



T.C

**KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİęİ PROGRAMI**

## **ÖDEV KONUSU**

### **ÇOK OYUNCULU SAVAŞ OYUNU**

**Hazırlayanlar**

**Şevval ÖZEREN**

**220501028**

**Esmanur AYDIN**

**220501005**

**DERS SORUMLUSU**

**Dr. Öğr. Üyesi ERCAN ÖLÇER**

**24/03/2024**

<https://github.com/sevvalozrn>

[esm4ydn \(Esmanur AYDIN\) · GitHub](#)

---

## İÇİNDEKİLER

1. GİRİŞ .....	3
2. GEREKSİNİM ANALİZİ .....	3
3. TASARIM .....	6
4. SONUÇ VE ÖĞRENİLEN DERSLER.....	9
5. KAYNAKÇA .....	15

## 1. GİRİŞ

### Projenin Amacı:

Bu proje, 2-boyutlu matris formunda bir dünyada geçen çok oyunculu savaş oyunu geliştirmeyi amaçlar. Oyuncular, kendi askeri birimlerini yönetir ve düşmanlarıyla savaşırken stratejik kararlar almak zorundadır. Oyun, kullanıcıların rekabet etmesine ve iş birliği yapmasına olanak tanır, aynı zamanda eğlenceli ve zorlayıcı oyun deneyimi sunmayı hedefler.

## 2. GEREKSİNİM ANALİZİ

### Arayüz Gereksinimleri:

- Oyuncu Sıraları Göstergesi:** Kullanıcılar, hangi oyuncunun sırasının geldiğini gösteren bir göstergeye ihtiyaç duyarlar. Bu, oyuncuların hangi adımda olduklarını takip etmelerini sağlar.
- Savaşçı Seçim Arayüzü:** Sıra kendilerine geldiğinde, kullanıcıların savaşçı seçmeleri için bir arayüze ihtiyacı vardır. Mevcut savaşçı seçeneklerini net bir şekilde göstermeli ve kullanıcıların seçim yapmalarını kolaylaştırmalıdır.
- Yerleştirme Alanlarının Gösterimi:** Oyuncuların sırası geldiğinde, savaşçıların yerleştirebilecekleri uygun alanlar belirtilmelidir. Yerleştirebilecekleri alanlar oyuncunun rengiyle vurgulanmalıdır.
- Hamle Onayı:** Kullanıcıların, savaşçıları yerleştirmeden önce seçimlerini onaylamak veya değiştirmek için onay seçeneği gerekmektedir. Yanlış yerlere savaşçı yerleştirmeyi önler.
- Oyun Durum Bilgisi:** Kullanıcılar, oyunun güncel durumu hakkında bilgi alabilmelidir. Bu, kullanıcıların oyunun nasıl ilerlediğini takip edip ona göre strateji geliştirmelerini sağlar.
- Bilgilendirme ve Yardım:** Kullanıcılar, oyun hakkında bilgi almak veya yardım almak için bilgilendirme veya yardım bölümüne erişebilmelidir. Yeni oyuncuların oyunu öğrenmelerine ve mevcut oyuncuların karşılaştıkları sorunları çözmelerine yardımcı olur.

### Donanım Gereksinimleri:

Ödev No:1	Tarih 24.03.2024	3/16
-----------	------------------	------

- **Ekran Çözünürlüğü:** Oyun iki boyutlu bir dünyada geçtiği için minimum ekran çözünürlüğü **800x600** pikseldir.
- **Bellek ve İşlemci:** Oyunun düzgün çalışabilmesi için yeterli bellek ve işlemci gücüne ihtiyacı vardır. Minimum bellek gereksinimi **2GB RAM**, minimum işlemci gereksinimi ise **Intel Core i3** veya eşdeğer **AMD** işlemcidir.
- **Grafik İşlemcisi:** Minimum grafik işlemci gereksinimi **Intel HD Graphics 4000** veya eşdeğer **AMD grafik kartıdır**.
- **Giriş Aygıtı:** Oyun için gerekli olan **klavye** ve **fare** gibi giriş aygıtları mevcut olmalıdır.
- **Depolama Alanı:** Oyun dosyaları ve gereksinimleri için yeterli depolama alanı sağlanmalıdır. Minimum depolama alanı gereksinimi **500MB**'dir.

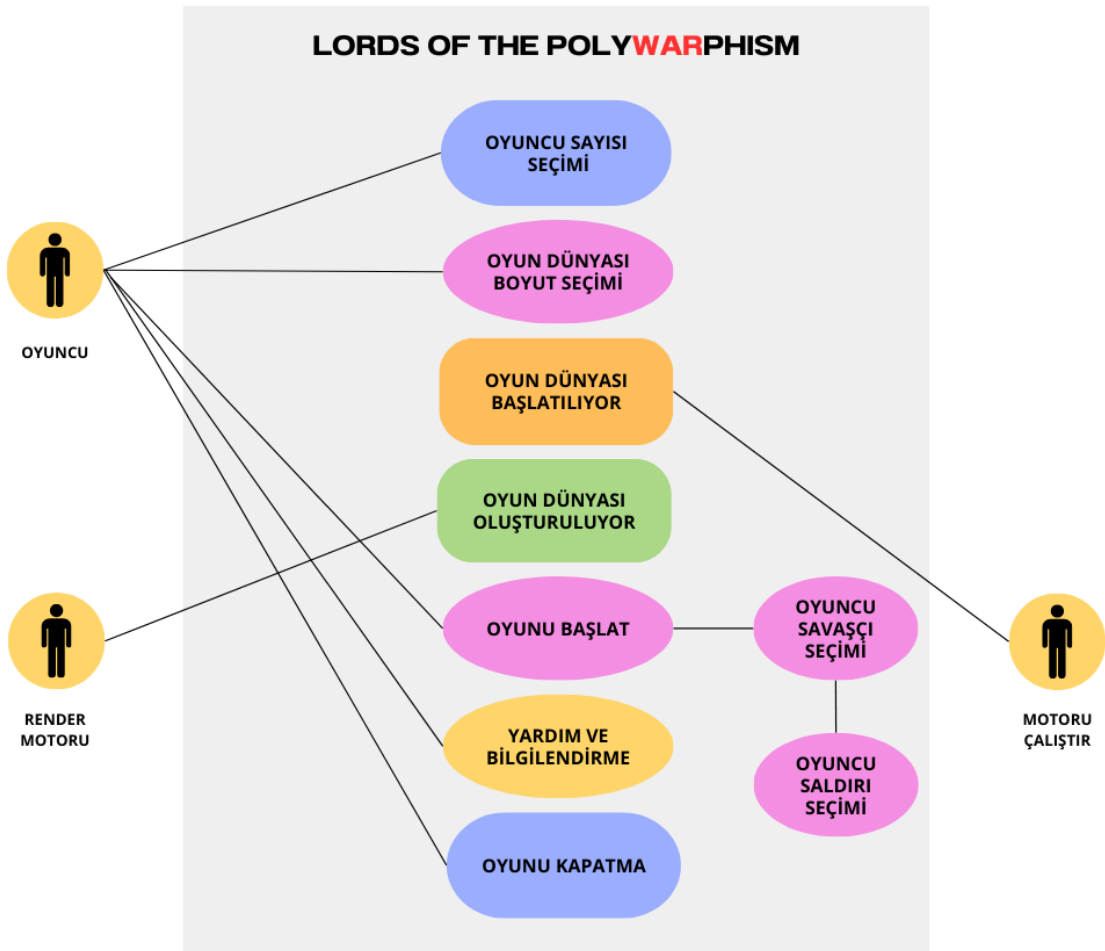
### Fonksiyonel Gereksinimler:

- **Oyun Dünyasının Oluşturulması:** Kullanıcılar oyun başladığında belirli bir boyutta oyun dünyası oluşturabilmelidir. Oyun dünyası, kullanıcıların seçtiği boyutta bir matris olarak temsil edilmelidir.
- **Oyuncu Sıralarının Belirtilmesi:** Oyun, sırası gelen oyuncuyu belirtmelidir. Sıra, oyuncular arasında dönüşümlü olarak ilerlemelidir.
- **Savaşçıların Yerleştirilmesi:** Her oyuncunun sırası geldiğinde, oyuncu seçtiği savaşçıyı uygun alana yerleştirebilmelidir. Savaşçılar, boş ve önceki savaşçıya komşu olacak alana ya da önceden yerleştirilmiş savaşçının üstüne yerleştirebilir.
- **Saldırı ve Savunma Mekanizmaları:** Oyun, savaşçıların yerleştirilmesinin ardından savaş mekanizmalarını uygulamalıdır. Savaşçılar, diğer oyuncuların savaşçılarına saldırabilir, iyileştirme ve savunma yapabilirler. Bu saldırı ve savunmaların sonunda hasar ya da can alabilirler.
- **Oyun Durumunun Güncellenmesi:** Her hamle ve saldırı sonrasında, oyun dünyasının güncel durumu oyuncuları göstermelidir. Oyuncuların son hamlelerinin etkilerini görmesini sağlar.
- **Oyunun Bitiş Koşulları:** Oyun, belirli bir bitiş koşulu sağladığında sona ermeli ve galip gelen oyuncuyu belirlemelidir. Örneğin, tüm düşmanlarının yok edilmesi durumunda oyuncuyu galip ilan edebilir.

Ödev No:1	Tarih 24.03.2024	4/16
-----------	------------------	------

- **Oyun Kontrolleri:** Oyun, oyuncuların savaşçıları seçmeleri, yerleştirmeleri ve hamle yapmaları için gerekli kontrolleri sağlamalıdır. Klavye, fare gibi giriş aygıtlarını içerir.

## Use-Case Diyagramı



### 3. TASARIM

#### Mimari Tasarım:

- **Kullanıcı Arayüzü (UI) Katmanı:** Tkinter kütüphanesi kullanılarak bir

Ödev No:1	Tarih 24.03.2024	5/16
-----------	------------------	------

---

kullanıcı arayüzü oluşturulmuştur. Oyuncuların oyunu başlatması, oyuncu sayısını seçmesi ve oyun dünyasının boyutunu belirlemesi gibi işlevler bu katmanda yer alır. Kullanıcı arayüzü, oyuncuların savaşçılarını dünyaya yerleştirmeleri için gerekli araçları sunar.

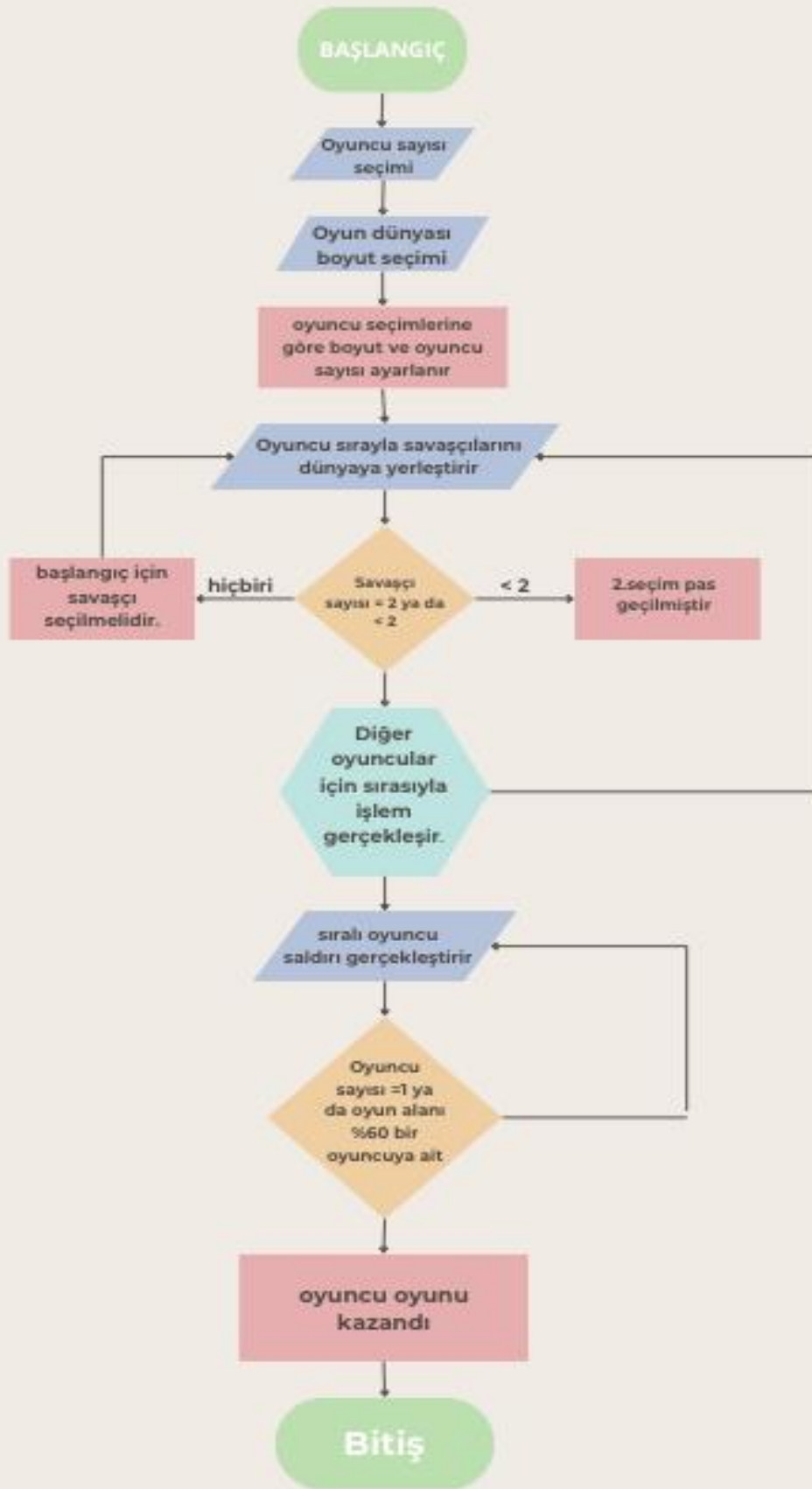
```
import tkinter as tk
from tkinter import ttk
```

- **Oyun Mantığı Katmanı:** ‘Warrior’ ve ‘Player’ sınıfları, oyun mantığını temsil eder. Savaşçıların özellikleri ve saldırı yetenekleri bu sınıflarda tanımlanır. Oyuncuların sahip olduğu kaynaklar, alanlar ve savaşçılar gibi bilgiler bu sınıflar aracılığıyla yönetilir.

```
class Warrior:
    def __init__(self, name, kaynak, can, hedef, hasar, yatay_menzil,
                 dikey_menzil, capraz_menzil):
        # Constructor
        pass
```

- **Savaşçı Seçim Ekranı:** Oyuncuların savaşçılarını seçip dünyaya yerleştirdikleri ekranı temsil eder. Savaşçı seçiminde kullanıcı arayüzü (UI) ve oyun mantığı (Warrior ve Player sınıfları) arasında iletişim sağlanır.
- **Oyun Dünyası Ekranı:** Tkinter Canvas kullanılarak oluşturulan oyun dünyasını temsil eder. Oyuncuların savaşçılarını dünyaya yerleştirmesi ve oyunun gerçek zamanlı olarak görselleştirilmesi sağlanır. Kullanıcı arayüzü (UI), oyun mantığı (Warrior ve Player sınıfları) ve görselleştirme (Canvas) arasında etkileşim gerçekleşir.
- **Yardım ve Bilgilendirme Ekranı:** Oyunculara oyun kuralları ve bilgileri hakkında yardım sağlayan bir ekranı temsil eder. Kullanıcı arayüzü (UI) ve bilgilendirme metni bu ekran aracılığıyla sunulur.
- **Oyun Motoru ve Veri Yapıları:** Oyunun temel işlevselliğini sağlayan veri yapıları (örneğin, savaşçılar, oyuncular, oyun dünyası) ve oyun motoru bu katmanda yer alır. Veri yapıları, savaşçıların özellikleri, oyuncuların durumu ve oyun dünyasının durumu gibi bilgileri depolar ve işler.

Ödev No:1	Tarih 24.03.2024	6/16
-----------	------------------	------



## Kullanılacak Teknolojiler:

- **Yazılımın Hangi Dilde Yazılacağı:** Bu yazılımın Python programlama dili kullanılarak yazılması planlanmaktadır. Python, basit ve okunabilir sözdizimiyle bilinen bir dil olduğu için bu projede tercih edilmiştir. Ayrıca Python'un geniş kütüphane desteği ve çeşitli modüllerle zengin bir ekosistemi bulunmaktadır.
- **Kullanılacak Harici Kütüphaneler:**
  - Tkinter: Python'da kullanıcı arayüzü oluşturmak için sıkça tercih edilen bir kütüphanedir. Tkinter, GUI (Grafiksel Kullanıcı Arayüzü) uygulamaları geliştirmek için standart kütüphanelerden biridir.
- **Diğer Teknolojilerle İlgili Açıklama:**
  - Bu projede Python dili dışında özel bir dış teknoloji kullanılması planlanmamaktadır. Ancak, platform bağımsız bir GUI oluşturmak için Tkinter kullanıldı.

## Kullanıcı arayüzü tasarımı:

- **Ana Pencere Tasarımı:**

Ana pencere başlığı "LORDS OF THE POLYWARPHISM" olarak belirlenir.

Pencere boyutu 450x500 piksel olarak ayarlanır ve ekranın (500, 100) konumunda açılır.

- **Oyuncu Sayısı Seçimi:**

Ana pencerede "1 player", "2 player", "3 player" ve "4 player" olarak dört buton bulunur.

Her buton, oyuncu sayısını belirlemek için kullanıcı tarafından tıklanabilir.

Kullanıcı bir butona tıkladığında, ilgili oyuncu sayısına göre oyuncu bilgileri ve oyunun boyutunu seçme seçenekleri görüntülenir.

- **Oyun Boyutu Seçimi:**

Kullanıcı oyuncu sayısını seçtikten sonra, "Dünya Boyutunu Seçiniz" başlığı altında 8x8, 16x16, 24x24 ve 32x32 butonları görüntülenir.

Kullanıcı bu butonlardan birine tıklayarak oyun dünyasının boyutunu seçebilir.

Oyun boyutunu seçtikten sonra, oyun dünyasının oluşturulması ve savaşçıların yerleştirilmesi için ekran geçiş yapılır.

- **Oyun Dünyası Tasarımı:**

Kullanıcı oyun boyutunu seçtikten sonra, seçilen boyuta göre bir oyun dünyası

Ödev No:1	Tarih 24.03.2024	8/16
-----------	------------------	------



oluřturulur.

Oyun dñnyası, kare hücrelerden oluřan bir matris řeklinde gösterilir.

Her hücrede bir nokta veya bařka bir sembol ile savařçıların yerini belirten bir iřaret bulunur.

Oyun dñnyası oluřturulduktan sonra, kullanıcı savařçıları seřip yerleřtirebilir ve oyunu bařlatabilir.

- **Yardıı ve Bilgilendirme:**

Ana pencerenin altında "Yardıı ve Bilgilendirme" adında bir buton bulunur.

Bu butona tıklandıęında, oyunun kuralları ve bilgilendirme metni iēeren bir yardıı penceresi aēılır.

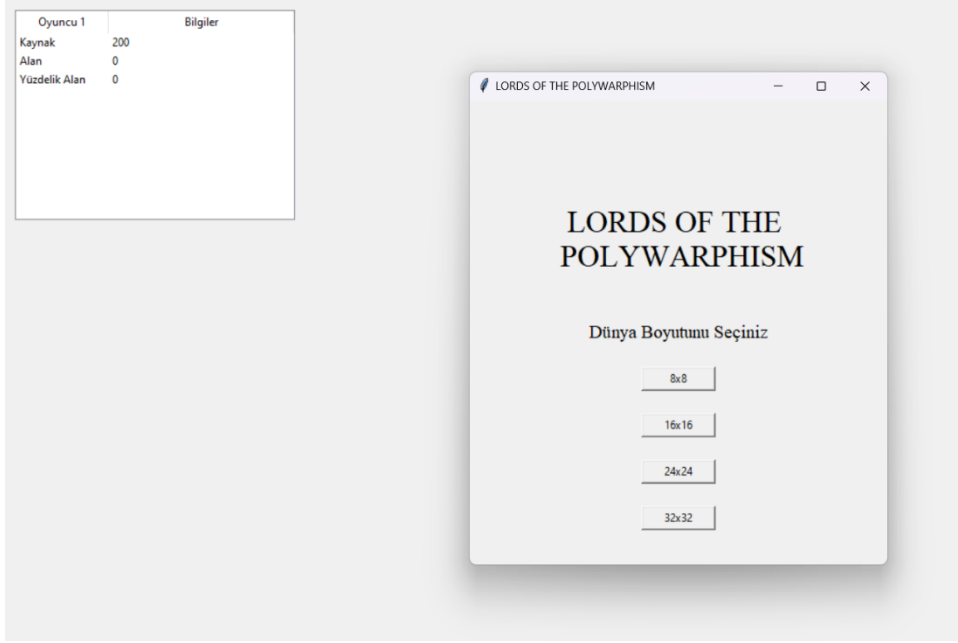
- **Yazılımdan ekran ēıktıları alınarak úzerinden aēıklama yapılması**

Oyuncu sayısı seēimi

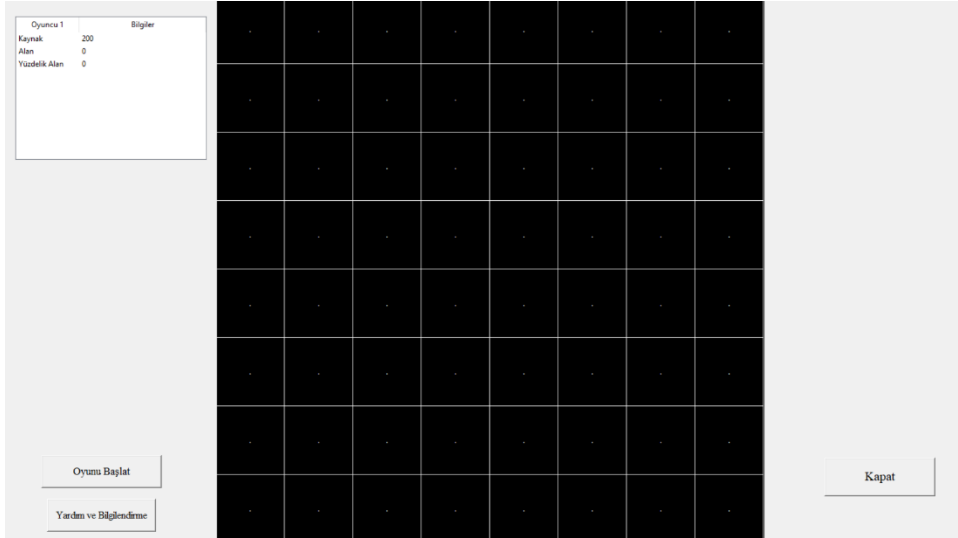


Oyuncu dñnya boyutu seēimi

Ödev No:1	Tarih 24.03.2024	9/16
-----------	------------------	------

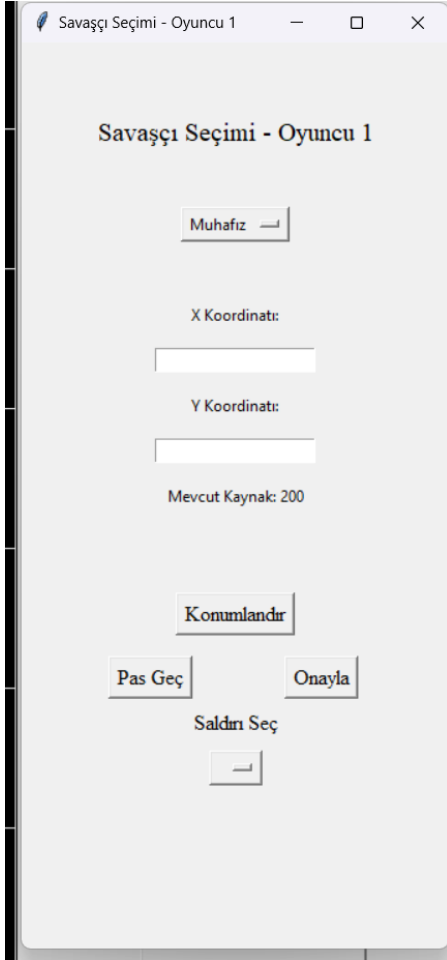
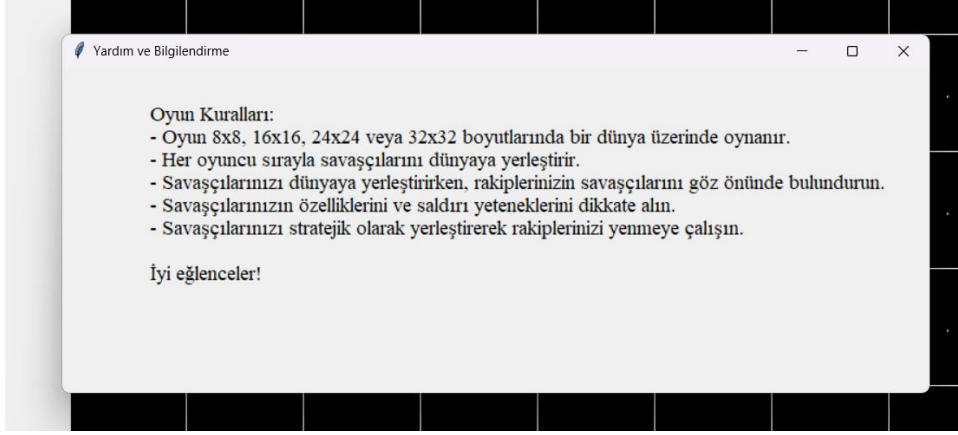


## Oyun ekranı



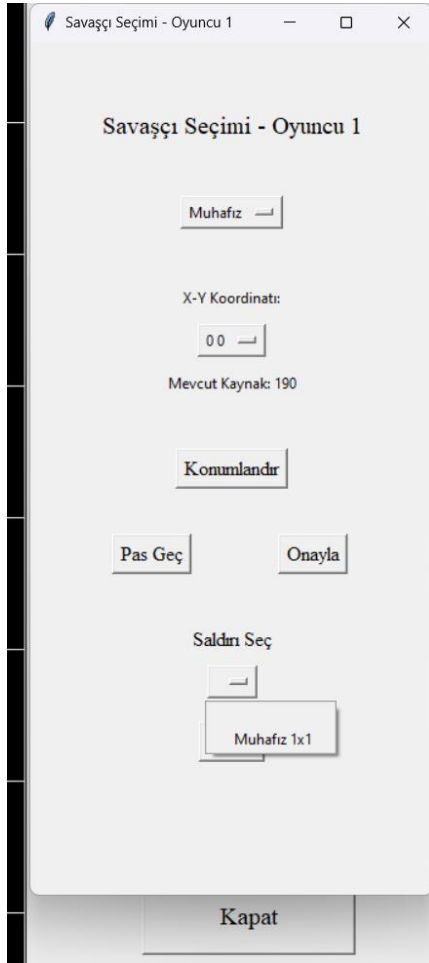
## Oyun kural bilgileri

Ödev No:1	Tarih 24.03.2024	10/16
-----------	------------------	-------



**Savařçı seçim ekranı1**

Ödev No:1	Tarih 24.03.2024	11/16
-----------	------------------	-------



**Savaşçı seçim ekranı 2**

- **Uygulamanın nasıl çalıştırılacağı ile ilgili açıklama**

Uygulamayı çalıştırmak için python üzerinden program açılmalı ve kod çalıştırılmalı. Çalıştırıldıktan sonra ise açılan pencerelerden istenilen veriler girilerek oyuna başlanabilir.

## 4. UYGULAMA

### 4.1 Kodlanan bileşenlerin açıklamaları

Oyuncu verileri tutmak ve işlemlerini gerçekleştirmek amaçlı oyuncu sınıfı, Savaşçı verilerini tutmak ve işlemlerini gerçekleştirmek amaçlı savaşçı sınıfı ayrıca oyun düzlemini seçim ekranı vb gibi oyunla ilgili işlemlerin gerçekleşmesi için ise oyun dünyası sınıfı oluşturuldu.

Ödev No:1	Tarih 24.03.2024	12/16
-----------	------------------	-------

## 4.2 Görev dağılımı

Tasarım kısmı tamamıyla tkinter kullanılarak oluşturuldu. Geliştirilme aşamasında ekstra bir kütüphane kullanılmadı.

## 4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

Oyunun kodlama sürecinde oyunun görsel biçimde ifade edilmesi gerekiyordu bunun için tkinter ile arayüz oluşturmayı tercih edildi. Oyunun kodlama süreci ilerledikçe yazılan kodlar daha karmaşık bir hal aldı bunun düzenlenmesi amacıyla da sınıflar oluşturuldu ve her sınıfın ilgili işlemleri gerçekleştireceği fonksiyonları tanımlandı.

## 4.4 Proje isterlerine göre eksik yönler

Oyunun bir sonu yok kazanan belirlenmiyor, ayrıca savaşçıların sağlık, can, kaynak vb verileri güncel tutulmuyor.

# 5. TEST VE DOĞRULAMA

## 5.1 YAZILIMIN TEST SÜRECİ

Yazılım test sürecinde sırasıyla;

- Oyun başlatıldığında dünya seçiminin ve oyuncu sayısının doğru alınıp alınmadığı
- Oyuncu sayısına göre oyuncu bilgi tablosunun düzgün oluşturulup oluşturulmadığı
- Seçilen dünya boyutuna göre, seçilmiş miktarda kareli düzlemin oluşturulup oluşturulmadığı
- Oyun seçilen kriterlere göre doğru biçimde başlatıldıktan sonra, ekrandaki oyuna başla butonuna basılmasıyla savaşçı seçim ekranı açılmalı ve oyuncular yerleştirecekleri savaşçıların koordinatlarını ve savaşçı türlerini seçebilecekler
- Seçim ekranındaki “Onayla” ve “Pas” butonları işlevlerine göre çalışmalı. “Onayla” butonuyla birlikte seçilmiş savaşçıların bilgileri oyuncu bilgilerine eklenir. “Pas” butonu ile birlikte savaşçı seçimi veya saldırı yapmaksızın seçim ekranı diğer oyuncuya geçer
- Her oyuncu ilk tur savaşçı yerleştirmesini yaptıktan sonra seçim tekrar başa döndüğünde bu sefer önceki yerleştirilen savaşçıların çevresine yerleşecek şekilde diğer savaşçılar konum seçip yerleştirilecek
- Ayrıca saldırı yapmak için de bir OptionsMenu açılacak. Bu menüde de sırası gelmiş oyuncu elindeki savaşçılardan birini saldırmak için seçebilecek ve saldırı yapabilecek
- Ayrıca savaşçıların can, kaynak bilgileri de saldırıya göre güncel tutulmalı
- Oyun sahasının %60’ına sahip olan veya sahada son kalan oyuncu oyunun kazananı olur

## 5.2 YAZILIMIN DOĞRULANMASI

- Oyuncu sayısı seçimi veya dünya boyutu seçimi projede doğru çalışmakta ve istenen sonucu vermektedir

Ödev No:1	Tarih 24.03.2024	13/16
-----------	------------------	-------

- 
- Alınan oyuncu sayısı ile oyuncu bilgi tablosu düzgün şekilde oluşturulmaktadır
  - Seçilen dünya boyutuna göre, seçilmiş miktarda kareli düzlemin oluşturulmaktadır
  - Oyun seçilen kriterlere göre doğru biçimde başlatıldıktan sonra, ekrandaki oyuna başla butonuna basılmasıyla savaşçı seçim ekranı açılmakta ve oyuncular yerleştirecekleri savaşçıların koordinatlarını ve savaşçı türlerini seçebilmektedir
  - Ayrıca oyuncu bilgilerinde de oyuncunun sahip olduğu savaşçı ve alan miktarı tutulmaktadır
  - Seçim ekranındaki “Onayla” ve “Pas” butonları işlevlerine göre çalışmaktadır.
  - Oyuncuların ilk turda savaşçı yerleşimi sorunsuz yapılmaktadır
  - Ancak saldırı aşamasında bazen istenmeyen hatalar oluşabilmektedir
  - Ayrıca oyuncunun sahip olduğu alan bilgisi tutulmakta ancak oyunun kazananı bilgisi elde edilememektedir
  - Savaşçıların saldırıya göre can eksiltme veya artırma bilgileri tutulamamaktadır

Ödev No:1	Tarih 24.03.2024	14/16
-----------	------------------	-------

---

## 6. KAYNAKÇA

- ❖ <https://chat.openai.com>
- ❖ <https://www.canva.com/design/DAGAcdmB1v4/NUeL-2TUUU3wfhq9rygEpA/edit>
- ❖ <https://www.youtube.com/watch?v=M2ZZvcgDots>
- ❖ <https://www.obenseven.com.tr/yazilim/python/nesne-tabanli-programlama/python-cok-bicimlilik-polymorphism/>
- ❖ <https://images.app.goo.gl/4ZCzCUjXLZ795qSz9>
- ❖ <https://sahinbolukbasi.medium.com/python-oop-polymorphism-2d52f8432d22>
- ❖ <https://1kodum.com/pythonda-polimorfizm/>
- ❖ <https://www.obenseven.com.tr/yazilim/python/nesne-tabanli-programlama/python-cok-bicimlilik-polymorphism/>
- ❖

Ödev No:1	Tarih 24.03.2024	15/16
-----------	------------------	-------

---

Ödev No:1	Tarih 24.03.2024	16/16
-----------	------------------	-------