

**T.C.**  
**BALIKESİR ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**  
**ORTAÖĞRETİM FEN VE MATEMATİK ALANLAR EĞİTİMİ**  
**ANABİLİM DALI**  
**FİZİK EĞİTİMİ**



**ELEKTROGAME EĞİTSEL OYUNUNUN TASARLANIP**  
**GELİŞTİRİLEREK BASİT ELEKTRİK DEVRELERİ**  
**KONUSUNDA BİLİŞSEL VE DUYUŞSAL DEĞİŞKENLERE**  
**ETKİSİNİN İNCELENMESİ**

**DOKTORA TEZİ**

**MEHMET EMİN KORKUSUZ**

**BALIKESİR, HAZİRAN - 2012**

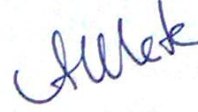
## KABUL VE ONAY SAYFASI

Mehmet Emin KORKUSUZ tarafından hazırlanan "ELEKTROGAME EĞİTSEL OYUNUNUN TASARLANIP GELİŞTİRİLEREK BASİT ELEKTRİK DEVRELERİ KONUSUNDA BİLİŞSEL VE DUYUSAL DEĞİŞKENLERE ETKİSİNİN İNCELENMESİ" adlı tez çalışmasının savunma sınavı 28.06.2012 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği / oy çokluğu ile Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Ortaöğretim Fen ve Matematik Alanlar Eğitimi Anabilim Dalı Fizik Eğitimi Doktora Tezi olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Danışman  
Yrd. Doç. Dr. Ayşen KARAMETE



Üye  
Doç. Dr. Hakan TÜZÜN



Üye  
Doç. Dr. Hüseyin KÜÇÜKÖZER



Üye  
Doç. Dr. M. Sabri KOCAKÜLAH



Üye  
Doç. Dr. Gamze SEZGİN SELÇUK



Jüri üyeleri tarafından kabul edilmiş olan bu tez BAÜ Fen Bilimleri Enstitüsü Yönetim Kurulunca onanmıştır.

Fen Bilimleri Enstitüsü Müdürü

Prof. Dr. Hilmi NAMLI

## ÖZET

### **ELEKTROGAME EĞİTSEL OYUNUNUN TASARLANIP GELİŞTİRİLEREK BASİT ELEKTRİK DEVRELERİ KONUSUNDA BİLİŞSEL VE DUYUŞSAL DEĞİŞKENLERE ETKİSİNİN İNCELENMESİ DOKTORA TEZİ**

**MEHMET EMİN KORKUSUZ**

**BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ  
ORTAÖĞRETİM FEN VE MATEMATİK ALANLAR EĞİTİMİ  
ANABİLİM DALI  
FİZİK EĞİTİMİ**

**(TEZ DANIŞMANI: YRD. DOÇ. DR. AYŞEN KARAMETE)**

**BALIKESİR, HAZİRAN - 2012**

Bu çalışmada, 9. sınıf fizik dersi “Basit Elektrik Devreleri” konusunu içeren Devasa Çok Kullanıcı Çevrim İçi Rol Yapma Oyunu (MMORPG - Massively Multiplayer Online Role Playing Game) türünde bir eğitsel oyun geliştirilmesi ve geliştirilen oyunun öğrencilerin bilgisayar tutumuna, fizik tutumuna, fizik başarısına etkisinin incelenmesi ve öğrencilerin oyun hakkındaki görüşlerinin belirlenmesi amaçlanmıştır.

elektroGame adı verilen bu eğitsel oyunun geliştirilmesi esnasında Kiili (2005) Deneyimsel Oyun Modeli temel alınmıştır. Açık kaynak kodlu MMORPG türü oyunlar incelenmiş, basit elektrik devreleri konusunun öğrenme hedefleri belirlenmiş ve tema oluşturulmuştur. “Mücadele”, “oyunsallık” ve “oynanabilirlik” bileşenleri göz önüne alınarak oyuncuların (öğrencilerin) kendilerine verilen görevleri yerine getirirken yaratıcı fikirler üretip çözüme ulaşmaları amaçlanmıştır.

Devreleri analiz ederek çözümün doğruluğunu kontrol eden bir simülatör hazırlanarak eğitsel oyuna entegre edilmiştir. Oyun hakkında genel bilgilerin bulunduğu, oyuncuların iletişiminin sağlandığı ve başarı derecelerine göre sıralandığı bir web sitesi oluşturulmuştur. Ayrıca, bu web sitesine blog ve wiki eklenerek oyun geliştiricileri için de bir paylaşım ortamı oluşturulmuştur.

Çalışma nicel ve nitel araştırma yöntemlerinin birlikte kullanıldığı karma-yöntem ile yürütülmüştür. Nicel araştırma yöntemi olarak yarı deneysel desenlerden kontrol gruplu ön-test son-test deney modeli kullanılmıştır. Katılımcılar olarak Balıkesir ilinden bir Anadolu Lisesi ile bir Meslek Lisesi seçilmiş ve bu okullardan ikişer sınıf rastgele deney ve kontrol grubu olarak alınmıştır. Uygulama sonunda, deney grubundan rastgele seçilmiş sekiz öğrenci ile yarı yapılandırılmış görüşme yapılmıştır.

elektroGame’i oynayan öğrencilerin fizik ve bilgisayar tutumlarında anlamlı bir değişim görülmemiştir. Oyundaki başarı seviyesiyle fizik başarısı arasında istatistiksel olarak anlamlı bir ilişki bulunmamış ancak oyundaki başarı seviyesiyle kavramsal anlama düzeyi ve emin olma düzeyi arasında anlamlı ve pozitif yönde orta düzeyli ilişki bulunmuştur. Ayrıca oyundaki başarı seviyesi cinsiyet faktörü açısından erkek oyuncular lehine çıkmıştır. Öğrencilerin çoğu elektroGame ortamında bulunmaktan keyif aldıklarını ancak grafiklerin iyileştirilmesi gerektiğini belirtmiştir.

**ANAHTAR KELİMELEER:** MMORPG, basit elektrik devreleri, bilgisayar destekli eğitim, eğitsel bilgisayar oyunu, çevrim içi oyun

## **ABSTRACT**

### **DESIGN AND DEVELOPMENT OF THE EDUCATIONAL GAME NAMED 'ELECTROGAME' AND ANALYSIS OF ITS EFFECT ON THE COGNITIVE AND AFFECTIVE VARIABLES REGARDING THE TOPIC OF SIMPLE ELECTRICAL CIRCUITS**

**PH. D THESIS**

**MEHMET EMİN KORKUSUZ**

**BALIKESİR UNIVERSITY INSTITUTE OF SCIENCE**

**SECONDARY SCIENCE AND MATHEMATICS EDUCATION**

**PHYSICS EDUCATION**

**(SUPERVISOR: ASST. PROF. DR. AYŞEN KARAMETE )**

**BALIKESİR, JUNE 2012**

The objective of this study is to develop a “MMORPG - Massively Multiplayer Online Role Playing Game)” including “Basic Electrical Circuit” that is a course topic for 9th grade physics lesson and to analyse the effects of the game on students’ attitudes to computer and physics and success on physics, and is to ascertain student’s opinions on the game.

“Experiential Game Model” (Kiili, 2005) was predicated on the development of this educational game named as elektroGame. The games with open source codes such as MMORPG were examined; the learning objectives of the basic electrical circuit topic were determined and the theme was finalized. By considering the “challenge”, “gamefulness” and “playability” components, it was aimed that players (students) can come up with creative thinking and reach a solution while fulfilling the duties given to them.

The simulator checking the accuracy of the solution by analysing the circuits was formed and integrated to the game. A web site that presents information about the game, that provides the contact among players and that sorts them by their success was built. A blog and wiki were attached to the site so that a sharing environment for developers can be created.

The study was carried out with mixed-method including both quantitative and qualitative methods. Of quasi-experimental patterns, pretest-posttest design with control group was used as the quantitative research method. As for the sample, an Anatolian High School and a Medical Vocational High School were selected and two for each class from these schools were randomly taken to be experimental and control group. After the experiment, eight students, were applied semi-structured interview.

In the study, a statistically significant relationship between the degree of game success and physics success was not reached but it was observed that there was a positive and significant relationship at the middle level between the degree of game success and the level of conceptual understanding or the level of sureness. Regarding the gender factor, the degree of game success gives result in favour of male players. Most of the students stated that they had enjoyed being in the elektroGame platform however the design of the game needed to be improved.

**KEYWORDS:** MMORPG, simple electrical circuits, computer aided education, educational computer game, online games



# İÇİNDEKİLER

Sayfa

<b>ÖZET</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>İÇİNDEKİLER</b> .....	<b>iii</b>
<b>ŞEKİL LİSTESİ</b> .....	<b>vi</b>
<b>TABLO LİSTESİ</b> .....	<b>x</b>
<b>ÖNSÖZ</b> .....	<b>xii</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1 Araştırmanın Amacı .....	2
1.2 Araştırmanın Önemi.....	2
1.3 Araştırma Soruları.....	6
1.4 Varsayımlar .....	8
1.5 Sınırlılıklar .....	9
1.6 Tanımlar ve Kısaltmalar .....	9
<b>2. LİTERATÜR</b> .....	<b>12</b>
2.1 Öğrenme ve Öğrenme Kuramları.....	12
2.2 Öğrenme Ortamları .....	19
2.3 Bilgisayar Oyunları .....	21
2.3.1 Oyun Türleri .....	23
2.3.2 Oyunların Özellikleri .....	25
2.3.3 Eğitsel Bilgisayar Oyunları.....	28
2.4 Çevrim İçi Oyun Tabanlı Öğrenme Ortamları.....	32
2.5 Eğitsel Bilgisayar Oyunu Geliştirilmesi .....	36
2.5.1 EFM: Eğitsel Oyun Tasarımı için Bir Model .....	38
2.5.2 FIDGE Modeli .....	42
2.5.3 Oyun Nesnesi Modeli .....	43
2.5.4 Dijital Oyun Tabanlı Öğrenme Modeli.....	45
2.5.5 Oyun Meydana Getirme Modeli .....	46
2.5.6 Deneysel Oyun Modeli .....	48
2.6 Basit Elektrik Devrelerinin Öğretimi.....	54
<b>3. YÖNTEM</b> .....	<b>57</b>
3.1 Araştırmanın Örnekleme .....	57
3.2 Verilerin Toplanması .....	58
3.2.1 Fizik Dersine Yönelik Tutum Ölçeği.....	58
3.2.2 Bilgisayara Yönelik Tutum Ölçeği .....	59
3.2.3 Üç Basamaklı Başarı Testi.....	59
3.2.4 Yarı Yapılandırılmış Görüşme.....	60
3.3 Deneysel Oyun Modeli ve elektroGame.....	60
3.4 Uygulama Süreci.....	65
3.4.1 Hazırlık Süreci .....	67
3.4.2 Uygulama Süreci.....	67
3.5 Verilerin Çözümlemesi .....	69
<b>4. ELEKTROGAME</b> .....	<b>71</b>
4.1 MMORPG Geliştirme .....	71
4.1.1 Oyun Mimarisi .....	72
4.1.2 Sunucu Mimarisi .....	74

4.1.3	İstemci Mimarisi .....	75
4.2	elektroGame Oyununun Özellikleri .....	75
4.2.1	elektroGame Oyununun Eğlenceye Yönelik Özellikleri .....	75
4.2.1.1	Tema.....	76
4.2.1.2	Oyuncu (Char).....	77
4.2.1.3	Robotlar.....	82
4.2.1.4	Bilgeler.....	83
4.2.1.5	Nesneler.....	85
4.2.1.6	Dünyalar.....	91
4.2.2	elektroGame Oyununun Öğretime Yönelik Özellikleri.....	91
4.2.2.1	Sunu Penceresi .....	92
4.2.2.2	Test Penceresi.....	93
4.2.2.3	Devre Penceresi.....	98
4.2.2.4	Görev Penceresi .....	100
4.2.2.5	Özel İçerikli Pencereleer.....	101
4.3	elektroGame Oyununun Geliştirilme Süreci.....	102
4.3.1	İstemcinin Geliştirilmesi.....	102
4.3.1.1	İstemcide Kullanılan Kütüphaneler.....	103
4.3.1.2	Sunu Penceresinin Geliştirilmesi .....	105
4.3.1.3	Test Penceresinin Geliştirilmesi.....	108
4.3.1.4	Elektrik Devre Elemanlarının Geliştirilmesi.....	114
4.3.1.5	Devre Penceresinin Geliştirilmesi.....	118
4.3.1.6	Görev Penceresinin Geliştirilmesi.....	132
4.3.1.7	Özel İçerikli Pencereleerinin Geliştirilmesi .....	135
4.3.1.8	Dünya Yapımı .....	136
4.3.1.9	Parçacık Efektleri .....	141
4.3.1.10	Nesne özellikleri.....	143
4.3.1.11	Bilge Özellikleri .....	147
4.3.1.12	Robot Özellikleri.....	149
4.3.2	Sunucu .....	153
4.3.2.1	Login sunucu .....	155
4.3.2.2	Char Sunucu .....	157
4.3.2.3	Map Sunucu .....	158
4.3.2.4	eAthena Script.....	162
4.3.2.5	Sunu Bilgeleri .....	165
4.3.2.6	Test bilgeleri.....	167
4.3.2.7	Devre Bilgeleri .....	178
4.3.2.8	Görev Bilgisayarı Bilgeleri .....	182
4.3.3	WEB Sitesinin Geliştirilmesi.....	183
4.3.3.1	Wiki.....	189
4.3.3.2	Blog.....	191
4.3.3.3	Yönetim Paneli.....	193
4.3.4	Yardımcı Programlar .....	198
4.3.4.1	Devre Tasarım Programı .....	198
4.3.4.2	Test Tasarım Programı .....	199
4.3.4.3	Sunucu Kontrol Programı .....	201
<b>5.</b>	<b>BULGULAR .....</b>	<b>202</b>
5.1	Uygulama Öncesinde Deney ve Kontrol Grubu Öğrencileri Arasındaki Farklılıklar.....	202
5.2	Uygulama Öncesi ve Sonrasında Değişim Var Mıdır? .....	203

5.3	Deney Grubu Öğrencilerinde Okul Türünün Etkisi.....	210
5.4	Deney Grubu Puanlarındaki İlişkiler .....	212
5.5	Cinsiyetin Deney Grubu Öğrencilerine Etkileri .....	213
5.6	Öğrenci Görüşleri.....	215
5.6.1	Bölüm I: Önceki deneyimler.....	215
5.6.2	Bölüm II: elektroGame hakkındaki öğrenci görüşleri .....	216
<b>6.</b>	<b>SONUÇ, TARTIŞMA VE ÖNERİLER.....</b>	<b>222</b>
6.1	Sonuçlar .....	222
6.2	TARTIŞMA .....	224
6.3	Öneriler .....	225
6.3.1	Geliştirmeye İlgili Öneriler.....	225
6.3.2	Uygulamaya İlgili Öneriler.....	226
<b>7.</b>	<b>KAYNAKLAR.....</b>	<b>227</b>
<b>8.</b>	<b>EKLER.....</b>	<b>239</b>
	EK A – Fizik Tutum Ölçeği .....	239
	EK B – Bilgisayar Tutum Ölçeği .....	241
	EK C – Üç Basamaklı Test.....	243
	EK D – Görüşme Formu .....	254

## ŞEKİL LİSTESİ

### Sayfa

Şekil 1.1: Türkiye'de oyun oynama oranlarının yaşa göre dağılımı. ....	3
Şekil 1.2: Türkiye'deki yaş gruplarının oyun oynama oranları. ....	4
Şekil 2.1: 1971 yılında geliştirilen Oregon Trail oyunu. ....	30
Şekil 2.2: Girdi-süreç-çıktı oyun modeli. ....	37
Şekil 2.3: İsmail İpek, Eğitsel Oyun Yazılımlarının Genel Yapısı.....	38
Şekil 2.4: Motivasyon, akış deneyimi, etkili öğrenme ortamı ve eğitsel oyun arasındaki bağlantı.....	41
Şekil 2.5: Eğitsel oyun tasarımı için EFM modeli.....	41
Şekil 2.6: Oyun Nesnesi Modeli (Amory ve Seagram 2003; Akt. Amory 2007).....	43
Şekil 2.7: Oyun Nesnesi Modeli II (Amory, 2007). ....	44
Şekil 2.8: Eğitsel tarih oyunlarının tasarımı için DGBL Modelinde önerilen bileşenler. ....	45
Şekil 2.9: DGBL Modeli. ....	46
Şekil 2.10: GAM'a göre eğitsel oyunun etkinliklerini oluşturan iki temel faktör. ....	47
Şekil 2.11: Oyun içindeki sahnelerin oluşturulması sürecindeki bileşenler. ....	47
Şekil 2.12: Sahneyi oluşturan üç alan ve alt öğeleri. ....	48
Şekil 2.13: Deneyimsel Oyun Modeli (Kiili, 2005a).....	50
Şekil 2.14: Deneyimsel Oyun Modelinin ikinci sürümü (Kiili, 2005b). ....	51
Şekil 3.1: elektroGame tasarım geliştirme süreci. ....	65
Şekil 3.2: Uygulama süreci. ....	66
Şekil 3.3: O2 okulundan bir görüntü. ....	68
Şekil 3.4: O1 okulundan bir görüntü. ....	69
Şekil 4.1: Kız oyuncu sprite'ının tüm kareleri. ....	78
Şekil 4.2: Erkek oyuncu sprite'ı. ....	79
Şekil 4.3: Char'ın durumunu gösteren durum penceresi. ....	81
Şekil 4.4: Oyunda kullanılan bir robotun sprite'ı. ....	83
Şekil 4.5: elektroGame'de kullanılan robotlar. ....	83
Şekil 4.6: Alışverişte kullanılan bir bilgenin açtığı pencere.....	84
Şekil 4.7: Oyuncunun nesnelerini gösteren heybe penceresi. ....	85
Şekil 4.8: Silah nesnesi örnekleri. ....	86
Şekil 4.9: Plazma silahlarında kullanılan plazma çeşitleri. ....	86
Şekil 4.10: elektroGame'de kullanılan bir kıyafet nesnesinin grafikleri. ....	87
Şekil 4.11: Ekipman penceresi. ....	89
Şekil 4.12: Basit elektrik devrelerinde kullanılan nesnelere. ....	89
Şekil 4.13: Diğer nesnelerin listesi .....	90
Şekil 4.14: Sunu penceresi.....	92
Şekil 4.15: Bir sorulu testlerde isteğe bağlı olarak eklenebilen açıklama penceresi.....	93
Şekil 4.16: Çoktan seçmeli bir soru penceresi.....	94
Şekil 4.17: Çoktan seçmeli bir soruluk testin sonuç penceresi.....	94
Şekil 4.18: Onay kutulu bir soru penceresi.....	95
Şekil 4.19: Onay kutulu tek sorulu testin rapor penceresi. ....	96
Şekil 4.20: Çoktan seçmeli birden fazla sorulu testin onay penceresi.....	96



Şekil 4.21: Çoktan seçmeli birden fazla sorulu test ekranı içeriği. ....	97
Şekil 4.22: Çok sorulu testin rapor ekranı. ....	98
Şekil 4.23: Devre penceresinin genel görünümü. ....	99
Şekil 4.24: Elektrik devrelerinde kullanılabilen elemanlar. ....	99
Şekil 4.25: Görev penceresi. ....	100
Şekil 4.26: Direnç renkleri penceresi. ....	101
Şekil 4.27: Sunu penceresinin özelliklerini bildiren tanımlayıcı XML. ....	106
Şekil 4.28: Slayt bilgisini ve içeriğini tanımlayan XML. ....	106
Şekil 4.29: Image nesnesinin elektroWidget sınıfında tanımlanan yapısı ve vektör tipi. ....	107
Şekil 4.30: Test penceresinde açılış mesajını oluşturan XML. ....	109
Şekil 4.31: Test penceresinin ve testin özelliklerini tanımlayan <starter> tag'ı. ....	109
Şekil 4.32: Test penceresinde soru oluşturulmasını sağlayan <question> örneği. ....	110
Şekil 4.33: Testin sonunda barajı aşamayan oyuncuya sunucu tarafından gönderilen rapor örneği. ....	111
Şekil 4.34: Testin sonunda barajı aşan oyuncuya sunucu tarafından gönderilen rapor örneği. ....	111
Şekil 4.35: Test aşamalarını kontrol etmekte kullanılan durumlar. ....	114
Şekil 4.36: Node sınıfının soy ağacı. ....	115
Şekil 4.37: Bir Component nesnesinin alabileceği durumlar. ....	116
Şekil 4.38: Component sınıftan türeyen devre elemanlarının listesi. ....	116
Şekil 4.39: Component sınıfının çizilmesinden sorumlu draw fonksiyonu. ....	118
Şekil 4.40: Devre penceresinde <b>açıklama</b> mesajı tanımlayan XML veri. ....	119
Şekil 4.41: Devre oluşturmak üzere gönderilen XML kodu. ....	121
Şekil 4.42: Devre penceresinin alabileceği durumlar. ....	122
Şekil 4.43: Devrenin analiz edilebilmesi için gerekli fonksiyonları çağıran fonksiyon. ....	123
Şekil 4.44: CircuitWindow sınıfının devre analizini açıklamak için kullanılan örnek devre. ....	123
Şekil 4.45: Birbirinin aynısı olan ilmekleri temizleyen kod. ....	125
Şekil 4.46: Birbirini kapsayan ilmekleri temizleyen kod. ....	126
Şekil 4.47: Matrislerin GSL kullanılarak çözülmesi. ....	127
Şekil 4.48: İlmek akımlarının node'lara aktarılması. ....	128
Şekil 4.49: Component'lerin silinmesini sağlayan döngü. ....	129
Şekil 4.50: Node'ların silinmesini sağlayan döngü. ....	129
Şekil 4.51: Sürüklenen node'un diğer node'larla çakışma kontrolü. ....	130
Şekil 4.52: Heybe penceresinde nesneyi kullan düğmesine basıldığında devre penceresine aktaran kod. ....	131
Şekil 4.53: Devre penceresinde kullanılan nesnelere sunucuya ileten sendUsedItem() fonksiyonu. ....	131
Şekil 4.54: Görev penceresine gelen XML veri. ....	132
Şekil 4.55: Ana görevi oluşturan yapı ve konteyner. ....	133
Şekil 4.56: Alt görevi oluşturan yapı ve konteyner. ....	134
Şekil 4.57: Mevcut görevlerin silinmesini sağlayan clearMission() fonksiyonu. ....	134
Şekil 4.58: Bilgeye tıkladığı zaman oluşturulan pencere. ....	136
Şekil 4.59: Dünya yapımında kullanılan sprite. ....	137
Şekil 4.60: TME (QT Editon) programı. ....	138

<b>Şekil 4.61:</b> Oluşturulacak olan dünyanın boyutlarının ve bir karenin boyutlarının tanımlandığı pencere. ....	138
<b>Şekil 4.62:</b> Bir dünya da bulunması gereken katmanlar. ....	139
<b>Şekil 4.63:</b> Tileset ekleme penceresi. ....	139
<b>Şekil 4.64:</b> TME programıyla tasarlanmış Merkez Dünyası. ....	140
<b>Şekil 4.65:</b> TMX uzantılı dosyaya eklenen parçacık efekt kodu. ....	141
<b>Şekil 4.66:</b> Bilgelerin efekt eklemesi için hazırlanmış fonksiyon. ....	142
<b>Şekil 4.67:</b> effects.xml dosyasının ilk satırları. ....	143
<b>Şekil 4.68:</b> Parçacık oluşturmak için kullanılan XML biçimli verinin genel yapısı. ....	143
<b>Şekil 4.69:</b> Base sprite'ların item olarak tanımlanması. ....	144
<b>Şekil 4.70:</b> Bir item tag'ının özellikleri. ....	144
<b>Şekil 4.71:</b> npcs.xml dosyasına örnek. ....	148
<b>Şekil 4.72:</b> elektroGame'de kullanılan bilgiler. ....	148
<b>Şekil 4.73:</b> Sprite üzerindeki npc bilgilerini tanımlayan veri. ....	149
<b>Şekil 4.74:</b> Monster.xml dosyasında kullanılan yapı. ....	150
<b>Şekil 4.75:</b> Virtual Host tanımlamaları. ....	155
<b>Şekil 4.76:</b> Robot ve geçit noktası ekleyen kod. ....	159
<b>Şekil 4.77:</b> Eğitim Dünyası'nın yapısal diyagramı. ....	161
<b>Şekil 4.78:</b> Döngü olarak kullanılan yapı. ....	163
<b>Şekil 4.79:</b> Sunu bilgilerinin akış diyagramı. ....	165
<b>Şekil 4.80:</b> Sunu özelliklerini gönderen script. ....	166
<b>Şekil 4.81:</b> Bir slaydı oluşturan script. ....	166
<b>Şekil 4.82:</b> Slaytlar arasında geçiş yapan menü script'i. ....	167
<b>Şekil 4.83:</b> Tek sorulu testin diyagramı. ....	168
<b>Şekil 4.84:</b> Tek sorulu bir testin başlık script'i. ....	169
<b>Şekil 4.85:</b> Tek sorulu bir testin soru oluşturma bölümü. ....	169
<b>Şekil 4.86:</b> Tek sorulu bir testte seçenekler menüsü ve değerlendirme bölümü. ....	170
<b>Şekil 4.87:</b> Tek sorulu bir testin doğru cevap script'i. ....	171
<b>Şekil 4.88:</b> Çoktan seçmeli bir testin yanlış cevap script'i. ....	171
<b>Şekil 4.89:</b> Onay kutulu bir sorunun diyagramı. ....	172
<b>Şekil 4.90:</b> Onay kutularının başlangıçta işaretli olmasını sağlayan ve doğru cevapları global_reg_value tablosuna kaydeden kod. ....	172
<b>Şekil 4.91:</b> Seçenek menüsü ve onay kutusu değişkeninin atanması. ....	173
<b>Şekil 4.92:</b> Onay kutulu soruların değerlendirilmesi. ....	174
<b>Şekil 4.93:</b> Çok sorudan oluşan bir testin diyagramı. ....	175
<b>Şekil 4.94:</b> Çok sorulu testte soru sonunda kullanılan menü. ....	176
<b>Şekil 4.95:</b> Çok sorulu testin seçeneklerinde çalışan kod. ....	176
<b>Şekil 4.96:</b> Doğru cevap tanımlama. ....	177
<b>Şekil 4.97:</b> Doğru ve boş cevapların hesaplanması. ....	177
<b>Şekil 4.98:</b> Doğru ve boş cevapların hesaplanması. ....	177
<b>Şekil 4.99:</b> Doğru ve boş cevapların hesaplanması. ....	178
<b>Şekil 4.100:</b> Devre bilgilerinin verisini oluşturan script'in diyagramı. ....	179
<b>Şekil 4.101:</b> Devreye bir bileşen yerleştirmek için kullanılan component fonksiyonunun kullanımı. ....	179
<b>Şekil 4.102:</b> component fonksiyonu. ....	180
<b>Şekil 4.103:</b> Node fonksiyonu çağırışı. ....	180
<b>Şekil 4.104:</b> Node fonksiyonu ve parametrelerinin açıklaması. ....	181
<b>Şekil 4.105:</b> connect fonksiyonunun çağırılması. ....	181

Şekil 4.106: connect fonksiyonu ve parametrelerinin açıklaması. ....	181
Şekil 4.107: circuitcondition tag'ını oluşturan kod. ....	182
Şekil 4.108: Görev bilgisayarına bir görev ve bir alt görev ekleyen kod. ....	182
Şekil 4.109: Bir görevin yerine getirilip getirilmediğini inceleyen kod bloğu. ....	183
Şekil 4.110: Doctrine ORM ile hazırlanmış olan oyuncu puanları tablosu. ....	185
Şekil 4.111: Login ve Char tablolarının gerekli alanları. ....	186
Şekil 4.112: login tablosunun Model bölümünde sınıf tanımlaması. ....	186
Şekil 4.113: Oyuncular.php dosyasındaki login sınıfının bir bölümü. ....	187
Şekil 4.114: Login sınıfı <i>setUp()</i> fonksiyonu. ....	187
Şekil 4.115: charview sınıfı <i>setUp()</i> fonksiyonu. ....	188
Şekil 4.116: IndexController'da yer alan <i>zirvedekilerAction()</i> fonksiyonu. ....	188
Şekil 4.117: Karakter verilerinin web sitesinde gösterilmesi. ....	189
Şekil 4.118: elektroGame wiki sayfası. ....	190
Şekil 4.119: <a href="http://wiki.elektrogame.net/index.php/Deneme">http://wiki.elektrogame.net/index.php/Deneme</a> sayfası. ....	191
Şekil 4.120: elektroBlog'un bir görüntüsü. ....	192
Şekil 4.121: elektroGame'in ana sayfasında blog'taki haberlerin gösterimi. ....	192
Şekil 4.122: Haberlerin ana sayfada gösterilmesi için veri tabanından verileri alan kod. ....	193
Şekil 4.123: Yönetim paneli kullanıcı giriş ekranı. ....	193
Şekil 4.124: Yönetim panelindeki sohbet kaydı bölümü. ....	194
Şekil 4.125: Görev Listele bölümünde öğrenci seçimi. ....	194
Şekil 4.126: Öğrencinin genel görevlerdeki durumu. ....	195
Şekil 4.127: Öğrencinin testlerdeki durumunun gösterimi. ....	195
Şekil 4.128: Okul yönetim bölümü. ....	196
Şekil 4.129: Oyuncu yönetimi bölümü. ....	197
Şekil 4.130: Yeni yönetici-öğretmen-oyuncu ekleme penceresi. ....	197
Şekil 4.131: Devre tasarım programının genel görünümü. ....	199
Şekil 4.132: Test tasarım penceresinin genel görünüşü. ....	200
Şekil 4.133: Bilgenin genel özelliklerini ayarlayan panel. ....	200
Şekil 4.134: Testin özelliklerini ayarlayan panel. ....	201
Şekil 4.135: Sunucu kontrol programı. ....	201
Şekil 5.1: Başarı testi ön-test son-test puan ortalamaları. ....	205
Şekil 5.2: Kavramsal anlama düzeyi ön-test son-test puan ortalamaları. ....	206
Şekil 5.3: Emin olma düzeyi ön-test son-test puan ortalamaları. ....	207
Şekil 5.4: Fizik tutumu deney ve kontrol gruplarının ön-test son-test ortalama puanları. ....	208
Şekil 5.5: Bilgisayar tutumu, deney ve kontrol gruplarının ön-test son-test ortalama puanları. ....	210

## TABLO LİSTESİ

### Sayfa

<b>Tablo 1.1:</b> Öğrencilerin bilgisayar kullanım amaçları. ....	3
<b>Tablo 2.1:</b> Öğrenme kuramlarının listesi. ....	13
<b>Tablo 2.2:</b> Tennis for Two ve Spacewar! oyunlarının görüntüsü. ....	21
<b>Tablo 2.3:</b> Modern bir tenis oyunu ile Tennis for Two oyununun karşılaştırılması. ....	22
<b>Tablo 2.4:</b> ARCS Motivasyon Modeli Stratejileri. ....	39
<b>Tablo 2.5:</b> EGM öğelerinin açıklaması. ....	51
<b>Tablo 2.6:</b> Tasarım döngüsünün aşamalarının özeti.....	53
<b>Tablo 3.1:</b> Örneklemenin özellikleri. ....	58
<b>Tablo 3.2:</b> Görüşme yapılan öğrenci bilgileri. ....	60
<b>Tablo 4.1:</b> Kıyafetlerin gruplandırılmış listesi. ....	88
<b>Tablo 4.2:</b> Sunu penceresini oluşturmak kullanılan XML yapısının tag'ları...	107
<b>Tablo 4.3:</b> Test XML'de kullanılan tag'lar. ....	112
<b>Tablo 4.4:</b> Testteki her bir soru için kullanılan vektörler ve görevleri. ....	113
<b>Tablo 4.5:</b> Switch sınıfının sprite setleri. ....	117
<b>Tablo 4.6:</b> Devre penceresinde basit elektrik devresi oluşturabilmek için sunucu tarafından gönderilen XML dosyasının tag'ları ve bu tag'ların özellikleri. ....	119
<b>Tablo 4.7:</b> 101 id'li Node'un komşu listesi. ....	124
<b>Tablo 4.8:</b> Tekrarlı ve uzun ilmekler temizlendikten sonra kalan ilmekler. ....	126
<b>Tablo 4.9:</b> Direnç ve pillerin oluşturduğu matrisler. ....	127
<b>Tablo 4.10:</b> Görev penceresini tanımlayan XML tag'lar ve açıklamaları. ....	133
<b>Tablo 4.11:</b> Bir pencerenin sahip olması gereken 3 temel fonksiyon. ....	135
<b>Tablo 4.12:</b> item.xml'de kullanılan tag'lar ve özellikleri.....	145
<b>Tablo 4.13:</b> item_db.txt dosyasında tanımlanan bir nesneye ait özellikler. ....	146
<b>Tablo 4.14:</b> Monster.xml'de kullanılan tag'lar ve özellikleri. ....	151
<b>Tablo 4.15:</b> mob_db.txt özellikleri. ....	152
<b>Tablo 4.16:</b> elektroGame sunucusunun donanımsal özellikleri. ....	154
<b>Tablo 4.17:</b> Char sunucusunun önemli ayar parametreleri. ....	157
<b>Tablo 4.18:</b> Sık kullanılan eAthena fonksiyonları. ....	164
<b>Tablo 5.1:</b> Deney ve kontrol grubu öğrencilerinin ön-test sonuçlarının karşılaştırılması. ....	202
<b>Tablo 5.2:</b> Başarı testi ortalama ve standart sapma değerleri.....	204
<b>Tablo 5.3:</b> Deney kontrol grubu kavramsal anlama düzeyi ön-test son-test için iki faktörlü ANOVA sonuçları. ....	204
<b>Tablo 5.4:</b> Kavramsal anlama düzeyi ortalama ve standart sapma değerleri...	205
<b>Tablo 5.5:</b> Deney kontrol grubu kavramsal anlama ön-test son-test iki faktörlü ANOVA sonuçları. ....	205
<b>Tablo 5.6:</b> Emin olma düzeyi ortalama ve standart sapma değerleri. ....	206
<b>Tablo 5.7:</b> Deney kontrol grubu emin olma düzeyi ön-test son-test iki faktörlü ANOVA sonuçları. ....	207
<b>Tablo 5.8:</b> Fizik tutum ölçeği ön-test son-test ortalamaları ve standart sapma değerleri. ....	207
<b>Tablo 5.9:</b> Deney kontrol grubu fizik tutumu ön-test son-test iki faktörlü ANOVA sonuçları.....	208



<b>Tablo 5.10:</b> Bilgisayar tutum ölçeği ön-test son-test ortalamaları ve standart sapma değerleri.....	209
<b>Tablo 5.11:</b> Deney kontrol grubu bilgisayar tutumları ön-test son-test iki faktörlü ANOVA sonuçları.....	209
<b>Tablo 5.12:</b> Bilgisayar tutumu değişim puanlarının okul türlerine göre t-testi sonuçları.....	210
<b>Tablo 5.13:</b> Fizik tutumu değişim puanlarının okul türlerine göre t-testi sonuçları.....	211
<b>Tablo 5.14:</b> Başarı testi değişim puanlarının okul türlerine göre t-testi sonuçları.....	211
<b>Tablo 5.15:</b> Kavramsal anlama değişim puanlarının okul türlerine göre t-testi sonuçları.....	211
<b>Tablo 5.16:</b> Emin olma düzeyi değişim puanlarının okul türlerine göre t-testi sonuçları.....	211
<b>Tablo 5.17:</b> elektroGame seviyelerinin okul türlerine göre t-testi sonuçları....	212
<b>Tablo 5.18:</b> elektroGame seviyeleri ile değişim puanları arasındaki ilişki.....	213
<b>Tablo 5.19:</b> Cinsiyetin elektroGame başarısına etkisinin t-testi sonuçları.....	214
<b>Tablo 5.20:</b> Cinsiyetin fizik başarısına etkisinin t-testi sonuçları.....	214
<b>Tablo 5.21:</b> Cinsiyetin kavramsal anlama düzeyine etkisinin t-testi sonuçları.....	214
<b>Tablo 5.22:</b> Cinsiyetin emin olma düzeyine etkisinin t-testi sonuçları.....	215
<b>Tablo 5.23:</b> elektroGame hakkındaki izlenimlerin tema ve frekansları.....	216
<b>Tablo 5.24:</b> Oyunda hoşça giden temalar ve frekanslar.....	217
<b>Tablo 5.25:</b> elektroGame'in olumsuz yönüne ilişkin tema ve frekansları.....	218
<b>Tablo 5.26:</b> Diğer oyuncularla etkileşim konusunda temalar ve frekanslar....	218
<b>Tablo 5.27:</b> Rekabet hissini artırılmasına ilişkin temalar ve frekanslar.....	220
<b>Tablo 5.28:</b> elektroGame'in ders mi oyun mu olarak nitelendirilmesi.....	220
<b>Tablo 5.29:</b> Dersin elektroGame'de işlenmesine ilişkin tema ve frekanslar. ...	220

## ÖNSÖZ

Uzun ve meşakkatli bir süreç sonunda ortaya çıkan bu çalışmada, bana yardımcı olan ve yol gösteren pek çok değerli insan oldu. Hepsinin ismini tek tek anamamış olmanın burukluğunu hissederek, kendilerine en içten teşekkürlerimi sunuyorum.

Rehberliğiyle ve manevi desteğiyle bir danışmandan çok daha fazlasını ortaya koyan değerli danışmanım Yrd. Doç. Dr. Ayşen KARAMETE'ye, her aşamada değerli fikirlerine başvurduğum hocalarım ve tez izleme komitesi üyelerim Doç. Dr. Hakan TÜZÜN'e ve Doç. Dr. Hüseyin KÜÇÜKÖZER'e ayrıca jüri üyesi olarak katkılarını esirgemeyen Doç. Dr. M. Sabri KOCAKÜLAH'a ve Doç. Dr. Gamze SEZGİN SELÇUK'a teşekkürü bir borç bilirim.

Değerli görüşlerini ve katkılarını esirgemeyen Yrd. Doç. Dr. Serkan PERKMEN'e, her konuda destek olan Arş. Gör. Gürhan DURAK'a ve tüm bölüm arkadaşlarıma ayrıca teşekkür ederim.

elektroGame'in geliştirilmesi sırasında, önce öğrenmek sonrada üretmek için hafta sonlarını ekibin bir parçası olmaya ayıran, meraklı ve gayretli lisans öğrencilerime, özellikle öğrencim, meslektaşım ve dostum olarak çok şey paylaştığım Öğr. Gör. Hüseyin GÜNEŞ'e ve paylaşımcı The Mana World geliştirme ekibine teşekkür ederim.

Bu çalışma sırasında, dualarıyla her zaman yanımda hissettiğim annem ve babam Zeliha ve Selami KORKUSUZ'a, motivasyon kaynağım kardeşim Şermin KORKUSUZ'a ve onlara ait olması gereken zamanı kullanmamı anlayışla karşılayan eşim Nihal ve oğlum Burak'a destekleri, yardımları ve varlıkları için teşekkürden öte minnet borçluyum. Onlar olmasaydı bu çalışma olmazdı.

Balıkesir, 2012

Mehmet Emin KORKUSUZ

# 1. GİRİŞ

Yapılan arařtırmalar, öğrencilerin fizik derslerindeki motivasyonlarının ve başarılarının diđer derslere oranla daha düşük olduğunu göstermektedir (Duit, Vorsteinung ve Physiklernen, 1992). Düşük başarının ve düşük motivasyonun sebepleri irdelendiğinde, mevcut öğretim yöntemlerinin fizik dersinin öğretiminde kısmen yetersiz kaldığı söylenebilir. Bu durum, geleneksel öğretim anlayışının dışında, çağdaş öğretim yaklaşımlarının denenmesi ihtiyacını doğurmaktadır (Şen ve Çıldır, 2006).

Çağdaş öğretim yaklaşımlarının temelinde bilişsel yaklaşım vardır (Temiz, 2010). Bilişsel yaklaşıma göre, öğrenciler öğrenme sürecinde aktiftirler ve bilgiyi kendileri yapılandırır ya da oluştururlar (Ülgen, 2011). Öğrenim sürecinde öğrencinin aktif katılımını mümkün kılan yöntemlerden biri de bilgisayar destekli öğretimdir (Uşun, 2000; Demirci, 2003).

Bilgisayar Destekli Öğretim (BDÖ), kendi kendine öğrenme ilkelerinin bilgisayar teknolojisi ile birleşmesinden oluşmuş bir öğretim yöntemi olup öğrenme sürecinde bilgisayarın, sistemi tamamlayıcı ve güçlendirici bir öge olarak kullanılmasıdır (Uşun, 2000; Altın, 2009). Bilgisayar destekli öğretimin pek çok türü bulunmaktadır. Bu türlerinden biri de eğitsel bilgisayar oyunlarıdır (İpek, 2001). Özellikle öğrencinin motivasyonunu ön plana çıkartan eğitsel bilgisayar oyunları birçok dersin öğretiminde kullanılabilir.

Eğitsel bilgisayar oyunlarının alt yapısını bilgisayar oyunları oluşturmaktadır. Bilgisayar oyunları, bilgisayarların yaygınlaşmasıyla birlikte ortaya çıkmış ve zamanla popülerleşerek günlük yaşamının bir parçası haline gelmiştir. Bu süreçte oyun sektörü, Hollywood filmleriyle yarışan yapım bütçeleriyle ve daha da önemlisi gelirleriyle ciddi bir endüstriye dönüşmüştür. Örneğin “Call of Duty Black Ops” oyunu piyasaya sürüldüğü 9 Kasım 2010’dan 21 Aralık 2010 tarihine kadar 8,7 milyon adet satılmış ve 1 milyar dolar hâsılat elde etmiştir (Malkoç, 2010).

Oyuncuların bilgisayar oyunlarında geçirdikleri zaman, televizyon ve sinema gibi diğer etkinliklere ayırdıkları zamanın çok üzerindedir. “World of Warcraft” isimli oyunu oynayanların, oyunu oynamak için haftada ortalama 22,7 saatlik bir zaman dilimini ayırdıkları görülmüştür (Konuksal, Bölüm Sonu Canavarı, 2009). Yukarıda bahsedilen “Call of Duty Black Ops” isimli oyunu oynayanların ise 1 ay içinde oyunda geçirdikleri toplam süre 600 milyon saati aşmıştır. Ayrıca bu oyunu oynayanların sayısı, ABD’nin en çok izlenen *talk show* programları olan; “Jay Leno”, “Devid Letterman” ve “Jimmy Fallon”nun toplam izleyici sayısından fazladır (Süngü, 2010).

Gençlerin, toplam oyuncu kitlesi içinde büyük bir çoğunluğu oluşturması, bilgisayar oyunlarının eğitimde etkili şekilde kullanılabileceğine yönelik önemli bir göstergedir (BitDünyası, 2009). Nitekim eğitimciler, 1980-90’lı yıllarda bilgisayar oyunlarının kazandığı başarılarından sonra, bu oyunları eğitimde kullanabilecekleri bir araç olarak görmeye başlamışlardır (Mayer, Schustack, & Blanton, 1999).

### **1.1 Araştırmanın Amacı**

Bu çalışmanın amacı, eğitsel amaçlar için kullanmaya elverişli, macera ve eğlence unsurlarını barındıran çok kullanıcı bir çevrim içi rol yapma oyunu geliştirmek, geliştirilen oyuna ortaöğretim 9. sınıf fizik dersi programında yer alan “Elektrik ve Manyetizma” ünitesindeki “Basit Elektrik Devreleri” konusunu entegre etmek, bu oyunun öğrenci başarısına olan etkisini incelemek ve elektroGame oynayan öğrencilerin oyun hakkındaki görüşlerini belirlemektir.

Çalışmada ayrıca, öğrencilerin bilgisayar tutumları ve fizik tutumlarının geliştirilen oyunu oynamaları ile ilişkisi incelenmiştir.

### **1.2 Araştırmanın Önemi**

Günümüzde bilgisayarların hemen her eve girmesiyle birlikte bilgisayar kullanım oranları hızla artmıştır. Türkiye İstatistik Kurumu’nun (TÜİK) yaptığı araştırmaya göre; 16–24 yaş arası gençlerin bilgisayar kullanma oranı %65,2;



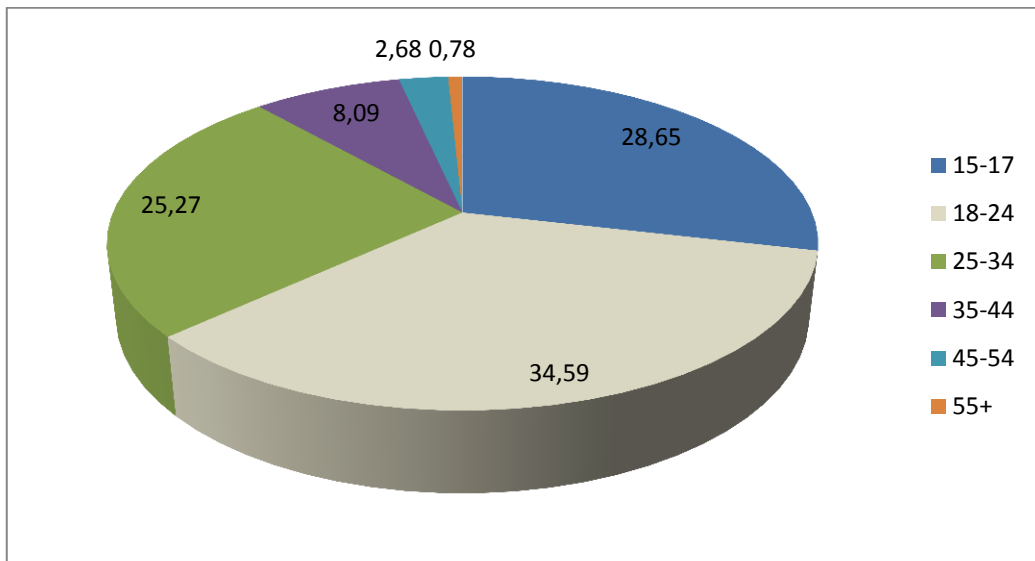
öğrencilerin ise %91,6'dır (TÜİK, 2011). Artık öğrenciler bilgisayar karşısında vakit geçirmeyi, televizyon seyretmek ya da kitap okumak gibi faaliyetlerden daha fazla tercih etmektedirler (Aksaçlıoğlu ve Yılmaz, 2007).

Bilgisayar başında vakit geçiren öğrenciler bu zaman zarfında bilgisayarı farklı amaçlar için kullanmaktadır. Deveci ve arkadaşlarının 2007 yılındaki çalışmasına göre öğrencilerin bilgisayarı kullanım amaçları ve dağılımları **Tablo 1.1**'de verilmiştir (Deveci ve Ark., 2007).

**Tablo 1.1:** Öğrencilerin bilgisayar kullanım amaçları.

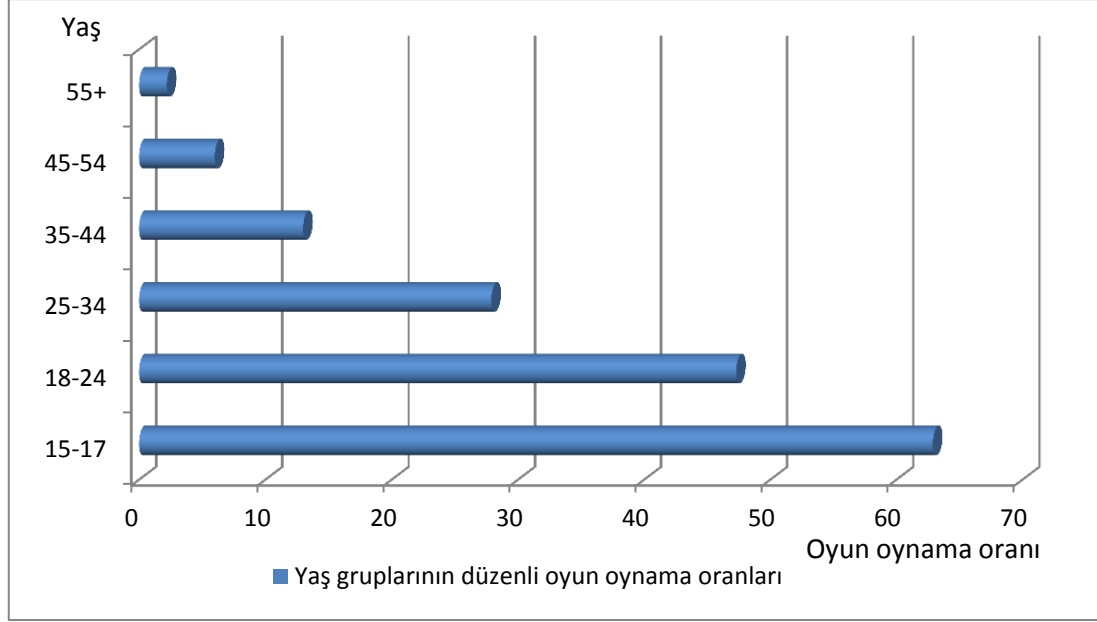
Kullanım Amacı	Öğrenci Sayısı	%
Oyun	2141	56.5
Ev Ödevi	1046	27.6
Ev Ödevi + Oyun	216	5.7
İnternet	161	4.2
Diğer	226	6.0
Toplam	<b>3790</b>	<b>100.0</b>

GfK isimli araştırma şirketinin Mayıs 2009'da ülkemizde yaptığı çalışmaya göre oyun oynama oranlarının yaşlara göre dağılımı **Şekil 1.1**'de verilmiştir. Buna göre oyun oynayan her 100 kişiden 63.24'ü 24 yaşın altındadır (Gfk, 2009).



**Şekil 1.1:** Türkiye'de oyun oynama oranlarının yaşa göre dağılımı.

Aynı arařtırmada, **Őekil 1.2**'de verilen yař gruplarının oyun oynama oranları da incelenmiř ve 15–24 yař arasındakilerin yarıdan fazlasının dzenli oyun oynadıđı gdrulmüřtür.



**Őekil 1.2:** Türkiye'deki yař gruplarının oyun oynama oranları.

Bu sonuçlar farklı ölkelerde yapılan çalıřmalarla da benzerlik göstermektedir. Örneđin, Malezya'da 236 öđrenci üzerinde yapılan bir çalıřmada oyun oynama oranının öđrenciler arasında ortalama %75,8 olduđu saptanmıřtır (erkek öđrencilerin %91,3'ü, kız öđrencilerin %54,1'i). Oyun oynamak için harcadıkları zamana bakıldıđında ise öđrencilerin haftada ortalama 8.47 saat bilgisayar oyunu oynadıkları belirlenmiřtir (Eow, Ali, Mahmud ve Baki, 2009).

Öđrencilerin bilgisayar oyunu oynama alışkanlıkları derinlemesine incelendiđinde, oyun türlerinin de oyun oynama alışkanlıklarına etkisi olduđu gdrulmektedir. Bilgisayar oyunu türleri arasında en fazla zaman harcanan ve oynama isteđi uyandıran tür MMORPG'dir (Shaw, 2010). Örneđin, MMORPG türünün en önemli oyunlarından biri olan World of Warcraft'ın 13,5 milyon aktif oyuncusu vardır ve oyuncular haftada 22,7 saatlerini bu oyunu oynamaya ayırmaktadır (Konuksal, Sayılarla World of Warcraft, 2009).

Arařtırmalarda da görüldüđu gibi öđrencilerin hayatında bilgisayarlar ve bilgisayar oyunları önemli bir yer tutmaktadır. Geçmiřte sokakta saklambaç oynayan

öğrenciler artık evlerinde, internet kafelerde bilgisayar oyunları oynamaya başlamıştır.

Öğrencilerin bilgisayar kullanmaya harcadıkları süre, oyunların bu süre içerisindeki büyük payı ve özellikle MMORPG türü oyunların etkisi göz önünde bulundurulduğunda, eğitsel açıdan yeni fırsatların ortaya çıktığı açık şekilde görülmektedir.

Literatürde MMORPG türünde oyunların eğitimde nasıl kullanılacağına yönelik birçok çalışma olmasına rağmen, geliştirilmiş çok az sayıda eğitsel MMORPG oyunu vardır ve bunlar da dar kapsamlı çalışmalardır (Güneş, 2010; Suh, Kim ve Kim, 2010). Genel olarak fizik eğitiminin, özel olarak basit elektrik devrelerinin konu edildiği bir MMORPG'ye ise bilgimiz dâhilinde rastlanmamıştır. Bu çalışmada ise farklı dersleri de içerebilecek geniş kapsamlı ve oynanabilir bir MMORPG hazırlanmaya çalışılmıştır. Böylece bu çalışma, fizik eğitimi alanında MMORPG'lerin kullanılması bakımından önem taşımaktadır.

Mevcut eğitsel MMORPG'ler araştırıldığında hiçbirinin açık kaynak kodlu olmadığı, şiddet öğeleri içerebildiği ya da ücretli olduğu bilgilerine ulaşılmıştır. MMORPG'lerin açık kaynak kodlu olmaması, bu alanda çalışmak isteyen araştırmacılar için kısıtlayıcı bir etkide bulunmaktadır. Bu nedenle elektroGame, GPL lisansı ile lisanslanmıştır, açık kaynak kodlu ve ücretsizdir. Ayrıca [www.elektrogame.net](http://www.elektrogame.net) sitesinde geliştiriciler için de kaynak ve bilgi paylaşımında bulunulmuştur. İstemci ve sunucu kaynak kodları ile oyunu geliştirmede kullanılan diğer materyallere web sitesinden erişilebilmektedir.

Ayrıca bu çalışma;

- Eğitsel bir MMORPG oyununun geliştirme aşamalarının ortaya konulması;
- İstemci-sunucu tabanlı bir oyunun mimarisinin oluşturulması,
- İki boyutlu haritaların geliştirilmesi,
- Oyun yazmaya uygun grafik kütüphanelerinin ve bunlar üzerindeki kullanıcı arabirimi kütüphaneleri kullanılarak yazılım geliştirilmesi,
- Oyun verilerinin veri tabanında saklanması ve gerektiğinde verilerin bir web sayfasında yayımlanması,

- Basit elektrik devrelerini çözen bir simülasyon oluşturulması

gibi aşamaların ortaya konması ile daha sonra benzer türde oyun geliştirmek isteyen araştırmacılara yardımcı olabilecektir.

elektroGame eğitsel oyununun ortamı, herhangi bir dersten bağımsız olarak geliştirildiği için farklı dersler ve konular ile ilgili etkinliklerin oyuna entegre edilebilmesi mümkündür.

Öğrenciler fizik derslerinde pek çok kavramsal sorun yaşamaktadır (Küçüközer, 2004; Uzunkavak, 2004). Özellikle basit elektrik devreleri konusunda ilköğretim birinci sınıftan başlayarak üniversite birinci sınıfa kadar her seviyede öğrencinin çeşitli kavram yanılgıları ve kavramları anlama düzeyinde yetersizlikleri vardır (Ateş ve Polat, 2005; Osborne, 1981; Shipstone ve Ark., 1988; Chambers ve Andre, 1997). Bunun yanında öğrencilerin fizik derslerine yönelik tutumları ve motivasyonları da çok olumlu değildir (Çokadar ve Külçe, 2008; Çakır, Şenler ve Taşkın, 2007; Kurnaz ve Yiğit, 2010; Alkan, 2006).

Geliştirilen eğitsel MMORPG elektroGame'e ortaöğretim 9. sınıf fizik dersi programında yer alan "Elektrik ve Manyetizma" ünitesindeki "Basit Elektrik Devreleri" konusu entegre edilmiştir. Fizik dersinin, başarısı en düşük derslerden birisi olması, bu alanda çok sayıda kavram yanılgısının bulunması, öğrencilerin fizik dersine karşı tutum ve motivasyonlarının diğer derslere oranla daha düşük olması gibi sebepler göz önüne alındığında; boş zamanlarının çoğunu bilgisayar başında oyun oynayarak geçiren öğrencilerin oyunlara yönelik ilgilerinin, fizik dersiyle ilgili etkinliklerin bulunduğu bu ortama yoğunlaştırılması önemlidir. Öğrencilerin oyunlara yönelik ilgisinin fizik dersine yönelik tutum, motivasyon ve başarılarını olumlu yönde etkilemesi umulmaktadır.

### **1.3 Araştırma Soruları**

1. Deney ve kontrol gurubu öğrencilerinin, uygulama öncesi ve sonrasında bilgisayar tutumu, fizik tutumu, başarı puanı, kavramsal anlama düzeyi ve emin olma düzeyi arasında anlamlı bir fark var mıdır?

1.1. Ön-test ve Son-test başarı puanları arasında anlamlı fark var mıdır?

- 1.2. Ön-test ve Son-test kavramsal anlama düzeyi puanları arasında anlamlı fark var mıdır?
- 1.3. Ön-test ve Son-test emin olma düzeyleri arasında anlamlı fark var mıdır?
- 1.4. Uygulama öncesi fizik tutumları ile uygulama sonrası fizik tutumları arasında anlamlı fark var mıdır?
- 1.5. Uygulama öncesi bilgisayar tutumları ile uygulama sonrası bilgisayar tutumları arasında anlamlı fark var mıdır?

2. Deney grubu öğrencilerinin, uygulama öncesi ve sonrasında, bilgisayar tutumu, fizik tutumu, başarı puanı, kavramsal anlama düzeyi, emin olma düzeylerindeki değişim puanları ve elektroGame oyunundaki seviyeleri arasında **okul türlerine** göre anlamlı bir fark var mıdır?

- 2.1. Deney grubu öğrencilerinin, uygulama öncesi ve sonrasında, bilgisayar tutumlarında okul türüne göre anlamlı bir fark var mıdır?
- 2.2. Deney grubu öğrencilerinin, uygulama öncesi ve sonrasında, fizik tutumlarında okul türüne göre anlamlı bir fark var mıdır?
- 2.3. Deney grubu öğrencilerinin, uygulama öncesi ve sonrasında, başarı testi değişim puanlarında okul türüne göre anlamlı bir fark var mıdır?
- 2.4. Deney grubu öğrencilerinin, uygulama öncesi ve sonrasında, kavramsal anlama düzeyi değişim puanlarında okul türüne göre anlamlı bir fark var mıdır?
- 2.5. Deney grubu öğrencilerinin, uygulama öncesi ve sonrasında, emin olma düzeyi değişim puanlarında okul türüne göre anlamlı bir fark var mıdır?
- 2.6. Deney grubu öğrencilerinin, elektroGame oyunundaki seviyelerinde okul türüne göre anlamlı bir fark var mıdır?

3. Deney grubu öğrencileri, elektroGame oyunundaki başarı düzeyleri ile bilgisayar tutumları, fizik tutumları, başarı testi, kavramsal anlama düzeyi ve emin olma düzeyi değişim puanları arasında istatistiksel olarak anlamlı bir ilişki var mıdır?

- 3.1. Deney grubu öğrencilerinin elektroGame oyunundaki başarı düzeyleri ile bilgisayar tutumları arasında anlamlı bir ilişki var mıdır?

- 3.2. Deney grubu öğrencilerinin, elektroGame oyunundaki başarı düzeyleri ile fizik tutumları arasında anlamlı bir ilişki var mıdır?
  - 3.3. Deney grubu öğrencilerinin, elektroGame oyunundaki başarı düzeyleri ile başarı testi değişim puanları arasında anlamlı bir ilişki var mıdır?
  - 3.4. Deney grubu öğrencilerinin, elektroGame oyunundaki başarı düzeyleri ile kavramsal anlama değişim düzeyi arasında anlamlı bir ilişki var mıdır?
  - 3.5. Deney grubu öğrencilerinin, elektroGame oyunundaki başarı düzeyleri ile emin olma düzeyi değişim puanları arasında anlamlı bir ilişki var mıdır?
4. Deney grubu öğrencilerinin elektroGame oyunundaki başarı düzeylerinin, başarı testi, kavramsal anlama ve emin olma düzeyi değişim puanları cinsiyete göre anlamlı farklılık göstermekte midir?
    - 4.1. Deney grubu öğrencilerinin, elektroGame oyunu başarısı cinsiyete göre anlamlı bir farklılık göstermekte midir?
    - 4.2. Deney grubu öğrencilerinin başarı testi değişim puanları, cinsiyetleri arasında anlamlı bir fark var mıdır?
    - 4.3. Deney grubu öğrencilerinin, kavramsal anlama düzeyleri değişim puanları, cinsiyete göre anlamlı bir farklılık göstermekte midir?
    - 4.4. Deney grubu öğrencilerinin, emin olma düzeyleri değişim puanları, cinsiyete göre anlamlı bir farklılık göstermekte midir?
  5. Hazırlanan elektroGame oyununun eğitimde kullanımına ilişkin öğrenci görüşleri nelerdir?

#### **1.4 Varsayımlar**

Bu araştırmada;

- Kullanılan ölçme araçlarının geçerlik ve güvenilirliği sağlanmış çalışmalar olduğundan araştırmanın amacına uygun olduğu,
- Katılımcıların veri toplama araçlarını içtenlikle yanıtladıkları,

- Çalışmaya katılan öğrencilerin bilgisayar okuryazarı olduğu,
- Öğrencilerin daha önce bilgisayar oyunu oynadıkları

kabul edilmiştir.

## 1.5 Sınırlılıklar

Bu araştırma;

- Balıkesir ilinde bir Anadolu Lisesi ile bir Meslek Lisesinde, 2009–2010 öğretim yılı 9. sınıfına devam etmekte olan 45 ve 44 öğrenci,
- MEB 9. sınıf fizik dersi programında yer alan “Elektrik ve Manyetizma” ünitesindeki “Basit Elektrik Devreleri” konusu,
- Oyunun oryantasyonu için ayrılan 1’er ders saati ve oyunun oynanması için ayrılan 4’er ders saati,
- Tüm öğrenciler derslerine normal akışında devam ederken deney grubu öğrencileriyle ayrıca bu uygulamanın yapılması

ile sınırlıdır.

## 1.6 Tanımlar ve Kısaltmalar

Bu bölümde, elektroGame oyunun geliştirilmesinde ve araştırmada kullanılan tanımlar ve kısaltmalar açıklanmıştır.

**RPG (Role Playing Game):** Oyuncuların oynadıkları karakterlerin rollerine bürünerek birleşik bir hikâye yarattıkları oyun türü.

**MMORPG (Massively Multiplayer Online Role Playing Game):** Devasa Çok Kullanıcı Çevrim İçi Rol Yapma Oyunu. RPG oyunların, çevrim içi tasarlanarak binlerce oyuncunun aynı anda oynamasının sağlandığı türü.

**TMW (The Mana World):** Açık kaynak kodlu 2D bir MMORPG’dir.

**NPC (Non Player Character):** Genel olarak rol yapma oyunlarında kontrol edilemeyen fakat etkileşime girebilen karakterler için kullanılır.

**SVN:** SubVersioN ifadesinden türetilmiştir. Yazılım geliştirmede sürüm kontrolü amacıyla kullanılan sistemlerden biridir.

**GPL (General Public License):** Free Software Foundation tarafından geliştirilen ve açık kaynak kodu savunan lisanstır.

**GCC (GNU Compiler Collection):** Özgür yazılımlar için geliştirilmiş bir derleyici ailesidir.

**MUD (Multi User Dungeon):** Çok kullanıcıli zindan. RPG oyunlarının atası olarak kabul edilen, metin ekrandaki betimler ve komutlarla oynanan oyun türü.

**Script:** Bir programlama diline göre daha serbest bir yapısı olan ve genellikle yorumlanarak çalıştırılan kod blokları.

**GUI (Graphic User Interface):** Kullanıcıların bilgisayarla etkileşimini komut yazarak değil, pencere, düğme gibi öğelerin imleç yardımıyla kullanmasını mümkün kılan grafik arabirim.

**XML (eXtensible Markup Language):** Genişletilebilir işaretleme dili, hem insanlar hem bilgi işlem sistemleri tarafından kolayca okunabilecek dokümanlar oluşturmaya yarayan, www standartlarını belirleyen örgüt olan [W3C](#) tarafından tanımlanmış bir standarttır. Bu özelliği ile veri saklamanın yanında farklı sistemler arasında veri alışverişi yapmaya yarayan bir ara format görevi de görür.

**TCP/IP (Transmission Control Protocol/Internet Protocol) :** Bilgisayarlar ile veri iletme/alma birimleri arasında organizasyonu sağlayan, böylece bir yerden diğerine veri iletişimini olanaklı kılan pek çok veri iletişim protokolüne verilen genel addır.

**Port:** TCP/IP ve UDP protokollerinde kullanılan mantıksal bağlantılardır.

**Fantastik:** Büyü ve doğaüstü güçlerin hâkim olduğu dünya (fantasy themes).

**Bilge:** NPC'lere elektroGame'de verilen isim.

**Char:** elektroGame'de oyuncuyu temsil eden avatar.



**Karakter Puanları:** elektroGame'de oyuncunun karakteristik özellikleri geliřtirmesi için oyuncuya verilen puan.

**N:** Örneklem Büyüklüğü

**sd:** Serbestlik Derecesi

**F:** Frekans

**p:** Anlamlılık Düzeyi

**S:** Standart Sapma

**2B:** 2 Boyut veya 2 Boyutlu

**3B:** 3 Boyut veya 3 Boyutlu

## 2. LİTERATÜR

Bu bölümde öğrenmenin tanımı, kapsamı, yapılandırmacı ve sosyal yapılandırmacı kuram, öğrenme ortamları, oyun tabanlı öğrenme ortamları, oyunların özellikleri, bilgisayar oyunları, eğitsel bilgisayar oyunları ve MMORPG'ler ile ilgili literatür incelenmiştir.

### 2.1 Öğrenme ve Öğrenme Kuramları

Öğrenme, doğası gereği karmaşık bir süreç olup pek çok araştırmacı tarafından anlaşılmaya ve açıklanmaya çalışılmıştır. Ancak araştırmacılar tarafından üzerinde anlaşılmış bir öğrenme tanımı bulunmamaktadır (Senemoğlu, 2005). Farklı araştırmacılar tarafından yapılmış, öğrenmeye ilişkin tanımların bazıları aşağıdaki gibidir;

- Pavlov, E. C. Tolman, Thorndike, Guthrie, Skinner ve Hull'a göre öğrenme, uyarıcı ile davranım arasında bir bağ kurmadır.
- Max, Wertheimer, W. Köhler, K. Kofka ve Ausubel'e göre öğrenme; hem zekânın hem güdülenmenin hem de transferin ürünüdür.
- N. E. Miller, J. Dollard ve Piaget'e göre öğrenme; kişinin yeteneklerine, onun biyolojik ve kültürel gelişimine, içinde yaşadığı toplumdaki kültüre, güdülenmişliğe, ilgisine, öğrenme ortamının havasına bağlıdır.
- Allen Nevell, Herbert A. Simon, Gagne ve Briggs'e göre öğrenme, bilgi işlem sürecine benzer bir biçimde oluşur (Senemoğlu, 2005).

Araştırmacıların öğrenmeyi açıklamak için bu kadar farklı tanımlar geliştirmelerinin nedeni, öğrenmeye farklı açılardan bakan kuramlardan yola çıkmalarıdır.

Öğrenme kuramları, genel olarak öğrenen bireyin öğrenme süreci ile psikolojik ve davranışsal süreçler arasındaki ilişki ile ilgilenir.

Öğrenmenin nasıl gerçekleştiğini açıklamaya çalışan pek çok farklı kuram bulunmaktadır. **Tablo 2.1**'de en sık karşılaşılan kuramların uluslararası literatürde geçen adları ve kuramların savunucuları iki sütun halinde yer almaktadır (Culatta, 2010).

**Tablo 2.1:** Öğrenme kuramlarının listesi.

ACT-R (J. Anderson)	Information Pickup Theory (J. J. Gibson)
Adult Learning Theory (P. Cross)	Information Processing Theory (G. A. Miller)
Algo-Heuristic Theory (L. Landa)	Lateral Thinking (E. DeBono)
Andragogy (M. Knowles)	Levels of Processing (Craik & Lockhart)
Anchored Instruction (J. Bransford)	Mathematical Learning Theory (R. C. Atkinson)
Aptitude-Treatment Interaction (L. Cronbach & R. Snow)	Mathematical Problem Solving (A. Schoenfeld)
Attribution Theory (B. Weiner)	Minimalism (J. M. Carroll)
Cognitive Dissonance Theory (L. Festinger)	Model Centered Instruction and Design Layering (Andrew Gibbons)
Cognitive Flexibility Theory (R. Spiro)	Modes of Learning (D. Rumelhart & D. Norman)
Cognitive Load Theory (J. Sweller)	Multiple Intelligences (H. Gardner)
Component Display Theory (M. David Merrill)	Operant Conditioning (B. F. Skinner)
Conditions of Learning (R. Gagne)	Originality (I. Maltzman)
Connectionism (E. Thorndike)	Phenomenonography (F. Marton & N. Entwistle)
Constructivist Theory (J. Bruner)	Repair Theory (K. VanLehn)
Contiguity Theory (E. Guthrie)	Script Theory (R. Schank)
Conversation Theory (G. Pask)	Sign Theory (E. Tolman)
Criterion Referenced Instruction (R. Mager)	Situated Learning (J. Lave)
Double Loop Learning (C. Argyris)	Soar (A. Newell et al.)
Drive Reduction Theory (C. Hull)	Social Development (L. Vygotsky)
Dual Coding Theory (A. Paivio)	Social Learning Theory (A. Bandura)
Elaboration Theory (C. Reigeluth)	Stimulus Sampling Theory (W. Estes)
Experiential Learning (C. Rogers)	Structural Learning Theory (J. Scandura)
Functional Context Theory (T. Sticht)	Structure of Intellect (J. Guilford)
Genetic Epistemology (J. Piaget)	Subsumption Theory (D. Ausubel)
Gestalt Theory (M. Wertheimer)	Symbol Systems (G. Salomon)
GOMS (Card, Moran & Newell)	Triarchic Theory (R. Sternberg)
General Problem Solver (A. Newell & H. Simon)	Transformational Theory (J. Mezirow)

Bu kuramlar, dayandıkları ortak ilkeler göz önüne alınarak farklı şekillerde gruplandırılabilir. Örneğin (Senemoğlu, 2005), öğrenme kuramlarını:

- Davranışçı-Çağrışım Kuramları
  - Pavlov'un Klasik Koşullanma Kuramı
  - Watson ve Guthrie'nin Bitişiklik Kuramı
  - Thorndike'in Bağ Kuramı
  - Skinner'in Edimsel Koşullanma Kuramı

- Hull'un Sistemantik Davranış Kuramı
  - Bilişsel Ağırlıklı Davranışçı Öğrenme Kuramları
    - Tolman'ın İşaret-Gestalt Kuramı
    - Bilişsel Sosyal Öğrenme Kuramı
  - Bilişsel Öğrenme Kuramları
    - Gestalt Kuramı
    - Bilgiyi İşleme Kuramı
  - Nörofizyolojik Öğrenme Kuramı
- şeklinde dört grupta toplamaktadır.

Deryakulu ise bu kuramları aşağıdaki gibi sınıflandırmaktadır (Deryakulu, 2001);

- Nesnelci Görüş (Objectivist)
  - Davranışçı Kuramlar
  - Bilgiyi İşlemeye Dayalı Bilişsel Kuramlar
- Yapılandırmacı Görüş (Constructivist)

Bu bölümde öğrenme kuramları Deryakulu'nun yaptığı sınıflamaya göre incelenmiştir.

Nesnelcilik, bilginin ne olduğuyla ve bir şeyi bilmenin ne anlama geldiğiyle ilgilenen felsefî bir görüştür. Lakoff'a göre bilginin bireyden bağımsız olduğu gerçekçiliğin bir sürümüdür (Lakoff, 1987). Dünya hakkında güvenilir bir bilginin varlığına inanır (Özden, 2003; Bednar, Cunningham, Duffy ve Perry, 1995, Akt. Deryakulu, 2001). Öğretim boyutunda ise hedefini, bu gerçekliği taşıyan kuramsal modelleri bireylere kazandırmak biçiminde belirler. Bireyin dışında nesnel bir gerçekliğin var olduğu inancı ile nesnelciler, öğrencilerin hedefinin bu bilgileri kazanmak; öğretmenlerin görevinin de bu bilgiyi aktarmak olması gerektiğini savunurlar. Nesnelci anlayışa göre; öğrenme nesnel gerçekliğin özümsemesi sürecidir (Jonassen, 1991a). Diğer bir deyişle öğrenme, objektif gerçekliği özümleme sürecidir. Eğitimin rolü, öğrenenlerin gerçek dünyayı öğrenmesine yardım etmektir. Öğretmenlerin amacı ise öğrenenler için olayları yorumlamaktır (Özden, 2003).

Nesnelci görüşün hâkim olduğu geleneksel öğretim uygulamalarının hedefi, öğrencinin zihninde tek doğru olarak kabul edilen önceden belirlenmiş bilgilerin istenilen şekilde oluşmasını sağlamaktır. Sonuç olarak, nesnelci öğretim yaklaşımlarının ulaşmak istediği, herkesin, dünya ile ilgili aynı nesnel, tam ve doğru olarak kabul edilen anlayışı kazanmasıdır (Jonassen, 1991b).

1950'li yıllarda eğitime hâkim olan davranışçılık, gelişimini daha çok hayvanlar üzerine yapılan deneylerle sağlamıştır. Bu yüzden insan davranışlarının nedeninden çok sonuçlarına odaklanmıştır (Crowley, 2002).

Davranışçı kurama göre, öğrenmenin gerçekleşmesi, yani organizmada istenilen davranışların oluşması, organizmaya dışarıdan gerekli uyarıcıların verilmesi ile gerçekleşir. Kısaca davranışçılara göre öğrenme etkiye verilen tepkidir. Öğrenmede dış koşullar önemli bir yer kaplamaktadır. Asıl ilgi dışsal çevrenin (öğretim ortamlarının, materyallerinin ve stratejilerinin) planlanması üzerine yoğunlaşmaktadır. Gerekli koşullar oluşturulduğunda her öğrencinin öğrenebileceğine inanılır.

Bilişsel kuram da davranışçı kuram gibi nesnelci felsefeye dayalıdır. Davranışçı kuramda odak noktası, bireyin davranışları iken, bilişsel kuramda odak noktası, bireyin zihinsel faaliyetleri, beynin yapısı ve düşünmenin nasıl gerçekleştiğidir. Bilişsel bakış açısını temel alan araştırmacılar öğrenmeyi doğrudan gözlemlenemeyen zihinsel bir süreç olarak tanımlamaktadır. Bilişsel kuramın önemli temsilcilerinden Piaget'e göre, öğrenme insanın dış dünyadan kendisine gelen her şeye anlam bulmaya çalıştığı dinamik, bilişsel bir yapı grubudur. Bu anlam yükleme, bireyin deneyimine, sahip olduğu kültüre, içinde öğrenmesini gerçekleştiği yere ve öğrencinin bu süreçteki rolüne göre değişim göstermektedir (Korkmaz, 2006).

Bilişsel kuramda öğrenmenin zihinsel süreçleriyle, davranışçı kuramda ise organizmanın verdiği gözlenebilir tepkilerle ilgilenilmesine rağmen iki kuramda da bilgiyi nesnel bir gerçeklik olarak ele alınır. Her iki kuramda da öğretim sürecine ait hedef ve etkinlikler önceden belirlenir ve bu hedef doğrultusunda kesin doğru olduğuna inanılan bilgi bireye aktarılır.

Yapılandırmacı öğrenme kuramının temelleri, Kant'tan fazlasıyla etkilenmiş olan Piaget'in fikirlerinden yola çıkılarak atılmıştır. Ayrıca eğitimin değişen ihtiyaçlar doğrultusunda yeniden yapılandırılması gerektiğini savunan Dewey'in ve eğitimin hayatın kendisi olduğunu söyleyen J. J. Rousseau'nun görüşleri de yapılandırmacı kuramın gelişmesinde etkili olmuştur (Duman, 2008).

Yapılandırmacı kuram ilk dönemlerinde, bilginin nasıl öğrenildiğini açıklamaya çalışırken zaman içinde öğrenenlerin bilgiyi nasıl yapılandırdıklarına ilişkin bir kuram haline dönüşmüştür (Demirel, 2002).

Yapılandırmacı kurama göre, bilgi bireyden bağımsız olarak bulunmaz ve bireye hazır olarak aktarılmaz. Birey, aktif olarak bulunduğu öğrenme sürecinin içinde, bilgiyi kendi zihninde oluşturur. Bunun için bir dizi deneyimi ve birtakım zihinsel faaliyetleri gerçekleştirilmesi ve bilgiyi özümlemesi gerekmektedir. Bu süreç içinde öğretmen de bireye bilgiyi inşa etmesi için gerekli ortamı hazırlamalı, deneme, keşfetme fırsatları vermeli, yönlendirici bir rol üstlenmelidir (Akpınar, 1999; Saban, 2002).

Öğrenenler öğrenme esnasında yeni bir bilgiyle ilk kez karşılaştıklarında, o an ham halde bulunan bu bilgiyi zihinlerinde eski bilgilerle karşılaştırarak yeni bir yapı oluştururlar. Oluşturulan bu yeni yapı içerisinde yeni karşılaşılan bilgi, eski bilgilerle çelişmeden aralarında çeşitli ilişkiler oluşturulabiliyorsa bu yeni bilgi öğrenenin zihinsel yapısının bir parçası haline gelir. Başlangıçta ham olan bilgi ilişkilendirme, özümleme ve içselleştirmeyle anlamlı bir bilgiye dönüştürülür. Ancak, yeni bilgi öğrenenin mevcut zihinsel yapısıyla çelişiyorsa ya da bir uyumsuzluk oluşuyorsa bir çatışma ortaya çıkar. Her yeni öğrenme, öğrencinin zihinsel yapısını tekrar gözden geçirdiği, ona bir şeyler ekleyerek geliştirdiği ya da gerektiğinde değişiklik yaptığı içsel bir deneyimdir. Öğrenenler öğrenme esnasında bu süreçlere hem fiziksel, hem de zihinsel yönden etkin olarak katılırlar. Bu süreç, “bilgi yapılandırma” olarak adlandırılır (Deryakulu, 2001).

Yapılandırmacı öğrenme yaklaşımı ile ilgili görüşler şu şekilde özetlenebilir (Saban, 2002):

1- Öğrenme pasif bir bilgi alma süreci değil, aktif bir anlam oluşturma sürecidir.

2- Öğrenme, kavramsal değişmeyi içerir. Öğrenme çeşitli kavramlar ile ilgili daha önceki anlayışlarını daha karmaşık ve daha geçerli hale getirmek için yeniden yapılandırmasıdır.

3- Öğrenme öznelidir. Öğrenme, bir bireyin öğrendiği şeyleri çeşitli semboller, metaforlar, imgeler, grafikler veya modeller yoluyla içselleştirmesidir.

4- Öğrenme durumsaldır ve çevresel şartlara göre şekillenir. Öğrenciler, alıştırma yapmak yerine, gerçek hayatta karşılaştıkları problemlere benzer nitelikteki problemleri çözmeyi öğrenirler.

5- Öğrenme sosyaldır. Öğrenme, bireylerin perspektiflerini paylaşmak, bilgi alışverişinde bulunmak ve problemleri işbirliği içinde çözmek üzere başkaları ile olan etkileşimleri sayesinde gelişir.

6- Öğrenme duygusaldır. Zihin ve duygu birbirleriyle ilişkilidir. Dolayısıyla, öğrenmenin doğası bireyin kendi becerileri hakkında sahip olduğu görüşler ve farkındalıklar, öğrenme amaçlarının açıklığı, kişisel beklentiler ve motivasyon gibi unsurlardan etkilenir.

7- Öğrenme gelişimseldir ve bireylerin sosyal, fiziksel, duygusal ve zihinsel gelişimleri ile doğrudan ilişkilidir.

8- Öğrenme, öğrenci merkezlidir.

9- Öğrenme sürekli dir. Öğrenme belli bir yer veya zamanda başlayıp belli bir yer veya zamanda durmaz, aksine sürekli devam eder.

Günümüzde yapılan araştırmalarda; bilişsel (cognitive), radikal (radical) ve sosyal (social) yapılandırıcılık yaklaşımları sıklıkla kullanılmaktadır (Kılıç, 2001). Küçüközer'e göre bilişsel ve sosyal yapılandırıcılık fen eğitimi araştırmalarında sıklıkla kullanılan yaklaşımlardır (Küçüközer, 2004).

Aşağıda bilişsel, radikal ve sosyal yapılandırıcılık hakkında bilgi verilmiştir.

Bilişsel yapılandırıcılar, bilginin oluşumunda Piaget'in öğrenme kuramından hareket etmektedir. Piaget, öğrenmeyi özümseme, uyum ve bilişsel denge kavramlarıyla açıklamaktadır (Glaserfeld, 1995). Burada yeni bilgi, birey tarafından özümseğinde eski bilgileriyle bağlantı kurulup uyumlu hale gelirse denge durumu oluşmaktadır. Yeni bilgi eski bilgilerle bir çelişki durumu

oluşturuyorsa bilişsel bir dengesizlik durumu yaşanır. Dengesizliği gidermek için birey, eski bilgilerin yerlerini değiştirerek, düzelterek, birleştirerek, parçalara ayırarak ve parçaları tekrar bir araya getirerek yeni bir yapı oluşturacaktır. Bu süreçlerden bir ya da birkaçını kullanan bireyde yeni bir bilişsel denge durumu ortaya çıkmaktadır (Philips ve Soltis, 2005).

Radikal yapılandırmacılar ise bilginin oluşumunu açıklamada Von Glasersfeld'in görüşlerini benimsemişlerdir. Radikal yapılandırmacılara göre bilgi, etkileşim yoluyla ve sezgilerle edilgen bir şekilde alınamaz, bireyin kendi zihninde oluşur (Glasersfeld, 1995). Bu anlamda bilgi, dış dünyanın gerçeğine benzemeyebilir. Bilgi, ontolojik gerçekliğin keşfedilmesine değil, bireyin dış dünyayı düzenlemesine hizmet eder (Glasersfeld, 1989).

Bu yaklaşımın kendine göre disiplinleri vardır ve bu yaklaşıma göre öğrenci bilgileri gerçek dünya içinde yaşadığı kendi deneyimleriyle yapılandırır. Bunu yaparken var olan problemlere çözüm yolları geliştirmeye çalışır (İşman, 1999).

Bu yaklaşım, öğrencilerin bilgiyi oluştururken, yetişkinler tarafından geliştirilen materyal ve açıklamaları temel almalarından ziyade, kendilerinin geliştireceği materyallerin kullanılmasının daha önemli olduğunu savunur (Doolittle, 1999).

Vygotsky, Piaget'in bireyin kendi kendine oluşturduğu şemalardan daha fazlasını başkalarından öğrendiğini savunmuştur. Özellikle insanların birbirleriyle etkileşim kurmak için kullandıkları psikolojik araçların yani sayılar, semboller, kavramlar, kelimeler gibi birlikte yaşamak için kullanılan araçların başkalarından öğrenilmesinin zorunlu olduğunu ifade etmiştir. Vygotsky de Dewey gibi dilin öncelikle bir iletişim aracı olduğuna ve dile ait kavramların sosyal bir ortamda aktarıldığına ve kazanıldığına inanır (Philips ve Soltis, 2005).

Sosyal yapılandırmacı anlayışa göre, “bilginin yapılanması sürecinde bireylerin içinde bulunduğu sosyal ortamın önemli bir rol oynaması” öğrenmede en önemli etkidir (Duit ve Treagust, 1998) . Bu anlayışa göre bilgi, belli bir sosyal ortam içinde yapılandırılmış olup ideoloji, din, politika, ekonomi, insan ilişkileri ve öğrenme çevresinin maddi özelliklerinden etkilenmektedir (Widodo, Duit ve Müller, 2002). 1980’li yıllara kadar yapılan araştırmalarda bilgi yapılanmasının sosyal



boyutu göz ardı edilmişken daha sonra bu boyut, sosyal yapılandırmacı anlayış çerçevesinde incelenmeye başlanmıştır. Sosyal perspektifler, yapılandırmacı anlayışın sosyal ve bireysel yapılandırmacılığın her ikisini de birleştiren bir görüşe doğru zenginleşmesini ve gelişmesini sağlamıştır (Duit ve Treagust, 1998; Akt. Küçüközer, 2004).

Benzer şekilde Sağıroğlu (2002)'na göre, öğrencinin öğrenme süreci sonunda zihninde bireysel olarak oluşturduğu bilgi, anlam ya da yorumlar üzerinde toplumsal olarak da uzlaşmış bilgi, anlam ya da yorumlar olmalıdır. Diğer bir deyişle, yapılandırmacı görüş, öğrenmenin bireysel olduğu kadar toplumsal bir etkinlik olduğunu da savunmaktadır (Saban, 2002).

Vygotsky (1987), bireyin bilişsel sisteminin kendisinin sosyal yaşantısından ayıramayacağını ve onun bir sonucu olarak görülmesi gerektiğini belirtmiş ve çocuğun bilişsel gelişiminde dil ve kültürün önemli rol oynadığını vurgulamıştır. Vygotsky'e göre öğrenciler problemlerini kendi bilişsel gelişim seviyelerinden ziyade, yetişkinlerin veya akran gruplarının yardımını alarak çözmektedir ve bundan ötürü sosyal etkileşim, bilişin gelişmesinde temel bir rol oynamaktadır. Bilgi edinmede öğrenme ortamının ve o ortamdaki bireylerle iletişimin büyük bir payı vardır. Öğrencinin daha deneyimli akran ve öğretmenlerle çalışırken bilişsel fonksiyonları daha iyi gelişir.

Sosyal yapılandırmacılara göre bilgi, sosyal ve kültürel olarak birey tarafından oluşturulan bir üründür (Prawat ve Floden, 1994). Birey, diğer kişilerle ve çevresiyle etkileşimde bulunarak kendi anlamlarını oluşturur. Anlamlı öğrenmeler, bireyin sosyal etkinliklere katılmasıyla oluşur. Sosyal oluşturmacılar göre gerçek keşfedilemez, toplumun üyeleri birlikte dünyanın gerçeklerini oluştururlar (Kukla, 2000).

## **2.2 Öğrenme Ortamları**

Brown (2002), öğrenmenin bir öğretim sonucu değil öğrenmeyi teşvik edici bir çevrenin sonucu olduğunu savunmuştur (Tüzün ve Bayırtepe, 2007). Öğrenmenin

gerçekleştiği bu çevreye öğrenme ortamı denilmektedir. Geleneksel sınıf ortamı, bilgisayar ortamında oluşturulmuş sanal sınıflar birer öğrenme ortamıdır.

Öğrenme-öğretme sürecinde öğrenmenin istenilen düzeyde gerçekleşebilmesinde öğrenme ortamının özellikleri önemli rol oynamaktadır. Bu yüzden öğrenme ortamının öğrenmeyi kolaylaştıracak şekilde tasarlanması, zengin araç, gereç ve kaynaklarla donatılması gerekir. İskoç Müfredat Danışma Konseyi'nin tanımına göre iyi bir öğrenme ortamının özellikleri şöyle olmalıdır (SCCC, 1996):

- İyi bir öğrenme ortamı üst düzey bir yarışma ortamı ve düşük endişe duygusu içermelidir.
- Öğretmenin davranışları olumlu öğeler içermelidir.
- Öğrenmenin daha kalıcı ve çabuk olması için öğrenme ortamı yeni materyallerle zenginleştirilmelidir.
- Bireysel öğrenmenin gerçekleşebilmesi için öğrenme ortamı farklı türde etkinlikler içermelidir.
- Kalıcı bir öğrenme için müzikten faydalanılabilir.
- Öğrenme stratejileri bir plan dâhilinde birleştirilmelidir.
- Öğrencilerin ihtiyaç duydukları kaynaklar ulaşılabilir, iyi tanımlanmış ve amaca uygun olmalıdır.
- Ortam aktif öğrenmeyi destekleyici özgür bir çevre sunmalıdır.
- Öğrencilerin problem çözerken işbirlikçi bir yaklaşım benimsemesi sağlanmalıdır.
- Tüm öğrencilere eşit fırsat sunulmalıdır.

Öğrenme ortamları, sahip olması gereken bu özellikler göz önünde bulundurularak çeşitli öğretim tekniklerinden yararlanılarak farklı şekillerde tasarlanmaktadır. Öğrenme ortamlarından birisi de eğitsel oyunlardır. Bu nedenle eğitsel oyunlar tasarlanırken de bu özellikler dikkate alınmalıdır.

Çocuklar için, kendilerinin isteyerek seçip yaptıkları ve nasıl yapacaklarının kararını kendilerinin verdikleri her tür etkinliğe “oyun” denmektedir. Aynı zamanda oyun çocuklar için ciddi bir uğraştır ve değerli bir aktif öğrenme yoludur (Bilgi, 2005).

Ward (2004)'a göre de bilgisayar oyunları çocukların öğrenmesine yardımcı olabilir. Tüm oyunlar gibi bilgisayar ve video oyunları da sosyal gelişimi hızlandırırken çocukları eğlendirir; ayrıca oyun oynama ve oyunlar hakkında konuşup tartışma gençlerin yaşamlarının önemli bir parçasını oluşturur.

Çevrim içi oyun tabanlı öğrenme ortamları; üst düzey bir yarışma ortamı sunma, bireysel ve işbirlikçi öğrenmeyi destekleme, istenilen kaynaklara internet aracılığıyla kolayca ulaşmayı sağlama ve herkese eşit fırsatlar tanıma gibi özellikleri bir arada bulundurmasından dolayı iyi öğrenme ortamlarından biridir (Kaya, 2005).

Takip eden bölümde çevrim içi oyun tabanlı öğrenme ortamlarının daha iyi tanımlanabilmesi için genel olarak bilgisayar oyunları, türleri ve özelliklerinden bahsedilmiştir.

### 2.3 Bilgisayar Oyunları

Yüksek hesaplama gücüne sahip bilgisayarlar ortaya çıkmaya başladıktan sonra, bilgisayarların eğlence amacıyla kullanılmaları fazla uzun sürmemiştir. İlk bilgisayar oyunu olarak kabul edilen [Tennis for Two](#) 1958 yılında monitör olarak osiloskop ekranını kullanan analog bir bilgisayar aracılığı ile oluşturulmuştur. İlk dijital oyun ise 1962 yılında, Massachusetts Institute of Technology laboratuvarında bulunan PDP-1 model bilgisayar kullanılarak [Spacewar!](#) ismiyle dizayn edilmiştir. Söz konusu bu oyunlar, nükleer santrallerde veya büyük üniversite laboratuvarlarında bulunan dönemin gelişmiş bilgisayarları için tasarlanmış, türünün ilk örneği oyunlardır (Wikipedia, Bilgisayar Oyunu, 2011).

**Tablo 2.2:** Tennis for Two ve Spacewar! oyunlarının görüntüsü.



Bilgisayar oyunlarının gelişimi, bilgisayarların gelişimi ve yaygınlaşması ile paralel olmuştur. Bilgisayarların işlem güçleri ve grafik yetenekleri arttıkça oyunların özellikleri gelişmiş, bilgisayarlar yaygınlaştıkça da oyun oynayanlar ve dolayısıyla da oyunlarla ilgilenen firmaların sayısı artmış, bunun sonucu olarak çok çeşitli ve kaliteli oyunlar ortaya çıkmıştır. Örneğin, ilk bilgisayar oyunu olan Tennis

for Two, osiloskop ekranında oynanabilen nokta ve çizgilerden ibaret bir oyunken günümüzde 3B olarak hazırlanmış onlarca tenis oyunu vardır (Wikipedia, Tennis video games, 2011).

**Tablo 2.3:** Modern bir tenis oyunu ile Tennis for Two oyununun karşılaştırılması.



Bilgisayarların gelişimi ve sayılarının artmasıyla birlikte bilgisayarlar birbirlerine bağlanarak bilgisayar ağları oluşturulmuştur. Bu bilgisayar ağları üzerinde oynanabilen ilk oyun ise 1973 yılında yazılan Empire oyunudur (Wikipedia, History of video games, 2011). Birden çok kişinin oynadığı bu tür oyunlara çok kullanıcı oyunları denmektedir.

Çok kullanıcı oyunlarının gelişimi internetin ortaya çıkması ve tüm dünyaya yayılmasıyla birlikte farklı bir boyut kazanmıştır. Önceleri bir ya da birkaç kişinin aynı ortamda oynayabileceği oyunlar yazılırken günümüzde milyonlarca oyuncunun aynı ortamda olduğu oyunlar yazılmaktadır. Ayrıca tek kişilik olarak yazılmış ve çok sevilmiş oyunlar da çok kullanıcı haline getirilmektedir. Örneğin, 1990'lı yıllarda çok sevilmiş bir oyun olan DOOM, aynı zamanda hem bilgisayara karşı hem de diğer insanlara karşı oynayabiliyorlardı. Ancak aynı oyuna katılan oyuncu sayısı sınırlıydı ve genellikle 4-8 kişi arasında kalıyordu. İnternet bağlantılarındaki hızın artmasıyla birlikte daha fazla sayıda oyuncunun eşzamanlı katılabileceği ortamlar çoğalmıştır. Katılımcı sayısının çokluğundan dolayı bu oyunlara “Devasa Çok Kullanıcı” (massively multiplayer) oyunları denilmektedir.

Günümüzde oyunlar, grafik, ses ve ağ kullanımı bakımından oldukça ilerlemiştir. Bazı oyunlarda hem sadece bir çerçeve (frame) için milyonlarca poligon çizilebilmekte hem gerçekçi ses efektleri üretilebilmekte hem de bu oyunları aynı anda milyonlarca kullanıcının birlikte oynayabilmesi mümkün olabilmektedir.

### 2.3.1 Oyun Türleri

Bilgisayar oyunları farklı yaşlarda, farklı cinsiyetlerde ve farklı kültürlerdeki oyuncular için çok çeşitli türlerde geliştirilmektedir. Örneğin, futbol veya basketbol gibi sporlardan hoşlanan oyuncular için spor oyunları üretilirken, kız çocukları için bebek giydirme, makyaj yapma gibi küçük çaplı oyunlar üretilmektedir (Wikipedia, Bilgisayar Oyunu, 2011).

Bilgisayar oyun türleri farklı kaynaklarda farklı şekillerde tanımlanmıştır (Kirriemuir ve McFarlane, 2004; Crawford, 1982). Prensky bilgisayar oyunlarını; aksiyon oyunları, macera oyunları, dövüş oyunları, bulmaca oyunları, rol yapma oyunları, simülasyon oyunları, spor oyunları ve strateji oyunları olarak sekiz başlık altında toplamıştır (Prensky, 2001).

**Aksiyon Oyunları:** Oyunların en temel ve en geniş türüdür. Aksiyon filmlerinin oyunlardaki karşılığı olarak düşünülebilir. Bu tür oyunlarda, oyuncu sürekli olarak oyuna anlık müdahaleler yapar ve aktif olarak oyunun içindedir. Savaş ve şiddet öğeleri yer alabilir. Birinci şahıs nişancı (FPS), araba yarışı gibi oyunlar bu türe ait oyunlardır.

**Macera Oyunları:** Oyuncu bilinmeyen sanal bir dünyanın içindedir. Araştırarak, keşfederek, bu dünyada nesnelere toplayarak çeşitli problemleri çözer. Oyunun akışı genellikle hikâye odaklıdır. Zork ve Resident Evil serileri, macera oyunlarının önemli örneklerindedir.

**Dövüş Oyunları:** Oyuncu, kendi seçtiği karakter ile bilgisayar ya da diğer oyuncu tarafından kontrol edilen diğer karakterler ile dövüşür. Klasik bir örneği Street Fighter oyunudur.

**Bulmaca Oyunları:** Oyuncuların labirent gibi bir ortamda dolaştığı ya da mantıksal problemler çözdüğü oyunlardır. Tetris türevleri bu tip oyunlara örnek gösterilebilir.

**Rol Yapma Oyunları (RPG):** Oyuncunun insan ya da herhangi bir canlı karakteri yönettiği oyunlardır. Oyuncu oyunda mücadele ettikçe deneyim ve çeşitli

ekipmanlar kazanarak karakterini geliştirir. Diablo, Fall Out oyunları RPG türüne örnektir.

Çevrim içi olarak ve çok kullanıcı ile oynanan RPG oyunlarına MMORPG (Massive Multiplayer Online Role Playing Game - Devasa Çok Kullanıcılı Rol Yapma Oyunları) denilmektedir.

**Simülasyon Oyunları:** Gerçek hayatta olan bir durumun oyunsallaştırılarak bilgisayar ortamına aktarılmış halleridir. Örneğin, bir şehir kurabilir, bir uçak kullanabilir ya da bir şirketin yöneticisi olabilirsiniz. Örnek olarak Electronic Arts firmasının SimCity serisi, Microsoft'un Flight Simulator serisi verilebilir.

**Spor Oyunları:** Fiziksel sporların bilgisayar ortamında olanlarıdır. Oyunlarda, bazen oyuncu olunurken bazen teknik direktör olunabilir. Futbol, basketbol gibi tek spor dalını içeren oyunlar olabildiği gibi olimpiyat sporları, kış sporları gibi başlıklar altında pek çok sporun yer aldığı paketler de bulunabilmektedir. FIFA Soccer, Fight Night Round, Winter Games spor oyunlarına örnek olarak verilebilir.

**Strateji Oyunları:** Oyuncunun ordu, halk gibi büyük grupları yönettiği oyunlardır. Gerçek zamanlı (Real-Time Strategy - RTS) ve Sıra Tabanlı (Turn-Based Strategy - TBS) olmak üzere iki ana türe ayrılmaktadır. Gerçek zamanlı strateji oyunlarında oyuncu rakipleriyle karşılıklı hamleler yoluyla mücadele eder. Sıra tabanlı strateji oyunları ise, satranç gibi, hamlelerin sırayla yapıldığı oyunlardır. The Battle for Wesnoth TBS'ye, Age Of Empires ise RTS'ye örnek olarak verilebilir.

Bu türlere ek olarak Prensky'nin taksonomisi dışında kalan ancak diğer araştırmacılarca bahsedilen oyun türleri ise şunlardır (Graser, 2009):

**Müzik Oyunları:** Belirli ritim, nota ya da hareket dizgelerini tekrar etmek üzerine kurulu oyunlardır. Genellikle gitar, dans pad gibi yardımcı bir araç kullanılır. [Guitar Hero](#), [Rock Band](#) ve Sing Star oldukça popüler olmuş müzik oyunlarından.

**Parti Oyunları:** Oyuncuların birlikte oynayabilmesi için özellikle çok oyunculu olarak geliştirilmiş küçük oyunlardır. Fiziki olarak bir araya gelen oyuncular, içlerinden kimin en iyi, en hızlı olduğunu belirlemeye çalışırlar. Mario

Party serisi, Crash Boom Bang! ve Rayman Raving Rabbids, türün önde gelen oyunlarından.

Bilgisayar oyunlarının türleri zaman içinde kullanıcı tercihlerine göre değişiklik gösterebilmektedir. Oyun firmaları geliştirdikleri oyunları daha fazla ilgi görmesi için farklı oyun türlerinden özellikler taşıyacak şekilde geliştirmektedir.

Oyun türlerinin sınıflandırılmasında kullanılan bir diğer yöntemde oyunların amaçlarına değerlendirilmesidir. Aşağıda amaçlarına göre oyun türleri verilmiştir.

### **Amaçlarına Göre Oyun Türleri**

- Yetişkin oyunları
- Reklam oyunları
- Sanat oyunları
- Dini oyunlar
- Basit (Casual) oyunlar
- Eğitsel oyunlar
- Elektronik spor oyunları
- Alıştırma oyunları
- Ciddi oyunlar

Bu çalışmada geliştirilen eğitsel oyunun türü olan MMORPG türü oyunlardan, Bölüm 4.1’de ayrıntılı olarak bahsedilmektedir.

### **2.3.2 Oyunların Özellikleri**

Bilgisayar ve bilgisayar oyunları insanlık tarihinin en ilgi çekici eğlencesi olma potansiyeline sahiptir. Ancak her eğlenceli etkinlik oyun olarak nitelendirilemez. Maroney, amaçları ve yapısı olan eğlenceyi oyun olarak tanımlamıştır (Maroney, 2001). Kramer, oyunun yapısını donanım, kurallarını ise yazılıma benzeterek yazılım ve donanımın bir araya gelerek oyunu oluşturduğunu söylemiştir (Kramer, 2000). Prensky ise daha kapsamlı bir tanım yapmış ve oyunların, on iki unsurun bir araya gelerek oluştuğunu söylemiştir. Bu on iki unsur şunlardır (Prensky, 2001):

1. Oyunlar eğlendiricidir ve zevk verir.
2. Oyunlar sürükleyicidir ve tutkuyla bağlanmayı sağlar.
3. Oyunların kuralları vardır. Kurallar oyunun yapısını oluşturur.

4. Oyunların motive edici amaçları vardır.
5. Oyunlar etkileşimlidir ve oyuncunun aktif katılımını sağlar.
6. Oyunların zorluğu oyuncunun başarısına göre ayarlanabilir. Bu durum, oyuncuları oyunda tutar.
7. Oyunların çıktıları ve dönütleri vardır. Oyuncular bu sonuçlar sayesinde öğrenirler.
8. Oyunlarda oyuncunun galip gelerek egosunu tatmin edebileceği durumlar vardır.
9. Oyunlarda mücadele, yarış, meydan okuma ve rekabet vardır. Bu durum oyunculara heyecan verir.
10. Oyunlarda çözülmesi gereken problemler vardır. Bu, oyuncuların yaratıcılığını geliştirir.
11. Oyunlarda oyuncular arası etkileşim vardır. Sosyal gruplar oluşmasını sağlar.
12. Oyunların sundukları hikâyeler vardır. Oyuncular hikâyenin bir parçası olarak hikâyedeki duyguları yaşarlar.

Prensky, bu özelliklerden altı tanesini oyunları eğlenceli kılan özellikler olarak tanımlarken diğer altısını temel yapısal özellikler olarak tanımlamaktadır. Bir oyunda mutlaka bulunması gereken altı temel unsur şunlardır (Prensky, 2001):

**Kurallar:** Oyunların diğer eğlence türlerinden farkı çeşitli kurallarının bulunmasıdır. Düzenlenmiş kuralları olmayan bir eğlenceyi, oyun olarak tarif etmek mümkün değildir. Kurallar oyunları tüm oyuncular için adil ve heyecan verici yapmalıdır.

Oyunda kaç oyuncunun olacağı, oyuncuların rolleri, geribildirimler, oyun düzeyleri arasındaki geçişler, oyuncunun davranışlarına göre oyun içerisinde ilerlemenin ne şekilde gerçekleşeceği, oyunun zorluk düzeyi, bunların oyuncuya sunulması ve hikâye ile bağdaştırılması gibi konular oyun kuralları ile belirlenir (Prensky, 2001; Kramer, 2000a; Maroney, 2001).

Özetle kurallar; oyunu yapılandırır, oyunun amaçlarının fark edilmesini sağlar, oyuncunun hareketlerini sınırlandırır, oyuncuyu strateji geliştirmeye zorlar (Kula, 2005).



**Hedefler ve Amaçlar:** Amacı olan bir oyunda hedefe ulaşmak motivasyonu büyük miktarda artırır. Oyunlarda hedefler “yüksek skor elde etmek”, “bayrak kazanmak”, “sona ulaşmak” gibi olabilmektedir ve bu hedefler oyunun kurallarıyla doğrudan ilişkilidir. Hedefe ulaşmak bazen sıkıcı olsa veya oyuncuları zorlasa da oyuncunun kapasitesini ortaya çıkarmak üzere ona baskı kurduğu ve kazanmaya ittiği için oldukça önemlidir.

**Çıktılar ve Geribildirimler:** Hedefe ne kadar ulaşıldığının ölçülmesi, kazanma ya da kaybetme bildirimlerinin verilmesi bakımından geribildirimler oyuncular için önemlidir. Bildirimler çok çeşitli formlarda olabilir ama temel özellikleri “anında” dönüt vermeleridir.

Geribildirimler, oyuncuya oyun anında yaptıkları aktivitelerin başarıya ulaştırılan aktivite olup olmadığıyla ilgili anında bilgi verdikleri için oyuncunun stratejisini gözden geçirmesini, gerekiyorsa farklı bir strateji denemesini sağlayarak oyun içinde öğrenmeyi kolaylaştırırlar.

**Çatışma/yarışma/mezdan okuma/rekabet:** İnsan olma doğası gereği, rekabet bulunduran ortamlarda birey, rakiplerinin önüne geçerek en başarılı olma eğilimindedir. Bu durum, oyuncuya kendi karakteriyle özdeşleşerek oyunda da başarılı olma isteği kazandırır. Genel oyunlar bir problem çözme etkinliği olarak düşünüldüğünde, oyuncunun o problemi çözmek için bazen bilgisayara karşı bazen de diğer oyunculara karşı mücadele etmesi, mezdan okuması, rakipleriyle çatışması gerekir.

**Etkileşim:** Etkileşimin, geribildirimler yoluyla oyun ve oyuncu arasındaki etkileşim ve oyuncunun diğer oyuncularla sosyal ortamdaki etkileşimi olmak üzere iki boyutu vardır. Bilgisayara karşı oynanan oyunların bazılarında da yapay zekâ algoritmaları sayesinde, bilgisayarın insana daha yakın düşünerek seçimler yapmasının sağlanabilmesi, gün geçtikçe daha iyi durumlara gelmektedir.

**Hikâye:** Bir oyunun ne hakkında olduğuyla ilgilidir. Örneğin, satranç rekabet, tetris doku inşa etme, The Age of Empires savaş stratejisiyle ilgilidir. Oyunun hikâyesinin oyuncuya çoğunlukla oyunun başında bildirilmesi gerekmektedir. Bu bildirim genellikle oyunun hedefleriyle beraber sunulur. Hedefleri çok açık olan satranç gibi oyunlarda hikâyeyi sunmaya gerek olmayabilir.

Oyunun güçlü ve etkili bir hikâyesinin olması, oyuncunun, oynadığı karakteri daha çok benimseyip oyunun hedeflerini gerçekleştirme isteğini arttırmada önemli bir yere sahiptir.

Garris, Ahlers ve Driskell oyun özelliklerini;

- Fantezi,
- Kurallar/amaçlar,
- Duyusal uyarıcılar,
- Mücadele,
- Gizem,
- Kontrol

olmak üzere 6 temel boyutta toplamıştır (Garris, Ahlers ve Driskell, 2002).

### 2.3.3 Eğitsel Bilgisayar Oyunları

Bilgisayar oyunlarının, oyuncuların motivasyonu üzerinde olumlu etkilere sahip olduğu kabul edilmektedir (Garris, Ahlers ve Driskell, Games, 2002; Malone, 1981; Alessi ve Trollip, 2001). Bu motivasyonun temel kaynağı olan eğlence sayesinde, oyuncular kendileri için gerekli olabilecek bilgileri oyun oynama sırasında kazanabilirler (Pillay, 2002). Crawford (1984)'a göre oyun oynamanın eğitimsel bir değeri vardır. Eğitimsel değerin yanında tamamen eğitim amacıyla geliştirilmiş oyunlar da vardır. Bu tür oyunlar eğitsel oyunlar olarak ifade edilmektedir ve bu tür oyunlar üzerinde yapılan çalışmalar nispeten yeni olmakla birlikte giderek çoğalmaktadır (Üçgül, 2006).

Young (2004), literatürü inceleyerek eğitsel oyunların sınıf içinde önemli bir yer tutabilmesi için üç konuda değişiklik yapılması gerektiğini belirtmiştir:

1. Yapılandırmacı kurama uygun olarak test performansına dayanmayan, çok geniş kapsamlı bir müfredat yerine hedeflerin daha açık sunulduğu, anlamlı öğrenmeye yönelik bir yapı hazırlanmalıdır.
2. Psikomotor becerilere yönelik oyunlar yerine, sosyal ve bilişsel becerileri ön plana çıkararak MMORPG oyunlar geliştirilmelidir.
3. Proaktif öğretim tasarımları geliştirmek için oyunların öğrenme kuramı açısından anlaşılması gerekir.

Demirel, Seferođlu ve Yađcı (2003)'ya gre eđitsel oyunlar, “Oyun formatını kullanarak đrencilerin ders konularını đrenmesini sađlayan ya da problem zme yeteneklerini geliřtiren yazılımlardır.” Oyunlar problem zmenin zellikleri olan problemin yapılandırılması, alternatif zmler geliřtirilmesi, bilinmeyen sonuca ulařılması gibi zelliklerin yanında řans, yarıř, rekabet gibi zellikler de ierir (Bottino, Ferlino ve Travella, 2006; Ebner ve Holzinger, 2007).

Kirriemuir ve McFarlane (2004), oyunların stratejik dřnme, planlama, iletiřim ve karar verme gibi nemli kazanımlar sađladığını vurgulamaktadır (gl, 2006). Benzer řekilde Shaffer, Squire, Halverson ve Gee (2005) video oyunlarının yeni sosyal ve kltrel dnyalar yarattığını ve bunun da dřnme, sosyal iletiřim ve teknolojiyi birleřtirme yoluyla đrenmeye yardımcı olduđunu savunmaktadır. Bunun yanında Whelan (2005) Gee (2003), iyi tasarlanmış bilgisayar ve video oyunlarının gerek nemini oyunların iindeki sanal dnyalarda insanların kendilerini yeniden yaratması ve hem eđlenme hem de đrenmenin eř zamanlı olarak elde edilebilmesi olarak gstermektedir (Bakar, Tzn ve ađıltay, 2008).

Eđitsel bilgisayar oyunları zerine arařtırmalar yapan Linderoth, Lantz-Andersson ve Lindstrm (2002), oyunların okullardaki ders konularını đretmek zere  biimde kullanıldığını ortaya koymuřtur: Simlasyonlar řeklinde gerek hayatın elektronik ortamda canlandırılması, đrenciler iin bir motivasyon kaynađı olarak derse gdlenmenin sađlanması ve konuların sunumu iin bir ara olarak kullanılması.

Eđitsel bilgisayar oyunlarının bir diđer avantajı da ok kolay đrenilebilmeleridir. Siang ve Rao (2003)'e gre bilgisayar oyunları tm bilgisayar programları arasında en kısa đrenme eđrisine sahip olan programlardır. Ayrıca oyuncular, yardım dosyaları okumak ya da ynergeleri takip etmek yerine, dođrudan oyunu oynayarak, yařayarak đrenirler.

Gnmzde geliřtirilmiř birok eđitsel oyun ve bu oyunlar zerine yapılmıř birok arařtırma mevcuttur. đrencileri hedef alan ilk eđitsel bilgisayar oyunu 1971 yılında MECC arařtırma merkezi tarafından Oregon Trail ismiyle retilmiřtir.



Şekil 2.1: 1971 yılında geliştirilen Oregon Trail oyunu.

1973 yılında eğitsel bilgisayar oyunlarının matematik eğitimi amacıyla kullanıldığı Plato isimli bir proje geliştirilmiştir. Bu proje kapsamında geliştirilen bilgisayar oyunu öğrenciler üzerinde denenmiştir. Sonuç olarak matematik başarısında ve matematiğe yönelik öğrenci tutumlarında olumlu etkiler gözlenmiştir.

Yıllar geçtikçe bilgisayar oyunlarının eğitimde kullanımı ve bunun sonucu olarak öğrencilerin eğitsel bilgisayar oyunlarına olan ilgileri giderek artmıştır. 1982 yılında lise öğrencilerinin dijital mantık devreleri tasarlayabilmeleri için geliştirilmiş Rocky Boots isimli oyun 100.000 adet satılmıştır.

İlerleyen yıllarda eğitsel oyun sektöründe gerileme yaşanmış, oyun firmaları küçük bütçelerle kalitesiz, yeniliklerden uzak oyunlar geliştirmişlerdir. Bu dönemde eğitsel oyunların yerini, Sim City, Lemmings, Civilization gibi eğlence amaçlı oyunların eğitsel amaçlı kullanımı almıştır (Egenfeldt-Nielsen, 2005).

Prensky (2001)'nin aktardığına göre, William Stock, 400 farklı okulda araştırma yaparak eğitsel bilgisayar oyunu kullanan deney grubu öğrencilerinin kontrol grubu öğrencilerine göre standart testlerde %30 daha başarılı olduğunu bulmuştur. William Stock'a göre bu başarının en büyük sebebi, deney grubu öğrencilerinin ders dışı zamanlarda da eğitsel bilgisayar oyunlarını oynayarak öğrenme süresini arttırmış olmalarıdır.

Etkileşimli oyun tabanlı öğrenme aracı geliştiren Mann ve arkadaşları (2002), cerrahi yönetim algoritmasının öğretimi konusunda çalışma yapmışlardır. Çalışma sonrasında, öğrencilerin test puanlarında artış gözlenmiş ve öğrencilerin ortamı eğlenceli buldukları sonucuna ulaşılmıştır.

Bartholomew ve arkadaşları (2006), geliştirdikleri bilgisayar oyununu astım hastası öğrencilere hastalıkla mücadele etmeyi öğretmek için kullanmıştır. Araştırma sonucunda, oyun oynayan astım hastası öğrencilerin hastaneye kaldırılma sayılarında azalma olduğu, daha az sayıda hastalık belirtisine rastlandığı ve bu öğrencilerin günlük faaliyetlerini daha rahat yapabildikleri ifade edilmiştir.

Bilgisayar oyunları ile stratejik ve mantıksal düşünme beceresinin ilişkisinin araştırıldığı çalışmada Bottino ve arkadaşları (2006), oyunların öğrencilerin stratejik ve mantıksal düşünme becerileri üzerinde olumlu etkisinin olduğunu belirlemişlerdir.

Brom, Preuss ve Klement (2011), Çek Cumhuriyeti'nde gerçekleştirdikleri araştırmada 4 farklı liseden 5 sınıf ile çalışmıştır. Çalışmada normal eğitimin ardından ek olarak oyun ile eğitim gerçekleştirilmiştir. Oyunun amacı öğrenilenleri pekiştirmek ve birleştirmektir. Öncelikle her sınıf rastgele ikiye ayrılmıştır. Deney grubu Orbis Pictus Bestialis isimli oyunu oynarken kontrol grubu çoklu-ortam öğeleri barındıran ek bir ders almıştır. Çalışma sonunda; oyun oynayarak öğrenmenin geleneksel öğretimle karşılaştırıldığında pozitif etkilerinin olduğu, oyun oynamanın zararlı olmadığı ancak öğretimi ek olarak güçlendirmedeği ve oyun grubunun başarısının iyi olduğu ancak kontrol grubuna göre az miktarda düşük olduğu ve oyun oynayarak öğrenmenin öğrencilerin ilgisini çektiği belirlenmiştir.

Garris, Ahlers ve Driskell (2002)'in aktardığına göre, McDonald (1993) ve Ricci (1996) yaptıkları araştırma sonucunda eğitsel özellikleri olan bilgisayar oyunlarının öğrenmeyi iyileştirdiğini göstermişlerdir. McDonald (1993)'a göre eğitsel bilgisayar oyunları öğrencilerin daha fazla risk almalarına, risk alan öğrencilerin oyundaki görevlerinde daha ısrarlı davranmalarına sebep olmaktadır ve dolayısıyla öğrencilerin öğrenme performansları daha fazla gelişmektedir. Ayrıca, Ricci (1996)'ye göre eğitsel bilgisayar oyunları öğrencileri motive etmektedir. Motive olan öğrenciler eğitim verilen konuya daha fazla odaklanmakta ve öğrenme performansları artmaktadır.

Günümüzde oyun sektörünün büyümesi, geniş bant internetin yaygınlaşması ve bilgisayar teknolojisinin ilerlemesiyle birlikte eğitsel oyunlar da gelişmiştir. Artık eğitsel oyunlar birçok öğrencinin sosyal bir yapının parçası olarak birlikte eğlenmelerine ve öğrenmelerine imkân tanımaktadır.

Bilgisayar oyunları gerçek yaşam ile sanal ortamlar arasında bir köprü kurarak bireylerin eleştirel düşünceye sahip olmasını sağlamaktadır (Turvey, 2006). Öğrenciler oyunun kurallarını, oyun ortamında deneyerek ve aldıkları geribildirimlerle öğrenmektedirler (Funk, 2003).

Ural'a göre, özellikle formal eğitim dönemindeki bireylerin bilgisayar oyunları ile bu derece yoğun ilgilenmeleri bilgisayar oyunlarından eğitimde faydalanmanın önemini arttırmaktadır. Öğrencilerin bu derece isteyerek katıldıkları bir aktivitenin aynı zamanda onların bilişsel, duyuşsal, psikomotor gelişimlerini desteklemesi de sağlanırsa bilgisayar oyunları eğitimciler için bulunmaz bir fırsat niteliği taşıyabilecektir (Ünal, 2009).

#### **2.4 Çevrim İçi Oyun Tabanlı Öğrenme Ortamları**

Steinkuehler (2004), çok kullanıcı çevrim içi oyun ortamlarını çevrim içi oynanan, oyuncularca tasarlanan karakterlerin, diğer bir deyişle avatarların aracılığıyla oyuncuların hem oyunla hem de diğer oyuncularla iletişimlerine olanak veren, yüksek grafik çözünürlükte iki veya üç boyutlu oyunlar olarak tanımlamaktadır. Griffiths, Davies ve Chappell (2003)'e göre, çevrim içi ortamlar temelde daha ayrıntılı ve gelişmiş sanal dünyaları insanlara sunmak amacıyla tasarlanmışlardır. Çevrim içi ortamların en önemli özelliklerinden biri, farklı kültürlerde, farklı yerlerde yaşayan, farklı yaşlardaki kişileri aynı anda bir araya getirmesidir. Çevrim içi bilgisayar oyunları da bu özelliği kullanarak milyonlarca oyuncuyu aynı ortamda buluşturmaktadır (Doğusoy ve İnal, 2006).

Aynı zamanda çevrim içi oyunlar sosyal olarak da bireylerin etkileşime girmesini sağlamaktadır. Squire (2005), çevrim içi oyunların karşılıklı etkileşim sayesinde sosyal öğrenmenin de gerçekleşmesini sağladığını belirtmiştir.

Çevrim içi oyunların çeşitli demografik özelliklere sahip oyuncularının olması ve bu oyuncuların aynı anda aynı ortamda bulunması, oyuncular arasındaki etkileşim sayesinde sosyal bir yapı oluşturmaktadır. Bu oyunlar, sosyo-teknik yapıları sayesinde, oyuncuların sosyal olaylara yönelik sorumluluk bilincinin artırılması gibi amaçlar içermektedirler. Bunun yanı sıra, onların eğitim, eğlence ve

sosyal sorumlulukların birleştiği bir ortak noktada bütünleşmesine imkân tanımaktadırlar (Tüzün, 2004).

Günümüzde oyun geliştiricileri, sosyal olgular içeren oyunlara da diğer oyun türleri kadar ağırlık vermeye başlamıştır. Bu oyunlardan biri de 9-12 yaş arasındaki çocukların çok kullanıcı bir öğrenme ortamında eğitsel görevleri yerine getirdiği çevrim içi ortam olarak tasarlanmış Quest Atlantis'tir (Barab, Thomas, Dodge, Carteaux ve Tüzün, 2005).

Quest Atlantis öğrenmeyi, eğlenceyi ve macerayı bir araya getiren üç boyutlu (3D) sanal bir oyun ortamıdır. Atlantis şehrini yaklaşan felaketten kurtarmak için oyuncular, "Quest" adı verilen görevleri tamamlamak zorundadırlar (Tüzün, 2006). (Barab, Dodge, Thomas, Jackson ve Tüzün (2007)'e göre; Quest Atlantis olarak adlandırılan teknoloji zengini eğitsel yenilik, 3B sanal ortamın kullanıldığı, 4500'den fazla kullanıcıya hizmet eden bir öğrenme ve öğretme projesidir. Quest Atlantis oyunu, ilköğretim okulundaki öğrencilerin okul dışında ve okul içinde eğitsel etkinliklere sanal ortamlar üzerinden katılmalarını, diğer kullanıcılarla ve danışmanlarla iletişim kurabilmelerini ve sanal kişilik oluşturmalarını sağlar (Sert, 2009).

Quest Atlantis dışında az sayıda da olsa çeşitli çevrim içi oyun tabanlı öğrenme ortamı geliştirilmiştir. Bunlardan Talking Island, Mithril, LMS of Nori School ve Zon gibi oyunlar MMORPG türündedir.

Talking Island, Island Teknoloji tarafından Tayvan'da geliştirilmiş eğitsel bir MMORPG'dir. İlköğretim öğrencilerine İngilizce öğretmeyi amaçlamaktadır. New York şehrinin günlük yaşamının canlandırıldığı oyunda oyuncular kendi karakterlerini oluşturarak çeşitli görevleri (alışveriş listesini tamamlamak vb.) yerine getirmektedir. Görevleri tamamlayan öğrenciler, daha zor görevlerin olduğu üst seviyelere çıkmaktadır. Oyunda yer alan ses tanıma modülüyle öğrencilerin telaffuzları kontrol edilerek başarıları ölçülmektedir. Ayrıca öğrenciler ses ya da metin yoluyla diğer öğrencilerle iletişim kurabilmekte, görevleri işbirliği yaparak tamamlayabilmektedir (Hou, 2012).

Mithril, Kate Swanson ve Chris Piech tarafından 2009 yılında Stanford Üniversitesi'nde geliştirilmiştir. Bu oyunda öğrenme ortamları ile MMORPG türü

oyun ortamlarının bir araya getirilmesi amaçlanmıştır. Oyun, ilköğretim öğrencilerine matematik öğretiminde yardımcı olmayı hedeflemektedir. Mithril, büyümlü bir ormandaki küçük bir köyde geçmektedir. Oyuncular birbirleri ile işbirliği yaparak çeşitli bulmacaları çözmektedir (Swanson & Piech, 2009).

LMS of Nori School, ilköğretim öğrencilerine İngilizce öğretmek amacıyla Korea Game Development Institute tarafından 2004 yılında geliştirilmiştir. Oyun bir LMS (Öğrenme Yönetim Sistemi)'nin parçası olarak kullanılmıştır. Oyuncuların canavarlar avlayarak çeşitli nesnelere kazandığı ve seviye atladığı bir oyundur. Oyuncuların görevleri her gün yenilenmektedir. Yazma, okuma, konuşma ve dinleme sınavları ile oyuncuların İngilizce seviyeleri ölçülmektedir.

Zon, Michigan State Üniversitesi (2009) tarafından geliştirilmiş Çince öğretmeyi amaçlayan bir MMORPG'dir. Çin dilini ve kültürünü öğretmek amacıyla çevrim içi öğrenme ortamı tasarlanmıştır (Zhao, 2004).

Ayrıca henüz tamamlanmamış olmakla beraber DimensionU şirketi tarafından 2012 yılında "Oyna Kazan" (U game. U gain) sloganıyla geliştirilen Dimension U oyunu K-12 grubu öğrencileri hedef almaktadır (U Game. U Gain., 2012).

Çevrim içi oyun tabanlı öğrenme ortamları ile ilgili de bir dizi çalışma gerçekleştirilmiştir. Aşağıda bu çalışmalardan ve sonuçlarından bahsedilmektedir.

Dickey (2003), 3B dünyaların eğitimsel olarak güçlü ve zayıf yanlarını ortaya çıkartmak için yaptığı araştırmada, 3B ortamların yapılandırmacı öğrenme ortamını desteklediği, uzaktaki öğrenenlere önemli fırsatlar sunduğu, sohbet aracını kullanmanın öğrenmeyi pekiştirdiği, ortamın sınırlılıklarının öğrenmeyi engellemediği ve bu ortamın uzaktan öğrenmede etkili bir ortam olduğu sonuçlarına ulaşmıştır.

Quest Atlantis'in Fen Bilgisi dersindeki çekiciliğini inceleyen Lim, Nonis ve Hedberg (2006), ortamın dersi sıkıcılıktan kurtarıp daha eğlenceli bir hale getirdiği, öğrencilerin derse karşı daha motive oldukları sonuçlarına ulaşmıştır. Ayrıca çalışmada, zaman yetersizliği ve sahiplenmeme faktörlerinin Quest Atlantis'in Fen Bilgisi dersine entegrasyonunu engellediği belirtilmiştir.



Quest Atlantis ortamında yapılan bir diğerk çalışmada Tüzün (2007), örgün eğitim ortamlarında bilgisayar oyunlarını kullanmanın önemli noktalarını ve zorluklarını arařtırmıřtır. Çalışmada, ortam tasarlanmanın uzun zaman alması, konuyu anlatmak için oluşturulan hikâye ile ortamın uyumunda zorluklar bulunduđu, öğrencilerin ticari oyunlardaki tecrübelerine göre ortamdaki beklentilerinin yüksek olduđu, ortama uyum sürecinin ve tasarımın dersin kavramsal çerçevesini kısıtladıđı gibi sonuçlar elde edilmiştir. Ayrıca, öğrenciler ortamı eğlenceli bulmuşlar, iş birliđi içinde ipuçlarını beraber deđerlendirerek verilen görevleri yerine getirmeye çalışmışlardır.

Sert (2009) tarafından, Quest Atlantis ortamında hazırlanan eğitsel bir bilgisayar oyununun “Bilgi ve İletişim Teknolojileri” dersinde bulunan İnternet konusunda öğrenci başarısına etkisi incelenmiştir. Lise 2. sınıflarla yapılan çalışmada, oyuncuların öğrenme stillerindeki farklılıkların başarıyı etkileyip etkilemediđi arařtırılmıştır. Arařtırma sonucunda deney ve kontrol grubu öğrencilerinin başarısında fark gözlenmemiştir. Oyun tabanlı ortamı kullanan deney grubu öğrencilerinin öğrenme stillerine göre başarılarında da fark bulunmamıştır.

Quest Atlantis ortamında, ilköğretim öğrencilerinin bilgisayar dersindeki başarılarını ve öz yeterlik algılarını inceleyen Yağız (2007), deney grubu öğrencileri ile kontrol grubu öğrencileri arasında anlamlı bir fark olmadığını ifade etmiştir. Ancak, oyun tabanlı öğrenme ortamının, öğrencilerin hoşuna gittiđini, kaygılarını azalttıđını, bireysel olarak öğrenmelerine yardımcı olduđunu ve öğrenmeyi görsel olarak desteklediđini belirtmiştir.

Güneş (2010), elektroGame ortamında ilköğretim Bilişim Teknolojileri dersi 4. basamak “Bilgilerimi Neden Unutuyorum?” ünitesinde öğrenci başarıları ile bilgisayar tutumlarının ve bilgisayar kaygılarının başarıya etkisini incelemiştir. Arařtırma sonucunda, deney ve kontrol grubu öğrencilerinin ders başarıları arasında anlamlı bir ilişki gözlenmemiştir.

Hwang, Wu ve Chen (2012), web tabanlı problem çözme etkinlikleri için çevrim içi bir oyun geliřtirmiştir. Oyunda, zar yardımıyla hamle yapan oyunculardan, yol üstündeki alanlarda verilen görevleri gerçekleřtirmeleri istenmektedir. Oyuncular, görevlerde öğrenme hedefleriyle ilgili bir dizi soru cevaplamak

zorundadır. Çalışma, Tayvan'da bir ilkokulun fen bilimleri kursunda 5. ve 6. sınıf düzeyinde 50 öğrencinin katılımıyla gerçekleştirilmiştir. Ön-test sonuçlarına göre deney ve kontrol grupları arasında anlamlı bir fark bulunmazken son-test sonuçlarında deney grubunun lehine anlamlı bir fark bulunmuştur.

Bunların dışında oyuncuların yeni bir simülasyon ortamını mobil teknolojileri kullanarak incelemesine izin veren 'Environmental Detectives' platformu da sosyal öğrenme ortamlarına bir örnek olarak verilebilir (Klopfer, Squire ve Jenkins, 2002)

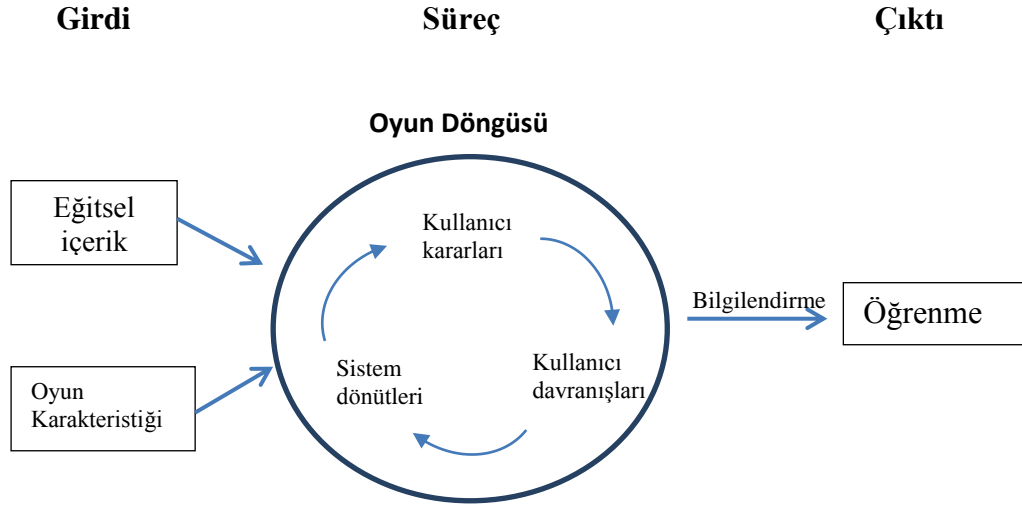
Çevrim içi oyunları geliştirmek, yüksek maliyet ve büyük bir iş gücü gerektirdiği için yeni eğitsel çevrim içi oyunlar geliştirmek yerine çevrim içi bilgisayar oyunlarının öğrenme ortamlarına uyarlanması günümüzde oldukça önemli hale gelmiştir (Pivec, Dziabenko ve Schinnerl, 2003). Civilization III (Squire, 2005), ve Battle Field (Manninen ve Kujanpää, 2007) gibi çok beğenilen oyunlar çeşitli şekillerde eğitsel amaçlı olarak kullanılmıştır. Ayrıca bu yöntemin bir avantajı da bu oyunlara olan ilgileri sayesinde öğrencilerin, oyun esnasında karşılaştıkları problemlere çözüm üretmek adına birden fazla strateji geliştirebilmeleridir (Pivec, Koubek ve Dondi, 2004).

## **2.5 Eğitsel Bilgisayar Oyunu Geliştirilmesi**

Önceleri bilgisayar oyunları sadece öğrencilerin dikkat ve motivasyonunu yükselten birer ders materyali olarak kullanılırken, artık dersin ana materyali olarak kullanılması daha çok tercih edilir hale gelmiştir. Eğitsel amaçlar ve oyun oynama boyutlarının her ikisi de dikkatlice analiz edilerek, aradaki dengenin etkili biçimde kurulması sağlanmalıdır (Kiili, 2005a). Bu durum göz önüne alınarak, bilgisayar oyunlarının tasarım aşamasında, oyundaki seviyelerin daha dikkatli tasarlanması gerekmektedir (Pelletier, 2005).

Eğitsel bilgisayar oyunu geliştirebilmek için, oyuncuların oyun ortamında çevreyle nasıl etkileşime geçtikleri, geri bildirimleri nasıl değerlendirdikleri ve nasıl öğrendiklerinin anlaşılabilmesi oldukça önemlidir (Ang & Rao, 2008).

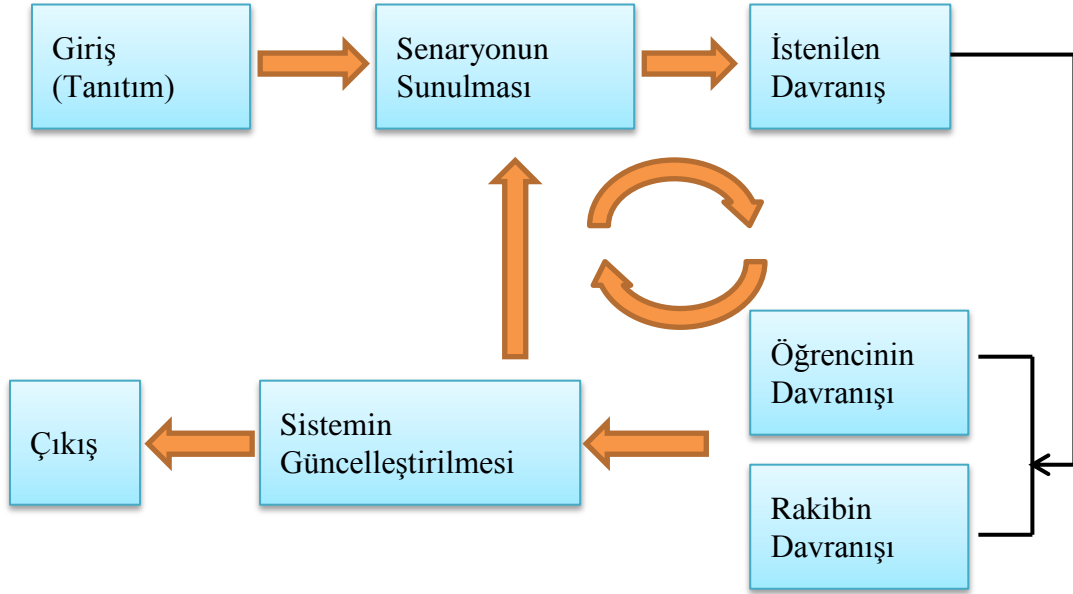
Garris, Ahlers ve Driskell (2002), oyun yoluyla öğrenmenin nasıl olacağına dair bir model ortaya koymuştur. Bu çalışmada önerilen oyun tabanlı öğrenme modeli **Şekil 2.2**'deki gibidir



**Şekil 2.2:** Girdi-süreç-çıkı oyun modeli.

Garris'in oyun tabanlı öğrenme modeline göre, oyun başladığında oyun döngüsü de başlar. Oyun döngüsüne oyun karakteristiği ve öğretim içeriği birlikte dâhil olur. Oyuncu oyun döngüsü içinde oyunun yapısını keşfeder ve öğretimsel içerikle karşılaşır. Oyun tamamlandığında bilgi alınmış ve öğretim gerçekleşmiştir (Garris, Ahlers & Driskell, 2002).

Eğitsel bir bilgisayar oyunu geliştirilirken oyun tabanlı öğrenmenin nasıl gerçekleştiği bilgisi ışığında, eğitsel yazılım geliştirme basamaklarını izlemek faydalı olacaktır. Ancak eğitsel bir oyun geliştirirken eğitsel öğeler ile oyunu bir araya getirmek karmaşık bir işlemdir (Güneş, 2010). Oyun geliştirilmeden önce eğitsel bir oyunun yapısının nasıl olması gerektiği bilinmelidir. Eğitsel oyunların yapısı ile ilgili olarak literatürde çeşitli çalışmalar bulunmaktadır. İpek (2001), eğitsel bir oyunun yapısının **Şekil 2.3**'teki gibi olması gerektiğini belirtmiştir.



**Şekil 2.3:** İsmail İpek, Eğitsel Oyun Yazılımlarının Genel Yapısı.

Bu eğitsel oyun yapısında oyuncu oyuna başladığında, ilk olarak oyunun tanıtımı ile karşılaşır. Burada oyunun oynanışı hakkında genel bilgilere sahip olan oyuncu, bir sonraki adımı oluşturan senaryonun sunulması aşamasına geçerek oyunun hikâyesini ve amacını öğrenir. Oyundaki öğelerle ve rakipleriyle etkileşimde bulunarak geri bildirimler alır. Oyunun her basamağında oyuncunun yaptığı işlemler listelenerek durumu oyuncuya bildirilir.

Bu genel eğitsel oyun tasarım modelindeki temel yapı, farklı araştırmacılar tarafından daha detaylı ve kapsamlı olarak literatürde incelenmiştir. Aşağıda çoğunlukla karşılaşılan oyun tasarım modellerinden bahsedilmektedir.

### 2.5.1 EFM: Eğitsel Oyun Tasarımı için Bir Model

EFM, etkili öğrenme ortamı (**E**ffective Learning Environment), akış deneyimi (**F**low Experience) ve motivasyon (**M**otivation) kelimelerinin baş harfleri ile isimlendirilmiş bir modeldir. Bu tasarım modelinde adı geçen üç kuramın özellikleri birleştirilerek eğitsel oyun geliştirmek için bir yapı sunulmuştur.

EFM Modeli, temelinde bulunan motivasyonu sağlamak için ARCS Motivasyon Modelini kullanır. ARCS Motivasyon Modeli Keller tarafından Beklenti Değer Teorisine (Expectancy-Value Theory) dayanarak geliştirilmiştir. Öğretim

boyutunda motivasyon faktörünün dikkate alındığı, hatta merkeze konularak sunulduğu bir modeldir. Bu model, eğitimcilere öğrencilerin derse karşı meraklarının uyandırılması ve sürdürülmesi için imkânlar sağlamaktadır (Dede, 2003).

ARCS Motivasyon Modeli (Keller, 1984), dört ana öğrenme modeli üzerine odaklanmaktadır. Bunlar (Keller & Suzuki, 1988; Bayram, 1999):

1. Dikkat (Attention)
2. Uygunluk (Relevance)
3. Güven (Confidence)
4. Doyum (Satisfaction)

şeklinde. **Tablo 2.4**'te ARCS Modelinin bu dört ana faktörü ve her faktörün alt basamakları listelenmiştir.

**Tablo 2.4:** ARCS Motivasyon Modeli Stratejileri.

DİKKAT (Attention)	UYGUNLUK (Relevance)	GÜVEN (Confidence)	DOYUM (Satisfaction)
Algısal Uyarılma	Yakınlık	Başarı Beklentisi	Doğal Sonuçlar
Araştırmaya Yönelik Uyarılma	Hedefe Yönelme	Güç Sınama	Olumlu Sonuçlar
Değişkenlik	Güdü Uygunluğu	Destekleme	Eşitlik

EFM Modeli, geliştirilen öğretim tasarımı ile ARCS Motivasyon Modeli arasındaki bağlantıyı kurabilmek için Chicago Üniversitesi'nden Csikszentmihalyi tarafından geliştirilen Akış Deneyimi Kuramını (Flow Experience) kullanmaktadır.

Akış Deneyimi Kuramı, motivasyonu anlamak ve uygulamak için, öğretim tasarımı ve motivasyon kuramı arasında teorik bir köprü oluşturmaktadır. Bu köprü dokuz adımdan oluşmaktadır (Csikszentmihalyi, 1991):

- Faaliyetin amaçları
- Belirsiz olmayan dönütler
- Mücadele beceri dengesi
- Konsantrasyon
- Kontrol
- Birleştiren çalışma farkındalığı
- Amacı olan deneyim
- Zamanın dönüştürülmesi
- Özbilinç kaybı

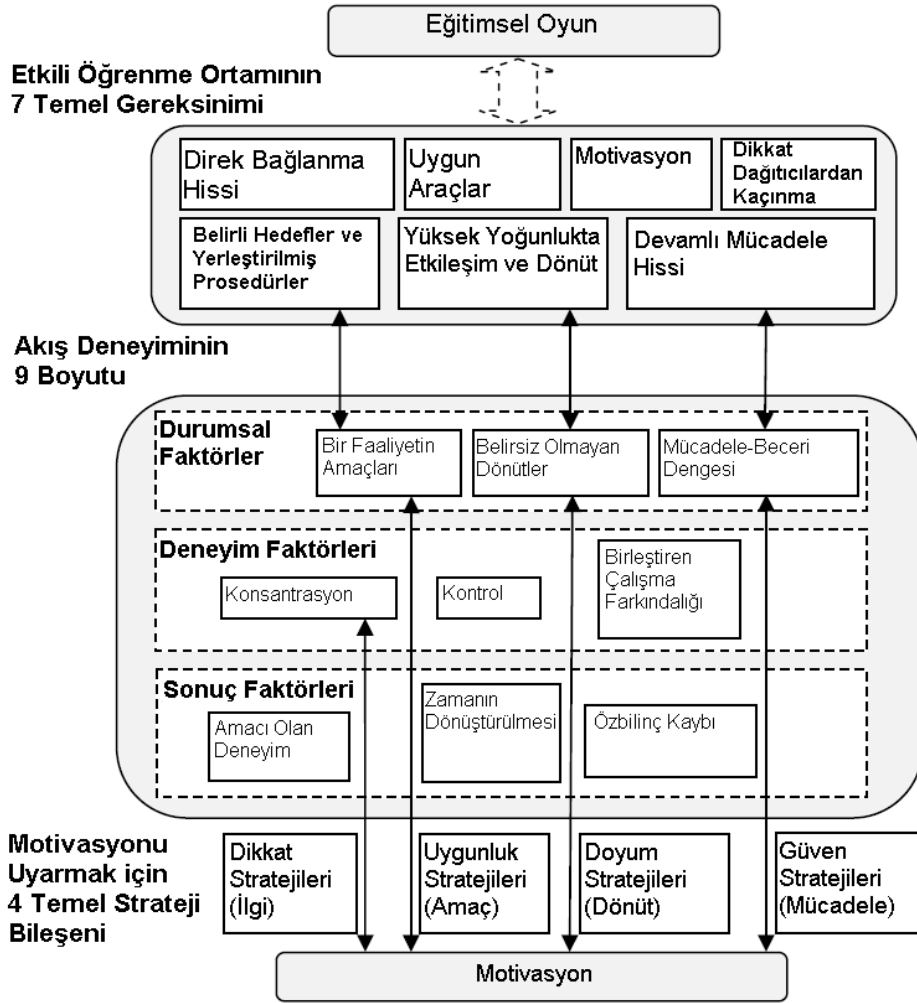
Ayrıca Novak ve diğerleri (2000), dokuz adımı üç kategori altında toplamışlardır:

- Durumsal faktörler
- Deneyim faktörleri
- Sonuç faktörleri

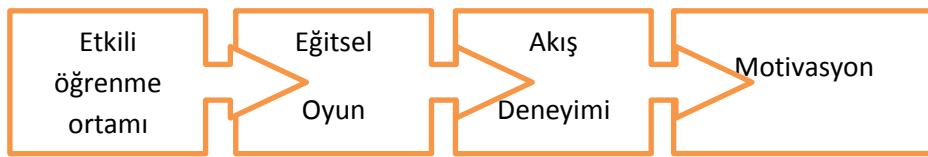
Etkili öğrenme ortamı tasarımı, öğrencilerin anlamalarına, ana içeriği öğrenmelerine ve kendi bilişsel yeteneklerini arttırmalarına yardımcı olmayı sağlayacak etkili ve pozitif öğrenme ortamları sağlamayı amaçlamaktadır. Bu ortamlar tasarlanırken dikkat edilmesi gereken kurallar, yedi temel basamak şeklinde aşağıdaki gibidir:

- Yüksek yoğunlukta etkileşim ve dönüt
- Belirli hedefler ve yerleştirilmiş prosedürler
- Motivasyon
- Devamlı mücadele hissi
- Direk bağlanma hissi
- Uygun araçlar
- Dikkat dağıtıcılardan kaçınma

Bütün bu kuramlar birlikte değerlendirildiğinde; motivasyon, akış deneyimi, etkili öğrenme ortamı ve eğitsel oyunlar arasındaki bağlantı **Şekil 2.4**'te ve **Şekil 2.5**'te gösterilmiştir (Song ve Zhang, 2008; Akgün, Nuhoğlu, Tüzün, Kaya ve Çınar, 2011).



Şekil 2.4: Motivasyonu, akış deneyimini, etkili öğrenme ortamı ve eğitsel oyun arasındaki bağlantı.



Şekil 2.5: Eğitsel oyun tasarımı için EFM modeli.

Sonuç olarak EFM Modeline göre, öğrenen etkili bir öğrenme ortamında akış deneyimini yaşadığında, bu ortam öğreneni uyararak kesin olarak bir motivasyonu sağlayacaktır. Bu açıdan bakıldığında iyi tasarlanan bir eğitsel oyun, motivasyonu sağlamak ve öğrenmeyi gerçekleştirmek için etkili bir öğrenme ortamı olabilmektedir.

### 2.5.2 FIDGE Modeli

FIDGE, oyun benzeri ortamlar için bulanıklaştırılmış öğretim tasarımı geliştirme (**F**uzzified **I**nstructional **D**esign **D**evelopment of **G**ame-like **E**nvironments) kelimelerinin baş harfleriyle isimlendirilmiş bir oyun tasarım modelidir.

Oyunlarla öğrenme ortamlarının nasıl birleştirilmesi gerektiğini açıklamaya çalışan kuramlardan biridir. Genel tasarım modellerinde bulunan analiz, tasarım, geliştirme, uygulama ve değerlendirme basamaklarına ek olarak bir ön analiz basamağına sahiptir. Modelin temelleri bulanık mantık göz önüne alınarak şekillendirilmiştir. Modelin geliştirilmesi esnasında gerçek hayat deneyimlerinden elde edilen veriler kullanılmıştır. Modelin aşamaları şöyle özetlenebilir:

**Ön Analiz Aşaması:** Bu aşamada değişebilir bir hedef grup belirlenir, hedef grup için onların önceki deneyimlerine bağlı olarak hedef konu seçilir. Küçük çaplı bir literatür incelemesi yapılarak seçilen konunun oyun benzeri öğrenme ortamı oluşturmak için uygun olup olmadığı araştırılır. Seçilen konuya göre belirlenen hedeflerin değişebilirliği gözden geçirilir. Konu alan uzmanlarının ve hedefi temsil eden grupların görüşme yoluyla görüş ve önerileri alınır. Yazılım geliştirme araçları araştırılır ve analiz edilir. Farklı oyunlar analiz edilir.

**Analiz Aşaması:** Bu aşamada, ihtiyaç, öğrenen, içerik, amaç ve duruma göre maliyet, risk analizleri yapılarak ön analiz kısmında başlanan oyun analizi derinleştirilir.

**Tasarım ve Geliştirme Aşaması:** Bu aşamada senaryolar belirlenerek içerikle ilgili uzman görüşü alınır. Motivasyon, geri bildirim, değerlendirme bileşenleri oluşturulur. Destek belgeleri ve veri toplama araçları hazırlanır.

Değerlendirme aşamasında konu alanı uzmanları ve hedef kitleyle görüşülerek biçimlendirici değerlendirme yapılır. Değerlendirme sonucunda elde edilen bulgulara göre gerekli düzeltmeler yapılır (Akıllı ve Çağıltay, 2006).

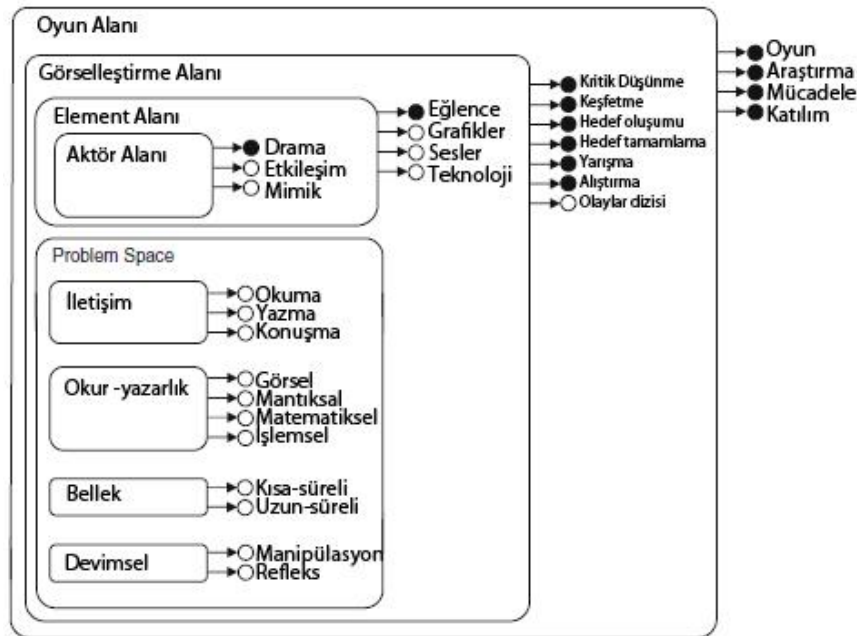


### 2.5.3 Oyun Nesnesi Modeli

Oyun Nesnesi Modeli (GOM - Game Object Model) ilk kez Amory tarafından 1999–2001 yıllarında tanımlanmıştır. GOM, nesne yönelimli programlamayı temel alarak öğrenmenin pedagojik boyutları ile oyun elemanları arasındaki ilişkiyi açıklamaya çalışan bir modeldir. Sistem, miras alma, çok biçimlilik ve kapsülleme gibi nesne yönelimli programlama öğelerini içerir.

Modelin ilk sürümü 2001 yılında sunulmuş, modelin kullanılmasıyla birlikte eksikliklerin değerlendirilip giderilmesiyle ve bakış açısındaki değişimler ve gelişmeler ile beraber 2007 yılında ilk sürümü de kapsayan daha geniş bir model haline getirilmiştir. Amory (2007)'e göre eğitsel oyunlar, bağlama uygun, araştırmaya yönelten, heyecan verici ve katılımı destekleyen, karmaşık mücadelelere verilen cevapların çeşitli zor diyaloglar gerektiren ortamlar sunmalıdır (Amory, 2007; Akgün ve Ark., 2011).

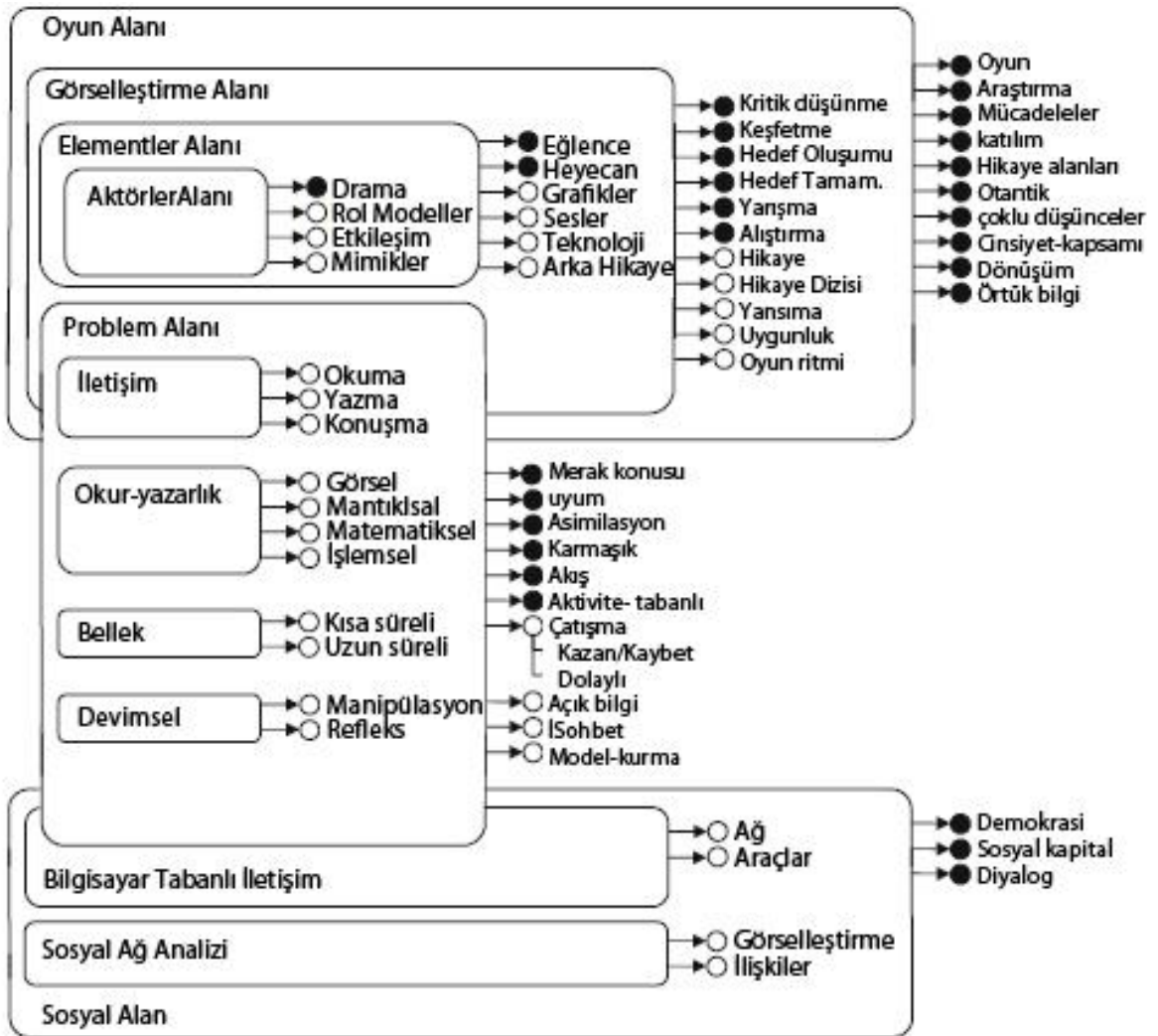
Modelin şematik gösteriminde yuvarlatılmış dikdörtgenler ile içi dolu ve boş daireler kullanılmıştır. Bu modele göre, eğitsel oyun somut ve soyut ara yüzler aracılığıyla açıklanan bileşenlerden oluşmaktadır. Somut ara yüzler eğitsel oyunun pedagojik ve teorik yapılarına; soyut ara yüzlerse tasarım bileşenlerine karşılık gelmektedir.



Şekil 2.6: Oyun Nesnesi Modeli (Amory ve Seagram 2003; Akt. Amory 2007)

Modelde içerdeki nesnelere genellikle somut ara yüzleri kapsarken, soyut ara yüzleri ifade eden içi dolu çemberler modelin dış kısımlarında konumlandırılmıştır.

Oyun Nesnesi Modeli akademik macera oyunlarının tasarımında başarı ile kullanılmıştır (Amory, 2007). Oyun Nesnesi Modelinin ikinci sürümü tanım, otantik öğrenme, hikâye, cinsiyet, sosyal işbirliği, mücadele olmak üzere altı noktada oyun tasarımını tartışmaktadır. Oyun Alanı, Görselleştirme Alanı, Elementler Alanı, Aktörler Alanı ve Problem Alanları korunurken; Sosyal Alanı bünyesine eklemiştir.

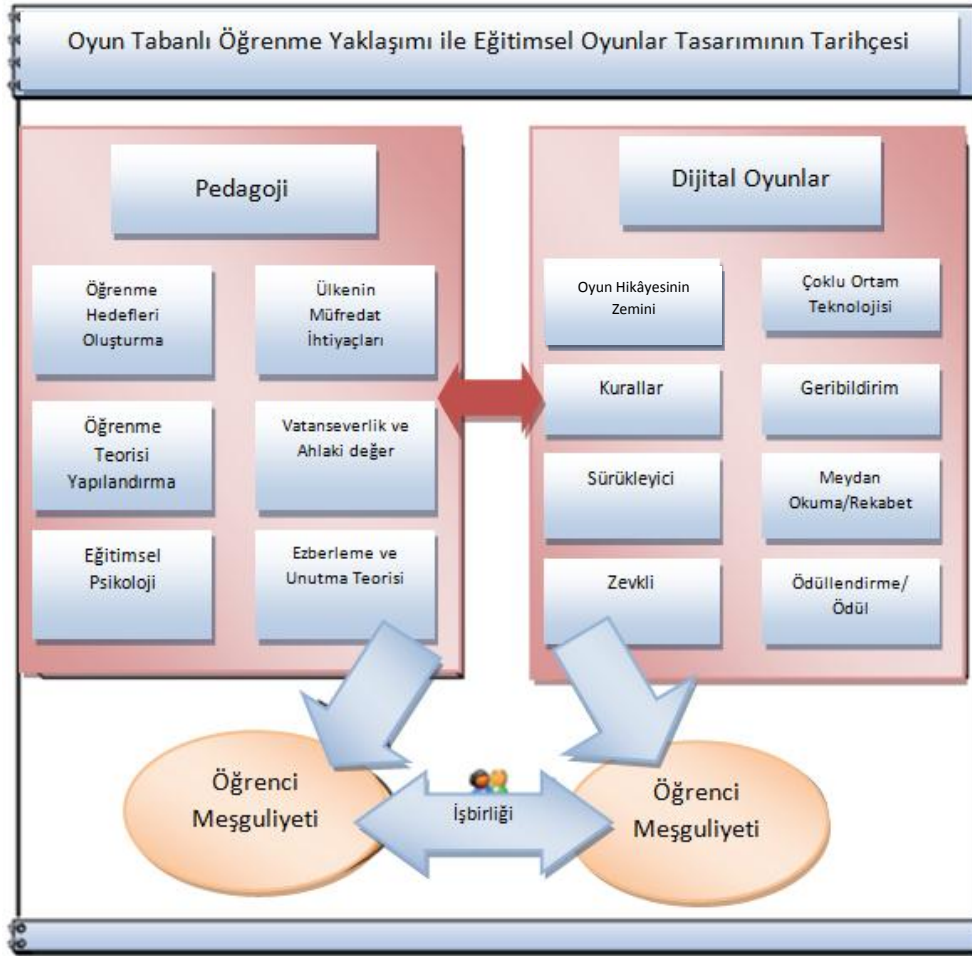


Şekil 2.7: Oyun Nesnesi Modeli II (Amory, 2007).

GOM II, birbirleriyle ilişkili nesnelere oluşmaktadır. Bu nesnelere mücadele, hikâye, sohbet olmak üzere eğitsel oyunları açıklamak için üç ana alan olarak gruplanmıştır.

## 2.5.4 Dijital Oyun Tabanlı Öğrenme Modeli

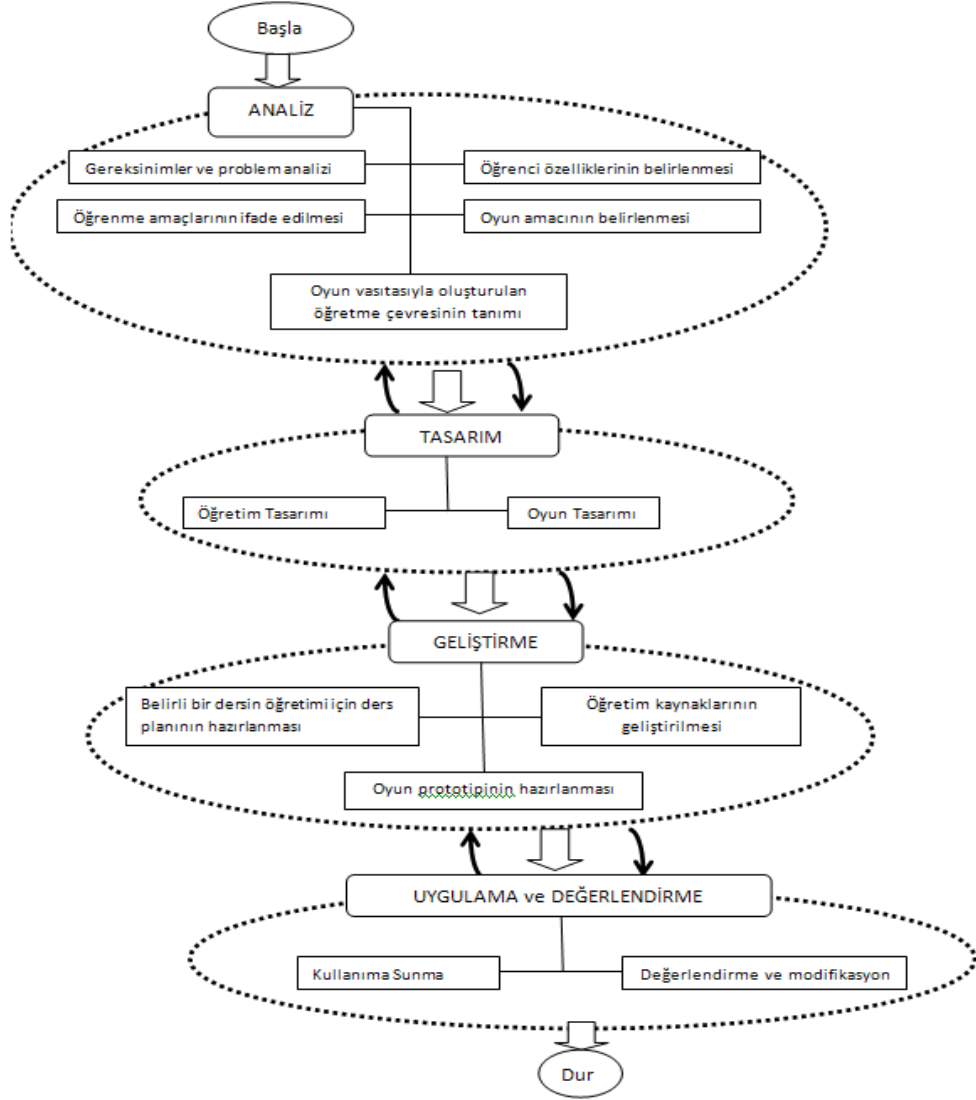
Dijital Oyun Tabanlı Öğrenme Modeli (DGBL - Digital Game Based Learning) oyun tasarım modelinde oyun, öğrenme içeriklerinin yerleştirildiği bir ortam olarak ele alınmıştır (Zin, Jaafar ve Yue, 2009). Öğrencilerin tarih dersine yönelik motivasyonunu arttırmak için çeşitli simülasyonlar kullanarak öğrenciyi, hazırlanan tarihsel ortamın içinde hissettirmek amacıyla yapılan çalışmalar sonucunda DGBL Modeli ortaya çıkmıştır.



Şekil 2.8: Eğitsel tarih oyunlarının tasarımı için DGBL Modelinde önerilen bileşenler.

Model, tarih öğretimi için geliştirilecek bir oyunun, pedagojik bölüm ve dijital oyun bölümü olmak üzere iki temel bileşenden oluşması gerektiğini belirtmektedir. DGBL Modeli, bu bileşenler dikkate alınarak tasarım için dört basamaklı bir yapı sunmaktadır. Genel tasarım modellerine benzer olarak analiz,

tasarım, geliştirme, kalite kontrolü, uygulama ve değerlendirme basamaklarından oluşmaktadır. Modelin basamakları ve her basamağın alt görevleri Şekil 2.9’da verilmiştir (Zin, Jaafar ve Yue, 2009; Akgün ve Ark., 2011).

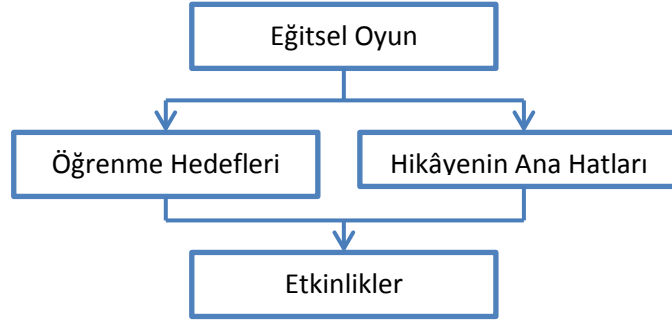


Şekil 2.9: DGBL Modeli.

### 2.5.5 Oyun Meydana Getirme Modeli

Oyun Meydana Getirme Modeli (Game Achievement Model - GAM), GOM’un eğitsel oyun tasarlamak ve üretmek için açıkça belirtmediği bir çerçeve sunmak üzere geliştirilmiştir. GOM ara yüzünün çerçevesi, oyunun kurgusudur ve bundan dolayı bu ara yüz oyun geliştirmenin temelini oluşturmalıdır. Şekil 2.10’da öncelikli öğrenme hedefleri, hikâye özeti, etkinliklerin tasarlanmasında nasıl

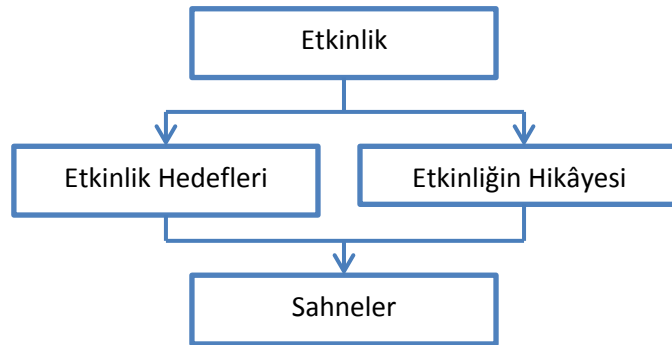
kullanılması gerektiği gösterilmiştir. Bu etkinlik ya da olaylar GAM'ın başlangıç noktasını oluştururlar (Amory ve Seagram, 2003).



**Şekil 2.10:** GAM'a göre eğitsel oyunun etkinliklerini oluşturan iki temel faktör.

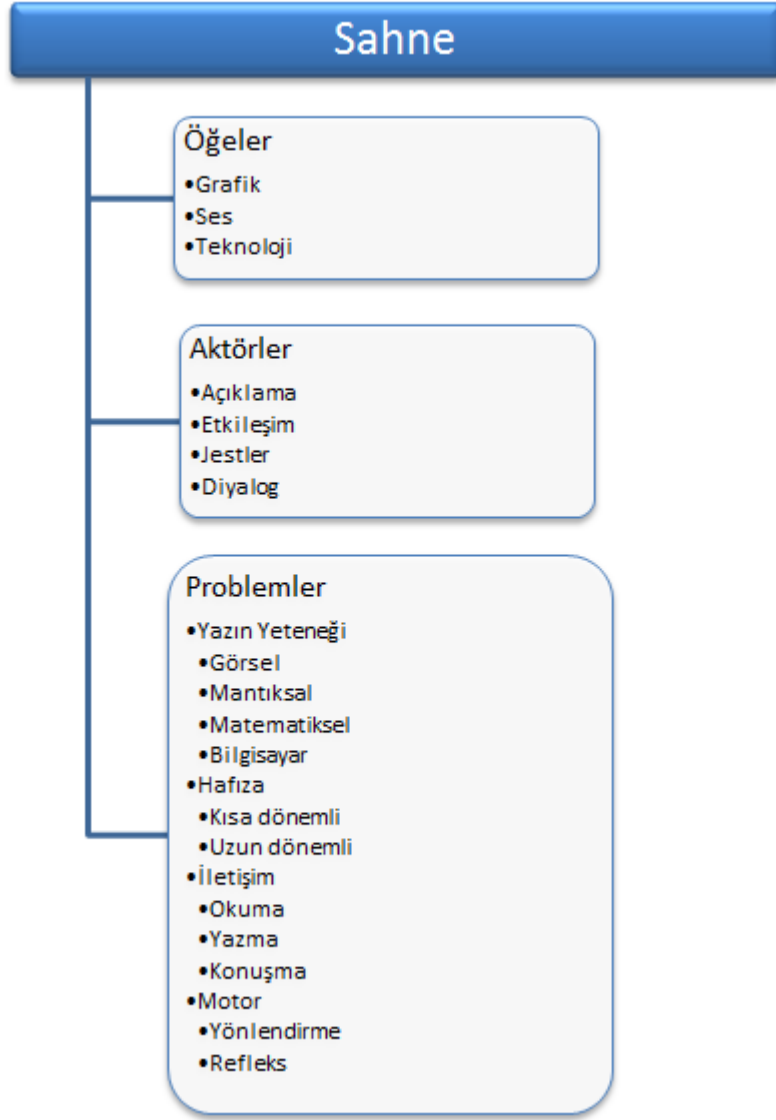
Bu modele göre, bilgisayar oyunları da tıpkı filmlerde olduğu gibi bir dizi etkinlikten oluşmalıdır. Ancak bir film içinde etkinlikleri ayırt etmek zor iken geleneksel oyunlarda etkinliklerin ayrımı kolayca fark edilebilir.

**Şekil 2.11'**de her eylemin bir amacı ve hikâyenin bir parçası olduğu ve sahnenin bunlara bağlı olarak tasarlanması gerektiği gösterilmiştir.



**Şekil 2.11:** Oyun içindeki sahnelerin oluşturulması sürecindeki bileşenler.

Bir sahneyi oluşturan üç alan bulunmaktadır: Öğeler, Oyuncular ve Problemler. **Şekil 2.12'**de sahneyi oluşturan bu üç alan, alt özellikleriyle birlikte gösterilmiştir (Amory ve Seagram, 2003).



Şekil 2.12: Sahneyi oluşturan üç alan ve alt öğeleri.

Model, eğitsel bir oyun geliştirmek için gerekli olan yaratıcılık gibi yetenekleri yok saymadan hikâye anlatımını ve grafik tasarımı içeren iki yönlü kavramsal bir çerçeve sunmaktadır (Amory ve Seagram, 2003).

### 2.5.6 Deneyimsel Oyun Modeli

Kiili (2005a) tarafından, oyun tasarım sürecine eğitsel kuramları entegre ederek eğitsel oyun tasarımcılarına, oyunlarla öğrenme mekanizmasını anlamada yardımcı olması için Deneyimsel Oyun Modeli (EGM- **Experiential Gaming Model**) önerilmiştir. Modelin Kiili (2005b) tarafından geliştirilmiş ikinci bir sürümü

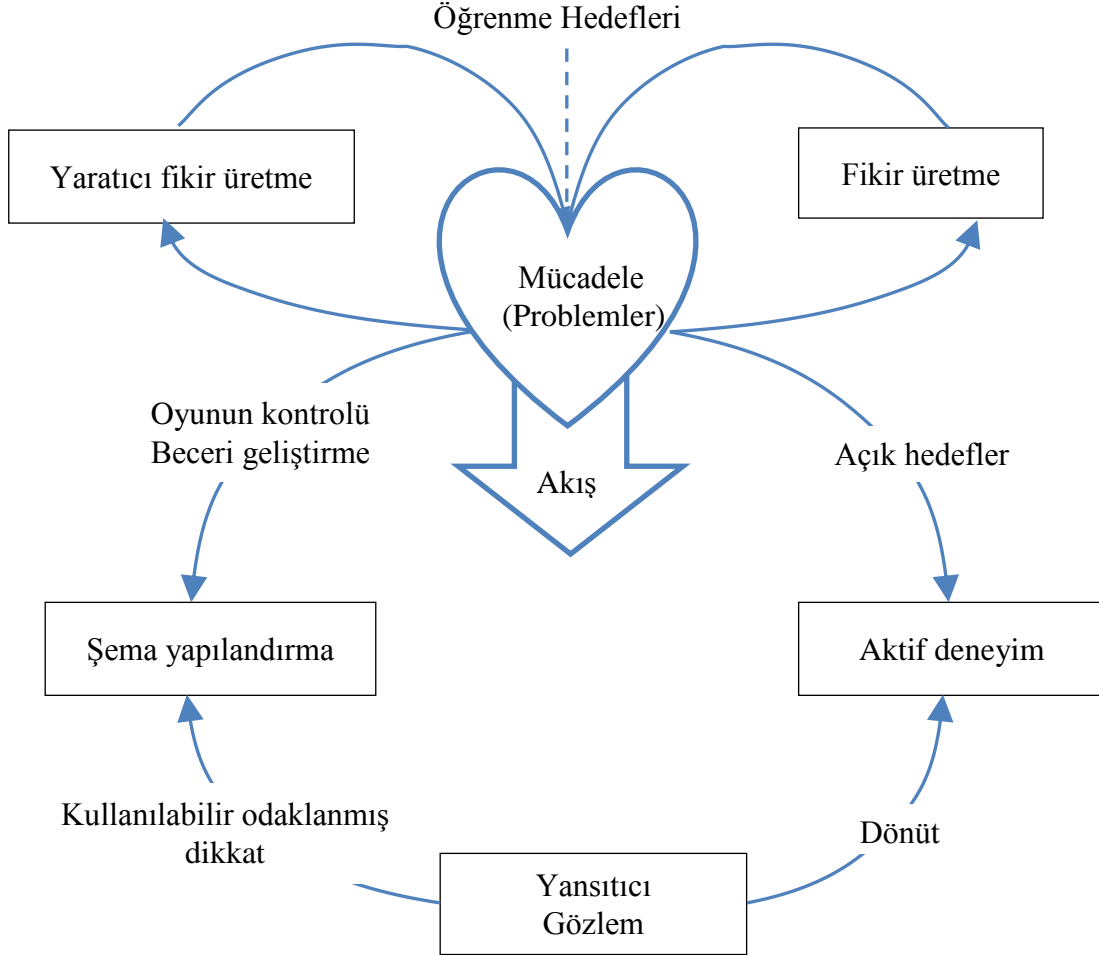
bulunmaktadır. EGM, Csikszentmihalyi'nin (1975) akış teorisini ve oyun tasarımını temel almaktadır.

Modelin amacı, oynanış ile deneysel öğrenmeyi ilişkilendirerek akış deneyimini kolaylaştırmaktır. Model, öğrenmeyi deneyim yoluyla doğrudan oyun dünyası içinde döngüsel bir süreç olarak tanımlar. İkinci sürümde bu döngüsel süreçte bir de tasarım döngüsü eklenmiştir.

Kiili, öğrenmenin sadece bilişsel değil, davranışsal da olduğunu vurgulayarak bilişsel yapıların oyun dünyası içinde yapılan etkinlik ve pratiklerle desteklenmesi gerekliliğini ortaya koymuştur. Deneysel Oyun Modeline göre, oyuncular akış deneyimi yaşadıklarında oyuna daha çok bağlanmaktadır. Bu nedenle, Deneysel Oyun Modeline göre, oyunlar tasarlanırken akış deneyimine katkıda bulunacak şekilde tasarlanmasının önemi vurgulanmaktadır.

Model, fikir üretme döngüsü, deneyim döngüsü ve mücadele öğelerini içerir. Modelin yapısı **Şekil 2.13**'teki gibi insan dolaşım sistemine benzetilerek geliştirilmiştir. Eğitim amaçlarına dayalı zorluklar, modelinin kalbini oluşturmaktadır. Kalp, oyuncuya çeşitli mücadele alanları ve problemler yaratarak motivasyonu sağladığı gibi, oyuncunun oyuna katılımını da artırır. Oyuncu mücadele sırasında karşılaştığı zorlukları aşmak için döngü içinde çözümler üretir. Burada oyuncu, öncelikle hiçbir kısıtlama olmaksızın birincil, yani yaratıcı fikirler geliştirir. Birincil fikirlerin, çözümü içeren fikirlere dönüşmesiyle küçük dolaşım tamamlanmış olur. Model'e göre, fikir üretme süreci gruplar halinde gerçekleştirildiğinde daha verimli olmaktadır. Fikir üretme aşamasından sonra, oyuncu ürettiği çözümleri büyük dolaşıma karşılık gelen deneyim döngüsünde uygulayarak dener ve eylemlerinin sonuçlarını gözlemler. Deneyim döngüsünde, oyunun kullanılabilir ve açık hedefler sunması gerekir. Ayrıca, oyun akış deneyimini kolaylaştırabilmek için uygun ve belirsiz olmayan geri bildirimler sağlamalıdır. Buna ek olarak odaklanmış dikkat akış deneyiminde önemli bir faktördür. Geri bildirimlerin oluşturacağı yansıtıcı gözlemler, problemlere daha iyi çözümler geliştirilmesini ve yeni şemaların oluşturulmasını sağlayabilir. Çözümlerin denenmesi aşamasında oyuncunun, oyun ve konu üzerindeki kontrol becerisi gelişir ve çözüme ilişkin şema güçlendirilir. Büyük dolaşım da böylece tamamlanmış olur.

Kalp, her oyuncunun beceri düzeyine uygun olarak farklı zorluk seviyeleri sağlamalıdır. Ayrıca oyunun zorluk düzeyleri dengeli olmalı ve bazen çok zor bazen çok kolay olmamalıdır.



Şekil 2.13: Deneyimsel Oyun Modeli (Kiili, 2005a).

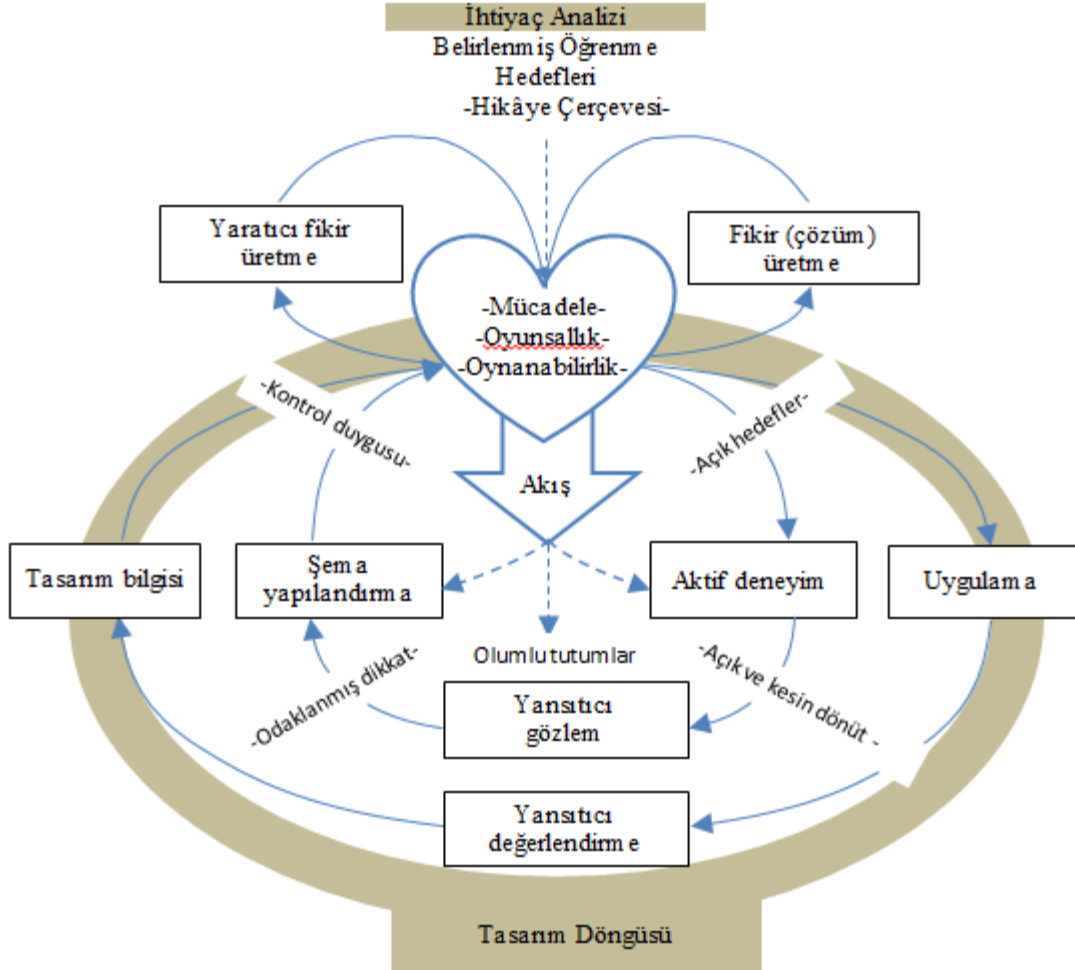
Genel olarak, kalbin amacı oyuncuya mücadele edeceği bir alan sunarak fikir ve deneyim döngüsüne girişi sağlamaktır. Fikir döngüsünün amacı yeni ve yaratıcı fikirler ortaya çıkarmak, deneyim döngüsünün amacı ise ortaya çıkan fikirlerin işe yarayıp yaramadığını test ederek fikirleri temizlemektir.

Deneyimsel Oyun Modeli, eğitsel oyunlar tasarlamak ve analiz etmek için kullanılabilir. Ancak, model eğitim kuramları ve oyun tasarımı arasında bir bağlantı olarak çalışır ve bir eğitsel oyun tasarım projesine baştan sona yol gösteremez.

Kiili, eğitsel oyun tasarımcılarının ihtiyaçlarını daha iyi karşılayabilmek için Deneyimsel Oyun Modelinin gözden geçirilmiş bir sürümünü sunmuştur. Bu sürümde modelin tasarım döngüsü genişletilmiştir. Böylece, sadece oyunlarla



öğrenme sürecini tanımlamak yerine, oyun tasarımı ve geliştirme süreci de modele dâhil edilmiştir. Ayrıca tasarım döngüsüne ek olarak akış elemanları da ilave edilmiştir. Deneyimsel Oyun Modelinin ikinci sürümü **Şekil 2.14**'te sunulmuştur.



**Şekil 2.14:** Deneyimsel Oyun Modelinin ikinci sürümü (Kiili, 2005b).

Deneyimsel Oyun Modelinin ikinci sürümünü oluşturan öğeler; mücadele, açık hedefler, dönüt, odaklanmış dikkat, kontrol duygusu, oynanabilirlik, hikâye çerçevesi ve oyunçuluktur. Kiili (2005b), bu öğeleri **Tablo 2.5**'teki gibi açıklamıştır.

**Tablo 2.5:** EGM öğelerinin açıklaması.

Mücadele	Zorluklar oyuncunun beceri düzeyine uygun olmalıdır. Zorluk seviyesi oyuncunun ilerleme seviyesine uygun olarak değişmelidir. Oyun, becerilerin gelişmesini desteklemeli ve gelişen becerileri ödüllendirmelidir. Oyun, yeni problemleri uygun hızda sunmalıdır. Problemler öğrenme hedeflerine uygun olmalıdır.
Açık hedefler	Oyun, en başında ana hedefi açıkça ortaya koymalıdır. Oyun, alt hedefleri uygun hızda ve açıkça vermelidir.

**Tablo 2.5'in devamı.**

Dönüt	Oyuncunun eylemlerine kesin ve anında dönüt verilmelidir. Oyun, hedeflere yönelik olarak dönüt sağlamalıdır. Oyunun durumu hakkında bilgi vermelidir.
Odaklanmış dikkat	Oyun, oyuncunun dikkatini çekmeli ve oyun boyunca bu dikkati korumalıdır. Oyuncunun dikkati öğrenme amaçlarına yönelik olmalıdır. Oyuncu, dikkatini dağıtan gereksiz şeylerle meşgul edilmemelidir. Oyuncuya bilişsel yapısına uygun olarak yüklenilmelidir.
Kontrol duygusu	Oyunun arayüzü, kontrolün oyuncuda olduğu hissini vermelidir. Oyuncunun özgürlüğü çok kısıtlanmamalıdır. Oyuncunun, oyun içindeki düzeltilmesi zor hataları önlenmelidir. Oyuncu, hata düzeyi çok az olan başarı düzeyine ulaşabilmelidir. Oyunda ilerlemenin, oyuncunun elinde olduğu hissi verilmelidir.
Oynanabilirlik	Kullanıcı arayüzü ve kontroller kolay öğrenilmeli ve kullanılmalıdır. Oyun akılda kalıcı olmalıdır. Hedeflere ulaşmak için uygun araçlar sağlamalıdır. Gereksiz ses ve grafiklerle oyuncu aşırı yüklenmemelidir. Animasyonlar en üst düzeyde fayda sağlamalıdır. Oyuncu, üst düzey yapılarla şaşırtılmamalıdır. Grafikler ve sesler uyumlu olmalıdır. Oyuncuların topluluk oluşturması desteklenmelidir.
Hikâye çerçevesi	Hikâye, hedefleri algılamak ve problemleri yapıya entegre etmek için kullanılmalıdır. Oyuncunun karşılaştığı problemler hikâyeye uyumlu olmalıdır.
Oyunsallık	Kazanımlar, oyundaki etkinlerle bağlantılı ve yapıcı olmalıdır. Oyun doğrusal olmayan geniş bir mücadele sunmalıdır.

Akış öğeleri, oyun tabanlı öğrenmeyi destekleyen bir yapı sunarlar. Akış deneyimi, kişilerin fiziksel veya zihinsel bir zorluğu aşmak için gönüllü olarak sınırlarını zorladıklarında etkili olmaktadır. Bu durum akış deneyimini desteklemek için oyuncunun kolay ve zahmetsizce sonuca ulaştığı bir eğitsel oyun geliştirilmesi gerektiği şeklinde yorumlanmamalıdır. Aksine, eğitsel oyunlar oyuncunun zorlukları aşabilmek için sınırlarını zorlamalıdır. Akış deneyiminin bu doğası, eğitsel oyun tasarlamak için uygun bir tasarım yöntemi sunar. Böylece yaşam boyu öğrenme stratejisine uygun olarak oyuncular başka bir ödül beklemezsizin eğitsel oyunun kendisini bir ödül olarak görür ve oyun oynayarak öğrenebilirler.

Genellikle, eğitsel oyun geliştirmek eğitim kurumlarının göze alamayacağı kadar büyük kaynaklar gerektirdiği için Deneyimsel Oyun Modeli zaman ve maliyet verimliliği ilkesi gözetilerek tasarlanmıştır. Modele ilave edilen tasarım döngüsünün amacı, tasarımcıları yönlendirmek ve çalışmalarını kolaylaştırmaktır. Tasarım döngüsünün çok temel öğeler barındırması farklı oyun türlerini içerecek şekilde özelleştirilebilmesini sağlamak içindir. Ayrıca, Crawford (1984)'a göre, oyun tasarımı tek bir yöntemle indirgenemeyecek kadar karmaşık bir süreç olduğundan, bu yaklaşım makul bir çözüm olarak görülebilir.

Tasarım döngüsü, öğrencilerin ihtiyaçlarını ve tasarım çözümlerini tanımlamak için ihtiyaç analizi ile başlar. Modelin döngü yapısı ihtiyaç analizi sürecini de açıklamaktadır. Yaratıcı (birincil) düşünme aşamasında tasarımcı mümkün olduğunca yaratıcı çözümler geliştirmeye çalışır. Ortaya çıkan fikirler; akış deneyimi öğeleri, öğretim tasarımı ilkeleri ve konu alanının kısıtlamaları dikkate alınarak değerlendirilir ve çözüm olarak geliştirilir. Bu aşamadan sonra, geliştirilen çözümler, oyuncuyla erken deneyim yaşamak için hızla uygulanır. Hızlı prototip geliştirme, tasarımcıya çabuk bir değerlendirme yapma imkânı sunar (Moonen, 1996). Prototip geliştirilmesi, son kullanıcıyı tasarım sürecine dahil etme avantajı sağlar. Bir seferde mükemmel sonuca ulaşmayı değil, yapılan değişikliklerle ürünün aşama aşama son halini almasını amaçlar.

Yansıtıcı değerlendirme aşamasında, gerçekleştirilen deneyler değerlendirilir. Odak ise oyun dünyasının ve deneylerin analizidir. Oyun dünyasının analiz edilmesindeki amaç, oyundaki özelliklerin kullanımının incelenmesidir. Deneylerin analizindeki amaç ise oyuncuların deneyimlerinin, oyun hakkındaki duygularının ve algılarının analiz edilerek tasarım için yeni fikirler ortaya çıkartılmasını sağlamaktır.

**Tablo 2.6'**da tasarım döngüsünde her bir aşama (ihtiyaç analizi, uygulama, yansıtıcı değerlendirme, tasarım bilgisi); ana görevler, olası yöntemler, çıktılar ve riskler için özetlenmiştir.

**Tablo 2.6:** Tasarım döngüsünün aşamalarının özeti.

	İhtiyaç Analizi	Uygulama	Yansıtıcı Değerlendirme	Tasarım Bilgisi
Görevler	İhtiyaçları belirlemek ve yaratıcı çözüm tasarımları oluşturma.	Tasarım çözümlerini uygulama.	Uygulanabilir çözümleri değerlendirme. Daha derinlemesine anlama.	Test edilmiş çözümlerin tasarım bilgisini yapılandırma.
Yöntemler	Akış öğelerinin, öğretim tasarımı ilkelerinin ve bağlamsal faktörlerin analiz edilmesi.	Hızlı prototipleme.	Oyun dünyasının ve deneyimlerin analiz edilmesi.	Başarılı ve başarısız çözümlerin raporlanması.
Çıktılar	Eğitsel, teknik ve psikolojik tasarım ilkeleri ve çözümler.	Akışı tetikleyen eğitsel oyun.	Oyun dünyasının özellikleri ve kullanıcı deneyimleri.	Gelecekteki ihtiyaçlar için tasarımcı örnekleri.
Riskler	Oyun tasarımı ve eğitimin anlamlı bir şekilde birleştirilmesi.	Oyunculara çok erken açıklama yapılması.	Uygun olmayan yöntemler: Hatalı sonuçlar.	İçerikle ilgili doküman yokluğu. Aktarım problemleri.

Oyun tasarım modellerinin tamamı göz önüne alındığında, ihtiyaç analizi yapılarak senaryonun oluşturulması ve eğitsel amaçlar doğrultusunda oluşturulan mücadele öğelerinin oyuna yerleştirilmesi öngörülmektedir. Oyuncuların mücadeleler boyunca yaptıkları eylemlere açıklayıcı dönütler vererek oyun oynarken öğrenmeleri hedeflenmektedir.

## 2.6 Basit Elektrik Devrelerinin Öğretimi

Bu bölümde “Basit Elektrik Devreleri” konusunda yapılmış çalışmalar ve sonuçlarından bahsedilmiştir.

Basit elektrik devreleri konusunda farklı ülkelerde, farklı yaş gruplarına yönelik yapılmış pek çok çalışma bulunmaktadır. Bu çalışmalar öğrencilerin fizik derslerinde özellikle de elektrik konusunda çok sayıda kavram yanılığına sahip olduklarını göstermektedir (Çepni ve Keleş, 2006; Yıldırım ve Ark., 2008; Chambers ve Andre, 1997; Cohen, Eylon ve Ganiel, 1983; Dupin ve Johsua, 1987; Heller ve Finley, 1992; Osborne, 1981; Küçüközer, 2004).

Öğrencilerin sahip olduğu kavram yanılıkları hakkında yapılan çalışmalarda farklı araştırmacılar tarafından çeşitli kavram yanılıkları belirlenmiştir.

Duit ve Rhöneck (1997), öğrencilerin pilin elektrik akımını depolayan bir araç olduğuna ve lambanın da bu akımı tükettiği düşüncesine inandıklarını göstermişlerdir. Aynı şekilde, Frederiksen, White ve Gutwill (1999) yaptıkları çalışmayla, öğrencilerin akımın tüketildiği fikrine inandıklarını ortaya koymuşlardır.

Cheng ve Kwen (1998) ise öğrencilerin elektrik ve elektrik akımı kavramlarını karıştırdıklarını ve sık sık birbirlerinin yerine kullandıklarını ifade etmişlerdir. Lee ve Law’ın (2001) yaptığı çalışmada da pilin akım kaynağı olduğuna ve pilden yayılan akımın dış devredeki değişiklikten etkilenmediği fikrinin öğrencilerde gözlemlendiği belirtilmiştir.

Literatürde sıklıkla bahsedilen kavram yanılıkları ve bu kavram yanılıklarına sahip olan öğrencilerin inanışları kısaca aşağıdaki şekilde özetlenebilir:

- Bölgesel düşünce modeli: Bu kavram yanılıgısına sahip olan öğrenciler, devrede herhangi bir deęişiklik yapıldığında, sadece deęişiklik yapılan bölümden sonrasının etkilendiğine inanırlar (Engelhardt ve Beichner, 2004; Heller ve Finley, 1992; Shipstone ve Ark., 1988).
- Tek kutuplu model: Bu kavram yanılıgısına sahip olan öğrenciler, pil ve lamba arasındaki tek bir telin lambayı yakmak için yeterli olduğuna ve akımın pilin pozitif kutbundan lambanın tabanına aktığına inanırlar (Chambers ve Andre, 1997; Osborne, 1981).
- Çarpışan akımlar modeli: Bu kavram yanılıgısına sahip olan öğrenciler, pozitif kutuptan gelen pozitif akım ile negatif kutuptan gelen negatif akımın lambada karşılaşarak lambanın yanması için gerekli enerjiyi oluşturduğuna inanırlar (Osborne, 1983; Chambers ve Andre, 1997).
- Zayıflayan akım modeli: Bu kavram yanılıgısına sahip olan öğrenciler, akımın devrede belli bir yönde aktığına inanmakla birlikte yol üstündeki devre elemanlarının akımı kullanarak azalttığına inanırlar (Chambers ve Andre, 1997; Heller ve Finley, 1992; Osborne, 1983).
- Paylaşılan akım modeli: Bu kavram yanılıgısına sahip olan öğrenciler, seri baęlı devrelerde, akımın devre elemanları tarafından eşit olarak paylaşıldığına ve aynı zamanda devre elemanları tarafından eşit olarak harcandığına inanırlar (Shipstone, 1985; Chambers ve Andre, 1997; Heller ve Finley, 1992).
- Deneysel kural: Bu kavram yanılıgısına sahip olan öğrenciler, pile yakın olan lambanın uzak olanlara göre daha parlak yanacağına inanırlar (Heller ve Finley, 1992).
- Sabit akım kaynağı modeli: Bu yanılıgıya sahip olan öğrenciler, üretcin, baęlandığı devreden baęımsız olarak, devreye sürekli aynı akımı veren bir devre elemanı olduğuna inanırlar (Kärqvist, 1985; Cohen ve Ark., 1983).
- Kısa devre önyargısı: Bu yanılıgıya sahip olan öğrenciler, devreye baęlanan boş bir telin devre üzerinde herhangi bir etkisi olmadığına inanırlar (Shipstone ve Ark., 1988).
- Eşdeęer direnç önyargısı: Bu yanılıgıya sahip olan öğrenciler, devreye baęlanan direnç sayısı arttıkça devrenin eşdeęer direncinin artacağını düşünürler (Chambers ve Andre, 1997; Dupin ve Johsua, 1987).

Osborne (1983) ve Shipstone (1985), öğrencilerde bulunan kavram yanlışlarının yaşa göre değişiklik gösterdiğini belirlemişlerdir. Öğrencilerin yaşı ilerledikçe, bilimsel olarak doğru kabul edilen modellere doğru geliştiğini ancak önceki yıllarda kavram yanlışlarının sıklıkla gözlemlendiğini ifade etmişlerdir.

Konuya ilişkin kavramların oldukça soyut ve karmaşık olmaları, araştırmacıları farklı modeller ve benzetmeler geliştirmeye itmiştir (Mulhall, McKittrick ve Gustone, 2001). Bu gibi çalışmalarda, elektrik devreleri konusunda hangi tür kavram yanlışları olabileceği genel olarak belirlenmesine rağmen, bu kavram yanlışlarının giderilmesi için hangi öğretim metotlarından nasıl yararlanılacağı hakkında benimsenmiş bir görüş yoktur (Shipstone ve Ark., 1988).

### 3. YÖNTEM

Çalışmanın bu bölümünde örneklem seçimi, veri toplama araçları, Deneyimsel Oyun Modeline göre elektroGame'in geliştirilmesi, uygulama süreci ve verilerin çözümlenmesi hakkında bilgi verilmiştir.

Çalışma nicel ve nitel araştırma yönteminin birlikte kullanıldığı karma-metot ile yürütülmüştür. Nicel araştırma yöntemi olarak yarı deneysel desenlerden kontrol gruplu ön-test son-test deney modeli (quasi-experimental pretest-posttest control group research design) kullanılmıştır (Fraenkel ve Wallen, 2003). Nitel veriler yarı yapılandırılmış yüz yüze görüşme yöntemi kullanılarak elde edilmiştir. Seçilen metodun, elde edilen verilerin daha güvenli ve kapsamlı olmasını sağlaması, böylece sonuçların güç ve kapsamını da arttırması beklenmiştir.

Araştırmada; fizik dersine yönelik tutum ölçeği, bilgisayara yönelik tutum ölçeği ve basit elektrik devreleri üç basamaklı testi kullanılmıştır. Tutumlardaki farklılıkların oyundaki başarıya ve oyundaki başarının fizik dersi başarısına etkisi olup olmadığı incelenmiştir.

Yüz yüze görüşmeler, araştırmacı tarafından ses kayıt cihazı kullanılarak gerçekleştirilmiştir.

#### 3.1 Araştırmanın Örnekleme

Araştırmanın örneklemini ise Balıkesir ilinden bir Anadolu Lisesi (O1) ve Meslek Lisesi (O2)'nden birer sınıf oluşturmaktadır.

O1 okulundan seçilen ilk sınıf (O1S1) 11 kız ve 11 erkek; diğer sınıf (O1S2) 11 kız ve 11 erkek öğrenciden oluşmaktadır; bu okulda araştırmaya dâhil edilen toplam öğrenci sayısı 44'tür. O2 okulundan seçilen ilk sınıfta (O2S1) 19 kız ve 5 erkek; diğer sınıfta ise (O2S2) 22 kız ve 2 erkek olmak üzere toplam 48 öğrenci mevcuttur.

**Tablo 3.1:** Örneklemin özellikleri.

	Deney Grubu	Kontrol Grubu	Toplam
O1	22	22	44
O2	24	24	48

Araştırma Milli Eğitim Bakanlığına bağlı okullarda yürütüldüğü ve sınıflar okul idareleri tarafından dönem başlarında oluşturulduğu için, deney ve kontrol grubu öğrencilerinin sınıflar arasında rastgele dağıtılması mümkün olmamıştır. Ancak deney ve kontrol grubunu oluşturacak olan sınıflar, tüm 9. sınıflar arasından rastgele seçilmiştir.

O2, kız öğrencilerin daha çok tercih ettiği bir meslek lisesi olduğu için sınıflarda kız ve erkek öğrenci sayılarının dağılımı kızlar lehine ağırlık göstermektedir. O1’de ise öğrenciler cinsiyete göre eşit dağıtılmıştır.

Her iki okulda da, ön-test başarı puanları, fizik tutumları ve bilgisayar tutumları açısından sınıflar arasında fark gözlenmemiştir. Deney ve kontrol gruplarında bilgisayar tutum, fizik tutum ve başarı (test, kavram, emin olma) ölçeklerine göre anlamlı bir farklılık bulunmadığı Bölüm 5.1’de ayrıntılı olarak incelenmiştir.

### 3.2 Verilerin Toplanması

Bu çalışmada veri toplamak amacıyla farklı araştırmacılar tarafından geliştirilmiş dört ölçek ve görüşme yöntemi kullanılmıştır. Ölçekler, uygulama başlamadan 1 hafta önce ön-test olarak, uygulamanın bitiminden 1 hafta sonra da son-test olarak uygulanmıştır.

Bu bölümde sırasıyla ölçme araçları hakkında bilgi verilmiştir.

#### 3.2.1 Fizik Dersine Yönelik Tutum Ölçeği

Bu araç iki farklı okuldaki öğrencilerin fizik dersine karşı tutumlarında farklılık olup olmadığını görmek ve öğrencilerin tutumlarıyla elektroGame’i oynamaları arasında bir ilişki olup olmadığını belirlemek amacıyla kullanılmıştır.



Fizik tutum ölçeği olarak, Baykul (1990) tarafından matematik dersi için geliştirilen ve Tekmen (2006) tarafından fizik dersine uyarlanan ölçek kullanılmıştır.

Tutum ölçeği güvenilirliği  $\alpha=0,93$  olarak bulunmuştur. Fizik Dersi Tutum Anketi 13 olumlu, 12 olumsuz cümle olmak üzere toplam 25 maddelik beşli Likert tipindedir (Ek A).

### **3.2.2 Bilgisayara Yönelik Tutum Ölçeği**

Ölçek, öğrencilerin bilgisayar tutumlarının elektroGame oyununu oynamalarına etkisini incelemek amacıyla kullanılmıştır. Bilgisayar tutum ölçeği olarak, Loyd ve Gressard (1984) tarafından geliştirilen ve Berberoğlu ve Çalikoğlu (1991) tarafından Türkçeye uyarlanıp, güvenilirliği  $\alpha=0.90$  bulunan ve Şerefhanoglu (2007) tarafından yeniden düzenlenen “Computer Attitude Scale (CAS) Bilgisayara Yönelik Tutum Ölçeği” kullanılmıştır. Yeniden düzenlenen ölçeğin güvenilirliği  $\alpha=0.87$  olarak bulunmuştur (Şerefhanoglu, 2007) (Ek B).

Bilgisayar tutum ölçeği Likert tipi 21 maddeden oluşmaktadır. Maddeler sırasıyla “güven” ( $\alpha=0.753$ ), “isteklilik” ( $\alpha=0.716$ ), “isteksizlik” ( $\alpha=0.723$ ) ve “inanç” ( $\alpha=0.720$ ) faktörleri altında toplanmıştır.

### **3.2.3 Üç Basamaklı Başarı Testi**

Öğrencilerin basit elektrik devreleri konusundaki başarılarını ölçmek amacıyla Peşman’ın (2005) geliştirdiği üç basamaklı test kullanılmıştır. Üç basamaklı test, 12 soru ile öğrencilerin başarı puanlarını, cevaplarını verirken geliştirdikleri düşünce biçiminin bir kavram yanlışlığı olup olmadığını ve verdikleri cevabın doğruluğuna güvenip güvenmediklerini ölçmektedir. Peşman’ın çalışmasında testin güvenilirlik katsayıları, başarı puanlarında  $\alpha=0.69$  olarak, kavram yanlışlığı puanlarında ise  $\alpha=0.33$  olarak bulunmuştur.

9. sınıf “elektrik ve manyetizma” konusunda bulunan fakat üç aşamalı testin içermediği öz direnç konusunun ölçülmesi amacıyla, araştırmacı tarafından bir iki kısımdan oluşan açık uçlu bir soru daha ilave edilmiş ve soru sayısı 13’e

yükselmiştir. Sorunun uygunluğu konu alanı uzmanlarına danışılarak kontrol edilmiştir (Ek C). Eklenen açık uçlu sorunun puanlaması için, öğrencilerin cevapları doğru ve yanlış olarak tasnif edildikten sonra verilen cevapların açıklamalarında bilimsel olarak doğru kabul edilecek bir açıklama yapıp yapılmadığı incelenmiştir. Bilimsel olarak doğru kabul edilen yanıtlar kavramsal anlama düzeyi sonuçlarına bir artı puan olarak ilave edilmiştir.

### 3.2.4 Yarı Yapılandırılmış Görüşme

Öğrencilerin elektroGame oyunu hakkındaki düşüncelerini daha iyi öğrenebilmek için 8 öğrenciyle yarı yapılandırılmış görüşme yapılmıştır. Öğrenciler, deney grubundaki gönüllü öğrenciler arasından rastgele seçilmiştir. Yaklaşık 15–20 dakika süren görüşmeler, araştırmacı tarafından ses kayıt cihazı ile kaydedilmiştir. Görüşme formu on beş sorudan oluşmaktadır (EK D).

Görüşme soruları analiz edilirken, öğrencilere 1k, 2e gibi kısaltmalar verilmiştir. Öğrencilerin takma isimleri, okulları ve yaşları **Tablo 3.2**'de verilmiştir.

**Tablo 3.2:** Görüşme yapılan öğrenci bilgileri.

Takma isim	Yaş	Okul	Cinsiyet
1k	16	O1	Kız
2k	16	O1	Kız
3k	15	O2	Kız
4k	15	O2	Kız
5e	16	O1	Erkek
6e	16	O1	Erkek
7e	16	O2	Erkek
8e	15	O2	Erkek

### 3.3 Deneyimsel Oyun Modeli ve elektroGame

elektroGame tasarlanırken Kiili'nin (2005b) Deneyimsel Oyun Modeli temel alınmıştır. Bu bölümde, modelin aşamalarına göre elektroGame'in tasarlanma sürecinden bahsedilmiştir. elektroGame oyununun eğitsel ve öğretimsel özellikleri ile geliştirme süreci Bölüm 4'te ayrı bir bölüm olarak ele alınmıştır.

Konu belirlenirken, ortaöğretimde çalışan fizik öğretmenleri ve fizik eğitimi uzmanlarıyla değerlendirmeler yapılmıştır. Literatürde, basit elektrik devreleri konusunda fazla kavram yanılgısı bulunduğu ve farklı yöntemlerin bu yanılgıları azaltmada yeterince başarılı olmadığı bilgileri göz önüne alınarak konunun 9. sınıf fizik dersi “basit elektrik devreleri” olmasına karar verilmiştir. Ardından, 9. sınıf fizik dersi öğretim programı incelenerek basit elektrik devreleri konusunun hedefleri belirlenmiştir. Bu bağlamda, öğrencilerin oyunda kalmasını ve mücadele etmesini sağlamak için oyunun hikâyesi oluşturulmuştur. Hikâye yazılırken, oyuncuları motive etmek için char’larını elektroGame dünyasını kurtarmak üzere güçlendirmelerini sağlayacak görevler tanımlanmıştır.

Kiili’nin Deneyimsel Oyun Modelinin kalbinde mücadele, oyunsallık ve oynanabilirlik öğeleri yer almaktadır. elektroGame’de bir oyuncu char’ını güçlendirebilmek için tanımlanmış görevleri yerine getirmek zorundadır. Genel görevlerin yanı sıra öğrenme hedeflerinin yerine getirilmesi sonucunda char’ın seviyesi, dayanıklılığı, çevikliği, sağlığı, atak gücü gibi özellikleri artmaktadır. Karşılıklı bir ilişki olarak bu özellikler arttıkça char, daha zor görevlerin üstesinden gelebilecek yeteneklere sahip olmaktadır. Dolayısıyla bu kazanımlar oyundaki etkinliklerle bağlantılı ve yapıcı olduğu için modeldeki “oyunsallık” (gamefulness) ögesi yerine getirilmiş olmaktadır.

elektroGame 2B tasarlandığı için doğal olarak sade bir yapıya sahiptir. Oyun esnasında aşırı grafik yükü oluşmadığından, istemci bilgisayarlar zorlanmamaktadır. Aynı durum, ses efektleri için de geçerlidir. Oyunda, oyuncuların dikkatlerinin dağılmaması ve oyun dünyasında kaybolmamaları için sade grafiklerin ve uyumlu seslerin kullanılmasına özellikle dikkat edilmiştir. elektroGame’de MMORPG’lerde klan sistemi anlamında bir grup oluşturabilme imkânı olmasa da oyuncular birbirleriyle nesne değiş tokuş edip, soru çözümlerinde fikir alışverişi yapmak gibi etkileşimlerde bulunabilmektedir. Bu özellikleriyle elektroGame, modelin kalbindeki “oynanabilirlik” (playability) ögesini sağlamaya çalışmaktadır.

Oyuncu Kiili’nin (2005b) Deneyimsel Oyun Modelinde; temada verilen görevleri yerine getirmek için yaratıcı (birincil) fikirler ve yaratıcı çözümler üretirler. Modelin kalbinde yer alan mücadele, oynanabilirlik ve oyunsallık öğeleriyle bu

fikirleri geliştirip olgunlaştırır. Böylece modelin küçük kan dolaşımı döngüsü tamamlanmış olur.

elektroGame’de oyuncu daima bir mücadele ve rekabet içindedir. Oyuncular görevleri yerine getirmek için robotlarla mücadele edip, oyunun üst seviyelerine çıkmak için diğer oyuncularla rekabet etmektedirler.

Görevler, oyuncunun seviyesine uygun olarak değişmektedir. Örneğin, oyuna ilk katılan oyuncunun daha basit görevleri olurken daha yüksek seviyedeki oyuncular, çözmesi daha karmaşık problemlerle karşı karşıya gelmektedir. Dolayısıyla zorluk seviyesi oyuncunun ilerleme seviyesiyle doğru orantılıdır. Daha zor görevlerin yerine getirilmesi sonucunda, daha özel ve daha kıymetli ödüller (devre elemanları, kıyafetler, silahlar vb.) kazanılmaktadır. elektroGame bu özellikleri kullanarak modelin kalbindeki “mücadele” ögesini gerçekleştirmeyi amaçlamaktadır.

elektroGame’de açık tanımlanıp net ifade edilmiş görevler, oyuncunun hedeften sapmadan denemeler yapmasını, dolayısıyla oyunda aktif olmasını sağlamaktadır. Bu deneyim esnasında oyuncuya sağlanan belirgin geri dönütler ile yansıtıcı gözlem yapması ve yolunda gitmeyen bir görev için farklı çözümler üretmesi beklenmektedir. Aynı zamanda, oyuncu oyunun her basamağında durum penceresi yardımıyla char’ını özellikleri hakkında bilgi alabilmektedir. elektroGame’de, oyuncu dikkatinin gereksiz şeylerle meşgul edilmesi önlenerek dikkatin öğrenme amaçlarına yönlendirilmesi sağlanmaya çalışılmaktadır. Bu sayede oyuncu zihnindeki kavramsal şemayı oluşturmakta ya da güçlendirmektedir. elektroGame’de, oyuncunun kontrolün kendisinde olduğunu hissetmesi için görevlerin seçimi ve sırası tamamen oyuncuya bırakılmıştır. Oyunda ilerlemek oyuncunun elindedir ve oyun boyunca oyuncunun bunu hissetmesi sağlanmaya çalışılmaktadır. Böylelikle modelin deneyim döngüsü tamamlanmış olmaktadır.

elektroGame’in tasarım döngüsünde, ihtiyaç analizi, uygulama, yansıtıcı değerlendirme ve tasarım bilgisi öğeleri; görevler, yöntemler, çıktılar ve riskler göz önüne alınarak aşağıdaki şekilde uygulanmıştır.

Fizik öğretmenleri ve konu alanı uzmanlarının görüş ve önerileri doğrultusunda, “basit elektrik devreleri” konusu belirlendikten sonra kullanılacak

grafik kütüphaneleri, kullanıcı ara yüzleri, oyun motorları, sunucu yazılımları incelenerek eğitsel, teknik ve psikolojik ilkeler göz önüne alınmış, uygun olanlar seçilmiştir.

Karar verilen yöntemler ile geliştirilen oyunun her bir bileşeninin denenmesi sağlanıp aksayan, çalışmayan, anlaşılmayan öğelerin yeniden gözden geçirilmesi sağlanmıştır. Bu durum, tasarlanan her bir öğe için ve tüm tasarım süreci boyunca tekrarlanmıştır.

Uygulama esnasında yapılan denemelerin analizi gerçekleştirilerek ve kullanıcı deneyimleri göz önüne alınarak yansıtıcı değerlendirme yapılmış, bu esnada hatalı sonuçlar veren ya da uygun olmayan yöntemler terk edilmiştir.

Bütün geliştirme sürecinde, test edilmiş ve başarıyla sonuçlandırılmış öğelerin hem oyuncularla hem de geliştiricilerle paylaşılması amacıyla, wiki, blog ve forum öğelerini içeren bir web sitesi hazırlanmıştır. Oyun geliştirme sürecinde tüm deneyimler belgelenerek web sitesindeki ilgili bölümlerde paylaşılmıştır.

Uygun bir grafik kütüphanesi araştırılarak tasarım sürecine başlanmıştır. Kullanımı kolay olduğu için kolayca hakkında kaynak bulunan Allegro kütüphanesinin uygun olduğu düşünülmüş ancak teknik yetersizlikleri görülünce daha gelişmiş bir kütüphane olan SDL'ye geçilmiştir. SDL kütüphanesi kendi GUI sistemine sahip olmadığı için uygun bir GUI araştırması yapılmış aynı zamanda GUI'lere uygun açık kaynak kodlu oyunlar ve oyun motorları da incelenmiştir. Bu aşama sonunda The Mana World oyunu hem SDL'yi kullandığı hem de GUIChan ara yüzünde yazıldığı için tercih edilmiştir.

Daha sonra oyunun hikâyesi oluşturulmuş ve hikâye çerçevesine uygun olacak şekilde öğretim ve eğlence öğeleri tasarlanmıştır.

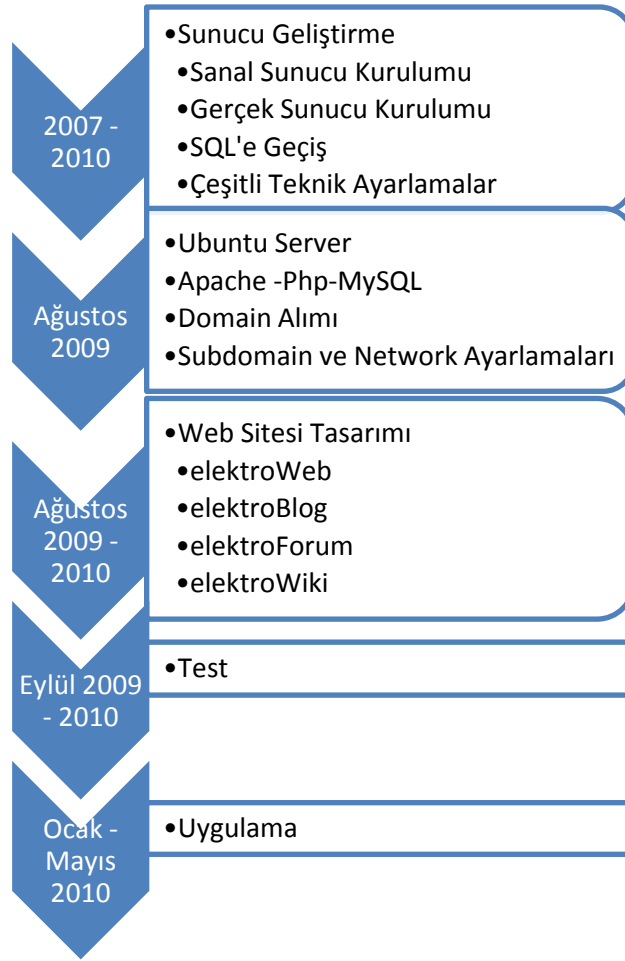
NPC, harita, karakter ve item çalışmaları, test ve sunum sistemi ile basit elektrik devrelerini çözen yapay zekânın geliştirilmesi ile elektroGame şekillenmeye başlamıştır. Eş zamanlı olarak istemciyle uyumlu sunucu yapısı geliştirilmiş ve elektroGame'e uyarlanmıştır. Daha sonra sistem ihtiyaçları göz önüne alınarak fiziksel sunucu yapılandırılmış ve internet erişimine açık hale getirilmiştir. Ayrıca,

oyuncuların ve geliştiricilerin birbirleriyle iletişim kurabilmeleri için web, blog, forum ve wiki sayfaları oluşturulmuştur.

Üç ay boyunca test edilerek uygulama aşamasına geçilmiştir.

**Şekil 3.1**'da elektroGame'in tasarım döngüsündeki adımlar ve tarihleri verilmiştir.





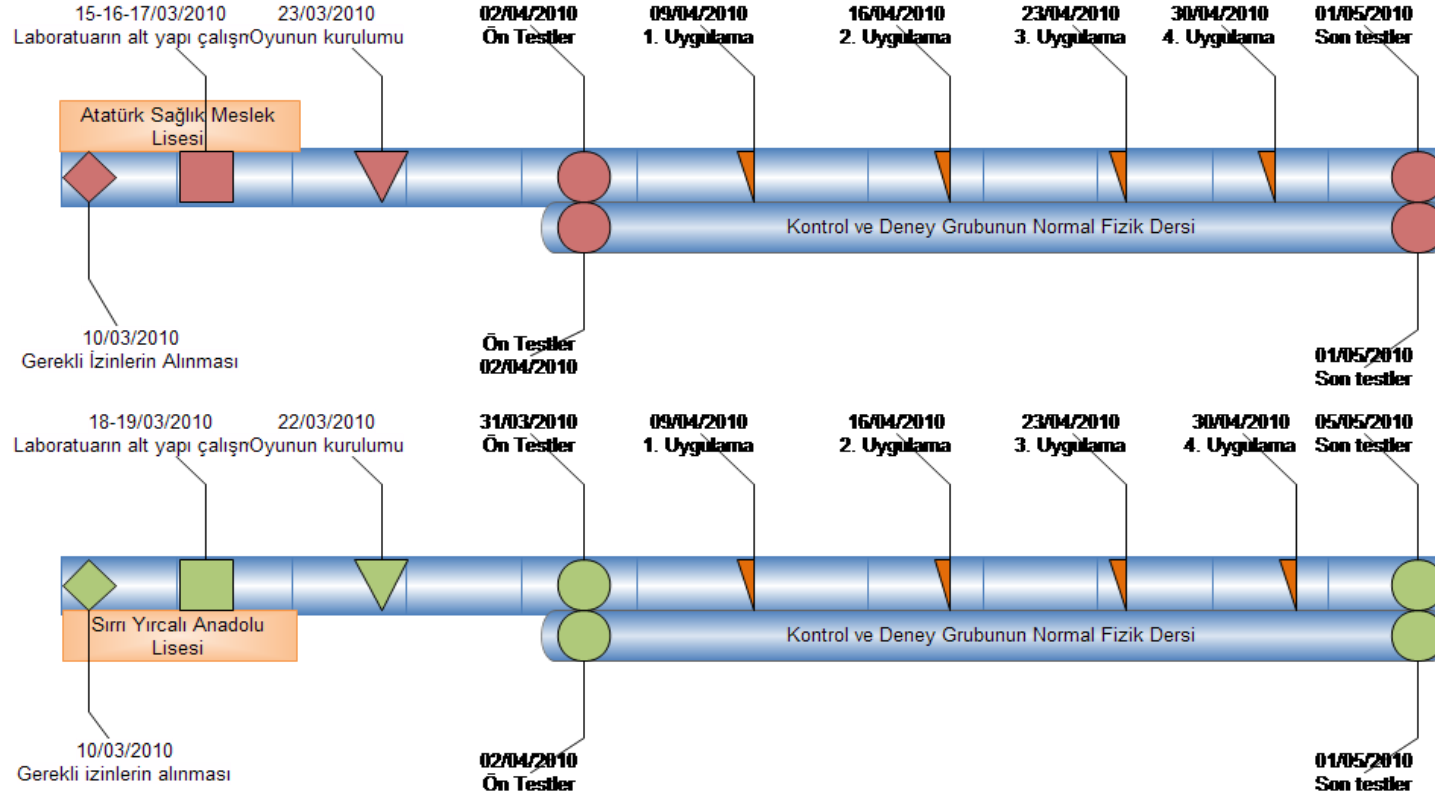
**Şekil 3.1:** elektroGame tasarım geliştirme süreci.

### 3.4 Uygulama Süreci

elektroGame, oynanabilir duruma getirildikten sonra, gerekli izinlerin alınmasını, uygulama yapılacak laboratuvarların hazırlanmasını ve bilgisayarlara elektroGame'in kurulmasını içeren bir hazırlık süreci başlamıştır. Hazırlıkların tamamlanmasının ardından, ön-testler yapılmış, bir hafta oryantasyon çalışması, dört hafta süren uygulama çalışması, ardından son-testler ve görüşmeler yapılmıştır.

Şekil 3.2'da çizelgesi verilen hazırlık süreci ve uygulama süreci açıklanmıştır.

# Uygulama Süreci



Şekil 3.2: Uygulama süreci.



### **3.4.1 Hazırlık Süreci**

Uygulama yapılmak istenen okulların yöneticileriyle ve uygulama sırasında yardımları gerekebilecek öğretmenlerle ön görüşmeler yapıldıktan sonra Balıkesir Milli Eğitim Müdürlüğünden gerekli izinler alınmıştır.

O2'nin bilgisayar laboratuvarı son bir sene içinde yenilediği için bilgisayarların teknolojisinin elektroGame'i rahatça oynatabilecek kapasitede olduğu tespit edilmiştir. Ancak O1'in bilgisayarları çok eski olduğu için özellikle grafik kalitesinde sıkıntı yaşanmasına yol açmıştır. Bu durumun oyunun oynanmasını etkilememesi için O1'e yüklenen oyunların grafik kalitesini düşüren ancak performansı yükselten ayarların seçili olduğu bir özel kurulum dosyası hazırlanmıştır.

Her iki okulda da internet bağlantısı MEB'in okullar için sağladığı sınırlı bağlantı üzerinden gerçekleştirildiğinden, elektroGame istemcilerinin sunucuyla haberleşmesi mümkün olamamıştır. Bu durumu aşabilmek için yöneticilerin kullandığı 8Mbit'lik ADSL hattından laboratuvarlara ek bir hat çekilmiştir. Uygulama sonrasında hattın devre dışı bırakılarak MEB'in güvenli internet bağlantısı tekrar aktif hale getirilmiştir.

İnternet bağlantısı hazır hale getirildikten sonra, kurulum programı ağ üzerinden dağıtılarak elektroGame bilgisayarlara kurulmuştur. Öğrencilerin elektroGame'e üye olabilmeleri için elektroGame web sitesi tarayıcılarda ana sayfa olarak ayarlanmıştır.

Web sitesine erişim ve oyunun sorunsuz çalışıp çalışmadığı test edilerek hazırlık aşaması sonlandırılmıştır.

### **3.4.2 Uygulama Süreci**

Uygulama sürecine ön-testlerin uygulanması ile başlanmıştır. Ardından deney ve kontrol grubu öğrencileri fizik derslerine normal şekilde devam ederken farklı olarak deney grubu bilgisayar derslerinde elektroGame oyununu oynamıştır.

Uygulama grubuna, ilk derste elektroGame'e nasıl üye olunacağı anlatılmış ve elektroGame'in web sitesi tanıtılmıştır. Ardından öğrenciler, web sitesi üzerinden üye kayıt formuna kendilerine ait bilgileri girmişlerdir. Kullanıcı adını ve şifresini tanımlarken dikkatli olmaları gerektiği, bu ad ve şifre ile oyuna giriş yapacakları ayrıca açıklanmıştır.



**Şekil 3.3:** O2 okulundan bir görüntü.

Üyelik işlemleri tamamlandıktan sonra öğrenciler eğitim haritasına yönlendirilmiş ve eğitim haritasında verilen görevleri tamamlamaları istenmiştir. Daha önceden RPG oyun deneyimi olan öğrenciler zorlanmadan görevleri tamamlamış ancak ilk kez bu tarz bir oyunla karşılaşanlar çeşitli aşamalarda yardıma ihtiyaç duymuşlardır. Böyle durumlarda, öğrencilere araştırmacı tarafından yardım edilmiştir. Ayrıca öğrencilerin de sık sık birbirlerine yardım ettikleri gözlenmiştir.



**Şekil 3.4:** O1 okulundan bir görüntü.

Deney gruplarındaki her öğrenci başına bir bilgisayar düşmüştür. Böylece öğrenciler oyunu tek başlarına oynama şansı yakalamışlardır.

Uygulama yapılan dört haftanın ardından deney ve kontrol gruplarına son testler uygulanmıştır.

### **3.5 Verilerin Çözümlemesi**

Elde edilen veriler, nitel ve nicel araştırma yöntemleri kullanılarak analiz edilmiştir. Nicel verilerin analizinde SPSS 19.0 programı kullanılmıştır. Yapılan uygulamalar esnasında kullanılan ölçeklerden elde edilen verilerin frekans, aritmetik ortalama, standart sapma ve dağılım yüzdeleri hesaplanmıştır.

İki değişken arasındaki ilişkinin miktarını bulup yorumlamak amacıyla korelasyon katsayısı hesaplanmıştır. Hesaplanan Pearson korelasyon katsayısı ile iki değişken arasındaki ilişkinin kuvveti (düşük, orta, yüksek) ve yönü (pozitif, negatif) hakkında yorum yapılabilmektedir (Büyüköztürk, 2003).

İki ilişkisiz örneklem ortalamaları arasındaki farkın anlamlı olup olmadığının test edilmesi amacıyla ilişkisiz örneklem için t-testi (independent sample t-test) kullanılmıştır (Büyüköztürk, 2003).

Zamana bağlı olarak tekrarlı ve ilişkisiz ölçümler için iki faktörlü ANOVA (two-way ANOVA for mixed measures) testi yapılmıştır. Toplam varyans, denekler arası ve denekler için olmak üzere iki bölümde incelenmiştir (Büyüköztürk, 2003).

Araştırma öncesinde iki farklı okulda, iki ayrı deney ve kontrol grubu seçilmiştir. Deney ve kontrol gruplarına uygulama öncesinde “basit elektrik devreleri üç basamaklı testi”, “bilgisayar tutum ölçeği” ve “fizik tutum ölçeği” uygulanmıştır. Araştırmanın bitiminde deney ve kontrol gruplarına aynı ölçekler tekrar uygulanmıştır. Böylece uygulama öncesi ve sonrası farklılıklar ölçülmeye çalışılmıştır.

Uygulama sonrasında her iki okuldan da gönüllüler arasından rastgele seçilen toplam sekiz öğrenci ile yüz yüze görüşme yapılmıştır. Görüşmeler yarı yapılandırılmış olup yaklaşık on beşer dakika sürmüştür ve kayıt altına alınmıştır. Sesli görüşmelerin çözümlenmesi yapılmış ve metin haline getirilen görüşmelerin içerik analizi gerçekleştirilmiştir. Görüşmelerden elde edilen veriler incelenerek kodlanmış, kodlar arasındaki ilişkiler incelenmiştir. Daha sonra temalar belirlenerek, aralarındaki ilişkiler açıklanmaya çalışılmıştır. Nitel verilerin analizinde sırasıyla aşağıdaki adımlar takip edilmiştir:

- Görüşme verilerinin incelenmesi
- Anlamlı verilerin işaretlenmesi
- Verilerin kodlanması
- Taslak temaların belirlenmesi
- Taslak temalara göre kodların düzenlenmesi
- Taslak temaların ve kodların gözden geçirilmesi ve son halinin verilmesi
- Kodlara ve temalara göre verilerin betimlenmesi
- Alıntılar eklenmesi, örneklendirilmesi
- Açıklanması, yorumlanması ve tablolandırılması
- Araştırma sonuçlarının yazılması

## 4. ELEKTROGAME

Bu bölümde MMORPG geliştirme, elektroGame oyununun eğlenceye ve öğretime yönelik özellikleri ele alınmıştır. İstemci, sunucu, web sitesi ve yardımcı diğer programlarla birlikte elektroGame'in geliştirme süreci detaylı olarak açıklanmıştır.

### 4.1 MMORPG Geliştirme

Bir MMORPG geliştirebilmek için, hata toleransı, çevrim içi sunucu yönetimi, hilelerin önlenmesi, dağıtık tutarlılığın sağlanması, ölçeklenebilirlik gibi pek çok zorluğun üstesinden gelinmesi gerekmektedir (Alexander, 2005).

En büyük zorluklardan birisi, çok geniş bir alana yayılan oyuncular arasında eş zamanlı olarak verileri dağıtmak ve oyuncuların aynı ortamda olduğu hissini verecek dağıtık veri tutarlılığını sağlamaktır (McFarlane, 2005). Bunun için her oyuncu kendi bilgisayarında oyunun kendisini ilgilendiren bölümünün anlık bir kopyasını buldurmalıdır. Ayrıca, herhangi bir oyuncu oyun dünyasını etkileyecek bir eylem gerçekleştirdiğinde, o eylemin etkilediği tüm oyuncuların anlık durum bilgisinin güncellenmesi gerekir (Boulanger, 2006). Nesnelerin ve oyuncu sayılarının artmasıyla birlikte, ağ üzerinden taşınması gereken bilginin miktarı hızla artar. Bu noktada değiştirilmesi gereken bilgi için işlemci ve ağ bant genişliği (network bandwidth) darboğazı oluşabilir (Smed, Kaukoranta ve Hakonen, 2001). Oyuncular arasında bilginin dağıtılması sırasında, insan algısının altında bir gecikme kabul edilebilirken daha fazlası oyun zevkini bozan takılmalar olarak kendini gösterir (Singhal ve Zyda, 1999). İyi bir MMO oyunun bu ağ gecikmesiyle (network latency) baş edecek yöntemler buldurması gerekmektedir.

Bant genişliği, birim zamanda ağ üzerinden aktarılabilen veri miktarıdır. Bant genişliği, bilgisayarları birbirine bağlayarak ağı oluşturan (ethernet kartı, hub vb.) temel donanım ile sınırlıdır. İnternette ise oyuncunun servis sağlayıcısı (IIS), servis sağlayıcının donanım ve yazılım alt yapısı, anlık trafik durumu, bağlı kullanıcı gibi bant genişliğini etkileyebilecek pek çok etken bulunmaktadır. Ayrıca her oyuncun

farklı bant genişliği bulunabileceği de göz önüne alınması gereken önemli bir etkidir. Sonuç olarak MMO bir oyun, değiştirilmesi gereken veri miktarını bant genişliğini göz önünde bulundurarak hesaplamalıdır. Mümkün olduğu oranda, oyuna katılan oyuncuların ihtiyaçları, bant genişliği ihtiyacını arttırmayacak yöntemlerle yapılmalıdır (Boulanger, 2006).

İşlem gücü ise bir bilgisayar tarafından birim zamanda gerçekleştirilebilen hesaplama miktarıdır. Her bilgisayarın işlem gücü donanımına bağlı olarak sabit bir değerdir. Bir oyunda işlemci gücü, fiziksel hesaplamalar, çakışma algılamaları (collision detection), grafik işleme ve ses gibi birden fazla işlemle baş etmek için kullanılır. MMO oyunlarda tüm bu ihtiyaçlara ek olarak ağ üzerinden paylaşılan bilgilerin hesaplanması da işlemci üzerine önemli bir yük olarak ortaya çıkar (Boulanger, 2006).

Ağ gecikmesi, bir bilginin ağ üzerinde kaynaktan hedefe ulaşması için gereken zamandır. Gecikme, temel olarak fiziksel donanımla sınırlanmış olsa da ağ üzerinden gönderilen büyük veriler önemli bir gecikme ortaya çıkarırlar. Gecikmeyi önlemek için geliştirilmiş, parakete hesabı (Dead Reckoning), mesaj toplama (Message Aggregation) gibi yöntemler bulunmaktadır (Boulanger, 2006).

MMORPG'ler üç bölümden oluşmaktadır, bu yüzden MMORPG geliştirme süreci de üç bölüme ayrılır. Bu bölümler (Sladjan Bogojevic, 2003):

- Oyun Mimarisi
- Sunucu Mimarisi
- İstemci Mimarisi

#### **4.1.1 Oyun Mimarisi**

MMORPG'lerde oyuncular sürekli olarak yaşamın devam ettiği bir dünyanın içindedir. Oyuncu oyunda olduğu zamanlarda sahip olduğu karakter ile bu dünyada yaşar. Oyunda olmadığı anlarda ise oyun dünyasında yaşam devam eder ancak oyuncunun karakteri bu dünyada yer almaz. Oyuncu tekrar oyuna girdiğinde her şeye bıraktığı yerden devam eder. Bu bilgiler ışığında, oyun mimarisinin kullanıcının bütün verilerini saklayacak şekilde geliştirilmesi gerekliliği ortaya çıkmaktadır.

Oyunda binlerce, hatta milyonlarca oyuncu yer alabilir. Her bir oyuncunun kişisel bilgileri, karakter bilgileri, sahip olduğu nesnelere ve diğer özellikleri göz önüne alındığında saklanacak veriler çok büyük olacaktır. Ayrıca, verilerin başkalarının eline geçmemesi ve hile yapılmaması için güvenli bir ortamda saklanması gerekmektedir. Bunun yanında, verilerin periyodik olarak yedeklenmesi gereklidir. Tüm bunlar göz önüne alındığında, veri tabanları (SQL Server, Oracle, MySQL vb.) verileri saklamak için en uygun yerlerdir (Mulholland & Hakala, 2004).

Oyun mimarisinin bir ayağı da oyunun içindeki görevler, maceralar ve bunların bir parçası olan diğer canlılar ve nesnelere arasındaki etkileşimdir. Oyundaki görevler ve maceralar geliştirilirken temel felsefe, oyun içerisinde herkesin her şeyi yapabilme şansının olmasıdır. Ancak bazı kriterlerle herkesin her görevi yapamaması sağlanarak oyuncular motive edilmelidir. Motivasyonu sağlamak için çeşitli görev türleri hazırlanabilir. Bunlar şöyle sıralanabilir (Hall & Novak, 2008):

*Tekrarlanan Görevler:* Görevi tamamlayana bir ödülün verileceği durumlarda bu tür görevler hazırlanabilir. Örneğin, oyuncu sağlık durumunu iyileştirmek için sağlık nesnesine her ihtiyaç duyduğunda aynı görevi tekrarlayabilir.

*Bir Kerelik Görevler:* Oyuncunun sadece bir defa tamamlayabileceği görevlerdir. Oyunda görevin açılacağı vakit, daha önceden oyunculara duyurulur ve rekabet ortamı oluşturulur. Ayrıca bu tür görevler sık sık oluşturularak oyuncuların oyunda daha uzun süreler geçirmesi sağlanabilir.

*Övündürücü Görevler:* Seviyesi yüksek ve düşük oyuncular için aynı zorlukta görevler hazırlanarak, yüksek seviyeli oyuncuların görevleri kolayca çözebildiği, düşük seviyeli oyunculara gösterilmelidir. Bu sayede yüksek seviyeli oyuncular kendileri ile övünürken, düşük seviyeli oyuncuların da kendilerini daha çok geliştirmeye çalışmaları sağlanabilir.

*Özel Hediyeli Görevler:* Tamamlayanlara çok özel hediyelerin verildiği görevlerdir. Örneğin, kırmızı bir şapka sadece 5. görevi bitirenlerde bulunur. Böylece o görevin değeri artırılmış olur.

Oyun mimarisinin ögelerinden biri de oyuncu topluluklarıdır. Oyuncular isterlerse kendi aralarında gruplar (clan, guild) kurabilmelidir ya da daha önce oluşturulmuş gruplara üye olabilmelidir.

Oyuncular yaşadıkları, görevler tamamladıkları ve gruplar kurdukları bu sanal dünyalarda karakterlerini sürekli olarak geliştirirler. Bu karakter gelişiminin doğru ve dengeli bir şekilde gerçekleşmesi çok önemlidir. Örneğin, oyuncu uzun süre oynayarak herhangi bir gelişim gösteremediğinde oyundan sıkılabilir.

Oyuncu oyundaki yeteneklerini yapacağı alıştırmalarla geliştirmeli ve bu gelişim konusunda uygun bir geri bildirimle bilgilendirilmelidir. Örneğin, robotlarla savaşırken lazer kılıcı kullanan bir karakter önceleri sürekli ıskalar; fakat oyunda pratik yaptıkça yeteneği gelişir ve daha az ıskalamaya başlar. Bu gelişim pratik yaptıkça devam eder. Ayrıca bu süreç bir ilerleme çubuğuyla oyuncuya gösterilebilir.

#### **4.1.2 Sunucu Mimarisi**

Sunucu mimarisi denilince ilk akla gelen, sunucu olarak kullanılacak bilgisayardır. MMORPG türünde yüzlerce, hatta binlerce kişinin aynı anda oynayabileceği bir oyun için sunucu bilgisayar kullanmak gerekir. Bunun yanında hızlı bir internet bağlantısı olmalıdır (McFarlane, 2005).

Sunucu mimari olarak, giriş sunucusu ve oyun sunucusu şeklinde iki parçaya ayrılmalıdır. Giriş sunucusu, kullanıcıların oyun sistemine giriş işlemlerini gerçekleştirirken oyun sunucusu da asıl işi olan oyun içi hesaplamaları yapar. Bu sayede, kullanıcılar sisteme giriş yaparken oluşabilecek gecikmeler engellenmiş olur.

Ayrıca oyunun sunucusunun yanı sıra oyun hakkında genel bilgilerin, kuralların, tartışma sayfalarının, oyunda yapılan geliştirmelerin, sıralama listelerinin, oyunun kurulum dosyalarının ve kurulumun nasıl yapılacağı gibi bilgilerin yer aldığı, oyuncularla oyun arasındaki iletişimin sağlandığı bir web sunucusuna da ihtiyaç vardır (Mulholland ve Hakala, 2004).

Web sunucusu üzerinde, oyun sunucusunun kullandığı veri tabanına erişim bulunmalıdır. Böylece oyundaki veriler gerçek zamanlı olarak web sayfalarına



yansıtılabilir. Aynı zamanda oyuncuların şifre değiştirmek, nesne satın almak gibi işlemleri web sunucusundan yapmaları, oyun sunucusun yükünün azaltılmasını sağlayabilmektedir.

### **4.1.3 İstemci Mimarisi**

Tüm grafiksel ara yüzün olduğu kısımdır. Oyuncu, grafik ortamda hareket eder ve çeşitli etkileşimlerde bulunur, istemci de oyuncunun yaptıklarını sunucuya gönderir ve dönen veriler doğrultusunda grafik ortamı düzenlenir. Kullanıcının gördüğü bu grafikler ve kullandığı ara yüz, olabildiğince kendisi tarafından, zevklerine uygun olarak yeniden tasarlanabilecek yapıda olmalıdır.

Oyun sisteminde zamanla çeşitli yenilikler yapılabilir ya da fark edilen hatalar düzeltilebilir. Böyle durumlar için istemci mimarisi, kendi kendini güncelleyebilen bir yapıda oluşturulmalıdır.

## **4.2 elektroGame Oyununun Özellikleri**

elektroGame oyunu geliştirilirken göz önünde bulundurulmuş, tipik bir MMORPG’de bulunması gereken eğlenceye yönelik özellikler ve eğitsel amaçları gerçekleştirebilmek için eklenmiş olan öğretime yönelik özellikler bu bölümde anlatılmıştır.

### **4.2.1 elektroGame Oyununun Eğlenceye Yönelik Özellikleri**

MMORPG türü oyunlarda bulunan ve oyuncuyu eğlendiren dolayısıyla ortama bağlanmasını sağlayan öğeler elektroGame oyununda da yer almaktadır. Eğlence unsuru olarak kullanılan öğelerin elektroGame’e özgü yapısı altı başlık altında incelenmiştir.

#### 4.2.1.1 Tema

Tema, MMORPG türündeki oyunların genel özelliklerinden birisidir. Oyunun oynanışı, ilerleyişi, görevleri, karakterleri, karakterlerin özellikleri, görsel öğeleri kısacası oyuncu tarafından görülen ve kontrol edilebilen her şey, bu tema temel alınarak inşa edilir. Oyunun hikâyesi de temanın temel ögesidir. elektroGame'in hikâyesi yazılırken öncelikle hem erkek hem de kız oyuncuların (öğrencilerin) ilgisini çekecek, güncel ve öğretimi yapılacak derslere uygun ve esnek bir konu olmasına dikkat edilmiştir.

Hikâye yazımına "Mana" adı verilen hayali (fantezi) bir dünya tasvir edilerek başlanmıştır. Mana RPG oyunlarda doğüstü gücü tanımlamak için kullanılan bir ifadedir. Ayrıca elektroGame için tasvir edilen bu dünyaya Mana adı verilmesi elektroGame'e kaynaklık eden The Mana World oyununa bir gönderme amacı taşımaktadır. Hikâyeye göre, tasvir edilen dünyada önceleri huzur hâkimken daha sonra insanlar arasında büyük savaşlar çıkmış ve dünyanın her yerini başıboş gezen savaş artığı robotlar sarmıştır. İşte oyuncuların görevi de bu robotlarla mücadele ederek Mana Dünyası'nı temizlemek ve etkisiz hale getirilen robotları geri dönüşüme göndererek Mana Dünyası'nın yeniden inşasına katkıda bulunmaktır.

Oyuncular robotlarla mücadele etmeyi ve kendilerini geliştirmeyi de yine bu dünyada yer alan bilgilerden öğreneceklerdir. Web sitesinde ve oyunun içinde oyunculara sunulan "oyunun hikâyesi" tam olarak şöyledir:

*"Çok eski zamanlarda, Mana Dünyası diye anılan güzel bir gezegen vardı. Burası masmavi okyanusları ve yemyeşil bitkileriyle insanların yaşaması için eşsiz bir cennetti. Burada suç diye bir şey yoktu, tüm canlılar birbirini sever ve birbirine saygı duyardı. Herkes eşit şartlarda mutluluk ve huzur içinde yaşardı.*

*Uzun süren mutlu zamanlardan sonra, insanoğlu suç eğilimli kuşaklar yetiştirmeye başladı. Cennetlerinin değerini çok iyi bilen eski nesiller, ona özen göstermeyi kendi çocuklarına yeterince aşılayamadılar. Mutlu günlerin sonsuza kadar süreceği düşünülürken, iyilik öğretilmemeye, bu yüzden de insanlar birbirlerini ve gezegenlerini sevmemeye başladılar. Her yerde savaşlar baş gösterdi, insanlar kendi cennetlerini tahrip etmeye başladı. İyi duyguların yerini kötülerini aldı. Sonunda kötü niyetli insanların ürettiği tehlikeli robotlar kontrolsüz biçimde çoğalarak her yeri işgal etmeye başladı.*

*Robot işgalleri nedeniyle insan soyu tükenme tehlikesiyle karşı karşıya kaldı. Bunun üzerine zaten sayıları çok azalmış olan insanlar başka bir diyara göç etmek zorunda kaldılar. Bir taraftan da özel olarak üretilmiş uçaklarıyla kendi gezegenlerine ulaşımı sağlamaya devam ettiler. Geride bıraktıkları dünyalarında kötü gidişi durdurmaya çalışan bir avuç insan kalmıştı. Bunlar robotlara karşı gezegenlerini savunabilmek için "Santral" adı verilen tampon bölgeler kurdu. Gezegenin tüm cadde ve sokaklarını robotları zararsız hale getiren geri dönüşüm kutularıyla donattılar.*

*Başka gezegene göç edenler, olanları duyunca Mana Dünyası'nda kalanlara yardım ederek gezegenlerini robotlardan birlikte kurtarmaya karar verdiler. Kalan savunmacılar, diğerlerine bu işin nasıl başarılacağına ilişkin planları anlatıp gerekli eğitimleri verdiler. Artık göç etmiş olanların yapması gereken, özel uçaklarıyla gezegenlerine dönmek ve sokaklarda tehlike saçarak dolaşan robotları geri dönüşüm kutularında zararsız hale getirmektir. Ancak bunu yaparken önlerine çıkan bazı soruları cevaplamaları ve kimi engelleri aşmaları gerekiyordu. Mana Dünyası ancak böyle eski mutlu günlerine kavuşacaktı..."*

elektroGame'in ilerleyen sürümlerinde hikâyenin oyuncuyu daha çok içine çekmesi amacıyla çizgi roman ve animasyon biçiminde de sunulması planlanmaktadır.

elektroGame haritalarında bulunan bazı ortamlar, bu tema düşünülerek sanayi ve savaş artıklarıyla kirlenmiş olarak tasarlanmıştır. Kullanılan renkler de bu yapıya uygun olarak orman haritalarında canlı ve daha renkli, diğer haritalarda biraz daha karanlık bir atmosfer oluşturacak şekilde tercih edilmiştir.

Her oyunda olduğu gibi elektroGame'de de atmosferi güçlendirmek için ses efektleri ve arka plan müziği kullanılmıştır. Kullanılan arka plan müzikleri (4 parça), tasarım grubuna yardımcı olan bir lisans öğrencisi tarafından bu oyun için bestelenmiştir.

#### **4.2.1.2 Oyuncu (Char)**

Tüm MMORPG'lerde olduğu gibi elektroGame'de de oyuncuyu temsil eden ve oyuncu tarafından özelleştirilebilen karakterler bulunmaktadır. RPG oyunlarda **avatar** ya da **char** olarak anılan bu karakterler hem oyuncuyu oyun içinde temsil etmektedir hem de yeni özellikler kazanarak oyuncunun oyun içindeki gelişimini

göstermektedir. Ayrıca, bu karakterlere oyuncu tarafından kıyafet ve nesnelere eklenerek bunların diğer oyuncuların karakterlerinden farklılaşması sağlanabilmektedir.

Şekil 4.1 ve Şekil 4.2’de görülen sprite, oyuncunun başlangıçta sahip olduğu en yalın karakterdir. İnsan şeklindeki karakterin hem erkek hem de kız şekli mevcuttur. Oyuncular hesapları ilk açtıklarında cinsiyet seçimini yaparlar. Daha sonra öğrencinin bu bilgiyi değiştirmesi mümkün değildir. Ancak başlangıçta hatayla yanlış girilen bir bilgi varsa öğretmenin ya da oyun yöneticisinin yetkisi ile düzeltilebilmektedir.



Şekil 4.1: Kız oyuncu sprite'ının tüm kareleri.



**Şekil 4.2:** Erkek oyuncu sprite'i.

Bir char'ın özelliklerini belirleyen pek çok değişken vardır. Char bu değişkenlerin birleşimi olarak oyun ortamıyla etkileşir. Örneğin, saldırı gücü 10 puan olan char, lazer tabancası kullanarak saldırı gücünü 15 puana çıkartabilir. Ancak saldırı gücünü arttıran silah, ağırlığı da arttırdığı için char'ın hareketlerini yavaşlatarak savunmada zafiyete neden olabilir. Aşağıda bu değişkenlerden en önemlileri açıklanmıştır.

**Seviye:** Oyuncunun oyunda ne kadar ilerlediğini gösteren en önemli değişkendir. Diğer bazı özelliklere sahip olup olamayacağına seviyeye göre karar vermek mümkündür. Örneğin, sağlık puanının daha hızlı yükselmesini sağlayan oturma özelliği 5. seviyeye ulaşıldıktan sonra mümkün olmaktadır. Ayrıca oyun içinde gelişimi teşvik etmek amacıyla bazı haritalara ve odalara giriş için, bilgilerle konuşabilmek için seviye sınırı konmuştur.

**Para:** Oyun parası altın olarak belirlenmiştir. Bu parayla, nesne satan bilgilerden alışveriş yapmak mümkündür. Özellikle kıyafet ve silah alımı için para gerekmektedir. Para kazanmak için robotlarla yapılan mücadelede elde edilen nesnelere satılabileceği gibi, bilgelerin verdiği görevler de tamamlanabilir.

**Uzmanlık:** elektroGame'in mevcut versiyonunda tam olarak kullanılmamaktadır. Ancak robotları geri dönüşüme gönderdikçe ve bilgelerin görevlerini tamamladıkça artan bir değerdir. Sonraki versiyonlarda "acemi", "usta", "pir" gibi derecelere ulaşmak için kullanılması planlanarak eklenmiştir.

**Sağlık:** Char'ın ne kadar sağlıklı olduğunu gösterir. Sağlık puanı, sıfır olduğunda, oyuncun başlangıç kısmına ışınlanarak oyunun baştan başlaması sağlanmaktadır. Sağlık puanı, robotlarla yapılan mücadelelerde ya da bilgelerin verdiği görevler tamamlanmadığında ceza olarak azalmaktadır. Düşen sağlık puanını daha hızlı yükseltmek için 5. seviyeden sonra kazanılan oturma özelliği kullanılabilir. Sağlık puanını yükseltmek için bir diğer seçenek de nesnelere

kullanmak veya doktora gitmektir. Nesnelere bölümünde açıklanacağı gibi, bazı nesnelere sağlık puanını yükseltici özelliği bulunmaktadır. Hastaneye gidildiğinde de doktora altın verilerek tedavi olunca sağlık puanı yükselmektedir.

**Tecrübe:** Seviye atlamak için toplanması gereken puandır. Belirli bir tecrübe puanına ulaşıldığında, oyuncu bir seviye yükselmektedir. Seviye arttıkça ilerlemek için gereken tecrübe puanı da yükselmektedir. Böylece ilerlemek isteyen oyuncu, eski görevlerle yetinmeyerek daha yüksek puan veren daha zorlu görevleri tamamlamak zorunda kalacaktır.

**Uzman:** Uzmanlık puanını yükseltmek için gerekli puan miktarıdır.

**Güç:** Char'ın bir atak sırasında karşısındaki robottan kaç sağlık puanı alabileceğini göstermektedir. Char'ın gücü kullanılan silah ve diğer nesnelere artabilmektedir. Burada belirtilen puan hiç bir nesnenin etkisi olmaksızın sahip olunan ham güç puanıdır.

**Çeviklik:** Çeviklik puanı yükseldikçe iki hamle arasındaki zaman kısalmaktadır. Böylece birim zamanda yapılan atak sayısı artarak robotlara karşı daha etkin mücadele sağlanmaktadır.

**Dayanıklılık:** Robotlar tarafından yapılan ataklara karşı savunma puanını gösterir. Atak gücü fazla olan robotlar, atak gücü oranında oyuncunun sağlık puanını düşürebilmektedir. Savunma puanı, karşı tarafın atak gücünün etkisini azaltarak char'ın daha az zarar görmesini sağlamaktadır.

**Zekâ:** Mevcut versiyonda aktif olarak kullanılmamaktadır. Sadece, zekâ puanı arttıkça robotların savunmasını daha kolay aşacağı düşüncesiyle, oyuncunun atak gücüne küçük bir katkı yapmaktadır. İlerleyen versiyonlarda robotlardan düşen parçaların niteliğini arttırmak gibi etkilerinin olması planlanmaktadır.

**Kabiliyet:** Nesnelere daha etkin kullanılmasını sağlar. Örneğin, atak gücünü 5 puan arttırabilen bir lazer kılıcı, daha yüksek kabiliyet puanı ile daha bu gücü fazla arttırabilmektedir.

**Şans:** Robotlardan nesne düşme ihtimalini yükseltir. Ayrıca bazı bilgelerin verdiği ödül miktarlarında artışa ve ceza oranlarında azalmaya neden olabilir.

**Savunma:** Ham dayanıklılık puanı ile nesnelerin kazandırdığı savunma puanı, toplam savunma gücünü gösterir.

**Atak:** Ham atak gücü ile nesnelerin kazandırdığı atak puanı, toplam atak gücünü gösterir.

Char'ın yukarıda anlatılan bu özellikleri ve puanları oyuncuya, **Şekil 4.3**'teki ekran yardımıyla gösterilmektedir.



**Şekil 4.3:** Char'ın durumunu gösteren durum penceresi.

**Karakter Puanı:** Seviye yükseldiğinde ve bazı bilgelerin ödülü niteliğinde karakter puanı kazanılır. Bu puan, yukarıda bahsedilen özelliklerin artırılmasında kullanılır. Artı düğmelerinin sağında bulunan sayı, o özelliği bir puan arttırmak için gereken karakter puanını göstermektedir.

Ayrıca, testlerde soru başına düşen zamanı etkileme, nesnelerin kullanım miktarını artırma gibi özellikler de tasarlanmıştır. Ancak uygulamanın yapıldığı versiyonda kullanılmamıştır.

### 4.2.1.3 Robotlar

Robotlar, elektroGame'in hikâyesinde yer alan Mana Dünyası'nda başıboş dolaşan savaş artığı makinelerdir. Oyuncuların temel görevi, bu robotlarla mücadele ederek onları geri dönüşüme göndermektir.

Robotlar, Mana Dünyası'nın her yerinde bulunmaktadır. Birçok farklı robot mevcuttur. Bunlar atak gücü, savunma gücü, görünüm gibi özellikleriyle birbirlerinden ayrılmaktadır. Robotların böyle farklı özelliklerde olmasının sebebi, her seviyede oyuncunun mücadele edebileceği türden robotlara ihtiyaç duyulmasıdır. Oyuna yeni başlayan bir oyuncu, atak ve savunma değerleri daha düşük zayıf robotlarla mücadele edebilirken, oyunda ilerlemiş, daha deneyimli ve güçlü char'a sahip oyuncular daha güçlü robotlarla mücadele edebilmektedir.

Oyuncular, robotlarla mücadele ederek onların geri dönüşüme gönderdiklerinde, rastgele zamanlamalı olarak bu robotlardan nesnelere düşmektedir. Düşen nesnelere oyun içindeki önemi, robotların güçleri ile doğru orantılıdır. Bu şekilde oyuncuların oyunda ilerlemeye teşvik edilmesi ve motivasyonlarının artması beklenmektedir.

Robotlar, saldırgan ve saldırgan olmayan olmak üzere iki türe ayrılır. Saldırgan olmayan robotlar yalnızca başıboş bir şekilde Mana Dünyası'nda dolaşırlar ve oyuncu onlara saldırmadığı sürece oyuncuya saldırmazlar. Saldırgan robotlar ise yakınlarında bulunan oyunculara, kendilerine bir hamle yapılmasını beklemeden saldırırlar. **Şekil 4.4**'te görülen sprite, saldırı yönünden güçlü, savunması zayıf fakat saldırgan olmayan Magot isimli robota aittir. **Şekil 4.5**'te elektroGame'de kullanılan bütün robotlar bir arada gösterilmiştir.





Şekil 4.4: Oyunda kullanılan bir robotun sprite'ı.



Şekil 4.5: elektroGame'de kullanılan robotlar.

#### 4.2.1.4 Bilgeler

Bilgeler MMORPG'lerde "NPC" (Non Player Character - Oyuncusuz Karakter) olarak bilinen, oyuncular tarafından kontrol edilemeyen ancak etkileşime girilebilen, oyunun kendi karakterleridir (Wikipedia, <http://tr.wikipedia.org/wiki/NPC>, 2011).

elektroGame'deki bilgiler, robotlar ya da karakterler gibi hareket etmezler, sabit bir konumları vardır. Ancak oyuncular etkileşime geçtiğinde bilgiler karşılık verirler. Oyuncular, bilgilerle etkileşime geçmek için bilgenin üzerine farenin sol

düğmesi ile tıklamalıdır. Şekil 4.6’da kıyafet satan bir bilge ve üzerine tıklandığında açılmış olan kıyafet alım satım penceresi görülmektedir.



Şekil 4.6: Alışverişte kullanılan bir bilgenin açtığı pencere.

Bilgeler Mana Dünyası'nın her yerinde bulunmakta ve birçok görevi yerine getirmektedir. Bilgelerin elektroGame'deki görevleri;

- Oyunun genel tanıtımını yapmak,
- Eğitim dünyasında oyunu öğretmek,
- Oyuncuya yapması gereken görevleri bildirmek ve yapmış olduğu görevleri listelemek,
- Oyuncuyu, Mana Dünyası'nda gideceği yerler hakkında bilgilendirmek ve yönlendirmek,
- Öğretimi yapılacak konular hakkında bilgi sunmak,
- Sorular sormak ve verilen cevapları değerlendirmek,
- Elektrik devreleri sunmak ve oyuncunun yaptıklarını değerlendirmek,
- Oyuncu ile nesne alışverişi yapmak

şeklinde dir.

#### 4.2.1.5 Nesnelere

Oyuncuların, oyun sırasında çeşitli amaçlarla kullanabildiği MMORPG’lerde “item” olarak bilinen öğelere nesne denir. elektroGame’de kullanış amaçlarına göre;

- Silah
- Mühimmat
- Yiyecek
- Kıyafet
- Basit elektrik devresi elemanları
- Diğer

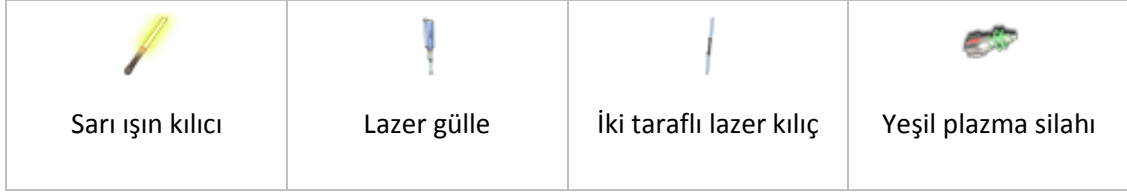
nesnelere mevcuttur. Oyuncu, oyun içindeki çeşitli görevleri tamamlayarak ya da satın alarak ihtiyacı olan nesnelere toplamaya çalışır. Toplanan nesnelere Şekil 4.7’de verilen “Heybe” penceresinden görülebilir. Oyuncu, heybesindeki nesnelere kullanarak kendi özelliklerini iyileştirebilir, bozulan sağlığını düzeltebilir ya da bir devreyi tamir edebilir. Nesnelere etkileri ve örnekleri aşağıda verilmiştir.



Şekil 4.7: Oyuncunun nesnelere gösteren heybe penceresi.

**Silah:** elektroGame’de robotları geri dönüştürmek için kullanılan nesnelere dir. Silahlar genellikle atak gücünü artırırken, hareket ve savunma gibi diğer yeteneklerin azalmasına neden olurlar. Ayrıca plazma silahları mühimmata ihtiyaç

duyarlar. Oyun içinde mühimmat genellikle parayla satın alınmakta olup bazı bilgiler de ödül olarak bir miktar mühimmat verebilmektedir. Mühimmatı biten silah etkisiz hale gelmektedir. **Şekil 4.8**'de, kullanılan silahlara örnek verilmiştir.



**Şekil 4.8:** Silah nesnesi örnekleri.

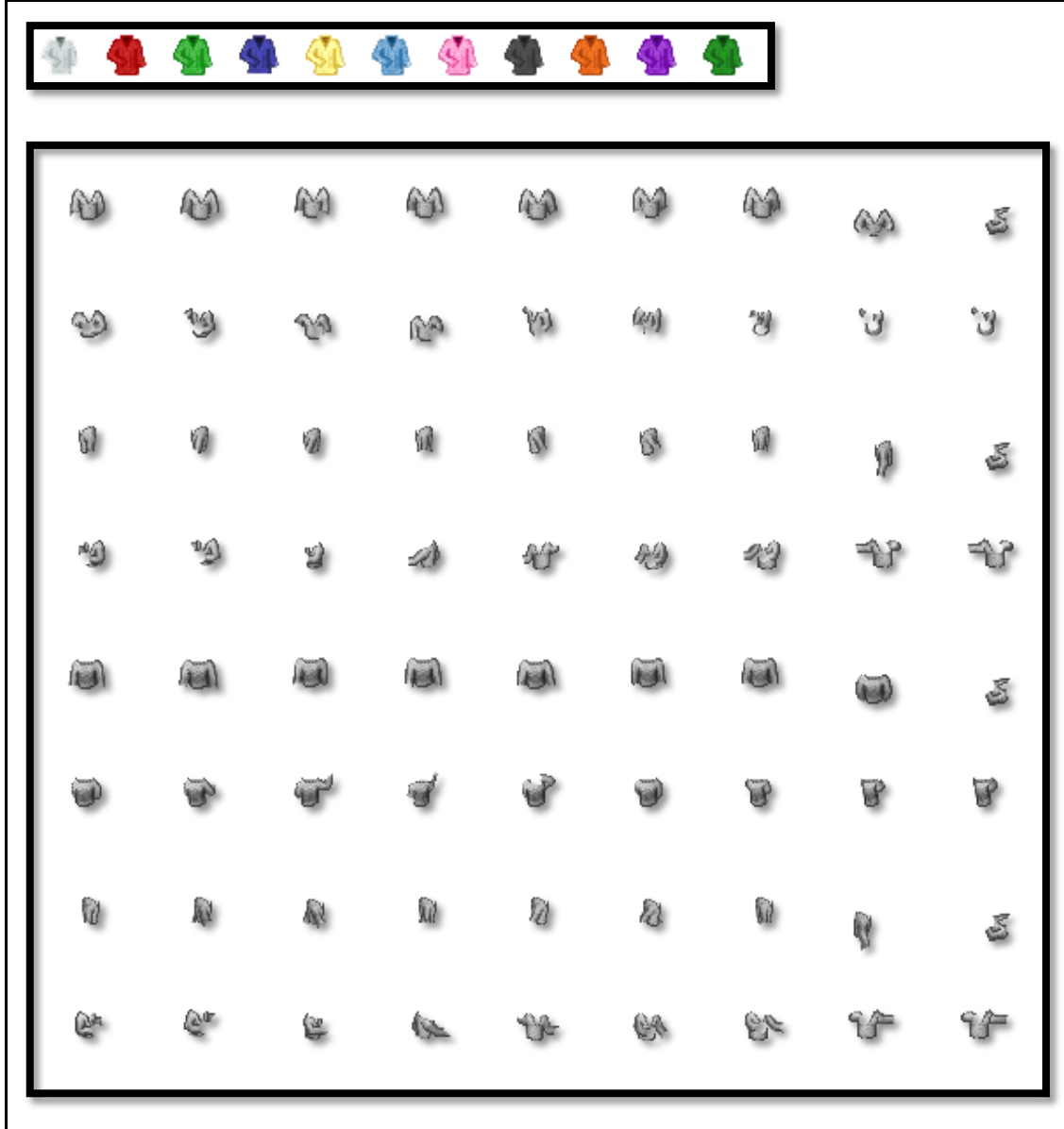
**Mühimmat:** Üç farklı enerji seviyesine sahip mühimmat bulunmaktadır (**Şekil 4.9**). Sadece plazma silahlarıyla birlikte kullanılabilir. Oyuncu, heybesinde iki veya daha fazla çeşit plazma bulunduğunda aralarında seçim yapabilmektedir.



**Şekil 4.9:** Plazma silahlarında kullanılan plazma çeşitleri.

**Yiyecek:** Oyunda yiyecekler robot saldırıları sonucunda char'ın düşen sağlığını düzeltmek için kullanılır. Bazı yiyecekler sağlık puanının yanı sıra yüksek hız ve atak gücünü geçici bir süre yükseltmektedir. Bu tip enerji veren yiyecek nesnelere, genellikle zor görevlerin tamamlanmasının ardından bilgiler tarafından verilmekte ve güçlü robotlarla mücadele etmek için kullanılmaktadır.

**Kıyafet:** elektroGame'de char'ın özelleştirilebilmesi ve savunma gücünün artırılması için kullanılırlar. Mümkün olan her kıyafet için 12 renk alternatifi hazırlanmıştır. Böylece aynı kıyafetleri seçmiş olsalar bile öğrencilerin farklı görünüme kavuşmaları sağlanmaya çalışılmıştır. **Şekil 4.10**'da ilk satırda alternatif renkleriyle görünen “v yaka kazağın” oyun içinde char'ın giyebilmesi için kullanılan sprite'ı verilmiştir.



Şekil 4.10: elektroGame’de kullanılan bir kıyafet nesnesinin grafikleri.
















elektroGame’de yaklaşık olarak 200 çeşit kıyafet bulunmaktadır. Bu kıyafetler vücudun bölgelerine göre:

- Baş
- Gövde
- Bacak
- Ayak
- El

olarak ayrılmıştır. Ayrıca kıyafetler türlerine göre aşağıdaki gibi gruplandırılmış ve kazandırdıkları özelliklerin tutarlı olması amaçlanmıştır. Kıyafet türleri ve alabilecekleri renkler **Tablo 4.1**’de görülebilir.



**Tablo 4.1:** Kıyafetlerin gruplandırılmış listesi.

Tişört	
V yaka kazak	
Balıkçı yaka	
Elbise	
Fanila	
Kısa tişört	
Zırh	
Etek	
Pantolon	
Ayakkabı	
Eldiven	
Çöl şapkası	
Saç bandı	
Şapka	
Maske	

Oyuncu bu kıyafetler içinden istediği kombinasyonu yaparak char'ını geliştirebilmektedir. Bazı kıyafetler ise sadece belli amaçlar için ayrılmıştır. Örneğin,

geliştiriciler ve öğretmenler için kullanılan şapkalar diğer oyuncular tarafından kullanılamamaktadır. Şekil 4.11’de bir oyuncunun ekipman penceresi görülmektedir.



Şekil 4.11: Ekipman penceresi.

**Basit Elektrik Devresi Elemanları:** Oyun içinde bilgelerin gösterdiği basit elektrik devrelerini tamir etmek, tamamlamak için kullanılan nesnelere. 1Ω, 50Ω arası 19 adet direnç, 1V, 20V arası 7 adet pil, 7 farklı renkte lamba, LED çeşitleri ile ampermetre, anahtar ve motordan oluşmaktadır. Şekil 4.12’de elektrik devrelerinde kullanılan nesnelerin listesi verilmiştir.



Şekil 4.12: Basit elektrik devrelerinde kullanılan nesnelere.

Devre elemanlarını elde etme yollarından biri robotları geri dönüşüm kutusuna göndermektir. Daha nadir bulunan nesnelere ise birkaç aşamadan oluşan bazı görevlerin tamamlanması halinde bilgiler tarafından hediye edilmektedir.

**Diğer:** Yukarıdaki grupların hiçbirine girmeyen, çoğunlukla da gerçekte bir karşılığı olmayan nesnelere. Kullanılış amaçları değişmekle birlikte, genellikle bir

bilgeyle konuşabilmek, bir binaya girebilmek gibi anahtar görevleri vardır. **Şekil 4.13**'te çeşitli nesnelerin listesi verilmiştir.



**Şekil 4.13:** Diğer nesnelerin listesi

Oyuncular çözdükleri sorular ve yaptıkları ticaret sonucunda, oyun içinde kullanabilecekleri altın biriminde para kazanmaktadırlar. Kazandıkları bu paraları oyunda;

- Nesne satan bilgilerden nesne almak,
- Doktora giderek sağlık seviyesini yükseltmek,
- Diğer oyuncularla ticaret yapmak

için kullanılabilirlerdir.

Oyuncunun karakterinin sahip olduğu para miktarı, char'ın durumunu gösteren durum penceresinde (**Şekil 4.3**) sağ üstte yer almaktadır. Oyundaki bazı görevleri tamamlamak için oyuncular nesnelere satın almak zorundadır. Nesne almak için para sahibi olmaları gerekmektedir. Bu da oyuncuların mutlaka para kazanmalarını gerektirmektedir. Böylece oyuncuların mücadele edebileceği bir alan daha oluşturulmaya çalışılmıştır.



#### 4.2.1.6 Dünyalar

elektroGame’de karakterlerin, nesnelere, robotların, bilgelerin ve oyundaki diğer her şeyin içinde bulunduğu ortamlara dünya denir. elektroGame’de, hikâye gereği birden çok dünya bulunmaktadır. Bu dünyalar şunlardır:

- **Ders Dağıtım Dünyası:** Yeni oluşturulan bir karakterin ilk olarak oyuna girdiğinde içinde bulunduğu dünyadır. Bu dünyada, oyuna ilk kez giren oyuncuya, oyunun oynanışıyla ilgili kısa bilgiler verilmektedir. Ayrıca elektroGame dünyasındaki derslerle ilgili bütün dünyalara gidebilmek için o dünyalara karakteri ışınlayan geçitler vardır.
- **Eğitim Dünyası:** Oyuna ilk giren oyuncuların dağıtım dünyasından sonra yönlendirildikleri dünyadır. Oyunu oynamayı bilmeyen oyuncuların, oyunu oynamayı öğrenmesi için tasarlanmış bu dünyada, aşama aşama oyun oynanırken kullanılacak bütün beceriler anlatılmaktadır.
- **Elektrik Dünyası:** Oyuncular elektrik konuları ile ilgili bütün çalışmalarını bu dünyada gerçekleştirmektedir. Basit elektrik devreleri konusunda bilgi veren ve soru soran bilgiler, direnç vb. elektrik devre elemanları düşen robotlar, elektrik konusundaki eğitim süresince onları yönlendirecek görev bilgisayarları bu dünyada yer almaktadır. Elektrik Dünyası çok geniş bir alan olduğundan, içinde farklı bölgelerine bir anda ulaşmayı sağlayan ışınlama noktaları vardır. Ayrıca, Elektrik Dünyası’nda yer alan birçok binanın kapısından içeri girildiğinde, ufak yeni dünyalara (müze, hastane vb.) ulaşılmaktadır.

#### 4.2.2 elektroGame Oyununun Öğretime Yönelik Özellikleri

Bu bölümde MMORPG’lerde bulunmayan elektroGame’e de eğitsel özellikler katmak amacıyla eklenmiş özelliklerden bahsedilmiştir.

#### 4.2.2.1 Sunu Penceresi

Sunu penceresi, bilgelerin oyunculara bilgi aktarması için geliştirilmiştir. Resim, metin ve animasyonların gösterilmesi için kullanılabilir. Şekil 4.14’te sunu penceresi gösterilmiştir.



Şekil 4.14: Sunu penceresi.

Pencere üzerinde standart olarak “ileri”, “geri” ve “kapat” düğmeleri bulunmaktadır. Bunlara ilave olarak genellikle sol üstte olan ve programcı tarafından yeri değiştirilebilen bir açılan kutu bulunur. Bu kutu gösterilen ekranın başlığını içerir. Henüz gösterilmemiş slaytlar ise sayı olarak görüntülenir. Sunucudan bilgiler geldikçe açılan kutunun içeriği de tamamlanmış olur.

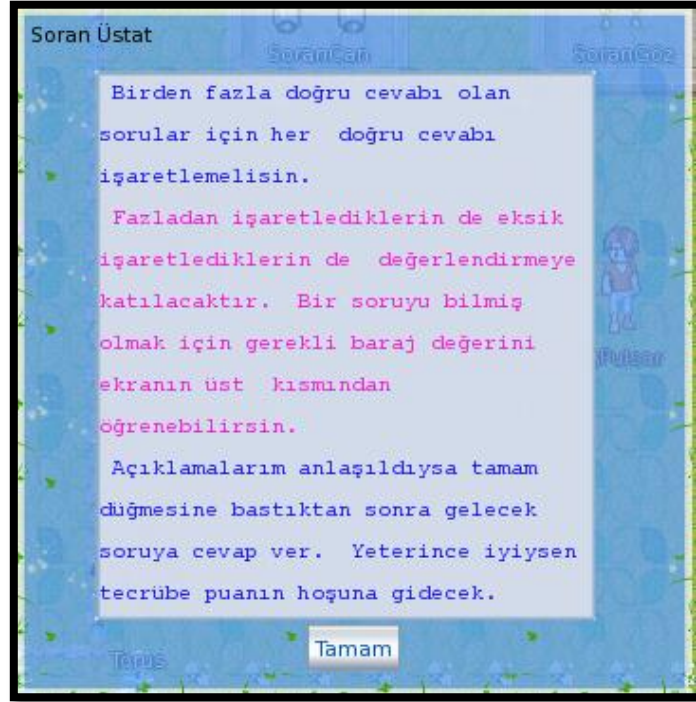
Sunu pencereleri, oyunun genelinde bilgi veren bir bilgenin üzerine tıklatıldığında açılan pencerelerdir, ancak bazı yerlerde oyuncu belli bir bölgeye girdiğinde otomatik olarak da açılabilir. Sunum pencereleri, oyuncuya bilgi vermek, ders anlatmak ve onu yönlendirmek, uyararak kullanılmaktadır.

#### 4.2.2.2 Test Penceresi

Bilgelerin oyunculara soru sorabilmesi için geliştirilmiştir. Test penceresi, sunucudan gelen XML formatındaki veriyi yorumlayarak test ekranının oluşmasını ve oyuncunun vereceği cevabın yeniden sunucuya iletilmesini sağlar. Test penceresi üç farklı yapıda oluşturulabilir:

##### 1. Çoktan seçmeli bir soru

Çoktan seçmeli bir soru olan testlerde, doğrudan soru görüntülenirken birden fazla sorusu olanlarda oyuncuya öncelikle bu teste başlamayı isteyip istemediğinin sorulduğu, Şekil 4.20'deki gibi bir onay ekranı gösterilir. Eğer istenirse Şekil 4.15'te görüldüğü gibi, onay ekranına benzeyen ancak sadece tamam düğmesi bulunan bir açıklama ekranı eklenebilmektedir.



Şekil 4.15: Bir sorulu testlerde isteğe bağlı olarak eklenebilen açıklama penceresi.

Açıklama ekranından sonra Şekil 4.16'da çoktan seçmeli bir soru penceresi gösterilmektedir.



Şekil 4.16: Çoktan seçmeli bir soru penceresi.

Bu pencere diğer soru türlerinde de kullanılmaktadır. Test penceresinin üst kısmında testle ilgili bilgiler yer almaktadır. Alt kısımda, soru ve sağ altta “değerlendir” düğmesi bulunmaktadır.

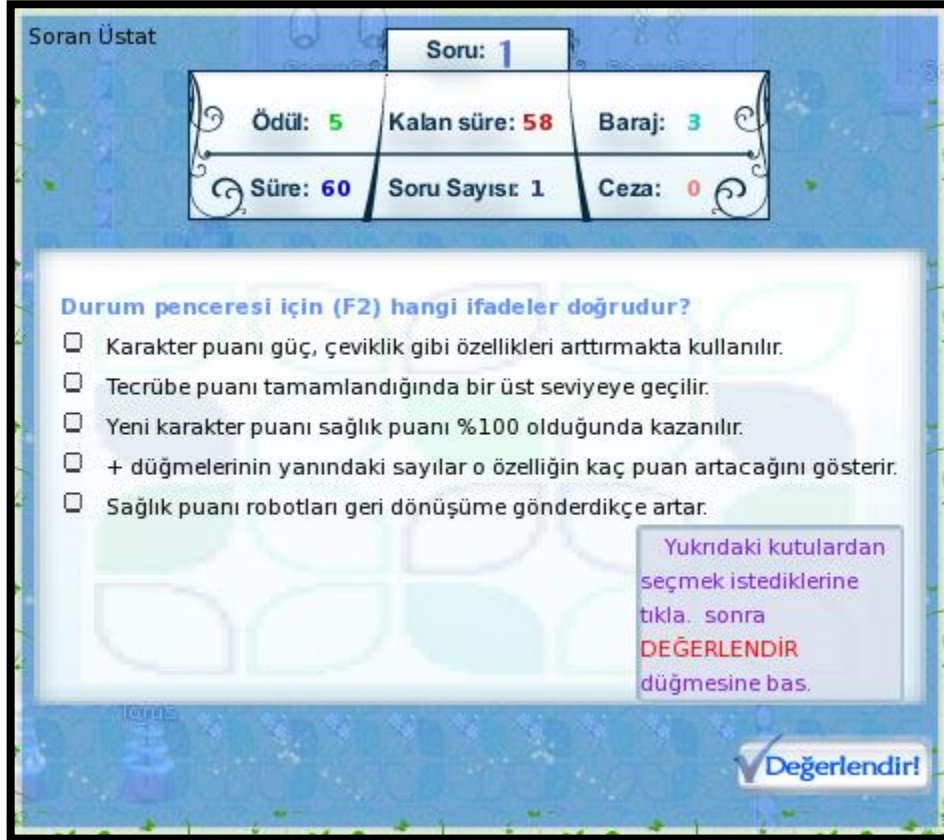
Test ekranlarında oyuncu hakkında daha fazla bilgi toplamak amacıyla, “Değerlendir” düğmesine basmasa bile oyuncunun hangi seçeneği seçtiği ya da değiştirdiği sunucuya gönderilerek kayıt altına alınmaktadır. Süre dolduğunda ya da oyuncu “Değerlendir” düğmesine bastığında durum sunucuya bildirilir. Sunucu cevabı değerlendirerek test ekranına yine XML formatında sonuç penceresinin içeriğini gönderir. Şekil 4.17’de sonuç penceresi görülmektedir.



Şekil 4.17: Çoktan seçmeli bir soruluk testin sonuç penceresi.

2. Onay kutuları ile birden fazla cevabın işaretlenebildiği bir soru

**Şekil 4.18**'de çok seçimli bir tek soru penceresi görülmektedir. Çoktan seçmeli bir test penceresinden farklı olarak burada seçeneklerin birden fazlası doğru olabilir. Uygulama yapılan sürümde, bu türdeki sorulardan oluşan çok sorulu bir test yapılamamaktadır. Ancak sunucu dosyalarında yapılacak bazı geliştirmelerle çok seçimli sorulardan oluşan çok sorulu bir test geliştirilebilir.



**Şekil 4.18:** Onay kutulu bir soru penceresi.

Çok seçimli sorularda, oyuncu "Değerlendir" düğmesine bastıktan ya da verilen süre dolduktan sonra oyuncunun verdiği cevaplar sunucuya gönderilir, değerlendirilir ve sonuçlar değerlendirme penceresinde gösterilir. **Şekil 4.19**'da görüldüğü gibi, her seçenek için oyuncunun verdiği cevabın doğru mu yanlış mı olduğunu gösteren rapor sunulur. Raporun sonunda oyuncunun aldığı ödül veya ceza belirtilir.





Şekil 4.19: Onay kutulu tek sorulu testin rapor penceresi.

### 3. Çoktan seçmeli birden fazla soru

Çok sorulu tek seçimli testlere başlamadan önce Şekil 4.20’de görülen onay penceresinde testteki soru sayısı, toplam süre, testten başarılı olmak için verilmesi gereken doğru cevap sayısı, bir soru için ayrılan zaman ve ödül olarak verilecek puanın bulunduğu bir özet yer alır. Oyuncunun uzun bir süreyi teste ayırmak isteyip istememesi kendi kararına bırakılmıştır; ancak başarılı testlerin sonunda elde edilen ödüller sayesinde ilerlemek isteyen oyuncular, bu testleri çözmeye yönlendirilmiştir.



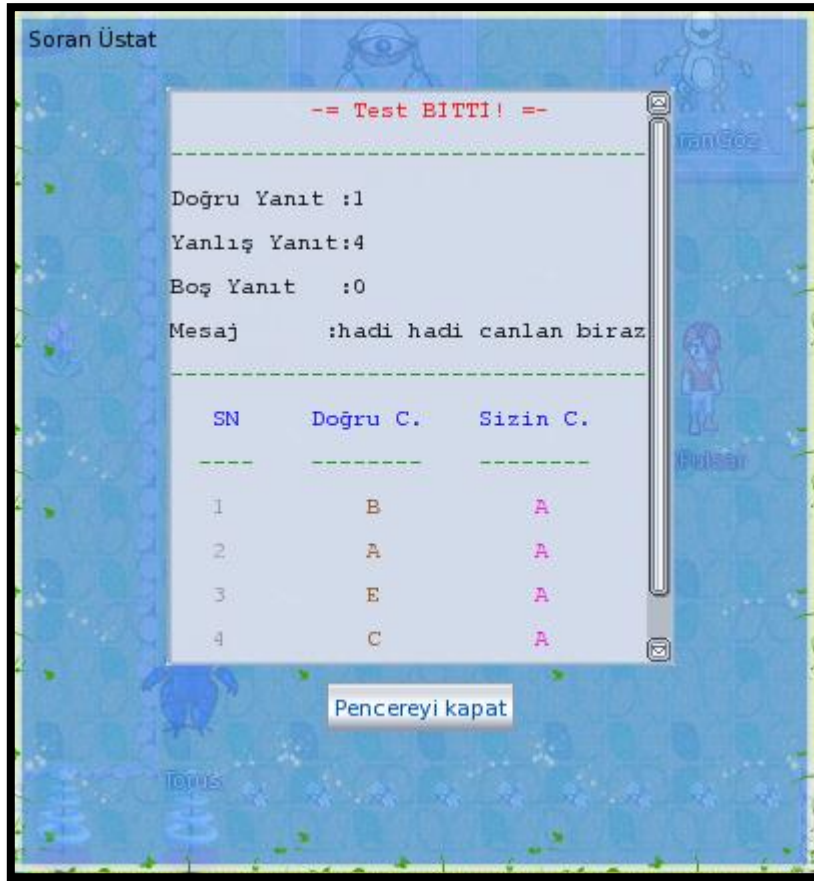
Şekil 4.20: Çoktan seçmeli birden fazla sorulu testin onay penceresi.

Onay penceresinden sonra standart soru penceresine ilave olarak altta sorular arasında dolaşmayı sağlayan yönetim panelinin bulunduğu tek seçimli çok sorulu test penceresi görüntülenir (**Şekil 4.21**). Oyuncu verdiği yanıtı değiştirdiğinde ve sorular arasında geçiş yaptığında, bu bilgi sunucuya gönderilerek kayıt altına alınır.



**Şekil 4.21:** Çoktan seçmeli birden fazla sorulu test ekranı içeriği.

Çok sorulu testin bitiminde (zaman dolduğunda ya da oyuncu “Değerlendir” düğmesine bastığında) rapor ekranı sunucu tarafında XML formatında hazırlanarak test penceresine gönderilir. Test penceresi sunucu tarafından gönderilen veriyi **Şekil 4.22**'deki gibi bir ara birimle görüntüler.



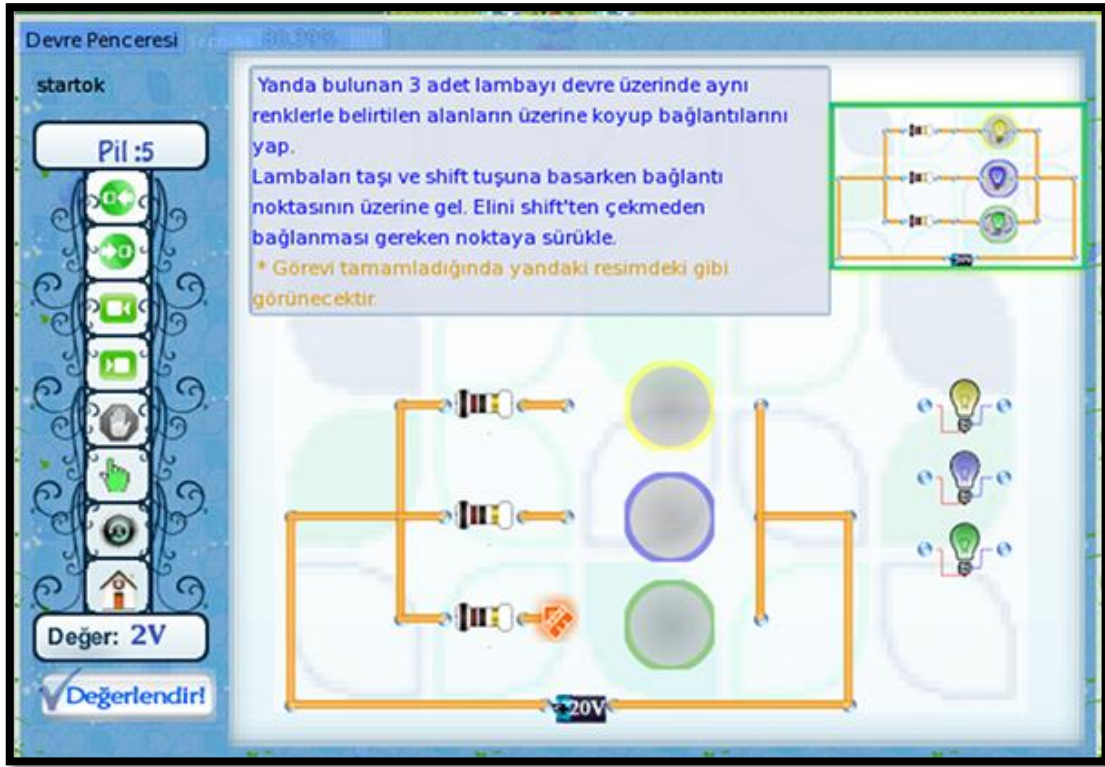
Şekil 4.22: Çok sorulu testin rapor ekranı.

Test penceresi farklı türde soru biçimlerini desteklemek için esnek bir yapıda geliştirilmiştir. Sonraki sürümlerde sürükle-bırak, eşleştirme gibi biçimler eklemek mümkün olabilir. Oyuncuların test penceresi açıkken yaptığı aktiviteler de daha detaylı kayıt altına alınarak ayrıntılı raporlar oluşturulabilir.

#### 4.2.2.3 Devre Penceresi

Devre penceresi, basit elektrik devrelerini oluşturacak bileşenlerin (direnc, lamba, pil, kablo vb.) ve bu devrenin nasıl kurulacağıyla ilgili açıklamanın bulunduğu bir penceredir (Şekil 4.23). Burada kurulan devre simülasyonu, devre çözümlenmesinde kullanılan metotlardan biri olan ilmek akımları metoduyla çözülmektedir. Bu süreçte öncelikle sunucudan gelen XML formatındaki dosya yorumlanarak oyuncunun görmesi gereken basit elektrik devresi oluşturulur ve çözümlenerek kollardan geçen akımlar hesaplanır. Hesaplama sonucunda üzerinden akım geçen lambaların yanması sağlanıp çalışması gereken motorları çalıştırılır.





Şekil 4.23: Devre penceresinin genel görünümü.

Devre penceresi iki ana bölümden oluşmaktadır. Pencerenin sol tarafında yer alan birinci kısım, devre elemanının özelliklerini gösteren bilgi paneli ve “Değerlendir” düğmesinden oluşmaktadır. İkinci kısım ise sunucudan gönderilen verinin yorumlandığı ve simülasyonun yapıldığı bölümdür.

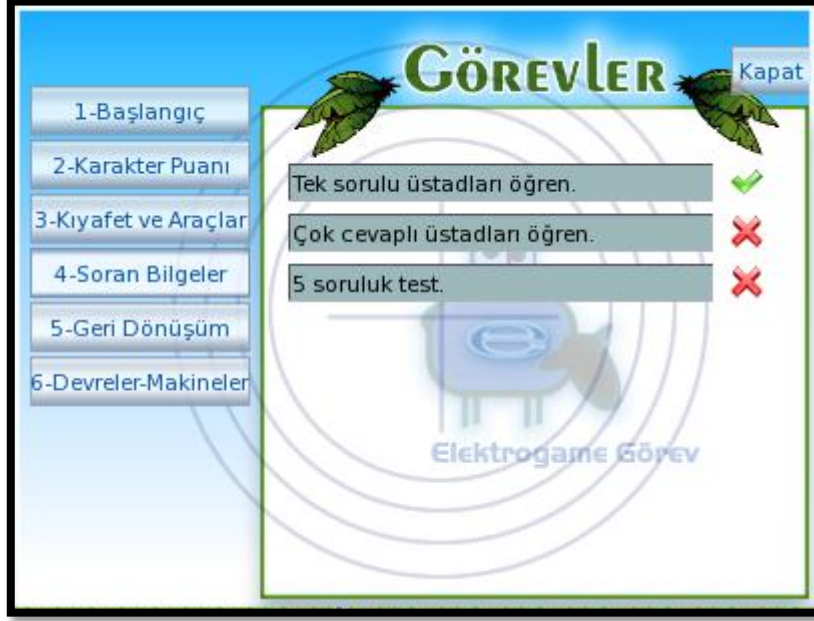
Bu pencerede kullanılacak basit elektrik devresi bileşenleri Şekil 4.24’te listelenmiştir.



Şekil 4.24: Elektrik devrelerinde kullanılabilen elemanlar.

#### 4.2.2.4 Görev Penceresi

Oyuncuların oyun içerisinde hangi görevleri gerçekleştirmeleri gerektiğini bildirmek için geliştirilmiştir. Bu görevler listelenirken oyuncu, o ana kadar hangi görevleri gerçekleştirip gerçekleştirmediğini görebilmektedir. Şekil 4.25'te, Elektrik Dünyası'ndan bir görev penceresi görüntüsü yer almaktadır.



Şekil 4.25: Görev penceresi.

Görev penceresi iki ana bölümden oluşmaktadır. Sırasıyla görevleri şunlardır:

**Görev Grupları:** Oyuncunun yapması gereken görevler birden çok olduğundan, aynı tür görevler düzenli ve anlaşılır şekilde bu bölümde listelenmektedir. Buradaki görevlerden biri seçildiğinde, bu göreve ait alt görevler sağ panelde listelenmektedir.

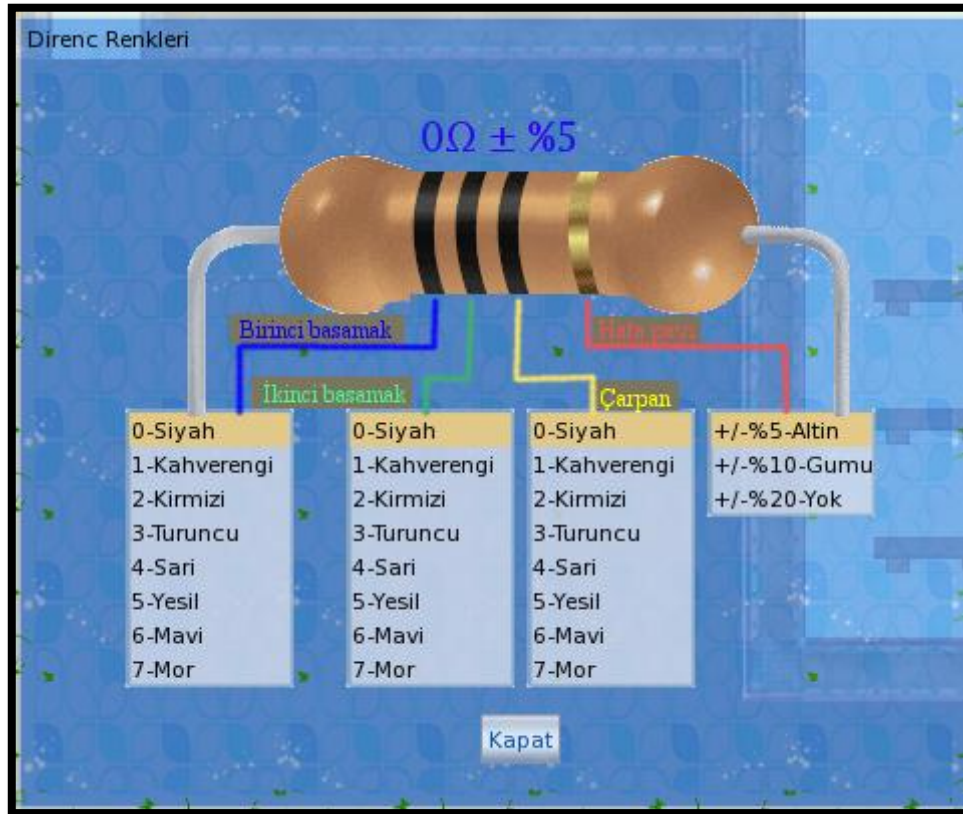
**Alt Görevler:** Oyuncunun tamamlaması gereken görevlerin listelendiği paneldir. Görevlerin üzerine fare ile gelindiğinde görev ile ilgili açıklamaların yer aldığı bilgilendirme kutucukları gösterilmektedir. Ayrıca görevin yanında gösterilen yeşil tik işareti o görevin oyuncu tarafından yerine getirildiğini, kırmızı çarpı işareti ise görevin henüz yerine getirilmediğini ifade etmektedir.

Görev pencereleri ile her dünyada oyuncudan yerine getirilmesi istenilen farklı görevler listelenir. Oyuncu görev penceresine, görev bilgisayarlarına tıklayarak

veya klavyeden backspace tuşuna basarak ulaşabilmektedir. Oyuncu bir görevi yerine getirdikten sonra listenin güncellenebilmesi için görevlerini görev bilgisayarlarından birisine gidip yüklemelidir. Bunun için de görev bilgisayarına tıklaması yeterlidir.

#### 4.2.2.5 Özel İçerikli Pencere

Özel içerikli pencereler, oyun içerisinde uygulayıcının istekleri doğrultusunda programlanabilen pencerelerdir. Örneğin, öğrencilerin direnç değerlerini hesaplayabilmeleri için hazırlanmış Direnç Renkleri Penceresi bunlardan biridir. Dirençlerin renk kodlarını göstermek üzere tasarlanmış özel içerikli pencere **Şekil 4.26**'da verilmiştir.



**Şekil 4.26:** Direnç renkleri penceresi

Bu tür pencereleri oluşturmak için ileri düzey programlama bilgisi ile kaynak kodda değişiklik yapılması gerekmektedir. Bu yüzden, ileriki sürümlerde özel içerikli pencerelerin kolayca oluşturulabilmesi için modüler bir yapıya dönüştürülmeleri düşünülebilir.

### 4.3 elektroGame Oyununun Geliştirilme Süreci

elektroGame'in geliştirilmesine karar verildikten sonra, pek çok MMORPG oyunu ve geliştirme yazılımı incelenmiştir. Özellikle açık kaynak kodlu yazılımlardan lisansı uygun olanlar, elektroGame'e temel olup olmayacakları açısından ele alınmıştır. Bu incelemeler ve değerlendirmeler sonucunda, istemci olarak **The Mana World (TMW)** sunucu olarak da **eAthena 1.0 text**'in kullanılmasına karar verilmiştir. Her iki yazılım da GNU lisanslı ve açık kaynaklı oldukları için kaynak kodlarında sorunsuzca değişiklik ve geliştirme yapılmıştır. Aynı şekilde elektroGame'in lisanslanması da GNU olarak yapılmıştır.

Çalışmanın bu bölümde elektroGame'i oluşturan istemcinin, sunucunun, web sitesinin ve yardımcı programların nasıl geliştirildiği teknik olarak ele alınmıştır. Ayrıca programın önemli sınıflarının ve sunucudaki önemli yapıların kodları belirtilen yerlerde ek olarak sunulmuştur. Deneysel oyun modelinin aşamalarına göre geliştirme sürecinden ve geliştirme basamaklarından Bölüm 3.3'te bahsedilmiştir.

#### 4.3.1 İstemcinin Geliştirilmesi

elektroGame'in temel mimarisi istemci-sunucu (client/server) yapısı üzerine kuruludur. Bu mimaride kullanıcı tarafından çalıştırılan ve merkezde bulunan yazılımla haberleşebilen yazılıma istemci denir. Sunucu ise tüm istemcilerden gelen talepleri karşılayan ve istemcilerin bilgilerini birbirlerine göndererek eş güdümü sağlayan yazılımdır.

elektroGame'in istemcisi C++ kullanılarak Code Blocks editöründe yazılmıştır. Derleyici olarak GNU GCC'nin minimal bir Windows uyarlaması olan minGW32 kullanılmıştır. Hem editörün hem de derleyicinin tercihinde, yazılımların GNU lisanslı özgür yazılımlar olmaları, kullanıcı dostu (user friendly) olmaları ve büyük bir kullanıcı kitlesine sahip olmaları etkili olmuştur.

Bölüm 4.1.1'de istemcinin yapısı, geliştirilmesi ve öğretim amaçlı bölümlerinin nasıl kodlandığı açıklanmıştır. Takip eden bölümlerde sunucunun, web sitesinin ve yardımcı programların geliştirilme evrelerinde bahsedilmiştir.

#### 4.3.1.1 İstemcide Kullanılan Kütüphaneler

elektroGame istemcisi geliştirilirken, programlama aşamasında birçok geliştirme kütüphanesinden (library) yararlanılmıştır. Geliştirme kütüphaneleri, programcılar tarafından geliştirilmiş, belli işleri yerine getirmeye yarayan ve ana programdan bağımsız olarak çalışan alt programlar olarak tanımlanabilir.

Geliştirme kütüphaneleri ile binlerce satır koddan oluşan programın daha sade bir kod yapısıyla daha hızlı ve sorunsuz bir şekilde çalışması mümkün olmaktadır. Ayrıca, geliştirme kütüphaneleri yazılımcılara çok büyük bir zaman tasarrufu sağlanmaktadır. Örneğin, **libcurl** kütüphanesi 35000 satır koddan oluşmaktadır. Bu çalışmada kullanılan kütüphanelerin tamamı daha önce birçok çalışmada kullanılmış, hızlı, güvenilir ve sorunsuz çalıştığı bilinen kütüphanelerdir.

Kullanılan kütüphaneler:

**SDL:** SDL, adını Simple Directmedia Layer kelimelerinin başharflerinden alan, ses kartı, klavye, fare, oyun çubuğu ve OpenGL aracılığıyla 3D grafik donanımına düşük seviyeli kontrol imkânı sağlayan 2D bir grafik motorudur (2D framebuffer). SDL bu programda da kullanılan birçok alt kütüphaneden oluşmaktadır (<http://www.libsdl.org/>).

**SDL mixer:** SDL ile kullanılan bir ses karıştırma kütüphanesidir. Programcının ek bir kod yazmadan çeşitli biçimlerdeki ses dosyalarını ve parçalarını oynatabilmesini sağlar. Oyun içerisinde yer alan efektler ve müzikler bu kütüphane yardımıyla çalınmaktadır ([http://www.libsdl.org/projects/SDL\\_mixer/](http://www.libsdl.org/projects/SDL_mixer/)).

**SDL image:** SDL ile kullanılan bir resim kütüphanesidir. Çeşitli biçimlerdeki resim dosyalarının ekranda görüntülenmesini sağlar. elektroGame'deki birçok grafik resim dosyalarından oluşmaktadır ve bunların ekranda görüntülenmeleri için bu kütüphane kullanılmaktadır ([http://www.libsdl.org/projects/SDL\\_image/](http://www.libsdl.org/projects/SDL_image/)).

**SDL net:** SDL ile kullanılan bir ağ iletişim kütüphanesidir. Ağ bağlantılarını ve ağ üzerinden veri transferini basitleştirir. elektroGame'deki istemci ile sunucu arasındaki iletişim bu kütüphane kullanılarak gerçekleştirilmektedir ([http://www.libsdl.org/projects/SDL\\_net/](http://www.libsdl.org/projects/SDL_net/)).

**SDL ttf:** elektroGame'in ilk aşamalarında yazı tipi dosyaları yerine tüm karakterlerin resimlerini içeren resim dosyaları kullanılmaktaydı. Ancak farklı yazı tiplerinin gerekliliği, tüm karakterlerin tek bir resim dosyasına aktarılmasının uzun zaman alması ve sonuçların da görsel açıdan yetersizliği sebebiyle, SDL ttf kütüphanesi kullanılmaya karar verilmiştir. SDL ttf, SDL ile kullanılan bir yazı tipi kütüphanesidir. ttf (true type font) uzantılı yazı tipi dosyalarının ekranda düzgün bir şekilde gösterilmesini sağlar. Bu kütüphane sayesinde elektroGame pencerelerinde çok çeşitli yazı tipleri kullanılabilir ( [http://www.libsdl.org/projects/SDL\\_ttf/](http://www.libsdl.org/projects/SDL_ttf/) ).

**Guichan:** Guichan oyunlar için özel olarak yazılmış, grafik tabanlı kullanıcı ara yüzüdür (GUI: **G**raphical **U**ser **I**nterface). Oyunda yer alan kullanıcı giriş penceresi, karakter oluşturma penceresi gibi pencereler Guichan kütüphanesi kullanılarak hazırlanmıştır. Guichan ufak ve sade olması ile birçok oyunda kullanılması sebebiyle tercih edilmiştir ( <http://guichan.sourceforge.net/wiki/index.php/> ).

**wsock32:** Windows'un ağ ve internet soketlerine erişmeye imkân tanıyan API'dir. elektroGame'in Windows sürümünde ağ bağlantılarına sorunsuzca erişmesini sağlar ( <http://support.microsoft.com/kb/151210> ).

**libxml2:** XML dokümanlarının ya da XML tipindeki verilerin çözümlenmesini sağlayan Gnome projesi kapsamında geliştirilmiş bir kütüphanedir. Oyunda robotların, bilgelerin ve daha birçok öğenin bilgileri XML biçimindeki dosyalarda saklanmaktadır. Bu bilgiler oyunda kullanılacağı zaman, libxml2 kütüphanesi aracılığıyla dosyadan ya da sunucudan gelen XML biçimindeki verilerden çözümlenerek değerlendirilir ve gerekli işlemler yapılır ( <http://xmlsoft.org/> ).

**OpenGL:** Open GL 2 ve 3 boyutlu grafik çizdirmek için kullanılan bir API'dir. OpenGL destekleyen donanımlarda yazılımcının doğrudan donanıma hükmetmesine izin verir. İşletim sisteminden bağımsız olduğu için yazılan kodun taşınabilir olmasına imkân tanır ( <http://www.opengl.org/> ).

**libpng12:** PNG (Portable Network Graphics) biçimindeki resim dosyalarını okumak ve o türde dosya oluşturmak için hazırlanmış bir kütüphanedir. elektroGame'de yer alan tüm resim dosyaları png biçimindedir ve bu dosyaların



okunup gösterilme işlemi png12 kütüphanesi aracılığıyla gerçekleştirilmektedir (<http://www.libpng.org/pub/png/libpng.html>).

**libcurl:** FTP, HTTP, POP3, SMTP, HTTPS, TELNET ve daha birçok protokolleri destekleyen, URL (Uniform Resource Locator, Düzenli Kaynak Konumlayıcı) sözdizimi ile veri transferi yapmaya yarayan bir kütüphanedir. elektroGame’de yapılan güncelleştirmeler bu kütüphane yardımıyla otomatik olarak sunucudan indirilir ve otomatik olarak güncelleme yapılır (<http://curl.haxx.se/libcurl/>).

**zlib:** Her platformda kullanılabilen, özgür, kayıpsız bir sıkıştırma ve açma algoritmasıdır (<http://zlib.net/>). elektroGame’de pek çok veri sıkıştırılarak saklanır. Örneğin, dünyaların saklandığı TMX dosyaları içindeki katman verileri, fazla yer kaplamamaları için XML biçimindeki dosya içinde sıkıştırılmış şekilde tutulur.

**PhysFS:** Oyunlarda kullanılmak için yazılmış, arşiv türündeki çeşitli dosyalara soyut erişim sağlamak için kullanılan bir kütüphanedir (<http://icculus.org/physfs/>). Programcı, fiziksel dosya sisteminde yazılabilir bir klasör belirler ve PhysFS aracılığıyla yapılan bütün işlemler sadece bu belirlenen alanda gerçekleştirilir. Bu sistem, oyuncuların bilgisayarlarının güvenliği açısından kullanılmaktadır.

**Libintl:** Yerel dil desteği sağlamak için geliştirilmiş bir kütüphanedir. Gettext’in bir parçasıdır (<http://gnuwin32.sourceforge.net/packages/libintl.htm>).

**GSL:** GNU Bilimsel Kitaplığı C ve C++ programcılar için sayısal bir kütüphanedir. Genel Kamu Lisansı altında özgür bir yazılımdır. Rasgele sayı üreteçleri, özel fonksiyonlar ve en küçük kareler gibi geniş yelpazede matematiksel fonksiyonlar sunar. Binden fazla işlevi içerir (<http://www.gnu.org/s/gsl/>).

#### 4.3.1.2 Sunu Penceresinin Geliştirilmesi

Bölüm 4.2.2.1’de görevi açıklanan sunu penceresi, elektroGame’in öğretime yönelik birimleri içinde en sade yapıya sahip olanıdır. Genellikle açıklama yapan bilgiler tarafından, oyunu öğretmek ya da belirli bir ders konusu hakkında bilgi

vermek için kullanılır. Oyunu öğretme amaçlı olanların pek çoğu, oyuncuyu temsil eden char belirli bir koordinata gelince otomatik olarak da açılabilir.

Çalışmanın bu bölümde istemci tarafında sunu penceresinin nasıl çalıştığıyla ilgili genel programlama algoritması açıklanacaktır.

Gönderilen kod, gelen mesajları değerlendiren sınıf olan **messagehandler** tarafından paketin bir bilgiye (NPC) ait olması durumunda, kontrolü üstlenen sınıf **npchandler**'a iletilir. **Npchandler** bu yapının bir sunu penceresine ait olduğunu anlayarak sunu penceresinin daha önceden oluşturulmuş bir örneğini (instance) çağırır. Gelen XML formatındaki veriyi `slideWindow::parseXML` fonksiyonu aracılığıyla sunu penceresinin gösterilmesinden ve kontrol edilmesinden sorumlu sınıf olan **slidewindow**'a aktarır. Bu durumda kullanılan XML kodu **Şekil 4.27**'de ve **Şekil 4.28**'de gösterilmiştir.

```
<?xml version="1.0" encoding="utf-8" ?>
<presentation>
  <presentationpro x="50" y="50" w="410" h="450" totalslides="7" />
  <slidebutton object="prev" visible="show" x="20" y="0" />
  <slidebutton object="combo" visible="show" x="60" y="0" />
  <slidebutton object="next" visible="show" x="220" y="0" />
  <slidebutton object="close" visible="show" x="360" y="0" />
  <image x="5" y="40" src="graphics/sunular/farabil/h_00.png" />
</presentation>
```

**Şekil 4.27:** Sunu penceresinin özelliklerini bildiren tanımlayıcı XML.

```
<?xml version="1.0" encoding="utf-8" ?>
<presentation>
  <slide slideid="1" name="Direnç"/>
  <image x="5" y="40" src="graphics/sunular/icerik.png" />
  <!-- Diğer içerik tagları -->
</presentation>
```

**Şekil 4.28:** Slayt bilgisini ve içeriğini tanımlayan XML.

Slidewindow'a gönderilen XML'de kullanılabilecek tag'ların açıklamaları **Tablo 4.2**'te verilmiştir.



**Tablo 4.2:** Sunu penceresini oluşturmak kullanılan XML yapısının tag'ları.

<b>Presentation</b>	XML dosyasının slidewindow penceresine ait olduğunu belirten kök tag'dır.
<b>Presentationpro</b>	Sunuya ait pencerelerin boyutlarını ve kaç slayttan oluştuklarını bildirir.
<b>Slidebutton</b>	İleri, geri, kapat ve açılan kutularının slayt üzerindeki koordinatlarını ve görünüp görünmeyeceklerini bildirir.
<b>Slide</b>	Sunuya ait kaçınıcı slayt olduğunu ve slaydın adını bildirir.
<b>Image</b>	Sununun arka planını ve sunu üzerinde yer alacak resimlerin istemci üzerindeki yolunu (path), ismini bildirir.
<b>Textbox</b>	Sunu içinde bir yazım alanı açar. Yazım alanının koordinatlarını, genişlik, yükseklik ve renk bilgilerini tanımlar. Textbox içine addrow tag'ı ile yazı eklenebilir. textbox'lar ##X gördükleri metinleri renkli halde yazabilirler.
<b>Label</b>	Sunu üzerine metin eklemek için kullanılır. Label tag'ları farklı ihtiyaçlar için kullanılabilirdiğinden çok fazla özelliği vardır.
<b>sendeffect</b>	Particalmanager'a efekt iletmekte kullanılır. Sunu penceresinde, genellikle sunu tamamlandığında, char'ın etrafındaki yıldız efektleri için kullanılır.
<b>Anim</b>	Sprite tekniği ile yapılmış animasyonları gösterir. Animasyonun koordinatlarını, animasyon oluşturmak için kullanılacak resmin yolunu ve ismini, her bir karenin genişlik ve yüksekliğini, animasyonun oynatma hızını bildirir.

Şekil 4.27, Şekil 4.28 ve Tablo 4.2'te yapısı açıklanan XML veriyi ayrıştırarak (parse) görselleştiren slidewindow sınıfı Window ve gcn::ActionListener sınıflarını miras alarak türetilmiştir. Slayttaki nesnelere bellekte yer kaplamaması için her nesne sadece gösterileceği zaman oluşturulmuş, görüntülenmesi sona erince de yok edilmiştir. Bu şekilde dinamik bellek yönetimini gerçekleştirebilmek için **elektroWidget** sınıfında nesnelere önce yapı (struct) olarak tanımlanmış, sonra da bu yapıyı saklayacak vektör tipi tanımlanmıştır. Şekil 4.29'da yapının ve vektörün kodu verilmiştir.

```
struct SmImage{
    Image *img;
    int x;
    int y;
    bool visible;
};

typedef std::vector< SmImage> TmvImage;
typedef std::vector< SmImage>::iterator TmiImage;
```

**Şekil 4.29:** Image nesnesinin elektroWidget sınıfında tanımlanan yapısı ve vektör tipi.

XML verilerin ayrıştırılması ve slayt üzerinde gösterilecek nesnelere oluşturulmasından sonra GUI'nin parçası olan nesnelere pencereye eklenirken GUI'ye eklenemeyen resim ve animasyonlar Window sınıfından alınarak ezilen Draw fonksiyonu ile ekrana basılır. Draw fonksiyonu oyunun saniyede gösterdiği kare sayısı (Frame Per Second - FPS) kadar çağırılarak kararlı bir görüntü oluşması sağlanır.

Pencerenin çizim işlemleri belirli aralıklarla tekrar edilirken `gc::ActionControler` sınıfı `SlideWindow::action(const gc::ActionEvent &event)` fonksiyonuna gerçekleştirilen olayları göndererek oyuncunun ileri, geri gibi düğmelere basıp basmadığını bildirir. Slaytlar arasında geçiş yapmak gibi `slidewindow` sınıfı tarafından kontrol edilmesi gereken bir olay olduğunda `slideStateControl()` fonksiyonu çağırılarak gerekli işlemler yapılır.

Sunu penceresinden çıkılırken oluşturulan her nesne, bellekten temizlenerek bir sonraki sunu için hafıza hazır hale getirilir.

#### 4.3.1.3 Test Penceresinin Geliştirilmesi

`elektroGame`'de oyuncunun, yükselebilmesi için sık sık bilgelerin sorduğu sorulara yanıt vermesi gerekmektedir. İstemcide soru sorma ve yanıtların değerlendirilmesinden sorumlu olan test penceresi, `testWindow` sınıfı tarafından oluşturulur. `testWindow` sınıfı, istemci ve sunucu arasında sürekli ve güvenli veri taşıyabilmesi, aynı zamanda da oyunun oynanış tarzını bozmaması için dikkatlice tasarlanmıştır.

Test penceresi `elektroGame`'in genel istemci-sunucu mimarisindeki XML veri yapısına uygun olarak, test yapan bir bilgeye tıklatıldığında sunucu tarafından bir test XML'inin gönderilmesiyle başlar. **Messagehandler**'ın gelen XML veriyi **npchandler**'a yönlendirmesinden sonra, gelen XML veri, **npchandler** tarafından ayrıştırılmak üzere, daha önceden oluşturulmuş olan **testWindow** sınıfının bir örneği çağırılarak `testWindow::parse()` fonksiyonuna aktarılır.

Test yapacak bir bilgiye tıkladığında, sunucu tarafından ilk olarak testin özelliklerini tanımlayan <mesaj> tag'ı gönderilir. **Şekil 4.30**'da verilen XML yapı **Şekil 4.15**'te görülen mesaj penceresi şeklinde görüntülenir.

```
<?xml version="1.0" encoding="utf-8" ?>
<test>
  <mesaj>
    <addrow text="##9Tanıtım mesajı" />
    <addrow test="##3 ve açıklamalar. " />
    <addrow text= "#autowrap#" />
  </mesaj>
</test>
```

**Şekil 4.30:** Test penceresinde açılış mesajını oluşturan XML.

addrow tag'ının içinde geçen #autowrap# ifadesi mesaj penceresine sığmayan satırların otomatik olarak bölünmesi için BrowserBox::autoWrap(ScrollArea \*textArea) fonksiyonunu çağırmaya yarayan anahtar bir ifadedir.

Testin açılış mesajına oyuncu tarafından "Tamam" düğmesine basılarak yanıt verilmesinden sonra, sunucu tarafından testin ve test penceresinin özelliklerini tanımlayan ve **Şekil 4.31**'de gösterilen veri gönderilir.

```
<?xml version="1.0" encoding="utf-8" ?>
<test>
  <starter>
    <window width= "480"
      height= "600"
      left= "0"
      top= "0" />
    <testpro totalquestion="10"
      totaltime="600"
      award="10"
      punish="0"
      successlimit="7" />
  </starter>
</test>
```

**Şekil 4.31:** Test penceresinin ve testin özelliklerini tanımlayan <starter> tag'ı.

<starter> tag'ı, test penceresinin genişlik, yükseklik ve koordinatlarını tanımlamak için kullanılır. Farklı boyutlarda pencereye ihtiyaç duyan sorular için

esneklik sağlama için <starter>'in dışında da kullanılabilir. Fakat böyle bir kullanım, test penceresinde titremeye neden olup oyuncuları rahatsız edeceği için gerekmedikçe tercih edilmemiştir.

<testpro> tag'ı ise testteki soru sayısı, toplam süre, ödül, ceza ve başarı limitini tanımlar. Tasarım aşamasında ödül ve cezanın sadece tecrübe puanını değiştirmesi öngörüldüğü için integer tipte bir sayı olarak tanımlanmıştır. Ancak uygulama aşamasında çeşitli nesnelere ve farklı puanların da ödül olarak verilebilmesi gerekliliği ortaya çıkmıştır. elektroGame'in sonraki sürümünde, iki değer de uygun şekilde değiştirilebilir.

```
<test>
<question qn="1" type="radio" />
  <image x="5" y="10"
    src="graphics/questions/test30/soru1.png" />
  <label x="20" y="30"
    text="Bu soru metni"
    font="speechFont"
    width="120"
    height="20"
    fcolorr="100"
    fcolorg="150"
    fcolorb="255"
    bcolorr="0"
    bcolorg="0"
    bcolorb="0"
    bordersize="1" />
  <radio x="75" y="380" text="A seçeneği" value="1" />
  <radio x="175" y="380" text="B seçeneği" value="2" />
  <radio x="275" y="380" text="C seçeneği" value="3" />
  <radio x="375" y="380" text="D seçeneği" value="4" />
</test>
```

**Şekil 4.32:** Test penceresinde soru oluşturulmasını sağlayan <question> örneği.

Oyuncu, testi bitirip “Değerlendir” düğmesine bastığında, o ana kadar verdiği yanıtlar sonucu tarafında değerlendirilerek test penceresinde görselleştirilmek üzere rapor hazırlanır. Oyun geliştirilirken rapor formatı **Şekil 4.34**'te verildiği gibi, oyuncunun yanıtıyla doğru yanıtı karşılaştıracak bilgileri içerecek şekilde oluşturulmuştur. Ancak uygulama sırasında öğrencilerin barajı aşmaya çalışmak yerine, doğrudan rapor ekranına geçerek yanıtları elde etmeye çalıştıkları

belirlenince, barajı aşamayan oyuncular için Şekil 4.33'te verilen format geliştirilmiştir.

```
<?xml version="1.0" encoding="utf-8" ?>
<test>
  <report numcorrectanswer="2"
        numwronganswer="1"
        numblankanswer="7" />
  <reportmesaj mesaj="hadi hadi canlan biraz!" />
</report>
</test>
```

Şekil 4.33: Testin sonunda barajı aşamayan oyuncuya sunucu tarafından gönderilen rapor örneği.

```
<?xml version="1.0" encoding="utf-8" ?>
<test>
  <report numcorrectanswer="8"
        numwronganswer="2"
        numblankanswer="0" />
  <reportmesaj mesaj="iyi gidiyor" />
  <question qn="1" correctanswer="2" studentanswer="3" />
  <question qn="2" correctanswer="3" studentanswer="2" />
  <question qn="3" correctanswer="3" studentanswer="3" />
  <question qn="4" correctanswer="3" studentanswer="3" />
  <question qn="5" correctanswer="1" studentanswer="1" />
  <question qn="6" correctanswer="4" studentanswer="4" />
  <question qn="7" correctanswer="2" studentanswer="2" />
  <question qn="8" correctanswer="4" studentanswer="4" />
  <question qn="9" correctanswer="4" studentanswer="4" />
  <question qn="10" correctanswer="4" studentanswer="4" />
</report>
</test>
```

Şekil 4.34: Testin sonunda barajı aşan oyuncuya sunucu tarafından gönderilen rapor örneği.

Test penceresini oluşturmakta kullanılan XML'de kullanılan tag'lar Tablo 4.3'te açıklanmıştır.

**Tablo 4.3:** Test XML’de kullanılan tag’lar.

Test	XML’in test penceresine ait olduğunu belirten kök tag’dır.
Starter	Test başlatılmadan önce test penceresi ve testin özellikleri hakkında bilgi verilebilmesi için kullanılan tag’dır.
Window	Test penceresinin genişlik, yükseklik ve koordinatlarını tanımlar.
Testpro	Oluşturulacak olan teste ait aşağıdaki özellikleri tanımlar. <ul style="list-style-type: none"><li>• <b>totalquestion:</b> Toplam soru sayısını bildirir.</li><li>• <b>totaltime:</b> Test için toplam süreyi bildirir.</li><li>• <b>award:</b> Ödül için verilecek puanı bildirir.</li><li>• <b>punish:</b> Ceza puanını bildirir.</li><li>• <b>successlimit:</b> Testten başarılı sayılmak için gereken soru sayısı</li></ul>
Mesaj	Test başlamadan önce gösterilecek olan mesaj için kullanılır.
Question	Testteki her bir sorunun başlangıcını ve bitişini belirten tag’dır. <ul style="list-style-type: none"><li>• <b>questionnumber:</b> Sorunun testte kaçınıcı soru olduğunu bildirir.</li><li>• <b>type:</b> Sorunun tek seçenekli mi çok seçenekli mi olduğunu bildirir.</li><li>• <b>selected:</b> Oyuncu bu soruyu daha önceden yanıtladıysa, o seçeneğin seçili olması için yanıtı gönderir.</li></ul>
Effect	Test ekranının kapanmasından sonra, ödül ya da ceza sonucu çeşitli parçacık efektleri oluşması için kullanılır.
Radio	Soruya bir seçenek ekler.
Check	Soruya onay kutulu bir seçenek ekler.
Label	Soruya bir metin kutusu ekler.
Image	Soruya bir resim ekler.
Text	Soruya bir metin alanı ekler.
Anim	Soruya sprite animasyonu ekler.
Report	Testin bitiminde raporun başladığını belirten tag’dır. Özellikleri şu şekildedir: <ul style="list-style-type: none"><li>• <b>numcorrectanswer:</b> Doğru cevap sayısı.</li><li>• <b>numwronganswer:</b> Yanlış cevap sayısı.</li><li>• <b>numblankanswer:</b> Boş cevap sayısı.</li></ul>
Question	Report tag’ının içinde tanımlı olan question tagı, test tag’ında tanımlı olandan farklı olarak aşağıdaki özellikleri tanımlar: <ul style="list-style-type: none"><li>• <b>qn:</b> Soru numarası.</li><li>• <b>correctanswer:</b> Sorunun doğru yanıtı.</li><li>• <b>studentanswer:</b> Oyuncunun verdiği yanıt.</li></ul>
Reportmessage	Raporda oyuncuya verilen bir cümlelik mesajı bildirir.

Özellikleri açıklanan XML formatındaki veriyi görselleştirmek için, TestDialog adında Window ve gen::ActionListener sınıflarını public olarak miras alarak türetilmiş bir sınıf yazılmıştır. Bu sınıf **gui** nesnesinde oluşturulmakta ve **npchandler** tarafından <test> tagı algılanması halinde çağrılarak görülebilir hale getirilmektedir.

TestDialog sınıfı **typedef.h** başlık dosyasında tanımlanan ve **elektroWidget** sınıfı tarafından oluşturulan nesnelere kendi konteynerlerinde tutarak kullanır. Bu durum **Şekil 4.29**'a benzer yapı ve vektörlerle sağlanır. **Tablo 4.4**'te TestDialog sınıfında kullanılan ve bir soru oluşması için gerekli vektörlerin tip tanımlamaları verilmiştir.

**Tablo 4.4:** Testteki her bir soru için kullanılan vektörler ve görevleri.

TmvLabel mvLabel;	typedef std::vector< gcn::Label * > TmvLabel;
TmvImage mvImage;	struct SmImage{ Image *img; int x; int y; bool visible;}; typedef std::vector<SmImage> TmvImage;
TmvAnim mvAnim;	struct SmAnim{ SimpleAnimation *anim; int x; int y; int v; bool visible;}; typedef std::vector< SmAnim> TmvAnim;
TmvRadio mvRadio;	struct SmRadio{ gcn::RadioButton *radio; int value;}; typedef std::vector<SmRadio> TmvRadio;
TmvCheck mvCheck;	struct SmCheck{ CheckBox *check; std::string value; }; typedef std::vector< SmCheck> TmvCheck;
TmvButton mvButton;	typedef std::vector< Button* > TmvButton;
TmvScrollArea mvScrollArea;	typedef std::vector< Component* > TmvComponent;
TmvBrowserBox mvTextBox;	struct SmTextBox{ BrowserBox *browserbox; ScrollArea *scrollarea;}; typedef std::vector <BrowserBox*> TmvBrowserBox;

TestDialog sınıfı, yapılmakta olan testin hangi aşamada olduğunu takip etmek için **Şekil 4.35**'te gösterilen durumları kullanır. Her bir durum **TestDialog::start()** fonksiyonu tarafından yönetilir. Test penceresinin görsel şekli **start()** fonksiyonunun ilgili fonksiyonları çağırmasıyla duruma uygun hale getirilir.

```
enum TestState {  
    MESSAGE_STATE,  
    ONEQ_START_STATE,  
    MANYQ_START_SATATE,  
    MANYQ_CONTINUE_STATE,  
    ONE_RADIO_STATE,  
    ONE_CEHCK_STATE,  
    TEST_FINISHED_STATE,  
    SHOW_RESULT_STATE  
};
```

**Şekil 4.35:** Test aşamalarını kontrol etmekte kullanılan durumlar.

Window sınıfından miras alınarak geliştirilen **logic()** fonksiyonu, TestDialog sınıfının testin süresini yönetmesinde kullanılır. Bu fonksiyon, oyundaki sunucuyla haberleşme, robotların ve diğer oyuncuların hareketleri gibi diğer tüm olaylar devam ederken testin bağımsız olarak devam edebilmesini sağlar. Ayrıca, test süresinin belirlenen süreye ulaşması halinde “Değerlendir” düğmesine basılma olayı üreterek TestDialog::action(const gcnc::ActionEvent &event) çağırılmasına neden olur.

Oyuncunun “Değerlendir” düğmesine basmasıyla ya da sürenin dolmasıyla **Şekil 4.35**'te verilen durumlardan **TEST\_FINISH\_STATE** durumuna geçilir. Bu durumda istemci sunucuya bilgeyle olan konuşmanın sonlandırıldığı mesajını gönderir. Bu mesajla birlikte oyuncunun verdiği yanıtlar, sunucu tarafından değerlendirmeye alınır. Böylece, oyuncuların istemci programın yapısını değiştirip değerlendirme işlemine müdahale etmesi engellenerek güvenlik artırılmış olur. Yine, değerlendirme sonucunda rapor ekranı görüntülenip görüntülenmeyeceğine, sunucu tarafında karar verilerek istemciye bildirilir.

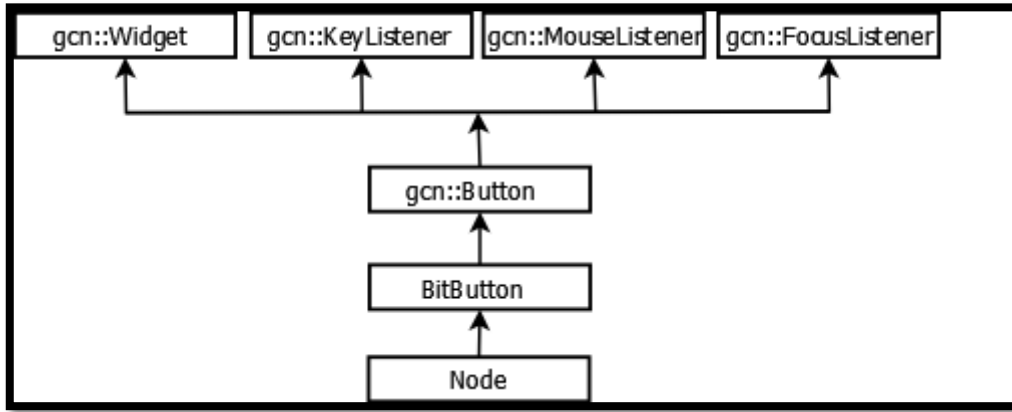
#### **4.3.1.4 Elektrik Devre Elemanlarının Geliştirilmesi**

Devre penceresinde kullanılan basit elektrik devresi elemanları, nesne yönelimli programlama ilkelerine uygun olarak ortak atadan türetilerek programlanmıştır. Bu sayede hem yeni bileşenlerin eklenmesiyle daha önce yazılan kodda değişiklik ihtiyacı en aza indirilmiş hem de verimlilik artırılmıştır.



Devre elemanlarının özelliklerini miras aldığı sınıf olan **Component** sınıfı yazılmadan önce, elemanlara kablo bağlanabilmesini sağlayan ve bağlantı noktalarını temsil eden **Node** sınıfı yazılmıştır.

**Şekil 4.36**'da **Node** sınıfının soy ağacı gösterilmektedir. Burada en üstte bulunan temel sınıflar ve bunlardan türeyen `gcn::Button` sınıfı, Guichan kütüphanesinden alınarak kullanılmaktadır. `BitButton` ve **Node** sınıfları ise araştırmacı tarafından geliştirilmiştir.



**Şekil 4.36:** Node sınıfının soy ağacı.

Bağlantı noktaları hazır olduktan sonra `gcn::Window` ve `gcn::ActionListener` sınıflarını miras alan **Component** sınıfının yazılmasına başlanmıştır. Component sınıfının fonksiyonlarının, kendisini miras alan sınıflar tarafından değiştirilip geliştirilebilmesi (ezilebilmesi) için tanımlamaları virtual yapılmıştır.

**Component** sınıfında bağlantı noktalarını tutmak amacıyla iki adet **Node** sınıfından nesne tanımlanmıştır. Bu durum mevcut devre elemanları için yeterli olduğundan, **Node** nesnelere tutan bir vektör tanımlanmamıştır. Sonraki sürümlerde ikiden fazla bağlantı noktası içerebilecek devre elemanları için (transistörler gibi) **Node** nesnelere uygun bir konteynerde saklanması düşünülebilir.

Devre penceresine eklenen bir devre elemanının alabileceği durumlar **Şekil 4.37**'de gösterildiği gibi **enum** tipinde tanımlanmıştır. Böylece **Component** sınıfından türeyen (child) diğer sınıflar için ortak bir tanım listesi oluşturulmuştur.

```
/**
 * Componentin bulunabileceği durumlar
 */
enum Status {
    PASIVE = 0, //Kapalı:Lambalar için sönmük, anahtar için açık
    ACTIVE, //Çalışıyor:Lambalar için yanıyor, anahtar için kapalı
    BURNED, //Bozuk
    PLUS, //Parlak
    PLUS2 //Daha Parlak
};
```

Şekil 4.37: Bir Component nesnesinin alabileceği durumlar.

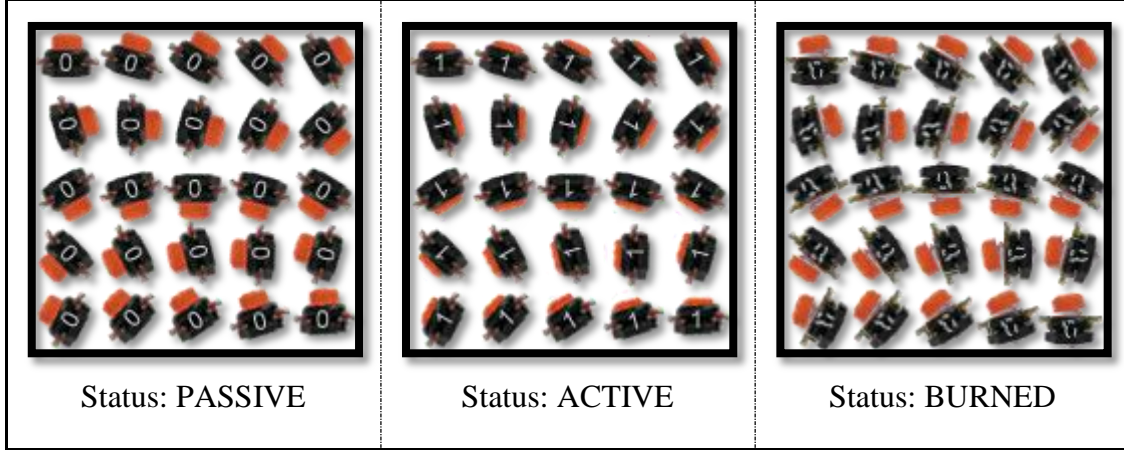
Devre penceresinde kullanılmakta olan devre elemanları yine küme (enum) tipi olarak tanımlanmıştır. Böylece **Component** sınıfından türeyen sınıfların birbirlerini tanıması ve gerektiğinde etkileşmeleri mümkün olmuştur. Şekil 4.38’de devre elemanlarının listesini tutan **Type** tipi verilmiştir.

```
enum Type {
    UNKNOWN,
    WIRE,
    RESISTANCE,
    BATTERY,
    DIODE,
    LAMP,
    SWITCH,
    MOTOR,
    AMPERMETRE};
```

Şekil 4.38: Component sınıfından türeyen devre elemanlarının listesi.

Tüm **Component** sınıfı, devre elemanlarını devre penceresine çizmek için elektroGame’in standart sprite sistemini kullanmaktadır. Her bir devre elemanı şekil 3.37’de verilen Status’u için 15 derecelik açılarla hazırlanmış Sprite Set’e sahiptir. Her **Component** kendi durumuna ve açısına uygun olan resmi seçerek devre penceresine çizilmesini sağlar. Tablo 4.5’de anahtarın (switch sınıfı) kullandığı sprite set serisi örnek olarak verilmiştir.

**Tablo 4.5:** Switch sınıfının sprite setleri.



Eğer sunucudan gönderilen XML’de devre elemanının dönmesine izin verildiyse, oyuncu ALT tuşuna basılı tutarken fare imlecini hareket ettirdiğinde, devre elemanı, bahsedilen sistem kullanılarak 15 derecelik açılarla döndürülebilmektedir.

Devre penceresi, ilmek akımları yöntemiyle devre elemanlarının üzerinden geçen akımı hesaplayabilmek için **Node**’ların birbirleriyle olan bağlantılarını takip etmektedir. Her **Component** kendi üzerinde bulunan iki **Node**’u bağlantılı göstererek devrenin çevrim oluşturmasını sağlamaktadır. Bir **Component**’in Status değeri “burned” ya da “passive” olarak değiştiğinde, **Node**’ların arasındaki bağlantı koparak çevrimin tamamlanmasına engel olmaktadır.

Oyuncu bir devre elemanını seçtiğinde, o elemanın seçili olduğunu göstermek üzere etrafında dönen bir halka şekli çıkmaktadır. Bu halka `Component::draw(gcn::Graphics *graphics)` fonksiyonda **Şekil 4.39**’da verilen kodda gösterildiği gibi, 10ms’de bir tazelenerek çizilir. Seçim halkasından her devre elemanı kendisi sorumlu olduğu için birden fazla elemanı seçmek için devre penceresinde yeni bir düzenleme yapmaya gerek kalmamaktadır.

```

void Component::draw(gcn::Graphics *graphics)
{
    // graphics->setColor(gcn::Color(0xff0000));
    // graphics->drawRectangle(gcn::Rectangle(0,0,getWidth(),getHeight()));
    Graphics *g = static_cast<Graphics*>(graphics);
    ResourceManager *resman = ResourceManager::getInstance();
    //Seçiliyse etrafına seçim halkasını ekle
    if (getSelected())
    {
        mSelectedAnime->update(10); //Halkayı 10ms'de bir tazele
        Image* targetCursor=mSelectedAnime->getCurrentImage();
        g->drawImage(targetCursor, -2, 2);
    }
    //Diğer pecere elemanlarını çiz -->Window
    drawChildren(graphics);
}

```

**Şekil 4.39:** Component sınıfının çizilmesinden sorumlu draw fonksiyonu.

**Component** sınıfı fare olaylarını CircuitWindow sınıfından önce ele alarak oyuncunun devre elemanları üzerinde değişiklik yapabilmesine olanak tanımaktadır. Örneğin, Switch sınıfı çift tıklama olayını algıladığında anahtarın açık kapalı konumunu değiştirmekte daha sonra yönetmesi için CircuitWindow'a iletmektedir.

**Component**, devre penceresinde oyuncunun sağ tıklamasıyla silinir. İlmek akımları hesabının bozulmaması ve sistemin kararsız hale gelmemesi için, silme işleminin devre penceresine bildirilmesi gerekmektedir. Bunu yapabilmek için, silinmek istenen component, kendisinin mDead özelliğini true olarak değiştirir. Bölüm 4.3.1.5'teki devre penceresinde gösterildiği şekilde, CircuitWindow::logic() fonksiyonu mDead özelliği true olan devre elemanları ve **Node**'lar silinir. Bu işlemi yapan kod bloğu bir sonraki bölümde verilmiştir.

#### 4.3.1.5 Devre Penceresinin Geliştirilmesi

Bu çalışma basit elektrik devreleri konusuna odaklandığı için devre penceresi, elektroGame'in en önemli pencerelerinden birisi olarak öne çıkmaktadır. Bu pencere, bir MMORPG'nin oynanış özelliklerini (game play) bozmadan bir elektrik devresi simülasyonu yapmak üzere geliştirilmiştir. İlgili bilgilerden birine oyuncunun tıklaması ile test ve sunu pencerelerinde anlatıldığı gibi **npchandler** aracılığı ile devre penceresi açılır ve sunucudan gelen XML verisine uygun olarak simülasyon başlatılır.

Sunucuda gelen devre XML verileri:

- Açıklama,
- Devre,
- Rapor

türlerinden birisi olabilir. Şekil 4.40'ta sunucudan gönderilen bir *açıklama* XML verisi gösterilmektedir. Bu yapı iki temel tag'dan oluşur. <window> tag'ı daha önce de belirtildiği gibi pencerenin özelliklerin tanımlar. <head\_message> tagı açıklamanın yapılacağı metin kutusunun açılmasını sağlar. #br# metin kutusu içinde satır başı yapmak için kullanılır.

```
<?xml version="1.0" encoding="utf-8" ?>
<circuit>
  <window width= "600"
height= "400" left= "20" top= "20" />
  <head_mesaj>
    <addrow text="    ##3 1. Sorudasın " />
    <addrow text= " #br# " />
    <addrow text="  ##2 Açıklama. " />
    <addrow text= "#autowrap#" />
  </head_mesaj>
</circuit>
```

Şekil 4.40: Devre penceresinde *açıklama* mesajı tanımlayan XML veri.

Devre tanımlamak için kullanılan XML yapısında, test ve sunu penceresinin oluşturulmasında kullanılan “label”, “image” gibi tag’ların tamamı kullanılabilir. Böylece devrenin açıklanmasında ve soru oluşturulmasında tasarımcı serbestçe dilediği görüntüyü elde edebilmektedir. Bu standart tag'lara ilave olarak **Tablo 4.6**'de verilen tag'lar ve özellikleri kullanılmaktadır. Bu tag'lar devre elemanlarının yerleşimini ve aralarındaki kablo bağlantısının yapılmasını sağlamaktadır.

**Tablo 4.6:** Devre penceresinde basit elektrik devresi oluşturabilmek için sunucu tarafından gönderilen XML dosyasının tag'ları ve bu tag'ların özellikleri.

Connect	İd	Node'lar arasında yapılacak bağlantının eşsiz bir id'ye sahip olması için verilen sayıdır.
	first	Bağlantı yapılacak olan Node'lardan birincisinin id'sini tanımlar.
	second	Bağlantı yapılacak olan Node'lardan ikincisinin id'sini tanımlar.
	active	Bağlantının üzerinden akım geçip geçemeyeceğini tanımlar.

**Tablo 4.6'nin devamı.**

Component	id	Yerleştirilen her componenti konteynerde tutmaya yarayan eşsiz bir sayıdır. Component'in sahip olduğu Node'lara da id verildiği için sadece tek sayılı değerler kullanılabilir.
	item_id	items.xml'de tanımlanan component id'si. Devre elemanları 1100-1140 arasındadır.
	x,y	Component'in pencere üzerindeki koordinatları.
	angel	Component'in açısını belirten 0-23 arası bir sayıdır. Gerçek açı değeri 15'le çarpılarak bulunur.
	status	<b>Şekil 4.37'</b> de verilen listeye uygun olarak durumu tanımlar.
	selectable	Component'in "seçilebilir" olduğunu gösterir. Bazı elemanlar seçilemez yapılarak oyuncunun hakkında bilgi edinmesi engellenmek istenebilir.
	movable	Component'in hareket ettirilebilmesini sağlar. Sadece seçilebilen nesnelere hareket ettirmek mümkündür.
	deletable	Oyuncunun sağ tıklayarak Component'i devre penceresinden yok etmesine izin verir. Silinen Component'in geri getirilmesi mümkün olmadığı için silinmesiyle devrenin yapısını bozacak elemanların silinmesine izin verilmez.
	fromlink1	Component'in sahip olduğu birinci Node'dan diğer bağlantı noktalarına oyuncu tarafından kablo takılabilesine izin verir.
	tolink1	Oyuncu tarafından Component'in sahip olduğu birinci Node'a diğer Node'lardan kablo bağlanabilmesine izin verir.
	fromlink2	Component'in sahip olduğu ikinci Node'dan diğer bağlantı noktalarına oyuncu tarafından kablo takılabilesine izin verir.
tolink2	Oyuncu tarafından Component'in sahip olduğu ikinci Node'a diğer Node'lardan kablo bağlanabilmesine izin verir.	
Node	id	Devre penceresi üzerine kabloların istenen şekilde görülebilmesi için eklenen bağımsız Node'ları eklemek için kullanılır.
	x,y	Node'un devre penceresindeki konumunu tanımlar.
	status	Node'un bağlantıya uygun olup olmadığını bildirir.
	selectable	Node'un seçilebilir olmasını sağlar.
	deletable	Oyuncu sağ tıkladığında Node'un pencereden kaldırılabilmesine izin verir. Seçilebilen ama silinemeyen Node'lara sağ tıkladığı zaman Node'un diğer Node'larla olan bağlantıları temizlenir.
	movable	Node'un oyuncu tarafından hareket ettirilebilmesine izin verir.

Tablo 4.6'nın devamı.

Circuitcondition	type	Verilen görevleri başarmak için oyuncunun gerçekleştirmesi gereken koşullardır. Dört tür koşul vardır. <ul style="list-style-type: none"><li>• <b>lampturnon:</b> Bir lambanın yanması gerektiğini tanımlar.</li><li>• <b>nodecurrent:</b> Herhangi bir bağlantı noktasından geçmesi gereken akım miktarını tanımlar.</li><li>• <b>locate:</b> Bir devre elemanını belirli bir koordinatta olması gerektiğini belirtir.</li><li>• <b>evulationswitch:</b> Bir anahtarın açık ya da kapalı olması gerektiğini belirtir.</li></ul>
	componentid	Belirtilen şartı hangi Component'in sağlayacağını tanımlar.
	current	nodecurrent tipinde geçecek akımın miktarını tanımlar.
	status	Anahtar ve lambalar için şartın açık mı kapalı mı olduğunu tanımlar.
	x,y,w,h	locate tipindeki şart için koordinat ve genişlik, yükseklik değerlerini tanımlar.

```
<?xml version="1.0" encoding="utf-8" ?>
<circuit>
<window width= "600" height= "400" left= "20" top= "20" />
<label x="10" y="10" text="Yeni hedefin 10ohm elde etmek."
font="txt_1" width="120" height="20" fcolorr="100"
fcolorg="150" fcolorb="255" />
<component id="1" item_id="1100" x="45" y="232"
angel="0" status="0" selectable="1" movable="1"
deletable="1" fromlink1="1" tolink1="1"
fromlink2="1" tolink2="1" />
<component id="9" item_id="1134" x="183" y="190"
angel="0" status="0" selectable="0" movable="0"
deletable="0" fromlink1="0" tolink1="0"
fromlink2="0" tolink2="0" />
<node id="101" x="41" y="93" status="1" selectable="0"
movable="0" deletable="0" fromlink="0" tolink="0" />
<node id="102" x="351" y="93" status="1" selectable="1"
movable="1" deletable="1" fromlink="1" tolink="1" />
<node id="104" x="41" y="143" status="1" selectable="0"
movable="0" deletable="0" fromlink="0" tolink="0" />
<node id="105" x="41" y="206" status="1" selectable="0"
movable="0" deletable="0" fromlink="0" tolink="0" />
<connect id="50" first="105" second="9" active="1" />
<connect id="51" first="10" second="106" active="1" />
<connect id="52" first="106" second="107" active="1" />
<connect id="53" first="107" second="102" active="1" />
<circuitcondition type="nodecurrent"
componentid="101" current="0.4" />
</circuit>
```

Şekil 4.41: Devre oluşturmak üzere gönderilen XML kodu.

Son olarak devre penceresinde “Değerlendir” düğmesine basıldığında <foot\_message> tag’ı gönderilir. Bu tag, sunucu tarafından şartların ne ölçüde sağlandığını değerlendiren ve oyuncuya geridönüt sağlayan bir mesaj iletilmesini sağlar. Ayrıca sonuca uygun parçacık ve ses efektleri, gösterilen mesaja eşlik edebilir. <head\_message> tag’ından farklı olarak #close# devre penceresini “Tamam” düğmesine basmadan otomatik olarak kapatmak için kullanılır.

Tag’ları ve özellikleri açıklanan XML kodu **CircuitWindow::circuitFromXML(std::string mDoc)** fonksiyonu tarafından ayrıştırılmaktadır. Bu fonksiyon gelen veriye uygun olarak ihtiyaç duyduğu nesnelere **elektroWidget** sınıfına ürettirerek uygun konteynerlerde saklar. Ayrıştırma işleminden sonra **Şekil 4.42**’de verilen durumlardan hangisinde olduğunu kontrol ederek devre penceresinin uygun görünümü alması için **stateCheck()** fonksiyonu çağrılır.

```
//Aşama türleri
enum CIRCSTATE{
    HEAD_MESSAGE_STATE = 0,
    FOOT_MESSAGE_STATE,
    CIRCUIT_STATE,
    TEST_STATE,
    WAIT_STATE,
    ENDSTATE
};
```

**Şekil 4.42:** Devre penceresinin alabileceği durumlar.

Eğer pencere üzerinde simülasyonun başlaması anlamına gelen CIRCUIT\_STATE durumuna gelinmişse, **Şekil 4.43**’te verilen devreAnaliz() fonksiyonu, ilmek akımları metoduna göre devreyi analiz etmeye başlar.



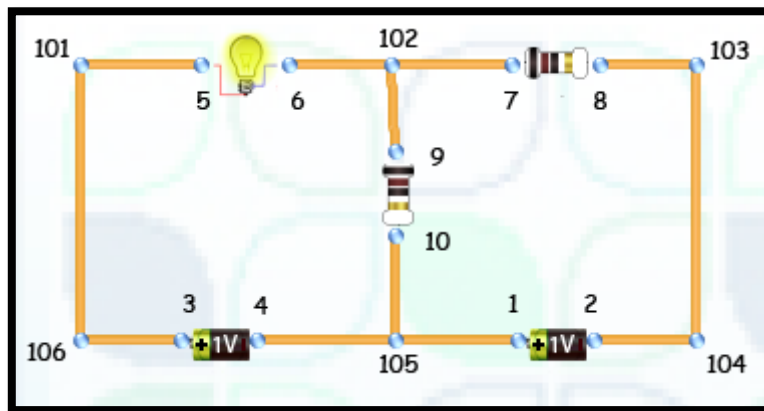
```

void
CircuitWindow::devreAnaliz()
{
    if (isVisible())
    {
        trashMeshMem();
        findConnectedNodeId();
        //mvNodeLoop için node id'lerini ekle: kontrol için hazırlık
        for (TmiNode nn = mvNode.begin(); nn<mvNode.end() ; nn++)
        {
            mvNodeLoop[mvNodeLoop.size()].push_back((*nn)->getId());
        }
        addLoopToMesh();
        // DEBUG
        // yaz_ConnectedNodeId();
        // showNodeLoop();
        // showMesh();
        winnowMesh();
        showMesh();
        makeMatris();
    }
}

```

**Şekil 4.43:** Devrenin analiz edilebilmesi için gerekli fonksiyonları çağıran fonksiyon.

Devre analizi uzun süren ve CPU'yu meşgul eden bir işlem olduğu için bu analizin sadece devre penceresi açıkken yapılmasını sağlayacak bir kontrolden sonra önceki devre analizlerinden kalan veriler silinerek işlem başlatılır. **Şekil 4.44**'te, analizi yapılacak örnek basit elektrik devresi verilmiştir. Bu devrede öncelikle ilmekleri belirleyebilmek için **Node**'ların listesi mvNodeLoop vektörünün içine yerleştirilir. Amaç, devrede mvNodeLoop vektörünün içinde ilmek oluşturan **Node**'ların sıralı listesini bulmaktır.



**Şekil 4.44:** CircuitWindow sınıfının devre analizini açıklamak için kullanılan örnek devre.

Başlangıçta, mvNodeLoop vektöründe 16 adet node'un id'leri yer almaktayken ikinci aşamada her Node'un komşusu listeye eklenir. Örneğin, 6 id'li Node'un 5 ve 102 id'li iki adet komşusu varken, 105 id'li Node'un 4, 10 ve 1 id'li üç komşusu vardır. İkidenden fazla komşusu olan Node'ların id'si, kabloların birleşme veya ayrılma noktasında olduğu sonucuna varılarak uzak komşularının bulunabilmesi için mvNodeLoop vektörüne eklenir. Bu kol için yine komşulara bakılır ve yeni komşu ya da komşular kola eklenerek mvNodeLoop ilmek oluşturma ihtimali olan her Node'u içine alarak genişler. **Şekil 4.44**'te verilen iki ilmekli 16 bağlı noktası içeren basit bir devre de bile 855 adet alternatif yol oluşmaktadır. **Tablo 4.7**'de örnek olarak 101 numaralı Node'un olası yolları verilmiştir.

**Tablo 4.7:** 101 id'li Node'un komşu listesi.

10 - 101 *
36 - 101 *5 *
37 - 101 *106 *
76 - 101 *5 *6 *
77 - 101 *106 *3 *
122 - 101 *5 *6 *102 *
123 - 101 *106 *3 *4 *
168 - 101 *5 *6 *102 *7 *
169 - 101 *5 *6 *102 *9 *
170 - 101 *106 *3 *4 *105 *
224 - 101 *5 *6 *102 *7 *8 *
225 - 101 *5 *6 *102 *9 *10 *
226 - 101 *106 *3 *4 *105 *1 *
227 - 101 *106 *3 *4 *105 *10 *
292 - 101 *5 *6 *102 *7 *8 *103 *
293 - 101 *5 *6 *102 *9 *10 *105 *
294 - 101 *106 *3 *4 *105 *1 *2 *
295 - 101 *106 *3 *4 *105 *10 *9 *
364 - 101 *5 *6 *102 *7 *8 *103 *104 *
365 - 101 *5 *6 *102 *9 *10 *105 *4 *
366 - 101 *5 *6 *102 *9 *10 *105 *1 *
367 - 101 *106 *3 *4 *105 *1 *2 *104 *
368 - 101 *106 *3 *4 *105 *10 *9 *102 *
440 - 101 *5 *6 *102 *7 *8 *103 *104 *2 *
441 - 101 *5 *6 *102 *9 *10 *105 *4 *3 *
442 - 101 *5 *6 *102 *9 *10 *105 *1 *2 *
443 - 101 *106 *3 *4 *105 *1 *2 *104 *103 *
444 - 101 *106 *3 *4 *105 *10 *9 *102 *7 *
445 - 101 *106 *3 *4 *105 *10 *9 *102 *6 *
536 - 101 *5 *6 *102 *7 *8 *103 *104 *2 *1 *
537 - 101 *5 *6 *102 *9 *10 *105 *4 *3 *106 *
538 - 101 *5 *6 *102 *9 *10 *105 *1 *2 *104 *
539 - 101 *106 *3 *4 *105 *1 *2 *104 *103 *8 *
540 - 101 *106 *3 *4 *105 *10 *9 *102 *7 *8 *
541 - 101 *106 *3 *4 *105 *10 *9 *102 *6 *5 *
620 - 101 *5 *6 *102 *7 *8 *103 *104 *2 *1 *105 *
621 - 101 *5 *6 *102 *9 *10 *105 *1 *2 *104 *103 *
622 - 101 *106 *3 *4 *105 *1 *2 *104 *103 *8 *7 *
623 - 101 *106 *3 *4 *105 *10 *9 *102 *7 *8 *103 *
680 - 101 *5 *6 *102 *7 *8 *103 *104 *2 *1 *105 *4 *

**Tablo 4.7'nin devamı.**

681	-	101	*	5	*	6	*	102	*	7	*	8	*	103	*	104	*	2	*	1	*	105	*	10	*				
682	-	101	*	5	*	6	*	102	*	9	*	10	*	105	*	1	*	2	*	104	*	103	*	8	*				
683	-	101	*	106	*	3	*	4	*	105	*	1	*	2	*	104	*	103	*	8	*	7	*	102	*				
684	-	101	*	106	*	3	*	4	*	105	*	10	*	9	*	102	*	7	*	8	*	103	*	104	*				
736	-	101	*	5	*	6	*	102	*	7	*	8	*	103	*	104	*	2	*	1	*	105	*	4	*	3	*		
737	-	101	*	5	*	6	*	102	*	7	*	8	*	103	*	104	*	2	*	1	*	105	*	10	*	9	*		
738	-	101	*	5	*	6	*	102	*	9	*	10	*	105	*	1	*	2	*	104	*	103	*	8	*	7	*		
739	-	101	*	106	*	3	*	4	*	105	*	1	*	2	*	104	*	103	*	8	*	7	*	102	*	9	*		
740	-	101	*	106	*	3	*	4	*	105	*	1	*	2	*	104	*	103	*	8	*	7	*	102	*	6	*		
741	-	101	*	106	*	3	*	4	*	105	*	10	*	9	*	102	*	7	*	8	*	103	*	104	*	2	*		
800	-	101	*	5	*	6	*	102	*	7	*	8	*	103	*	104	*	2	*	1	*	105	*	4	*	3	*	106	*
801	-	101	*	106	*	3	*	4	*	105	*	1	*	2	*	104	*	103	*	8	*	7	*	102	*	9	*	10	*
802	-	101	*	106	*	3	*	4	*	105	*	1	*	2	*	104	*	103	*	8	*	7	*	102	*	6	*	5	*
803	-	101	*	106	*	3	*	4	*	105	*	10	*	9	*	102	*	7	*	8	*	103	*	104	*	2	*	1	*

Komşu bulma işlemine, takip edilen yol Node'un kendisine çıkıncaya ya da komşusu olmayan bir Node'a gidinceye kadar devam edilir. Tüm Node'lar için liste hazırlandığında oluşturulan listenin ayıklanması için winnowMesh() fonksiyonu çağrılır.

winnowMesh() fonksiyonunda iki temel temizlik işlemi vardır: İlki, farklı yönlerden elde edilen ve birbirinin aynı olan ilmeklerin elenmesidir. Bu eleme işlemini yapabilmek için öncelikle ilmeği oluşturan Node'ların id'leri küçükten büyüğe sıralanır ve eşitlikleri kontrol edilir. Her iki adımda, standart c++ şablonunda yer alan std::sort ve std::equal fonksiyonlarıyla yerine getirilir. Şekil 4.45'te aynı ilmekleri temizleyen kod parçası verilmiştir.

```
miMesh=mvMesh.begin();
while(miMesh != mvMesh.end())
{
    TmiIntMatris iki = miMesh;
    iki++;
    while(iki != mvMesh.end())
    {
        TmvInt yedek1, yedek2;
        yedek1 = miMesh->second;
        yedek2 = iki->second;
        std::sort(yedek1.begin(), yedek1.end());
        std::sort(yedek2.begin(), yedek2.end());

        if (std::equal(yedek1.begin(),
                     yedek1.end(),
                     yedek2.begin()))
        {
            mvMesh.erase(iki--);
        }
        iki++;
    }
    miMesh++;
}
```

**Şekil 4.45:** Birbirinin aynısı olan ilmekleri temizleyen kod.

İkinci adımda birbirini kapsayan ilmekler bulunarak vektörden çıkartılır. Şekil 4.46’da birbirini kapsayan ve uzun ilmekleri bulup çıkartan kod verilmiştir.

```

miMesh=mvMesh.begin();
while(miMesh != mvMesh.end())
{
    TmiIntMatris iki = miMesh;
    iki++;
    while(iki != mvMesh.end())
    {
        TmvInt yedek1, yedek2;
        yedek1 = miMesh->second;
        yedek2 = iki->second;
        if (*(yedek1.end()-1) == *(yedek2.end()-1) &&
            *yedek1.begin() == *yedek2.begin())
        {
            mvMesh.erase(iki--);
        }
        iki++;
    }
    miMesh++;
}

```

Şekil 4.46: Birbirini kapsayan ilmekleri temizleyen kod.

Temizleme işleminden sonra **Tablo 4.7**’deki bağlantı listelerinden geriye sadece gerçekten ilmek oluşturan ve **Tablo 4.8**’da verilen iki ilmek listede kalmayı başarır. Bu temizleme aşamasından sonra ilmek üzerindeki Node’ları herhangi bir bileşene ait olup olmadığının ve eğer Component’in yönü önemliyse ilmekle Component’in yönünün aynı olup olmadığının belirlenmesine geçilir.

**Tablo 4.8:** Tekrarlı ve uzun ilmekler temizlendikten sonra kalan ilmekler.

0	-	1	*	2	*	104	*	103	*	8	*	7	*	102	*	9	*	10	*	105	*
1	-	3	*	4	*	105	*	10	*	9	*	102	*	6	*	5	*	101	*	106	*

Node sınıfının owner() fonksiyonu, eğer node bir component’e aitse ait olduğu component’in işaretçisini döndürmektedir. Örneğin, **Tablo 4.8**’da verilen sıfır numaralı ilmek üzerinde 1, 2, 8, 7, 9 ve 10 numaralı node’ların owner fonksiyonları null dışında değer döndürürler. Bir component’in o ilmeğe ait olabilmesi için her iki node’unun da ilmek üzerinde olması gerektiği için ilmek üzerinde olduğu öngörülen component’lerin node çiftinin kontrolü yapılır. Bu kontrol sırasında eğer component’in yönü ilmeğe ters ise CircuitWindow::elemanYonKontrol (int satir, int node1, int node2) fonksiyonu -1 değilse 1 değerini döndürerek işaretlenmesi sağlanır.

Yukarıda belirtilen hazırlıklardan sonra ilmek akımlarının çözümü için gerekli olan denklemler oluşturulur ve katsayılar matrisi hazırlanmasına geçilir. Bu aşamada öncelikle matrisi oluşturacak olan component'lerin id'leri sonra da değerleri matris formunda yazılır. **Tablo 4.9**'da makeMatris() fonksiyonu tarafından oluşturulan debug.txt'den alınan component id'si ve değerleri matrisleri yer almaktadır.

**Tablo 4.9:** Direnç ve pillerin oluşturduğu matrisler.

<pre>##1~~~~~Direnç Component Matris~~~~~ 0   7 +9 - 9   1   9 - 9 +5  </pre>
<pre>##1~~~~~Pil Component Matris~~~~~ 0   1   1   3  </pre>
<pre>##1~~~~~Direnç Değer Matris~~~~~ ##4   2 1   ##4   1 3  </pre>
<pre>##1~~~~~Pil Değer Matris~~~~~ ##4   -1   ##4   -1  </pre>

Devredeki ilmek sayısına bağlı olarak oluşan matrislerin boyutları oldukça büyük olabilmektedir. Bu matrislerle ilgili hesaplarda sıkıntı yaşanmaması için daha önce açıklanan kütüphanelerden biri olan GNU Scientific Library (GSL) kullanılmıştır. **Şekil 4.47**'de gsl\_linalg\_LU\_solve() fonksiyonu ile çözülen matrisin gsl\_vector olarak tanımlanan x değişkenine aktarılması gösterilmektedir.

```
// gsl kullanarak oluşan matrisi çöz
// bilinen hata: birbirinden bağımsız iki
// devre oluşturursan tek devre gibi çözmeye çalışıyor-->kısmen aşıldı
TmiFloatMatris a = resistanceMatris.begin();
gsl_matrix_view r = gsl_matrix_view_array (
    direnc,
    resistanceMatris.size(),
    (*a).size());
gsl_vector_view v = gsl_vector_view_array ()
    pil,
    resistanceMatris.size());
gsl_vector *x = gsl_vector_alloc (resistanceMatris.size());
int s;
gsl_permutation * p = gsl_permutation_alloc (resistanceMatris.size());
gsl_linalg_LU_decomp (&r.matrix, p, &s);
gsl_linalg_LU_solve (&r.matrix, p, &v.vector, x);
```

**Şekil 4.47:** Matrislerin GSL kullanılarak çözülmesi.

GSL yardımıyla akımlar bulunduktan sonra tüm node'ların akımları önce sıfıra eşitlenir. Daha sonra toplam akım, **Şekil 4.48**'de verilen kod yardımıyla akımın geçtiği ilmek üzerinde bulunan node'lara aktarılır.

```
// yeni yöntem : her nodun ait olduğu ilmek'leri bul. ilmek akımlarını topla.
for(TmiNode nit = mvNode.begin(); nit != mvNode.end(); nit++)
{
    int i = 0;
    for(miMesh=mvMesh.begin(); miMesh != mvMesh.end(); miMesh++)
    {
        if(isNodeInMesh(*nit, miMesh->second))
        {
            if(gsl_isinf(gsl_vector_get (x, i))
                || gsl_isnan(gsl_vector_get (x, i)))
            {
                (*nit)->setCurrent(0);
            }
            else
            {
                (*nit)->setCurrent((*nit)->getCurrent()
                    + gsl_vector_get (x, i) );
            }
        }
        i++;
    }
}
```

**Şekil 4.48:** İlmek akımlarının node'lara aktarılması.

Her devre elemanı, üzerinden geçen akıma göre gerekli tepkiyi verebilmektedir. Böylece, lamba üzerinden geçen akıma göre uygun parlaklığı ayarlayabilmektedir.

Node ve component'lerin silinmesini sağlayan mDead özellikleri, logic() fonksiyonundaki bir döngüyle kontrol edilirler. Component'leri ve node'ları silen döngüler **Şekil 4.49** ve **Şekil 4.50**'de verilmiştir.

```

{
//silinen component'leri temizle
std::vector<Component*>::iterator i = mvComponent.begin();
while (i != mvComponent.end())
{
    Component *compo = (*i);

    if (compo->mDead)
    {
        compo->node1->setDead(true); //node'ların silinmesi için
        compo->node2->setDead(true);
        compo->node1=NULL;
        compo->node2=NULL;
        compo->mDead = false;
        delete compo;
        i = mvComponent.erase(i);
        mRefresh = true; // devreyi yeniden analiz et
    }
    else {
        i++;
    }
}
}

```

Şekil 4.49: Component'lerin silinmesini sağlayan döngü.

```

//silinen nodları temizle
std::vector<Node*>::iterator j = mvNode.begin();
while (j != mvNode.end())
{
    Node *nod = (*j);

    if (nod->getDead())
    {
        clearNodeConnection(nod);
        nod->setDead(false); //henüz silme gerçekleşmeden
        delete nod;
        j = mvNode.erase(j);
        mRefresh = true;
        circuitWindow->setNodeCreate(true);
    }
    else {
        j++;
    }
}
}

```

Şekil 4.50: Node'ların silinmesini sağlayan döngü.

Bir node sürüklenirken eğer başka bir node'un üzerine gelirse, bağlantı işareti ortaya çıkar. Bu bağlantı işareti çıktığı sırada, sürüklenen node bırakılırsa bir bağlantı oluşturulur. Sürüklenen yerde bir node olup olmadığını kontrol eden ve bağlantı gerekiyorsa gerekli değişkenleri hazırlayan kod Şekil 4.51'de verilmiştir.

```

//çakışma kontrolünü sadece pencere görünürken yap
if (collisionCheck && isVisible())
{
    int mx=0, my=0;
    bool found=false;
    j = mvNode.begin();
    while (j != mvNode.end())
    {
        Node *nod = (*j);
        if (nod->getMoving())
        {
            mx = nod->getX();
            my = nod->getY();
            movingNode = nod;
            found = true;
            break;
        }
        j++;
    }
    if (found)
    {
        j = mvNode.begin();
        while (j != mvNode.end())
        {
            Node *nod = (*j);
            if (nod != movingNode)
            {
                targetNode = nod;
                int tx = nod->getX();
                int ty = nod->getY();
                if (mx > tx-10 && mx < tx+10 &&
                    my > ty-10 && my < ty+10 &&
                    nod->getFromLink())
                {
                    collisionNodeX = tx;
                    collisionNodeY = ty;
                    nodeCollision = true;
                    break;
                }
                else
                    nodeCollision = false;
            }
            j++;
        }
    }
}
}
}

```

Şekil 4.51: Sürüklenen node'un diğer node'larla çakışma kontrolü.

Oyuncu, “heybe penceresi” açıkken herhangi bir elektrik devre elemanını “Kullan” düğmesine basarak devre penceresine geçirebilir. Bunun için inventoryWindow::action() fonksiyonunda uygulanan kontrol kodu Şekil 4.52’de verilmiştir.



```

if (event.getId() == "use")
{
    if (item->isEquipment())
    {
        if (item->isEquipped())
            Net::getInventoryHandler()->unequipItem(item);
        else
            Net::getInventoryHandler()->equipItem(item);
    }
    else if (item->getId()>1099 && item->getId()<1200)
    {
        if (circuitWindow->isVisible())
        {
            circuitWindow->distributeOlay(item);
        }
        else
        {
            localChatTab->chatLog("Bu nesneyi sadece \
devre tamir ederken kullanabilirsin.", BY_SERVER);
        }
    }
}

```

**Şekil 4.52:** Heybe penceresinde nesneyi kullan düğmesine basıldığında devre penceresine aktaran kod.

Şekil 4.52’de CircuitWindow::distributeOlay(Item \*it) fonksiyonu ile “devre penceresine” aktarılan nesnenin heybeden çıkartılması gerekmektedir. Ancak bir bilgi ile konuşma sürerken aynı zamanda bir nesnenin kullanılması eAthena 1.0 txt’de mümkün olmadığı için devre penceresinde bulunan bir konteyner, kullanılan nesnelere biriktirir ve pencere kapanırken toplu bir şekilde sunucuya iletir. Şekil 4.53’te verilen bu kod sayesinde, oyuncunun istemcide yaptığı nesne transferleri doğru bir şekilde aktarılmış olur. Ayrıca oyuncunun nesneyi kullanmaktan vazgeçerek heybeye iade etmesi de bu yöntemle mümkün olabilmektedir.

```

void
CircuitWindow::sendUsedItem ()
{
    for(mIterUsedItem = mUsedItem.begin();
        mIterUsedItem != mUsedItem.end();
        mIterUsedItem++)
    {
        Net::getInventoryHandler()->useItem(*mIterUsedItem);
    }
    mUsedItem.clear();
}

```

**Şekil 4.53:** Devre penceresinde kullanılan nesnelere sunucuya ileten sendUsedItem() fonksiyonu.

### 4.3.1.6 Görev Penceresinin Geliştirilmesi

Bölüm 4.2.2.4'te bahsedildiği gibi görev penceresi oyuncuların görevlerini ve hangi görevleri gerçekleştirdiklerini öğrenmesi amacıyla geliştirilmiştir. Mevcut sürümde görev penceresinin güncellenebilmesi için görev bilgisayarından birisiyle iletişim kurmak gerekmektedir. Otomatik güncelleme yapılabilmesi için eAthena tarafından ayrılmış paket adreslerinin değiştirilmesi ve istemcinin de paketleri alacak şekilde uyarlanması gerekmektedir. Bir görev bilgisayarı arka planında diğer bilgilerle aynı mekanizmaya sahiptir. Ayrıca Backspace tuşuna her basıldığında görev penceresi en son güncellendiği haliyle görüntülenebilmektedir. **npchandler** gelen verinin görev bilgisayarına ait olduğuna karar verdikten sonra bu veriyi `missionWindow::parse()` fonksiyonu aracılığıyla görev penceresine aktarır. **Şekil 4.54**'te görev penceresine gelen XML verinin sadeleştirilmiş haliyle sunulmuştur.

```
<?xml version="1.0" encoding="utf-8" ?>
<mission>
  <mainmission name = "Görev 1." >
    <submission status="0"
      image="elektrik/check_it.png"
      label="Alt görev 1." >
      <text>
        <addrow text=" ##1 Açıklama" />
      </text>
    </submission>
    <submission status="0"
      image="elektrik/check_it.png"
      label="Alt görev 2." >
      <text>
        <addrow text=" ##1 Açıklama" />
      </text>
    </submission>
  </mainmission>
  <mainmission name = "Görev 2." >
    <submission status="0"
      image="elektrik/check_it.png"
      label="Alt görev." >
      <text>
        <addrow text=" ##1 Açıklama" />
      </text>
    </submission>
  </mainmission>
</mission>
```

Şekil 4.54: Görev penceresine gelen XML veri.

<mission> tag'ıyla görev bilgisayarına ait verinin başladığı bildirildikten sonra, yapı 2 temel kısımdan oluşmaktadır. Birinci kısım, ana görev tanımlamasında kullanılan <mainmission> tag'ıdır. Bu tag görev penceresinin sol kısmında yer alan düğmelerin oluşmasını sağlar. İkinci kısım ise <submission> ile tanımlanan ana görevin altında alt görevlerin açılmasını sağlayan tag'lardır. XML yapıyı oluşturan tag'lar ve açıklamaları **Şekil 4.54**'te verilmiştir.

**Tablo 4.10:** Görev penceresini tanımlayan XML tag'lar ve açıklamaları.

mission	Görev penceresine gelen kök tag.
mainmission	Ana görev eklemek için kullanılan tag. name parametresi ile pencerede bulunan düğmenin başlığını tanımlar.
submission	Her ana görevin altında alt görevler açmak için kullanılır. <ul style="list-style-type: none"> <li>• <b>status:</b> Görev tamamlanıp tamamlanmadığını belirten bir sayı belirtir.</li> <li>• <b>image:</b> Görevin yan tarafında bulunan resmin yolunu ve adresini belirtir. Varsayılan olarak onay ve çarpı işaretleri içerir.</li> <li>• <b>label:</b> Alt görevi tanımlayan, bir cümle içerir.</li> </ul>
text	Alt görevlerin üzerine gelindiğinde çıkan açıklama metnini belirtir. addrow alt tag'ıyla bu açıklamaya satır eklenir.

**Şekil 4.55**'de bir ana görevi saklamak için kullanılan yapı (struct), yapıyı saklamakta kullanılan vektör ve vektörün elemanlarına ulaşmak için iteratör tip tanımları verilmiştir. Yapıdan da görüleceği gibi, ana görevler sadece kendi bilgilerini tutmakta ve alt görevlerin bilgisine sahip olamamaktadırlar. Bu durumun ana görevler ile alt görevler arasında ilişki kurmaya yetmemesinden dolayı, alt görevlerde sahibi olduğu ana görev açıkça belirtilmiştir.

```

struct SmMainMission{
    Button *mainButton;
    TSubMissions subMissions;
};

typedef std::map<std::string, SmMainMission*> TMainMissions;
typedef std::map<std::string, SmMainMission*>::iterator TMainMissionsIter;

```

**Şekil 4.55:** Ana görevi oluşturan yapı ve konteyner.

Bir alt görev; sahibi olan ana görevi, alt görev bilgilerini ve görevin yapılıp yapılmadığını SmSubMisson yapısı ile saklayabilir. Ayrıca imleç üzerine geldiğinde

açıklama kutusunu görüntüleyebilmek için görünebilir olup olmadığını tutan bir boolean değişkene sahiptir. Şekil 4.56'da bir alt görevin yapısı, vektör ve iteratörünün tip tanımı verilmiştir.

```
struct SmSubMission{
    std::string mainName;
    TextBox *oneTarget;
    ImageWidget *oneImage;
    BrowserBox *oneExplain;
    int oneStatus;
    bool oneVisible;
};

typedef std::vector<SmSubMission*> TSubMissions;
typedef std::vector<SmSubMission*>::iterator TSubMissionsIter;
```

Şekil 4.56: Alt görevi oluşturan yapı ve konteyner.

Ana görev ve alt görevlerin birlikte çalışmalarına örnek olarak Şekil 4.57'de yeni görev bildirimini geldiğinde eskilerinin temizlenmesini sağlayan MissionWindow::clearMissions() fonksiyonu verilmiştir. Bu fonksiyonda ana görevlere ait alt görevler, ilgili vektör içinde bulunarak silinmekte, alt görevi kalmayan ana görev daha sonra silinmektedir.

```
void
MissionWindow::clearMissions()
{
    for(mMainMissionIter = mMainMission.begin();
        mMainMissionIter != mMainMission.end();
        mMainMissionIter++)
    {
        SmMainMission *temp;
        temp = mMainMissionIter->second;
        TSubMissions subtemp= temp->subMissions;

        for(mSubMissionsIter = subtemp.begin();
            mSubMissionsIter != subtemp.end();
            mSubMissionsIter++)
        {
            if ((*mSubMissionsIter)->oneTarget != NULL)
                delete (*mSubMissionsIter)->oneTarget;
            if ((*mSubMissionsIter)->oneImage != NULL)
                delete (*mSubMissionsIter)->oneImage;
            if ((*mSubMissionsIter)->oneExplain != NULL)
                delete (*mSubMissionsIter)->oneExplain;
        }
        subtemp.clear();
        delete temp->mainButton;
    }
    mMainMission.clear();
}
```

Şekil 4.57: Mevcut görevlerin silinmesini sağlayan clearMission() fonksiyonu.

### 4.3.1.7 Özel İçerikli Pencerelelerin Geliştirilmesi

elektroGame içinde bir pencere, kısayol tuşlarından birine basıldığında görüntülenebilir ya da oluşturulabilir. Ayrıca bilgiler **npchandler** aracılığıyla bir pencerenin oluşmasına ya da görüntülenmesine neden olabilirler. Özellikle eğitsel içerik, oyun veya simülasyon eklemek amacıyla programcı kendi özel içerikli penceresini geliştirebilir. Teknik olarak özel içerikli bir pencerenin diğer pencerelerden bir farkı bulunmamaktadır. Ancak oyunun oynanışı bir bütün olarak ele alındığında, standart MMORPG özelliklerinin dışına çıkan ve elektroGame'e eğitsel özellikler katan pencereler için bu ifade kullanılabilir. Örneğin, dirençlerin renk kodlarını anlatan ve **DiRenK** adlı bilgenin oluşturduğu pencere bu yapıya örnek olarak verilebilir.

Bir pencere, eğer grafik yapısı elektroGame'in tema yapısını miras alacaksa Window sınıfından türetilebilir. Ancak daha yalın ve farklı amaçlar için tasarlanan bir pencere, doğrudan `gcn::Window` ve `gcn::WidgetListener` sınıflarından türetilerek daha özgün bir yapıya sahip olabilir. Ayrıca pencerenin kendi olay yönetimine sahip olması gerekiyorsa, türetilen sınıflara `gcn::ActionListener` da eklenmelidir.

Window ve `gcn::ActionListener` sınıfından türetilen bir pencerenin temel olarak 3 fonksiyonu ederek yeniden tanımlaması gerekmektedir. Bu fonksiyonlar ve kısaca görevleri **Tablo 4.11**'de verilmiştir.

**Tablo 4.11:** Bir pencerenin sahip olması gereken 3 temel fonksiyon.

Logic	Oyunun çalışması sırasında her 10ms de bir çağırılan fonksiyondur. Zamanla değişmesi olası değişkenlerin kontrolünü kolaylaştırmaktadır. Ancak fonksiyonun yoğun işleme neden olması CPU kullanım oranını aşırı yükselterek yavaşlamaya neden olabilmektedir.
Action	Widget nesnelere üreteceği olayları yöneten action listener tarafından çağırılır. Gerekli durumlarda yazılımcı aşağıdaki kodu kullanarak kendi olayını üretebilir. <pre>const std::string &amp;actionEventId="node_shift"; setActionEventId(actionEventId); distributeActionEvent();</pre>
Draw	Pencere üzerinde çizilmesi gereken grafikler için tazeleme hızına uygun olarak çağırılan fonksiyondur. Kendi draw() fonksiyonu olmayan her nesne için kullanılabilir.

C++'ın nesne yönelimli yapısına uygun olarak her sınıfın kendi kurucu (constructor) ve yıkıcı (destructor) fonksiyonlarının da olması beklenebilir.

Açıklanan fonksiyonlar yazılarak elektroGame'in yapısıyla uyumlu hale getirilen bir pencere, ilaveten kendi görevlerini de yapacak şekilde geliştirilebilir. Geliştirilen pencerenin oyunda eklenebileceği temel iki yer mevcuttur. Birincisi game.cpp üzerinde **static void createGuiWindows()** fonksiyonudur. Bu fonksiyon tüm oyun boyunca açık kalacak pencereler için hem **create** etme hem de pencereye erişimi sağlayan işaretçiyi global olarak tanımlama alanıdır.

İkinci yöntem, pencerenin gerektiği an oluşturulması ve pencere kapandığı an sistemden aldığı kaynakları iade ederek kapanması şeklindedir. Bu yöntem için pencerenin açılmasını tetikleyecek olayı yönetmek gerekmektedir. Örneğin, bir bilgeye basıldığında pencere açılacaksa gerekli kod **npchandler** sınıfının içinde yazılmalıdır. **Şekil 4.58**'de void NpcHandler::handleMessage(MessageIn &msg) fonksiyonu içinde 253 id'li bilgenin açtığı pencerenin kodu görülmektedir. Gelen mesajın türü msg.getId() değişkeninde saklandığı için öncelikle değeri kontrol (switch) edilir. Eğer gelen paket **SMSG\_NPC\_MESSAGE** sabitiyle tanımlanmış 0x00b4 başlıklı paket ise bilgenin üzerine tıkladığı bilgisi sunucu tarafından istemciye gönderilmiştir.

```
else if (being->getJob() == 253) //DirenK
{
    ResColorWindow *direnk = new ResColorWindow;
    direnk->setVisible(true);
    npcText.str("");
}
```

**Şekil 4.58:** Bilgeye tıkladığı zaman oluşturulan pencere.

#### 4.3.1.8 Dünya Yapımı

elektroGame'de oyuncunun oyunda gördüğü ve içinde dolaştığı ortama *dünya* denilmiştir. Üç boyutlu oyunlardan farklı olarak elektroGame dünyaları farklı katmanların (layer) üst üste basıldığı grafiklerden oluşmaktadır. Katmanlar ise 32x32 piksellik karelere bölünebilen sprite yardımıyla oluşturulur. **Şekil 4.59**'da katmanları oluşturmakta kullanılan sprite'a bir örnek verilmiştir.

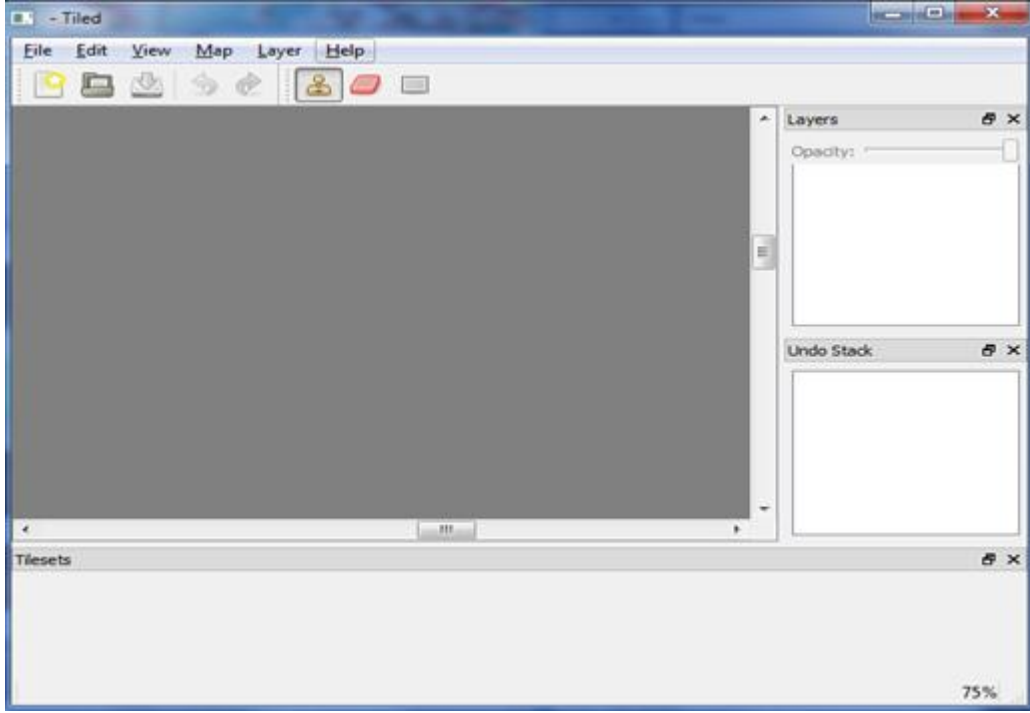


Şekil 4.59: Dünya yapımında kullanılan sprite.

2D oyunlarda sprite'lardan oluşan dünyaları yapmak için çeşitli yardımcı programlar bulunmaktadır. elektroGame'de TMW'nin yapısıyla uyumlu olan ve TMW için yeni özellikler eklenen **Tiled Map Editor** (TMP) kullanılmıştır. TMP programının iki farklı sürümü bulunmaktadır. İlki Java ile yazılmıştır. Bu sürümde, hazırlanan dünyanın, eAthena sunucu için geliştirilmiş olan WLK formatına dönüşüm yapılabilmektedir. Ancak bazı yapısal problemler içerdiği için geliştirme sırasında tasarımcılar için sıkıntı oluşturabilmektedir. İkinci sürüm bu sorunları gidermek için QT ile baştan yazılmıştır. Yeni haliyle çok daha hızlı ve kullanışlıdır. Henüz WLK formatına destek sağlanmadığı için dünyaların sunucuya uygun hale getirmek için eski sürümünün kullanılması gerekmektedir.

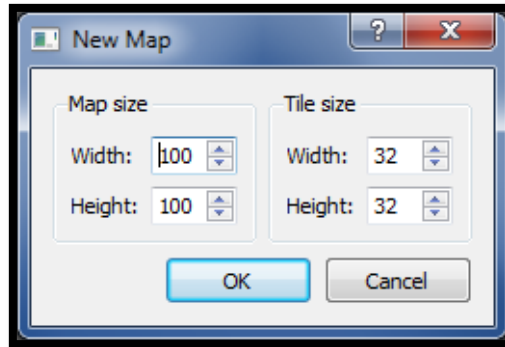
elektroGame için hazırlanan bir dünya temel olarak dört katmandan oluşur: Bu katmanlar sırasıyla **ground** (zemin), **fringe** (kenar), **top** (üst), **collision** (çakışma) katmanlarıdır. Ground, char'ın üzerinde yürüyeceği, fringe, char'la aynı seviyedeki nesnelerin, top char'ın altından geçebileceği, collision ise char'ın ulaşamayacağı alanların işaretlendiği katmandır. Ayrıca dünya üzerine parçacık efektleri eklemek, bilgelerin yerini göstermek ve geçiş noktalarının koordinatlarını göstermekte kullanılan object group isimli katman eklenebilir. Araştırmacı tarafından, bu katmanlara hareketli tiled set'ler kullanabilmek için üç katman daha ilave edilmiştir. Bu katmanlar zemin, kenar ve üst katmanlarla birlikte çizilir ve ışık yanıp sönme gibi etkiler elde etmek amacıyla kullanılabilir.

TME (QT Edition) programının görüntüsü **Şekil 4.60**'ta verilmiştir. Pencere üç temel panelden oluşmaktadır: Hiçbir dünya açık olmadığı için koyu gri olarak görünen düzenleme alanı, dünya üzerine yerleştirilecek sprite'ların listelendiği "tilesets" alanı ve katmanların listelendiği "layers" paneli. Layers panelinin altında yer alan "undo stack" ise yapılan işlemleri listeleyerek geri almayı kolaylaştırır.



**Şekil 4.60:** TME (QT Editon) programı.

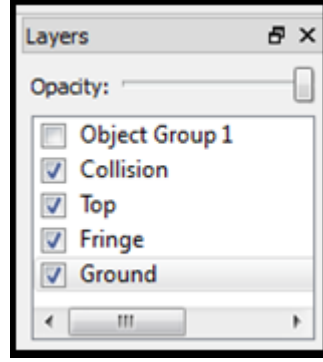
File menüsünden New seçeneği seçildiğinde, **Şekil 4.61**'de verilen, açılacak olan dünyanın boyutlarını soran iletişim penceresi gösterilmektedir. "Tile size" başlıklı grupta dünyayı oluşturacak olan resimlerin kaç piksel genişlik ve yüksekliğinde olacağı belirtilir. "Map size" bölümünde ise dünyanın kaç kareden oluşacağı tanımlanır.



**Şekil 4.61:** Oluşturulacak olan dünyanın boyutlarının ve bir karenin boyutlarının tanımlandığı pencere.

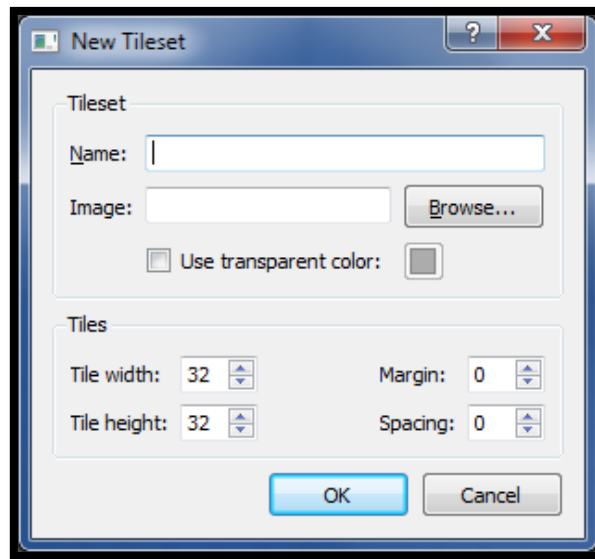


Yeni dünya oluştuktan sonra, Layer menüsünden Add Tile Layer seçeneği seçilerek sırasıyla ground, fringe, top ve collision katmanları oluşturulur. Katmanların isimleri ve sıraları istemci tarafından dikkate alındığı için **Şekil 4.62**'de görülen layer panelindeki gibi oluşturulmalıdır. Bu dört katmana ilave olarak Layer menüsünden Add Object Layer seçeneği ile object katmanı da eklenebilir.



**Şekil 4.62:** Bir dünya da bulunması gereken katmanlar.

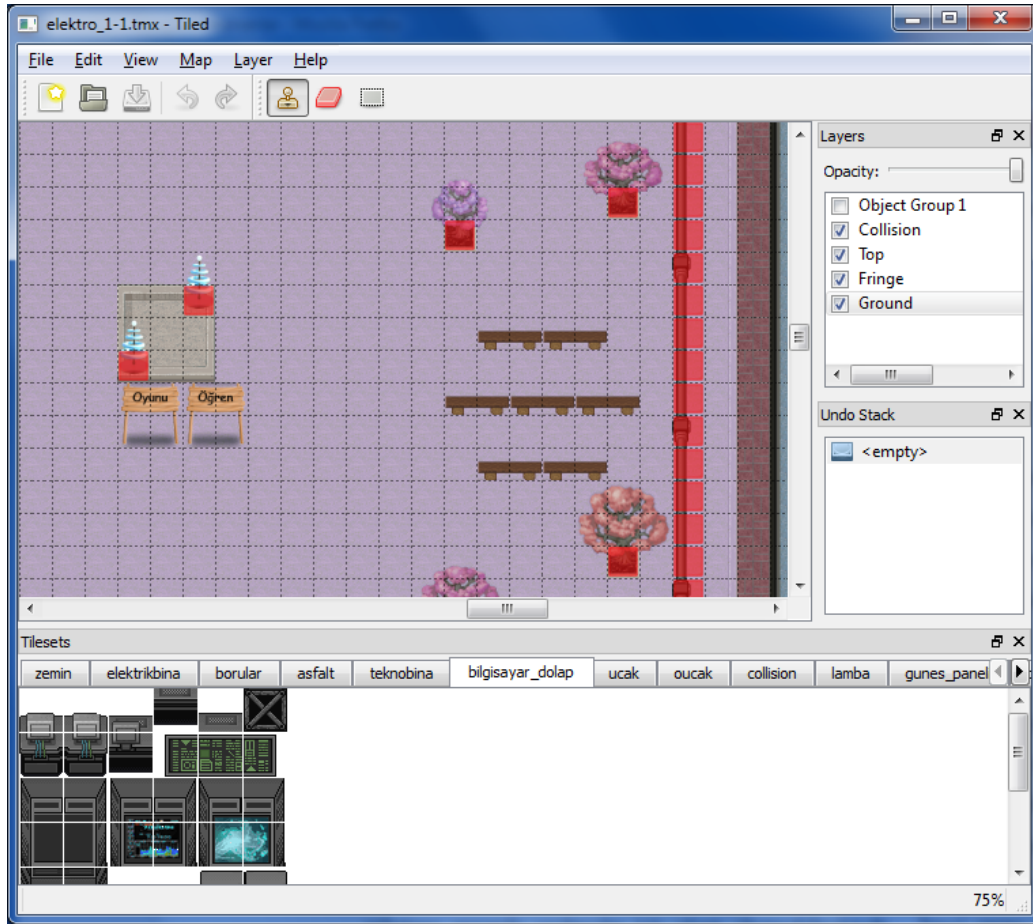
Katmanlar hazırlandıktan sonra, tileset'ler map menüsünden New Tileset seçeneği seçilerek eklenir. **Şekil 4.63**'te verilen New Tileset penceresinde genişlik ve yüksek değerleri ayarlanmalıdır. elektroGame'de tileset genişliği 32 piksel olmak zorundadır. Yükseklik ise 32'nin tam katları olabilir. Böylece ağaç, direk gibi nesnelere yerleştirmek kolaylaşmakla birlikte, oyuncunun hareketiyle pencerenin kayması sırasında bu şekilde yerleştirilen nesnelere ekrandan olması gerekenden biraz daha erken yok olabilmektedir.



**Şekil 4.63:** Tileset ekleme penceresi.

Ground, fringe ve top katmanları ile görüntü oluşturulduktan sonra, char'ın geçemeyeceği yerleri belirten collision katmanı hazırlanır. Bu katman iki parçadan oluşan özel bir tileset kullanır. Collision katmanı sadece tasarım aşamasında görüntülenir, oyuncu katmana ait grafiği göremez. Oyuna etkisi tileset'in 2. parçası basıldığında oyuncuların ve robotların o kareye girememesi şeklindedir.

**Şekil 4.64**'te elektro-1\_1.tmx olarak kaydedilen elektroGame'in Merkez Dünyasının tasarım ekranı verilmiştir.



**Şekil 4.64:** TME programıyla tasarlanmış Merkez Dünyası.

Görsel tasarım tamamlandıktan sonra, efekt eklemek için object katmanına gerekli açıklamaları içeren bir XML veri eklenmelidir. Bunun için TMX uzantısıyla kaydedilen istemci dosyası bir metin editöründe açılmalıdır. Editörde açılan dosya XML biçiminde görüntülenecektir. Layer tag'larının sonunda object tagı içine gerekli eklemeler yapılabilir. **Şekil 4.65**'te eklenen parçacık efektinin kodu verilmiştir.

```
<objectgroup name="Object Group 1"
    width="0" height="0"
    visible="0">
    <object name="graphics/particles/wispdance.particle.xml"
        type="PARTICLE_EFFECT"
        x="720"
        y="520"
        width="32"
        height="32"/>
</objectgroup>
```

**Şekil 4.65:** TMX uzantılı dosyaya eklenen parçacık efekt kodu.

İstemci üzerinde grafik, efekt ve ses gibi özelliklerin haritalarda bulunması gerekirken sunucu tarafında haritaların bu bilgilerin tamamını içermesine gerek bulunmamaktadır. Haritanın boyutu, robotların ve char'ların bulunamayacağı alanları gösteren collision katmanı gibi sunucuyu ilgilendiren bilgileri içeren özetlenmiş dosya, TMW (Java edition) WLK uzantılı dosya olarak kaydedilmelidir.

WLK uzantılı harita dosyalarının, sunucu üzerinde data klasöründe bulunması gereklidir. Ayrıca dosya isimlerinin, karakter kodlaması farklılıkları gibi durumlarda sunucunun çalışmasında problem çıkartma ihtimalini engellemek için resnametable.txt dosyası kullanılır.

#### 4.3.1.9 Parçacık Efektleri

RPG oyunlarda animasyon ve efektler, oyuncuları etkilemek için bol miktarda kullanılmaktadır. elektroGame'de de aynı amaçla görsel efekt oluşturmak için parçacık efekt tekniği kullanılmıştır. TMW geliştiricileri tarafından oluşturulan particle, particleemitter ve particlecontainer adlı üç adet sınıf, iki boyutlu harita üzerine değişik parametreler yardımıyla parçacık efekti ekleyebilir. Parçacık efektlerine çeşitli sesler de eklenebilmektedir.

Parçacık efektleri oyunda dünyanın hazırlanması aşamasında **Şekil 4.65**'te gösterildiği şekilde kod ekleyerek yapılabileceği gibi, bilgilerden gelen mesajlara uygun komutları ekleyerek de yapılabilir. Test penceresini oluşturmakta kullanılan XML'de kullanılan tag'lar **Tablo 4.3**'te açıklanmıştır.

**Tablo 4.3**'te test bilgilerinin doğru ya da yanlış yanıtlar sonrasında göstermesi için kullanılan effect tag'ı bu duruma örnek olarak verilebilir.

Test penceresinde ve devre penceresinde efekt eklemeyi kolaylaştırmak için `makeEffect(std::string type, std::string name, std::string ssound)` fonksiyonu tanımlanmıştır. Bu fonksiyon belirtilen tipe göre efekti oluşturur. **Şekil 4.66**'da `makeEffect` fonksiyonu verilmiştir. `particleEngine` sınıfına `text` ve `particle` olmak üzere iki tür bildirim yapılabilir.

```
void TestDialog::makeEffect(std::string type, std::string name, std::string ssound)
{
    if (type=="particle")
    {
        Particle *dogruFX;
        dogruFX = particleEngine->addEffect("graphics/particles/"+name+".particle.xml", 0, 0);
        player_node->controlParticle(dogruFX);
    }
    else
    {
        particleEngine->addTextSplashEffect(name,
                                           player_node->getPixelX() + 16,
                                           player_node->getPixelY() + 16,
                                           sgc::Color(255, 0, 255),
                                           boldFont);
    }
    if (ssound != "")
    {
        sound.playSfx("sfx/"+ssound+".ogg");
    }
}
```

**Şekil 4.66:** Bilgelerin efekt eklemesi için hazırlanmış fonksiyon.

Kullanılacak olan efektler daha düzenli olmaları ve tanımlarının kolay yapılması amacıyla id'lerini ve yollarını gösteren `effects.xml` dosyasında listelenmişlerdir. **Şekil 4.67**'de `effects.xml` dosyasından birkaç satır gösterilmiştir.

```

<being-effects>
  <effect id="0" audio="sfx/system/levelup.ogg"
    particle="graphics/particles/levelup.particle.xml" />
  <effect id="1" audio="sfx/system/levelup.ogg"
    particle="graphics/particles/skillup.particle.xml" />
  <effect id="2" particle="graphics/particles/magic.generic.xml"/>
  <effect id="3" particle="graphics/particles/magic.white.xml"/> <!-- life -->
  <effect id="4" particle="graphics/particles/magic.black.xml"/> <!-- war -->
  <effect id="5" particle="graphics/particles/magic.red.xml"/> <!-- transmute -->
  <effect id="6" particle="graphics/particles/magic.green.xml"/> <!-- nature -->

```

Şekil 4.67: effects.xml dosyasının ilk satırları.

effects.xml dosyasında tanımlanan her bir efekt, efekti gerçekleştirmek için gerekli grafik dosyalarının yolu, parçacık sayısı gibi bilgilerin verildiği XML biçimli bir dosyayı işaret eder. Parçacık efekti oluşturmak için kullanılan bir dosyanın genel yapısı Şekil 4.68’de verilmiştir. Burada emitter olarak tanımlanan her bir tag yeni bir parçacık serisinin başlamasına neden olur. Parçacıklar bağımsız hareket edebilecekleri gibi bir başka parçacığın alt parçaları da olabilirler. Bunun için emitter tag’ı içindeyken yeni bir emitter tag’ı açılarak kendisine ait özelliklerin tanımlanması yeterli olmaktadır.

```

<effect>
  <particle
    position-x = "0"
    position-y = "0"
    position-z = "50"
    lifetime = "1">
    <emitter>
      <property name="position-x" value = "10"/>
      <emitter>
        <property name="position-x" value = "10"/>
      </emitter>
    </emitter>
  </particle>
</effect>

```

Şekil 4.68: Parçacık oluşturmak için kullanılan XML biçimli verinin genel yapısı.

#### 4.3.1.10 Nesne özellikleri

elektroGame’de kullanılan nesnelerin oyundaki görevleriyle ilgili açıklamalar Bölüm 4.2.1.5’te verilmişti. Nesnelerin oyuna eklenmesi için kullanılan yöntem, elektroGame’de kullanılan verilen XML yapılar içinde tutulması felsefesine uygun

olarak yapılandırılmıştır. İstemci nesnelerin resim, isim gibi özellikleri data klasöründe bulunan items.xml dosyasından öğrenir. Bu dosyada bulunan bilgiler, sadece oyuncuya bilgi vermek amacıyla kullanıldığı için oyuncular tarafından bu dosyalarda yapılan değişiklikler sadece görünümü değiştirecek ancak sonucu etkilemeyecektir. Bir nesnenin gerçek özellikleri, oyuncular tarafından erişilemeyen sunucu tarafında tutularak güvenlik artırılmaya çalışılmıştır.

Items.xml dosyası üç ana bölümden oluşmaktadır. Birinci bölümde saç özellikleri bir nesne sprite'ı gibi tanımlanarak char'ı oluşturan katmanlar gibi işlenebilmesi mümkün hale getirilmiştir. İkinci bölümde ise char'ın base sprite'ı (Şekil 1.1 ve Şekil 1.2) cinslere göre tanımlanmıştır. Mevcut sürümde her cinsi tek sprite set temsil etmekle birlikte, bu yöntem sayesinde kolayca yeni resimler eklemek mümkün olabilecektir. Şekil 4.69'da base sprite'ların tanımı verilmiştir. Üçüncü bölüm tanıtımı yapılan tüm nesnelerin sırayla tanımlandığı bölümdür.

```
<item id="-100" type="racesprite" name="Human">
  <sprite gender="male">player_male_base.xml</sprite>
  <sprite gender="female">player_female_base.xml</sprite>
</item>
```

Şekil 4.69: Base sprite'ların item olarak tanımlanması.

Item tag'ının özellikleri özellikle heybe penceresinde nesne hakkında bilgi veren kutu açıldığında etkisini gösterir. Şekil 4.70'te bir nesnenin özelliklerinin tanımlandığı kod parçası verilmiştir. Tablo 3.11'de ise özellikler açıklanmıştır.

```
<item id="1100"
  image="/elektrik/item-direnc-1.png"
  name="Direnç 1 Ohm"
  description="Direnç 1 ohm."
  effect=""
  type="usable"
  weight="1"
  max-per-slot="100"
  elektro-type="resistance"
  elektro-value="1"/>
```

Şekil 4.70: Bir item tag'ının özellikleri.

**Tablo 4.12:** item.xml’de kullanılan tag’lar ve özellikleri.

id	Nesneye has bir kimlik numarasıdır. Sunucuda ve istemcide nesnelerin eşgüdümü bu id sayesinde yapılır. ID’lerin aralıkları şöyledir: <ul style="list-style-type: none"><li>• Yiyecekler: 500-649</li><li>• Silahlar: 650-689</li><li>• Mühimmat: 690-699</li><li>• Nesne: 700-999</li><li>• Kıyafetler: 2001-2500</li><li>• Bilgisayar parçaları: 1020-1099</li><li>• Devre elemanları: 1100-1200.</li></ul>
image	Nesnenin heybede temsil edildiği resmin yolunu ve adını verir. Kök olarak ./data/graphics olarak belirtildiği için yol tanımı buna göre yapılmalıdır.
name	Nesne adını veren string.
description	Nesneyi kısaca açıklayan metin.
Effect	Nesnenin etkisini tanımlar. Örneğin, “+50HP” sağlık puanını 50 puan arttıracığını gösterir.
type	Nesnenin kullanım yerine göre tipini belirtir. Type özelliği; <ul style="list-style-type: none"><li>• usable</li><li>• equip-1hand</li><li>• cephane</li><li>• generic</li><li>• equip-torso</li><li>• equip-legs</li><li>• equip-feet</li><li>• equip-head</li><li>• equip-arms</li><li>• equip-shield</li><li>• equip-ring</li><li>• equip-charm</li></ul> değerlerinden birini alabilir.
weight	Nesnenin ağırlığıdır. Bir char gelişmişlik seviyesine göre sınırlanmış ağırlığı yanında taşıyabilir. Bazı nesnelerin ağırlık seviyeleri yükseltilerek oyuncu bazı nesneler arasında tercih yapmaya zorlanabilir.
max-per-slot	Heybede taşınabilecek maksimum nesne sayısı 100’dür. Ancak pek çok nesne gruplanarak heybede tek nesne olarak görünür. Bu özellik nesnenin grupta en fazla kaç tane olacağını belirtir.
hp	Nesnenin sağlık puanına etkisi.
mp	Nesnenin uzmanlık puanına etkisi.
slot	Nesne silah, kıyafet gibi char’ın belli bir alanında bulunuyorsa, yerleşim yerini gösterir. F4 ile açılan penceredeki yerleşim de slot numarasına göre yapılır.
weapon-type	Nesne bir silaha silahın türünü tanımlar.
attack-min	Silahın en düşük etki düzeyidir.
attack-delta	Silahın etkisinin değişim miktarıdır.
attack-range	Silahın etki edebileceği uzaklıktır.

**Tablo 4.12'**nin devamı.

elektro-type	elektroGame'de, TMW'deki nesne tiplerine ek olarak kullanılan nesne türleridir.
elektro-value	Direnç, pil gibi nesnelerin değeridir.
sprite	sprite bir özellik değil tag'dır. Kıyafetler gibi item tag'ıyla, tanımlanan nesnenin kullandığı sprite set'in yolu ve adı tanımlanır. Ortak nesneden renk değişimiyle türeyen nesnelere için renk kodu da tanımlanabilir.
sound	Nesnenin kullanımı ile seslendirilmesi istenen ses efektinin yolu ve adı tanımlanır.

İstemci tarafında XML dosyada tutulan bu bilgiler sunucu üzerinde metin (text) dosyada ve veri tabanı üzerinde tutulmaktadır. eAthena 1.0 TXT sürümü veri tabanı bağlantısını desteklemediği için bu özellikler Bölüm 4.3.2'de anlatıldığı gibi, araştırmacı tarafından eklenerek geliştirilmiştir.

Nesne bilgilerinin tutulduğu item\_db.txt dosyası, betik (script) dosyalarına benzer bir yapıda tanımlanmıştır. Diyez (#) işareti ile başlayan satırlar açıklama satırı olarak yorumlanır ve dikkate alınmazlar. Satır sonu işaretinden bölünen ifadeler virgül (,) ifadesinden parçalanarak yorumlanır. Yapının parçalanması virgülle ayrılan kaçınıcı eleman olduğuna göre yapıldığı için her satırdaki elemanların sırasına özellikle dikkat edilmelidir.

Bir nesnenin sunucu tarafında belirtilmesi gereken özellikler **Tablo 4.13'**te açıklanmıştır. Açıklamalar uyulması gereken sıralamaya göre yapılmıştır.

**Tablo 4.13:** item\_db.txt dosyasında tanımlanan bir nesneye ait özellikler.

ID	Nesnenin tanımlayıcı numarası.
Name	Nesnenin görünür ismi.
Label	Nesnenin uzun ismi.
Type	Nesnenin kullanım türünü belirler. <ul style="list-style-type: none"><li>• 0 - Usable</li><li>• 3 - Object</li><li>• 4 - Silah</li><li>• 5 - Kıyafet</li><li>• 10 - Cephane</li></ul>
Price	Nesnenin standart fiyatı.
Sell	Nesnenin satılıp satılamayacağı.
Weight	Nesnenin ağırlığı.
ATK	Nesnenin atak gücüne katkısı.



**Tablo 4.13**'ün devamı.

DEF	Oyuncunun hangi bölgesinin savunma gücüne kaç birim katkı yapacağı. 0 - All 1 - Lower Head 2 - Right Hand 4 - Robe/Garment 8 - Left Accessory 16 - Body/Armor 32 - Left Hand 64 - Feet 128 - Right Accessory 256 - Top Head 512 - Mid Head
Range	Nesnenin etki alanı.
Slot	Ekipman penceresinde hangi slota yerleştirileceği.
Gender	Nesnenin cinsiyet kısıtlaması.
View	Varsa nesnenin farklı görünüm seçenekleri.
{UseScript}	Nesne kullanıldığında çalıştırılacak olan script.
{EquipScript}	Nesne giyildiğinde çalıştırılacak script.

Eğer kullanılıyorsa veri tabanındaki nesne bilgilerinin, item\_db.txt dosyasının ve items.xml dosyasının eşgüdümlü olarak düzenlenmesi gereklidir. Aralarında meydana gelebilecek uyumsuzluklar hem oynanabilirliği düşürecektir hem de kararlılık sorunlarına yol açacaktır.

#### 4.3.1.11 Bilge Özellikleri

Oyunda kullanılan bilgelerin özellikleri tıpkı diğer oyun öğeleri gibi önceden detaylı şekilde tanımlanmalıdır. İstemcide bilgelerin özelliklerinin tanımlandığı dosya, npcs.xml dosyasıdır. Sunucu da ise, Bölüm 4.3.2.4'te anlatılacak kapsamlı bir betik dilinin kullanıldığı, map sunucu tarafından yönetilen bir yapıyla yönetilir.

npcs.xml dosyasının biçimi diğer XML biçimli verilere nispetle daha sadedir. **Şekil 4.71**'de bilge özelliklerini göstermek için iki adet bilgenin tanımlandığı temsili bir veri yapısı gösterilmiştir.

```

<npcs>
  <npc id="100">
    <sprite variant="0">npc1.xml</sprite>
  </npc>
  <npc id="132">
    <sprite variant="5">npc1.xml</sprite>
    <particlefx>graphics/particles/p1.xml</particlefx>
  </npc>
</npcs>

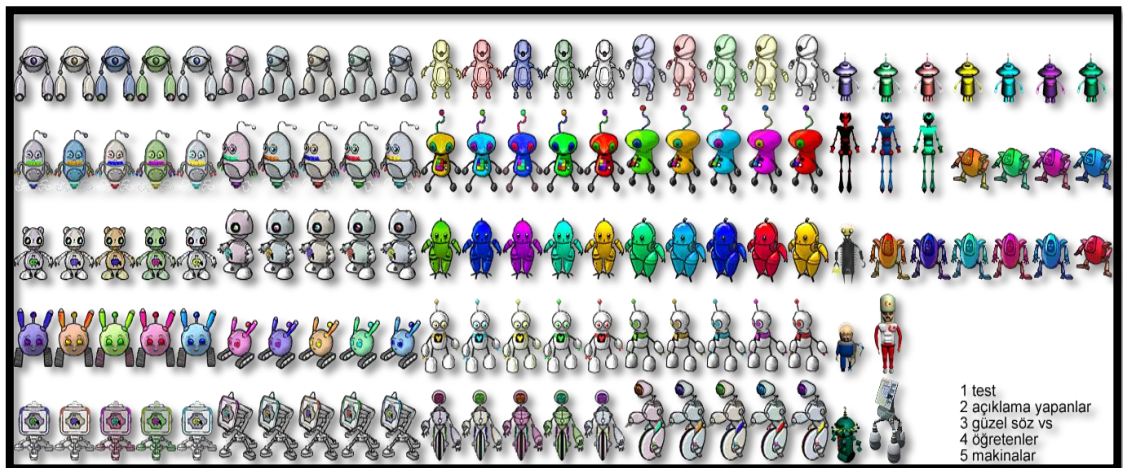
```

Şekil 4.71: npcs.xml dosyasına örnek.

Örnekte bilgenin id'sinin verildiği NPC tag'ı yeni bir bilge türü tanımlamak için kullanılmaktadır. npc tag'ları arasında kullanılan sprite tag'ı bilgenin görüntüsüne ait bilgilerin bulunduğu XML dosyasına işaret eder. variant özelliği ise bilgenin işaret edilen dosyada kaçınıcı sırada tanımlanan resim olduğunu ifade etmek için kullanılır.

Bilgelerin etrafında ışık, duman gibi efektler kullanılması gerekebileceği için Bölüm 4.3.1.9'da bahsedilen parçacık efektin tanımlandığı dosyaya gönderme yapmak için particlefx tag'ı npc tag'ları arasına eklenebilir.

elektroGame'de kullanılan bilgiler Şekil 4.72'de verilmiştir. Bu bilgelere ulaşmak için kullanılan npcs.xml'de npc tagı içinden işaret edilen XML veri ise Şekil 4.73'te sunulmuştur.



Şekil 4.72: elektroGame'de kullanılan bilgiler.

```
<sprite variants="27" variant_offset="1">
  <imageset name="base"
    src="graphics/sprites/npcs1.png"
    width="50"
    height="80" />
  <action name="stand" imageset="base">
    <animation direction="default">
      <frame index="0" />
    </animation>
  </action>
</sprite>
```

Şekil 4.73: Sprite üzerindeki npc bilgilerini tanımlayan veri.

Sprite içinde kullanılan variants özelliği, npc serisini oluşturan sprite'da kullanılabilir kaç adet bilge resmi olduğunu göstermektedir. animation özelliği ise bu seri için kullanılacak olan sprite için gerekli özellikleri tanımlamakta kullanılır. elektroGame'de animated bilge kullanılmamıştır ancak yapı animated bilgiler oluşturmak için hazır hale getirilmiştir. Bundan sonraki sürümlerde uygun sprite set'ler hazırlanarak bu özellikten yararlanılabilir.

#### 4.3.1.12 Robot Özellikleri

MMORPG oyunlarda oyuncunun mücadele edeceği karakterler oyunun temasına göre değişiklikler gösterebilmektedir. elektroGame'de Bölüm 4.2.1.3'te anlatılan robotlar, oyuncunun geri dönüşüne göndermek için çalışacağı ve bunu başardığında çeşitli açılardan ödüllendirileceği oyun öğeleridir. Nesnelerde olduğu gibi, robotlarda da hem istemci tarafında tanımlanması gereken grafik, ses gibi özellikler hem de sunucu tarafından tanımlanması gereken tecrübe puanı, vereceği hediyeler gibi özellikler mevcuttur. elektroGame'in genel hiyerarşisine uygun olarak istemci tarafında robotların özelliklerini tanımlayan XML biçiminde tanımlanmış veriler bulunmaktadır. Sunucu tarafında ise eAthena'nın kullandığı metin tabanlı veri yapısı kullanılmaktadır.

İstemcide robotları tanımlamak için kullanılan temel veri dosyası, monster.xml dosyasıdır. Şekil 4.74'te Afacan Kırmızı Robot için kullanılan yapı gösterilmiştir.

```
<monster id="0" name="Afacan Kirmizi Robot" targetCursor="small">
  <sprite>elektroadd/robot_afacan_kirmizi.xml</sprite>
  <sound event="hit">monsters/maggot/maggot-hit1.ogg</sound>
  <sound event="hit">monsters/maggot/maggot-hit2.ogg</sound>
  <sound event="miss">monsters/maggot/maggot-miss1.ogg</sound>
  <sound event="die">monsters/maggot/maggot-dying1.ogg</sound>
  <drop item="505" percent="8"/>
  <drop item="518" percent="4"/>
  <drop item="501" percent="1.5"/>
  <drop item="533" percent="1.5"/>
  <drop item="502" percent="0.7"/>
  <drop item="522" percent="0.1"/>
```

```
<attributes
  hp="20"
  size="4"
  speed="64"
  attack-min="10"
  attack-delta="2"
  attack-magic="0"
  hit="10"
  evade="5"
  physical-defence="5"
  magical-defence="0"
  mutation="50"
/>
```

```
<vulnerability element="fire" factor="1.5"/>
<vulnerability element="earth" factor="0.7"/>
<exp>10</exp>
<!-- average stroll- and track range-->
<behavior
  aggressive="false"
  cowardly="false"
  track-range="5"
  stroll-range="2"
  attack-distance="32"
/>
```

```
<attack id="1"
  priority="1"
  type="physical"
  pre-delay="10"
  aft-delay="5"
  damage-factor="1"
  range="32"
  animation="attack"
/>
</monster>
```

Şekil 4.74: Monster.xml dosyasında kullanılan yapı.

Monster.xml’de kullanılan tag’lar ve özellikleri **Tablo 4.14**’te verilmiştir.

**Tablo 4.14:** Monster.xml’de kullanılan tag’lar ve özellikleri.

monster	id: Robotun ayırt edici numarası name: Robot hedef alındığında görünecek ismi targetcursor: Robot hedeflendiğinde görünecek fare ikonunun boyutu small: Küçük medium: Orta large: Büyük
sprite	Robotun grafiksel görünümü için gerekli olan resim dosyası
sound	hit: Robot saldırdığında çıkacak ses miss: Robotun yapacağı saldırıyı kaçırmaması durumunda çıkacak ses hurt: Robot darbe aldığıda çıkacak ses die: Robot geri dönüşüme gönderildiğinde çıkacak ses
drop	id: Robot geri dönüşüme gönderildiğinde düşüreceği nesne percent: Robot geri dönüşüme gönderilirken nesnelere hangi oranda düşüreceği
attributes	hp: Robotun gücü size: Robotun etki alanının boyutu speed: Robotun hareket hızı attack-min: Robotun en küçük saldırı gücü değeri attack-delta: Robotun en küçük ve en büyük saldırı gücü değerleri arasındaki fark hit: Robotun hangi çabuklukta saldırı yapabileceği evade: Robotun kendisine gelen saldırılardan kaçabilme seviyesi physical-defence: Robotun normal saldırılara karşı savunma gücü
vulnerability	element: Robotun zafının olduğu elementin adı factor: Robotun bu elementten alacağı saldırı sonucu kaybedeceği sağlık değerinin çarpılacağı değer
exp	Robot geri dönüşüme gönderildiğinde kazanılacak deneyim puanı
behavior	aggressive: Robotun saldırı gelmeden saldırı yapacağını yada yapmayacağını belirler cowardly: Robotun saldırıp kaçarak daha akıllıca saldırıp saldırmayacağını belirler track-range: Robotun oyuncuyu ne kadar mesafe takip edeceğini belirler stroll-range: Robotun saldırı olmayan zamanlarda gezebileceği yerlerin mesafesi attack-distance: Robotun oyuncuya saldıracağı mesafe
attack	id: Saldırının ayırt edici numarası priority: Robotun birkaç saldırı tipinde bu saldırıyı seçme önceliği type: Saldırının tipi (fiziksel yada sihir) pre-delay: Robotun saldırıya karar verdikten sonra saldırıya başlaması ile aradan geçen zaman aft-delay: Robotun saldırdıktan sonra tekrar saldırabilmesi için geçmesi gereken süre damage-factor: Yapılan saldırının gücünün çarpılacağı değer range: Saldırının yapılabileceği mesafe animation: Robotun bu saldırıyı kullandığında oynatılacak animasyon particle-effect: Robotun bu saldırıyı kullandığında gösterilecek parçacık efekti

Sunucu tarafından tanımlanan robotların özellikleri, görüntü gibi istemciyi ilgilendiren özellikler yerine, robotların hareketlerine ve oyuncu üzerindeki etkilerine odaklanmıştır. eAthena tarafından kullanılan mob\_db.txt dosyasında yer alan

özellikler, **Tablo 4.15**'da verilmiştir. Buradaki bazı özellikler, elektroGame içinde kullanılmamıştır. Ancak ilerleyen versiyonlarda sunucu üzerinde değişiklik yapılmadan yeni özellikler eklenerek oyunun zenginleştirilmesi mümkündür.

**Tablo 4.15:** mob\_db.txt özellikleri.

ID	Robotun ayırt edici numarası
Name	Robotun adı
Jname	Robotun sunucuda görünen ismi
LV	Robotun seviyesi
HP	Robotun sağlık puanı
SP	Robotun dayanıklılık puanı
EXP	Robotun vereceği deneyim puanı
JEXP	Robotun görev sırasında vereceği deneyim puanı
Range	Robotun saldırı yapabileceği mesafe
ATK1,ATK2	Robotun en düşük saldırı gücü - Robotun en yüksek saldırı gücü
DEF	Robotun savunma gücü
STR	Robotun gücü
AGI	Robotun çevikliği
VIT	Robotun dayanıklılık seviyesi
INT	Robotun zekâ seviyesi
DEX	Robotun beceri seviyesi
LUK	Robotun şans seviyesi
Scale	Robotun görüntülenme ölçeği
Race	Robotun cinsi
Mode	Robotun davranışlarını tanımlar
Speed	Robotun yürüme hızı
Adelay	Robotun saldırı gecikmesi
Amotion	Robotun saldırısı sırasında gösterilecek animasyon
Dmotion	Robotun hasar aldığıda gösterilecek animasyon
Drop1id..8	Robot geri dönüşüme gönderildiğinde düşecek nesnenin id numarası
Drop1per..8	Robotun geri dönüşüme gönderildiğinde düşecek olan nesnelere düşme oranı

**Tablo 4.15**'in devamı.

Item1,2	Robotun kullanabildiği nesnelere
MEXP	Robot geri dönüşüme gönderildiğinde en değerli oyuncuya verilecek deneyim puanı
MVP1id..3	Robotun en değerli oyuncuya düşürdükleri nesne
MVP1per..3	Robotun nesneyi düşürme oranı

### 4.3.2 Sunucu

Çevrim içi oyunlarda sunucu, oyunun en önemli parçasını oluşturmaktadır. İstemcide meydana gelebilecek bir istisna (exception) hata, sadece bir oyuncuyu etkilemekteyken, sunucuda oluşan bir hata tüm oyuncuları etkileyecek sonuçlar doğurabilmektedir. Bu yüzden, elektroGame sunucusu kaynaklı sorunların oluşmaması için sunucu üzerinde oldukça fazla deneme yapılmıştır.

Açık kaynak kodlu (GPL) olarak lisanslanmış Ragnarok isimli oyun için kullanılan eAthena sunucusu, pek çok RPG projesinde sunucu olarak kullanılmaktadır. elektroGame sunucusu olarak da eAthena 1.0 TXT sürümü kullanılmaktadır. elektroGame'in temelini oluşturan The Mana World oyununun bu sunucuyla çalışacak şekilde yazılmış olması, uyumluluk problemlerini azaltacağı için özellikle tercih edilmiştir. Ancak bu sürümün dayanıklılık, esneklik gibi avantajlarının yanı sıra, veri tabanı ve web sitesi ile entegre çalışmama gibi önemli eksiklikleri bulunmaktadır. Bahsedilen problemleri giderebilmek için sunucu üzerinde pek çok değişiklik yapılmıştır. Ayrıca TMW ekibi de sunucunun eksikliklerinden dolayı yeni bir sunucu yazmaya başladığını duyurmuştur.

elektroGame'in oynanabilmesi için gerekli olan sunucu programları, kaynak tüketme konusunda dikkatli bir şekilde yazılmış olsalar da aynı anda çevrim içi olan oyuncu sayısı arttığında yüksek performanslı bir bilgisayara ihtiyaç duyulmaktadır. Ayrıca Bölüm 4.3.3'te özellikleri anlatılan web sitesi için ayrı bir sunucu olmadığından, web sunucu ve oyunun sunucusu aynı bilgisayarı paylaşmaktadır. Sunucu olarak kullanılan bilgisayarın donanım özellikleri **Tablo 4.16**'da verilmiştir.

**Tablo 4.16:** elektroGame sunucusunun donanımsal özellikleri.

Model	Sun Fire x4100
İşlemci	AMD opteron 2000 64 Bit 2x Dual Core
Bellek	2x 2GB
HDD	35GB SAS

İşletim sistemi olarak Ubuntu 9.10 Carmic kurulmuştur. Bu işletim sisteminden önce işlemciyi daha verimli kullanabilmek için 64 bit işletim sistemleri denenmiş ancak GCC'nin (GNU Compiler Collection) 64 bit derleme problemleri nedeniyle vazgeçilmiştir. Ubuntu'nun başarılı paket yönetimi, derleme sırasında ihtiyaç duyulan paketleri otomatik olarak sağladığı için Windows işletim sistemlerinde ortaya çıkan pek çok kütüphane sorununu ortadan kaldırmıştır.

Web sunucu olarak Apache 2.2.12, veri tabanı yönetim sistemi olarak MySQL 5.1.37 ve betik dili olarak da PHP Version 5.2.10-2 kullanılmıştır.

Uzak bilgisayarda yapılan çalışmalarda ağ üzerinde bulunmanın getirdiği yavaşlama, bağlantı kopması gibi bazı sıkıntılar yaşanabilir. elektroGame'in geliştirilmesi sırasında bu sorunun üstesinden gelebilmek için sanal bilgisayar (virtual box) üzerine sunucu kurulmuştur. Bunun için sanal bilgisayara Pardus 2008.2 işletim sistemi kurulmuş ve sunucuya ait kaynak dosyalar SVN'den indirilerek gerekli ayarlamalar yapılmıştır. Böylece gerektiğinde sanal bilgisayara ait kalıp dosyası kopyalanarak sunucunun diğer bilgisayarlar üzerinde kolaylıkla çalıştırılabilmesi mümkün olabilmektedir.

Sunucuda çalışan web siteleri birden fazla olduğu için web sunucu olarak kurulan Apache'de çeşitli ayarlamalar ve tanımlamalar yapılmıştır. Örneğin, <http://forum.elektrogame.net> bağlantısı ile web sunucuya bağlanan istemci, doğrudan elektroForum'a yönlendirilmektedir. Bu yönlendirmeler için **Şekil 4.75**'teki gibi virtualhost tanımlamaları yapılmıştır. Ayrıca blog, sunucu, görev, güncelle ve wiki içinde bu tür tanımlamalar yapılmıştır.



```
<VirtualHost 193.255.184.205>
  DocumentRoot /var/www/forum
  ServerName forum.elektrogame.net
  <Directory /var/www/forum>
    Options Indexes FollowSymLinks MultiViews +Includes
    AllowOverride All
    Order allow,deny
    allow from all
  </Directory>
</VirtualHost>
```

**Şekil 4.75:** Virtual Host tanımlamaları.

### 4.3.2.1 Login sunucu

Login sunucunun görevi, sunucuya bağlanmak isteyen istemcilerin doğru kullanıcı adı ve parolayı girip girmediğini kontrol etmektir. Bu amaçla veriler athena.txt’den çekilir ve istemciden gelen veriyle karşılaştırılır. “athena.txt” dosyası, program tarafından bir kez okunduktan sonra, ayar dosyalarında belirtilen sürelerde yapılan değişikliklerin tekrar diske yazılmasını sağlar. Bu durum, oyuna kayıt olmak için istemcinin kullanılmasını zorunlu kılmaktadır. Ancak istemciden alınan bilgilerin türü ve gönderilen veri miktarı sınırlıdır. Bu yüzden araştırmacı tarafından sunucunun verileri athena.txt’den alması ve buraya kaydetmesi engellenerek MySQL üzerinde tutulan bir veri tabanında login isimli bir tabloya yönlendirilmiştir. Böylece hem oyuncular web sitesi üzerinden kayıt olabilmekte hem de web sayfası için yazılmış program kolaylıkla oyundaki son bilgilere ulaşabilmektedir.

Sunucu başlatıldığında, ilk olarak çalışma ayarlarının bulunduğu login\_athena.conf dosyasını okur. Bu dosya, tipik bir ikili anahtar-değer listesi şeklinde tasarlanmıştır. Özelliğin karşısına iki nokta (:) işareti konarak değeri belirtilmektedir. Sunucunun ayarlanması sırasında özellikle üç önemli değer vardır. **login\_port** özelliği istemcinin sunucuya isteklerini hangi port üzerinden göndereceğini tanımlar. Bu port, özel olarak dinlemeye alınır ve gelen her mesaj login sunucu tarafından değerlendirilir. admin\_state ve admin\_pass özellikleri login sunucuya uzaktan admin olarak erişilip erişilmeyeceğini ve erişilecekse erişimin parolasını bildirir. Diğer değerler varsayılan ayarlarında bırakılabilir.

Login tablosunda kullanıcı adı, parola gibi bilgilerin yanı sıra son girişin yapıldığı ip, kaçınca giriş olduğu ve saat kaçta girildiği gibi güvenlik amaçlı veriler

de saklanmaktadır. elektroGame'in uygulaması sırasında kullanılmayan ancak geniş kapsamlı kullanıma açıldığında istenmeyen oyuncuları uzak tutmak amacıyla cezalı, kovulmuş adaylar hem IP'leriyle hem de kullanıcı adlarıyla bu tabloda saklanabilmektedir.

Oyuncu, istemcinin kullanıcı girişi ekranında (login screen) kullanıcı adı ve şifresini girdiğinde LoginHandler sınıfı sendLoginRegister() fonksiyonu ile 0x0064 başlıklı bir paketi sunucuya gönderir. Login sunucu çalışmakta olduğu IP üzerinde, ayar dosyalarında tanımlanmış olan porttan gelen paketleri parse\_login fonksiyonuna yönlendirir. Bu fonksiyon, istemciden gelmesi muhtemel bütün paket başlıkları için yapılması gerekenleri icra etmek üzere tasarlanmıştır. 0x0064 başlıklı paket çözümlendikten sonra veri tabanında login tablosundan kontrol edilir. Kullanıcı adı ve parola doğru şekilde girilmişse oyuncu için bir oturum açılır ve tablodaki diğer veriler okunarak oturumda kayıt altına alınır.

Kullanıcı adı ve parola doğru şekilde girilmediyse ya da oyuncunun giriş yapmasını engelleyecek başka bir problem bulunursa SMSG\_LOGIN\_ERROR (0x006a) başlıklı paket ile uygun hata mesajı istemciye iletilir. İstemci gelen mesajı yine LoginHandler aracılığıyla çözümlenerek görselleştirilmek üzere login penceresine iletir.

Oyuncun giriş süreci tamamlandığında yani sunucudan 0x0069 başlıklı paket geldiğinde oyuncuya ait account\_id, sessiond\_id gibi kullanıcı bilgilerini arkasından da sunucu adı, çevrim içi kullanıcı sayısı gibi sunucuya ait bilgileri çözümler. Çözümleme tamamlandıktan sonra istemci güncellemeleri kontrol eder. Güncelleme bilgi penceresi kapatılınca oyuncunun kendisine ait char'lerden birini seçebilmesi ya da yeni bir char oluşturabilmesi için karakter seçim penceresi açılır.

Login sunucu, oyuncu bir kez giriş yaptıktan sonra yasaklanarak oyundan atılma yönünde kayıt altına alınması durumu hariç, diğer sunucu ve istemciler ile paket alışverişi yapmaz. Bu yüzden bazı oyunlarda char sunucu ile login sunucu birleştirilerek yazılır.

### 4.3.2.2 Char Sunucu

Char sunucu, oyuncuların char'larına ait bilgilerin kontrol edilmesinden sorumludur. elektroGame'de oyuncuların sadece bir char'ı olacak şekilde kısıtlama yapılmıştır ancak char sunucu her oyuncuya üç char'a kadar destek verebilmektedir.

Char sunucu, login sunucuda olduğu gibi, bir ayar dosyasını okuyarak çalışma parametrelerini ayarlar. char\_athena.conf adlı bu dosyanın önemli parametreleri **Tablo 4.17**'de açıklanmıştır.

**Tablo 4.17:** Char sunucunun önemli ayar parametreleri.

userid	Sunucular arası haberleşme için özel olarak tanımlanan kullanıcının adı.
passwd	Sunucular arası kullanıcının parolası.
login_ip	Login sunucusunun IP'si.
login_port	Login sunucusunun portu.
char_ip	Char sunucusunun IP'si.
char_port	Char sunucusunun portu.
start_point	Yeni bir char'ın hangi haritada ve hangi koordinatlarda başlayacağı.
start_weapon	Yeni char'ın başlangıç silahının id'si.
start_armor	Yeni char'ın başlangıç kıyafeti.
start_zeny	Yeni char'ın parası.

Char sunucu çalışmaya başladığında, tanımlanan IP ve port üzerinden login sunucu ile iletişim kurmaya çalışır. İletişimin başarılı olması halinde, login sunucuya normal bir oyuncu gibi kullanıcı adı ve parola gönderir. Ancak gönderilen kullanıcı, login tablosunda oyuncuların cinsiyetlerinin belirtildiği alanda "S" olarak tanımlanmış özel bir kullanıcı olduğu için login sunucu tarafından yönetici ayrıcalıkları verilerek sisteme kabul edilir.

Bir oyuncu, login sunucu tarafından onaylanarak sisteme giriş yaptıktan sonra, char sunucu devreye girerek eğer varsa oyuncuya ait ilk char'ın özelliklerini istemciye gönderir. Bu özellikler 0x006b başlığını taşırlar ve istemcide **charserverhandler** tarafından çözümlenirler. 0x006b paketi sunucu ve istemcinin char'ı doğru tanımlayabilmesi için hayati öneme sahiptir. eAthena'nın ve TMW'nin farklı sürümlerinde farklı tanımlar yapılabildiği dikkate alınarak uyumlu sürümlerin

kullanılması gerekmektedir. elektroGame, 106 Byte uzunluğunda paket yapısı kullanmaktadır.

Seçilen char'ın özellikleri veri tabanında char, inventory, guild, global\_reg\_value gibi tablolara bölünmüş olarak saklandığı için sırayla uygun tablodan verileri okuyarak açılan oturuma yapılar (struct) halinde kaydeder. Bölüm 4.2.2.2'de bahsedildiği gibi, elektroGame'in oyuncuların testlere verdiği yanıtlar gibi çok sayıda girdiyi saklaması, uygulama aşamasında bazı değişkenlere erişememeye neden olmuştur. Problemin, C programlama dilinin yapısal olarak 4096 Byte'tan fazla yer kaplayan dizilere izin vermemesinden kaynaklandığı anlaşılmıştır. Problemi aşabilmek için sunucunun konvansiyonel çalışma prensibi değiştirilerek fazla yer kaplayan verileri, yapılardan oluşan diziler yerine, doğrudan veri tabanından çekmesi ve kaydetmesi sağlanmıştır.

Char sunucu açılışını tamamladıktan sonra, oyuncuların giriş yapabilmesi için oyunun işlemesinden sorumlu olan map sunucunun yetkili kullanıcı olarak kendisine talep göndermesini bekler. Talep geldikten, kullanıcı adı ve parola onaylandıktan sonra, belirtilen porttan oyuncu taleplerini dinler.

### **4.3.2.3 Map Sunucu**

Map sunucu, eAthena'nın en önemli ve sistem kaynaklarını en fazla tüketen parçasıdır. Dünyalar üzerinde bulunan robotların, bilgelerin ve oyuncuların etkileşimini yönetmekten sorumludur. Dünyalar çok geniş bir alana yayılabileceği için bir sunucu sisteminde birden fazla map sunucu çalıştırmak mümkündür. Aynı şekilde, char sunucuda çok fazla oyuncunun sistemde olması durumunda, birden fazla sunucu çalıştırılabilir. Performansı arttırmak için ek sunucuların farklı makineler üzerinde olması da mümkündür.

Diğer iki sunucuda olduğu gibi map sunucu da, çalışması için gerekli olan değerleri ikili liste şeklinde hazırlanmış bir text dosyadan alır. map\_athena.conf isimli dosyanın başlangıç kısmı char\_athena.conf dosyasına benzer özellikler gösterir. Char sunucunun ve kendisinin IP'si ve portu burada tanımlıdır. Farklı olarak oyunda kullanılan haritalar, bilgiler ve haritaya ait değişkenler burada yüklenir.

Dosya bu şekliyle çok büyük ve karmaşık olacağı için düzenlemenin zor olmaması amacıyla, bu özellikler ayrı bir dosyaya yazılmış ve *import* komutu ile dosyanın adı ve yolu belirtilmiştir.

map\_conf.txt dosyası map anahtar kelimesinin karşısına kullanılacak olan dünya isimlerinin yazıldığı bir liste şeklindedir. Burada tanımlanan haritalar Bölüm 4.3.1.8’de anlatılan WLK formatında kaydedilmiş dosyalardır. eAthena bir iç kontrol mekanizması olarak harita isimlerini bir dönüşüm yaparak kullanmaktadır. İç isimlerin tanımlandığı dosya, data dizinin altındaki resnametable.txt dosyasıdır. Burada iç ismi belirtilmemiş bir harita map sunucuda kullanılamaz.

Oyunda kullanılan bilgelerin yeri ve yapacakları da map sunucu tarafından kontrol edilmektedir. Eklenecek olan bilgiler, map\_athena.conf dosyasına farklı dosyalardan import edilerek daha düzenli bir yerleşim yapılmıştır. Bu dosyalara npc anahtar kelimesinin karşısına npc’yi tanımlayan script dosyasının tam yolu ve adı yazılır. elektroGame’de npc’ler yer aldıkları dünyaların ismiyle açılmış klasörlere konulmuştur. Ayrıca ilk harfleri yaptıkları göreve uygun olarak verilmiştir. Örneğin, *npc/oda1\_12/t\_doktor.txt* dosyası devre dünyasında 12 numaralı binada bulunan test bilgesinin tanımını ve script blokunu belirtir.

eAthena’da ışınlanma noktaları ve robotlar da map\_athena.conf dosyasında yer alırlar. Bilgeler gibi bu script’ler de import edilmek üzere ayrı dosyalarda tutulmuş ve buldukları dünyanın klasörüne robot.txt ve gecit.txt kaydedilmiştir. **Şekil 4.76**’da bir robot ve bir geçit noktası eklemek için kullanılan kod satırları verilmiştir.

```
elektro_1-1.gat,111,88,160,80 monster robot1 1018,18,1000,0,{}  
elektro_1-1.gat,162,92 warp gecit1 0,0,elektro_1-8.gat,21,26
```

**Şekil 4.76:** Robot ve geçit noktası ekleyen kod.

Robot eklemek için kullanılan kodun 12 parametresi bulunmaktadır. Sırasıyla parametrelerin görevleri şu şekildedir:

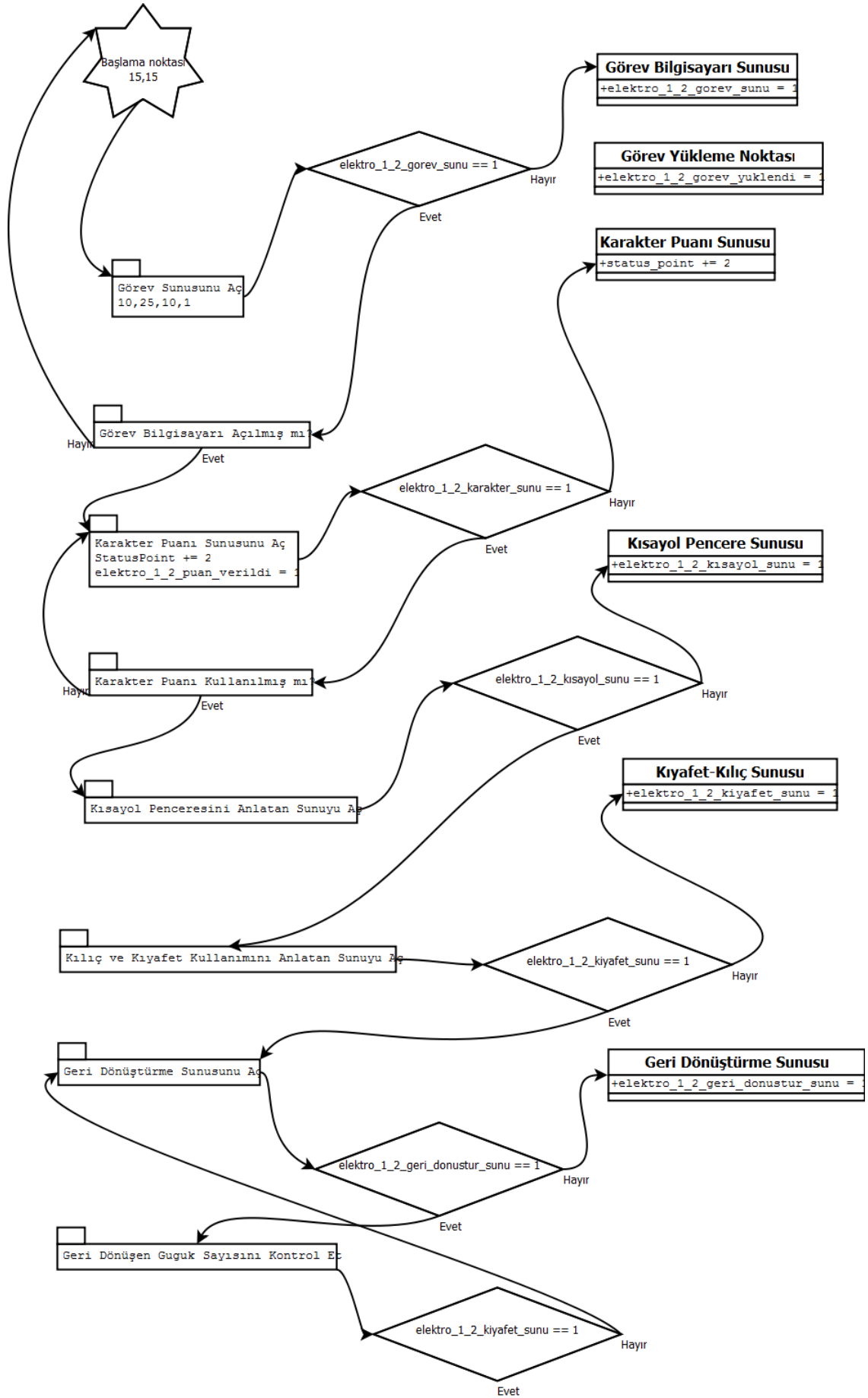
1. Robotların yerleştirileceği dünyanın iç adı,
- 2-3. Robotların yerleşim merkezinin x ve y koordinatları,
- 4-5. Verilen koordinatlardan kaç birim uzaklaşabileceği,
6. Verilen koordinata robot yerleştirileceğini belirten anahtar ifade,
7. Robotun görünen adı,
8. Robotun monster\_db.txt dosyasına tanımlanan id’si,

9. Robot sayısı,
- 10-11. Yeniden ortaya çıkma süresi,
12. Olay script'i.

Geçit komutu ise 10 parametreden oluşur. Bu parametrelerin görevleri şu şekildedir:

1. Geçit yerleştirileceği haritanın iç adı,
- 2-3. Geçidin x, y koordinatları,
4. İfadenin bir geçit olduğunu belirten anahtar kelime,
5. Geçidin adı,
- 6-7 Geçidin genişlik ve yüksekliği,
8. Hedef haritanın iç adı,
- 9.-10 Hedefin koordinatları.

Oynanabilirliği yüksek bir oyun ortaya çıkartabilmek için yukarıda belirtilen özellikleri dengeli ve doğru bir şekilde kullanmak gerekmektedir. elektroGame dünyaları tasarlanırken her biri için geçitler, robotlar, bilgiler ve değişkenleri gösteren diyagramlar hazırlanmıştır. **Şekil 4.77**'de Eğitim Dünyası'nın diyagramı gösterilmiştir.



Şekil 4.77: Eğitim Dünyası'nın yapısal diyagramı.

#### 4.3.2.4 eAthena Script

Sunucunun bilgeleri, robotları ve nesneleri eklemek için kullandığı script (betik) dili eAthena'dır. Bu dilin yapısının anlaşılması sunu, test, devre ve görev bilgilerinin hazırlanması için gereklidir. Bu bölümde eAthena dilinin en temel kuralları ve elektroGame'de çok sık kullanılan fonksiyonların görevleri açıklanacaktır.

**Bilge Eklemek:** Bilge eklemek için map\_athena.txt'de tanımlanan dosyaya `elektro_1-16.gat,20,21,0 script Tyrex 323,{}` yapısına uygun bir script başlığı eklenir. Başlıkta dünya adı, x koordinatı, y koordinatı, bakış yönü tanımlandıktan sonra script anahtar kelimesi eklenir. Daha sonra bilgenin adı ve id'si belirtilir. Küme parantezleri arasına ise bu bölümde kuralları verilen script kodu yazılır. Başlık ve script Map sunucunun script.c dosyası tarafından çözümlenir ve çözümlene kuralları esnek değildir. Örneğin, başlıkta uzun boşluklarla verilen aralıklar tab ile yapılmıştır. Arada boşluk vb. karakterler bulunması çözümlene hatalarına neden olur. Geçit ve robot ekleme **Şekil 4.76**'da açıklandığı gibi bilgelere benzer bir yapıdadır.

**Değişkenler:** Değişken yapısı çok gelişmiş değildir. Bu yüzden eAthena'nın ileri sürümlerinde en çok üzerinde durulan konulardan birisi olmuştur. Kullanılan sürümde temel olarak geçici ve kalıcı olarak iki tür değişken vardır. Değişkenler yalnızca integer (tam sayı) değerler alabilirler. Geçici değişkenler tanımlandıkları küme parantezinin kapatılmasıyla birlikte hafızadan atılırlar. Başlarındaki @ işareti ile geçici oldukları belirtilir. Kalıcı değişkenler ise elektroGame için yeniden düzenlenerek veri tabanında **global\_reg\_value** tablosunda saklanacak şekilde dönüştürülmüştür. Ayrıca, sadece içinde bulunan dünyayı etkileyecek map\_reg\_value türü tasarlanmış ancak uygulamada kullanılmamıştır.

Değişkene değer atamak için **set** anahtar kelimesi kullanılır. Kullanımı set değişken, "değer" şeklindedir. Değeri sıfır olarak atanan bir değişken saklanmaz ve hafızadan atılır. Eğer kalıcı bir değişkense global\_reg\_value tablosundan silinir. Bu yüzden bir unset fonksiyonu yoktur. Değeri sıfır olan bir değişken saklanamadığı için (örneğin, bir testte hiç boş bırakılmamışsa boş cevap değişkeni) kontroller sırasında



tabloda bulunamayan deęişkenler için özel önlem almak gerekmektedir. elektroGame geliştirilirken bu noktalara özellikle dikkat edilmiştir.

elektroGame script'lerinde dizi (array) kullanmanın sunucuyu kararsız hale getirdiđi fark edildiđi için script'lerde diziler kullanılmamıştır. eAthena'da dizi kullanımı için birkaç fonksiyon bulunsa da tercih edilmemelidir.

**Label ve Goto:** Script içinde etiketler tanımlayarak gerektiğinde etikete atlama yapmak için kullanılır. Yapısal (fonksiyonel) programlamada tavsiye edilen bir yöntem olmasa da script dilleri için kolay bir çözüm sunar. elektroGame'de jargon olarak etiketler büyük harfle yazılırlar ve tanım geređi iki nokta ile biterler. Aynı küme parantezi içinde yazılan blokta istendiđi zaman **goto** ile tanımlanan etikete geçiş yapılabilir. Etiket, **goto** komutundan sonra olsa bile ön tanımlama gerektirmez.

**Koşul ve Döngüler:** Diğer dillerde de olduđu gibi koşul operatörü “**if**” olarak kullanılmıştır. Yazım kalıbı C jargonuna benzer. if (şart) goto <label>; biçiminde kullanılır. Bir koşul ifadesinden sonra sadece **goto** komutu kullanılabilir. Bu kullanım şekli script'lerin gereksiz yere uzamasına neden olmakla birlikte mevcut sürümde başka bir çözüm yolu yoktur.

Döngü komutları 1.0 versiyonunda tanımlanmamıştır. Onun yerine döngüler koşul ve etiket kullanılarak yapılabilir. **Şekil 4.78**'de sadece yöneticilerince girilen, ihtiyaç duyulan tüm nesnelere veren ve silen kodun döngü oluşturan kısmı verilmiştir.

```
set @silinecekitem,1100;
LOOP4:
delitem @silinecekitem,1;
set @silinecekitem,@silinecekitem + 1;
if (@silinecekitem<1200) goto LOOP4;
goto MENU_;
```

**Şekil 4.78:** Döngü olarak kullanılan yapı.

**Fonksiyonlar:** Fonksiyonlar sık sık kullanılan kodların yeniden yazılmasını gereksiz kılan ve kodun güvenilirliğini arttıran yapılardır. eAthena'da fonksiyonlar, diğer tanımların aksine aynı dosyada ve küme parantezleri arasında olmak zorunda değildir. Fonksiyon tanımı **function script fonk\_adi {}** şeklindedir. Uzun aralıklar tab

işareti ile yapılmaktadır. Fonksiyon çağrısı ise diğer dil yapılarından çok farklı olarak `callfunc "fonksiyon_adi", <parametre1>, <parametre2>, ..., <parametreN>;` biçimindedir.

Gönderilen parametreler fonksiyon içinde `getarg(n)` fonksiyonu ile alınır. Bu yüzden parametrelerin gönderilme sırası önem taşımaktadır. Gönderilen parametreler içyapıda bir değişkende saklandığı için tam sayı dışında değer alamamaktadır. Fonksiyon yapısına örnek, ileriki konularda özel fonksiyonlar açıklanırken verilecektir.

**İstemciye Mesaj Gönderme:** Sunucu istemci tabanlı dillerde genellikle çıktı veren komutlar çıktıyı ekran gibi ara birimler yerine istemciye gönderirler. eAthena'da da `mes` komutu istemciye çıktı göndermek için kullanılır. Komutun yapısı `mes "mesaj 1"+"mesaj 2"+ "mesaj N"`; biçimindedir.

**Hazır Fonksiyonlar:** elektroGame'deki bilgelerin en sık kullandığı fonksiyonlar ve görevleri **Tablo 4.18'**da verilmiştir (eAthenawiki, 2010).

**Tablo 4.18:** Sık kullanılan eAthena fonksiyonları.

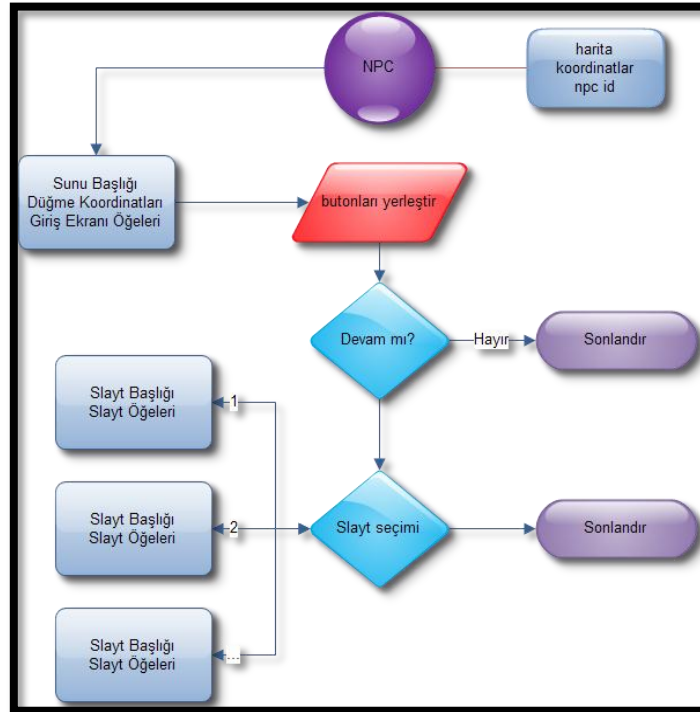
<code>heal &lt;hp&gt;, &lt;sp&gt;;</code>	Sağlık değerlerini mutlak olarak değiştirir.
<code>percentheal &lt;hp&gt;, &lt;sp&gt;;</code>	Sağlık puanlarını yüzde olarak değiştirir.
<code>statusup &lt;st&gt;;</code> <code>statusup2 &lt;st&gt;, &lt;n&gt;;</code>	Char'ı bir seviye yukarı çıkarır. Char'ın seviyesini istenen derece yükseltir ya da azaltır.
<code>bStr, bVit, bInt, bAgi,</code> <code>bDex, bLuk</code>	Dayanıklılık, enerji, zekâ, çeviklik, beceri, şans değerleri için sistem değişkenleri.
<code>rand(&lt;n1&gt;[,&lt;n2&gt;]);</code>	Belirtilen sınırlar içinde rastgele sayı üretir.
<code>countitem(&lt;itemid&gt;)</code>	ID'si verilen nesnenin sayısını döndürür.
<code>getitem &lt;itemid&gt;, &lt;num&gt;;</code>	ID'si verilen nesneden char'a vermek içindir.
<code>delitem &lt;itemid&gt;, &lt;num&gt;;</code>	ID'si verilen nesneyi char'dan alır.
<code>getexp</code> <code>&lt;base_exp&gt;, &lt;job_exp&gt;;</code>	Char'a tecrübe puanı verir.
<code>strcharinfo(&lt;n&gt;)</code>	Char hakkında istenen bazı parametreleri geri döndürür.

Çıktıları XML biçimli veri üretecek olan script'ler, programcı için fazlasıyla karmaşık bir yapı oluşturduğu için sadelik ve kullanım kolaylığını arttırmak amacıyla tüm bilgiler için kullanılacak olan fonksiyonlar tek bir dosyada (`\npc\function.txt`) toplanmış ve npc (bilge) listesi yüklenmeden önce içe aktarılmıştır.

**Menü:** Menü komutu C’de kullanılan switch komutuyla **goto** komutunun birleştirilmiştir bir varyasyonu sayılabilir. İstemciden gelen değer 1’den başlayan bir tam sayı olarak beklenir ve uygun sıradaki etikete gönderme yapar. **Şekil 4.82**’de sunu bilgilerinde kullanılan **menu** yapısı verilmiştir. elektroGame’de etiket isimlerinin başında bulunan metin ifade, sadece programcı için açıklama olarak kullanılmıştır. Ancak bu başlıklar da istemciye gönderilir. Eğer istenirse bu seçeneklerin istemcide bir menü oluşturması sağlanabilir.

#### 4.3.2.5 Sunu Bilgeleri

Bölüm 4.2.2.1’de anlatılan sunu bilgilerinin kullandığı XML biçimindeki verinin sunucu tarafından oluşturulabilmesi için eAthena script kurallarına göre çıktı veren betiklerin yazılması gerekmektedir. eAthena’nın 1.0 sürümünde, az sayıda fonksiyonu bulunan ve bazı sınırlılıkları bulunan script dili daha sonraki sürümlerde oldukça geliştirilmiştir. Ancak elektroGame’de uyumluluk problemleri nedeniyle eAthena sunucunun gelişmiş sürümlerini kullanmak mümkün olmamıştır.



**Şekil 4.79:** Sunu bilgilerinin akış diyagramı.

Diyagramda görülen bilgi koordinatları, ismi ve id’si yapısal olarak script’in başlığında belirtmek zorundadır. Daha sonra gönderilecek olan XML biçimindeki

verinin başlığı ve sunu penceresinin temel özellikleri gönderilir. Önceki bölümde anlatılan function.txt dosyasında tanımlanan fonksiyonlar yardımıyla sunu penceresinin koordinatları ve düğmelerin yerleri istemciye gönderilir. İstemciden sunuyu görmek isteyip istemediğinin cevabını alabilmek için iki seçenekli bir menü bu kodun sonunda yer alır. Akış, yapılan seçime göre uygun etikete yönlendirilir. **Şekil 4.80**'de sunu özelliklerinin XML biçimindeki verisini oluşturan script'in kodu verilmiştir.

```
elektro_1-8.gat,49,25,0 script Farabi 235,{
  set elektro_1_8_farabi_sunuc, 1;
  callfunc "presentationstarter";
  callfunc "presentationpro",50,50,410,450,7;
  set @x, 20;
  set @y, 0;
  callfunc "slidebutton","prev","show",@x,@y;
  set @x, @x + 40;
  callfunc "slidebutton","combo","show",@x,@y;
  set @x, @x + 160;
  callfunc "slidebutton","next","show",@x,@y;
  set @x, 360;
  callfunc "slidebutton","close","show",@x,@y;
  callfunc "image","5","40","graphics/sunular/farabi1/h_00.png";
  callfunc "presentationstarter_";
menu
  "quit",CANCEL,
  "start",s1;
```

**Şekil 4.80:** Sunu özelliklerini gönderen script.

Bölüm 4.3.1.2'de anlatılan elemanlar, slayt içinde farklı miktarlarda kullanılabilir. **Şekil 4.81**'de sadece bir resim içeren slaydın kodu görülmektedir. Kodun başında bulunan değişken ataması, yanlış cevap verildiğinde tekrar sorusunu göstermeyen bir bilgenin, bu slaydın görülmesiyle yeniden soru sormasına izin vermek için eklenmiştir. Böylece, oyuncuların sıkıştıkları yerlerde çeşitli hatırlatmalar ve ipuçları bulunan sayfaları okuduktan sonra yeniden şanslarını denemelerine imkân verilmektedir.

```
s1:
  set elektro_1_8_satx_yanlis , 0;
  callfunc "slide","1","Direnc";
  callfunc "image",5,40,"graphics/sunular/farabi1/h_01.png";
  callfunc "slide_";
  goto MENUSUB;
```

**Şekil 4.81:** Bir slaydı oluşturan script.

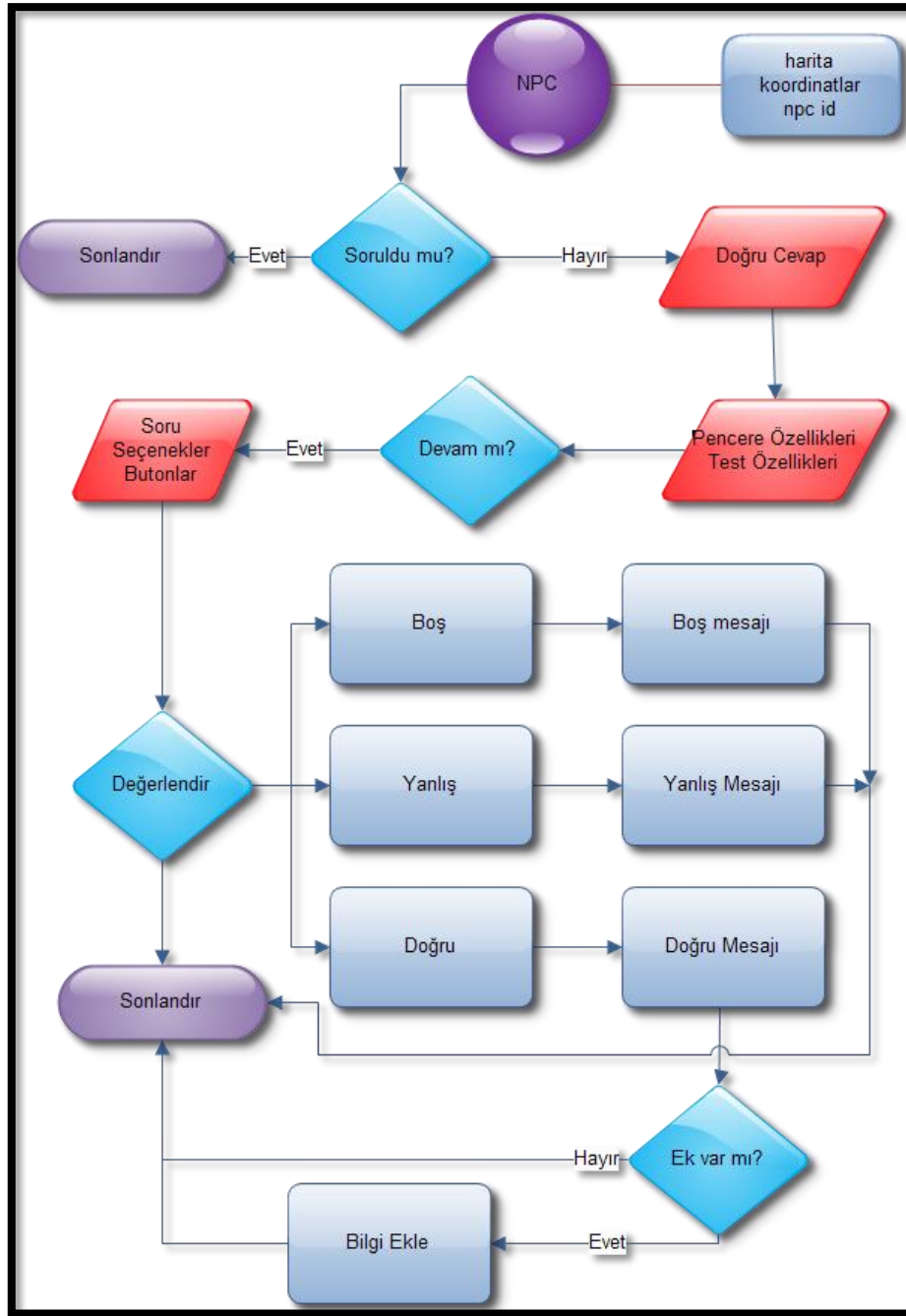
Oyuncu açılan kutu yardımıyla istediği slayda doğrudan gidebileceği gibi ileri geri düğmeleriyle de slaytlar arasında geçiş yapabilir. Bu etkileşimi sağlayabilmek için her slayt görsel bilgilerini gönderdikten sonra bir menü bloğuna gönderme yapar. Menüde istemciden gelen bilgiye göre yönlendirme yapılır. İlk iki seçenek, oyuncunun sunuyu iptal ederek çıkması ile sona gelerek çıkması arasında fark olması halinde programcıya yönetme imkânı sunmak için eklenmiştir. **Şekil 4.82**'de slaytlar arası geçiş yapmayı sağlayan menünün kodu verilmiştir.

```
MENUSUB:  
  menu  
  "iptal",CANCEL,  
  "sonlandir",SONLANDIR,  
  "1",s1,  
  "2",s2,  
  ...;  
  close;  
}
```

**Şekil 4.82:** Slaytlar arasında geçiş yapan menü script'i.

#### 4.3.2.6 Test bilgeleri

Test bilgilerinin üç tür soru yapısına sahip olduğu Bölüm 4.2.2.2 ve Bölüm 4.3.1.3'de açıklanmıştı. Sunucuda da XML veriyi oluşturan script üç farklı yapıdadır. Bu yapılardan ilki olan tek soru soran bilge oluşturmak için **Şekil 4.83**'te verilen yapıya uygun script yazılmalıdır.



**Şekil 4.83:** Tek sorulu testin diyagramı.

Sorunun ve test penceresinin özelliklerini tanımlayan başlık kısmına bir örnek **Şekil 4.84**'te verilmiştir. Başlıkta öncelikle gerektiğinde başlık kısmına dönebilmek için bir etiket tanımlanmıştır. Daha sonra sorunun doğru cevabı geçici bir değişkene aktarılarak aynı kodun diğer sorularda küçük değişikliklerle kullanılması sağlanmıştır. windowpro fonksiyonu pencerenin koordinatlarını ve genişlik, yükseklik değerlerini bildiren XML veriyi oluşturur. testpro fonksiyonu ise soru sayısı, süre, ödül ve ceza değerlerini tanımlar. Kodun sonundaki menü tek soruluk testlerde sadece istemciden gelecek paket alındı bilgisini beklemek için

kullanılmıştır. Veri transferinde herhangi bir sorun yaşanmadığı sürece test penceresi otomatik olarak iki değerini göndererek kodun devam etmesini sağlar.

```
elektro_1-2.gat,83,14,0 script SoranCan 100,{
Sor_1:
  set @correct,1; //doğru cevap
  if (elektro_1_2_soran1==1) goto SORULDU;
  callfunc "teststarter";
  callfunc "windowpro",400,400,"","";
  callfunc "testpro",1,60,1,0,1;
  callfunc "teststarter_";
  menu
  "quit",CANCEL,
  "start",Sor_1_devam;
```

**Şekil 4.84:** Tek sorulu bir testin başlık script'i.

Başlık bölümünden sonra soru başlangıcını bildiren XML veriyi oluşturan testquestion fonksiyonu çağrılır. Fonksiyon aynı zamanda sorunun çoktan seçmeli (radio) bir soru olacağını da bildirir. Daha sonra sorunun görsel çıktısını oluşturacak olan fonksiyonlar çağrılarak istemciye gönderilir. **Şekil 4.85'**te sorunun test penceresinde oluşmasını sağlayan XML veriyi gönderen script verilmiştir.

```
Sor_1_devam:
  callfunc "testquestion","1","radio",0;
  callfunc "label","10","10","Robotları geri dönüşüm kutusuna
göndermek için","txt_1","120","20","100","150","255","","","";
  callfunc "label","10","30","hangi tuşlar birlikte
kullanılır?","txt_1","","20","100","150","255","","","";
  callfunc "label","80","50","Yardım almak için L tuşuna basabilirsin
;)","el_1","","","20","40","60","","","";

  callfunc "textbox",230,100,140,100,0,0,0,0,1,1;
  callfunc "addrow"," ##7Yandaki yuvarlaklardan";
  callfunc "addrow","##7seçmek istediğine tıkla. ";
  callfunc "addrow","##7sonra ##1DEĞERLENDİR";
  callfunc "addrow","##7düğmesine bas. ";
  callfunc "textbox_";

  callfunc "radio","10","100","A -> Ctrl","1";
  callfunc "radio","10","120","A -> Alt","2";
  callfunc "radio","10","140","A -> Shift+A","3";
  callfunc "radio","10","160","C ->Ctrl","4";
  callfunc "radio","10","180","F2","5";
  callfunc "test_";
```

**Şekil 4.85:** Tek sorulu bir testin soru oluşturma bölümü.

Sorunun oluşturulmasından sonra istemciden gelen yanıtın beklendiği menü oluşturulur. Menünün gönderdiği kod blokları oyuncu tarafından işaretlenen seçeneğin boş, doğru ya da yanlış yanıt olup olmadığını kontrol ederek ilgili koda

gönderme yapar. **Şekil 4.86**'da seçenek menüsü ve değerlendirme yapan kod kısmı verilmiştir.

```
menu "0",SOR_1_0,  
"1",SOR_1_a,  
"2",SOR_1_b,  
"3",SOR_1_c,  
"4",SOR_1_d,  
"5",SOR_1_e;  
SOR_1_0:  
goto BOS;  
SOR_1_a:  
if (@correct == 1) goto DOGRU;  
goto YANLIS;  
close;  
SOR_1_b:  
if (@correct == 2) goto DOGRU;  
goto YANLIS;  
close;  
SOR_1_c:  
if (@correct == 3) goto DOGRU;  
goto YANLIS;  
close;  
SOR_1_d:  
if (@correct == 4) goto DOGRU;  
goto YANLIS;  
close;  
SOR_1_e:  
if (@correct == 5) goto DOGRU;  
goto YANLIS;  
close;
```

**Şekil 4.86:** Tek sorulu bir testte seçenekler menüsü ve değerlendirme bölümü.

**Şekil 4.87**'de verilen yanıtın doğru olması halinde, sunucuda gerçekleştirilen işlemler ve istemciye gönderilecek kodu hazırlayan fonksiyonlar verilmiştir. Kodda ilk önce soruya doğru yanıt verildiğini veri tabanında saklayan değişkene, 1 değeri atanmıştır. Arkasından sorunun ödülü olan 775 id'li nesne olan kirpiotrus'un oyuncuya verilmesi işlemi gerçekleştirilmiştir. Ödülün verildiğini gösteren mesaj ve parçacık efekti, tasarımcının isteğine göre bu alana eklenebilir. Kodda *DEVAM* etiketlerinin bulunduğu bölüm oyuncunun tamamlaması gereken ardışık görevler hakkında bilgi vermek için eklenmiştir. Eğer oyuncu görevi tamamlamışsa, mesaj penceresini altına sıradaki görev hakkında açıklamalar eklenebilmektedir.

```
DOGRU:  
set elektro_1_2_soran1,1;  
if (countitem(775) > 0) goto KIRPI_VAR1;  
getitem 775,1;  
KIRPI_VAR1:  
callfunc "testmesaj";
```



```
callfunc
"sendeffect","particle","skillup","levelup";
callfunc "sendeffect","text","Aferin","aferin";
callfunc "addrow", "##3Tebrikler :)";
callfunc "addbr";
callfunc "addrow", "##2Kirpiotrus'un depona
eklendiđini";
callfunc "addrow", "##2sohbet penceresinden
görebilirsin.";
callfunc "autowrap";
if (elektro_1_2_soran2 != 1) goto EKLE_1;
DEVAM_1:
if (elektro_1_2_soran3 != 1) goto EKLE_2;
DEVAM_2:
callfunc "testmesaj_";
close;
```

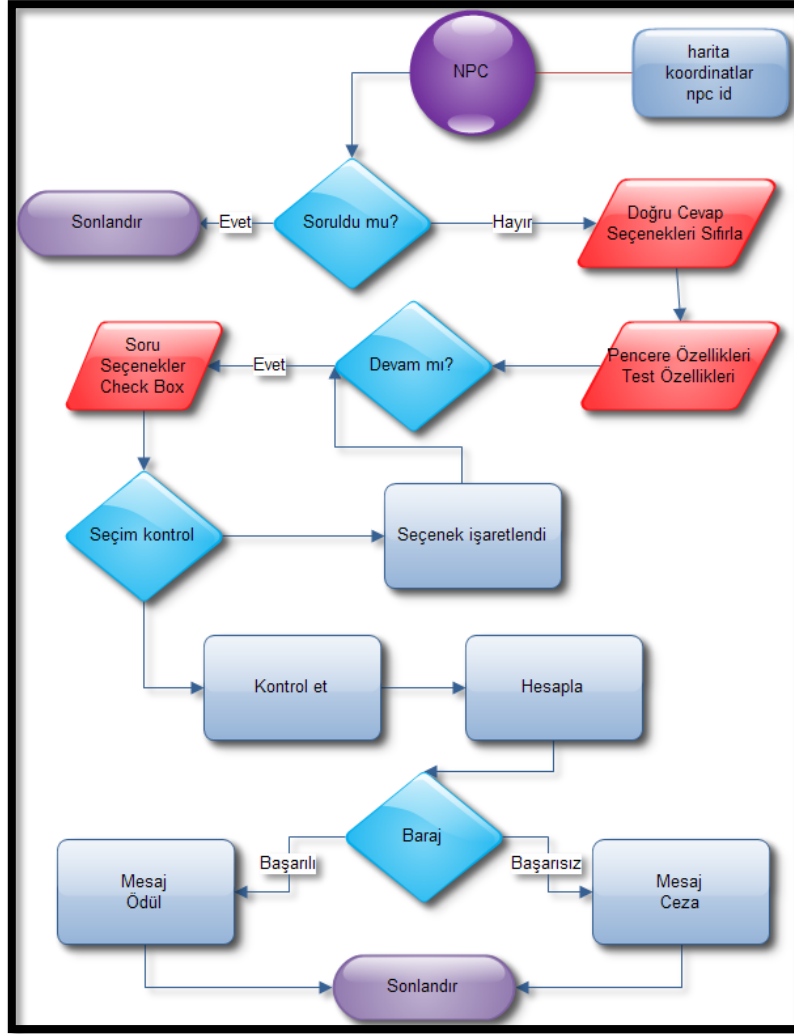
**Şekil 4.87:** Tek sorulu bir testin doğru cevap script'i.

**Şekil 4.88**'de oyuncu yanlış yanıt seçtiğinde yapılacak kod verilmiştir. Doğru cevap için geçerli olan mesaj kuralları yanlış ve boş yanıt için de geçerlidir.

```
YANLIS:
callfunc "testmesaj";
callfunc "addrow", "##1 Üzgünüm :( ";
callfunc "testmesaj_";
close;
```

**Şekil 4.88:** Çoktan seçmeli bir testin yanlış cevap script'i.

Onay kutulu soruları hazırlayan ikinci tipteki script'in yapısı ilkin benzemekle birlikte, seçilen her onay kutusu script'i yeni bir değerlendirme sürecine soktuđu farklıklar da içermektedir. **Şekil 4.89**'da onay kutulu bir sorunun diyagramı verilmiştir.



Şekil 4.89: Onay kutulu bir sorunun diyagramı.

Onay kutularının seçilmiş olması 1 (bir) seçilmemiş olması ise -1 (eksi bir) ile ifade edilmektedir. Script'e ilk girildiğinde onay kutularını saklayan değişkenler 0 (sıfır) değerini taşıyacakları için tüm onay kutusu değişkenlerinin değerinin -1'e eşitlenmesi gerekmektedir. Şekil 4.90'da değişkenleri hazırlama ve doğru cevapları atama komutlarına örnek verilmiştir.

```

if (elektro_1_2_soran2_op_1==0) set elektro_1_2_soran2_option_1, -1;
if (elektro_1_2_soran2_op_2==0) set elektro_1_2_soran2_option_2, -1;
if (elektro_1_2_soran2_op_3==0) set elektro_1_2_soran2_option_3, -1;
if (elektro_1_2_soran2_op_4==0) set elektro_1_2_soran2_option_4, -1;
if (elektro_1_2_soran2_op_5==0) set elektro_1_2_soran2_option_5, -1;
set @elektro_1_2_soran2_correct_1,1;
set @elektro_1_2_soran2_correct_2,1;
set @elektro_1_2_soran2_correct_3,-1;
set @elektro_1_2_soran2_correct_4,-1;
set @elektro_1_2_soran2_correct_5,-1;
  
```

Şekil 4.90: Onay kutularının başlangıçta işaretli olmasını sağlayan ve doğru cevapları global\_reg\_value tablosuna kaydeden kod.

Sorunun sonunda oyuncunun seçeceği onay kutusunun işlenmesi için bir menü bulunur. Menüde onay kutularından birisi seçildiğinde ilgili etikete gönderme yapılır. Burada seçimin hatırlanmasını sağlayan değişkene değeri atanır. Değişkenin hazırlanmasından sonra soru yeniden oyuncuya gönderilir. Eski sorudan farklı olarak onay kutusunu oluşturan XML veriye onay kutusunun seçili olduğunu bildiren bir mesaj ilave edilir. Bu yöntemle her seçimden sonra soruda farklı renk, resim gösterme imkânı da doğmaktadır. Tasarımcı isterse oyuncunun seçimleriyle değişen ekranlar hazırlayabilir. **Şekil 4.91**'de soru sonundaki menü ve ilk onay kutusunun seçilmesi ya da seçiminin kaldırılması durumunda çalışan kod gösterilmiştir.

```
menu "1", KONTROL,  
    "2", SECENEK_1,  
    "3", SECENEK_2,  
    "4", SECENEK_3,  
    "5", SECENEK_4,  
    "6", SECENEK_5;  
  
SECENEK_1:  
    set elektro_1_2_soran2_op_1, elektro_1_2_soran2_op_1 * -1;  
    goto Sor_2_devam;  
SECENEK_2:  
    ...
```

**Şekil 4.91:** Seçenek menüsü ve onay kutusu değişkeninin atanması.

Oyuncu soruyu “Değerlendir” düğmesine bastığında KONTROL bloğu çalıştırılmaya başlar. Bu bölümde öncelikle oyuncuya gönderilecek mesajı içeren metin kutusu oluşturulur. Daha sonra her bir onay kutusunun doğru işaretlenip işaretlenmediğini kontrol eden fonksiyon çağrılır. Seçeneklerin kontrolü bittiğinde metin kutusuna, oyuncunun hangi seçenekleri doğru, hangi seçenekleri yanlış işaretlediği eklenir. Son olarak baraj aşılmışsa ödül verilir. **Şekil 4.92**'de değerlendirmenin yapılabildiği mesajın gönderildiği kod kısmı verilmiştir.

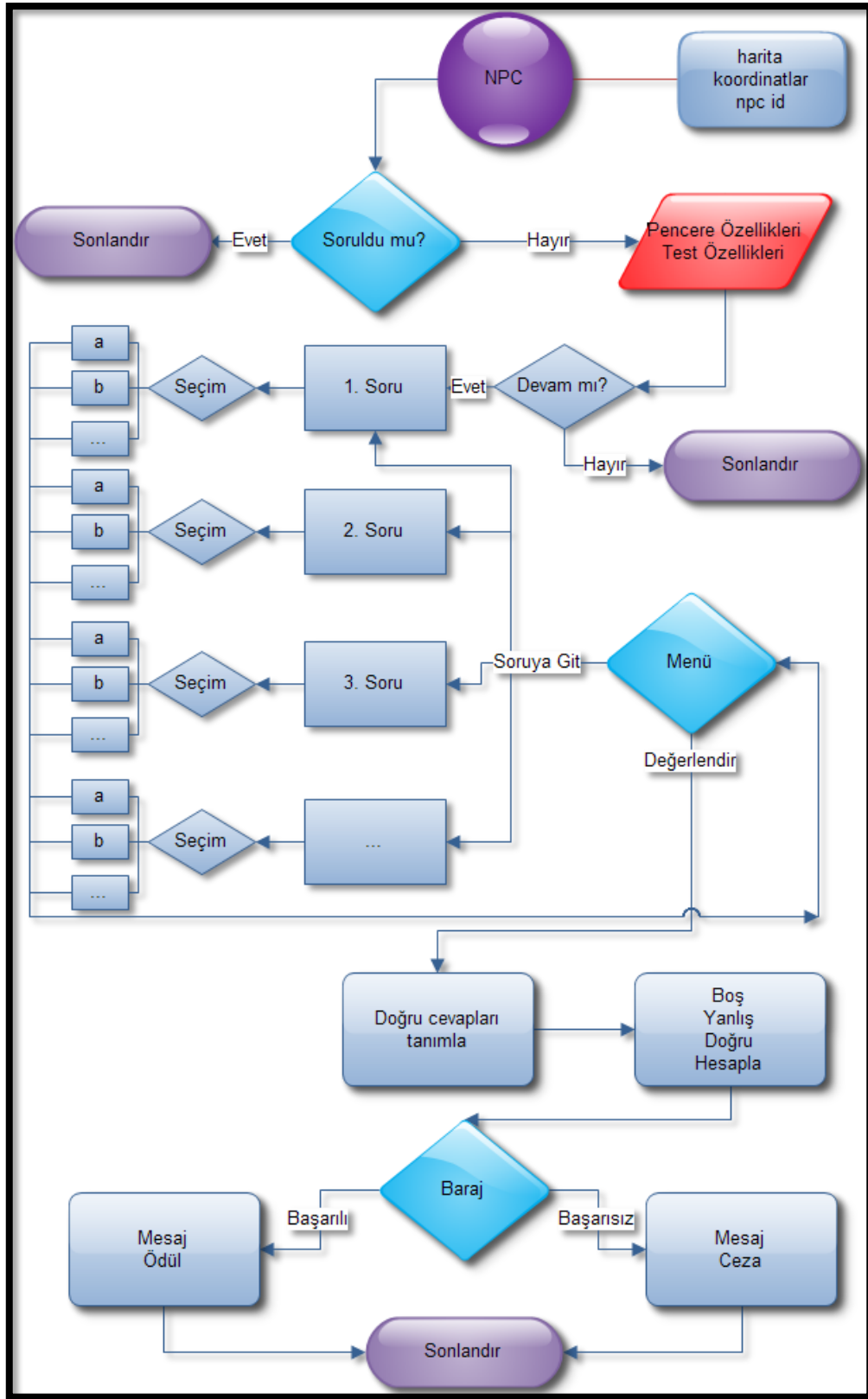
```
KONTROL:  
    callfunc "testmesaj";  
    callfunc "addrow", "##8 == SONUÇ == \" />";  
    callfunc "addrow", "";  
    set elektro_1_2_soran2_result, 0;  
    //seçenek 1  
    set @temp, callfunc ("addcheckkontrol", elektro_1_2_soran2_op_1,  
        @elektro_1_2_soran2_correct_1, "1");  
    set elektro_1_2_soran2_result, elektro_1_2_soran2_result + @temp;  
  
    //seçenek 2  
    ...
```

```
callfunc "addrow","=====";
callfunc "addrow"," ##4Toplam Doğru Cevap : ##6"+
elektro_1_2_soran2_result;

if (elektro_1_2_soran2_result >= elektro_1_2_soran2_limit) goto
ODUL;
ODUL_DEVAM:
callfunc "testmesaj_";
close;
```

**Şekil 4.92:** Onay kutulu soruların değerlendirilmesi.

Test bilgelerine XML veri hazırlayan üçüncü tip script ise çok sorulu bir test hazırlamak için kullanılır. eAthena'nın bahsedilen kısıtlamaları yüzünden bu script'ler oldukça uzun ve karmaşık bir yapıya sahip olabilmektedirler. **Şekil 4.93'**te çok sorulu test hazırlamak için kullanılan script'in diyagramı verilmiştir.



Şekil 4.93: Çok sorudan oluşan bir testin diyagramı.

Çok sorulu testlerde başlık bilgisi soru sayısı dışında Şekil 4.84'te verilen kodla aynıdır. Başlık tanımlandıktan sonra, bu testin oyuncu tarafından daha önce

yapılıp yapılmadığı kontrol edilir. Tasarım aşamasında bir kez açılan testin tekrar açılmasına izin verilmemesi kararı verilmiş ancak uygulamada oyuncuların sadece ne olduğuna bakmak için testi başlattıkları ve sonra geri dönemedikleri için sıkıntı yaşadıkları gözlenince, testin baraj aşıncaya kadar görüntülenebilmesine izin verilmiştir. Bu kez de oyuncuların, barajın altında kaldığı zaman alınan teselli puanını alabilmek için, barajı aşmayacak kadar doğru cevap vererek testi tekrar tekrar aldıkları görülmüştür. Bu durumun önüne geçebilmek için oyuncuların testi tekrar almalarına verilen izin devam ettirilmiş fakat her görüntülemeye baraj altı puan düşürülerek önemsiz hale getirilmiştir.

Çok sorulu testlerde, soruyu oluşturan blok **Şekil 4.85**'te verilen kodla aynı yapıya sahiptir. Ancak seçenekler için kullanılan menüde seçenek işaretlenebileceği gibi sorular arası geçiş de olabileceği için **Şekil 4.94**'teki menü kullanılır.

```
menu
"iptal",CANCEL,
"degerlendir",DEGERLENDIR,
"1",s1,
"2",s2,
"3",s3,
"4",s4,
"5",s5,
"a",s1_a,
"b",s1_b,
"c",s1_c,
"d",s1_d,
"e",s1_e;
```

**Şekil 4.94:** Çok sorulu testte soru sonunda kullanılan menü.

Menüde ilk kısım, soruları tanımlayan etiketlere gönderme yapar. Son kısım ise oyuncunun yaptığı seçimin global\_reg\_value tablosuna kaydedilmesini ve yeniden sorunun başına dönmesini sağlar. **Şekil 4.95**'te her seçenekte seçimi kaydeden ve başa döndüren kod verilmiştir.

```
s1_a:
set elektro_1_2_soran3_s1,1;
goto s1;
```

**Şekil 4.95:** Çok sorulu testin seçeneklerinde çalışan kod.

Oyuncu, "Değerlendir" düğmesine bastığında, hangi soruda olursa olsun test script'inin sonundaki DEGERLENDIR blokuna gönderme yapılır. Bu blokta testin değerlendirme bölümüne gelindiğini gösteren bir değişken kaydedilir. Daha sonra

testin doğru cevapları tanımlanır. **Şekil 4.96**'da, değerlendirme blokunun doğru cevaplarının tanımladığı bölüm verilmiştir.

```
DEGERLENDIR:
//cevap anahtari
set elektro_1_2_soran3,1; // soran_3 ==1 ise soran_3 değerlendirildi
set elektro_1_2_soran3_d1,2; //1=a, 2=b ... soran_3 'ün 1. sorusunun
doğru cevabı
set elektro_1_2_soran3_d2,1;
set elektro_1_2_soran3_d3,5;
set elektro_1_2_soran3_d4,3;
set elektro_1_2_soran3_d5,4;
```

**Şekil 4.96:** Doğru cevap tanımlama.

Oyuncunun verdiği doğru, yanlış ve boş cevapların hesaplanması **Şekil 4.97**'de verilen kontrol bölümünde yapılır. Sonuçlar geçici değişkenlerde saklanır. Hesaplama bittikten sonra **Şekil 4.98**'de verilen kodda olduğu gibi global\_reg\_value tablosuna aktarılarak kalıcı olması sağlanır.

```
set @bos, 0;
if (elektro_1_2_soran3_s1==0) set @bos, (@bos+1);
...
if (elektro_1_2_soran3_s1 == elektro_1_2_soran3_d1) set
@dogru, (@dogru+1);
...
```

**Şekil 4.97:** Doğru ve boş cevapların hesaplanması.

```
set elektro_1_2_soran3_dcev ,@dogru;
set elektro_1_2_soran3_bcev ,@bos;
set @yanlis, @totalquestion - @dogru - @bos;
```

**Şekil 4.98:** Doğru ve boş cevapların hesaplanması.

Son olarak, oyuncunun testte belirlenen barajı aşıp aşmadığı kontrol edilir. Oyuncu eğer barajı aşmışsa, başlangıçta bildirilen ödüle sürpriz bir ek yapılarak kendisine verilir. Tam barajda kaldıysa, sadece açıklanan ödül verilir. Bu durumun, sürprizi almak isteyen oyuncuları daha çok motive ettiği uygulamalar sırasında gözlenmiştir. **Şekil 4.99**'da ödüle gönderme yapan şart komutları ve geri dönmeyi sağlayan etiketler ve reportline fonksiyonu ile mesajın oluşturulması verilmiştir.

```
if (@dogru > @successlimit) goto ODUL;
ODUL_:

if (@dogru == @successlimit) goto ODUL1;
ODUL1_:
```

```
if (@dogru<@successlimit) goto ODUL2;
ODUL2_:
  //duruma göre ödül ya da cezayı ver!!

  //her soru için doğru cevabı ve öğrencinin cevabını gönder.
  callfunc "reportline", "1", elektro_1_2_soran3_d1,
  elektro_1_2_soran3_s1;
  ...

  callfunc "report_"; //raporu bitir

menu
  "1",CANCEL;
CANCEL:
  close;
```

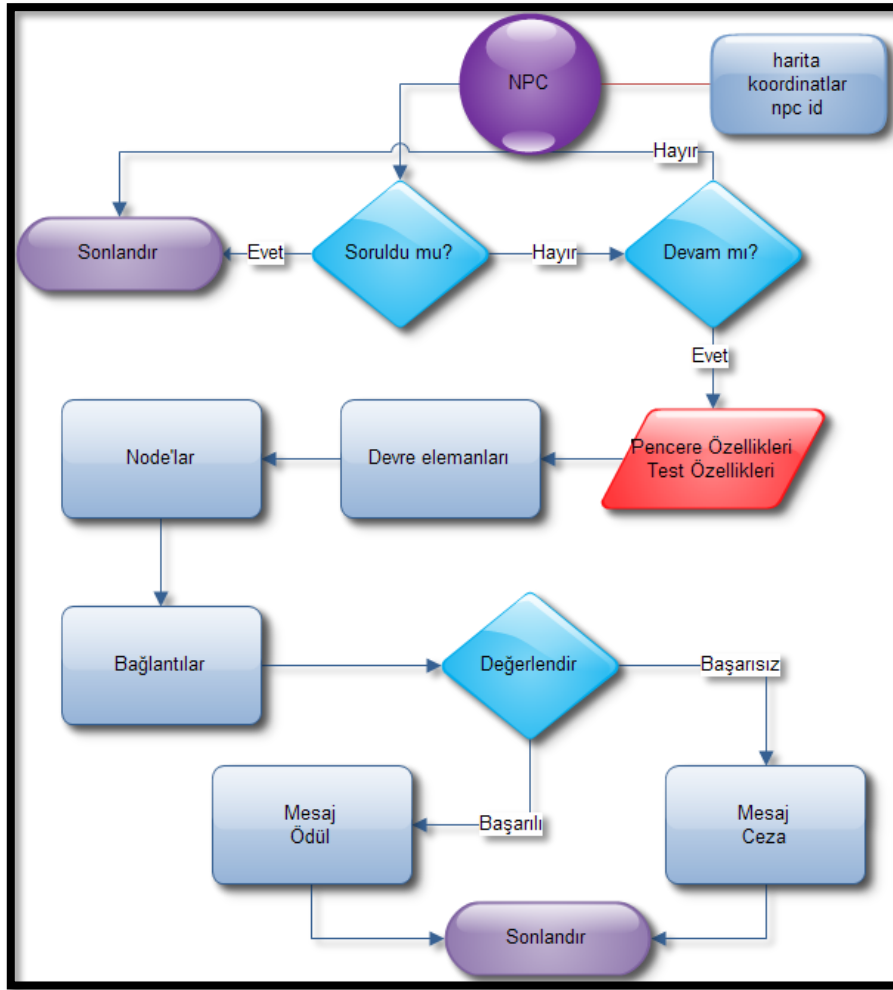
**Şekil 4.99:** Doğru ve boş cevapların hesaplanması.

Test bilgilerinin hazırlanması oldukça zaman alan ve parametrelerin doğru girilmesinde sıkıntılar yaşanan bir aşama olduğu için Bölüm 4.3.4'te bahsedilecek olan yardımcı bir araç geliştirilmiştir.

#### **4.3.2.7 Devre Bilgeleri**

Devre bilgilerinin script'leri devre elemanlarının yerleşimi, bağlantıları ve hedefi içermektedir. **Şekil 4.100**'de bir devre bilgisi oluşturmak için gereken kodun yapısı gösteren diyagram verilmiştir.





**Şekil 4.100:** Devre bilgilerinin verisini oluşturan script'in diyagramı.

Diğer bilgelere benzer şekilde XML verinin başlığı oluşturulur ve pencerenin koordinat gibi özellikleri gönderilir. Devrenin oluşturulması için component, node, connect ve circuitcondition fonksiyonları hazırlanmıştır. **Şekil 4.101**'de component fonksiyonunun çağırısı verilmiştir.

```
//id, item, x, y, aci, st, sl, mv, del, f11, t11, f12, t12
callfunc "component", 1, "1100", 119, 173, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
```

**Şekil 4.101:** Devreye bir bileşen yerleştirmek için kullanılan component fonksiyonunun kullanımı.

Fonksiyonun parametre sayısı çok fazla olduğu için açıklama satırı ile parametrelerin görevleri açıklanmaya çalışılmıştır. Fonksiyonun bulunduğu function.txt dosyasında diğer tüm fonksiyonlarda olduğu gibi daha detaylı bir açıklama yapılmıştır **Şekil 4.102**'de açıklamalarıyla birlikte fonksiyonun kodu verilmiştir.

```

//@@@
// Devre pencerelerine component eklemek için kullanılır
// @id: devre penceresinde nesneyi tanımlayacak sayı !!! Yalnızca tek sayılar
// @item_id: item_db'de tanımlı id'si-> 1100-1137 type ve value item_id tam
çalıştığıında kaldırılabilir.
// @x: koordinat
// @y: koordinat
// @angel: dönme açısı ->0-23
// @status: item sağlam mı-> 0: bozuk 1:çalışıyor
// @selectable: Nesnenin seçilince etrafında mavi halka çıkması ve bazı bilgileri
göstermisi ->0: seçilemez 1:seçilebilir
// @movable: Seçilebilen nesnelere hareket ettirilebilmesi ->0:hareketsiz
1:hareketli
// @fromlink1: Nesnenin 1. node'una dışardan bağlantı yapılabilir
// @tolink1: Nesnenin 1. node'undan diğer node'lara bağlantı yapılabilir
// @fromlink2: Nesnenin 2. node'una dışardan bağlantı yapılabilir
// @tolink2: Nesnenin 2. node'undan diğer node'lara bağlantı yapılabilir
//@@@

function script component {
  mes "<component id=\"" + getarg(0) + "\"";
  mes " item_id=\"" + getarg(1) + "\"";
  mes " x=\"" + getarg(2) + "\"";
  mes " y=\"" + getarg(3) + "\"";
  mes " angel=\"" + getarg(4) + "\"";
  mes " status=\"" + getarg(5) + "\"";
  mes " selectable=\"" + getarg(6) + "\"";
  mes " movable=\"" + getarg(7) + "\"";
  mes " deletable=\"" + getarg(8) + "\"";
  mes " fromlink1=\"" + getarg(9) + "\"";
  mes " tolink1=\"" + getarg(10) + "\"";
  mes " fromlink2=\"" + getarg(11) + "\"";
  mes " tolink2=\"" + getarg(12) + "\" />";
  return;
}

```

**Şekil 4.102:** component fonksiyonu.

Devre elemanları yerleştirildikten sonra, bağlantıların doğru noktalardan geçmesi amacıyla node'ların yerleştirilmesi gerekmektedir. Node'ların yerleşimini sağlayan tag'ı oluşturmak amacıyla node fonksiyonu yazılmıştır. **Şekil 4.103**'te node fonksiyonu çağrısı verilmiştir.

```
callfunc "node",101, 28, 251, 1, 0, 1, 0, 0, 0;
```

**Şekil 4.103:** Node fonksiyonu çağrısı.

Node fonksiyonu parametrelerinin açıklamasıyla birlikte **Şekil 4.104**'te verilmiştir.

```

//@@@
// Devre pencerelerine NODE eklemek için kullanılır
// @id: devre penceresinde nesneyi tanımlayacak sayı
// @x: koordinat
// @y: koordinat
// @status: item sağlam mı-> 0: bozuk 1:çalışıyor
// @selectable: Nesnenin seçilince etrafında mavi halka çıkması ve bazı bilgileri
göstermisi ->0: seçilemez 1:seçilebilir
// @movable: Seçilebilen nesnelere hareket ettirilebilmesi ->0:hareketsiz

```

```

1:hareketli
// @deletable: Silinebilir
// @fromlink: node'a dışardan bağlantı yapılabilir
// @tolink: node'dan diğer node'lara bağlantı yapılabilir
//@@@

function script node {
  mes "<node id=\"" + getarg(0) + "\"";
  mes " x=\"" + getarg(1) + "\"";
  mes " y=\"" + getarg(2) + "\"";
  mes " status=\"" + getarg(3) + "\"";
  mes " selectable=\"" + getarg(4) + "\"";
  mes " movable=\"" + getarg(5) + "\"";
  mes " deletable=\"" + getarg(6) + "\"";
  mes " fromlink=\"" + getarg(7) + "\"";
  mes " tolink=\"" + getarg(8) + "\"";
  mes " />";

  return;
}

```

**Şekil 4.104:** Node fonksiyonu ve parametrelerinin açıklaması.

Devre elemanları ve bağlantı noktaları yerleştirildikten sonra, bağlantıların yapılmasına geçilir. **Şekil 4.105**'te node'lar arasında bağlantı kurmak için kullanılan fonksiyon çağrısı verilmiştir.

```
callfunc "connect",50, 113, 101, 1;
```

**Şekil 4.105:** connect fonksiyonun çağrılması.

Connect fonksiyonu yapı olarak basit olmakla birlikte bağlantı yapılacak node'ların id'si doğru verilmediğinde kararlılık sorunlarına ya da devrenin doğru çalışmamasına neden olabilmektedir. Devre yapımı sırasında en fazla özen bu fonksiyonu kullanırken gösterilmelidir. **Şekil 4.106**'da fonksiyon ve açıklamaları verilmiştir.

```

//@@@
// Devre pencerelerine NODE'lar arası bağlantı yapmak için kullanılır.
// @id: devre penceresinde bağlantıyı tanımlayacak sayı
// @first: bağlantının 1. node'u
// @second: bağlantının 2. node'u
// @active: bağlantı sağlam mı?-> 0: bozuk 1:çalışıyor
//@@@

function script connect {
  mes "<connect id=\"" + getarg(0) + "\"";
  mes " first=\"" + getarg(1) + "\"";
  mes " second=\"" + getarg(2) + "\"";
  mes " active=\"" + getarg(3) + "\" />";
  return;
}

```

**Şekil 4.106:** connect fonksiyonu ve parametrelerinin açıklaması.

Oyuncunun verilen devrede yapması gereken hedefleri circuitcondition tag'ı tanımlamaktadır. Bu tag'ı oluşturmak için kullanılan kod **Şekil 4.107**'de verilmiştir.

```
mes "<circuitcondition type=\"lampturnon\" componentid=\"13\"
status=\"1\" />";
mes "<circuitcondition type=\"locate\" componentid=\"9\" x=\"218\"
y=\"168\" w=\"50\" h=\"50\" />";
```

**Şekil 4.107:** circuitcondition tag'ını oluşturan kod.

Devre elemanlarının yerleştirilmesi ve bağlantılarının doğru yapılabilmesi için çok zaman alan bir işlem olduğu için Bölüm 4.3.4.1'de anlatılan yardımcı program geliştirilmiştir.

### 4.3.2.8 Görev Bilgisayarı Bilgeleri

Görev bilgisayarları, sunucu açısından bakıldığında sıradan bir bilgedir. Onları diğerlerinden farklı kılan, sunucunun hazırlayıp gönderdiği XML veriyi yorumlayan istemci yapısıdır. Görevleri oluşturan tag'ların hazırlanması için öncelikle bir ana görev tanımlanmalı ve ana görev kapatılmadan alt görevler yerleştirilmelidir. **Şekil 4.108**'de bir ana görev ve alt görevin sadeleştirilmiş kodu verilmiştir.

```
callfunc "missionstarter";
callfunc "mainmission","1-Ana Görev";
callfunc "submission",gorev_yuklendi,"check_it.png","alt görev";
mes "<text>";
callfunc "addrow"," Alt görev açıklamaları.";
mes "</text>";
callfunc "submission_";
callfunc "mainmission_";
```

**Şekil 4.108:** Görev bilgisayarına bir görev ve bir alt görev ekleyen kod.

Görev tanımlama ve alt görev ekleme basit bir yapıya sahiptir. Ancak bir görevin yapılıp yapılmadığının anlaşılması, bazı durumlarda eAthena'nın verdiği imkânlarla mümkün olamadığı için script'i yorumlayan kodda ekleme ve düzenlemeler yapılması gerekmiştir. **Şekil 4.109**'da Eğitim Dünyası'ndaki görevlerden birisi olan kıyafet giyme ve ışın kılıcı kuşanma görevinin yapılıp yapılmadığını kontrol eden kod verilmiştir.

```
KIYAFET_KONTROL:
  if (getequipname(9) != "Head 2-[Not Equipped]" && getequipname(4)
  != "Right hand-[Not Equipped]") goto KIYAFET_OK;
  goto KIYAFET_NO;
KIYAFET_OK:
  set @kiyafet , 1;
  goto KIYAFET_RETURN;
KIYAFET_NO:
  set @kiyafet , 0;
  goto KIYAFET_RETURN;
return;
```

**Şekil 4.109:** Bir görevin yerine getirilip getirilmediğini inceleyen kod bloğu.

### 4.3.3 WEB Sitesinin Geliştirilmesi

MMORPG türü oyunların hemen hepsinin bir web sitesi vardır. Web siteleri bu türdeki oyunların bir parçasıdır. MMORPG'ler, oyuncular için ilk karşılaştıklarında oldukça karmaşık ve gizemli dünyalardır. Her oyuncu bu dünyaya ilk adımı web sitesi üzerinden atmaktadır. Oyunu indirmek ve üyelik başta olmak üzere, oyuncunun bu dünyayı keşfetmesini sağlayacak birçok bilgi ve işlem web sayfasında bulunmaktadır.

elektroGame oyununun web sitesi geliştirilmeden önce web sayfasında neler olacağı belirlenmiştir. Buna göre web sitesinde;

- Oyunun genel tanıtımı, hikâyesi ve kuralları,
- Üyelik sistemi,
- Zirvedekiler (Oyuncuların seviyelerinin listelendiği bölüm)
- Robotlar ve nesnelere hakkında ayrıntılı bilgiler,
- Oyunla ilgili haber ve yeniliklerin duyurulacağı bölüm,
- Oyunun ekran görüntüleri,
- Yapımcılarla iletişim,
- Oyuncuların kendi aralarında kuracağı iletişim(Forum)
- Yönetim paneli,

bölümlerinin yer almasına karar verilmiştir.

Web sitesinde yer alacak bölümler belirlendikten sonra, bütün bu ihtiyaçları karşılayıp gelecekte olabilecek değişimlere ve yeniliklere uyumlu olabilecek bir web sitesi tabanı oluşturulmasına karar verilmiştir.

Taban oluşturulması aşamasında, güvenlik, performans, düzenli ve sade kodlama, ekip halinde çalışırken görev paylaşımında kolaylık ve MVC yapısı gibi kriterler göz önünde bulundurularak bir framework altında web sitesinin

geliştirilmesinin uygun olacağına karar verilmiştir. Ayrıca sunucunun teknik özellikleri göz önüne alındığında, yapılacak çalışmaya en uygun web programlama dili olarak PHP, veri tabanı olarak MySQL ve web sunucu olarak da Apache uygun görülmüştür.

PHP framework'ler (Zend, Codeigniter, Symfony, CakePhp) arasında yapılan incelemeler sonucunda, kararlı ve esnek yapısı ve çok geniş bir kütüphaneye sahip olmasından dolayı Zend Framework'te karar kılınmıştır. Ayrıca veri tabanı işlemlerini kolaylaştırmak ve daha anlaşılır bir hale getirmek için “**Doctrine**” isimli “**PHP ORM**” kullanılmasına karar verilmiştir.

### **Zend Framework ve MVC Yapısı**

Framework, yazılımın iskeletini oluşturan, sınıfları ve fonksiyonları içinde barındıran, geniş çaplı bir kütüphanedir (Wikipedia, Software framework, 2011). Zend Framework, Zend firması tarafından geliştirilen bir framework'tür. Tüm kodları test edilmiştir. Bu da güvenliği en üst seviyede tutmakla birlikte özellikle çocuklara yönelik geliştirilen bir sitede karşılaşılabilecek güvenlik açıklarının sonuçları açısından geliştiricilere kolaylık sağlamaktadır.

MVC, Model View Controller kelimelerinin baş harflerinin birleşiminden oluşmaktadır. Bu yapıda web sitesi üç ayrı bölümden oluşmaktadır. Bu bölümler kendi içlerinde bağımsız olarak birbirleri ile bağlantılı olarak çalışmaktadır (Sauter, Vögler, Specht, & Flor, 2005).

**Model:** Veri tabanı işlemlerinin yürütüldüğü bölümdür.

**Controller:** View ile model arasındaki bağlantıyı sağlar. **View**'de kullanıcıdan gelen verileri **Model**'e düzenleyerek gönderir. Aynı işlemi ters yönlü de gerçekleştirir.

**View:** Siteyi gezen kullanıcıların gördüğü kısımdır. Görsel öğelerin tamamı bu kısımda yer alır.

**Doctrine ORM:** Doktrin, PHP için bir object-relational mapper (nesneli ilişkisel Eşleştiricisi - ORM) yapısıdır. Bir veri tabanı soyutlama katmanı üstünde oturur. Doktrin'in en önemli özelliklerinden birisi de SQL'e alternatif özel bir nesne

yönelimli sorgu yazma lehçesi olan, Doktrin Query Language (DQL) bulundurmasıdır.

Şekil 4.110’da Doctrine ile hazırlanmış olan, kullanıcıların karakter ve kullanıcı isimlerine, seviye, para, güç ve benzeri özelliklerine göre listelenebildiği “Zirvedekiler” tablosu verilmiştir.

Karakter	Kullanıcı	Seviye	Para	Güç	Çeviklik	Dayanıklılık	Zeka	Ustalık	Şans
Avatar10can	Avatar10can	43	29134	43	24	26	7	23	18
abcd	silent hunter	36	2551	47	21	22	5	2	2
menderes15	menderes15	35	29	22	22	22	21	21	21
Şukufe	iremyelken	27	46	35	25	21	1	1	1
DeVIL	menderes7	26	1208	32	18	15	7	8	11
uygar	menderes2	23	25324	24	15	11	9	9	15
obendegilim	obendegilim	22	315	32	12	22	2	1	4
NoN-St0P	NoN-StoP	21	53408	29	11	15	7	8	7
al2008	menderes13_	19	103498	22	12	20	2	3	12
KaaN	menderes1	18	14	28	14	10	4	2	9

Şekil 4.110: Doctrine ORM ile hazırlanmış olan oyuncu puanları tablosu.

MVC yapısı kullanılarak Şekil 4.110’da gösterilen tablonun kodlanması şu şekilde gerçekleştirilir:

**Model-Zirvedekiler:** Zirvedekiler bölümündeki listelenen kullanıcılara ait karakter bilgileri sunucuda yer alan elektroGame veri tabanındaki login ve charview tablolarından çekilmektedir. Login tablosu ve charview tablosunun gerekli kısmı Şekil 4.111’de gösterilmiştir.

Login Tablosu		Char Tablosu	
Alan	Türü	Alan	Türü
<u>account_id</u>	int(11)	<u>char_id</u>	int(11)
ad	varchar(20)	account_id	int(11)
soyad	varchar(25)	char_num	tinyint(1)
userid	varchar(23)	name	varchar(30)
user_pass	varchar(32)	class	smallint(6)
lastlogin	datetime	base_level	bigint(20)
sex	enum('M','F','S')	job_level	bigint(20)
logincount	mediumint(9)	base_exp	bigint(20)
email	varchar(39)	job_exp	bigint(20)
level	tinyint(3)	zeny	int(11)
error_message	smallint(11)	str	smallint(4)
connect_until	int(11)	agi	smallint(4)
last_ip	varchar(100)	vit	smallint(4)
memo	smallint(11)	int	smallint(4)
ban_until	int(11)	dex	smallint(4)
state	int(11)	luk	smallint(4)
yetki	enum('oyuncu','ogretmen','yonetic','gelistiric','superyonetic')	max_hp	mediumint(8)
okul	int(11)	hp	mediumint(8)
sinif	int(11)	max_sp	mediumint(6)
numara	varchar(20)	sp	mediumint(6)
aktif	tinyint(4)	status_point	smallint(4)
uyeliktarihi	datetime	skill_point	smallint(4)

**Şekil 4.111:** Login ve Char tablolarının gerekli alanları.

Bu tablolarla verilerin alınabilmesi ve aynı şekilde verilerin yazılabilmesi için, tabloların birbirleriyle ilişkisi gözetilerek model tanımlamalarının yapılması gerekmektedir. Tanımlamalar *Oyuncular.php* dosyasında sırası ile şu şekilde yapılmaktadır:

Model tanımlaması yapılırken ilk olarak her tablo için *Doctrine\_Record* (veri tabanı işlemleri için Doctrine kullanılmasından dolayı) sınıfını miras alan bir sınıf tanımlanır. **Şekil 4.112'**de login sınıfının tanımı verilmiştir.

```
class login extends Doctrine_Record
{
```

**Şekil 4.112:** login tablosunun Model bölümünde sınıf tanımlaması.

Dikkat edilmesi gereken nokta, sınıf adı ile tablo adının aynı olması zorunluluğudur. Sınıf tanımlamadan sonra, sınıfın içine tablonun içindeki alanların



tanımlamalarının yapılacağı *setTableDefinition()* fonksiyonu yazılır. Bu fonksiyonun içerisinde, tablonun içinde yer alan her alan ve özellikleri birer birer *hasColumn()* fonksiyonu aracılığı ile sınıfa eklenir. Login ve charview tablolarının alanlarının eklendiği Oyuncular.php'nin bir kısmı **Şekil 4.113**'te gösterilmektedir.

```
class login extends Doctrine_Record
{
    public function setTableDefinition()
    {
        $this->hasColumn('account_id', 'integer', 11, array('type' => 'integer', 'length' => 4, 'primary' => true, 'autoincrement' => true));
        $this->hasColumn('ad', 'string', 20);
        $this->hasColumn('soyad', 'string', 25);
        $this->hasColumn('userid', 'string', 23);
        $this->hasColumn('user_pass', 'string', 32);
        $this->hasColumn('lastlogin', 'datetime');
        $this->hasColumn('sex', 'enum', null, array ( 'values' => array ('M', 'F', 'S')));
        $this->hasColumn('logincount', 'integer', 9);
        $this->hasColumn('email', 'string', 39);
        $this->hasColumn('level', 'integer', 3);
        $this->hasColumn('error_message', 'integer', 11);
        $this->hasColumn('connect_until', 'integer', 11);
        $this->hasColumn('last_ip', 'string', 100);
        $this->hasColumn('memo', 'integer', 11);
        $this->hasColumn('ban_until', 'integer', 11);
        $this->hasColumn('state', 'integer', 11);
        $this->hasColumn('yetki', 'enum', null, array ( 'values' => array ('oyuncu', 'ogretmen', 'yonetici')));
        $this->hasColumn('okul', 'integer', 11);
        $this->hasColumn('sinif', 'integer', 11);
        $this->hasColumn('numara', 'string', 20);
        $this->hasColumn('aktif', 'integer', 4);
        $this->hasColumn('uyelik tarihi', 'datetime');
    }
}
```

**Şekil 4.113:** Oyuncular.php dosyasındaki login sınıfının bir bölümü.

Alanların tanımlamaları *hasColumn()* fonksiyonu ile gerçekleştirildikten sonra eğer tablolar arası ilişkiler mevcutsa *setUp()* fonksiyonu ile bu ilişkiler karşılıklı olarak tanımlanır. Bu tanımlamalar hem login sınıfında hem de charview sınıfında ayrı ayrı yapılmaktadır. **Şekil 4.114**'te login sınıfındaki, **Şekil 4.115**'te charview sınıfındaki *setUp()* fonksiyonları yer almaktadır. login sınıfındaki tanımlamada bir oyuncunun birden fazla karakteri olabileceği için *hasMany()*, charview sınıfında ise bir karakter sadece tek bir oyuncuya ait olabileceği için *hasOne()* fonksiyonu kullanılarak ilişkilendirilecek alanlar belirtilmiştir.

```
public function setUp()
{
    $this->hasMany('charview as charviews', array(
        'local' => 'account_id',
        'foreign' => 'account_id'
    ));
}
```

**Şekil 4.114:** Login sınıfı *setUp()* fonksiyonu.

```

public function setUp()
{
    $this->hasOne('login', array(
        'local' => 'account_id',
        'foreign' => 'account_id'
    ));
}

```

Şekil 4.115: charview sınıfı *setUp()* fonksiyonu.

**Controller-Zirvedekiler:** Model tanımlamaları bitirildikten sonra veri tabanındaki işlemlerin kontrollerinin yapılacağı Controller bölümünde veri tabanındaki kayıtlar, ilişkiler göz önünde bulundurularak alınır, işlenmesi gerekiyorsa işlenir ve View'e gönderilir. Yapılacak işlemler sitenin genel görünümünün kontrollerinin yer aldığı IndexController.php dosyasında *zirvedekilerAction()* fonksiyonunda yapılır.

```

public function zirvedekilerAction()
{
    $zirvedeki = new Model_Oyuncular();
    $q = Doctrine_Query::create()
        ->from('charview o')
        ->leftJoin('o.login k');
    $this->view->zirvedekiler = $q->execute();
}

```

Şekil 4.116: IndexController'da yer alan *zirvedekilerAction()* fonksiyonu.

İlk olarak login ve charview tablolarının modellerinin kullanılabilir hale gelmesi için sınıflarının çalıştırılması gerekmektedir. Bunun için MVC yapısı gereği login ve char sınıflarının yer aldığı Oyuncular.php dosyasında bu sınıflara ek olarak bu sınıflardan türetilmiş olan *Model\_Oyucular()* sınıfı çağrılır. Bu çağrı ile birlikte Oyuncular.php çalışarak charview ve login sınıfları kullanılabilir hale gelir.

İkinci aşamada Doctrine kullanılarak, karakter bilgilerinin yer aldığı tablodan kullanıcı tablosu ile ilişki de kurularak karakter bilgileri alınır. Daha sonra da alınan veriler View'a gönderilir.

**View-Zirvedekiler:** Controller tarafından gönderilen veriler HTML kodları yardımı ile istenilen şekilde kullanıcıya sunulur. Veriler Şekil 4.117'deki kodlarla ekranda gösterilmiştir. Bu kodlar zirvedekiler.phtml dosyasında yer almaktadır.

```

$zirvedekiler = $this->zirvedekiler;
foreach($zirvedekiler as $karakter)
{
    echo "<tr>";
    echo "<td>".$karakter['name']."</td>";
    echo "<td>".$karakter->login['userid']."</td>";
    echo "<td>".$karakter['base_level']."</td>";
    echo "<td>".$karakter['zeny']."</td>";
    echo "<td>".$karakter['str']."</td>";
    echo "<td>".$karakter['agi']."</td>";
    echo "<td>".$karakter['vit']."</td>";
    echo "<td>".$karakter['int']."</td>";
    echo "<td>".$karakter['dex']."</td>";
    echo "<td>".$karakter['luk']."</td>";
    echo "</tr>";
}

```

**Şekil 4.117:** Karakter verilerinin web sitesinde gösterilmesi.

Bu bölümde, sitedeki ziyaretçilerin istedikleri istatistiklere göre karakterleri listeleyebilmesi için jQuery Javascript kütüphanesi ile geliştirilmiş tablesorter isimli tablodaki verileri istatistiklerine göre alfabetik, büyükten küçüğe ya da küçükten büyüğe sıralayabilen bir Javascript betiği kullanılmıştır (**Şekil 4.115**).

#### 4.3.3.1 Wiki

Özgür ansiklopedi sanılarak Wikipedia ile karıştırılan, herkesin katkıda bulunabildiği bir sözlük olarak bilinen Wiki aslında “GNU Özgür Belgeleme Lisansı altında kullanıcıların yeni sayfa yaratmasına, sayfalarda düzenlemeler yapmasına ve bu sayfaları birbirine bağlamasına olanak sağlayan bir yazılımdır” (Wikipedia, Wiki, 2011).

Açık kaynak kodlu yazılımların genelinde yer alan wiki sayfaları, programın geliştiricileri tarafından geliştiricilere ya da farklı türde programa katkı sağlayacak kişilere programın kodları, grafikleri, ayarları ya da bu gibi konularda ayrıntılı açıklamalar yaptığı sayfalardır.

Günümüzde birçok açık kaynak kodlu wiki sistemi vardır (MediaWiki, TWiki, PmWiki, DokuWiki, MoinMoinWiki). Bu çalışmada dünyada en çok

kullanılan wiki olan Wikipedia'nın da kullandığı wiki sistemi olan MediaWiki kullanılmıştır.

Bu sistemi kullanabilmek için öncelikle MediaWiki, internet sitesinden indirilip sunucumuza kurulmuştur ve yeni wiki sistemine subdomain ayarları ile doğrudan bağlantı verilmiştir (wiki.elektrogame.net).

elektroGame'in wiki sayfasına içerik olarak geliştiriciler ve kullanıcılar için iki ayrı bölüm hazırlanmıştır (**Şekil 4.118**). Geliştiriciler bölümünde elektroGame'in geliştiricilerine ve gelecekte elektroGame kullanarak yeni eğitsel uygulamalar hazırlayacak kişilere yönelik, Sunucu-İstemci yapısı, NPC'ler, harita yapımı gibi konularda ayrıntılı bilgiler bulunmaktadır. Kullanıcılar bölümünde ise genel web sitesindeki bilgilere ek olarak oyun ile ilgili daha ayrıntılı bilgiler yer almaktadır.



**Şekil 4.118:** elektroGame wiki sayfası.

Wiki sistemine yeni bir sayfa eklemek için tarayıcının adres çubuğuna, istenen sayfa ismi yazılır ve sayfa açılır. Örneğin, Deneme isimli yeni bir sayfa oluşturmak için <http://wiki.elektrogame.net/index.php/Deneme> yazılıp sayfa

açıldığında **Şekil 4.119**'daki gibi bir sayfa görünür. Açılan sayfadan “*oluştur*”a tıkladığında, sayfa içine eklenebilecek bilgiler yazılarak “*Sayfayı Kaydet*”e tıklanır ve yeni bir sayfa oluşturulmuş olur.



**Şekil 4.119:** <http://wiki.elektrogame.net/index.php/Deneme> sayfası.

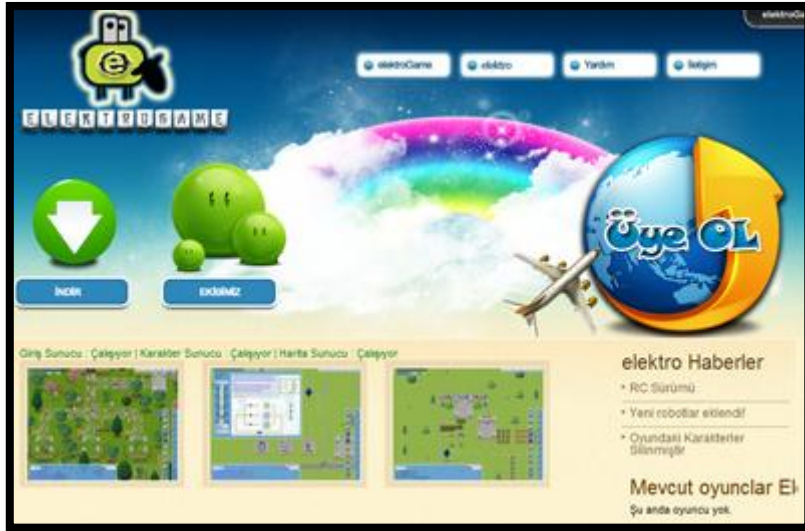
Deneme sayfası oluşturulduktan sonra index sayfadan bu Deneme sayfasına bağlantı vermek için index sayfasına “[<http://wiki.elektrogame.net/index.php/Deneme> Deneme]” satırı eklenir.

#### 4.3.3.2 Blog

Sürekli olarak geliştirilen elektroGame’de, oyuncuların yapılan değişikliklerden haberdar olması için bir blog sayfası hazırlanmıştır. Ayrıca hazırlanan blog sayfası ile oyunun ana sayfası birbirine entegre hale getirilerek elektroBlog’a yazılan haberlerin elektroGame’in ana sayfasında da otomatik olarak gösterilmesi sağlanmıştır. **Şekil 4.120**'te elektroBlog, **Şekil 4.121**'de ise elektroGame’in ana sayfasında blog’tan alınan haberlerin gösterimi vardır.



Şekil 4.120: elektroBlog'un bir görüntüsü.



Şekil 4.121: elektroGame'in ana sayfasında blog'taki haberlerin gösterimi.

Blog sistemi olarak açık kaynak kodlu wordpress sistemi kullanılmıştır. Wordpress sistemi sitesinden indirilerek sunucumuza kurulmuş ve gerekli subdomain ayarlamaları ile doğrudan erişim sağlanmıştır (blog.elektrogame.net).

Blog içeriğini sunucuda yer alan MySQL veri tabanında saklayacak şekilde ayarlanmıştır. Bu sayede blog'ta yer alan haber bilgileri, elektroGame ana sayfasında da gösterilmiştir. Bu işlem için Şekil 4.122'deki kod kullanılmıştır.

```

$haberler = new Model_Haberler();
$haberler = Doctrine_Query::create()
    ->from('Model_Haberler h')
    ->orderBy('h.tarih DESC')
    ->limit(3)
    ->execute();
$this->view->haberler = $haberler;

```

**Şekil 4.122:** Haberlerin ana sayfada gösterilmesi için veri tabanından verileri alan kod.

Öncelikle, Model\_Haberler modeli ile haberler veri tabanından tarihe göre sıralanmış olarak son 3 tanesi olacak şekilde alınır ve ana sayfanın gösterileceği bölüme *haberler* dizisi ile gönderilir.

#### 4.3.3.3 Yönetim Paneli

elektroGame’de oynayan öğrencilerin, oyun dünyası içerisinde neler yaptıklarını, neler konuştuklarını, hangi görevleri tamamlayıp hangilerini tamamlamadıklarını takip edebilmek, kullanıcı bilgilerini, okulları ve sınıfları yönetebilmek için yönetim bölümü geliştirilmiştir.

Yönetim panelini öğretmenler, yöneticiler ve geliştiriciler kullanabilmektedir. Bu bölümde her grubun ulaşabileceği bölümler belirlidir ve her kullanıcı sadece kendine izin verilen bölümleri görebilir ve izin verildiği kadar işlem yapabilir.

Yönetim bölümüne adres girilerek erişildiğinde, ilk olarak **Şekil 4.123**’te görülen kullanıcı giriş ekranı gelmektedir. Kullanıcı adı ve şifre ile giriş yapıldığında menüden istenilen işlemlerin yer aldığı sayfalar açılarak işlemler yapılabilir.

**Şekil 4.123:** Yönetim paneli kullanıcı giriş ekranı.



Yönetim panelindeki bölümlerin başında öğrencilerin aralarındaki sohbetlerin incelenebildiği sohbet kaydı bölümü yer almaktadır. Sohbet kaydı bölümüne girildiğinde öncelikle tüm sohbet kayıtları **Şekil 4.124**'teki gibi listelenmektedir.

id	Kullanıcı	Karakter	Kime	İleti	Zaman	Harita
171	alihadar123	koroner	Herkese	kimsa yokmu	2010-09-01 19:16:26	elektro_1-1
172	alihadar123	koroner	Herkese	kimsa yokmu	2010-09-01 19:16:31	elektro_1-1
173	alihadar123	koroner	Herkese	slm	2010-09-01 19:17:00	elektro_1-1
174	alihadar123	koroner	Herkese	slm	2010-09-01 19:17:04	elektro_1-1
263	puls		Herkese	adinda yada şifresinde türkiye karakter olanlar yine giremiyor oyuna :(	2010-04-19 22:09:28	elektro_0-1
262	Ufuk	cuteBoss	Herkese	ayrı haritaymış hocam hüseyin hocanın kide	2010-04-19 22:09:19	elektro_0-1
258	puls		Herkese	Hoggeldiniz	2010-04-19 22:08:11	elektro_0-1
259	Ufuk	cuteBoss	Herkese	selam hocam hüseyim hocanın haritada açık	2010-04-19 22:08:43	elektro_0-1
282	Ufuk	cuteBoss	Herkese	ilk başta ki ekranın üst ve solu biraz kanşık geldi ama	2010-04-19 22:21:01	elektro_1-1
281	puls		Herkese	sen bir şey anlatıyordun	2010-04-19 22:20:10	elektro_1-1

**Şekil 4.124:** Yönetim panelindeki sohbet kaydı bölümü.

Başlık bölümlerine tıklatılarak sohbet kaydı alfabetik olarak kullanıcı adı, karakter, mesajın gönderildiği kişi ve mesaja göre ya da tarih olarak mesajın gönderildiği tarihe göre sıralanabilir. Ayrıca sayfanın üst kısmında yer alan arama bölümünden oyuncu adı, karakter, mesajın gönderildiği kişi ya da mesaja göre arama yapılabilir. Örneğin, sadece tek bir öğrencinin tüm mesajları ya da sadece tek bir öğrenciye gönderilmiş tüm mesajlar listelenebilir.

Yönetim panelindeki en önemli bölüm, öğrencilerin oyun içinde ne yaptıklarının takip edilebildiği **Görev Listele** bölümüdür. Bu bölüme giriş yapıldığında sırasıyla okul, sınıf, öğrenci ve öğrencinin görevlerinin takip edilmek istendiği dünyanın **Şekil 4.125**'teki gibi seçilmesi gerekmektedir.

<b>Okul Seç</b> Atatürk Sağlık Mes. I	<b>Sınıf Seç</b> 9A	<b>Örenci Seç</b> melek1995	<b>Karakter Seç</b> melek1995	<b>Görev Seç</b> Eğitim Odası	<b>Göster</b>
--	------------------------	--------------------------------	----------------------------------	----------------------------------	---------------

**Şekil 4.125:** Görev Listele bölümünde öğrenci seçimi.

Öğrenci seçimi yapıp “Göster” düğmesine basıldığında, aşağıda öğrencinin seçilen bölümde hangi görevleri yaptığı veya yapmadığı, hangi testlerde nasıl cevaplar verdiği gibi bilgiler **Şekil 4.126** ve **Şekil 4.127**'teki gibi listelenmektedir.



Eğitim Odası Görevleri				
Ad	Soyad	Hesap ID	Karakter Adı	Karakter ID
özge	özdemir	2000232	ilkay	150105

Genel	Durum
elektro_1_2_gorev_sunu_1	✗
elektro_1_2_karakter_sunu	✓
elektro_1_2_kisayol_sunu	✗
elektro_1_2_geri_donusum_sunu	✓
elektro_1_2_kiyafet_sunu	✗

Şekil 4.126: Öğrencinin genel görevlerdeki durumu.

Şekil 4.126’da öğrencinin genel bilgileri ve Eğitim Dünyası’ndaki genel görevlerdeki durumu gösterilmektedir. Yeşil tik işaretli olanlar öğrencinin başarıyla gerçekleştirdiği görevler, kırmızı çarpı işaretli olanlarsa başaramadığı görevlerdir.

elektro_1_2_soran2	Durum	Başarılı mı
elektro_1_2_soran2_s1	✗	✗
elektro_1_2_soran2_s2	✗	
elektro_1_2_soran2_s3	✓	
elektro_1_2_soran2_s4	✗	
elektro_1_2_soran2_s5	✗	
<b>Toplam Doğru Cevap Sayısı</b>		1
<b>Başarılı olma limiti</b>		3


elektro_1_2_soran3	D. Cevap	Ögr. Cevabı	Durum
elektro_1_2_soran3_s1	b	e	✗
elektro_1_2_soran3_s2	a		✗
elektro_1_2_soran3_s3	e		✗
elektro_1_2_soran3_s4	c		✗
elektro_1_2_soran3_s5	d		✗
<b>Toplam Doğru Cevap Sayısı</b>			0

Şekil 4.127: Öğrencinin testlerdeki durumunun gösterimi.

Şekil 4.127’de, öğrencinin yine aynı dünyada cevaplamış olduğu testlerdeki durumu gösterilmektedir. *elektro\_1\_2\_soran2* bilgesinin sorduğu sorularda öğrenci

sadece 3. soruyu doğru cevaplamış ve başarı limiti olan 3'ü aşmadığı için testte başarısız olmuştur. *elektro\_1\_2\_soran3* bilgesinin sorduğu çoktan seçmeli sorulara öğrencinin verdiği cevaplar, doğru cevaplar ve toplam doğru cevap sayısı gösterilmektedir.

Yönetim panelinde okul, sınıf, öğretmen, öğrenciler ve yöneticiler eklenebilir, silinebilir ve düzenlenebilir. Okul eklemek için yönetim bölümünden okullar bölümüne girildikten sonra, okulların listesi ile karşılaşılmaktadır. Okulların üstünde yer alan ve **Şekil 4.128**'de de görülen “yeni”, “düzenle” ve “sil” düğmeleri kullanılarak yeni okul eklenebilir, var olan okullar düzenlenebilir ya da silinebilir.



id	Okul Adı	Seç
20	Atatürk Sağlık Mes. L.	<input type="radio"/>
10	Balıkesir Üniversitesi	<input type="radio"/>
1	Menderes	<input type="radio"/>
21	Sım Yırcalı A. L.	<input type="radio"/>

**Şekil 4.128:** Okul yönetim bölümü.

Öğretmen, öğrenci (oyuncu) ve yöneticiler için işlem yapmak için “oyuncular” bölümüne girilir. Burada kayıtlı tüm kullanıcılar **Şekil 4.129**'da görüldüğü gibi listelenir. “Önceki” ve “sonraki” düğmeleri kullanılarak bütün oyuncular sırasıyla görülebilir. Aynı sayfada 10'dan fazla kayıt, “kayıt göster” isimli açılır kutunun değeri değiştirilerek listelenebilir.

Yeni bir oyuncu, yönetici ya da öğretmen oluşturmak için “yeni” düğmesi tıklanır ve açılan pencerede (**Şekil 4.130**) gerekli bilgiler doldurularak “kaydet” düğmesine basılır. Bilgiler girilirken yetki kısmından, oluşturulan yeni oyuncunun yönetici, öğretmen ya da öğrenci mi olacağı belirlenir.

Oyuncular arasında istenilen oyuncuyu bulmak için arama bölümüne “oyuncu” ismi girildiğinde ve “göster” düğmesine basıldığında sonuçlar listelenir. Bu bölümde, yöneticilere kolaylık olması için aranmak istenen oyuncu bölümüne henüz tam olarak girilmeden harfler girildiğinde, o harfleri içeren oyuncular **Şekil 4.129**'daki gibi otomatik olarak listelenmektedir.

Oyuncu bilgilerini düzenlemek ya da oyuncuyu silmek için en sağ tarafta yer alan “seç” bölümünden silmek ya da düzenlemek istenilen oyuncu seçilip aşağı kısımda yer alan “düzenle & sil” düğmesine tıklanır. Bu işlemden sonra açılan pencereden, silme ya da düzenleme işlemi yapılabilir.

elektroWeb -> Oyuncu Yönetim

Bir Autocomplete input text örneği... Göster

(Yeni) kayıt göster

Özellikler: Oyuncu Adı, Yetki, Level, Eposta, Cinsiyet, Okul, Sınıf, Numara, Aktif, Seç

Oyuncu Adı	Yetki	Level	Eposta	Cinsiyet	Okul	Sınıf	Numara	Aktif	Seç
10	oyuncu	0							
10kolikiçi	oyuncu	0							
1234	oyuncu	0							
3444	oyuncu	0	muhammed_44_1998@hotmail.com	M	20	13	112	0	
34sagopa44	oyuncu	0	muhammed_44_1998@hotmail.com	M	20	13	112-104	0	
34sagopa44	oyuncu	0	muhammed_44_1998@hotmail.com	M	1	0	1098	0	
aaaa	oyuncu	0	a@a.com	F	0	0	0	0	
abcd	oyuncu	0	a@a.com	M	0	0	0	0	
Abeja	oyuncu	0	sago_mhmt@hotmail.com	M	20	13	105	0	
affetmez	oyuncu	0	a@a.com	M	0	0	0	0	

1 -> 10 kadar gösteriliyor 245 toplam kayıt

Düzenle & Sil İlk Önceki 1 2 3 4 5 Sonraki Son

Şekil 4.129: Oyuncu yönetimi bölümü.

Oyuncu Kayıt

Oyuncu Adı:

Şifre:

Şifre (Tekrar):

Cinsiyet:

Eposta Adresiniz:

Yetki:

Okul Adı:

Sınıf:

Numara:

Aktif:

kaydet

Şekil 4.130: Yeni yönetici-öğretmen-oyuncu ekleme penceresi.

#### 4.3.4 Yardımcı Programlar

MMORPG oyunların programlanması kadar programa data olarak girilen verilerin yapılandırılması ve sınanması da uzun, dikkat gerektiren ve yorucu bir süreçtir. Genellikle programcılar, uzun ve zor olan bu süreci tasarımcılar açısından kolaylaştırmak için, programlama zamanından ciddi bir pay ayırırlar. elektroGame’de de bir yandan istemci ve sunucunun hedefler doğrultusunda kararlı çalışması için programlama yapılırken bir yandan da eğitsel içeriğin oyuna eklenebilmesi amacıyla yardımcı bazı programlar hazırlanmıştır.

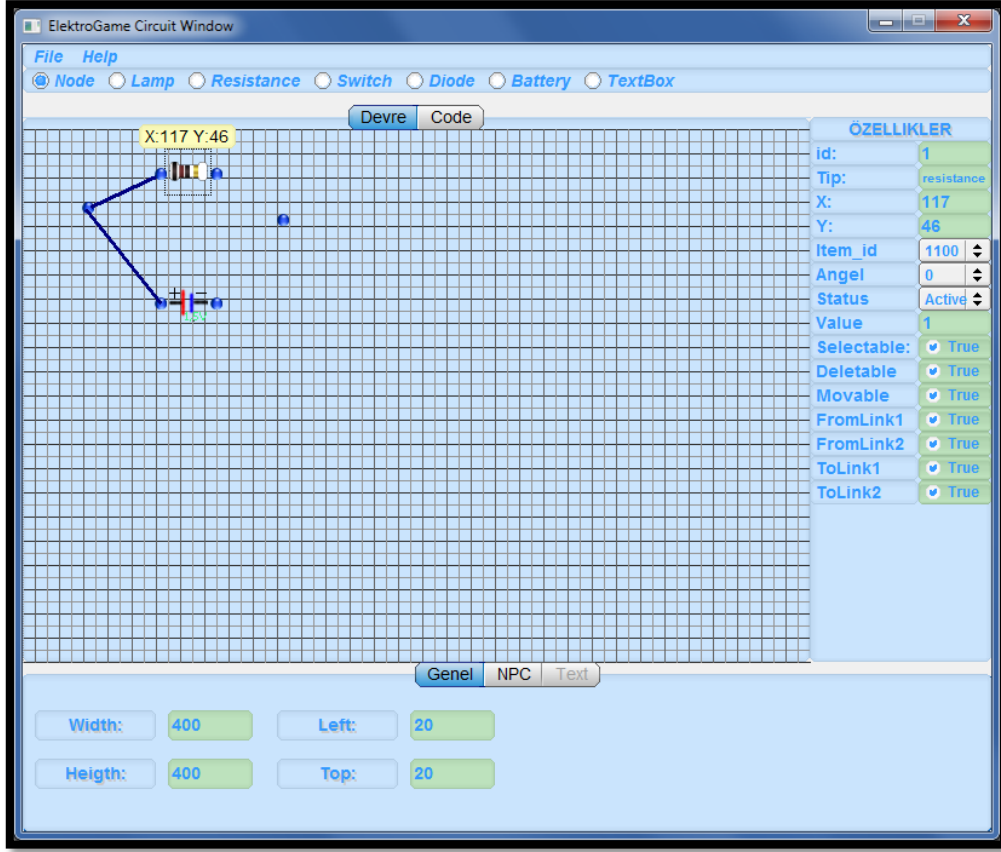
Yardımcı yazılımlar çoğunlukla FLTK kütüphanesi kullanılarak geliştirilmiş olmakla birlikte sunucu için hazırlanmış olanlardan bir kısmı **sh** betikleri ile yazılmıştır.

Bu bölümde elektroGame’e eğitsel içeriğin eklenmesinde en fazla kullanılan devre tasarım ve test tasarım programları ile sunucunun çalıştırılması, yeniden başlatılması gibi işlerde kullanılan sunucu kontrol programı tanıtılacaktır.

##### 4.3.4.1 Devre Tasarım Programı

Devre tasarım programı Bölüm 4.3.2.7’de bahsedildiği gibi eAthena script için gerekli kodu üretmek amacıyla geliştirilmiştir. Script ise çıktı olarak uygun XML veriyi üretecek şekilde hazırlanmıştır. FLTK kütüphanesi kullanılmıştır.

Devre bilgilerinin hazırlanmasında karşılaşılan üç önemli zorluk bulunmaktadır. Birincisi bağlantıyı sağlayan “connect” tag’larının doğru id’leri referans vermesi. İkincisi kabloların görsel yapıyı bozmayacak şekilde koordinatlarının ayarlanması. Üçüncüsü ise her elemanın id’sinin çakışmaları bozmayacak şekilde verilmesidir. Hazırlanan yardımcı tasarım programının öncelikli amacı, bu üç sorunun üstesinden gelmektir. **Şekil 4.131**’de devre tasarım programının genel görünümü verilmiştir.



Şekil 4.131: Devre tasarım programının genel görünümü.

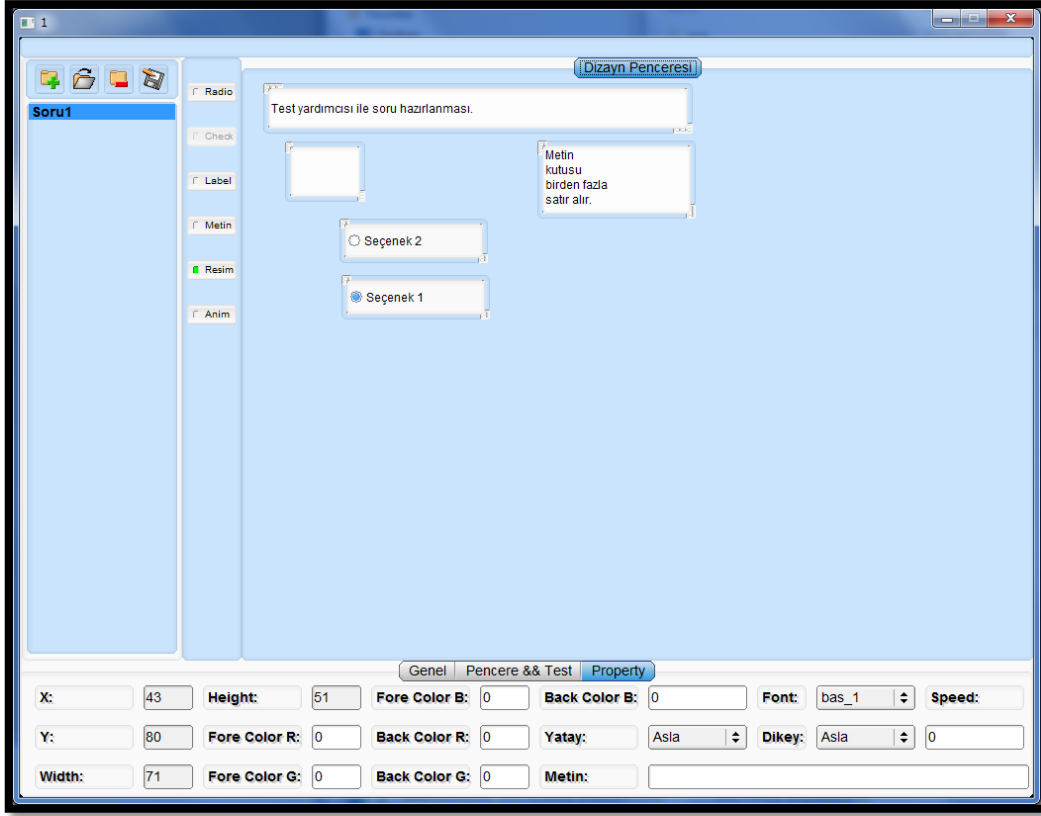
Programın yapısı; bir yerleşim alanı, devre elemanı özellikleri paneli, bilgi özellikleri paneli ve araç çubuğundan oluşmaktadır. Araç çubuğunda seçili olan devre elemanı, farenin orta tuşuna tıklanarak yerleştirilir. Bağlantı noktası (node) çok sık yerleştirildiği için farenin sağ tuşuna atanmıştır. Node'lar arası bağlantı ise CTRL tuşuna basılı tutularak sürüklendiğinde ortaya çıkartılmaktadır.

Seçilen her devre elemanı için özellikler panelinde component tag'ının özellikleri listelenir ve düzenlenebilmesi sağlanır.

Tasarım bittiğinde Code düğmesine basılarak eAthena script dilinde yazılmış kod görüntülenir. İstenirse bir TXT dosyaya yazdırılır ya da panoya kopyalanır.

#### 4.3.4.2 Test Tasarım Programı

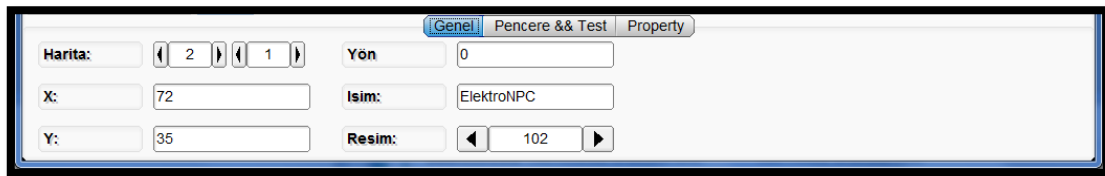
Test bilgilerinin eAthena Script kodunu üretmek üzere tasarlanmıştır. FLTK kütüphanesi kullanılmıştır. Üç soru tipini de desteklemektedir. Şekil 4.132'de programın genel görünüşü verilmiştir.



Şekil 4.132: Test tasarım penceresinin genel görünüşü.

Programın sol tarafında bulunan panel, çok sorulu test yapımı için soru eklemek ve silmek için kullanılan düğmeleri ve eklenmiş sorular arasında geçiş için soru listesini içermektedir. Sağ panel, tasarımın yapıldığı bölümdür ve hemen solunda yerleştirilmek istenen bileşenin seçildiği düğmeler yer alır.

Pencerenin alt kısmına üç farklı sekme yerleştirilmiştir. Property sekmesi, seçili bileşenin özelliklerinin değiştirilebilmesini sağlar. Şekil 4.133'te bilgeye ait özelliklerin girildiği sekmenin görünüşü sunulmuştur.



Şekil 4.133: Bilgenin genel özelliklerini ayarlayan panel.

Test penceresinin özellikleri ayrı ise Şekil 4.134'te verilen resimdeki sekmede ayarlanmaktadır.

Width:	400	Top:	20	Punish:	10
Height:	400	Total Time:	50	Success Limit:	5
Left:	20	Award:	20		

**Şekil 4.134:** Testin özelliklerini ayarlayan panel.

Tasarım bittiğinde “kaydet” butonuna basılarak oluşturulan script, diske bilgenin ismiyle kaydedilir.

#### 4.3.4.3 Sunucu Kontrol Programı

Sunucu kontrol programı, bilgelerin özellikleri gibi verilerde değişiklik yapıldığında kapatılıp açılması gereken sunucuyu kolayca resetlemek için oluşturulmuştur. Ancak sonradan eklenen, çöken sunucuyu algılayıp yeniden başlatabilme özelliği sayesinde, oyunun oynanması sırasında oluşan istisnai hatalar için oldukça geçerli bir çözüm olmuştur.

Yazılım, UNIX tabanlı işletim sistemleri için geliştirilmiştir. Sunucuların ekrana uyarı olarak verdiği bilgileri logs sekmelerinde gösterdiği için terminal ekranına geçiş yapma ihtiyacını azaltmıştır. Sunucunun yoğunluk durumuna göre, çökme kontrolünün sıklığı artırılarak istenilen güvenlik düzeyi ayarlanabilmektedir.

ip pid:2310 Refresh Time

Sunucu Kapat

Otomatik Baslat

Login

Char

Map

Logs

**Şekil 4.135:** Sunucu kontrol programı.

## 5. BULGULAR

Bu bölümde öncelikle deney ve kontrol grubu öğrencilerinin uygulama öncesinde aralarında anlamlı bir fark olup olmadığı incelenmiştir. Daha sonra yapılan uygulamanın öğrencilerin fizik ve bilgisayar tutumlarında, başarı puanlarında ve kavramsal anlama ve emin olma düzeylerinde bir değişime yol açıp açmadığına bakılmıştır. Bilgisayar tutumunun, başarı ve fizik tutumu ile ilişkisi incelenmiştir. Okul türünün başarıya olan etkisi, elektroGame oyunundaki başarı düzeyi ile diğer ölçek sonuçlarının ilişkili olup olmadığı ve cinsiyetin etkisi de bu bölümde incelenmiştir. Son olarak elektroGame hakkında öğrenci görüşlerine yer verilmiştir.

### 5.1 Uygulama Öncesinde Deney ve Kontrol Grubu Öğrencileri Arasındaki Farklılıklar

“Deney ve kontrol grubu öğrencileri arasında, uygulama öncesinde bilgisayar tutumları, fizik tutumları ve başarı puanları açısından istatistiksel olarak anlamlı bir fark var mıdır?” başlığı altında incelenmiştir.

Deney ve kontrol grubu öğrencilerinin arasında uygulama öncesinde bilgisayar tutumları, fizik tutumları ve ön-test başarı puanları arasında istatistiksel olarak anlamlı bir fark olup olmadığını belirlemek için veriler ilişkisiz örneklem t-testi (independent samples t-test) ile incelenmiştir. Sonuçlar **Tablo 5.1**'de verilmiştir.

**Tablo 5.1:** Deney ve kontrol grubu öğrencilerinin ön-test sonuçlarının karşılaştırılması.

Okul	Ölçek	Gruplar	N	$\bar{X}$	S	sd	t	p
01	Bilgisayar Tutum	Deney Grubu	22	85,09	12,839	41	0,943	0,351
		Kontrol Grubu	21	88,29	8,923			
	Fizik Tutum	Deney Grubu	22	68,59	14,950	41	0,910	0,928
		Kontrol Grubu	21	69,59	17,755			
	Fizik Motivasyon	Deney Grubu	22	102,59	11,228	41	0,160	0,874
		Kontrol Grubu	21	102,14	6,398			
	Başarı (Test)	Deney Grubu	23	7,09	2,193	43	0,614	0,543
		Kontrol Grubu	22	6,68	2,234			
	Başarı (Kavram)	Deney Grubu	23	5,91	2,429	43	0,481	0,633
		Kontrol Grubu	22	5,59	2,039			
	Başarı (Emin olma)	Deney Grubu	23	7,23	3,033	43	0,470	0,641
		Kontrol Grubu	22	6,74	3,903			



**Tablo 5.1**'in devamı.

Okul	Ölçek	Gruplar	N	$\bar{X}$	S	sd	t	p
O2	Bilgisayar Tutum	Deney Grubu	24	83,54	15,209	46	0,35	0,972
		Kontrol Grubu	24	83,42	8,637			
	Fizik Tutum	Deney Grubu	24	56,54	20,076	46	0,090	0,928
		Kontrol Grubu	24	56,08	14,626			
	Başarı (Test)	Deney Grubu	24	5,63	1,740	46	0,255	0,799
		Kontrol Grubu	24	5,75	1,648			
	Başarı (Kavram)	Deney Grubu	24	4,13	2,271	46	0,138	0,891
		Kontrol Grubu	24	4,21	1,911			
	Başarı (Emin olma)	Deney Grubu	24	6,75	3,710	46	0,485	0,630
		Kontrol Grubu	24	7,21	2,766			

Hem O1 hem de O2 okullarından seçilen deney ve kontrol gruplarının, bilgisayar tutumu, fizik tutumu ve başarı (test, kavramsal anlama düzeyi, emin olma düzeyi) faktörleri açısından uygulama öncesinde birbirlerine denk olduğu görülmüştür.

Kolmogorov-Smirnov Testi sonucunda bilgisayar tutumu, fizik tutumu ve başarı (test) ölçeklerine göre grupların normal bir dağılım gösterdiği görülmüştür.

## 5.2 Uygulama Öncesi ve Sonrasında Değişim Var Mıdır?

İkinci olarak, “Deney ve kontrol gurubu öğrencilerinin, uygulama öncesi bilgisayar tutumu, fizik tutumu ve ön-test puanları ile uygulama sonrası bilgisayar tutumu, fizik tutumu ve son-test puanları arasında anlamlı bir fark var mıdır?” sorusunun cevabı aranmıştır.

Bu araştırma sorusuna ait alt sorular için, karışık ölçümler için iki faktörlü ANOVA testinden elde edilen bulgular sırasıyla verilmiştir.

Ön-test ve son-test olarak uygulanan başarı testi üç aşamalı olarak incelenmiştir (başarı, kavramsal anlama, emin olma). İlk üç bulgu, başarı testinin bu üç bileşenine ait bulgulardır.

- Ön-test ve son-test başarı puanları arasında anlamlı fark var mıdır?

**Tablo 5.2:** Başarı testi ortalama ve standart sapma değerleri.

Grup	Ön-test			Son-test		
	N	$\bar{X}$	S	N	$\bar{X}$	S
Deney	47	6,34	2,09	47	7,15	2,90
Kontrol	46	6,20	1,99	46	6,37	3,72

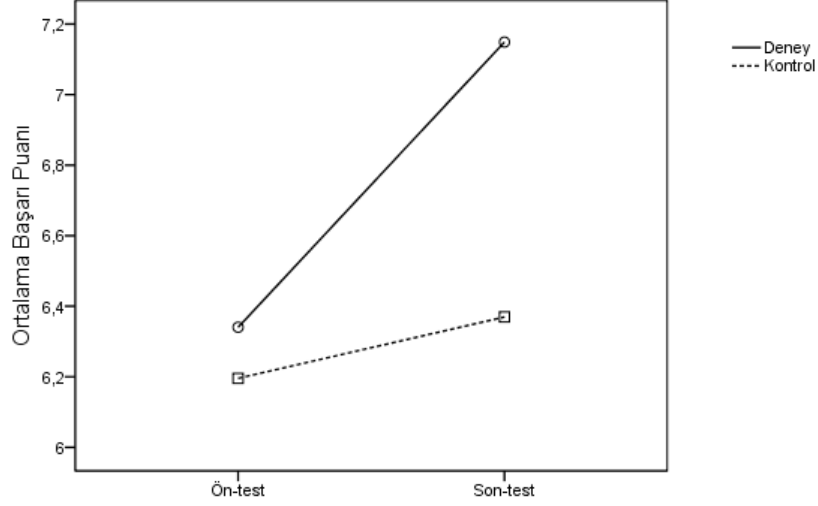
Deney grubu öğrencilerinin uygulama öncesi ortalama puanı 6,34 iken bu ortalama uygulama sonrasında 7,15 olarak ölçülmüştür. Kontrol grubu öğrencilerinin uygulama öncesindeki ve sonrasındaki ortalama puanları 6,20 ve 6,37 şeklindedir. Buna göre uygulamanın deney grubu öğrencilerinin başarı testi ortalamalarında bir artış gözlemlendiği söylenebilir.

Söz konusu değişimin anlamlı bir farklılık gösterip göstermediğinin belirlenmesi için yapılan iki faktörlü ANOVA testi sonuçları **Tablo 5.3**'te verilmiştir.

**Tablo 5.3:** Deney kontrol grubu kavramsal anlama düzeyi ön-test son-test için iki faktörlü ANOVA sonuçları.

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	P
Denekler Arası	898,452	92			
Deney/Kontrol	9,927	1	9,927	1,020	0,315
Hata	885,525	91	9,731		
Denekler İçi	516,843	93			
Ön-test/Son-test	11,219	1	11,219	2,038	0,157
<b>Grup*Ölçüm</b>	<b>4,681</b>	<b>1</b>	<b>4,681</b>	<b>0,850</b>	<b>0,359</b>
Hata	500,943	91	5,505		
Toplam	1415,295	185			

Deney ve kontrol grubu öğrencilerinin başarı ön-test son-test değerlendirmelerinde anlamlı bir fark gözlenmemiştir ( $F=0,850$ ;  $p>0,05$ ). Bu sonuca göre, elektroGame oynayan deney grubu öğrencileri, oynamayan kontrol grubu öğrencilerine göre daha yüksek başarı puanları elde etmişlerdir (**Şekil 5.1**). Ancak bu fark istatistiksel olarak anlamlı çıkmamıştır.



**Şekil 5.1:** Başarı testi ön-test son-test puan ortalamaları.

- Ön-Test ve son-test kavramsal anlama düzeyi puanları arasında anlamlı fark var mıdır?

**Tablo 5.4:** Kavramsal anlama düzeyi ortalama ve standart sapma değerleri.

Grup	Ön-test			Son-test		
	N	$\bar{X}$	S	N	$\bar{X}$	S
Deney	47	5,00	2,49	47	5,87	3,21
Kontrol	46	4,87	2,07	46	5,72	3,54

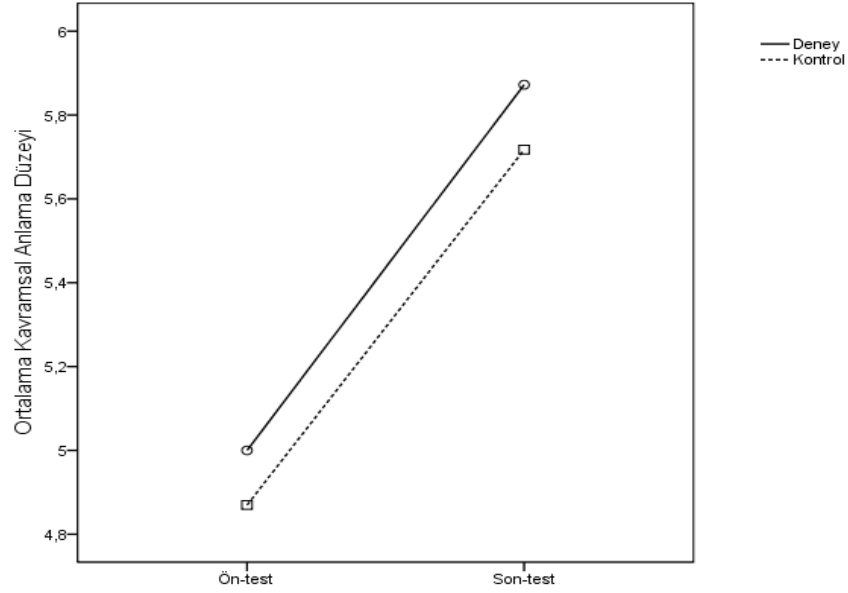
Kavramsal anlama düzeyinde deney grubu öğrencilerinin uygulama öncesi ortalama puanı 5,00 iken bu ortalama uygulama sonrasında 5,87'ye yükselmiştir. Kontrol grubu öğrencilerinin uygulama öncesindeki ve sonrasındaki ortalama puanları 4,87'den 5,72'ye değişmiştir. Buna göre iki grupta da ortalama puanlarda bir artış olduğu söylenebilir.

Bu değişimin anlamlı olup olmadığının belirlenmesi için yapılan iki faktörlü ANOVA testi sonuçları **Tablo 5.5**'te verilmiştir.

**Tablo 5.5:** Deney kontrol grubu kavramsal anlama ön-test son-test iki faktörlü ANOVA sonuçları.

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	P
Denekler Arası	1075,14	92			
Deney/Kontrol	0,947	1	0,947	0,080	0,778
Hata	1074,193	91	1074,193		
Denekler İçi	477,985	93			
Ön-test/Son-test	34,394	1	34,394	7,056	0,009
<b>Grup*Ölçüm</b>	<b>0,007</b>	<b>1</b>	<b>0,007</b>	<b>0,001</b>	<b>0,970</b>
Hata	443,584	91	4,875		
Toplam	1553,125	185			

Deney ve kontrol grubu öğrencilerinin kavramsal anlama düzeyi ön-test son-test değerlendirmelerinde anlamlı bir fark gözlenmemiştir ( $F=0,001$ ;  $p>0,05$ ). İki grubun da kavramsal anlama düzeylerinin birbirine benzer şekilde değiştiği **Şekil 5.2**'de görülmektedir.



**Şekil 5.2:** Kavramsal anlama düzeyi ön-test son-test puan ortalamaları.

- Ön-Test ve son-test emin olma düzeyleri arasında anlamlı fark var mıdır?

**Tablo 5.6:** Emin olma düzeyi ortalama ve standart sapma değerleri.

Grup	Ön-test			Son-test		
	N	$\bar{X}$	S	N	$\bar{X}$	S
Deney	47	6,74	3,359	47	8,38	3,745
Kontrol	46	7,22	3,319	46	7,00	4,619

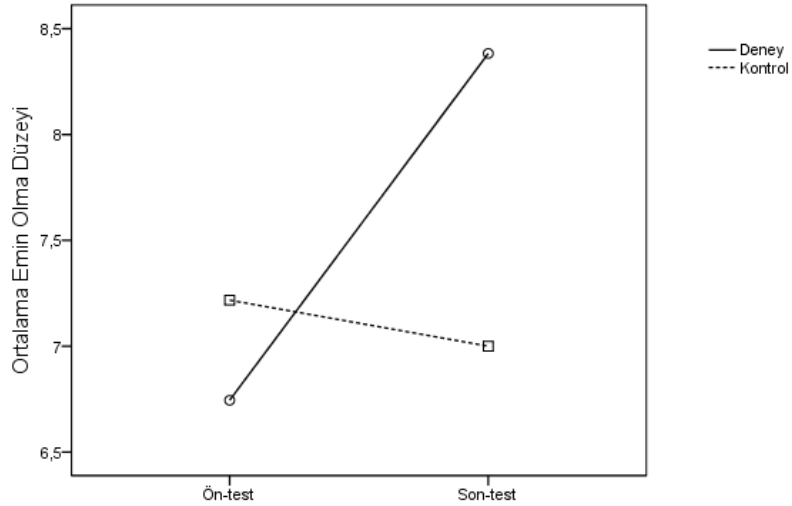
Emin olma düzeyinde deney grubu öğrencilerinin uygulama öncesi ortalama puanı 6,74 iken bu ortalama, uygulama sonrasında 8,38'e yükselerek yaklaşık iki puan artmıştır. Kontrol grubu öğrencilerinin uygulama öncesindeki ve sonrasındaki ortalama puanlarının 7,22'den 7,00'ye düştüğü gözlenmektedir. Buna göre, deney grubunun emin olma düzeyinde artış olurken kontrol grubu öğrencilerinin düzeylerinde düşüş görülmüştür.

Bu değişmelerin anlamlı olup olmadığının belirlenmesi için yapılan iki faktörlü ANOVA testi sonuçları **Tablo 5.7**'de verilmiştir.

**Tablo 5.7:** Deney kontrol grubu emin olma düzeyi ön-test son-test iki faktörlü ANOVA sonuçları.

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	P
Denekler Arası	1595,161	92			
Deney/Kontrol	9,631	1	9,631	0,553	0,459
Hata	1585,530	91	1585,530		
Denekler İçi	1097,834	93			
Ön-test/Son-test	23,468	1	23,468	2,065	0,154
<b>Grup*Ölçüm</b>	<b>40,027</b>	<b>1</b>	<b>40,027</b>	<b>3,522</b>	<b>0,064</b>
Hata	1034,339	91	11,366		
Toplam	2692,995	185			

Deney ve kontrol grubu öğrencilerinin emin olma düzeyi ön-test son-test değerlendirmelerinde anlamlı bir fark gözlenmemiştir ( $F=3,522$ ;  $p>0,05$ ). İstatistiksel olarak anlamlı bir fark çıkmasa bile emin olma düzeyinin kontrol grubu ve deney grubu öğrencilerine göre değişiminin grafiği **Şekil 5.3**'te görüldüğü gibi deney grubu öğrencilerinin lehine olduğu söylenebilir.



**Şekil 5.3:** Emin olma düzeyi ön-test son-test puan ortalamaları.

- Uygulama öncesi fizik tutumları ile uygulama sonrası fizik tutumları arasında anlamlı fark var mıdır?

**Tablo 5.8:** Fizik tutum ölçeği ön-test son-test ortalamaları ve standart sapma değerleri.

Grup	Ön-test			Son-test		
	N	$\bar{X}$	S	N	$\bar{X}$	S
Deney	46	62,30	18,637	46	63,20	18,942
Kontrol	43	61,86	17,268	43	60,44	15,867

Deney grubu öğrencilerinin uygulama öncesi fizik tutumu ortalama puanı 62,30'dan 63,20'ye yükselmiştir. Kontrol grubu öğrencilerinin uygulama

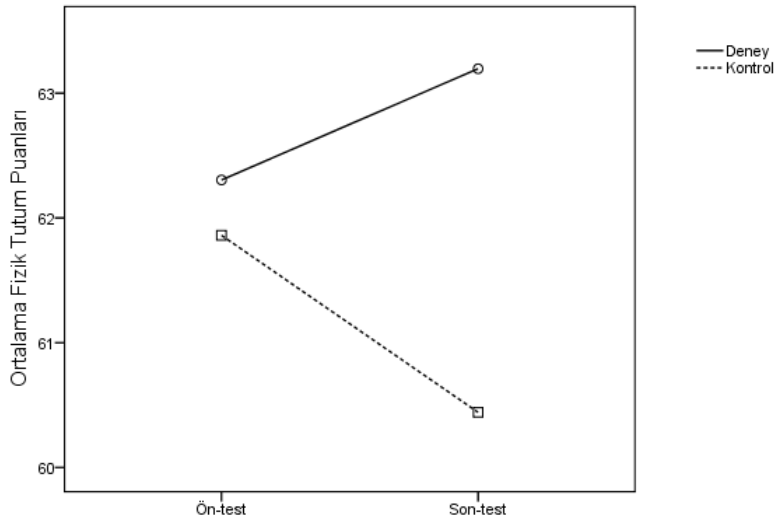
öncesindeki ve sonrasındaki ortalama puanlarının 61,86'den 60,44'e düştüğü gözlenmektedir.

Bu değişmelerin anlamlı olup olmadığının belirlenmesi için yapılan iki faktörlü ANOVA testi sonuçları **Tablo 5.9**'de verilmiştir.

**Tablo 5.9:** Deney kontrol grubu fizik tutumu ön-test son-test iki faktörlü ANOVA sonuçları.

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	P
Denekler Arası	50188,91	88			
Deney/Kontrol	113,625	1	113,625	0,197	0,658
Hata	50075,285	87	575,578		
Denekler İçi	4859,843	89			
Ön-test/Son-test	3,090	1	3,090	0,056	0,813
<b>Grup*Ölçüm</b>	<b>59,292</b>	<b>1</b>	<b>59,292</b>	<b>1,075</b>	<b>0,303</b>
Hata	4797,461	87			
Toplam	55048,75	177			

Deney ve kontrol grubu öğrencilerinin fizik tutumları ön-test son-test değerlendirmelerinde anlamlı bir fark gözlemlenmemiştir ( $F=1,075$ ;  $p>0,05$ ). Uygulamanın toplam dört hafta sürdüğü ve tutumların kısa zamanda değişmediği bilgisi göz önüne alındığında, ortalamaların uygulama öncesi ve sonrasında birbirine yakın olması beklentiye uygundur. Fizik tutumu ortalamalarının kontrol grubu ve deney grubu öğrencilerine göre değişiminin grafiği **Şekil 5.4**'te sunulmuştur.



**Şekil 5.4:** Fizik tutumu deney ve kontrol gruplarının ön-test son-test ortalama puanları.

- Uygulama öncesi bilgisayar tutumları ile uygulama sonrası bilgisayar tutumları arasında anlamlı fark var mıdır?

**Tablo 5.10:** Bilgisayar tutum ölçeği ön-test son-test ortalamaları ve standart sapma değerleri.

Grup	Ön-test			Son-test		
	N	$\bar{X}$	S	N	$\bar{X}$	S
Deney	46	84,28	13,992	46	82,00	15,042
Kontrol	43	85,84	8,960	43	84,12	9,160

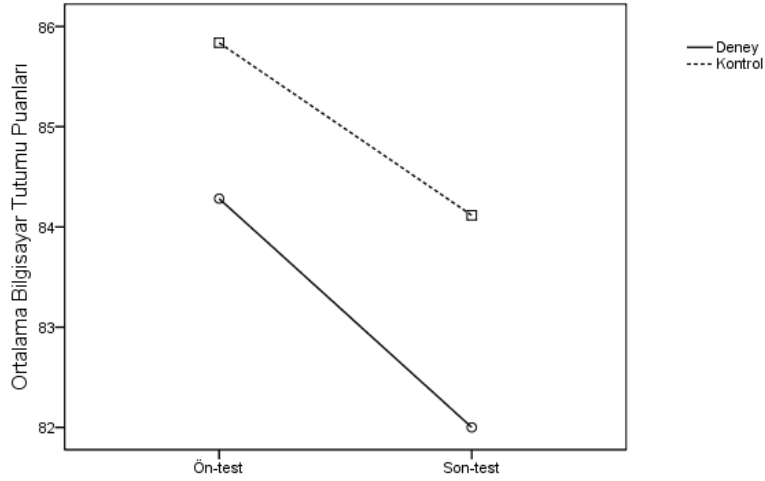
Bilgisayar tutumu ortalama puanları deney grubu öğrencilerinin uygulama öncesi ortalama puanı 84,28 iken bu ortalama uygulama sonrasında 82,00'ye düşmüştür. Kontrol grubu öğrencilerinin uygulama öncesindeki ve sonrasındaki ortalama puanlarının 85,84'ten 8,12'ye düştüğü gözlenmektedir. Buna göre deney ve kontrol grubunun bilgisayara yönelik tutumlarında düşüş oluşmuştur.

Bu değişmelerin anlamlı olup olmadığının belirlenmesi için yapılan iki faktörlü ANOVA testi sonuçları **Tablo 5.11**'te verilmiştir.

**Tablo 5.11:** Deney kontrol grubu bilgisayar tutumları ön-test son-test iki faktörlü ANOVA sonuçları.

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	P
Denekler Arası	23228,36	88			
Deney/Kontrol	149,743	1	149,743	0,564	0,454
Hata	23078,617	87	265,271		
Denekler İçi	2990,607	89			
Ön-test/Son-test	178,112	1	178,112	5,517	0,021
<b>Grup*Ölçüm</b>	<b>3,506</b>	<b>1</b>	<b>3,506</b>	<b>0,109</b>	<b>0,743</b>
Hata	2808,989	87			
Toplam	26218,97	177			

Deney ve kontrol grubu öğrencilerinin bilgisayar tutumu ön-test son-test değerlendirmelerinde anlamlı bir fark gözlemlenmemiştir ( $F=0,109$ ;  $p>0,05$ ). Aynı şekilde uygulamanın toplam dört hafta sürmesi ve tutumların kısa zamanda değişmemesi bilgisayar tutum ortalamalarının da uygulama öncesi ve sonrasında birbirine yakın olması beklentisi ile uyumludur. Kontrol grubu ve deney grubu öğrencilerinin bilgisayar tutumlarının değişim grafiği **Şekil 5.5**'de verilmiştir.



**Şekil 5.5:** Bilgisayar tutumu, deney ve kontrol gruplarının ön-test son-test ortalama puanları.

### 5.3 Deney Grubu Öğrencilerinde Okul Türünün Etkisi

Dördüncü araştırma sorusunda, deney grubu öğrencilerinin, uygulama öncesi ve uygulama sonrasında, bilgisayar tutumunda, fizik tutumunda, başarı, kavramsal anlama düzeyi ve emin olma düzeyilerindeki değişim puanları ile elektroGame oyunundaki seviyelerinin arasında **okul türlerine** göre anlamlı bir fark olup olmadığı incelenmiştir. Verilerin analizinde iki ilişkisiz örneklem ortalamaları arasındaki farkın anlamlı olup olmadığını test etmek için *ilişkisiz örneklem için t-testi* kullanılmıştır.

- Deney grubu öğrencilerinin, bilgisayar tutumlarında uygulama öncesi ve uygulama sonrasında, okul türüne göre anlamlı bir fark var mıdır?

**Tablo 5.12:** Bilgisayar tutumu değişim puanlarının okul türlerine göre t-testi sonuçları.

Grup	N	$\bar{X}$	S	sd	t	p
O1	22	1,09	11,747	44	0,816	0,419
O2	24	3,38	6,788			

Bilgisayar tutumlarındaki değişim bakımından iki okul öğrencileri arasında anlamlı bir fark gözlenmemiştir ( $t=0,816$ ;  $p>0,05$ ). O1 öğrencilerinin O2 öğrencilerine göre tutumlarında olumlu yönde daha fazla değişim olmuştur.

- Deney grubu öğrencilerinin, fizik tutumlarında uygulama öncesi ve uygulama sonrasında, okul türüne göre anlamlı bir fark var mıdır?



**Tablo 5.13:** Fizik tutumu deęişim puanlarının okul türlerine göre t-testi sonuçları.

Grup	N	$\bar{X}$	S	sd	t	p
O1	22	2,3182	9,687	44	0,806	0,425
O2	24	0,4167	12,938			

Fizik tutumlarındaki deęişim bakımından iki okul öğrencileri arasında anlamlı bir fark gözlenmemiştir ( $t=0,806$ ;  $p>0,05$ ). O1 öğrencilerinin O2 öğrencilerine oranla fizik tutumları pozitif yönde daha fazla deęişim göstermiştir.

- Deney grubu öğrencilerinin, başarı testi deęişim puanlarında okul türüne göre anlamlı bir fark var mıdır?

**Tablo 5.14:** Başarı testi deęişim puanlarının okul türlerine göre t-testi sonuçları.

Grup	N	$\bar{X}$	S	sd	t	p
O1	23	2,00	2,335	45	1,765	0,084
O2	24	0,88	2,028			

Başarı testi deęişim puanları okul türlerine göre anlamlı bir farklılık ( $t=1,765$ ;  $p>0,05$ ) göstermemekle birlikte, O1 öğrencilerinin deęişim puanlarının O2 öğrencilerinin deęişim puanlarından daha yüksek olduğu görülmektedir.

- Deney grubu öğrencilerinin, kavramsal anlama düzeyi deęişim puanlarında okul türüne göre anlamlı bir fark var mıdır?

**Tablo 5.15:** Kavramsal anlama deęişim puanlarının okul türlerine göre t-testi sonuçları.

Grup	N	$\bar{X}$	S	sd	t	p
O1	23	2,35	2,854	45	2,620	0,012
O2	24	0,42	2,165			

O1 öğrencilerin kavramsal anlama düzeyleri, O2 öğrencilerinin kavramsal anlama düzeyinden istatistiksel olarak anlamlı olacak şekilde artmıştır ( $t=2,620$ ;  $p<0,05$ ).

- Deney grubu öğrencilerinin, emin olma düzeyi deęişim puanlarında okul türüne göre anlamlı bir fark var mıdır?

**Tablo 5.16:** Emin olma düzeyi deęişim puanlarının okul türlerine göre t-testi sonuçları.

Grup	N	$\bar{X}$	S	sd	t	p
O1	23	2,83	3,962	45	1,124	0,267
O2	24	1,67	3,074			

Emin olma düzeyi bakımından iki okul öğrencileri arasında anlamlı bir fark gözlenmemiştir ( $t=2,620$ ;  $p>0,05$ ). Aynı şekilde fark istatistiksel açıdan anlamlı olmasa bile O1 öğrencilerinin emin olma düzeyinin O2 öğrencilerinden daha fazla artış gösterdiği görülmektedir.

- Deney grubu öğrencilerinin, elektroGame oyunundaki seviyelerinde okul türüne göre anlamlı bir fark var mıdır?

**Tablo 5.17:** elektroGame seviyelerinin okul türlerine göre t-testi sonuçları.

Grup	N	$\bar{X}$	S	sd	t	p
O1	23	11,48	11,004	45	2,030	0,048
O2	24	6,79	2,587			

O1 öğrencilerinin elektroGame oyunundaki seviyeleri O2 öğrencilerinin seviyelerine göre anlamlı bir şekilde farklıdır ( $t=2,030$ ;  $p<0,05$ ). İki okul arasındaki ortalamalara bakıldığında da bu farkın O1 lehine olduğu görülmektedir.

#### 5.4 Deney Grubu Puanlarındaki İlişkiler

Üçüncü araştırma sorusunda, deney grubu öğrencilerinin, elektroGame oyunundaki başarı düzeyleri ile bilgisayar tutumları, fizik tutumları, başarı testi, kavramsal anlama düzeyi ve emin olma düzeyi değişim puanları arasında istatistiksel olarak anlamlı bir ilişki olup olmadığı araştırılmıştır.

- Deney grubu öğrencilerinin bilgisayar tutumları ile elektroGame oyunundaki başarı düzeyleri arasında anlamlı bir ilişki var mıdır?
- Deney grubu öğrencilerinin, fizik tutumları ile elektroGame oyunundaki başarı düzeyleri arasında anlamlı bir ilişki var mıdır?
- Deney grubu öğrencilerinin, başarı testi değişim puanları ile elektroGame oyunundaki başarı düzeyleri arasında anlamlı bir ilişki var mıdır?
- Deney grubu öğrencilerinin, kavramsal anlama değişim düzeyi ile elektroGame oyunundaki başarı düzeyleri arasında anlamlı bir ilişki var mıdır?
- Deney grubu öğrencilerinin, emin olma düzeyi değişim puanları ile elektroGame oyunundaki başarı düzeyleri arasında anlamlı bir ilişki var mıdır?

**Tablo 5.18:** elektroGame seviyeleri ile deęişim puanları arasındaki ilişki.

		Oyun Seviyesi	Başarı Testi	Kavramsal Anlama	Emin Olma	Fizik Tutum	Bilgisayar Tutum
Oyun Seviyesi	Pearson K.	1	,079	,473**	,315*	-,023	,062
	Anlamlılık		,598	,001	,031	,878	,682
	N	47	47	47	47	46	46
Başarı Testi	Pearson K.	,079	1	,634**	,421**	,277	,019
	Anlamlılık	,598		,000	,003	,063	,900
	N	47	47	47	47	46	46
Kavramsal Anlama	Pearson K.	,473**	,634**	1	,386**	,124	,035
	Anlamlılık	,001	,000		,007	,410	,815
	N	47	47	47	47	46	46
Emin Olma	Pearson K.	,315*	,421**	,386**	1	-,057	,034
	Anlamlılık	,031	,003	,007		,708	,823
	N	47	47	47	47	46	46
Fizik Tutum	Pearson K.	-,023	,277	,124	-,057	1	-,186
	Anlamlılık	,878	,063	,410	,708		,216
	N	46	46	46	46	46	46
Bilgisayar Tutum	Pearson K.	,062	,019	,035	,034	-,186	1
	Anlamlılık	,682	,900	,815	,823	,216	
	N	46	46	46	46	46	46

elektroGame oyunu seviyesi ile kavramsal anlama düzeyi deęişim puanları arasında ( $r=0,473$ ;  $p<0,01$ ), başarı testi deęişim puanları ile emin olma düzeyi deęişim puanları arasında ( $r=0,421$ ;  $p<0,01$ ), başarı testi deęişim puanları ile kavramsal anlama düzeyi deęişim puanları arasında ( $r=0,634$ ;  $p<0,01$ ) ve kavramsal anlama düzeyi deęişim puanları ile emin olma düzeyi deęişim puanları arasında ( $r=0,386$ ;  $p<0,01$ ) pozitif yönde kuvvetli bir ilişki vardır.

elektroGame oyunu seviyesi ile emin olma düzeyi deęişim puanları arasında ( $r=0,315$ ;  $p<0,05$ ) olacak şekilde pozitif yönlü bir ilişki vardır.

Diğer bileşenler arasında anlamlı bir ilişki gözlenmemiştir.

## 5.5 Cinsiyetin Deney Grubu Öğrencilerine Etkileri

Altıncı araştırma sorusu olarak, deney grubu öğrencilerinin elektroGame oyunundaki başarı düzeylerinin, başarı testi, kavramsal anlama ve emin olma düzeyi

değişim puanlarının cinsiyete göre anlamlı farklılık gösterip göstermediği incelenmiştir. Verilerin analizinde iki ilişkisiz örneklem ortalamaları arasındaki farkın anlamlı olup olmadığını test etmek için *ilişkisiz örneklem için t-testi* kullanılmıştır.

- Deney grubu öğrencilerinin, elektroGame oyunu başarısı cinsiyete göre anlamlı bir fark göstermekte midir?

**Tablo 5.19:** Cinsiyetin elektroGame başarısına etkisinin t-testi sonuçları.

Cinsiyet	N	$\bar{X}$	S	sd	t	p
Erkek	17	13,53	11,325	45	3,047	0,004
Kız	30	6,57	4,141			

Cinsiyetin, elektroGame oyunu seviyelerinde ilerlemeye anlamlı bir etkisinin olduğu gözlenmiştir ( $t=3,047$ ;  $p<0,01$ ). Yaklaşık iki katı ortalama ile bu fark erkek öğrenciler lehinedir.

- Deney grubu öğrencilerinin başarı testi değişim puanları ile cinsiyetleri arasında anlamlı bir fark var mıdır?

**Tablo 5.20:** Cinsiyetin fizik başarısına etkisinin t-testi sonuçları.

Cinsiyet	N	$\bar{X}$	S	sd	t	p
Erkek	17	1,59	1,698	45	0,372	0,711
Kız	30	1,53	2,510			

Cinsiyetin, başarı testindeki değişime anlamlı bir etkisinin olmadığı görülmektedir ( $t=0,372$ ;  $p>0,05$ ).

- Deney grubu öğrencilerinin, kavramsal anlama düzeyleri değişim puanları ile cinsiyetleri arasında anlamlı bir fark var mıdır?

**Tablo 5.21:** Cinsiyetin kavramsal anlama düzeyine etkisinin t-testi sonuçları.

Cinsiyet	N	$\bar{X}$	S	sd	t	p
Erkek	17	1,94	2,561	45	1,118	0,269
Kız	30	1,03	2,735			

Cinsiyetin, kavramsal anlama düzeyindeki değişime anlamlı bir etkisinin olmadığı görülmektedir ( $t=1,118$ ;  $p>0,05$ ).

- Deney grubu öğrencilerinin, emin olma düzeyleri ile cinsiyetleri arasında anlamlı bir fark var mıdır?

**Tablo 5.22:** Cinsiyetin emin olma düzeyine etkisinin t-testi sonuçları.

Cinsiyet	N	$\bar{X}$	S	sd	t	p
Erkek	17	1,94	3,418	45	0,422	0,675
Kız	30	2,40	3,663			

Cinsiyetin, emin olma düzeyindeki değişime anlamlı bir etkisinin olmadığı görülmektedir ( $t=0,422$ ;  $p>0,05$ ).

## 5.6 Öğrenci Görüşleri

Yedinci araştırma sorusunda, hazırlanan elektroGame oyununun eğitimde kullanımına ilişkin öğrenci görüşleri incelenmiştir. Uygulama sonrasında deney gruplarından rasgele seçilen sekiz öğrenci ile yüz yüze görüşme yapılmıştır.

Görüşme soruları 2 bölümden oluşmuştur.

Öğrencilerin görüşme sorularına yönelik cevapları, kimlikleri gizli tutularak verilmiştir. Bu amaçla öğrenciler 1k, 2e gibi kısaltmalarla anılmıştır. Öğrencilerin sorulara verdiği cevaplar değişiklik yapılmadan aynen aktarılmıştır.

### 5.6.1 Bölüm I: Önceki deneyimler

Öğrencilere *her zaman ulaşabilecekleri bilgisayar ve internet bağlantılarının olup olmadığı ve bilgisayarı daha çok hangi amaçla kullandıkları sorulmuştur.*

- Öğrencilerin büyük çoğunluğunun kendi bilgisayarının olduğu, kendi bilgisayarı olmayan öğrencilerin ise kolayca bir yakınlarının bilgisayarı kullanabildiği veya okulda bilgisayara erişebildiği görülmüştür. Ulaşılabilen ve sahip olunan bilgisayarların çoğu internete bağlanabilmektedir.
- Bilgisayarı kullanma amacı çoğunlukla internet üzerinden sohbet etmek, film izlemek, müzik dinlemek, oyun oynamak ve ödevler hakkında araştırma yapmak şeklinde belirtilmiştir.

Öğrencilere *oyun oynayıp oynamadıkları, oynuyorlarsa hangi tür oyunları tercih ettikleri sorulmuştur.*

- Erkek öğrencilerin tamamı bilgisayar oyunu oynadıklarını ifade ederken, kız öğrencilerin sadece bir tanesi eskiden oynadığını ancak artık oyun oynamadığını belirtmiştir.
- Oyun oynadığını söyleyen erkek öğrencilerin çoğunluğu FPS tarzı macera oyunlarını tercih ettiğini söylemiştir.

Öğrencilere *genel olarak fizik dersi hakkındaki düşünceleri ve basit elektrik devreleri konusunda kendilerini ne düzeyde gördükleri* sorulmuştur.

- Öğrenciler fizik dersini zor bulmakla birlikte fizik dersi hakkında olumlu görüşler bildirmişlerdir. 9. sınıf fizik konularının eski konularına göre daha kolay olduğunu belirtmişlerdir.
- Basit elektrik devreleri konusu, öğrenciler tarafından zor bir konu olarak ifade edilmesine karşın eğlenceli de bulunmuştur.
- Öğrenciler, elektroGame oyununu oynamanın basit elektrik devreleri konusuna daha sıcak bakmalarına neden olduğu konusunda ortak görüş belirtmişlerdir.

### 5.6.2 Bölüm II: elektroGame hakkındaki öğrenci görüşleri

Bu bölümde öğrencilerin elektroGame hakkında görüşleri alınmıştır. Sorular ve bu sorulardan elde edilen çözümlenmiş veriler ve yorumlar sırasıyla yer almaktadır.

1. elektroGame hakkındaki ilk izlenimlerin nelerdir?

**Tablo 5.23:** elektroGame hakkındaki izlenimlerin tema ve frekansları.

Temalar	f
Öğrenmeyi eğlenceli hale getirme	4
Gerçeklik hissi	2
Alışlagelmiş oyunlardan farklı oluşu	2
Karmaşık yapısı	3
Daha iyi oyun grafiği beklentisi	2
Amacına uygunluk	1

Öğrencilerin yarısı elektroGame'in konuyu öğrenmeyi eğlenceli hale getirdiğini ifade etmiştir. Oyundaki gerçeklik hissi, oyunun alışlagelmiş oyunlardan

farklı oluşu ifade edilirken; yapının karmaşık olduğu ve grafiklerin daha iyi olabileceği belirtilmiştir.

8e kodlu öğrenci bu konuda “*Bence zevkliydi. Eğitim almak zorundayız zaten ama oyun oynayarak olunca, ikisi birleşince iyi oluyor.*” şeklinde görüşünü belirtmiştir.

## 2. Oyunda hoşuna giden şeyler nelerdir?

**Tablo 5.24:** Oyunda hoşça giden temalar ve frekanslar.

Temalar	f
Maceracı oyun yapısı	3
Yardımlaşma hissi	2
Özgün karakterler ve oyun yapısı	2
Öğretici olması	3
Bilginin sunumunun iyi olması	1
Oynanabilirliğin keyifli olması	4

Öğrencilerin büyük çoğunluğu oyunun keyif veren maceracı bir yapısı olduğunu ifade etmiştir. Ayrıca oyun aracılığıyla bilginin iyi sunulması ve öğretici olması, karakterlerin özgün olması ve yardımlaşabilme hissi oyunda olumlu sayılabilecek görüşler olarak ifade edilmiştir.

Oyunu oynayan öğrencilere oyunda en çok nelerden hoşlandıkları sorulduğunda, karakterlerini özelleştirebilme ve devre penceresinde simülasyon yapabilme özelliklerinin öne çıktığı görülmektedir. Ayrıca, oyuncuların birbirleriyle sohbet penceresi yardımıyla kurdukları iletişim hoşça giden özelliklerden birisi olarak belirtilmiştir.

2k kodlu öğrenci “*Hepimizin kendine özgü karakterlerimizin olması çok hoşuma gitti. Biz onlara absürt isimler verdik. Hani böyle biz oyunda kendimizi gösterdik. Hiç tanımadığımız birileriyle konuştuk. Ben ona yardım ettim. O bana yardım etti. Yardımlaşabilmek çok güzel bir şeydi.*” şeklinde görüşünü ifade etmiştir.

## 3. Sıkıcı, zorlayıcı bulduğun, sevmediğin yerler var mı?

**Tablo 5.25:** elektroGame'in olumsuz yönüne ilişkin tema ve frekansları.

Temalar	f
Renkler-Grafikler	4
Oyun içi yönlendirme, görevler ve haritalar	4
Karakterlerde görülen eksiklikler	2
Soruların zorluğu	3

Öğrencilerin yarısı, oyunun renk ve grafiklerinin daha iyi olabileceğini belirtmiştir. Yine öğrencilerin yarısı, oyundaki haritalar ve yönlendirmelerin karışık olduğunu ancak oynadıkça anlaşıldığını ve haritaya hâkim olduğunu söylemiştir. Ayrıca elektroGame'de kullanılan fizik sorularının zor olduğu bazı öğrenciler tarafından ifade edilmiştir.

1k kodlu öğrenci “Renkler çok parlaktı. Çok fazla yeşildi. İlk giriş haritası çok yeşil. Ağaçlar yeşil, yerler yeşil. Ben nereye gideceğimi bilemedim. Hatta arkadaşımın üzerindeki kazak da yeşil. Onu da kaybettim. Hangi doğru kapıdan nereye gideceğimi bulamadım. Bir kapıya gidiyorum bütün devreleri yapmışım. Başka bir tanesine gidiyorum yapmışım. Yapmadığım devreyi bulamıyorum. Haritayı karıştırmıştım ben. Yani o hoşuma gitmedi.” şeklinde görüşünü ifade etmiştir.

4. Eğitim haritasının yeterli olduğunu düşünüyor musun?

Öğrencilerin büyük çoğunluğu elektroGame'in eğitim haritasının yeterli olduğu görüşünde birleşmişlerdir. Ayrıca, harita kullanmada teknik problemlerden kaynaklanan sorunlar yaşandığı ifade edilmiştir.

5e kodlu öğrenci “Oyunu öğrenelim bölümünü geçmek için devreleri falan yapmak için oradan öğrendik hep. Bence her şey açıktı zaten. Öğrenmesi çok kolay oldu.” şeklinde görüşünü ifade etmiştir.

5. Oyun içinde diğer oyuncularla etkileşimin nasıldı?

**Tablo 5.26:** Diğer oyuncularla etkileşim konusunda temalar ve frekanslar.

Temalar	f
Yardımlaşmaya teşvik ediyor	4
Eğlenceli bir iletişim ortamı sağlandı	5
Daha kalabalık olsa daha iyi olurdu	2
Yazılanların kaydedilmesi sıkıntısı	1

Öğrencilerin büyük çoğunluğu, oyunda eğlenceli bir iletişim ortamı sağlandığı konusunda görüş bildirmiş, bu ortamın öğrenciler arasında yardımlaşmayı



desteklediği hususunu ifade etmiştir. Öğrencilerden bir kısmı oyunda çevrim içi oyuncu sayısının daha fazla olması gerektiğini belirtmiştir.

1k kodlu öğrenci “Biriyle şey yapmıştık. Onlar yardımcı oldu. Kuzenim oyun hastasıdır ondan biliyorum Knight online, GTA falan. Ondan biliyorum. Burada da aşağıda pencerede takım arkadaşlarıyla veya herkesle konuşabiliyorlar. Sadece istedikleriyle. Bu iletişim benim çok hoşuma gitti. Sadece oynuyordum sıkıldım bir süre sonra. Arkadaşım ile konuştum sen neredesin dedim. Ben şuradayım dedi. Ben bu kademedeyim. İşte burada daha eğlenceli şeyler var. Bunlar teşvik ediyor. Oyunda bir yerde kaldım. Ben ne yapacağım derken orda bir yardım çubuğu yoksa anlatan birisi olması iyi oluyor.” şeklinde görüşünü ifade etmiştir.

Oyun içi etkileşim konusunda öğrenci görüşlerine örnek oluşturması için elektroGame’in sunucusunda tutulan sohbet kayıtlarından kısa bir bölüm sunulmuştur.

- A hangi görevleri tamamladın  
B bilmiyorum desem :D  
A o zaman beni takip et  
B efelerin ne yapmam gerekiyor  
A içerdeki diğerler görevleri yaptıysan  
A demir cevheri buluncaya kadar savaşmalısın  
B biliyorum ama nasıl savaşacağım?  
A önce A tuşuna bas  
A sonra CTRL  
B denedim ama olmuyor  
A yeterince yakınsan vurmaya başlıyor  
A q klavye kullanıyorsun değil mi?  
B evet, q tipi  
A robota yaklaş  
B ok  
A düşeni al  
A çok şanslısın  
A daha ilk seferinde düşürdün  
A ben 14. de bulmuştum  
A şimdi gel  
B sağ ol  
A :)  
B sen olmasan yapamazdım

Buradan öğrencilerin samimi bir şekilde iletişime geçtikleri ve yardımlaştıkları anlaşılmaktadır.

6. Oyundaki rekabet hissini artırılması konusunda ne düşünüyorsun?

**Tablo 5.27:** Rekabet hissini artırılmasına ilişkin temalar ve frekanslar.

Temalar	f
Heyecan veriyor ve hırslandırıyor	6
Daha büyük gruplar arası rekabet olsa daha iyi olurdu	4
Grup içi öne çıkma dürtüsü	5

Oyundaki rekabet hissini artırılması, öğrencilerin büyük bir bölümü tarafından, heyecan verici ve hırslandırıcı etkiye sebep olabileceği şeklinde değerlendirilmiştir. Grup içinde ön plana çıkma isteğinin, öğrencileri oyuna karşı daha istekli hale getirdiği ifade edilmiştir.

Grupların büyütülmesi öğrenciler açısından arzu edilen bir özellik olarak gözlenmiştir. Öğrencilerin yarısı rekabetin, sadece sınıf arkadaşları arasında değil, farklı okullar hatta iller arasında olması gerektiği konusunda görüş bildirmiştir.

6e kodlu öğrenci “*Bu yaşlarda falan çoğu kişi ben şundan daha iyi olmalıyım dediği için. Grupça oynamak güzel olurdu. O zaman daha fazla vakit ayırırdım.*” şeklinde görüşünü ifade etmiştir.

7. elektroGame’i daha çok oyun olarak mı yoksa ders olarak mı sınıflandırırısın?

**Tablo 5.28:** elektroGame’in ders mi oyun mu olarak nitelendirilmesi.

Temalar	f
Her ikisi de	3
Oyun kısmı daha ağır basıyor	4
Ders kısmı daha ağır basıyor	3
Ne tam olarak oyun ne de ders	1

Öğrencilerin büyük çoğunluğu elektroGame’in oyun mu ders mi olarak nitelendirilebileceği konusunda net bir görüş ifade edememişlerdir.

2k kodlu öğrencinin “*Galiba yani, hem oyun oynuyorum hem bilgi ediniyorum. Oyun lezzetini aldım. Sıkılmadım hem faydalandım. Oyun tadında ders yazılımı denmeli. Yani öyle bence. Oyun oynuyorsunuz ama ders de öğreniyorsunuz.*” şeklindeki görüşü tablodan çıkan sonuçları destekler niteliktedir.

8. Herhangi bir dersi bir sene boyunca elektroGame ile işlemek ister misin?

**Tablo 5.29:** Dersin elektroGame’de işlenmesine ilişkin tema ve frekanslar.

Temalar	f
Evet	3
Destekleyici nitelikte olması daha iyi olur	3
Dersin konusuna göre değişir	1

Öğrencilerin bazıları, dersin tamamının elektroGame’de işlenebileceğini belirtirken bazıları da elektroGame’in destekleyici bir unsur olarak kullanılmasının daha faydalı olabileceği görüşünü ifade etmişlerdir.

5e kodlu öğrenci “Okulda dersleri göreceğiz. Hafta sonları oyun oynayacağız çok güzel olurdu. Ama sadece ders yerine oyun güzel olmazdı. Okuldan sonra bunlarla pratik yapsak çok güzel olurdu.” şeklinde görüş bildirirken 8e kodlu öğrenci ise görüşünü, “Dersine bağlı. Geometri dersinde ne kadar işe yarar bilmiyorum ama bir tarih olsun hafızada kalma amacıyla olsun kullanılırsa olurdu. O zaman oynardım ben. Akılda kalma amacıyla güzel olurdu.” şeklinde ifade etmiştir.

## 6. SONUÇ, TARTIŞMA VE ÖNERİLER

Bu çalışmanın amacı, eğitsel amaçlar için kullanmaya elverişli, macera ve eğlence unsurlarını barındıran çok kullanıcıli bir çevrim içi rol yapma oyunu geliştirmek, geliştirilen oyuna ortaöğretim 9. sınıf fizik dersi programında yer alan “Elektrik ve Manyetizma” ünitesindeki “Basit Elektrik Devreleri” konusunu entegre etmek, bu oyunun öğrenci başarısına olan etkisini incelemek ve elektroGame oynayan öğrencilerin oyun hakkındaki görüşlerini belirlemektir.

Bu bölümde, araştırma sonucunda elde edilen bulguların sonuçlarına ve bu sonuçlar doğrultusunda ileride yapılacak çalışmalara ilişkin önerilere yer verilmiştir.

### 6.1 Sonuçlar

Literatürde yapılan araştırma sonucunda, fikir ya da geliştirme aşamasında çalışmalar olmasına rağmen tamamlanabilmiş çok az çalışma bulunmuştur. Geliştirilmesi planlanan ve geliştirilmeye başlandığı duyurulan çalışmalardan da çoğunun yarım kaldığı belirlenmiştir. Bu durumun temel nedeni, kaliteli bir MMORPG üretmenin çok yüksek maliyetler, büyük bir ekip ve uzun zaman gerektirmesidir. Örneğin, toplam yedi yılda Sega-AM2 tarafından geliştirilmiş olan Shenmue isimli MMORPG 70 milyon dolarlık bir bütçeye sahiptir. Bioware tarafından geliştirilen Star Wars MMORPG'nin ise 100 milyon dolarlık bir bütçesi vardır. Microsoft'un XBOX'ta oldukça popüler olan HALO'nun MMORPG versiyonu için, proje iptal edilinceye kadar, 90 milyon dolar harcandığı açıklanmıştır (MMO Budget Busters, 2012; Thorsen, 2012).

[www.elektrogame.net](http://www.elektrogame.net) adresinde yayınlanmakta olan web sitesinde, araştırmacıların hem kaynak kodlarına hem de kaynak kodda gerekli değişikliklerin nasıl yapılacağını anlatan açıklamalara yer verilmektedir. Böylece benzeri bir eğitsel oyun geliştirmek isteyen ya da elektroGame oyununu farklı alanlara uyarlamak isteyen araştırmacılara açık kaynak kodlu bir MMORPG sunulmuştur.

elektroGame tasarlanırken şiddet ögesi içermeyen, eğlence ögesini de azaltmayan bir yöntem izlenmeye çalışılmıştır.

elektroGame tasarlandıktan sonra oyuna 9. sınıf fizik dersi “Basit Elektrik Devreleri” ünitesi entegre edilmiştir. Deney grubu olarak belirlenen gruplarla, fizik dersi dışında beş hafta süre uygulama yapıldıktan sonra elde bulgulardan ulaşılan sonuçların bazıları şöyledir:

- Uygulama öncesinden ve sonrasına başarı, kavramsal anlama düzeyi ve emin olma düzeyi ortalama puanları, deney grubu öğrencilerinde kontrol grubu öğrencilerine göre istatistiksel olarak anlamlı olmamakla birlikte daha fazla artış göstermektedir.
- Benzer şekilde bilgisayar ve fizik tutumlarında da anlamlı bir değişiklik gözlenmemiştir. Uygulamanın beş hafta gibi kısa zamanda yapıldığı ve tutumların da kısa zaman aralıklarında fazla değişmediği bilgisi göz önüne alındığında, elde edilen sonuç beklentilere uygun çıkmıştır.
- Daha başarılı öğrencilerin bulunduğu O1’deki öğrenciler elektroGame’den olumlu yönde daha fazla etkilenmişlerdir.
- elektroGame oyunundaki başarı açısından bakıldığında, erkek öğrenciler kız öğrencilere göre anlamlı olarak daha başarılıdır.
- Öğrenciler genelde elektroGame’i eğlenceli bulmuşlardır. Çoğunluğu grafiklerin iyileştirilmesini istemiştir.
- Öğrenciler elektroGame’de oynarken sanal karakterlerinin kişiselleştirebilmesi özelliğini beğendiklerini ifade etmişlerdir.
- elektroGame’in en çok beğenilen özelliği devre penceresinde yapılan simülasyonlar olarak öne çıkmıştır.
- Oyun esnasında oyuncuların birbirleriyle yazılı iletişim kurabilme özelliği, öğrencilerin sık kullandıkları ve beğendiklerini ifade ettikleri özelliklerdir.
- elektroGame’e alışmanın ve oyunu kullanmanın kolay olduğu, öğrenciler tarafından ifade edilmiştir.
- Öğrenciler, oyunda her zaman danışabilecekleri, oyunu kendilerinden daha iyi bilen kişiler olmasını beklemektedirler.

- Öğrenciler, sınıf içinde zirvede olmak yerine, sınıf veya okul grubuyla diğer gruplara karşı mücadele etme beklentisi içindedirler.

## 6.2 TARTIŞMA

Bu bölümde, çalışmadan elde edilen sonuçlar eğitsel MMO ve MMORPG türünde bilgisayar oyunları ile yapılmış çalışmalardan elde edilmiş sonuçlarla karşılaştırılmıştır.

İngilizce öğretiminde kullanılmak üzere geliştirilen bir MMORPG ile Suh, Kim ve Kim (2010) tarafından çalışma yapılmıştır. 118’i kontrol 102’si deney grubu olmak üzere toplam 220 öğrenci bulunan çalışma sonucunda, deney grubu öğrencileri kontrol grubu öğrencilerinden daha başarılı olmuşlardır. Ancak bu çalışmada da deney ve kontrol grubu arasında istatistiksel olarak anlamlı bir fark çıkmamıştır. Araştırmacılara göre, seçilen konunun kısa ve basit olması oyunun etkililiğini incelemeye yetersizlik doğurmuştur. Ayrıca, katılan öğrenci sayısının az olması, farkın anlamlı çıkmasını engelleyen başka bir neden olarak sunulmuştur.

elektroGame ortamında 4. basamak bilişim teknolojileri dersi “bilgilerimi neden unutuyorum?” ünitesiyle ilgili bir bölüm hazırlanmıştır. Bu öğrenciler için elektroGame’de ilköğretim seviyesine uygun donanım parçalarını içeren bir harita tasarlanmıştır. Kontrol ve deney grubu arasında, istatistiksel bakımdan anlamlı bir fark bulunamamıştır (Güneş, 2010).

2009 yılında Sert tarafından, Lise 2. sınıf seçmeli bilgi ve iletişim teknolojileri dersi internet konusunun öğretiminde, 3B bilgisayar oyununun kullanıldığı oyun-tabanlı öğrenme ortamı ile anlatıma dayalı öğrenme ortamı arasındaki farkın incelendiği bir çalışma yapılmıştır (Sert, 2009). Quest Atlantis’de geliştirilen oyunda; deney ve kontrol grubu öğrencilerinin başarıları istatistiksel olarak anlamlı bir fark göstermemiştir. Ayrıca, öğrenci başarılarının cinsiyetten etkilenmediği görülmüştür.

Aynı şekilde Quest Atlantis ortamında, ilköğretim 7. sınıf bilgisayar dersi donanım ünitesi ile ilgili yapılan araştırmada, deney grubu ile kontrol grubu arasında anlamlı bir fark çıkmamış; fakat deney grubu öğrencilerinin başarı ortalamalarının

daha fazla arttığı gözlenmiştir. Bu çalışmada da cinsiyetin başarıyı etkilemediği sonucuna ulaşılmıştır (Yağız, 2007).

Çalışmaların tamamında olumlu yönde gelişme elde edilmesine rağmen anlamlı bir fark bulunamamıştır. Araştırmacılar tarafından bu durum, konuların kısa ve basit olmasına, uygulama süresinin azlığına bağlanmıştır.

### **6.3 Öneriler**

Bu bölümde, oyunu geliştirme süreci ve geliştirilen oyunu uygulama sürecine göre öneriler iki başlık altında sunulmuştur.

#### **6.3.1 Geliştirmeyle İlgili Öneriler**

- Oyun geliştirme süreci, ileri düzeyde programcılık bilgisi, grafik tasarımı, veri tabanı yönetimi bilgisi, farklı işletim sistemlerinde deneyim, alan bilgisi, öğretim materyali tasarımı gibi farklı uzmanlık alanlarının birleşimini kapsadığı için konusunda uzman kişilerden oluşan bir ekip ile çalışılmalıdır.
- Yeni baştan MMORPG yapısı oluşturmak yerine, mevcut açık kaynak kodlu MMORPG'lerden yararlanıp üzerlerinde eğitsel içeriğe uygun değişiklikler yapılabilir.
- Öğrencilerin motivasyonunu ve oyunda geçirdikleri süreyi arttırmak için görsel öğelerin mevcut eğilimlere cevap verecek şekilde seçilmesi gerekmektedir.
- Üç boyutlu oyunlar iki boyutlu oyunlara göre daha fazla tercih edildiği için, geliştirmede üç boyutlu bir oyun motoru kullanılabilir.
- Oyuncuların günün her saati oyuna katılabilmeleri için sunucuların her zaman erişilebilir olmaları sağlanmalıdır.

### 6.3.2 Uygulamayla İlgili Öneriler

- Oyunla ilgili bir oryantasyon alanı oluşturulup oyuncuların bu alanda oyunu tanımaları sağlanmalıdır.
- Oyun oynamaya alışkın olmayan öğrencilerin oryantasyon alanındaki tüm eğitimleri başarıyla tamamlayıp tamamlamadıkları kontrol edilmelidir.
- Oryantasyon döneminde, hem sanal ortamda hem de uygulama ortamında rehberlik edecek birden fazla kişinin bulunması sağlanmalıdır.
- Uygulama yapılacak ortamın ağ alt yapısının yeterli olup olmadığı kontrol edilmelidir.
- Günün belli saatlerinde ya da haftanın belli günlerinde ortam hakkında tartışmaların yapılabileceği ortak bir zaman dilimi belirlenip oyunculara rehberlik edilebilir.
- Sadece ortamın eğlence kısmıyla ilgilenen oyuncular tespit edilip eğitsel hedefleri de yerine getirmeleri hususunda motive edilebilir.
- elektroGame bir ders ya da bir üniteden bağımsız olarak tasarlanmıştır. Uygun eğitsel içerikler belirlenerek farklı ders ya da konuların öğretiminde kullanılabilir.



## 7. KAYNAKLAR

- MMO Budget Busters*. (2012, 6 8). mmohut: <http://mmohuts.com/editorials/mmo-budget-busters> adresinden alınmıştır
- U Game. U Gain*. (2012, 6 8). Dimension U: [https://dimensionu.com/home/landing\\_flash.aspx](https://dimensionu.com/home/landing_flash.aspx) adresinden alınmıştır
- Akgün, E., Nuhoğlu, P., Tüzün, H., Kaya, G., & Çınar, M. (2011). BİR Eğitsel Oyun Tasarımı Modelinin Geliştirilmesi. *Eğitim Teknolojisi Kuram ve Uygulama*, 41-61.
- Akıllı, K., & Çağiltay, K. (2006). An instructional design/development model for the creation of game-like learning environments: Fidge model. *Affective and Emotional Aspects of Human-Computer Interaction: Game-based and Innovative Learning Approaches*, 93-112.
- Akpınar, Y. (1999). *Bilgisayar Destekli Öğretim ve Uygulamalar*. Ankara: Anı Yayıncılık.
- Aksaçoğlu, A., & Yılmaz, B. (2007). Öğrencilerin Televizyon İzlemeleri ve Bilgisayar Kullanmalarının Okuma Alışkanlıkları Üzerine Etkisi. *Türk Kütüphaneciliği*, 21(1).
- Alessi, S. M., & Trollip, S. R. (2001). *Multimedia for learning: Methods and Development (third ed.)*. Massachusetts: Allyn & Bacon: Needham Heights.
- Alexander, T. (2005). *Massively Multiplayer Game Development 2*. Massachusetts: Charles River Media Game Development.
- Alkan, A. (2006). İlköğretim Öğrencilerinin Fen Bilgisine Karşı Tutumları. Afyon: Afyon Kocatepe Üniversitesi.
- Altın, K. (2009). *Bilgisayar Destekli Fen ve Teknoloji Öğretimi*. İstanbul: Beta Basın Yayın.
- Amory, A. (2007). Game object model version II: a theoretical framework for educational game development. *Education Tech Research Dev*, 55, 51-77.
- Amory, A., & Seagram, R. (2003). Educational Game Models: Conceptualization and Evaluation. *South African*, 206-217.
- Ang, C. S., & Rao, R. K. (2008). Computer Game Theories for Designing Motivating Educational Software: A Survey Study. *International Journal on E-Learning*, 7(2).
- Ateş, S., & Polat, M. (2005). Elektrik devreleri konusundaki kavram yanlışlarının giderilmesinde öğrenme evreleri metodunun etkisi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi* (28), 39-47.
- Bakar, A., Tüzün, H., & Çağiltay, K. (2008). Öğrencilerin Eğitsel Bilgisayar Oyunu Kullanımına İlişkin Görüşleri: Sosyal Bilgiler Dersi Örneği. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 27-37.

- Barab, S., Dodge, T., Thomas, M. K., Jackson, C., & Tuzun, H. (2007). Our Designs and the Social Agendas They Carry. *The Journal Of The Learning Sciences* , 16(2), 263–305.
- Barab, S., Thomas, M., Dodge, T., Carteaux, B., & Tuzun, H. (2005). Making learning fun: Quest Atlantis, a game without guns. *Educational Technology Research and Development* (53), 86-107.
- Bartholomew, L. K., M.Sockrider, M., Abramson, S. L., Swank, P. R., Czyzewski, D. I., Tortolero, S. R., . . . Tyrrell, S. (2006). Partners in School Asthma Management: Evaluation of a Self-Management Program for Children With Asthma. *Journal of School Health*, 76(3), 283-290.
- Baykul, Y. (1990). *İlkokul 5. Sınıftan Lise ve Dengi Okulların Son Sınıflarına Kadar Matematik ve Fen Derslerine Karşı Tutumda Görülen Değişmeler ve Öğrenci Seçme Sınavındaki Başarı İle İlişkili Olduğu Düşünülen Bazı Faktörler*. Ankara: ÖSYM Yayınları.
- Bayram, S. (1999). *Bilgisayar Destekli Öğretim Teknolojileri*. İstanbul: Marmara Üniversitesi Teknik Eğitim Fakültesi Yayınları.
- Bednar, A. K., Cunningham, D., Duffy, T. M., & Perry, J. P. (1995). *Theory into practice: How do we link? In G.J. Anglin (Ed.), Instructional technology: Past, present and future*. Englewood: Libraries Unlimited, Inc.
- Bilgi, A. (2005). *Bilgisayar Oyunu Oynayan Ve Oynamayan İlköğretim Öğrencilerinin Saldırganlık, Depresyon Ve Yalnızlık Düzeylerinin İncelenmesi*. İstanbul: Marmara Üniversitesi.
- BitDünyası. (2009, 07 13). *Bilgisayarda Oyun Oynama Oranı Hızla Artıyor*. 12 06, 2011 tarihinde Bilgi ve İletişim : <http://www.bitdunyasi.com/tr/?Sayfa=Detay&Id=568> adresinden alındı
- Bottino, R., Ferlino, L., & Travella, M. O. (2006). Developing strategic and reasoning abilities with computer games at primary school level. *Computers & Education*, 49(4), 1272–1286.
- Boulanger, J.-S. (2006). *Interest Management for Massively Multiplayer Games*. Montreal, Quebec: McGill University.
- Brom, C., Preuss, M., & Klement, D. (2011). Are Educational computer micro-games engaging and effective for knowledge acquisition at high-schools? A quasi-experimantal study. *Computers & Education*(57), 1971-1988.
- Büyüköztürk, Ş. (2003). *Sosyal Bilimler İçin Veri Analizi El Kitabı* (3. b.). Ankara: Pegem-A Yayınları.
- Chambers, S., & Andre, T. (1997). Gender, prior knowledge, interest, and experience in electricity and conceptual change text manipulations in learning about direct current. *Journal of Research in Science Teaching* (34), 107-123.

- Cheng, A. K., & Kwen, B. H. (1998). *Primary pupils' conceptions about some aspect of electricit*. Australian Association for Research in Education: <http://www.aare.edu.au/98pap/ang98205.htm> adresinden alınmıştır
- Cohen, R., Eylon, B., & Ganiel, V. (1983). Potential differences and current in simple electric circuits: A study of students' concepts. *American Journal of Physics*, 51(5), 407-412.
- Crawford, C. (1984). *Art of computer game design*. Osborne: McGraw-Hill.
- Crowley, J. (2002). Analogies constructed by students in a selective high school. . Curtin University of Technology.
- Csikszentmihalyi, M. (1991). *Flow: The Psychology of Optimal Experience*. Harper Perennial.
- Culatta, R. (2010, 1 1). *Learning Theories*. 01 05, 2012 tarihinde Instructional Design: <http://www.instructionaldesign.org/theories/index.html> adresinden alındı
- Çakır, N., Şenler, B., & Taşkın, B. (2007). İlköğretim İkinci Kademe Öğrencilerinin Fen Bilgisi Dersine Yönelik Tutumlarının Belirlenmesi. *Türk Eğitim Bilimleri Dergisi*, 5, 637-655.
- Çepni, S., & Keleş, E. (2006). Turkish Students' Conceptions about the Simple Electric Circuits. *International Journal of Science and Mathematics Education*, 4(2), 269-291.
- Çokadar, H., & Külçe, C. (2008). Pupil's Attitudes Towards Science: A Case Of Turkey. *World Applied Sciences Journal*, 3, 102-109.
- Dede, Y. (2003). Arcs Motivasyon Modeli'nin Öğrencilerin Matematiğe Yönelik Motivasyonlarına Etkisi. *Pamukkale Üniversitesi Eğitim Fakültesi Dergisi*, 2(14), 173-182.
- Demirci, N. (2003). *Bilgisayarla Etkili Öğretme Stratejileri ve Fizik Öğretimi*. Ankara: Nobel Yayın Dağıtım.
- Demirel, Ö. (2002). *Kuramdan Uygulamaya Eğitimde Programram Geliştirme*. Ankara: Pegem A Yayıncılık.
- Demirel, Ö., Seferoğlu, S., & Yağcı, E. (2003). *Öğretim teknolojileri ve materyal geliştirme*. Ankara: PegemA Yayıncılık.
- Deryakulu, D. (2001). *Yapıcı Öğrenme*. Ankara: Eğitimsen Yayınları.
- Deveci, S. E., Açık, Y., Gülbayrak, C., Demir, A. F., Karadağ, M., & Koçdemir, E. (2007, 4 12). İlköğretim Öğrencilerinin Cep Telefonu, Bilgisayar, Televizyon Gibi Elektromanyetik Alan Oluşturan Cihazları Kullanım Sıklığı. *Fırat Tıp Dergisi*, 279-283.
- Dickey, M. D. (2003). Teaching in 3D: Pedagogical affordances and constraints of. *Distance Education*, 24(1), 105-121.

- Doğusoy, B., & İnal, Y. (2006). Çok Kullanıcılı Bilgisayar Oyunları ile Öğrenme. 7. *Ulusal Fen Bilimleri ve Matematik Eğitimi Kongresi*.
- Doolittle, P. (1999). Constructivism and Online Education. *Online Conference on Teaching Online in Higher Education*. Virginia Polytechnic Institute & State .
- Duit, R. (1992). Vorsteinerung und Physiklernen. *Physik in der Schule* (30), 282-285.
- Duit, R., & Rhöneck, C. v. (2012, 6 1). *Learning and Understanding Key Concepts of Electricity*. The Ohio State University Department of Physics:  
<http://www.physics.ohio-state.edu/~jossem/ICPE/C2.html> adresinden alınmıştır
- Duit, R., & Treagust, D. (1998). Learning in Science. B. J. Fraser, & K. Tobin içinde, *From Behaviourism Towards Social Constructivism and Beyond*. In *International Handbook of Science Education*. Kluwer Academic Publishers.
- Duman, B. (2008). *Öğrenme- Öğretim Kuramları ve Süreç Temelli Öğretim* (Genişletilmiş 2. Baskı b.). Ankara: Anı Yayıncılık.
- Dupin, J. J., & Johsua, S. (1987). Conceptions of French pupils concerning electric circuits: Structure and evolution. *Journal of Research in Science Teaching*, 24(9), 791-806.
- eAthenawiki. (2010, 09 24). *eAthena wiki*. eAthena:  
[http://eathena.ws/wiki/index.php/Category:Script\\_Command](http://eathena.ws/wiki/index.php/Category:Script_Command) adresinden alınmıştır
- Ebner, M., & Holzinger, A. (2007). Successful implementation of user-centered game based learning in higher education: An example from civil engineering. *Computers & Education*, 3(49), 873-890.
- Egenfeldt-Nielsen, S. (2005). *Beyond Edutainment: Exploring the Educational Potential of Computer Games*. <http://www.itu.dk/>. adresinden alınmıştır
- Engelhardt, P. V., & Beichner, R. J. (2004). Students' understanding of direct current resistive electrical circuits. *American Journal of Physics*, 72, 98-115.
- Eow, Y., Ali, W., Mahmud, R., & Baki, R. (2009, Haziran). Form one students' engagement with computer games and its effect on their academic achievement in a Malaysian secondary school. *Computers & Education*, 1082-1091.
- Ertürk, S. (1993). *Eğitimde Program Geliştirme*. Ankara: Meteksan Yayınları.
- Fraenkel, J. R., & Wallen, N. (2003). *How to design and evaluate research in education*. New York: McGraw-Hill Companies.
- Frederiksen, J., White, B. Y., & Gutwill, J. (1999). Dynamic mental model in learning science: The importance of constructing derivational linkages among models. *Journal of Research in Science Teaching*, 36(7), 806-836.
- Funk, J. B. (2003). How children experience playing video games. *Proceedings of Acm International Conference*, (s. 1-14).
- Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. *Simulation & Gaming*, 33(4), 441-467.

- Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York: Palgrave/St. Martin's.
- Gfk. (2009, 6 15). *GfK Arařtırmalar*. 9 10, 2010 tarihinde GfK Türkiye: [http://www.gfk.com/gfkturkiye/sectors/it\\_and\\_telecom/index.tr.html](http://www.gfk.com/gfkturkiye/sectors/it_and_telecom/index.tr.html) adresinden alındı
- Glaserfeld, E. V. (1995). A constructivist approach to teaching. I. L. (Eds.) içinde, *Constructivism in Education*. Lawrence Erlbaum Associates.
- Graser, M. (2009, 08 21). *Videogames rock song sales*. www.variety.com: <http://www.variety.com/article/VR1118007573?refCatId=16> adresinden alınmıřtır
- Griffiths, M., Davies, M., & Chappell, D. (2003). Breaking the stereotype: The case of on-line gaming. *Cyberpsychology and Behavior*, 6(1), 81-91.
- Güneř, H. (2010). Geliřtirilen Çevrimiçi Elektrogame Oyununun İlköğretim 4. Basamak Biliřim Teknolojileri Dersi Bařarısına Etkisi. Balıkesir.
- Hall, R., & Novak, J. (2008). *Game Development Essentials: Online Game Development*. New York: Delmar Cengage Learning.
- Heller, P. M., & Finley, F. N. (1992). Variable uses of alternative conceptions: a case study in current electricity. *Journal of Research in Science Teaching*, 29(3), 259-272.
- Hou, H.-T. (2012). Exploring the behavioral patterns of learners in an educational massively multiple online role-playing game (MMORPG). *Computers & Education*, 58(4), 1225-1233.
- Hwang, G.-J., Wu, P.-H., & Chen, C.-C. (2012). An Online Game Approach for Improving Students' Learning Performance in Web-Based Problem-Solving Activities. *Computers & Education*(59), 1246-1256.
- İpek, İ. (2001). *Bilgisayarla Öğretim, Tasarım, Geliřtirme ve Yöntemler*. Ankara: Tıp Teknik Yayınevi.
- İřman, A. (1999). "Eğitim Teknolojisinin Kuramsal Boyutu: Yapısalcı Yaklařımın (Constructivism) Eğitim Öğretim Ortamlarına Etkisi. *Öğretmen Eğitiminde Çağdař Yaklařımlar Sempozyumu*. İzmir: Dokuz Eylül Üniversitesi, Buca Eğitim Fakültesi.
- Jonassen, D. (1991b). Objectivism versus constructivism: Do we need a new philosophical paradigm? *Educational Technology Research and Development*, 5-14.
- Jonassen, D. H. (1991a). Objectivism vs. constructivism: Do we need a new philosophical paradigm? *Educational Technology, Research and Development*, 5-13.
- Kärrqvist, C. (1985). The development of concepts by means of dialogues centered on experiments. R. Duit, W. Jung, & C. Rhöneck içinde, *Aspects of Understanding Electricity* (s. 215-226). Kiel, Germany: IPN.
- Kaya, Z. (2005). *Öğretim Teknolojileri ve Materyal Geliřtirme*. Ankara: Pegem Yayıncılık.

- Keller, J. M. (1984). The use of the ARCS model of motivation in teacher trainin. *Aspects of educational technology*, 140-145.
- Keller, J. M. (1984). *The use of the ARCS model of motivation in teacher training*. London: Kogan Page.
- Keller, J. M., & Suzuki, K. (1988). Application of the ARCS model to courseware design. *Instructional Designs for Microcomputer Courseware*, 401-434.
- Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. *The Internet and Higher Education*, 13-24.
- Kiili, K. (2005b). *Educational Game Design: Experiential gaming model revised*. Pori: Tampere University of Technology.
- Kılıç, G. (2001, Haziran). Oluşturmacı Fen Öğretimi. *Kuram ve Uygulamada Eğitim Bilimleri Dergisi* (1), 23-38.
- Klopfer, E., Squire, K., & Jenkins, H. (2002). Environmental Detectives: PDAs as a Window into a Virtual Simulated World. *Wireless and Mobile Technologies in Education (WMTE'02)* (s. 95-98). IEEE Computer Society.
- Konuksal, S. (2009, 12 15). *Bölüm Sonu Canavarı*. 12 06, 2011 tarihinde Sayılarla World of Warcraft: [http://www.bolumsonucanavari.com/Haberler-Sayilarla\\_World\\_of\\_Warcraft-3030.htm](http://www.bolumsonucanavari.com/Haberler-Sayilarla_World_of_Warcraft-3030.htm) adresinden alındı
- Konuksal, S. (2009, 12 15). *Sayılarla World of Warcraft*. Bölüm Sonu Canavarı: [http://www.bolumsonucanavari.com/Haberler-Sayilarla\\_World\\_of\\_Warcraft-3030.htm](http://www.bolumsonucanavari.com/Haberler-Sayilarla_World_of_Warcraft-3030.htm) adresinden alınmıştır
- Korkmaz, A. İ. (2006). *Anlamli Öğrenme Yaklaşımına Dayali Bİlgİsayar Destekli 7. Sınıf Fen Bİlgİsİ Dersİ İçİN Hazirlanan BİR Ders Yaziliminin ÖğrencİlerİN AkademİK Başarılarına Ve Kaliciliğİna Etkİsİ*. Adana: Çukurova Üniversitesi.
- Kramer, W. (2000a, July). *What Makes a Game Good?* The Games Journal: <http://www.thegamesjournal.com/articles/WhatMakesaGame.shtml> adresinden alınmıştır
- Kramer, W. (2000b, December). *What is a Game?* The Games Journal: <http://www.thegamesjournal.com/articles/WhatIsaGame.shtml> adresinden alınmıştır
- Kula, A. (2005). *Öğretimsel Bilgisayar Oyunlarının Temel Aritmetik İşlem Becerilerinin Gelişimine Etkisi*. Ankara: Hacettepe Üniversitesi.
- Kurnaz, M. A., & Yiğit, N. (2010). Physics Attitude Scale : Development , Validity and Reliability. *Necatibey Eğitim Fakültesi Elektronik Fen ve Matematik Eğitimi Dergisi (EFMED)*, 4(1), 29-49.
- Küçüközer, H. (2004). Yapılandırmacı Öğrenme Kuramına Dayalı Olarak Geliştirilen Öğretim Modelinin Lise 1. Sınıf Öğrencilerinin Basit Elektrik Devrelerine İlişkin Kavramsal Anlamalarına Etkisi. Balıkesir: Balıkesir Üniversitesi.

- Lakoff, G. (1987). *Women, fire, and dangerous things*. Chicago: University of Chicago Press.
- Lee, Y., & Law, N. (2001). Explorations in promoting conceptual change in electrical concepts via ontological category shift. *International Journal Science Education*, 23(2), 111-149.
- Lim, C. P., Nonis, D., & Hedberg, J. (2006). Gaming in a 3D multiuser virtual. *British Journal of*, 37(2), 211-231.
- Linderöth, J., Lantz-Andersson, A., & Lindström, B. (2002). Electronic exaggerations and virtual worries: Mapping research of computer games relevant to the understanding of children's game play. *Contemporary Issues in Early Childhood*, 3(2), 226-250.
- Loyd, B., & Gressard, C. (1984). Reliability and Factorial Validity of Computer Attitude Scales. *Educational and Psychological Measurement*, 44(2), 501-505.
- Malkoç, A. (2010, 12 23). *Rekorları Alt Üst Eden oyun!* 12 05, 2011 tarihinde Chip Dergisi: <[http://www.chip.com.tr/konu/rekorlari-alt-ust-eden-oyun\\_24233.html](http://www.chip.com.tr/konu/rekorlari-alt-ust-eden-oyun_24233.html)> adresinden alındı
- Malone, T. (1981). What makes computer games fun? *Byte*, 6(12), 258-277.
- Mann, B. D., Eidelson, B. M., Fukuchi, S. G., Nissman, S. A., Robertson, S., & Jardines, L. (2002). The development of an interactive game-based tool for learning surgical. *The American Journal of Surgery*, 3(183), 305-308.
- Manninen, T., & Kujanpää, T. (2007). The Value of Virtual Assets – The Role of Game Characters in MMOGs. *Int. Journal of Business Science and Applied Management*, 2(1), 21-33.
- Maroney, K. (2001, Mayıs). *My Entire Waking Life*. The Games Journal: <http://www.thegamesjournal.com/articles/MyEntireWakingLife.shtml> adresinden alınmıştır
- Mayer, R. E., Schustack, M. W., & Blanton, W. E. (1999). What Do Children Learn from Using Computers in an Informal, Collaborative Setting? *Educational Technology*, 39, 27-31.
- McFarlane, R. D. (2005). *Network software architecture for real-time massively-multiplayer online games*. Montreal, Quebec, Canada: McGill University.
- Mulhall, P., McKittrick, B., & Gustone, R. (2001). A perspective on the resolution of confusions in the teaching of electricity. *Research in Science Education*, 31, 575-587.
- Mulholland, A., & Hakala, T. (2004). *Programming Multiplayer Games*. Texas: Wordware Publishing.
- Osborne, R. (1981). Children's Ideas About Electric Circuits. *New Zealand Science Teacher* (29), 12-19.

- Osborne, R. (1983). Towards modifying children's ideas about electric current. *Research in Science and Technological Education*, 1, 79-82.
- Özden, Y. (2003). *Öğrenme ve Öğretme*. Ankara: Pegem A.
- Pelletier, C. (2005). Studying games in school: A framework for media education. *Research Association Conference*.
- Peşman, H. (2005, September). Development of A Three-Tier Test To Assess Ninth Grade Students' Miconceptions About Simple Electric Circuits. Ankara: Middle East Technical University.
- Philips, D., & Soltis, J. (2005). *Perspicities on Learning*. Ankara: Nobel Yayın Dağıtım.
- Pillay, H. (2002). An investigation of cognitive processes engaged in by recreational computer game players: An implication for skills of the future. *Journal of Research on Technology in Education*(34), 336–350.
- Pivec, M., Dziabenko, O., & Schinnerl, I. (2003). Aspects of Game- Based Learning. *I-KNOW 03, the Third International Conference on Knowledge Management*. Graz, Austria.
- Pivec, M., Koubek, A., & Dondi, C. (2004). *Guidelines for Game-Based Learning*. Eichengrund: Pabst Science Publishers.
- Prensky, M. (2001). *Fun, play and games: What makes games engaging. From digital game-based learning*. Mark Prensky: <http://www.marcprensky.com/writing/Prensky%20-%20Digital%20Game-Based%20Learning-Ch5.pdf> adresinden alınmıştır
- Saban, A. (2002). *Öğrenme Öğretme Süreci*. Ankara: Nobel Yayın Dağıtım.
- Sauter, P., Vögler, G., Specht, G., & Flor, T. (2005). A Model–View–Controller extension for pervasive multi-client user interfaces. *Journal*, 9(2), 100-107.
- SCCC. (1996). *Teaching for Effective Learning Dundee*. Scottish Consultative Council on the Curriculum.
- Senemoğlu, N. (2005). *Gelişim, Öğrenme ve Öğretim: Kuramdan Uygulamaya*. Ankara: Gazi Kitabevi.
- Sert, S. (2009). *Eğitsel Bilgisayar Oyunlarının Lise Öğrencilerinin İnternete İlişkin Bilgi Düzeyi Performansına Etkisi: Quest Atlantis Örneği*. Ankara: Hacettepe Üniversitesi.
- Shaw, A. (2010, 6 10). *Aspen Achievement Academy*. 7 20, 2011 tarihinde The Most Addictive Types of Video Games: [http://www.aspenacademy.com/addictive\\_games.html](http://www.aspenacademy.com/addictive_games.html) adresinden alındı
- Shipstone, D. M. (1985). Electricity in simple dc circuits. R. Driver, E. Guesne, & A. Tiberghin içinde, *Children's Ideas in Science* (s. 33-51). Milton Keynes, England: Open University Press.



- Shipstone, D., Rhöneck, C., Jung, W., C. Kärrqvist, J. D., Joshua, S., & Licht, P. (1988). A study of students' understanding of electricity in five European countries. *International Journal of Science Education*, 3(10), 303-316.
- Siang, A. C., & Rao, R. K. (2003). Theories of Learning: A Computer Game Perspective. *Proceedings of the IEEE Fifth International Symposium on Multimedia Software Engineering (ISMSE'03)*, (s. 239-244).
- Singhal, S., & Zyda, M. (1999). *Networked Virtual Environments: Design and Implementation*. New York: Addison-Wesley and ACM Press.
- Sladjan Bogojevic, M. K. (2003). *The Architecture of Massive Multiplayer Online Games*. Lund: Lund University.
- Smed, J., Kaukoranta, T., & Hakonen, H. (2001). Aspects of networking in multiplayer computer games. *In Proceedings of International Conference on Application and Development of Computer Games in the 21st Century*, (s. 74-81).
- Song, M., & Zhang, S. (2008). EFM: A Model for Educational Game Design. *Technologies for ELearning and Digital Entertainment* (5093), 509-517.
- Squire, K. (2005). Changing the Game: What Happens When Video Games Enter the Classroom? *Innovate Journal of Online Education*, 1(6).
- Steinkuehler, C. A. (2004). Learning in massively multiplayer online games. *Proceedings of the 6th International Conference of the Learning Sciences*, (s. 521-528). Mahwah, NJ, USA: Erlbaum.
- Suh, S., Kim, S., & Kim, N. (2010). Effectiveness of MMORPG-based instruction in elementary English education in Korea. *Journal of Computer Assisted Learning* (26), 370-378.
- Süngü, E. (2010, 12 12). *Call of Duty'e Harcanan Zaman İnanılmaz!* 12 06, 2011 tarihinde Chip Online Tr: [http://www.chip.com.tr/konu/call-of-duty-e-harcanan-zaman-inanilmaz\\_24274.html](http://www.chip.com.tr/konu/call-of-duty-e-harcanan-zaman-inanilmaz_24274.html) adresinden alındı
- Swanson, K., & Piech, C. (2009). *A Math Oriented Massively Multiplayer Online Role Playing Game*. Mithril: <http://stanford.edu/~pnaqlada/mithril/> adresinden alınmıştır
- Şen, A., & Çıldır, I. (2006). Lise Öğrencilerinin Elektrik Akımı Konusundaki Kavram Yanılgılarının Kavram Haritalarıyla Belirlenmesi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi* (30), 92-101.
- Şerefhanoglu, H. (2007). İlköğretim İkinci Kademe Öğrencilerinin Bilgisayara Yönelik Tutumları İle Çoklu Zeka Alanlarının Karşılaştırılması. Balıkesir: Balıkesir Üniversitesi.
- Tekmen, S. (2006). Fizik Dersinde, Bilgisayar Destekli Eğitimin Öğrencilerin Erişisine, Derse Karşı Tutumlarına ve Kalıcılığa Etkisi. Bolu: Abant İzzet Baysal Üniversitesi.
- Temiz, N. (2010). *Eğitime Bakış*. 11 10, 2011 tarihinde Ders Destek: <http://www.dersdestek.com/deneme/egitimebakis/index.html> adresinden alındı

- Thorsen, T. (2012, 6 8). *Halo MMORPG had \$90 million budget pre-cancellation*. Gamespot: <http://www.gamespot.com/news/halo-mmorpg-had-90-million-budget-pre-cancellation-6257799> adresinden alınmıştır
- Turvey, K. (2006). Towards deeper learning through creativity within online communities in primary education. *Computers & Education*(46), 309–321.
- TÜİK. (2011, 06 30). *Hanelerde Bilişim Teknolojileri Kullanımı*. 12 06, 2011 tarihinde Türkiye İstatistik Kurumu: [http://www.tuik.gov.tr/VeriBilgi.do?tb\\_id=60&ust\\_id=2](http://www.tuik.gov.tr/VeriBilgi.do?tb_id=60&ust_id=2) adresinden alındı
- Tüzün, H. (2004). *Motivating learners in educational computer games*. . Bloomington: Indiana University.
- Tüzün, H. (2006). *Affective and emotional aspects of human-computer interaction: Game-based and innovative learning approaches*. Amsterdam, Netherlands: IOS Press.
- Tüzün, H. (2007). Blending video games with learning: Issues and challenges with. *British Journal of Educational Technology*, 38(3), 465–477.
- Tüzün, H., & Bayırtepe, E. (2007). Oyun-Tabanlı Öğrenme Ortamlarının Öğrencilerin Bilgisayar Dersindeki Başarıları Ve Öz-Yeterlik Algıları Üzerine Etkileri. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi* (33), 41-54.
- Uşun, S. (2000). *Dünyada ve Türkiye’de Bilgisayar Destekli Öğretim*. Ankara: Pegem A Yayınevi.
- Uzunkavak, M. (2004). Lise Ve Dengi Okul Öğrencilerinin Elektrik Ve Manyetizma Öğreniminde Karşılaştığı Kavram Yanılgıları. Isparta: Süleyman Demirel Üniversitesi.
- Üçgül, M. (2006). *The Impact Of Computer Games On Students’ Motivation*. Ankara: Middle East Technical University.
- Ülgen, G. (2011). *Kavram Geliştirme kuramlar ve uygulamalar* (3 b.). Ankara: Pegem A yayınevi.
- Ünal, M. N. (2009). *Eğitsel Bilgisayar Oyunlarının Eğlendirici Ve Motive Edici Özelliklerinin Akademik Başarıya Ve Motivasyona Etkisi*. Eskişehir: Anadolu Üniversitesi.
- Widodo, A., Duit, R., & Müller, C. (2002). Constructivist views of teaching and learning in practice: Teachers’ views and classroom behaviour. Paper. *Annual Meeting of the National Association for Research in Science Teaching*. New Orleans.
- Wikipedia. (2011). *Bilgisayar Oyunu*. Wikipedia: [http://tr.wikipedia.org/wiki/Bilgisayar\\_oyunu](http://tr.wikipedia.org/wiki/Bilgisayar_oyunu) adresinden alınmıştır
- Wikipedia. (2011). *History of video games*. Wikipedia: [http://en.wikipedia.org/wiki/History\\_of\\_video\\_games](http://en.wikipedia.org/wiki/History_of_video_games) adresinden alınmıştır
- Wikipedia. (2011, 11 13). <http://tr.wikipedia.org/wiki/NPC>. 12 10, 2011 tarihinde wikipedi özgür ansiklopedi: <http://tr.wikipedia.org/wiki/NPC> adresinden alındı

- Wikipedia. (2011, 4). *Software framework*. www.wikipedia.org:  
[http://en.wikipedia.org/wiki/Software\\_framework](http://en.wikipedia.org/wiki/Software_framework) adresinden alınmıştır
- Wikipedia. (2011). *Tennis video games*. Wikipedia:  
[http://en.wikipedia.org/wiki/Category:Tennis\\_video\\_games](http://en.wikipedia.org/wiki/Category:Tennis_video_games) adresinden alınmıştır
- Wikipedia. (2011). *Video oyun konsolu*. Wikipedia:  
[http://tr.wikipedia.org/wiki/Video\\_oyun\\_konsolu](http://tr.wikipedia.org/wiki/Video_oyun_konsolu) adresinden alınmıştır
- Wikipedia. (2011, 10 23). *Wiki*. wikipedia.org web sitesi: <http://tr.wikipedia.org/wiki/Wiki> adresinden alınmıştır
- Yağız, E. (2007). *İlköğretim Öğrencilerinin Bilgisayar Dersindeki Başarıları ve Öz-Yeterlik Alguları Üzerine Etkileri*. Ankara: Hacettepe Üniversitesi.
- Young, M. (2004). An Ecological Description of Video Games in Education. *Proceedings of the International Conference on Education and Information Systems Technologies and Applications (EISTA)*. Orlando, FL.
- Zhao, Y. (2004). *ZON*. Zon: www.enterzon.com adresinden alınmıştır
- Zin, N. A., Jaafar, A., & Yue, W. S. (2009). Digital Game-based learning (DGBL) model and development methodology for teaching history. *WSEAS TRANSACTIONS on COMPUTERS*, 8(2), 322-333.

# **EKLER**

## 8. EKLER

### EK A – Fizik Tutum Ölçeđi

# FİZİK DERSİNE YÖNELİK TUTUM ÖLÇEĐİ

Okulu	
Sınıfı	
Numarası	
Adı Soyadı	
Cinsiyeti	<input type="radio"/> Erkek <input type="radio"/> Kız

### Yönergeler

1. Ölçeđi doldurmaya başlamadan önce yukarıda verilen kısma kendinize ait bilgileri giriniz. Bu bilgiler sadece bu çalışmada ve **sadece istatistiksel veri amacıyla kullanılacaktır.**
2. Sonraki sayfada Fizik dersine ilişkin tutumları ölçmek üzere hazırlanmış 20 maddeden oluşan bir tutum ölçeđi yer almaktadır. Ölçekteki maddelerin karşısında görüşünüzü belirteceđiniz beş seçenek vardır. Her bir maddeyi dikkatle okuduktan sonra bu seçeneklerden **sizce en uygun olanını işaretleyiniz.**

**Katılımınız için teşekkür ederim.**

	FİZİK DERSİYLE İLGİLİ CÜMLELER	Tamamen Katılıyorum	Katılıyorum	Kararsızım	Katılmıyorum	Hiç Katılmıyorum
1.	Fizik çok sevdiğim dersler arasındadır.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.	Fizik derslerindeki konuların azaltılmasından mutlu olurum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	Fizik dersi ile uğraşmak beni eğlendirir.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.	Fizik dersine çalışırken canım sıkılır.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.	Fizik dersinin beni düşündürtmesinden büyük zevk alırım.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.	Fizik dersinden korkarım.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	Fizik derslerin en güzelidir.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8.	Fizik dersinden hiç hoşlanmam.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.	Fizik ile ilgili her şey ilgimi çeker.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.	Yetki verseler okuldaki bütün Fizik derslerini kaldırırım.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.	Dersler arasında en çok Fizik dersinden hoşlanırım.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12.	Mümkün olsa Fizik yerine başka bir ders alırım.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13.	Fizik ödevlerini sıkılmadan, zevkle yaparım.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.	Fizik dersinden çekinirim.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15.	Fizikle ilgili bir problemi çözmek bana zevk verir.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16.	Fizik ders konuları ilgi duyduğum konular değildir.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
17.	Boş zamanlarımda fen konularıyla uğraşmaktan hoşlanırım.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18.	Fizik ile ilgili kitap okumanın pek yararlı bir iş olduğuna inanmıyorum.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19.	Fizik dersinde yapılan sınıf çalışmalarını (etkinliklerini) severim.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20.	Fizik dersinde düşünmek çok sıkıcıdır.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

# BİLGİSAYARA YÖNELİK TUTUM ÖLÇEĐİ

Okulu	
Sınıfı	
Numarası	
Adı Soyadı	
Cinsiyeti	<input type="radio"/> Erkek <input type="radio"/> Kız

## Yönergeler

- Ölçeđi doldurmaya başlamadan önce yukarıda verilen kısma kendinize ait bilgileri giriniz. Bu bilgiler sadece bu çalışmada ve **sadece istatistiksel veri amacıyla kullanılacaktır.**
- Sonraki sayfada bilgisayara ilişkin tutumları ölçmek üzere hazırlanmış 21 maddeden oluşan bir tutum ölçeđi yer almaktadır. Ölçekteki maddelerin karşısında görüşünüzü belirteceğiniz beş seçenek vardır. Her bir maddeyi dikkatle okuduktan sonra bu seçeneklerden **sizce en uygun olanını işaretleyiniz.**

**Katılımınız için teşekkür ederim.**

	BİLGİSAYARLA İLGİLİ CÜMLELER	Tamamen Katılıyorum	Katılıyorum	Kararsızım	Katılmıyorum	Hiç Katılmıyorum
1.	Bilgisayar dersinden iyi notlar alabilirim.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.	Bilgisayar kullanmak sanırım benim için çok zor olur.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	Bilgisayar kullanmada iyi olabilecek tipte biri değilim.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.	Bilgisayar dersinde başarılı olabileceğimi sanmıyorum.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.	Bilgisayarlarla çalışmak gerektiğinde kendime yeterince güvenirim.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.	Bir bilgisayar programında hemen çözemediğim bir sorun olduğunda, cevabı bulana kadar vazgeçmem.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	Bilgisayar kullanmam gerektiğinde kendimi rahat hissedirim.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8.	Bilgisayarlarla çalışmaya bir kez başlayınca bırakmak benim için çok zor olur.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.	Hayatımda hiçbir zaman bilgisayarları istekli olarak kullanacağımı zannetmiyorum.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.	Günlük hayatımda bilgisayarları çok az kullanacağımı tahmin ediyorum.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.	Bilgisayarlar beni huzursuz eder.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12.	Bilgisayarlar kendimi rahatsız hissetmeme neden oluyor.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13.	İçimden bilgisayarları parçalamak geliyor.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.	Bilgisayarla problemleri çözmek bana çekici gelmiyor.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15.	Bilgisayarlarla çalışmak sinirlerimi bozar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16.	Bilgisayar kursları almak için zahmete girmem.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
17.	Bilgisayar derslerinde başarılı olmak benim için önemlidir.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18.	Gelecekteki çalışmalarım için bilgisayarda ustalaşmam gerekecek.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19.	Bir bilgisayar dili öğrenmek istiyorum.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20.	Bilgisayarlar hakkında sürekli bilgi edinirim.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
21.	Bilgisayarlarla çalışmak zevklidir.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



# BASİT ELEKTRİK DEVRELERİ

## ÜÇ BASAMAKLI TEST





Okulu	
Sınıfı	
Numarası	
Adı Soyadı	
Cinsiyeti	<input type="radio"/> Erkek <input type="radio"/> Kız

### Yönergeler

1. Sınava başlamadan önce yukarıda verilen kısma kendinize ait bilgileri giriniz. Bu bilgiler sadece bu çalışmada ve sadece istatistiksel veri amacıyla kullanılacaktır.
2. Bütün sorulara cevap vermek için gayret gösteriniz.
3. Devrede kullanılan pil ve ampuller özdeştir.
4. Devrelerde kullanılan piller ideal pillerdir. Yani pillerin iç dirençleri önemsizdir.
5. İkinci basamak sorularda, verilen sebeplerden birini tercih etmezseniz, boş bırakılan şıkka kendi sebebinizi yazmalısınız.

**Katılımınız için teşekkür ederim.**

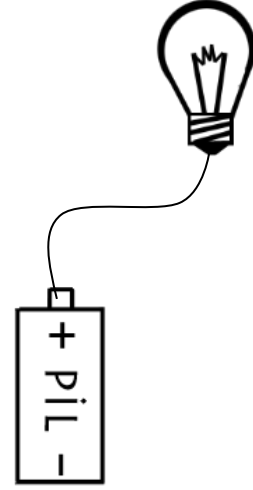
Sınavda Kullanılan Semboller

		
Pil	Lambalar	Direnç
		
		Ampul takılı duym

### SORULAR

1.1. Şekil 1’de gösterilen devredeki ampul ışık verir mi?

- (a) Evet, ışık verir.
- (b) Hayır, ışık vermez.



Şekil 1

1.2. Yukarıda verdiğim cevabın sebebi:

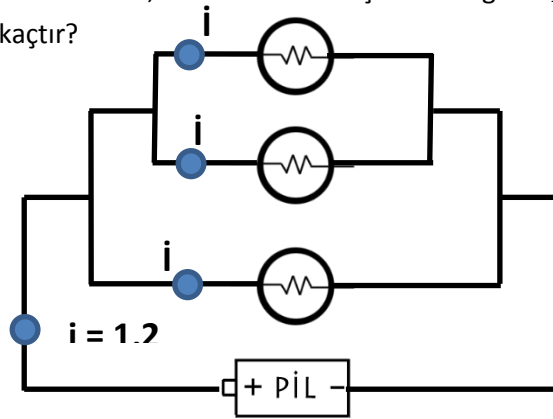
- (a) Pil ve ampul temas halindedir.
- (b) “+” ve “-” yüklerin ampulde birleşmesi için pilin “-” ucundan ampulün yan metal kısmına bir tel bağlanmalıdır.
- (c) Ampulden akım geçmesi için pilin “-” ucundan ampulün yan metal kısmına bir tel bağlanmalıdır.
- (d)

1.3. Yukarıdaki iki soruya verdiğim cevaptan;

- (a) Eminim.
- (b) Emin değilim.

2.1. Şekil 2’deki elektrik devresinde ana koldaki akım 1,2 A olarak verilmiştir. Buna göre  $i_1$ ,  $i_2$  ve  $i_3$  akımlarının büyüklüklerinin değerleri kaçtır?

- (a) 0,6 / 0,3 / 0,3
- (b) 0,4 / 0,4 / 0,4



Şekil 2

**2.2.** Yukarıdaki iki soruya verdiğim cevabın sebebi:

(a) Elektrik akımı ilk kol ayırımından eşit olarak ikiye ayrıldıktan sonra, ikinci kol ayırımında tekrar eşit olarak ikiye ayrılır.

(b) Üç özdeş ampul birbirine paralel bağlı olduğundan ana koldaki elektrik akımı üç ampul üzerinden de aynı büyüklükte geçecek şekilde kollara ayrılır.

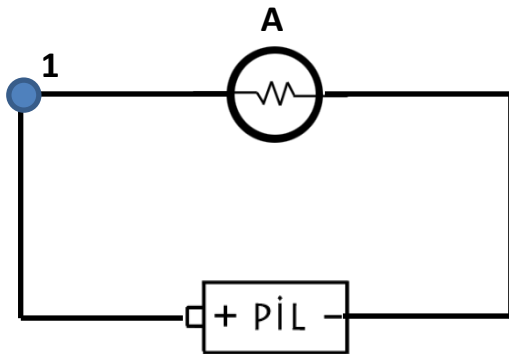
(c)

**2.3** Yukarıdaki iki soruya verdiğim cevaptan;

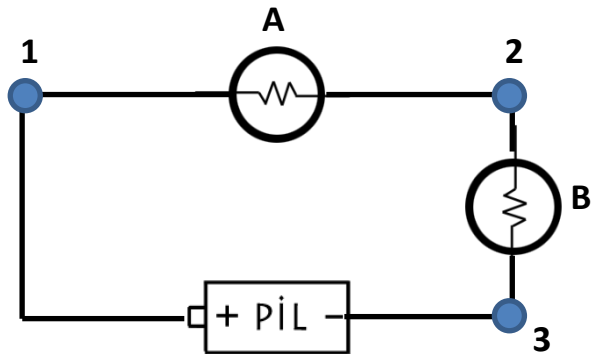
(a) Eminim.

(b) Emin değilim.

Şekil 3'te gösterilen bir elektrik devresine, Şekil 4'te gösterildiği gibi bir B ampulü ekleniyor. 3. ve 4. soruları bu bilgiye göre cevaplayınız.



**Şekil 3**



**Şekil 4**

**3.1.** Şekil 3 ve Şekil 4'teki 1 noktalarından geçen elektrik akımlarının büyüklüklerini karşılaştırınız.

(a) Şekil 3'te daha fazladır.

(b) Şekil 4'te daha fazladır.

(c) Her iki şekilde de eşittir.

**3.2.** Yukarıda verdiğim cevabın sebebi:

- (a) Pilden gelen aynı büyüklükteki elektrik akımları her iki şekildeki 1 noktasında da henüz kullanılmamıştır.
- (b) Her iki şekilde de pillerin sağladığı potansiyel farklar aynıdır; ama Şekil 4'teki eşdeğer direnç daha büyüktür.
- (c) Şekil 3'te bir ampulün kullandığı elektrik akımı, Şekil 4'te ise iki ampulün kullandığı elektrik akımı vardır.
- (d)

**3.3.** Yukarıdaki iki soruya verdiğim cevaptan;

- (a) Eminim.
- (b) Emin değilim.

**4.1.** Şekil 4'te 1, 2 ve 3 noktalarındaki akımların büyüklükleri ile A ve B ampullerinin parlaklıklarını karşılaştırınız.

	Akım	Parlaklık
<b>a)</b>	$i_1 = i_2 = i_3$	A ve B ampulleri aynı parlaklıktadır.
<b>b)</b>	$i_3 > i_2 > i_1$	B ampulü daha parlaktır.
<b>c)</b>	$i_1 > i_2 > i_3$	A ampulü daha parlaktır.
<b>d)</b>	$i_1 > i_2 > i_3$	A ve B ampulleri aynı parlaklıktadır.

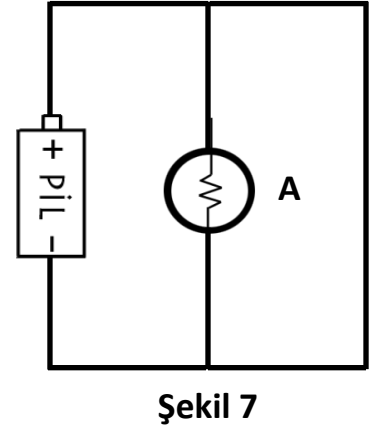
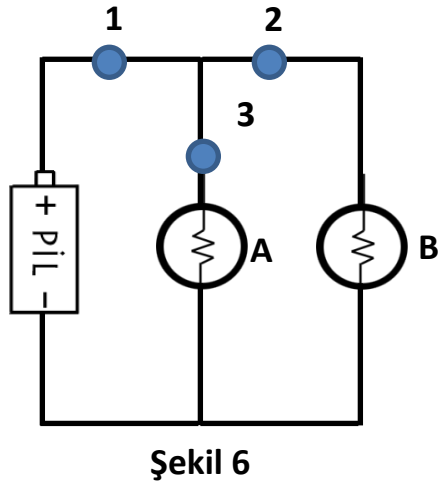
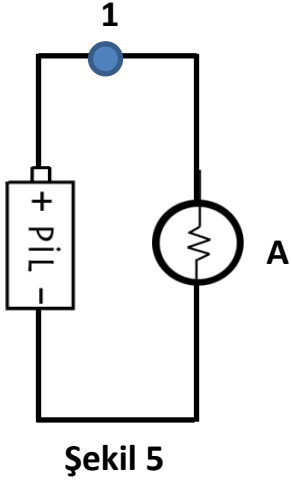
**4.2.** Yukarıda verdiğim cevabın sebebi:

- (a) Pile ne kadar çok yakın olunursa, elektrik akımı da o kadar çok olur.
- (b) Seri bağlı devrelerde akım şiddeti her yerde aynıdır.
- (c) Elektrik akımı ampuller tarafından kullanıldığı için azalır.
- (d)

**4.3.** Yukarıdaki iki soruya verdiğim cevaptan;

- (a) Eminim.  
(b) Emin değilim.

Şekil 5'te bir elektrik devresi verilmiştir. Öncelikle bir B ampulü Şekil 6'da görüldüğü gibi devreye ekleniyor. Daha sonra B ampulü devreden çıkarılarak ve onun yerine bir tel kullanılarak Şekil 7'deki devre oluşturuluyor. 5., 6., 7. ve 8. soruları bu bilgilere göre cevaplayınız.



5.1. Şekil 5 ve Şekil 6'daki elektrik akımlarının büyüklüklerini karşılaştırınız.

- (a) Şekil 5'te daha büyüktür.  
(b) Şekil 6'da daha büyüktür.  
(c) Şekil 5'te ve Şekil 6'da eşittir.

5.2. Yukarıda verdiğim cevabın sebebi:

- (a) Şekil 6'da iki ampul olduğundan, eş değer direnç fazladır.  
(b) Şekil 6'da, pilden gelen akım iki kola ayrılır.  
(c) Şekil 5'te, pil tek ampule; Şekil 6'da ise iki ampule akım verir.  
(d) Şekil 6'daki paralel devrede eşdeğer direnç daha küçüktür.

(e) Her iki şekilde de, 1 noktalarına gelen aynı büyüklükteki elektrik akımı henüz kollara ayrılmamıştır.

(f)

5.3. Yukarıdaki iki soruya verdiğim cevaptan;

- (a) Eminim.

(b) Emin değilim.

**6.1.** Şekil 6'da 1, 2 ve 3 kollarındaki akımların büyüklüklerini karşılaştırınız.

(a)  $i_1 > i_2 > i_3$

(b)  $i_1 > i_2 = i_3$

**6.2.** Yukarıda verdiğim cevabın sebebi:

(a) Akım kollarına ayrılırken gidiş yönüne düz kola daha çok, kıvrılan kola daha az akım geçer.

(b) Ampuller özdeş olduğundan, akım kol ayrımına geldiğinde eşit bir şekilde iki kola ayrılır.

(c)

**6.3.** Yukarıdaki iki soruya verdiğim cevaptan;

(a) Eminim.

(b) Emin değilim.

**7.1.** Şekil 6'da A ve B ampullerinin parlaklıklarını karşılaştırınız.

(a) A ve B ampullerinin parlaklıkları eşittir.

(b) A ampulü daha parlaktır.

(c) B ampulü daha parlaktır.

**7.2.** Yukarıda verdiğim cevabın sebebi:

(a) Kollara ayrılan akımın çoğu B ampulünden geçer.

(b) A ampulü pile daha yakındır.

(c) A ve B ampullerinden aynı büyüklükte akım geçer.

(d)

**7.3.** Yukarıdaki iki soruya verdiğim cevaptan;

(a) Eminim.

**(b)** Emin deęilim.

**8.1.** Őekil 6 ve Őekil 7'deki A lambalarının parlaklıklarını karşılaştırınız.

**(a)** Őekil 6'daki daha parlaktır.

**(b)** Őekil 7'deki daha parlaktır.

**(c)** İki Őekildeki de eőit parlaklıktadır.

**8.2.** Yukarıda verdięim cevabın sebebi:

**(a)** Őekil 7'deki A ampulünün üzerinden akım gemez.

**(b)** Őekil 6'daki akım iki ampul; Őekil 7'de ise tek ampul tarafından kullanılır.

**(c)** Her iki Őekilde de akım iki kola ayrılır.

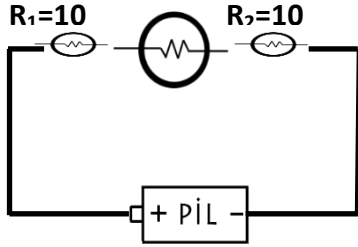
**(d)**

**8.3.** Yukarıdaki iki soruya verdięim cevaptan;

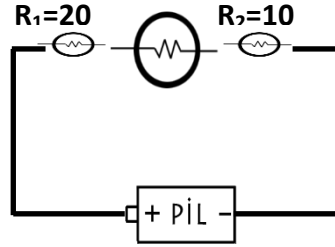
**(a)** Eminim.

**(b)** Emin deęilim.

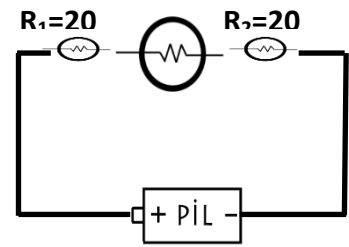
**9.1.** Şekil 8'deki devrede bir pil, ampul ve iki dirençten oluşan bir düzenek gösterilmektedir. İlk önce sadece 10 ohm'luk  $R_1$  direncinin yerine 20 ohm'luk bir direnç bağlanıyor ve Şekil 9 elde ediliyor. Daha sonra Şekil 8'deki 10 ohm'luk  $R_2$  direncinin yerine 20 ohm'luk bir direnç bağlanıyor ve Şekil 10 elde ediliyor. Şekil 9 ve Şekil 10'daki ampullerin parlaklıklarında Şekil 8'deki ampule göre değişiklik olur mu?



**Şekil 8**



**Şekil 9**



**Şekil 10**

Şekil 9'da Şekil 10'da

- (a) Değişir Değişmez
- (b) Değişir Değişir
- (c) Değişmez Değişir

**9.2.** Yukarıda verdiğim cevabın sebebi:

- (a) Akım ampule ulaşmadan önce  $R_1$  direnci tarafından etkilenir.
- (b) Akım ampule ulaşmadan önce  $R_2$  direnci tarafından etkilenir.
- (c) Her iki şekilde de eşdeğer direnç Şekil 8'e göre değiştiğinden, ana koldaki akım da değişir.
- (d) Piller aynı olduğundan ana kollardaki akımlar aynıdır.
- (e)

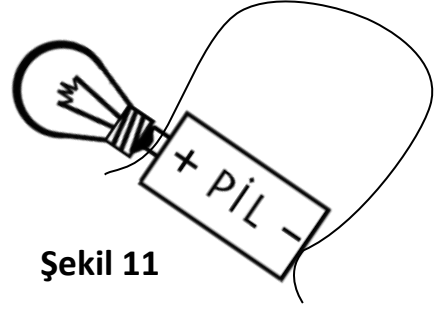
**9.3.** Yukarıdaki iki soruya verdiğim cevaptan;

- (a) Eminim.
- (b) Emin değilim.



10.1. Şekil 11'deki ampul ışık verir mi?

- (a) Evet, ışık verir.
- (b) Hayır, ışık vermez.



Şekil 11

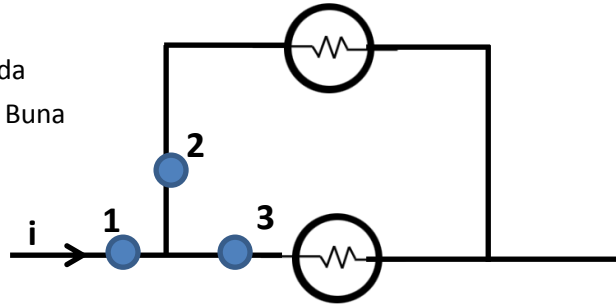
10.2. Yukarıda verdiğim cevabın sebebi:

- (a) Ampul "+" ve "-" yüklerine dokundurulduğu için "+" ve "-" yükler ampulde birleşebilirler.
- (b) Ampul üreticinin "+" kutbuna değiyor.
- (c) Ampulden elektrik akımı geçer.
- (d) Ampulden elektrik akımı geçmez.
- (e)

10.3. Yukarıdaki iki soruya verdiğim cevaptan;

- (a) Eminim.
- (b) Emin değilim.

11.1. Şekil 12'de verilen devre parçasında elektrik akımının yönü ok ile gösteriliyor. Buna göre, 1, 2 ve 3 noktalarındaki akımların büyüklüklerini karşılaştırınız.



Şekil 12

- (a)  $i_1 > i_3 > i_2$
- (b)  $i_1 > i_2 = i_3$

11.2. Yukarıda verdiğim cevabın sebebi:

- (a) Ampullerin dirençleri eşit olduğu için, akım kol ayrımına geldiğinde eşit bir şekilde kollara ayrılır.
- (b) Akım kol ayrımına geldiğinde gidiş yönündeki düz kola daha çok akım, kıvrılan kola ise daha az akım geçer.
- (c)

**11.3.** Yukarıdaki iki soruya verdiğim cevaptan;

(a) Eminim.

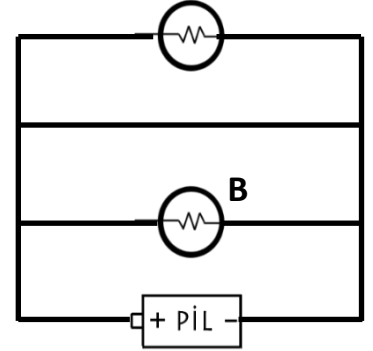
(b) Emin değilim.

**12.1.** Şekil 13'te gösterilen devrede A ve B ampullerinin parlaklıklarını karşılaştırınız.

(a) B ampulü daha parlaktır.

(b) İki ampulün de parlaklıkları eşittir.

(c) İki ampul de yanmaz.



**Şekil 13**

**12.2.** Yukarıda verdiğim cevabın sebebi:

(a) Akım A ve B ampullerinin arasındaki boş telden geçer.

(b) B ampulü pile daha yakındır.

(c) Akım ilk iki kola ayrıldığında, akımın yarısı B ampulüne geçer. Daha sonra tekrar kol ayrımına geldiğinde akımın hepsi boş telden geçer ve pile döner.

(d) İki ampul birbirine paralel olduğundan üzerlerinden eşit büyüklükte akım geçer.

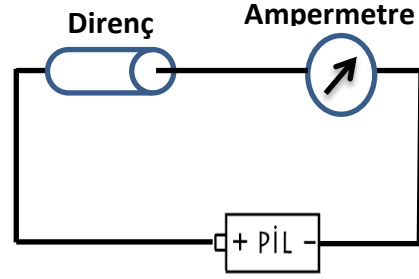
(e)

**12.3.** Yukarıda iki soruya verdiğim cevaptan;

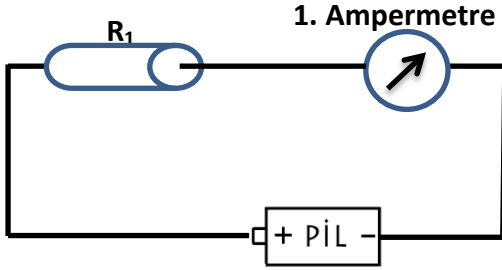
(a) Eminim.

(b) Emin değilim.

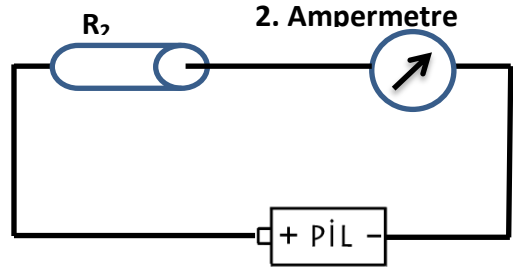
1. 12 voltluk pil takılmış Şekil 1'deki devrede, direncin hamur gibi esnek bir malzemeden yapıldığı varsayılmıştır. Yani **direncin** şekli elle kolayca değiştirilebilmektedir. **Pilin değiştirilemediği** bu devrede Ampermetreden geçen akımı arttırmak için neler **yapabileceğinizi** açıklayarak yazınız.



Şekil 1



Şekil 2



Şekil 3

2. Şekil 2 ve şekil 3'teki dirençler şekilleri bakımından özdeşler. Devredeki piller de özdeş olduklarına göre, 1. Ampermetreden 2. Ampermetreye göre daha fazla akım geçmesinin sebebi sizce neler olabilir? Açıklayarak yazınız.

## **EK D – Görüşme Formu**

- **Bölüm I: Önceki deneyimler**
  - Her zaman ulaşabileceğin bir bilgisayar var mı?
  - İnternete bağlantın var mı?
  - Bilgisayarı daha çok hangi amaçla kullanıyorsun?
  - Oyun oynuyor musun? Hangi tür oyunları tercih ediyorsun?
  - Fizik dersi hakkında neler düşünüyorsun?
  - Basit elektrik devreleri konusundaki bilgi düzeyini nasıl değerlendirirsin?
- **Bölüm II: elektroGame hakkındaki öğrenci görüşleri**
  - elektroGame hakkındaki ilk izlenimlerin nelerdir?
  - Oyunda hoşuna giden şeyler nelerdir?
  - Sıkıcı, zorlayıcı bulduğun sevmediğin yerler var mı?
  - Eğitim haritasının yeterli olduğunu düşünüyor musun?
  - Oyun içinde diğer oyuncularla etkileşimin nasıldı?
  - Oyundaki rekabet hissinin arttırılması konusunda ne düşünüyorsun?
  - elektroGame'i daha çok oyun olarak mı yoksa ders olarak mı sınıflandırırın?
  - Herhangi bir dersi bir sene boyunca elektroGame ile işlemek ister misin?
  - Eklemek istediğin başka bir şey var mı?