

GAZİ ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ

BM 314 YAZILIM MÜHENDİSLİĞİ  
SDD BELGESİ

Abdullah Akalın  
Bekir Aydın  
Karim El Guermai  
Muhammed Emre Emrah

05.04.1017

# İçindekiler

<b>1 Genel Bakış</b>	<b>1</b>
1.1 Kapsam . . . . .	1
1.2 Amaç . . . . .	1
1.3 Hedef Kitle	1
<b>2 Tanımlar</b>	<b>1</b>
<b>3 Yazılım Tasarım Tanımları İçin Kavramsal Model</b>	<b>2</b>
3.1 Bağlam Dahilinde Yazılım Tasarımı . . . . .	2
3.2 Yazılım Döngüsü İçinde Yazılım Tasarımı . . . . .	2
3.2.1 Belgenin Hazırlanmasındaki Etkiler . . . . .	2
3.2.2 Yazılım Döngüsü Ürünlerindeki Etkiler . . . . .	2
3.2.3 Tasarımın ve Tasarım Rolünün Tasdiki . . . . .	2
<b>4 Tasarım Bilgi İçeriği</b>	<b>2</b>
4.1 Giriş . . . . .	2
4.2 YTB Tanımlaması . . . . .	3
4.2.1 Tasarım Paydaşları ve Beklentileri . . . . .	3
4.2.2 Tasarım Beklentileri . . . . .	3
4.2.3 Tasarım Bakış Açıları . . . . .	4
4.2.4 Tasarım Unsurları . . . . .	4
4.2.5 Tasarım Katmanları . . . . .	4
4.2.6 Tasarım Gerekçeleri . . . . .	4
4.2.7 Tasarım Dilleri . . . . .	4
<b>5 Tasarım Bakış Açıları</b>	<b>4</b>
5.1 Giriş . . . . .	4
5.2 Bağlam Bakış Açısı . . . . .	5
5.2.1 Tasarım Beklentileri . . . . .	5
5.2.2 Tasarım Unsurları . . . . .	5
5.3 Yapısal Bakış Açısı . . . . .	5
5.3.1 Tasarım Beklentileri . . . . .	5
5.3.2 Tasarım Unsurları . . . . .	7
5.4 Mantıksal Bakış Açısı . . . . .	7
5.4.1 Tasarım Beklentileri . . . . .	7
5.4.2 Tasarım Unsurları . . . . .	7
5.5 Bilgisel Bakış Açısı . . . . .	9
5.5.1 Tasarım Beklentileri . . . . .	9
5.5.2 Tasarım Unsurları . . . . .	10

5.6	Etkileşimsel Bakış Açısı . . . . .	10
5.6.1	Tasarım Beklentileri . . . . .	10
5.7	Arayüz Bakış Açısı . . . . .	10
5.8	Durum Dinamikleri Bakış Açısı . . . . .	11
5.8.1	Tasarım Beklentileri . . . . .	11
5.8.2	Tasarım Unsurları . . . . .	11
<b>6</b>	<b>Sonuç</b>	<b>15</b>

# 1 Genel Bakış

## 1.1 Kapsam

Bu belge, Kahvaltı oyununun tasarımını ayrıntıyla ele alacaktır. Yazılım ürünün geliştirilme ve tasarım süreçlerini belirtmek için bakış açıları ele alınacaktır ve bu bağlamdaki tasarım beklentileri açıklanacaktır. Bununla birlikte her bakış açısıyla ilintili tasarım unsurlarından bahsedilecektir. Hülasa, bu belge projenin yapılandırılışı hakkında etraflı bilgi sunacak ve sistemin gelişim evrelerine ışık tutacaktır.

## 1.2 Amaç

Yazılım tasarımı belgesi Kahvaltı oyununun yapısal modelini sağlar. Bu belgenin en önemli ve ana gayesi, tasarım unsurlarına dair kapsamlı açıklamalar getirmek ve bu unsurların birbirleriyle olan etkileşimlerini göstermektir. Denebilir ki yazılım tasarım belgesi, geliştiriciler için pek yüksek önemi haiz bir belgedir ve dikkatle incelenmesi gerekir.

## 1.3 Hedef Kitle

Bu belgenin hedef kitle, bu projenin yöneticisi, geliştiricileri, kısmen kullanıcıları ve testçileridir. Belge, belirlenmiş hedef kitesine yönelik tasarım ve yapısal bilgiler sağlayacaktır.

# 2 Tanımlar

**UML:** Unified Modeling Language

**YTB:** Yazılım Tasarım Belgesi

**SRS:** Yazılım Gereksinim Belgesi

**Java:** Programlama dili.

**LibGDX:** Java oyun kütüphanesi.

**SQLite:** Hafif veri tabanı programı.

## **3 Yazılım Tasarım Tanımları İçin Kavramsal Model**

### **3.1 Bağlam Dahilinde Yazılım Tasarımı**

Projede nesne yönelimli tasarım anlayışı benimsenecektir. Bunun başlıca sebebi taşınabilirlik, tekrar kullanılabilirlik sağlaması ve projedeki bileşenler arasında bütünlüğü temin etmesidir.

### **3.2 Yazılım Döngüsü İçinde Yazılım Tasarımı**

#### **3.2.1 Belgenin Hazırlanmasındaki Etkiler**

Daha önce sunulan yazılım gereksinimleri belgesi, bu projenin tasarımına ışık tutar. Buna göre, bu belge yazılım tasarım belgesinde belirtilen ve ele alınan gereksinimler doğrultusunda hazırlanmıştır.

#### **3.2.2 Yazılım Döngüsü Ürünlerindeki Etkiler**

Bu belge, daha önce sunulan yazılım gereksinim belgesiyle birebir etkileşimlidir. Ayrıca yazılım geliştirme süresince projeye dair bir kısım işlev ve özellikte ekleme/çıkarma yapılmıştır. Benzer şekilde bir kısım gereksinim ve kısıtlamalar da gereksinim belgesinin ardından güncellenmiştir.

#### **3.2.3 Tasarımın ve Tasarım Rolünün Tasdiki**

Test senaryoları yazılım doğrulanmasında çok önemli bir rol oynar. Bu yazılım çeşitli test senaryoları ile yazılımın tüm işlevselliğini kapsayacak testlere tabi tutulacaktır. Denebilir ki, yazılımın başarısı, test sonuçlarının başarısına doğrudan bağlıdır. Test ve sınama işlemlerinin ardından, gerçekleştirilen projenin gerçekleştirilmesi istenen proje olup olmadığını sorgulayıcı doğrulamalardan geçecektir.

## **4 Tasarım Bilgi İçeriği**

### **4.1 Giriş**

Bu belgenin esas gayesi,, projenin mimari yapısı hakkında bilgi sunmaktır. Bu bölümde ele alınacak alt başlıklar şu şekilde sıralanabilir:

- YTB Tanımlaması
- Tanımlı Tasarım Paydaşları

- Tanımlı Tasarım Beklentileri
- Seçili Tasarım Bakış Açıları (herbiri tasarım unsurları ve tasarım dilleri tanımlarıyla birlikte)
- Tasarım Niyetleri
- Tasarım Katmanları
- Tasarım Gerekçeleri

## 4.2 YTB Tanımlaması

Bu belge yazılım tasarım tanımlamalarının başlangıç sürümü olup, ilerleyen süreçlerde değişmesi ihtimal dahilindedir. Ayrıca bu YTB IEEE 1016-2009 standartlarına göre hazırlanmış olup, diyagramların çiziminde Draw.io yazılımından yararlanılmıştır. Proje organizasyonu ve raporun teslim ediliş tarihi kapak sayfasında belirtilmiştir.

Belgenin birinci bölümünde YTB'ye genel bakış sunulmuştur. Ardından belgenin kapsamı, yazılış amacı ve hedef kitlesi tanımlanmıştır. Kavramsal model ve tasarım tanımlamaları 3. Bölümde ele alınmış olup son olarak 5. Bölümde bağlam bakış açısı, yapısal bakış açısı, mantıksal bakış açısı, arayüz bakış açısı gibi tasarım bakış açılarından bahsedilmiştir.

### 4.2.1 Tasarım Paydaşları ve Beklentileri

Tasarım paydaşları, bu projenin geliştiricileridir. Bizim ana beklentilerimiz şu şekildedir:

- Yazılım gerçekleştirmesi güvenlik, sürdürülebilirlik ve tekrar kullanılabilirlik gibi önemli gereksinimleri karşılamalıdır.
- Arayüzün kullanımı kolay olmalı ve gereksiz karmaşıklıklardan uzak durulmalı.
- İstenen sonuçlar, geliştirilmiş sistem tarafından elde edilebilmelidir.

### 4.2.2 Tasarım Beklentileri

Proje, modüler bir yaklaşım anlayışıyla gerçekleştirilecektir. Bu yaklaşım, değişen ihtiyaçlar karşısında, sadece değişmesi gereken modüllerin değiştirilmesini gerektirmesi nedeniyle daha esnek bir yapı sağlar. Ayrıca hata ayıklama işlemleri de modüller sayesinde daha başedilebilir hale gelecektir. Nesne yönelimli tasarım prensiplerinin uygulanması suretiyle de yeni özelliklerin eklenmesi veya var olan özelliklerin güncellenmesi az bir eforla mümkün olabilecektir. Kullanıcılar Android telefonlarında programı çalıştırarak programla etkileşime geçeceklerdir. Kazandıkları skorlar veritabanında saklanacaktır.

### 4.2.3 Tasarım Bakış Açıları

Bu bölüm, 5. Bölümde belirtilen tasarım bakış açıları hakkında genel bilgi vermektedir.

- Bağlam bakış açısı kullanıcı ve sistem arasındaki etkileşimi ifade eder. Her bir fonksiyon için kullanım durumu diyagramı kullanılır.
- Yapısal bakış açısı uygulamanın ana yapısını tarif eder. Sistemin kendi içindeki bileşenleri arasındaki etkileşimle ilgilenir.
- Etkileşimsel bakış açısı varlıklar arasındaki etkileşim ve ilişkileri tarif eder.

### 4.2.4 Tasarım Unsurları

Tasarım unsurları varlıklar, özellikler ve bazı diğer modül-kullanıcı haberleşmesinden ve ilişkilerinden müteşekkildir. Daha ayrıntılı bilgi 5. Bölümde sunulmuştur.

### 4.2.5 Tasarım Katmanları

İlgili bilgiler Tasarım Bakış Açıları bölümünde izah edilmiştir. Bu bölüm için ilave bilgi gerekmemektedir.

### 4.2.6 Tasarım Gerekçeleri

Bu proje, Android platformu üzerinde çalışacak bir oyun projesidir. Projenin sürdürülebilirliği ve geliştiriminin kolaylığı için nesne yönelimli yaklaşım benimsenmiştir. Nesne yönelimli tasarım yazılım nesnelerini, eklenip silinebilen bileşenler şeklinde ele almayı mümkün kılar. İlaveten, yazılım içinde kullanılacak değişken ve metod isimleri için, neyi ifade ettiğini veya ne yaptığını açıklayıcı isimlendirmeler tercih edilecektir. Bu da sürdürülebilirliğe katkı sağlayacaktır.

Ayrıca popüler Java oyun geliştirme kütüphanesi LibGDX kullanılacaktır.

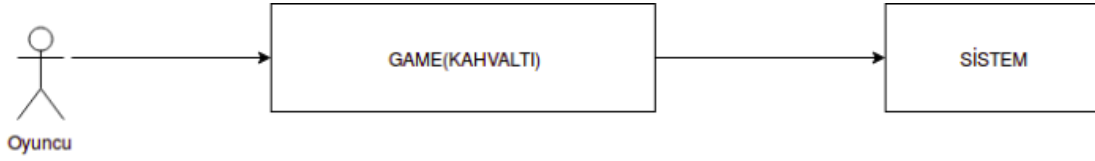
### 4.2.7 Tasarım Dilleri

Bu belgenin tasarımında UML diyagramlarından ve akış şemalarından yararlanılmıştır.

## 5 Tasarım Bakış Açıları

### 5.1 Giriş

Tasarım bakış açıları, mimari modeller, notasyonlar ve çeşitli diller yardımıyla bir sistemi nitelendirmeye yarar. Bu tasarım bakış açıları sayesinde, burada Kahvaltı oyu-



Şekil 1: Sistem bağlam diyagramı.

nunun tasarım yapıları ve kısıtlamaları ele alınacaktır. Tasarım gösteriminde UML'den faydalanılacaktır.

## 5.2 Bağlam Bakış Açısı

Kahvaltı oyunuyla kullanıcı arasındaki etkileşimler ve sistem sınırları bu bölümde incelenecektir. Kahvaltı uygulaması aracılığıyla birbirleriyle etkileşen ilgili taraflar UML ile kullanım senaryoları biçiminde gösterilecektir.

### 5.2.1 Tasarım Beklentileri

Kahvaltı oyunu, oyuncuya ilk açıldığında menü ekranını sunar. Oyun içi fonksiyonlarda internet bağlantısına gerek yoktur; oyuncu, “single player” olan oyunu her zaman çevrim dışı oynayabilecektir. Bu durumlarda, uygulamanın ilgili paydaşlar yalnızca oyuncudan ibarettir.

### 5.2.2 Tasarım Unsurları

Uygulamada en önemli tasarım elemanı oyuncudur. Oyuncu oyun ile etkileşime geçer. Tüm bu etkileşimler kullanım senaryolarında belirtilmiştir. Her ne kadar uygulamanın ücretsiz olacağına şimdilik karar verilmiş olsa da, ücretli olması durumunda ya da reklam vb. gelirleri elde edecek yer uygulamayı yapanlar olarak tarafımızdır.

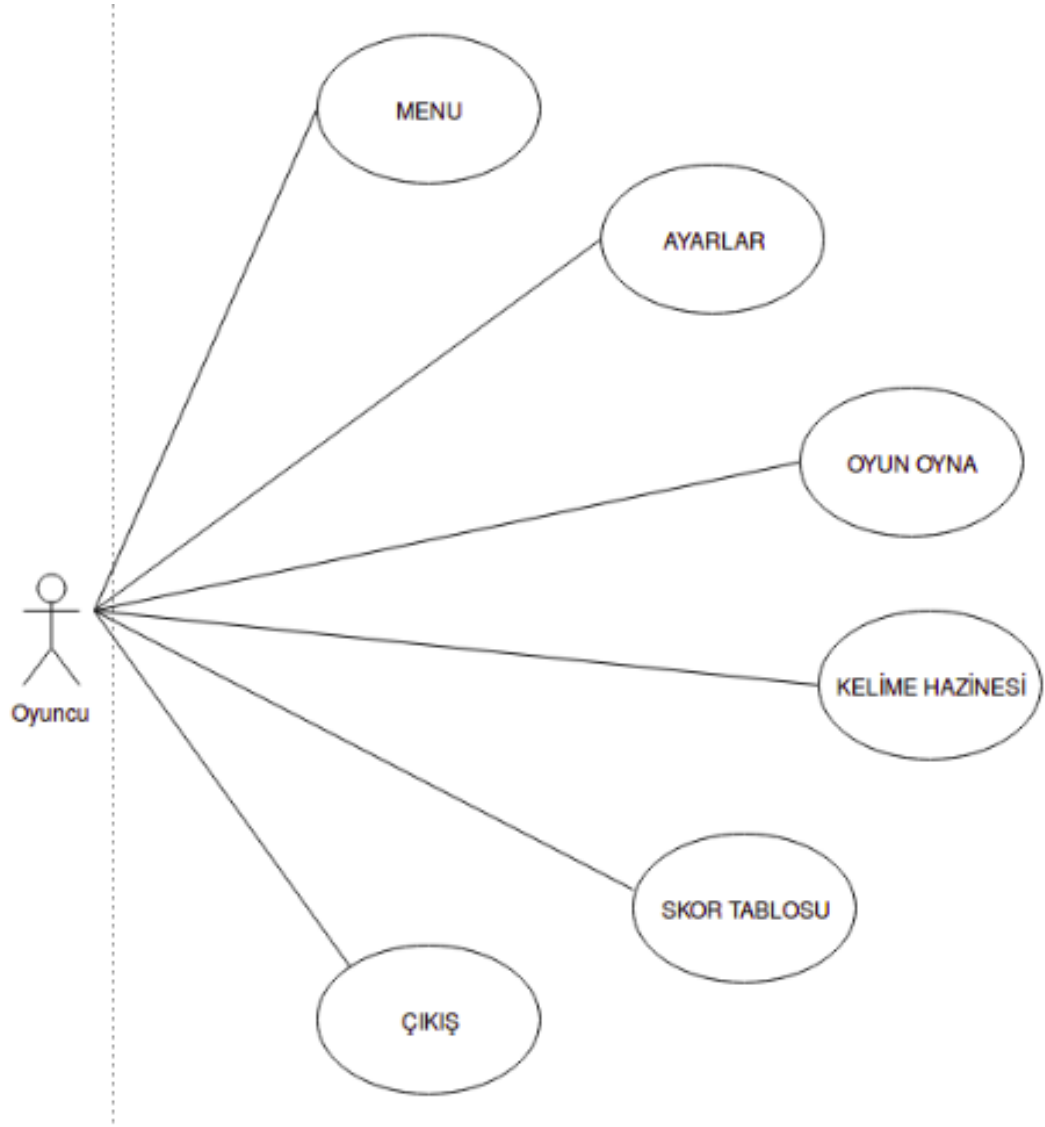
## 5.3 Yapısal Bakış Açısı

Bu bölümde, proje elemanları ve bu elemanlar arasındaki etkileşimler açıklanacaktır. Bu açıklamalar diyagram ve şekillerle detaylandırılmıştır.

### 5.3.1 Tasarım Beklentileri

Kullanıcı uygulamayı API 19 versiyonu ve üzeri Android cihazlarda çalıştırabilecektir. Uygulamanın harici bağlantıları, çevrimdışı olarak skorların tutulduğu yerel bir SQLite veritabanından ibarettir.





Şekil 2: Kullanım senaryoları diyagramı.

### 5.3.2 Tasarım Unsurları

- Ana Uygulama:
  - Fonksiyonel nitelikler: Kullanıcı ihtiyacı olan tüm fonksiyonları tek ve başka uygulama çalıştırmayan ana uygulama sayesinde gerçekleştirir.
  - Alt yordam nitelikleri: Ana uygulama veriyi yerel saklayacak olan veritabanı ile etkileşir.
- Veri Tabanı:
  - Fonksiyonel nitelikler: Yerel olarak en yüksek skorun tutulduğu SQLite veritabanıdır. Uygulama sayesinde, kullanıcının haberi ya da etkisi olmadan kullanılır.
  - İkincil fonksiyon nitelikleri: SQLite sakladığı verileri ana uygulamaya sunar.
- UI:
  - Fonksiyonel nitelikler: Kullanıcı oyuna başlamak için, yandığında tekrar başlatmak için, ana menüde seçenekler menüsüne girmek için, skorları görmek için arayüzü kullanır.

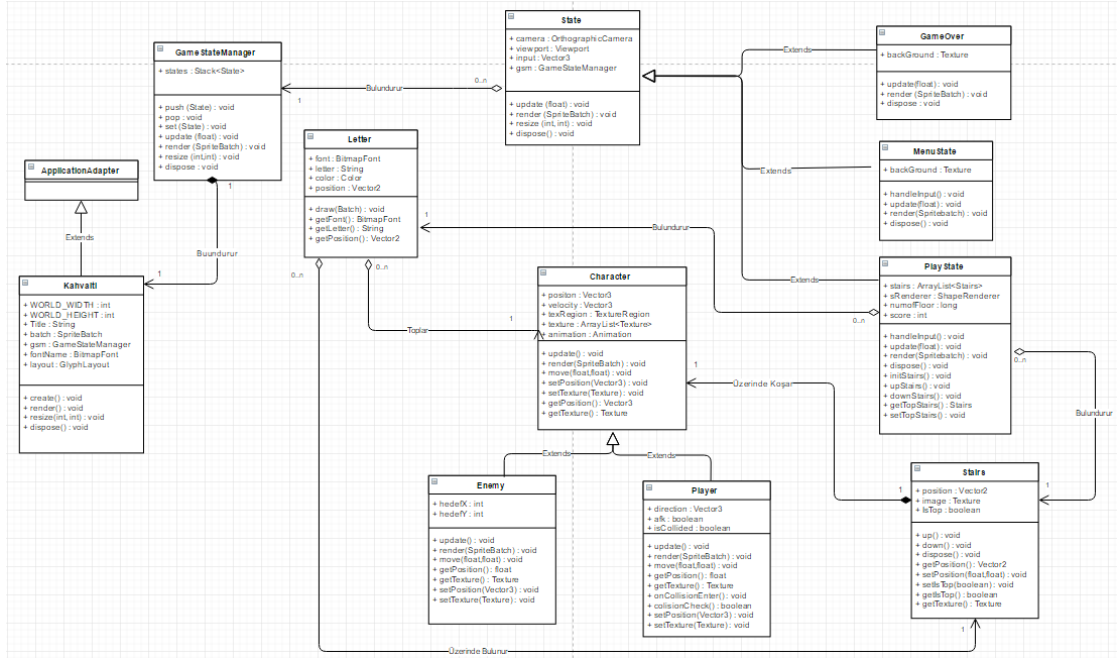
## 5.4 Mantıksal Bakış Açısı

### 5.4.1 Tasarım Beklentileri

Bu bölüm sınıfların organizasyonu hakkında detaylı bilgi içerir. Uygulama nesne yönelimli programlama metoduyla yazıldığından ve devam edeceğinden, yapımı sırasında değişiklikler gözlenebilir.

### 5.4.2 Tasarım Unsurları

- Kahvalti: Main sınıfımız olup oyunun değişmeyecek olan temel fonksiyonlarını ve niteliklerini barındırır.
- GameStateManager: Oyuncunun etkileşimine göre oyun, menü ekranları gibi GameState'leri yöneten bir sınıf. Menü ekranında, menü ekranına ilişkin şeyler render ve update edilir. Oyun ya da Game Over durumlarında ise, o sınıflara ait nitelikler idare edilir ve ekrana çizilir.
- State: PlayState, MenuState, GameOverState gibi sınıfların extend ettiği, oyunun durumlarına özel kamera pozisyonları gibi niteliklerin ayarlandığı soyut sınıftır.



Şekil 3: UML sınıf diyagramı

- **GameOver:** Oyuncu, yandığı zaman o duruma özel update ve render metotlarının uygulanacağı oyun durumudur.
- **MenuState:** Oyuncu oyunu ilk açtığında ve ya oyun sırasında menü tuşuna tıkladığında karşısına gelecek bir oyun ekranıdır. Bu ekranda da kendisine özel arkaplan, tuşlar gibi render edilecek parçalar bulunur.
- **PlayState:** Oyuncunun bizzat oyun sırasında karşısına gelen ekrandır. Karmaşık hesaplamalar, çizimler, o anki skor gösterimi, karakter animasyonları gibi görüntülerin ekrana çizildiği, merdiven hareketlerinin karmaşık bir şekilde ayarlandığı sınıftır. Oyuncunun yanması durumunda bu sınıfın üstüne otomatik olarak GameOver durumu GameStateManager sayesinde “push” edilir.
- **Stairs:** PlayState sınıfında kullanılan merdiven sınıfı.
- **Character:** Player ve Enemy sınıfları için ortak bilgileri ve metotları sağlayan bir üst sınıftır.
- **Player:** Oyun esnasında kontrol ettiğimiz karakterin pozisyon, “afk mı” gibi gerekli bilgilerini tutan sınıftır. Karakter sınıfından kalıtılır.

Skor		
PK	<u>userID</u>	int
	name	string
	score	int

Şekil 4: Skor tablosu.

Kelimeler		
PK	<u>kellme</u>	string
	puan	int

Şekil 5: Kelime tablosu.

- Enemy: Oyuncumuza engel olması amacıyla koyulmuş düşmanların kontrol edileceği sınıftır.
- Letter: Oyuncumuzun birincil amacı olan harf toplama için gerekli harf görüntülerini ve pozisyonlarını sağlayan sınıftır.

## 5.5 Bilgisel Bakış Açısı

“Entity relationship” yapısı ve dili kullanılarak, bu bölümde tasarım bilgileri ve implementasyonları sunulacaktır.

### 5.5.1 Tasarım Beklentileri

Sadece yerel bir depolama kullanıldığı ve yüksek skor tutulduğundan, basit bir tablo ile durumu izah edebiliriz:

Toplanan	
PK	<u>kelime</u>
	kelime
	score

Şekil 6: Toplanan kelimeler tablosu.

### 5.5.2 Tasarım Unsurları

Şekil 4 üzerinde gösterilen tablonun içerikleri:

- userID: Oyuncunun bir etkisi olmayan, sistem tarafından verilecek olan benzersiz bir anahtar.
- name: Oyuncunun yüksek skor yapması durumunda kendisinden istenecek olan oyuncu ismi.
- score: Oyuncunun yapmış olduğu o ana kadarki en yüksek skor.

## 5.6 Etkileşimsel Bakış Açısı

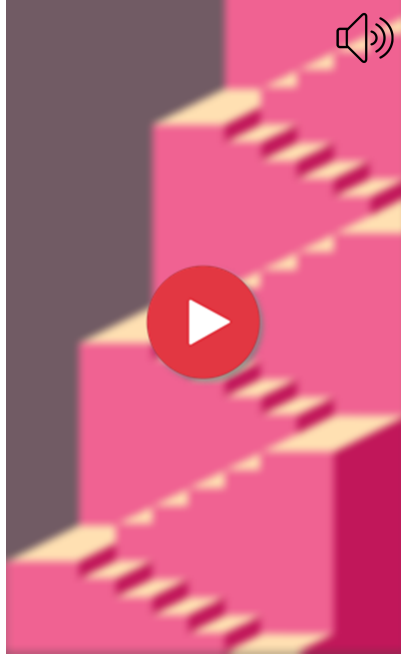
Kullanıcının uygulama ile etkileşimi ve kullanımı senaryoları, UML Durum Diyagramları ile bu bölümde irdelenecektir.

### 5.6.1 Tasarım Beklentileri

Kullanıcının uygulamayla etkileşimi doğrultusunda, uygulamanın işleyişi ve farklı etkilemlere karşı tepkileri inceleniyor.

## 5.7 Arayüz Bakış Açısı

Arayüz, kullanıcının/oyuncunun uygulamayla etkileşimini, uygulamanın farklı fonksiyonlarını kullanmasını sağlar. Bu bölümde, oyunumuzun farklı durumlardaki arayüzleri incelenecektir. Uygulama hâlâ yapım aşamasında olduğundan, burada alınan ekran görüntüleri, yapısal olarak buradakilere benzer olmakla birlikte, uygulamanın son hâliyle birlikte tasarımında ve işlevinde değişiklikler gösterebilir.



Şekil 7: Uygulama çalıştırma ekranı.

Kullanıcının oyunu ilk çalıştırması ya da “ana menü” butonuna tıklaması hâlinde karşısına gelecek, “oyna”, “ses aç/kapa” tuşunu ve hali hazırda en yüksek skoru gösterecek olan ekrandır.

Oyuncu yandığında, bölüme tekrar başlama ve ses açma/kapama seçeneklerini sunacak olan ekrandır.

## 5.8 Durum Dinamikleri Bakış Açısı

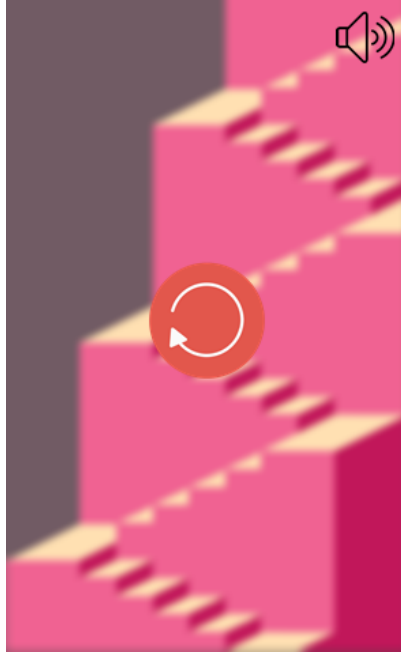
Bu bölüm, farklı durumlardaki sistem davranışlarını irdeleyecektir. Bunun gösteriminde durum diyagramlarından faydalanılacaktır.

### 5.8.1 Tasarım Beklentileri

Tasarım beklentileri Şekil 9, Şekil 10, Şekil 11, Şekil 12 ve Şekil 13 üzerinde gösterilmiştir.

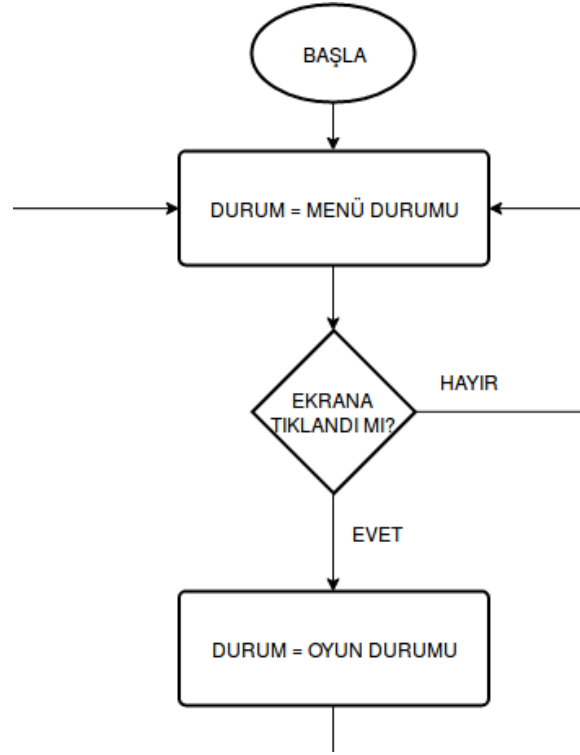
### 5.8.2 Tasarım Unsurları

Bu diyagram sistemin kullanıcının ektilerine karşı tepkilerini göstermektedir. Başlangıç durumunda, kullanıcının karşısına menü durumu gelecektir. Ekranı tıklanılana kadar

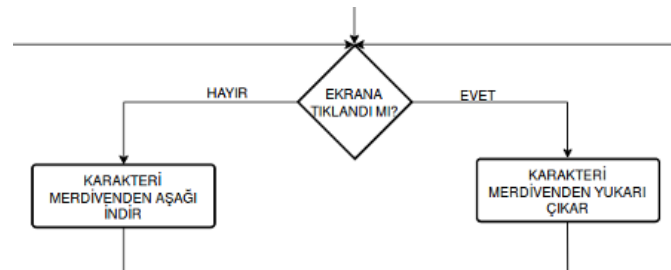


Şekil 8: Game over ekranı.

menü ekranında kalacaktır. Ekranı tıklanıldığında oyunun asıl oynandığı “PlayState” durumuna geçecektir. Burada, oyuncu harfleri toplayıp kelimeyi tamamlayana kadar kalacaktır. Eğer karakter bir düşman ile çarpışır, bir harf kaybedecek ve harfleri biterse de, oyun sona erecektir ve “game over” ekranına geçecektir. “Game over” ekranında da, yeniden oyna butonuna tıklanana kadar burada kalacaktır. Kullanıcı istediği zaman cihazın kendisinde bulunan “geri” tuşuna basarak oyundan çıkabilecektir.

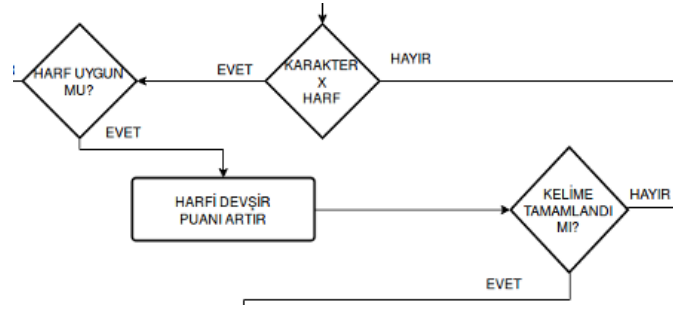


Şekil 9: Menü seçim gereksinimi akış şeması

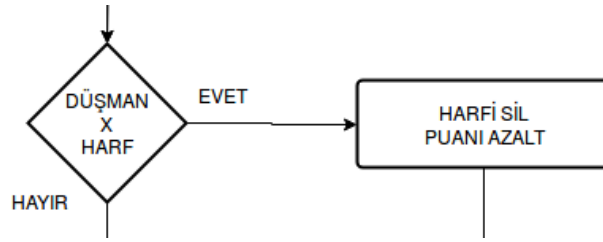


Şekil 10: Merdiven çıkma/inme diyagramı.

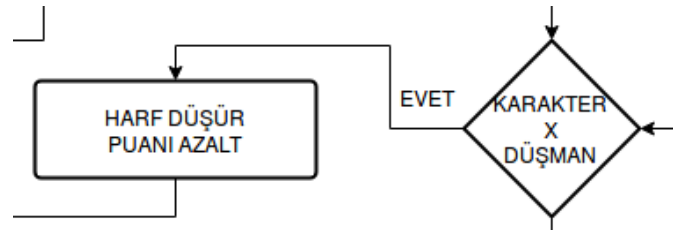




Şekil 11: Karakter-harf çarpışması diyagramı.



Şekil 12: Düşman-harf çarpışması diyagramı.



Şekil 13: Karakter-düşman çarpışması diyagramı.

## **6 Sonuç**

Kahvaltı projesinin uygulanma ayrıntıları, sistem mimarisi ve tasarım anlayışları açıklanmıştır. Gerekli tasarım bilgileri çeşitli gösterim yöntemleri kullanılarak sunulmuştur. Son olarak, tasarım bakış açıları sayesinde tasarım meseleleri ayrıntılı olarak izah edilmiştir.