

OYUN PROJESİ ANALİZİ

Dalga Tabanlı Savunma Oyunu: Teknik Mekanik İncelemesi

"SYSTEM INITIALIZED"

"ARCADE TERMINAL v1.0"

"READY FOR ANALYSIS"

PROJE ÖZETİ: TEMEL MEKANİKLER

Oyuncu (Sarı Küp), Base'i (Mavi Küp) dalgalar halinde gelen düşmanlara (Beyaz Küreler) karşı korur. Zorluk, dalga ilerledikçe dinamik olarak artırılır.



Wave Sistemi

Her 1 dakikada dalga atlanır.
Düşman sayısı ve hızı artar.



Oyuncu Kontrolü

W/A/S/D ile hareket, fare ile dönüş.
Space ile ateş, LeftShift ile hızlı mod.



Puan Sistemi

Her düşman 10 puan değerinde. Mermiler çarptığında puan eklenir.



TEMEL OYNANIŞ: ATEŞ VE TARAMALI MOD

Mermi Sistemi

Space tuşu ile mermi atılır. Mermi hızı 10 birim/saniye, yaşam süresi 5 saniyedir.

```
public float speed = 10f; public float lifeTime =  
5f; transform.Translate(Vector3.forward * speed *  
Time.deltaTime);
```

Çarpışma algılandığında (Enemy tag'i), 10 puan eklenir ve hem mermi hem düşman yok edilir.

Taramalı Mod (Burst Fire)

→ Etkinleştirme

LeftShift tuşu ile etkinleştirilir.

→ Süre ve Bekleme

5 saniye boyunca hızlı ateş, ardından 10 saniye bekleme süresi (cooldown).

→ Avantaj

Stratejik avantaj sağlayan güçlü bir mekaniktir.

Enemy Spawner (Mono Script) Import Settings

Open Execution Order...

Enemy Prefab: Enemy
Target: None (Transform)

Default references will only be applied in edit mode.

Imported Object: Enemy Spawner (Mono Script)

Assembly Information: Assembly-CSharp.dll

```
using System.Collections;
using UnityEngine;

public class EnemySpawner : MonoBehaviour
{
    [Header("Ayarlar")]
    public GameObject enemyPrefab; // Spawning enemy prefab
    public Transform target; // Target object
    public float spawnDelay = 0.5f; // Time between spawns
    public float enemySpeed = 3f; // Enemy speed
    public string spawnerTag = "Spawner"; // Spawner objects' tag

    private Transform[] spawners;
    private int currentWave = 1;

    // UI'nin okuyabilmesi için static değişken
    public static int CurrentWave = 1;

    void Start()
    {
        // Tüm spawnerları tag'e göre bul
        GameObject[] spawnerObjects = GameObject.FindGameObjectsWithTag(spawnerTag);

        if (spawnerObjects.Length == 0)
        {
            Debug.LogError("Spawner objeleri bulunamadı! Tagları kontrol et: " + spawnerTag);
            return;
        }

        // Transform array'ını doldur
        spawners = new Transform[spawnerObjects.Length];
        for (int i = 0; i < spawnerObjects.Length; i++)
        {
            spawners[i] = spawnerObjects[i].transform;
        }
    }

    if (enemyPrefab == null || target == null)
    {
        Debug.LogError("Enemy prefab veya target atanmadı!");
        return;
    }
```

DÜŞMAN HAREKETİ VE ZORLUK YÖNETİMİ

Düşmanlar Base objesini hedefler ve dalga ilerledikçe hızları dinamik olarak artırılır.

Hedef Belirleme

Düşmanlar Base objesini hedef olarak belirler. `target = GameObject.Find("Base");`

Dinamik Hız Hesaplaması

Hız, mevcut dalgaya göre güncellenir. Örnek: Wave 5'te hız 6 birim/saniye olur.

$$\text{DüşmanHızı} = \text{baseEnemySpeed} + (\text{currentWave} \times \text{speedIncreasePerWave})$$

Wave Yönetimi

EnemySpawner scripti, düşmanları belirli aralıklarla üretir ve zorluk parametrelerini günceller.

ARAYÜZ VE KONTROL ENTEGRASYONU

Oyuncu kontrolü (TowerMove) ve nişan alma/ateş etme (TowerAimAndShoot) scriptleri birlikte çalışır. Arayüz sistemleri (Score, Wave, Game Over) oyun durumunu yansıtır.



Oyuncu Kontrolü

Hareket hızı 10f. W/A/S/D tuşları ile hareket, fare ile dönüş sağlanması sağlanır.



Ateş Mekanikleri

Dönüş hızı 10f, ateş aralığı 0.25f. Hızlı ateş modu 3 saniye sürer, 7 saniye bekleme süresi vardır.



Score Manager

Statik değişken ile puan yönetimi, çarpışmada puan ekleme ve UI güncelleme.



Wave UI ve Zamanlama

Dalga numarasını gösterir. `InvokeRepeating(60f)` ile 60 saniyede bir dalga artırılır.

SONUÇ: BAŞARILAR VE GELİŞTİRME ALANLARI

Proje, dalga tabanlı savunma mekanığını dinamik zorluk artışı ve temiz kod yapısıyla başarıyla uygulamıştır. Teknik temel sağlamdır.

Proje Başarıları



Mekanik Uygulama

Dalga tabanlı savunma ve dinamik zorluk artışı başarıyla entegre edildi.



Kod Yapısı

Modüler ve temiz kod yapısı, 8 script ile global veri yönetimi sağlandı.



Sistem Optimizasyonu

Mermi-Enemy çarpışma sistemi ve Time.timeScale kullanımı optimize edildi.

Geliştirilmesi Gerekenler



Taramalı mod için görsel/ses geri bildirimi eklenmesi.



Farklı düşman türleri, davranışları ve Power-up sistemi eklenmesi.



Yüksek skor tablosu, başarı sistemi ve müzik entegrasyonu.

8

Script Entegre

10

Mekanik Detay

∞

Potansiyel