

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**PROJE KONUSU:**

**STAR WARS LABİRENT OYUNU**

**ÖĞRENCİ ADI:**

**Ataol Ali Topal**

**Yiğit Atik**

**ÖĞRENCİ NUMARASI:**

**220501029  
220501041**

**DERS SORUMLUSU:**

**Arş Gör. Eray Dursun**

**Arş Gör. Candide Öztürk**

**TARİH: 12.12.2024**

**1. GİRİŞ**

**1.1 Projenin Amacı**

Bu projenin amacı, Star Wars temalı bir labirent oyununu, Nesneye Dayalı Programlama (OOP) prensiplerine uygun olarak Python programlama dili ile geliştirmektir. Oyun, oyuncunun bir karakter seçip düşmanlardan kaçarak ödüle ulaşmasını hedefler.

**Projede Gerçekleştirilmesi Beklenenler**

* Kullanıcı karakter seçebilmeli.
* Oyuncu, harita üzerinde yön tuşlarıyla hareket edebilmeli.
* Düşman karakterler oyuncuya doğru en kısa yolu hesaplayarak hareket etmeli.
* Oyuncu ve düşmanlar çarpıştığında can kaybı olmalı.
* Oyuncu ödüle ulaştığında oyun kazanılmalı.
* Oyuncunun canı bittiğinde oyun kaybedilmeli.
* Grafik arayüz ile oyun durumu görselleştirilmeli.

**2. GEREKSİNİM ANALİZİ**

**2.1 Arayüz Gereksinimleri**

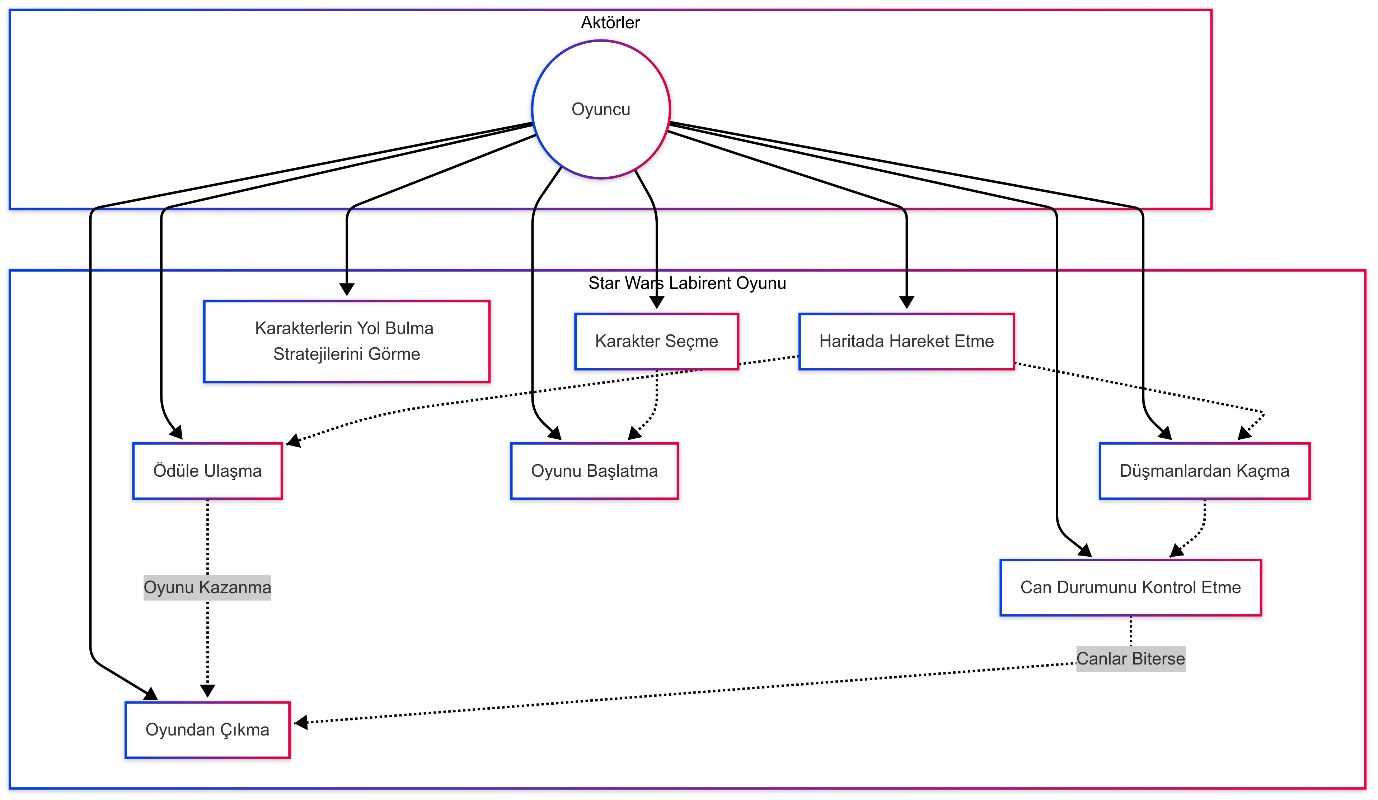
* Oyuncunun karakter seçimi yapabileceği bir başlangıç ekranı.
* Harita üzerinde oyuncu ve düşman karakterlerin görselleştirilmesi.
* Oyuncunun can durumunun ekranda gösterilmesi.
* Oyun sonu kazanma/kaybetme durumlarının gösterimi.
* Düşmanların yolunu görsel olarak gösterebilme.

**2.2 Donanım Gereksinimleri**

* Standart bir bilgisayar ve klavye.
* Python ve gerekli kütüphanelerin kurulu olduğu ortam.

**2.3 Fonksiyonel Gereksinimler**

* Haritanın dosyadan okunması.
* Oyuncunun klavye ile kontrol edilmesi.
* Düşmanların farklı algoritmalarla yol bulması:
  + Stormtrooper: BFS (Breadth-First Search)
  + Darth Vader: Dijkstra (duvarlardan geçebilir)
  + Kylo Ren: İkili hareket eden özel BFS (Breadth-First Search)
* Çarpışma kontrolü ve can yönetimi.
* Oyunun kazanma ve kaybetme kontrolü.

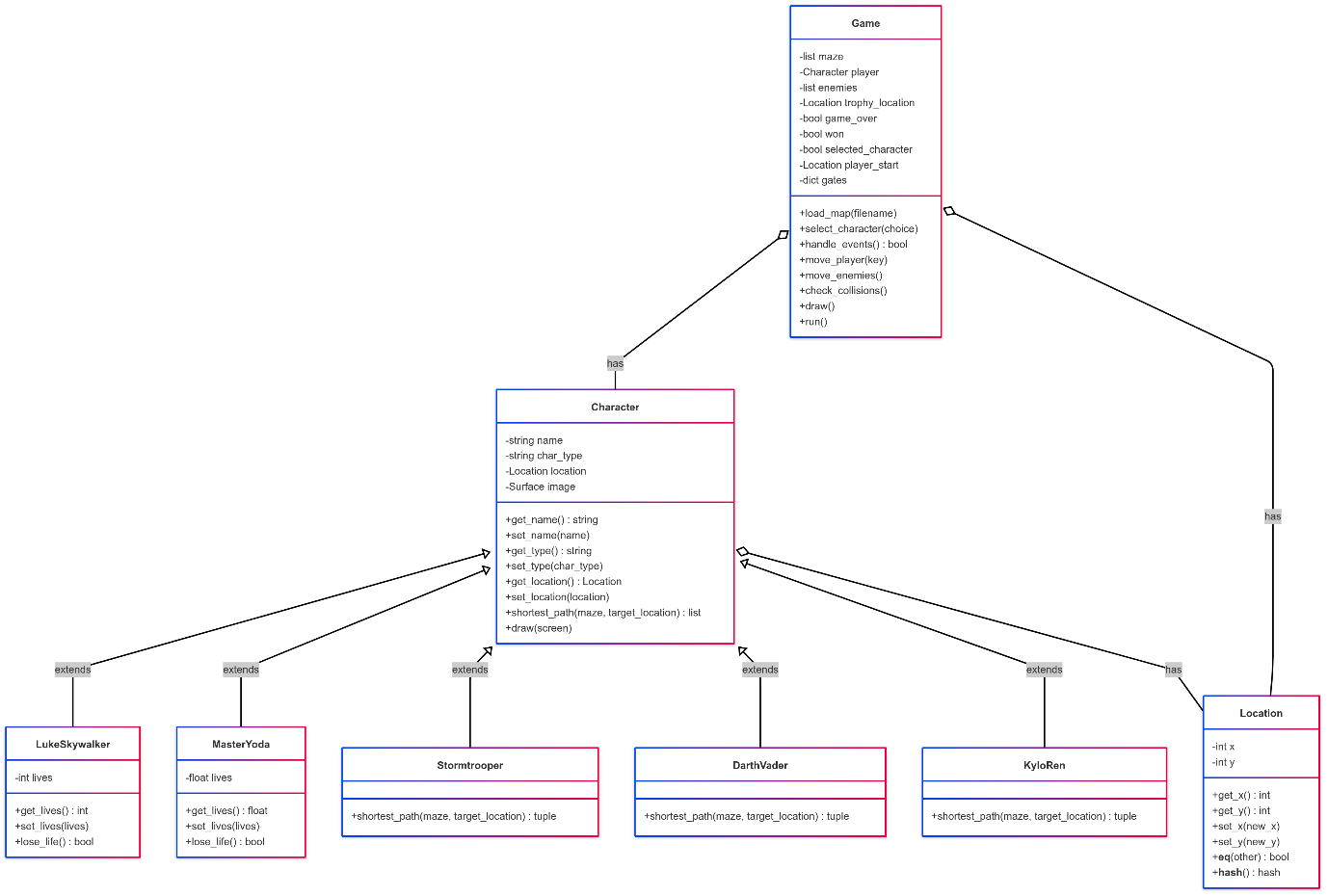
**Use-Case Diyagramı**

**3. TASARIM**

**3.1 Mimari Tasarım**

* Karakter Sınıfı (Temel karakter özellikleri)
* LukeSkywalker ve MasterYoda (İyi karakterler)
* Stormtrooper, DarthVader, KyloRen (Kötü karakterler)
* Konum Sınıfı (Koordinat yönetimi)
* Oyun Sınıfı (Oyun motoru ve mantığı)

**Modül Diyagramı**

****

**3.2 Kullanılacak Teknolojiler**

* Python 3
* Pygame kütüphanesi (Arayüz ve görselleştirme için)

**3.3 Veritabanı Tasarımı**

* Veritabanı kullanılmamıştır.

**3.4 Kullanıcı Arayüzü Tasarımı**

* Labirent görünümü
* Karakterlerin renklerle gösterimi
* Can durumunun ekranın üstünde gösterilmesi
* Başlangıç ve bitiş ekranları

**4. UYGULAMA**

**4.1 Kodlanan Bileşenler**

* Karakter: Ortak özellikler
* LukeSkywalker: 3 can, 1 can kaybı
* MasterYoda: 6 yarım can, 1 yarım can kaybı
* Stormtrooper: BFS ile yol bulur (Breadth-first search)
* DarthVader: Dijkstra ile, duvardan geçer
* KyloRen: Özel BFS ile çift adım hareket eder (Breadth-first search)
* Oyun: Harita okuma, çizim, oyun döngüsü

**4.2 Görev Dağılımı**

* Tüm tasarım ve geliştirme iki kişi tarafından yapıldı.

**4.3 Karşılaşılan Zorluklar ve Çözüm Yöntemleri**

* Yol bulma algoritmalarını karakterlere uygun uyarlamak - Farklı sınıflar ve algoritmalar yazıldı.
* Oyun içi çarpışma kontrolü - Lokasyon sınıfı ve koordinat kontrolü ile çözüldü.

**4.4 Proje İsterlerine Göre Eksik Yönler**

* Proje isterlerine uygun olarak tüm gereksinimler karşılanmıştır. Eksik yön bulunmamaktadır.

**5. TEST VE DOĞRULAMA**

**5.1 Yazılımın Test Süreci**

* Harita dosyasının doğru okunması test edildi.
* Her karakterin doğru hareket edip etmediği kontrol edildi.
* Çarpışma durumları test edildi.
* Oyun kazanma ve kaybetme koşulları test edildi.

**5.2 Yazılımın Doğrulanması**

| **Test Edilen Bileşen** | **Sonuç** |
| --- | --- |
| Harita Okuma | Başarılı |
| LukeSkywalker Hareketi | Başarılı |
| MasterYoda Hareketi | Başarılı |
| Stormtrooper Yol Bulma | Başarılı |
| Darth Vader Yol Bulma | Başarılı |
| Kylo Ren Yol Bulma | Başarılı |
| Çarpışma ve Can Azaltma | Başarılı |
| Kazanma ve Kaybetme | Başarılı |