



T.C
KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR/YAZILIM MÜHENDİSLİęİ

PROJE KONUSU:

Nesne Tabanlı Programlama ile Labirent Oyunu Uygulaması

ÖęRENCİ ADI:Ahmet BEKMEZCİ Ahmet ERİM

ÖęRENCİ NUMARASI: 230501022 230501027

DERS SORUMLUSU: Dr. Öğr. Üyesi Fulya AKDENİZ

TARİH:06.04.2025

1. Giriş

1.1 Projenin Amacı

Bu projenin amacı, nesne yönelimli programlama tekniklerini kullanarak kullanıcıyla etkileşime giren ve labirent üzerinde yönlendirme yapılabilen bir oyunun geliştirilmesidir. Projede Flask web çatısı ile temel bir karakter hareket sistemi tasarlanmıştır.

1.2 Projede Gerçekleştirilmesi Beklenenler

- Karakterin harita üzerinde yön tuşlarıyla hareket ettirilmesi
- Karakterin kupa simgesine ulaşarak oyunu kazanması
- OOP ilkeleriyle karakter modelleme

2. Gereksinim Analizi

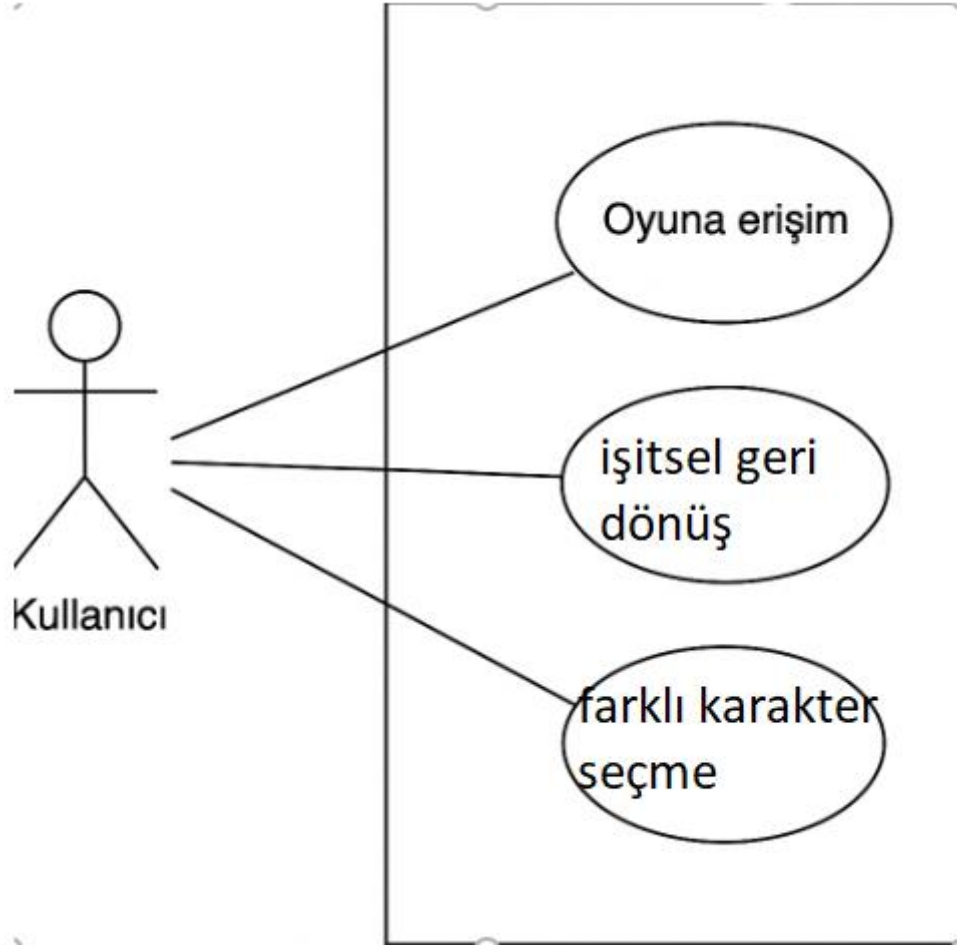
2.1 Arayüz Gereksinimleri

- Butonlar ile işlem başlatma ve sonuç ekranı gösterimi
- Farklı karakter seçim ekranı
- İşitsel geri bildirim

2.2 Fonksiyonel Gereksinimler

- Arayüz Basit ve Kullanıcı Dostu Olmalıdır

2.3 Use-Case Diyagramı



3. Tasarım

3.1 Mimari Tasarım

- Değerlendirme Fonksiyonu
- Sonuç Gösterimi
- Modül yapısı tek dosyada modüler fonksiyonlar olarak kurgulanmıştır.

3.2 Kullanılan Teknolojiler

- **Yazılım Dili:** Python
- Pygame kütüphanesi kullanıldı.

3.3 Veri Tabanı Tasarımı

Bu projede veri tabanı kullanılmamaktadır.

3.4 Kullanıcı Arayüzü Tasarımı

- Fazlalıklardan arındırılmış basit bir arayüze sahiptir.
- Göz yormayan anlaşılır renkler kullanılmaktadır.

4. Uygulama

4.1 Kodlanan Bileşenlerin Açıklamaları

Location Sınıfı

Karakterlerin koordinat bilgilerini tutmak için kullanılır:

```
class Location:
    def __init__(self, x, y):
        self.x = x
        self.y = y
```

Character Sınıfı

Tüm karakterlerin kalıtım aldığı ana sınıftır:

```
class Character:
    def __init__(self, name, tur, konum):
        self.name = name
        self.tur = tur
        self.location = konum
```

İyi Karakterler:

Luke Skywalker:

```
class LukeSkywalker(Character):
    def __init__(self, konum):
        super().__init__("Luke Skywalker", "iyi", konum)
        self.can = 3
```

Master Yoda:

```
class MasterYoda(Character):
    def __init__(self, konum):
        super().__init__("Master Yoda", "iyi", konum)
        self.can = 6
```

Kötü Karakterler:

Stormtrooper:

```
class Stormtrooper(Character):
    def __init__(self, konum):
        super().__init__("Stormtrooper", "kotu", konum)

    def enKisaYol(self, baslangic, hedef, harita):
        # BFS algoritması
```

BFS (Breadth-First Search) ile En Kısa Yol

Düşman karakterlerin oyuncuya en kısa yoldan ulaşması için kullanılır.

```
def bfs(self, baslangic, hedef, harita):
    visited = set()
    queue = [(baslangic, [baslangic])]
    # Algoritma devamlılığı
```

Harita yapısı "harita.txt" dosyasından okunur. Her karakterin konumu dosyadan belirlenir:

```
with open("harita.txt", "r") as file:
    lines = file.readlines()
    for i, line in enumerate(lines):
        for j, ch in enumerate(line.strip()):
            # konum belirleniyo
```

Tkinter ile arayüz geliştirilmiştir. Harita görsel olarak kutucuklarla gösterilir. Oyuncu ok tuşları ile hareket ettirilir.

```
self.root.bind("<Up>", self.hareketEt)
```

Oyun Kuralları ve Mantığı

1. Kullanıcı oyun başlangıcında iyi karakteri seçer.
2. Oyuncu yön tuşlarıyla hareket eder.
3. Düşmanlar her hamlede BFS ile yaklaşır.
4. Yakalanan karakter can kaybeder.
5. Can 0 olursa oyun biter. Kupa'ya ulaşırsa kazanır.

4.2 Görev Dağılımı

Tasarımdaki Görev Dağılımı:

- Kod : Ahmet Erim .
- Rapor : Ahmet Bekmezci .

4.3 Karşılaşılan Zorluklar ve Çözüm Yöntemleri

Zorluklar:

- Pygame kütüphanesi ilk kez kullanılacağı için gerekli kurulumlar ve bağımlılıkların yüklenmesinde sorunlar yaşandı.

Çözümler:

- Pygame kütüphanesi araştırılarak kullanıldı .

4.4 Proje İsterlerine Göre Eksik Yönler

- Geliştirme sırasında eksik kalan modül yoktur.

5. Test ve Doğrulama

5.1 Yazılımın Test Süreci

- Karakterler istenildiği gibi hareket tuşları ile yönlendirilebiliyor ve hareket ettirildiği zaman haritada istenilen yerlerine uygun şekilde ilerliyor.
- Düşman karakterlerde yapması istenildiği doğrultuda ilerliyor.

5.2 Yazılımın Doğrulanması

İşlevler tam olarak çalışmaktadır .