



T.C

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

PROJE KONUSU

LORDS OF THE POLYWARPHISM

HAZIRLAYAN

DAMLA KEKLİK

230501033

DERS SORUMLUSU

Dr. Öğr. ERCAN ÖLÇER

TARİH

21/03/2024

İÇİNDEKİLER

| | |
|------------------------------------|-----------|
| 1. GİRİŞ | 3 |
| 2. GEREKSİNİM ANALİZİ | 4 |
| 3. TASARIM | 5 |
| 4. UYGULAMA | 8 |
| 5. TEST VE DOĞRULAMA | 10 |
| 6. GİTHUB | 10 |

1. GİRİŞ

Projenin amacı, bir strateji oyununda oyuncuların savaşçı seçimini ve yerleştirmesini sağlayan bir oyun arayüzü oluşturmaktır. Bu arayüz, oyuncuların oyuna başlamadan önce seçecekleri matris boyutu, oyuncu sayısı ve insan oyuncu olup olmadığı gibi seçenekler sunarak oyunun özelliklerini özelleştirmelerine imkan tanır.

Projenin gerçekleşmesinde beklenenler şunlardır:

1. Ana menü arayüzü:

- Oyunculara “PLAY”, “OPTION” ve QUIT” seçeneklerini sunan bir ana menü arayüzüyle karşılanırsınız.
- “PLAY” ve “OPTIONS” seçeneklerine tıklandığında ilgili ekranlara geçiş yapılması.
- “QUIT” seçeneğine tıklandığında oyunun kapatılması.

2. Seçenekler ekranı (OPTIONS):

- Matris boyutu, oyuncu sayısı ve insan oyuncu olup olmadığı gibi seçeneklerin bulunduğu bir ekranın oluşturulması.
- Oyuncuların seçimlerini yapabileceği butonlar veya metin giriş alanları sağlanması.
- Seçimlerin doğrulanması ve geçerli olduğunda “PLAY” seçeneğine geçiş yapılması.

3. Oyun ekranı (PLAY):

- Seçilen matris boyutuna uygun bir oyun alanının oluşturulması ve görselleştirilmesi.
- Oyuncuların savaşçıları seçip yerleştirebilecekleri bir mekanizmanın sağlanması.
- Her oyuncunun kaynaklarını ve savaşçılarının görsel olarak takip edilmesi.
- Oyuncuların seçimlerinin oyuna etkilerinin görsel olarak anlaşılabilir olması.

4. Savaşçı sınıflarının oluşturulması:

- Farklı savaşçı tiplerinin (Muhafız, Okçu, Topçu, Atlı, Sağlıklı) sınıflarının tanımlanması.
- Her savaşçı tipinin özelliklerine uygun metotların sınıflara eklendi.

5. Oyuncu seçimlerinin değerlendirilmesi:

- Oyuncuların savaşçı seçimlerinin puan/fayda oranına göre değerlendirilmesi ve en uygun savaşçının seçilmesi.
- Seçilen savaşçının oyunculara görsel olarak bildirilmesi.

Bu adımların tamamlanmasıyla birlikte kullanıcılar oyunun seçeneklerini özelleştirerek oynamaya başlayabilirler ve savaşçılarını stratejik bir şekilde yerleştirerek oyundaki hakimiyetlerini sağlamaya çalışabilir.

2. GEREKSİNİM ANALİZİ

2.1 Arayüz Gereksinimleri:

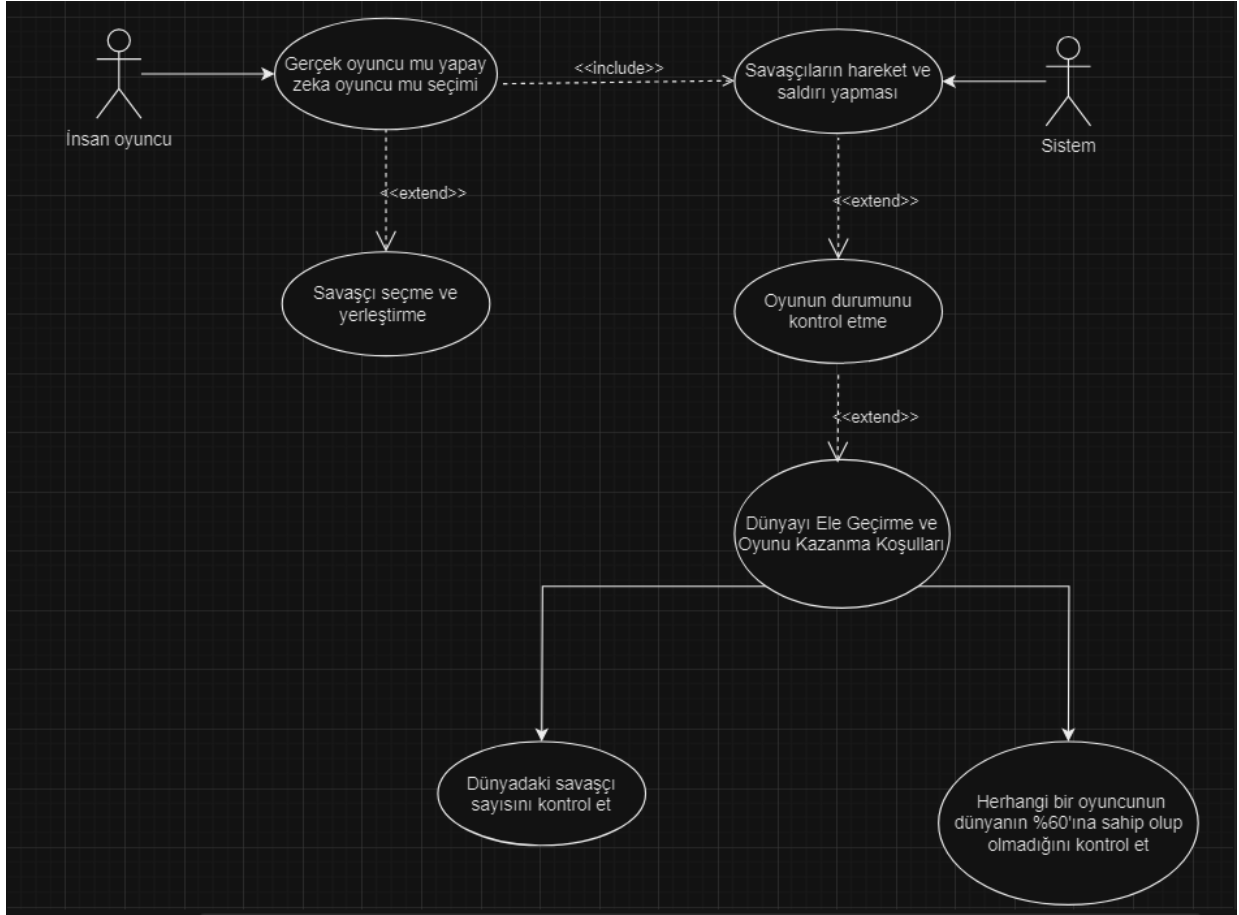
- Kullanıcı Arayüzü:
 - Ana menüde “PLAY”, “OPTIONS” ve “QUIT” seçeneklerinin görüntülenmesi.
 - Options menüsünde matrix boyutu oyuncu sayısı ve insan oyuncu seçimi için düğmelerin gösterilmesi.
 - Oyun ekranında matrisin ve oyuncuların savaşçıların görüntülenmesi.
- Donanım Arayüzü:
 - Oyunun çalışması için minimum ekrana çözünürlüğü ve fare girişi gerekmektedir.

2.2 Fonksiyonel Gereksinimleri:

- Ana Menü:
 - Kullanıcı “PLAY” seçeneğine tıkladığında oyun başlar.
 - Kullanıcı “OPTIONS” seçeneğine tıkladığında ayarlar menüsünün açılması.
 - Kullanıcı “QUIT” seçeneğine tıkladığında oyun kapatılır.
- Ayarlar Menüsü:
 - Kullanıcı matrix boyutunu, oyuncu sayısını ve insan oyuncu seçimini ayarlayabilir.
 - Kullanıcı ayarlar belirledikten sonra “BACK” seçeneğine tıklayarak ana menüye geri dönebilir.
- Oyun:
 - Oyuncu sırayla savaşçı seçimini yapar.
 - İnsan oyuncu fare ile savaşçı seçer.
 - AI oyuncular rastgele savaşçı seçimi yaparlar.
 - Oyuncular savaşçıların matrise yerleştirirler.

- Savaşçılar saldırıları gerçekleştirirler ve hasar alırlar.
- Oyun, tüm matris dolana kadar devam eder.

2.3 Use-Case Diyagramı:

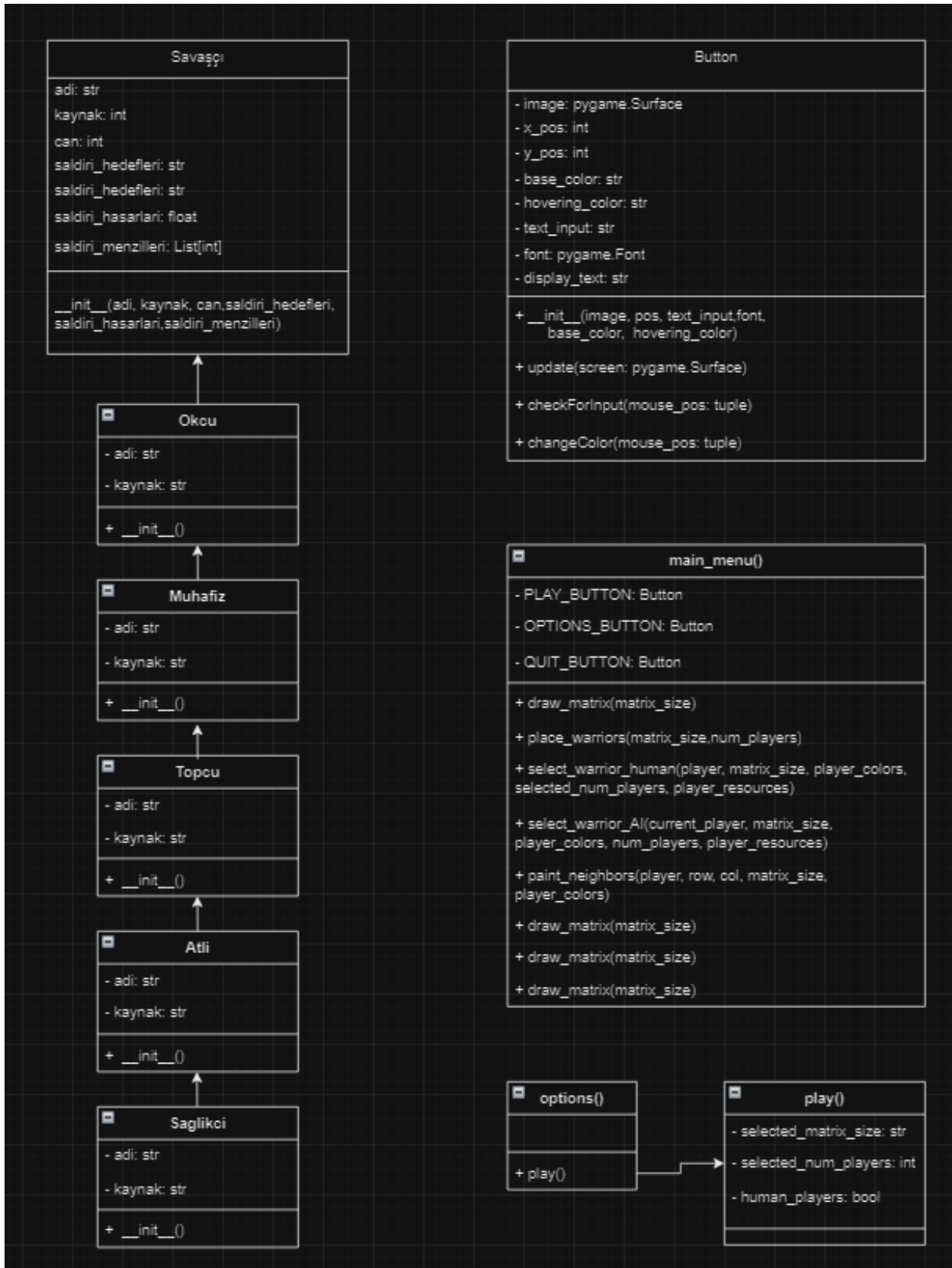


3. TASARIM

3.1 Mimari Tasarım:

- Modüler Yapı: Yazılım, modüler bir yapıda olacak şekilde tasarlanır. Bu, farklı birleşenlerin birbirinden bağımsız olmasını sağlayarak, kodun daha okunabilir sürdürülebilir ve genişletilebilir olmasını sağlar.
- Katmanlı Mimari: Yazılım, katmanlı bir mimariye dayanarak tasarlanacak. Bu, veritabanı erişim katmanı, iş mantığı katmanı ve kullanıcı arayüzü katmanı gibi farklı katmanları içerecek şekilde olacak. Bu katmanlar arasında net bir sınır çizilerek, her birinin kendi sorumluluk alanına odaklanması sağlanır.

- Modül Diyagramı:

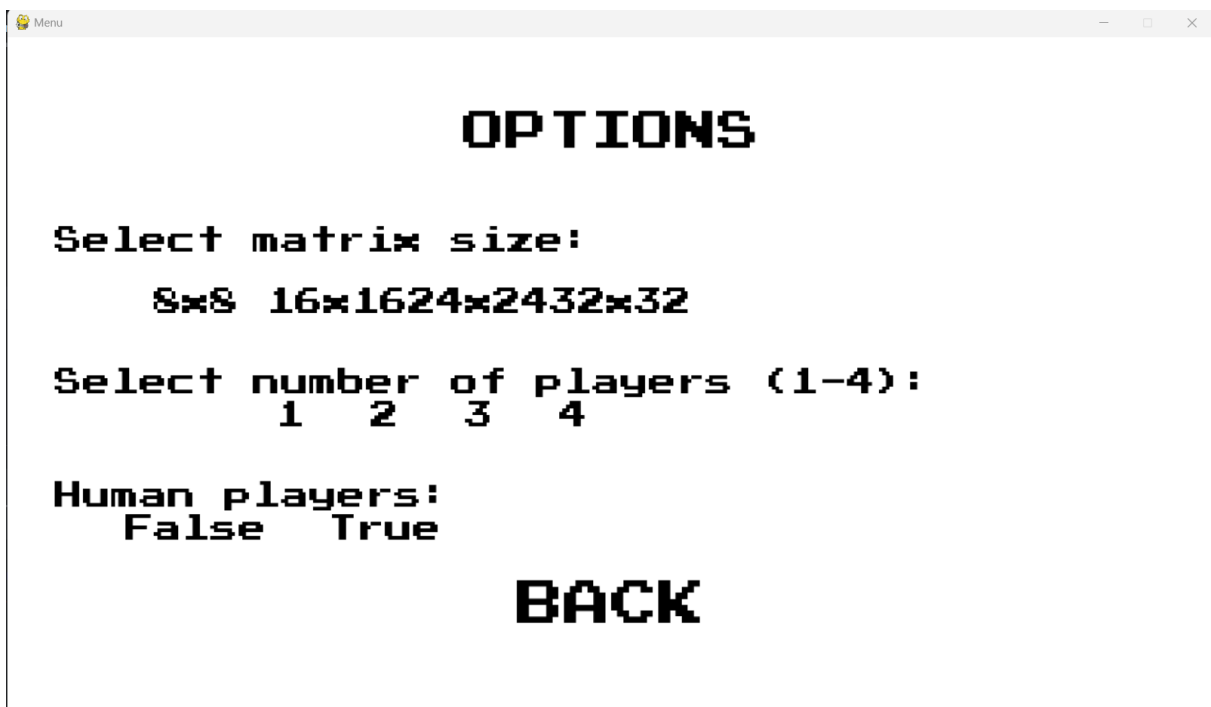


3.2 Kullanılacak Teknolojiler

- Programlama Dili: Yazılımı Python programlama dili kullanılarak geliştirildi. Pygame kütüphanesi kullanıldı.
- Harici Kütüphaneler: Pygame gibi harici kütüphane kullanıldı. Kullanıcı etkileşimi, grafikler gibi birçok oyun birleşenini destekler. Bu sebeple bu kütüphaneyi kullandım.

3.3 Kullanıcı Arayüzü Tasarımı:

Kullanıcı arayüzü basit ve kullanıcı dostu olacak şekilde tasarlandı. Ana menü, ayarlar menüsü ve oyun ekranı gibi farklı bölümler bulunuyor.



Play tıkladığımızda üstteki options menüsü açılıp kaç oyunculu, kaç kaçlık kareler üzerinde savaş yapılacağı seçilir ve bilgisayara karşımı yoksa insanlar mı oynayacak ona karar verilir. Kullanıcı oynamaya başlamadan önce ayarları yapmaları gerekmektedir.

4. UYGULAMA

1. Buton Sınıfı:

- Bir butonun özelliklerini ve davranışlarını tanımlar.
- `__init__` metodu ile butonun görüntüsü, konumu, metin içeriği, yazı tipi, temel renk ve üzerine gelindiğinde değişen renk gibi özelliklerini belirler.
- `Update` metodu, butonun ekran üzerinde güncellenmesini sağlar.
- `checkForInput` metodu, fare pozisyonunu alır ve pozisyonun butonun içinde olup olmadığını kontrol eder.
- `changeColor` metodu, fare pozisyonunu alır ve pozisyonun butonun üzerinde olup olmadığını kontrol ederek butonun metninin rengini değiştirir.

2. Options Fonksiyonu:

- Oyunun ayarlarını değiştirmek için kullanılan menüyü gösterir.
- Oyuncunun matris boyutunu, oyuncu ayısını ve insan oyuncu olup olmadığını seçmesini sağlar.
- Seçimler yapıldıktan sonra play fonksiyonuna geçiş yapar.

3. Play Fonksiyonu:

- Oyunun ana oynanabilir alanını oluşturur.
- Matris boyutuna ve oyuncu sayısına göre savaşçıları yerleştirir.
- Sıradaki oyuncunun savaşçı seçimini yapmasını sağlar.

4. Main_menu Fonksiyonu:

- Ana menü gösterir ve kullanıcının seçim yapmasını bekler.
- Play, options ve quit seçeneklerini içeren butonları gösterir.
- Kullanıcının seçimine göre ilgili işlevi çağırır.

5. Draw_matrix Fonksiyonu:

- Oyun alanındaki matrisi çizer.
- Matris boyutuna göre hücreleri ve arka plan çizgilerini oluşturur.

6. Place_warriors Fonksiyonu:

- Belirli bir matris boyutunda oyuncuların savaşçıları yerleştirir.
- Oyuncuların savaşçıları belirli köşelere yerleştirir.
- Her oyuncunun savaşçıları matrise yerleştirir ve ekranı günceller.

7. Select_warriors_human Fonksiyonu:

- İnsan oyuncunun sırası geldiğinde savaşçı seçimini yapmasını sağlar.
- Fare tıklaması ile seçilen hücreye savaşçı yerleştirir.
- İnsan oyuncunun seçimine göre savaşçısı matrise yerleştirir.

8. Select_warriors_AI Fonksiyonu:

- Bilgisayar oyuncunun sırası geldiğinde savaşçı seçimini yapmasını sağlar.
- Rasgele bir savaşçı seçerek matrise yerleştirir.

9. Paint_neighbors Fonksiyonu:

- Seçilen hücrenin komşularını belirler ve bu komşu hücrelerin arka planını belirli bir renkle boyar.
- Oyuncunun savaşçısını yerleştirdiği hücrenin etrafındaki hücreleri işaretler.

4.2 Görev Dağılımı

Grup arkadaşlarımla iletişime geçmeme rağmen bir dönüş yapmadı. Diğer arkadaşımız da oyunu tasarlamakta yardımcı olmadı.

4.3 Karşılaşılan Zorluklar ve Çözüm Yöntemleri

4.4 Projede İstenilene Göre Eksik Yönler:

İstenilen savaşma mantığını anlamadım. Bu sebeple savaş kısmı eksiktir. Oyunun sonlanması için bütün hücrelerin dolması lazımdır.

5. TEST VE DOĞRULAMA

5.1 Yazılımın test süreci:

- Buton sınıfının metotları ve özellikleri doğru çalışıyor mu kontrol edilmiştir.
- Seçenekler menüsünün doğru bir şekilde görüntülendiği ve kullanıcı girişlerinin doğru bir şekilde işlendiği kontrol edilmiştir.
- Oyun başlatıldığı ve savaşçıların doğru bir şekilde yerleştirdiği kontrol edilmiştir.
- Matrisin doğru bir şekilde çizildiği ve hücrelerin doğru boyutta olduğu kontrol edildi.
- Savaşçılar doğru bir şekilde matrise yerleştirildiği ve oyuncu kaynaklarının doğru bir şekilde güncellendiğine bakıldı.

5.2 Yazılımın doğrulanması:

- Button sınıfı ve metotları doğru bir şekilde çalışmaktadır.
- Options fonksiyonu, seçenekler menüsü doğru bir şekilde görüntülenmekte ve kullanıcı girişleri doğru bir şekilde işlemektedir.
- Play fonksiyonu oyunun başlamasını ve oyuncuların savaşçılarını doğru bir şekilde yerleştirilmesini sağlamaktadır.
- Draw_matrix fonksiyonu matrisin doğru şekilde çizilmesini sağlar.
- Place_warriors fonksiyonu, savaşçıların doğru bir şekilde matrise yerleştirme ve oyuncu kaynaklarını doğru şekilde günceller.

6. GİTHUB

- https://github.com/Damla-keklik/programlama_lab.git