



T.C
KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ

PROJE KONUSU: Programlama Labı Ödev-1 Lords Of The Polywarphism

ÖĞRENCİ ADI: İBRAHİM TEKİN
ÖĞRENCİ NUMARASI: 220502037
ÖĞRENCİ ADI: ALPEREN ALSAN
ÖĞRENCİ NUMARASI: 220502024

DERS SORUMLUSU:
DR. ÖĞR. ÜYESİ NUR BANU ALBAYRAK
TARİH: 24.03.2024

1 GİRİŞ

1.1 Projenin amacı

- Projenin amacını 2 boyutlu matris formunda bir dünyada çok oyunculu bir savaş oyunu oluşturmaktır.

2 GEREKSİNİM ANALİZİ

2.1 Arayüz gereksinimleri

- 2-boyutlu dünyanın boyutları oyunun başında seçilecektir. Kullanıcılar 16x16, 24x24, 32x32 veya kullanıcı tanımlı bir kare dünyada oyun oynayabilecektir. Dünyanın bir boyutu 8x8'den küçük, 32x32'den büyük olamayacaktır.
- 1. Oyuncu kırmızı, 2. Oyuncu mavi, 3. Oyuncu yeşil ve 4. Oyuncu sarı renk ile gösterilecektir.
- Boş hücreler beyaz renk “.” karakteri ile savaşçılar ise ilk harfleri ile gösterilecektir. Örneğin bir muhafız büyük M harfi ile bir okçu ise büyük O harfi ile ekranda gösterilir.

2.2 Fonksiyonel gereksinimler

- Oyunda en az 1 gerçek oyuncu olmak üzere maksimum 4 oyuncu olabilecektir.
- Oyuncular bir sıraya uygun olarak oyun oynayacaktır. Sırası gelen oyuncu bir el içinde kaynakları yetiyorsa en fazla 2 savaşçı oluşturabilecektir. Savaşçılar aşağıdaki tablodaki herhangi bir savaşçı olabilir. Oyuncu isterse aynı tür savaşçıdan 2 tane üretebilir. Oyuncu isterse bir adet savaşçı üretebilir veya hiç üretmeden pas geçebilir.
- Gerçek insan oyuncular bir menüden savaşçı türünü seçtikten sonra savaşçıyı matrise yerleştirecekleri hücrenin x ve y koordinatlarını vermelidir. Bu süreç uygun seçimler yapılana kadar devam etmelidir.
- Tüm oyuncular için savaşçı seçim ve yerleştirme işlemi tamamlandığında sahadaki tüm savaşçılar için sahaya ilk yerleşen savaşçı en önce olmak üzere saldırılarını gerçekleştirirler.

3 TASARIM

3.1 Yazılım Tasarımı

- Bu projede nesneye yönelik programlama tasarımı kullanılmıştır

3.2 Kullanılacak teknolojiler

- Projemiz Python dilinde kullanılmıştır
- Projemizde kullandığımız harici kütüphaneler 'pygame' ve 'random' dur.

3.3 Kullanıcı arayüzü tasarımı

- Arayüz bakımından user friendly bir yaklaşım sergileyip oldukça sade bir arayüz tercih ettik.
- Aşağıda kodda kullandığımız bazı fonksiyonlar bulunmaktadır.

```
2 usages
def initiate_combat(attacker, defender):

    random_damage = random.randint(1, 10)

    if (attacker.menzil_x >= abs(attacker.pozisyon_x - defender.pozisyon_x) and
        attacker.range_y >= abs(attacker.pozisyon_y - defender.pozisyon_y) and
        attacker.health > 0 and defender.health > 0):

        damage = attacker.damage_modifier + random_damage

        defender.health -= damage

        print(f"{attacker.name} attacked {defender.name} and dealt {damage} damage.")

        if defender.health <= 0:
            defender.health = 0
            print(f"{defender.name} has been defeated!")
        else:
            print(f"{attacker.name} cannot attack {defender.name}. Out of range or one of them is defeated.")

pygame.init()
```

```
def place_random_warrior(player, world):
    available_cells = []

    for y in range(world.size):
        for x in range(world.size):
            if can_place_warrior(x, y, player.name, world):
                available_cells.append((x, y))

    if available_cells:
        x, y = random.choice(available_cells)

        warrior_type = random.choice(list(savasci_simgeleri.keys()))

        if warrior_type and player.spend_resources(WARRIOR_COSTS[warrior_type]):
            player.savascilar.append(warrior_type)
            world.grid[y][x] = savasci_simgeleri[warrior_type]
            print(f"{player.name} pastgele {warrior_type} savascısını ({x}, {y}) koordinatına yerleştirdi.")
```

4 UYGULAMA

4.1 Kodlanan bileşenlerin açıklamaları

Aşağıda gördüğünüz kodda sıra geçme butonu kodlanmıştır

```
1 usage
def create_pass_button():
    button_width = 150
    button_height = 50
    x = SCREEN_WIDTH + 10
    y = SCREEN_HEIGHT - button_height - 400
    button_rect = pygame.Rect(x, y, button_width, button_height)
    button_color = (200, 200, 200)
    return button_rect, button_color
```

Aşağıdaki kodda savaşçı oluşturma kısmı kodlanmıştır

```
1 usage
def create_warrior_buttons():
    button_width = 150
    button_height = 50
    margin = 10
    x = SCREEN_WIDTH + 10
    y = SCREEN_HEIGHT - button_height - margin
    buttons = []

    savasci_listesini_ters_dondur = list(savasci_simgeleri.keys()[::-1])

    for i, warrior_type in enumerate(savasci_listesini_ters_dondur):
        icon = savasci_simgeleri[warrior_type]
        button_rect = pygame.Rect(x, y - i * (button_height + margin), button_width, button_height)
        button_color = (150, 150, 150)
        buttons.append((button_rect, warrior_type, icon, button_color))

    return buttons
```

4.2 Görev dağılımı

- Sorumlulukları aramızda eşit şekilde paylaşıp zorlukların üstesinden beraber geldik

4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- Çok fazla fonksiyon kullanıldığı için bu fonksiyonların beraber sorumsuz çalışması konusunda biraz zorlandık

4.4 Proje isterlerine göre eksik yönler

- Yapay zeka modülü ve savaşma mekaniğini oyuna entegre edemedik.

5 TEST VE DOĞRULAMA

5.1 Yazılımın test süreci

- Yazılımın Testini kendimiz yaptık

5.2 Yazılımın doğrulanması

- Yazılımın doğrulanmasını kendimiz yaptık.

<https://github.com/Alpyinwonderlands>
<https://github.com/IbrahimTekin03>