



T.C.
ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

STAJ RAPORU

SERCAN DORMAN

NİTRA OYUN YAZILIM LTD. ŞTİ.

TEMMUZ 2017
İSTANBUL



T.C.
ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

FOTOĞRAF

Öğrencinin Adı ve Soyadı : Sercan DORMAN
Sınıf ve No : 4 / 120401054
Kurumun Adı ve Adresi : Nitra Oyun Yazılım Ltd. Şti. / KOSGEB Teknoloji Geliştirme Merkezi,
İstanbul Aydın Üniversitesi H Blok Ofis:6022, Beşyol Mah. İnönü Cad. No:38 Küçükçekmece / İstanbul
Staja Başlayış ve Bitiş Tarihleri : 24/07/2017 – 18/08/2017
Raporun Sunulduğu Tarih :
Öğrencinin İmzası :

Raporu İnceleyen Öğretim Üyesi :
Verilen Not : Başarılı/Başarısız :
Tarih :
İmza :

T.C.
ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

STAJ YÖNERGESİ

MADDE: 1- 2. sınıftan 3. sınıfa ve 3.sınıftan 4. sınıfa geçen her öğrencinin yaz stajı yapması zorunludur. Bu stajlara 2. ve 3. sınıf stajları denilecektir.

MADDE: 2- Staj özel haller haricinde yaz aylarında yapılmalıdır.

MADDE: 3- 2. sınıf ve 3. sınıf stajları en az 4 hafta (20 iş günü) olmak üzere toplam 8 hafta (40 iş günü) olarak belirlenmiştir.

MADDE: 4- 4 hafta (20 iş günü) olan staj süresi bölünemez. Staj devamsızlıkları Staj Kurulu Başkanlığı'na değerlendirilir. **5 (Beş) günü aşan devamsızlık stajın yenilenmesini gerektirir.**

MADDE: 5- İlke olarak stajlar, olanaklı ise, işletmelerin bilgi işlem merkezlerinde, yazılım projesi geliştiren özel ya da devlet kuruluşlarında yapılacaktır. Böylece öğrencinin daha etkin olması, işletmelerdeki sorunları görmesi ve projelendirme süreçlerini gözlemlenmesi amaçlanmıştır.

MADDE: 6- Staj yapacak olan öğrencinin, ilgili kuruluşun tüm yasal ve yönetsel kurallarına uyması zorunludur. Aksi davranışları tespit edilen öğrencilerin staj görevi geçersiz sayılır ve haklarında disiplin soruşturması açılması için durum Fakülte Dekanlığı'na bildirilir.

MADDE: 7- Öğrencilerin stajlarını farklı kurumlarda yapması önerilir. Bölüm onayı alınmadan iki staj aynı kuruluştaki yapılamaz.

MADDE: 8- Stajlardan biri, bölümün ihtiyaç görmesi halinde, Üniversite laboratuvarlarında bölümün onayladığı bir danışman gözetiminde yapılabilir.

MADDE: 9- Her öğrenci staj ile ilgili gerekli belgeleri (**Staj Raporu Yazım Kılavuzu, Staj Başarı Belgesi**) bölümün internet sayfasından temin eder.

MADDE: 10- Öğrenci, staj süresince, yürüttüğü çalışmalardan yararlanarak staj raporunu bölümün öngördüğü “**Staj Raporu Yazım Kılavuzuna**” uygun bir şekilde hazırlar. Staj raporunun her sayfası kurumdaki ilgili yetkili kişi tarafından imzalanır. Staj raporunun “Devam Çizelgesi” sayfası rapor sahibince imzalanır ve kuruma onaylatılır.

MADDE: 11- “Staj Başarı Belgesi” staj sonunda ilgili kuruluşun yetkilisince doldurulduktan sonra, öğrenci tarafından **kapalı ve imzalı bir zarf içinde** Çanakkale Onsekiz Mart Üniversitesi Bilgisayar Mühendisliği Bölümü Staj Kurulu Başkanlığı’na elden veya posta ile iletilir.

MADDE: 12- Staj raporları kişiseldir, aynı staj raporu birden fazla öğrenci adına hazırlanamaz.

MADDE: 13- “Staj Raporu” ve “Staj Başarı Belgesi”, **en geç, izleyen ders dönemi başlangıcından 1 (bir) ay içerisinde bölüme teslim edilir. Geç gelen raporlar kabul edilmez ve öğrencinin stajı geçersiz sayılır.**

MADDE: 14- Bölüm tarafından oluşturulan Staj Komisyonu, bölümün belirlediği bir tarihte staj raporlarını değerlendirir. Bu değerlendirmede öğrenciler sözlü sınava alınır, yazmış oldukları staj raporlarının ve içeriğinin yazım kılavuzuna göre uygunluğu ve teknik açıdan yeterliliği sorgulanır. Bölüm Staj Değerlendirme Formu doldurularak öğrencilerin staj başarı durumları tespit edilir. Sonuçlar komisyon toplantı tarihinden sonraki 2 hafta içerisinde bölüm ilan panosuna asılarak öğrencilere açıklanır.

T.C.
ANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ÖĞRENCİNİN ÖZGEÇMİŞ FORMU

ADI SOYADI : SERCAN DORMAN

DOĞUM TARİHİ : 04/06/1993

TELEFON NO : (+90) 5380242011

EV ADRESİ : Talaytepe Mah. Laleş Bulvarı Memursen Konutları DG 11B
Blok D:3 Kayapınar / DİYARBAKIR

ÖĞRENCİNİN ÖZGEÇMİŞİ

MEZUN OLDUĞU LİSE : Diyarbakır Anadolu Teknik ve Endüstri Meslek Lisesi

MEZUN OLDUĞU BRANŞ : Web Tasarım

MEZUNİYET TARİHİ : 2010

Daha Önce Staj Yapmış ise Kurum Adı ve Çalışma Tarihleri : Yapılmadı.

Bu İşyerinde Çalıştı ise İşyeri Adı ve Çalışma Tarihleri : Çalışılmadı.

Diğer Hususlar :

Öğrencinin
Adı Soyadı ve İmzası

SERCAN DORMAN

Not: Verilen Bilgilerden Öğrenci Sorumludur.

T.C.
ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
STAJ EĞİTİM ÇALIŞMASI GÜNLÜK DEVAM ÇİZELGESİ

Sıra no	Tarih	İmza
1	24/07/17	
2	25/07/17	
3	26/07/17	
4	27/07/17	
5	28/07/17	
6	31/07/17	
7	01/08/17	
8	02/08/17	
9	03/08/17	
10	04/08/17	
11	07/08/17	
12	08/08/17	
13	09/08/17	
14	10/08/17	
15	11/08/17	
16	14/08/17	
17	15/08/17	
18	16/08/17	
19	17/08/17	
20	18/08/17	
Çalıştığı Bölüm Oyun Yazılımı		Bölüm Yetkilisinin İmza ve Kaşesi

FİRMA HAKKINDA BİLGİ

Gamelab İstanbul, oyun yazılımları ve oyun motorları geliştiren bir şirkettir. Kuruluş amacı ve genel hedefi, uzun vadede AAA kalitesinde ürünlerden oluşan büyük bir oyun portföyü kurmaktır. Gamelab İstanbul'un 2017 başında faaliyete geçmesine karşın, kurucu ekibi mobil oyun sektörünün dünyadaki en eski ve tecrübeli ekiplerinden birisidir.

ÖZET TARİHÇE

MOBILENTER DÖNEMİ

Kurucu ekibin oyun geliştirme macerası 2006 yılında Mobilenter Mobil Oyun Stüdyoları'nı kurarak başladı. Mobilenter, sektörün ülkemizdeki ve dünyadaki ilk firmalarından birisidir. 2010 sonuna vardığında, firma 21 oyunluk bir portföye ve 3 ofise sahip olmuş durumdaydı. Ofisler İstanbul, Lancaster/İngiltere ve North San Jose/ABD lokasyonlarında bulunuyordu. O dönemde dünyanın en büyük appstore sitesi olan Getjar'ın tarihinde en çok indirilen 10 oyundan 2 tanesi Mobilenter oyunlarıdır. Firmanın oyunları 11 dilde hazırlıyor ve 46 ülkede 100'ün üzerinde operatörde yayınlanıyordu. İlerleyen yıllarda Java telefonlarının piyasadan kalkması sürecinde Mobilenter küçülmeye gitti ve yurtdışındaki ofislerini kapattı. Ekip 2014 Nisan'ında eski Java oyunlarının Android versiyonlarını yayınlamak üzere Nitra Oyun Yazılım Ltd. Şti. firmasını kurdu.

NİTRA DÖNEMİ

Kurucu ekip 2007 başından 2016 sonuna kadar geçen 11 yıllık süreçte, 1 OpenGL ES benzeri grafik arayüzü, 13 oyun motoru ve 26 oyun geliştirip yayınladı. Nitra, küçük ölçekli bir firma olmasına karşın, bugün kendi alanında dünyanın en büyük know-how'ına sahip firmalarından bir tanesidir. Oyunların motor yazılımı dahil tamamını geliştirmesi sebebiyle, Türkiye'de AR-GE desteği almaya hak kazanan ilk oyun yazılım firması Nitra Oyun Yazılım Ltd. Şti. olmuştur.

GAMELAB İSTANBUL DÖNEMİ

KOSGEB'in İstanbul Aydın Üniversitesi Teknoloji Geliştirme Merkezi içerisinde ücretsiz olarak tahsis ettiği ofiste, Nitra yöneticileri ve 24 üniversite öğrencisinin katılımıyla kurulan Gamelab İstanbul, 13 Şubat 2017 günü Türkiye'nin en büyük mobil oyun yazılım ekibi olarak faaliyetine başladı.

İÇİNDEKİLER

GİRİŞ.....	I
Android Studio ve emülatörlerin kurulumu.....	1
Kaynakların ekrana çizdirilmesi.....	2
Ses ve müziğin eklenmesi.....	3
Buton ekleme ve tıklama olayı.....	4
Photoshop kullanımı.....	5
Oyuna logo ekleme.....	6
Resimlerin yüklenmesi ve tıklama olayları.....	7
Özellik (skill) ekleme.....	8
Özelliklerin (skill) kodlaması.....	9
Skor tablosu tasarımı.....	10
Oyunun tüm ekran boyutlarına göre ölçeklenmesi.....	11
Çarpışma kontrolü.....	12
Puanlama sistemi.....	13
Dil desteğinin veritabanına eklenmesi.....	14
Skor tablosunun farklı dildeki görünümü.....	15
Özelliklerin (skill) yeniden dizaynı.....	16
Özelliklerin zamanının ve en yüksek skorun veritabanına kaydı.....	17
Devam etmek ister misiniz ekranının buton tasarımı.....	18
Devam etmek ister misiniz ekranını çizdirme ve dil desteği.....	19
Oyundaki lazere ses efekti ekleme.....	20
Stajer memnuniyet anketi.....	21

GİRİŞ

Staja başlamadan önce oyun yazılımı nedir, nasıl geliştirilir gibi sorular hakkında verebilecek cevaplarım yoktu. Şirkette staja başladıktan sonra aldığım bir haftalık eğitim ile birlikte bu soruların cevaplarını öğrenmiş oldum ve benim için işler daha zevkli hale gelmeye başladı.

Şirketin devam eden 4 tane oyun yazılımı bulunmaktaydı ve eğitimden sonra direkt projelere dahil edildim. Bu benim için çok önemli bir deneyimdi ve gerçekten çok şey öğrendim, çok yol katettim. Daha önce iyi bir Java eğitimi almıştım ve bu yönde kendimi geliştirmek istiyordum. Bu yüzden Android projelerinde çalışmak benim için iyi bir deneyim oldu.

Projelerde Android Studio ve Git kullanılarak projeler geliştirildi. İkisini de daha önce hiç kullanmamıştım ve özellikle Git aracını kullanmayı öğrenmek benim için önemli bir kazanımdı.

STAJ METNİ

Sayfa No : 1

Bugün, bir haftalık eğitimin ilk günüydü. Bu yüzden öncelikle şirketin devam eden projeleri hakkında bilgi edindik. Devam eden 4 adet oyun projesi bulunmaktaydı. Bunlar; futbol oyunu olan Taso16, Candy Crush tarzında fakat karşılıklı oynanabilen Jewel Multiplayer, kule savunma oyunu olan Fantastic War ve RPG oyun türünde Robocowboy'du.

Öncelikle Android Studio ortamında çalışacağımız için Android Studio ve oyunların üzerinde koşacağı emülatörlerin kurulumunu gördüm. Daha sonra Git ortamında birer hesap oluşturduk ve Git'in çalışma mantığını öğrendim. Projelerin kendi depomuza (repository) nasıl çatallanacağını (fork) ve Android Studio ortamına nasıl klonlanacağını (clone) öğrendim.

Çalışma Tarihi : 24/07/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 2

Bugün, oyunlarda kullanılan resimlerin ekrana nasıl yerleştirildiğini, arka plan çizdirmeyi, karakterlerin, atılan mermilerin vb. ekranda yani oyunun içinde nasıl hareket ettirildiğini öğrendik.

Bir karakterin sayfa içindeki tüm hareketlerinin, oyunun kaynaklarının tutulduğu klasörde (assets) tek bir parça halinde tutulduğunu öğrendim. 5 satır 5 sütun olacak şekilde 100x100'lük yani toplamda 500x500'lük bir resimde 25 tane hareketin tutulduğunu ve bunları ekrana çizdirmek için kodlamada Rectangle denilen sınıfın metodları ile 100x100'lük kareler oluşturup bu kareleri sıra sıra oynatarak yani Rectangle'ın kaynağını (source) değiştirerek karakterin oynatıldığını öğrendim. Daha sonra ise bu karakterin ekranda yer, yön değiştirmesi için hedefinin (destination) nasıl değiştirildiğini gördüm. Daha sonra ise bunların ekrana çizdirilmesini öğrendim.

Çalışma Tarihi : 25/07/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 3

Bugün, ses ve müziğin nasıl kesildiğini, oyuna nasıl eklendiğini, nasıl oynatıldığını ve nasıl durdurulduğunu öğrendim.

Örnek kod parçası:

```
if(root.getVoice()){  
    root.soundManager.play(oynatilacakSes);  
}  
if(root.getVoice() == false){  
    root.soundManager.stop(oynatilacakSes);  
}
```

Çalışma Tarihi : 26/07/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 4

Bugün, menü oluşturmaya, buton eklemeyi, butona tıklayınca ne olacağını nereden kodlanacağını ve hangi metodun çağrılacağını gördüm. Kanvas kavramını ve kanvaslar arası geçişleri öğrendim.

Örnek kod parçası :

```
private class MenuCanvas{
    private void touchUp(int x, int y){
        if(playButtonDst.intersect(x, y)){
            GameCanvas gc = new GameCanvas();
            root.canvasManager.setCurrentCanvas(gc);
        }
    }
}
```

Çalışma Tarihi : 27/07/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 5

Bugün, eğitimin son günüydü. Kısa bir Photoshop eğitimi verildi ve örnek bir tasarım yaptık. Var olan resimleri düzenledik ve küçük bileşenleri bir araya getirerek bir oyun arka planı tasarladık. Katmanlarla (layer) çalışmayı öğrendim.

Çalışma Tarihi : 28/07/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

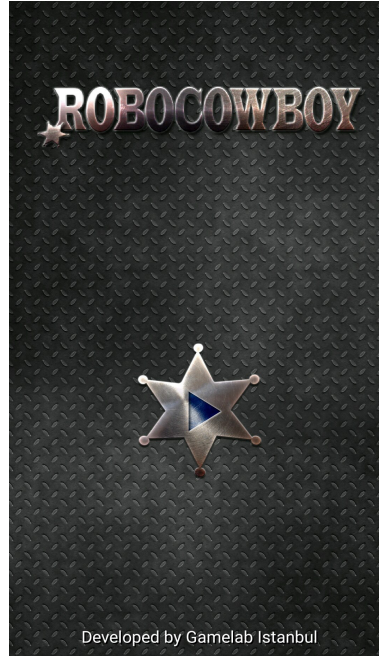
STAJ METNİ

Sayfa No : 6

Bugün, staj gören öğrenciler yedişer kişilik iki gruba ayrıldı ve yakın zamanda yayınlanması planlanan iki projeye atandı. Ben Robocowboy adlı RPG oyununa atandım. Projeyi Git üzerinden kendi depoma çatallayıp klonladıktan sonra bir süre oyunun çalışma mantığını anlamaya çalıştım. Daha sonra ilk görevim verildi ve oyunun logosunun değiştirilmesi istendi.

Yaptığım kodlamadan bir kesit:

```
karakter = Utils.loadImage(root, "images/logo.png");  
logoSrc = new Rect(0, 0, 500, 500);  
logoX = 0; logoY = 0;  
logoGenislik = getWidth(); logoYukseklık = getHeight();  
logoDst = (logoX, logoY, logoX + logoGenislik, logoY + logoYukseklık);  
canvas.drawBitmap(logo, logoSrc, logoDst, null);
```



Resim 1. Logonun oyun ekranında görünümü

Çalışma Tarihi : 31/07/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 7

Bugün kodları anlamak için zaman ayırdım öncelikle. Oyunlarda kullanan kaynakların hangi metodlarda hangi metodların yardımıyla çağrıldığını, bunların ekrana nerede ve hangi metodların yardımıyla çizdirildiğini, değişkenlere ilk değer atamasının nerede yapıldığını, ekrana tıklama olaylarının nerede işlendiğini vb. şeyleri öğrendim.

Çalışma Tarihi : 01/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 8

Bugün oyuna yeni özellikler (skill) ekleme görevi verildi. Oyunda, hızlandırma, tüm düşmanları yok etme ve kalkana sahip olma özellikleri vardı. Ben, bir adet rastgele ve iki adet negatif özellik ekledim. Negatif özelliklerden biri bizim ana karakterimizi yavaşlatan özellikti. Öteki ise rakiplerimize kalkan veren bir özellikti. Rastgele özellik ise edinildiği takdirde, iki negatif, üç pozitif olmak üzere var olan toplam 5 özellikten birini rastgele sağlayacaktı. Bugün bunların Photoshop ile tasarımını yaptım.



Resim 2. Özelliklerin çizimleri



Resim 3. Skillerin oyun içi görünümü

Çalışma Tarihi : 02/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 9

Bugün eklenecek olan özelliklerin kodlamasını yaptım.

Kodlamanın bir bölümü:

```
if (skillNumarasi == 5 ) {  
    dusmanKalkani = true;  
}  
  
if (skillNumarasi == 4) {  
    ilerlemeHiziX[0] = 5; // default 20  
    ilerlemeHiziY[0] = 5; // default 20  
}  
  
if(skillSayaci == bitisSuresi) {  
    skill = false;  
    ilerlemeHiziX[0] = 20;  
    ilerlemeHiziY[0] = 20;  
}
```

Bu kodlama tamamlandıktan sonra Android Studio üzerinden Commit, Push, Merge Request olaylarını yapmayı öğrendim.

Çalışma Tarihi : 03/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 10

Bugün, oyun sona erince ekranda gösterilen skor tablosunun tasarımının değişmesi istendi. Kaynakların bulunduğu klasörlerde var olan butonlar ve arka planları kullanıp, kodlarla birleştirerek aşağıdaki tasarımı gerçekleştirdim.

Kodlamanın bir bölümü:

```
exit_buttondst = new Rect((int) ((972 - 256) * screenrate),  
                           (int) ((1094.4) * screenrate),  
                           (int) ((972) * screenrate),  
                           (int) (((1094.4)+256)*screenrate));
```



Resim 4. Skor tablosu

Çalışma Tarihi : 04/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 11

Bugün oyunun tüm cihazların ekranlarında orantılı bir şekilde açılabilmesi ve oynanabilmesi için tüm bileşenleri oranladık.

Kodlamanın bir bölümü:

```
private float screenrate = getWidth() / 1080f;
public int getScreenRate(int value) {
    return (int) (value * screenrate);
}
private void drawSkills(Canvas canvas){
    if (skill) {
        skillDst.set(skillX, skillY,
            skillX + getScreenRate(skillW), skillY + getScreenRate(skillH));
        canvas.drawBitmap(skillbitmap, skillkaresrc, skillkaredst, null);
    }
}
```

1533	-	fireSolUstdst.set(circlex, circley, circlew, circleh);//firesolÜst köşe
1534	-	fireSagUstdst.set(getWidth() - circlew, circley, getWidth() - circlex, circleh);//firesağÜst köşe
1535	-	fireSolAltdst.set(circlex, getHeight() - circlew, circlew, getHeight() - circleh);//fireSolAlt köşe
1536	-	fireSagAltdst.set(getWidth() - circlew, getHeight() - circlew, getWidth() - circlex, getHeight() - circleh);//firesağAlt köşe
1548	+	fireSolUstdst.set(root.getScreenRate(circlex), root.getScreenRate(circley), root.getScreenRate(circlew), root.getScreenRate(circleh));//firesolÜst köşe
1549	+	fireSagUstdst.set(getWidth() - root.getScreenRate(circlew), root.getScreenRate(circley), getWidth() - root.getScreenRate(circlex), root.getScreenRate(circleh));//firesağÜst köşe
1550	+	fireSolAltdst.set(root.getScreenRate(circlex), getHeight() - root.getScreenRate(circlew), root.getScreenRate(circlew), getHeight() - root.getScreenRate(circleh));//fireSolAlt köşe
1551	+	fireSagAltdst.set(getWidth() - root.getScreenRate(circlew), getHeight() - root.getScreenRate(circlew), getWidth() - root.getScreenRate(circlex), getHeight() - root.getScreenRate(circleh));//firesağAlt köşe
1537	1552	//end region circleengel
1538	1553	}

Resim 5. Oranlanan bileşenlerden bazıları

Çalışma Tarihi : 07/08/2017

Bölüm Yetkilisinin İmza ve Kaşesi :

Çalıştığı Bölüm : Oyun Yazılımı

STAJ METNİ

Sayfa No : 12

Bugün karakterlerin birbiriyle ve engellerle olan çarpışma olayını kontrol edip düzeltmeler yaptım. Ayrıca mermilerin karakterlere ve engellere olan çarpışma olaylarını da kontrol ettim. Ve bunları tüm ekran boyutlarında çalışacak şekilde düzenledim. Daha sonra oyun içindeki yazıların yeniden boyutlandırılmasını yaptım.

Kodlamanın bir bölümü:

```
// Yazıyı yeniden ve tüm ekranlarda çalışacak şekilde boyutlandırma;  
paint.setTextSize((int) (110 * screenrate));  
// Ekranı yeni yazı çizdirme, koordinatlarını belirleme ve boyutlandırma;  
canvas.drawText("Start Game", (int) ((score_dialogue_x + 50 + 80) * screenrate),  
                (int) ((score_dialogue_y + 70 + 1200) * screenrate), paint);  
// Mermileri tüm ekranlara göre boyutlandırma;  
mermikare.set(mermix[k][f], mermiy[k][f], mermix[k][f] +  
              root.getScreenRate(mermiw), mermiy[k][f] + root.getScreenRate(mermih));
```

Çalışma Tarihi : 08/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 13

Oyunda iki farklı mod vardı. Bunlardan biri ölüm maçıydı öteki ise zamana karşı olan moddu. Ölüm maçında zaman sınırı yoktu ve karakter ölene kadar oynanabiliyordu. Zamana karşı olanda ise hem zaman sınırı hem de ölüm kontrolü vardı. Bu iki modun puanlaması ayrı olması gerekiyordu ve bunu ben sağladım. Ayrıca skillerde olan bazı hataları düzelttim.

Kodlamanın bir bölümü:

```
if (skillnum == 1) {  
    if (root.getVoice()) {  
        root.soundManager.play(SESEFEKTI_PATLAMA);  
    }  
    if (root.getVoice() == false) {  
        root.soundManager.stop(SESEFEKTI_PATLAMA);  
    }  
    if (root.getModeNo() == 01) {  
        score += 10;  
    } else {  
        score += 5;  
    }  
}
```

Çalışma Tarihi :09/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 14

Bugün oyunun veritabanının nasıl çalıştığıyla ilgilendim ve oyuna dil desteği eklenmesi görevini üstlendim. Dil desteğinin veri tabanına kaydedilmesi gerekiyordu çünkü kanvas değıştikçe dil aynı kalmalıydı ve oyunda çıkacak olan her ekranın ve her yazının ana menüde seçili olan dile göre gelmesi gerekiyordu.

Kodlamanın bir bölümü:

```
private static final String PREFKEY_LANGUAGE = "language";

public void setLanguage(String language) {
    this.language = language;
    prefs.putString(PREFKEY_LANGUAGE, language);
}

public String getLanguage() {
    return language;
}
```

Çalışma Tarihi :10/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

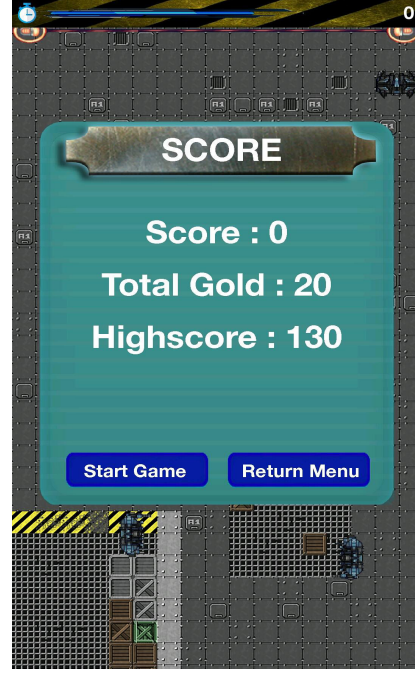
STAJ METNİ

Sayfa No : 15

Bugün oyun bitince ekranda çıkan skor tablosunun dizaynını değiştirdim ve dil desteği ekledim.



Resim 6. Türkçe



Resim 7. İngilizce

Kodlamanın bir bölümü:

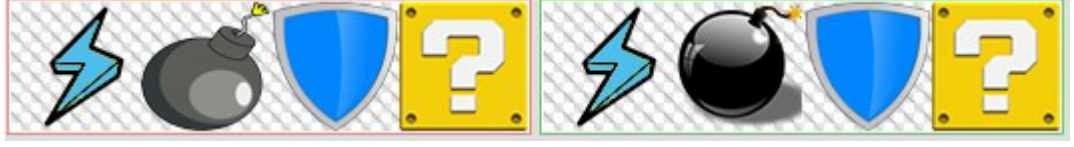
```
paint.setColor(Color.WHITE);  
paint.setTextAlign(Paint.Align.CENTER);  
paint.setTextSize((int) (90 * screenrate));  
paint.setTypeface(Typeface.createFromAsset(baseapp.activity.getAssets(),  
"fonts/HelveticaNeueBold.ttf"));
```

Çalışma Tarihi : 11/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

Bugün skillerden birinin görüntüsünü Photoshop ile değiştirdim ve skillerin ekranda durma sürelerini birbirinden farklı hale getirdim.



Resim 9. Özelliğin eski ve yeni hali

Kodlamanın bir bölümü:

```
private int speedSkillCounter, shieldSkillCounter, otherSkillCounter,  
        skillSpawnCounter;  
private void skillControl() {  
    skillcounter++;  
    speedSkillCounter++; shieldSkillCounter++; otherSkillCounter++;  
    skillSpawnCounter++;  
    if (speedSkillCounter >= 70 && skillnum == 0) {  
        skill = false;  
    }  
    if skillSpawnCounter >= 150) {  
        speedSkillCounter = 0; shieldSkillCounter = 0; otherSkillCounter = 0;  
        skillSpawnCounter = 0;  
    }  
}
```

Çalışma Tarihi : 14/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 17

Bugün skillerin ekranda kalma zamanlarını veritabanına ekledim. Ayrıca farklı oyun modlarında farklı en yüksek skor olmasını sağladım.

Kodlamanın bir bölümü:

```
private int score;
private int deathmatch_score;
if (root.getModeNo() == 0l) {
    score += 10;
} else {
    deathmatch_score += 5;
}

if (root.getModeNo() == 0l) {
    if (score > root.getUserScore()) {
        root.setHighScore(score);
        root.topTenTable(score);
    }
} else {
    if (deathmatch_score > root.getDM_UserScore()) {
        root.setDM_HighScore(deathmatch_score);
        root.topTenTable(deathmatch_score);
    }
}

isGameOverVisible = true;
}
```

Çalışma Tarihi : 15/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

Bugün, karakter öldükten sonra kullanıcıya 100 altın vererek veya reklam izleyerek oyuna devam etme hakkı tanıma görevini üstlendim. Bunun için kullanılacak olan butonları Photoshop ile tasarladım ve bunları farklı dillere göre dizayn ettim.



Resim 10. Altın vererek veya reklam izleyerek devam etme butonları

Kodlamanın bir bölümü:

```
private int continueButton_x, continueButton_y, continueButton_w,
        continueButton_h, watchvideoButton_x, watchvideoButton_y,
        watchvideoButton_w, watchvideoButton_h, deadCount,
        continueButtonWaitingTime;

continueButton = Utils.loadImage(root, "images/continue_Button.png");
watchvideoButton = Utils.loadImage(root, "images/watchvideo.png");
if (continuegame) {
    if (deadCount < 2 && continueButtonWaitingTime < 50
        && !karakterhayatta[0]) {
        canvas.drawBitmap(continueButton, continueButtonsrc,
            continueButtondst, null);
        canvas.drawBitmap(watchvideoButton, watchvideoButtonsrc,
            watchvideoButtondst, null);
        continueButtonWaitingTime++;
    }
}
```

Çalışma Tarihi : 16/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 19

Bugün oyuna devam etme seçeneğinin ekranda gösterilmesi görevini üstlendim. Ayrıca dil desteği sağladım. Ve karakter devam etmeyi seçtiği takdirde kaldığı yerden devam edebilmesini sağladım.



Resim 11. Oyun içi görünümü

Kodlamanın bir bölümü:

```
if (root.getLanguage().equalsIgnoreCase("en")) {  
    canvas.drawBitmap(continueandwatchEN, continueandwatchsrc,  
        continueandwatchdst, null);  
}
```

Çalışma Tarihi : 17/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :

STAJ METNİ

Sayfa No : 20

Bugün karakterin yaşam süresi bitince yeniden oynama şansı vermeyi ve oyunda karakteri öldürmeyi amaçlayan lazere, lazerin geldiğine ve lazerin ekranda olduğuna dair sesler ekledim.

Kodlamanın bir bölümü:

```
public void touchUp(int x, int y) {  
    if (oyunbitti && continuegame) {  
        if (continueButtondst.contains(x, y)) {  
            if (root.getCoin() >= 100) {  
                if (batterybarw <= 0) {  
                    batterybarw = root.getScreenRate(700);  
                }  
                root.setCoin(root.getCoin() - 100);  
                karakterhayatta[0] = true;  
                oyunbitti = false;  
                continue_right = false;  
            }  
        }  
    }  
}
```

Çalışma Tarihi : 18/08/2017

Çalıştığı Bölüm : Oyun Yazılımı

Bölüm Yetkilisinin İmza ve Kaşesi :