



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

PROJE KONUSU: PYTHON STRATEJİ OYUNU

HAZIRLAYANLAR

ÖęRENCİ ADI: BAŞAR BURAK ÜNAL
ÖęRENCİ NUMARASI: 220501025
ÖęRENCİ ADI: EFE ÖZTAŞ
ÖęRENCİ NUMARASI: 220501014

DERS SORUMLUSU:
DR. ÖęR. ÜYESİ ERCAN ÖLÇER

TARİH: 24.03.2024

1 GİRİŞ

1.1 Projenin amacı

- Bu projenin temel amacı, kullanıcıların stratejik düşünme ve kaynak yönetimi becerilerini geliştirebilecekleri çok oyunculu bir strateji oyunu geliştirmektir. Oyuncular, farklı yeteneklere sahip savaşçılar arasından seçim yaparak, bu savaşçıları stratejik olarak harita üzerinde yerleştirir ve rakiplerini yenmek için kaynaklarını yönetir.
- **Projede Gerçekleştirilmesi Beklenenler:**
- - Çok oyunculu oyun desteği.
- - Farklı savaşçı türleri ve özellikleri.
- - Oyuncuların kaynaklarını yönetebilmesi.
- - Stratejik düşünme ve planlama gerektiren bir oyun mekaniği.
- - Yapay zekaya karşı oynanabilirliği.

2 GEREKSİNİM ANALİZİ

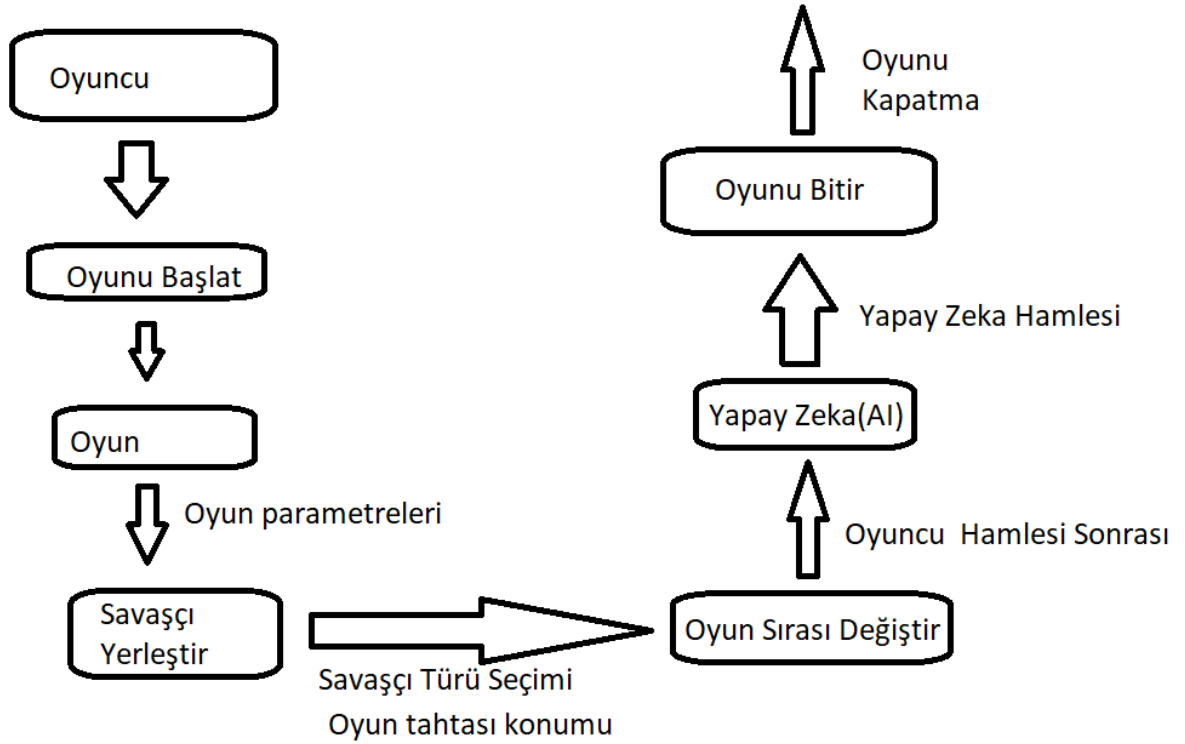
2.1 Arayüz gereksinimleri

- Oyunun durumunu, oyuncu kaynaklarını ve mevcut savaşçıları net bir şekilde gösteren kullanıcı dostu bir arayüz.
- Oyun içi eylemleri kolayca gerçekleştirebilmek için inisiyatif kontroller.
- **Donanım Arayüzü Gereksinimleri:**
- Fare ve klavye ile tam uyumluluk.

2.2 Fonksiyonel gereksinimler

- Oyuncu kaydı ve girişi.
- Savaşçı seçimi ve yerleştirme.
- Sıralı oyun ilerlemesi.
- Kaynak kontrolü.

2.3 Use-Case diyagramı



Oyunu Başlat: Kullanıcı oyuncu sayısını ve oyun alanı boyutunu belirler. Oyun bu bilgileri kullanarak oyunu başlatır.

Hücre Seç: oyun alanında bir hücre seçer.

Savaşçı Seç: Kullanıcı yerleştirmek istediği savaşçı türünü seçer.

Savaşçı Yerleştir: Oyun, seçilen hücreye seçilen savaşçı türünü yerleştirir.

Kaynakları Güncelle: Oyun, savaşçı yerleştirmenin maliyetini oyuncunun kaynaklarından düşer.

Oyuncu Değiştir: Oyuncu sırası bir sonraki oyuncuya geçer.

Oyun Alanını Göster: Oyun, oyun alanını kullanıcıya gösterir.

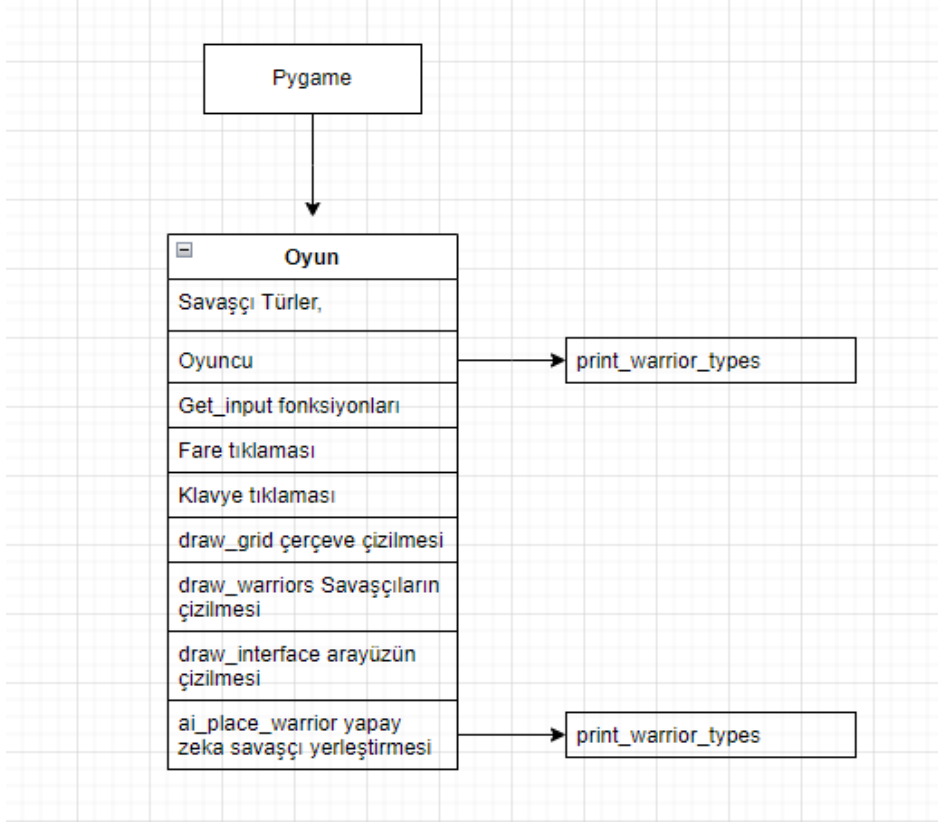
Oyuncu Bilgilerini Göster: Oyun, oyuncunun ismini ve kaynak miktarını kullanıcıya gösterir.

Oyundan Çık: Kullanıcı oyunu sonlandırır.

3 TASARIM

3.1 Mimari tasarım

- Oyun, modüler bir mimari kullanılarak tasarlanmıştır. Ana modüller; Kullanıcı Arayüzü, Oyun Mantığı ve Oyuncu Yönetimi'ni içerir.



3.2 Kullanılacak teknolojiler

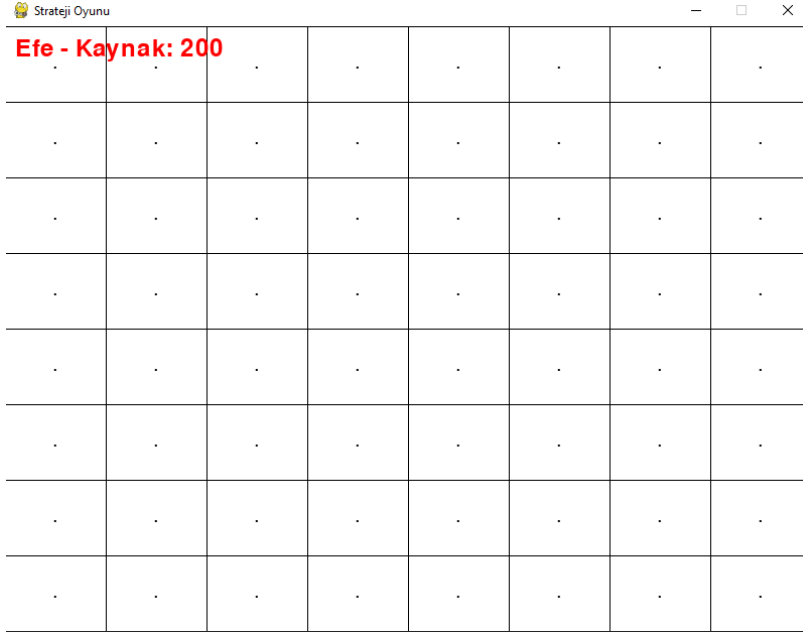
- Yazılım Dili: Python
- Kullanılan kütüphane: Pygame
- Bu teknolojiler, oyunun grafiksel arayüzünü ve etkileşimli öğelerini kolayca geliştirmek için seçilmiştir.

3.3 Veri tabanı tasarımı

- Veri tabanı tasarımı yoktur.
- Veri tabanı kullanımı olmadığı için ER diyagramı oluşturulmamıştır.

3.4 Kullanıcı arayüzü tasarımı

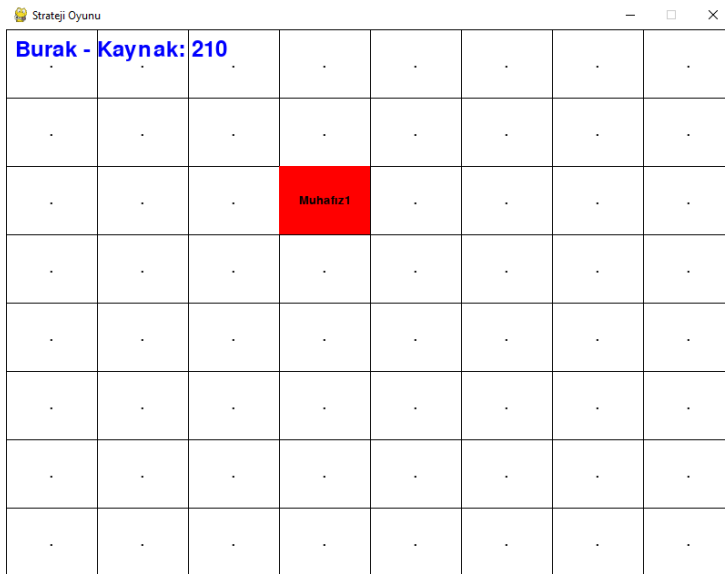
- Kullanıcı arayüzü tasarımı ile ilgili açıklama.



Efe - Kaynak: 200
.
.
.
.
.
.
.

-Uygulamayı açıp terminalde oyun boyutu ve oyuncu sayısını belirlediğinizde karşınıza böyle bir sekme çıkacak.

-Karşınıza çıkan bu karelerden herhangi birine tıklayıp, terminalden savaşçı türünü seçtikten sonra tıkladığınız kareye savaşçınız yerleşecektir.



Burak - Kaynak: 210
.
.	.	.	Muhafız1
.
.
.
.
.

-Örneğin bir oyuncu muhafızın olduğu kareyi seçip bir muhafız koydu. Ve sıra diğer oyuncuya geçti.

4 UYGULAMA

4.1 Kodlanan bileşenlerin açıklamaları

- Oyun, savaşçı yerleştirme, kaynak yönetimi ve sıra bazlı ilerleme gibi temel fonksiyonları içeren bir yapıya sahiptir.
- Oyuncu ve Oyun sınıfları, oyunun temel bileşenleridir. Oyuncular kendi sıralarında kaynak ekleyebilir, savaşçı yerleştirebilir ve oyunun durumunu kontrol edebilirler.
- Oyuncular oyunu tek veya çoklu oynayabilirler.
- Oyuncular oyunu yapay zekaya karşı da oynayabilirler.

4.2 Görev dağılımı

- Ekip olarak, oyunun her bir aşamasında birlikte ilerledik; oyun motorunun temelinden başlayarak grafik tasarımıyla birlikte görsel bir estetik oluşturduk ve son olarak yapay zeka modülüyle oyunun canlılığını ve dinamizmini sağladık.
- Raporu ekip üyeleri arasında iletişimi koparmadan birlikte hazırlanmıştır.

4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- Görselleştirme için kütüphane arayışı. Arayışların sonucunda bulunması.
- Yapay zekaya karşı oynayış. Yapay zekaya karşı oynadığımızda yapay zekanın kaynağının gözükmemesi.

4.4 Proje isterlerine göre eksik yönler

- Savaşçıların savaşmaması.
- Oyunu kazananın kaynak bittiğinde belli olması.
- Yapay zekaya karşı oynarken kaynak görülememesi zorluğu.

5 TEST VE DOĞRULAMA

5.1 Yazılımın test süreci

- Herhangi bir test uygulaması geliştirilmemiştir. Yazılan kodda testler kullanıcı girişleri ile sağlanmaktadır.

5.2 Yazılımın doğrulanması

- Test uygulaması ile yazılımın test edilmesi sonucunda elde edilen yazılım doğruluğu açıklanmalıdır.
- **Test sonucunda eksik ya da hatalı çalışan bileşenler:**
- Savaşçıların savaşmaması
- Yapay zekanın oyun sırasında kaynağı gözükmemesi.

Kaynakça

<https://www.pygame.org/>

https://www.w3schools.com/python/module_random.asp

<https://www.geeksforgeeks.org/pygame-tutorial/>

https://www.w3schools.com/python/python_polymorphism.asp

<https://www.geeksforgeeks.org/polymorphism-in-python/>