



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

PROJE KONUSU:
ÇOK OYUNCULU SAVAŞ OYUNU GELİŞTİRMEK

ÖęRENCİ ADI:
ZEHRA YARDIMCI
ŞEYMA LAÇİN

ÖęRENCİ NUMARASI:
220502038
220502035

DERS SORUMLUSU:
PROF. DR. NEVCİHAN DURU

TARİH:24.03.2024

1 GİRİŞ

1.1 Projenin amacı

Bu projenin amacı, çok oyunculu bir savaş oyunu geliştirmektir.

Projede gerçekleştirilmesi beklenenler şunlardır:

- Oyuncuların birbiriyle etkileşim içinde olduğu bir oyun ortamı oluşturulması.
- 2-boyutlu bir dünyada oyuncuların savaşçıları kontrol etmesine izin veren bir oyun mekanizması tasarlanması.
- Farklı savaşçı türlerinin (örneğin, muhafız, okçu, topçu) oyuncular tarafından seçilmesi ve stratejik olarak kullanılması.
- Oyuncuların savaş sırasında hamlelerini yapmaları ve savaşçılar arasındaki etkileşimin gerçekleştirilmesi.
- Oyuncuların kaynakları yönetmesi ve savaş sırasında stratejik kararlar alması için sistem oluşturulması
- Oyuncuların sırayla hamle yapması ve savaşçılarını hareket ettirmesi.
- Savaşçıların belirlenen saldırı yöntemlerine göre düşmanlarına saldırması.
- Grafik arayüzünün kullanıcıya oyunun durumunu görsel olarak aktarması.

2 GEREKSİNİM ANALİZİ

2.1 Arayüz gereksinimleri

Kullanıcı arayüzü gereksinimleri:

- **Oyun Ekranı:** Oyun sırasında kullanıcılar için oyun ekranı sunulmalıdır. Bu ekran, oyuncuların dünya haritasını görmelerini sağlamalı ve savaşçılarını yerleştirebilmeleri için gerekli kontrolleri içermelidir.
- **Savaşçı Seçim Ekranı:** Yeni savaşçı eklenirken veya mevcut savaşçılar hareket ettirilirken, kullanıcıların savaşçı türünü seçebilecekleri bir ekran sunulmalıdır. Bu ekran, mevcut savaşçı türlerini listelemeli ve kullanıcıların seçim yapmasını sağlamalıdır.

Donanım arayüzü gereksinimleri:

- **İşlemci ve Bellek:** Oyunun akıcı bir şekilde çalışabilmesi için

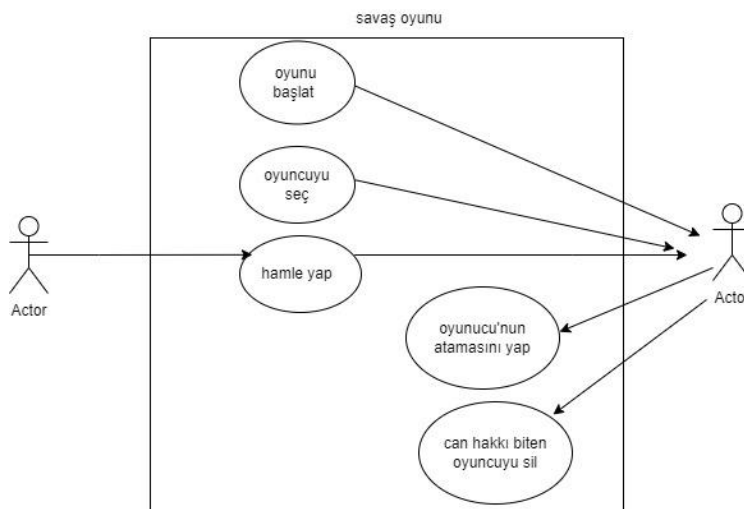
minimum işlemci ve bellek gereksinimleri belirlenmelidir.

- **Giriş Aygıtları:** Kullanıcıların oyunu oynamak için mouse, klavye gibi giriş aygıtlarını kullanabilecekleri bir donanım arayüzü sağlanmalıdır.

2.2 Fonksiyonel gereksinimler

- **Oyun Başlatma:** Kullanıcılar yeni bir oyun başlatma işlemini gerçekleştirebilmelidir.
- **Oyuncu Kontrolü:** Oyun sırasında her oyuncunun sırayla hareket etmesi ve savaşçılarını yerleştirmesi sağlanmalıdır.
- **Savaşçı Seçimi:** Kullanıcılar yeni bir savaşçı eklerken veya mevcut bir savaşçıyı hareket ettirirken, savaşçı türünü seçebilmelidir.
- **Savaşçı Hareketi:** Kullanıcılar savaşçılarını uygun koordinatlara hareket ettirebilmelidir. Hareket, belirlenen kurallara göre gerçekleşmelidir.
- **Savaşçı Saldırısı:** Kullanıcılar savaşçılarını düşman savaşçılarına saldırabilmelidir. Saldırı, savaşçı türüne ve belirlenen saldırı menziline göre gerçekleşmelidir.
- **Oyun Sonu Koşulları:** Oyunun bitiş koşulları belirlenmelidir.
- **Hata Kontrolleri:** Kullanıcıların yanlış girişler yapması durumunda hata mesajları gösterilmeli ve uygun şekilde yönlendirilmelidirler. Örneğin, geçersiz koordinatlar girme veya hatalı bir savaşçı seçimi yapma durumlarında kullanıcıya uygun bir geri bildirim sağlanmalıdır.

2.3 Use-Case diyagramı



3 TASARIM

3.1 Mimari tasarım

- **Nesneye Yönelik Programlama:** Projede nesneye yönelik

programlama prensiplerine uygun bir mimari kullanılacak. Bu sayede kod tekrarları önlenerek daha okunabilir, bakımı kolay ve genişletilebilir bir kod tabanı oluşturulacak.

- **Veri Yapıları:** Oyun dünyasını temsil etmek için uygun veri yapıları kullanılacak. Örneğin, oyun dünyasını temsil etmek için 2 boyutlu bir matris kullanılacak ve bu matris üzerinde savaşçıların yerlerini ve durumlarını izlemek için veri yapıları oluşturulacak.
- **İşlemsel Mantık:** Oyunun işlemsel mantığı, kullanıcıların sıra ile oyunu oynamasını, savaşçıların hareket etmesini ve saldırı yapmasını sağlayacak şekilde tasarlanacak. Bu işlemsel mantık, oyunun akışını doğru bir şekilde yönetecek ve oyun kurallarına uygunluğu sağlayacaktır.
- Modül diyagramı eklenmesi

3.2 Kullanılacak teknolojiler

- **Yazılım Dili:** Proje, Python programlama dili kullanılarak geliştirilecektir.
- **Harici Kütüphaneler:**
- **Pygame:** Oyunun grafiksel kullanıcı arayüzünü ve etkileşimlerini yönetmek için Pygame kütüphanesi kullanılacaktır. Pygame, Python dilinde oyun geliştirmeyi kolaylaştıran bir kütüphanedir ve grafikler, animasyonlar ve ses efektleri gibi çeşitli özellikleri destekler.
- **NumPy:** Proje boyunca 2 boyutlu matrisler üzerinde işlem yapmak için NumPy kütüphanesi kullanılacaktır. NumPy, Python dilinde yüksek performanslı matematiksel işlemleri kolaylaştıran bir kütüphanedir ve oyun dünyasını temsil etmek için kullanılabilir.
- **random:** Rastgele sayı ve ögeler oluşturmak için kullanılan bir Python kütüphanesidir. Projede, savaş oyunu senaryosuna rastgelelik eklemek ve belirli olayları rastgele gerçekleştirmek için kullanılacaktır.
- **Kullanılan Modüller:**
- **ABC (Abstract Base Classes):** Soyut sınıflar oluşturmak ve soyut yöntemler tanımlamak için kullanılan bir Python modülüdür. Projede, farklı savaşçı sınıflarının bir arayüze sahip olmasını sağlamak için soyut sınıflar kullanılacaktır.

3.3 Kullanıcı arayüzü tasarımı

- Projenin kullanıcı arayüzü, pygame kütüphanesi kullanılarak tasarlanmıştır. Pygame, oyun geliştirme için popüler bir Python kütüphanesidir ve grafiksel kullanıcı arayüzlerinin oluşturulmasını sağlar. Bu arayüz, oyun dünyasının görsel olarak temsil edilmesini, kullanıcıların oyunu görmesini ve etkileşime girmesini sağlar.

- Oyun alanı, karelerden oluşan bir ızgara şeklinde ekranda gösterilir. Her kare, bir hücreyi temsil eder ve oyundaki farklı nesnelerin (savaşçılar, boşluklar vb.) konumunu göstermek için kullanılır. Oyun dünyasının durumu, her çerçeve yenilendiğinde ekrana çizilir ve güncellenir.
- Kullanıcı arayüzü, oyun dünyasının yanı sıra kullanıcı girdilerini de işler. Kullanıcılar, savaşçıları yerleştirmek için koordinatları seçer ve bu girdiler oyunda bir savaşçının yerini belirlemek için kullanılır.

- Kullanıcıya oyun alanının boyutunu girmesi için bir talimat verilir.

```
Oyun alanı boyutunu girin (örneğin, 8x8 için 8): 8
```

- Kullanıcıya oyuncu sayısını girmesi için bir talimat verilir.

```
Oyuncu sayısını girin (minimum 1, maksimum 4): 2
```

- Girilen oyuncu sayısının doğrulandığı ve kullanıcının seçiminin onaylandığı bir mesajdır.

```
Oyuncu sayısı: 2
```

- Oyun sırasının belirlendiği bir mesajdır. Burada, 2 numaralı oyuncunun birinci sırada olduğu ve 1 numaralı oyuncunun ikinci sırada olduğu görülmektedir.

```
Oyuncu sırası: [2, 1]
```

- Sıradaki oyuncunun (2 numaralı oyuncu) savaşçıları yerleştirmek için gerekli talimatlar verilir. Kırmızı, mavi, yeşil ve sarı renkler kullanılarak savaşçıları temsil eden renkler belirtilir.

```
Oyuncu 2'in savaşçıları ((0, 0, 255)):
```

- Sıradaki oyuncunun savaşçı seçimini yapması için bir talimat verilir.

```
Savaşçı seçimi 1:
```

- Oyuncudan savaşçı türünü seçmesi istenir.

```
Savaşçı türünü seçin (Muhafız, Okçu, Topçu, Atlı, Sağlıkçı): Okçu
```

- Oyuncudan savaşçıyı yerleştirmek istediği satır numarasını girmesi istenir

```
Yerleştirmek istediğiniz satır numarasını girin: 3
```

- Oyuncudan savaşçıyı yerleştirmek istediği sütun numarasını girmesi istenir

```
Yerleştirmek istediğiniz sütun numarasını girin: 5
```

Ve bu şekilde devam eder. Yukarıdaki adımlar, kullanıcı arayüzü etkileşimlerinin konsol tabanlı bir simülasyonunu temsil eder.

```
Oyun alanı boyutunu girin (örneğin, 8x8 için 8): 8
Oyuncu sayısını girin (minimum 1, maksimum 4): 2
Oyuncu sayısı: 2
Oyuncu sırası: [2, 1]
Oyuncu 2'in savaşçıları ((0, 0, 255)):
Savaşçı seçimi 1:
Savaşçı türünü seçin (Muhafız, Okçu, Topçu, Atlı, Sağlaksız): Okçu
Yerleştirmek istediğiniz satır numarasını girin: 3
Yerleştirmek istediğiniz sütun numarasını girin: 5
Savaşçı seçimi 2:
Savaşçı türünü seçin (Muhafız, Okçu, Topçu, Atlı, Sağlaksız): topçu
Yerleştirmek istediğiniz satır numarasını girin: 2
Yerleştirmek istediğiniz sütun numarasını girin: 7
Oyuncu 1'in savaşçıları ((255, 0, 0)):
Savaşçı seçimi 1:
Savaşçı türünü seçin (Muhafız, Okçu, Topçu, Atlı, Sağlaksız): atlı
Yerleştirmek istediğiniz satır numarasını girin: 1
Yerleştirmek istediğiniz sütun numarasını girin: 6
Savaşçı seçimi 2:
Savaşçı türünü seçin (Muhafız, Okçu, Topçu, Atlı, Sağlaksız): okçu
Yerleştirmek istediğiniz satır numarasını girin: 4
Yerleştirmek istediğiniz sütun numarasını girin: 4
```

- **Uygulama nasıl çalıştırılır:**
- Oyuncuların sırası geldiğinde, konsolda "Savaşçı seçimi 1:" gibi metin tabanlı talimatlar belirir.
- Oyuncu, savaşçı türünü ("Muhafız", "Okçu", vb.) ve savaşçıyı yerleştirmek istediği satır ve sütun numarasını girmek üzere yönlendirilir.
- Girilen girdilere bağlı olarak, konsol çıktısı savaşçının yerleştirildiği konumu ve herhangi bir eylemin gerçekleştiğini bildiren metin tabanlı mesajları içerir.
- Savaş devam ettikçe, konsol çıktısı sıradaki oyuncunun hamlesini ve oyun durumunu gösterir.

4 UYGULAMA

4.1 Kodlanan bileşenlerin açıklamaları

- Bileşenlerin (fonksiyonlar, sınıflar ya da modüller) açıklanması Tabii, işte kodlanan bileşenlerin açıklamaları rapor formatında:
- **GameWorld Sınıfı**
- Oyun dünyasını temsil etmek için kullanılan sınıftır. Oyun alanını ve oyuncuların savaşçılarını yönetir.
- - **__init__(self, size):** Oyun dünyasını oluşturan yapıcı metod. Parametre olarak oyun alanının boyutunu alır ve başlangıçta tüm hücreleri '.' karakteriyle doldurur.

- - **place_warrior(self, row, col, warrior, player_color):** Savaşçıyı belirtilen konuma yerleştiren metod. Geçerli koordinatlar verilmezse hata mesajı verir.

- **Savaşçı Soyut Sınıfı**

- Oyundaki savaşçıları temsil etmek için soyut bir sınıftır. Muhafız, Okçu, Topçu, Atlı ve Sağlıkçı sınıfları bu sınıftan türetilir.

- - **__init__(self, sembol, renk):** Savaşçının sembolünü ve rengini tanımlayan yapıcı metod.

- - **get_symbol(self):** Savaşçının sembolünü döndüren soyut metoddur.

- - **get_color(self):** Savaşçının rengini döndüren soyut metoddur.

- **Muhafız, Okçu, Topçu, Atlı ve Sağlıkçı Sınıfları**

- Oyundaki farklı savaşçı türlerini temsil eden sınıflardır.

- - **__init__(self):** İlgili savaşçının sembolünü ve rengini tanımlayan yapıcı metodlar.

- - **saldırı_yap(self, game_world, row, col):** Belirli bir hedefe saldırı gerçekleştiren metodlar.

- **Diğer Yardımcı Fonksiyonlar**

- Oyun sırasını belirleme, oyuncu ve hazine oluşturma gibi yardımcı işlevleri yerine getiren fonksiyonlar.

- - **get_world_size():**Kullanıcıdan oyun alanının boyutunu alan ve bu boyutun 8x8 ile 32x32 arasında olmasını sağlayan fonksiyon.

- - **get_player_count():**Kullanıcıdan oyuncu sayısını alan ve bu sayının 1 ile 4 arasında olmasını sağlayan fonksiyon.

- - **determine_player_order(player_count):** Oyuncuların sırasını belirleyen ve bu sırayı rastgele karıştıran fonksiyon.

- **Ana Oyun Döngüsü ve Grafiksel Arayüz**

- Oyunun ana döngüsü ve grafiksel arayüzünü sağlayan kısımları içerir.

- - **Ana Döngü:** Oyunun ana döngüsünü başlatır ve pygame

kütüphanesini kullanarak oyun dünyasını ve savaşçıları ekrana çizer.

- **- Grafiksels Arayüz:** Oyunun grafiksels arayüzünü oluşturan kodlar yer alır. pygame kütüphanesi kullanılarak oyun dünyası ekran üzerine çizilir.

4.2 Görev dağılımı

- **Bileşenlerinin tasarım ve geliştirme aşamalarındaki görev Dağılımı:**
Projenin tasarım ve geliştirme aşamalarında iş bölümü yapılmamıştır. Tüm bileşenlerin tasarımı ve geliştirilmesi Zehra YARDIMCI tarafından gerçekleştirilmiştir.
- **Raporun hazırlanması sürecindeki görev dağılımı:**
Use Case Diyagramı ve Modül Diyagramı Tasarımı: Şeyma LAÇİN
Raporun Geri Kalan Kısımları: Zehra YARDIMCI

4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- Projede, savaşçılara renk atama sürecinde bir hata yapıldı. Aslında projenin gereksinimleri oyunculara renk atamamızı gerektiriyordu, ancak yanlışlıkla savaşçılara renk atandı. Bu hatayı fark ettiğimde kodun ilgili bölümlerini değiştirmeye çalıştım. Ancak, maalesef bu değişiklikleri yaparken bazı zorluklarla karşılaştım ve beklenen sonuçları elde edemedim.

4.4 Proje isterlerine göre eksik yönler

Projenin gerçekleşmesi beklenen görevlerden bazıları tam olarak kodlanamamıştır. Bunlar:

- Oyuncuların sırayla oynayabilmesi ve her elde savaşçı yerleştirme yeteneklerine yönelik kodlama eksikliği bulunmaktadır.
- Savaşçıların birbirlerine olan etkileşimleri ve savaş stratejilerine göre saldırılarını gerçekleştirecek kodlama bulunmamaktadır.
- Oyuncuların sahip oldukları kaynakların yönetimi ve her elde kazanılan kaynak miktarına ilişkin kodlama bulunmamaktadır.
- Oyuncuların oyun alanında yerleştirebilecekleri savaşçıların kısıtlamalarına ilişkin kodlama eksikliği mevcuttur.

5 TEST VE DOĞRULAMA

5.1 Yazılımın test süreci

- Yazılımın test süreci kapsamında bir test uygulaması geliştirilmedi.

5.2 Yazılımın doğrulanması

- Test Yazılımın doğrulanması için geliştirilen test uygulaması

olmadığından, yazılımın doğruluğu hakkında kesin bir değerlendirme yapılamamaktadır. Ancak, yazılımın bazı bileşenlerinin manuel olarak test edildiği ve bazı senaryolar altında doğru çalıştığı gözlemlenmiştir.

GİTHUB HESAPLARI

<https://github.com/Seyma-lacin>

<https://github.com/Zehrayardimci>