

KULE SAVUNMA OYUNU - DETAYLI YAPILACAKLAR LİSTESİ



PROJE GENELİ

Başlangıç Hazırlıkları

- ☐ Proje ekibini oluştur (2 kişilik)
- ☐ Programlama dilini seç (C#/Java/Python vb.)
- ☐ GUI kütüphanesini belirle (WinForms/WPF/JavaFX/Swing)
- ☐ Proje klasör yapısını oluştur
- ☐ Versiyon kontrol sistemi kur (Git)



FAZA 1: KONSEPT TASARIMI

Oyun Konsepti Seçimi

- ☐ Özgün bir tema seç (Antik Mısır, Uzay, Siber Şehir vb.)
- ☐ Tema ile ilgili isimlendirme stratejisi belirle
- ☐ Düşman türlerinin konseptte uygun isimlerini belirle
- ☐ Kule türlerinin konseptte uygun isimlerini belirle
- ☐ Oyun değişkenlerinin konseptte uygun isimlerini belirle (can → shieldIntegrity gibi)

Görsel Tasarım

- ☐ Arka plan tasarımını hazırla
- ☐ Yol geometrisi tasarımını yap
- ☐ Her kule türü için sembol/ikon tasarla (3 adet)
- ☐ Her düşman türü için ikon tasarla (3 adet)
- ☐ Efekt renklerini belirle (saldırı, yavaşlatma vb.)
- ☐ UI elemanlarının görsel tasarımını yap



FAZA 2: TEMEL SINIF MİMARİSİ

Soyut/Base Sınıflar

- ☐ Enemy (Düşman) base sınıfını oluştur
 - ☐ Özellikler: can, maksimumCan, hız, pozisyon, zırh
 - ☐ Metotlar: hareket(), hasarAl(), öl(), ussUlas()
- ☐ Tower (Kule) base sınıfını oluştur
 - ☐ Özellikler: hasar, menzil, ateşHızı, maliyet, pozisyon
 - ☐ Metotlar: hedefBul(), ateşEt(), hasarHesapla()

Düşman Sınıfları (3 adet)

- ☐ **Standart Düşman** sınıfını oluştur
 - ☐ Can: 50
 - ☐ Hız: 50
 - ☐ Zırh: 0
 - ☐ Uçma: Hayır
 - ☐ Ödül: 10 para
 - ☐ Üs hasarı: 5 can
- ☐ **Zırhlı Düşman** sınıfını oluştur
 - ☐ Can: 75 (50'nin %150'si)
 - ☐ Hız: 25 (50'nin %50'si)
 - ☐ Zırh: 50-100 arası rastgele
 - ☐ Uçma: Hayır
 - ☐ Ödül: 20 para
 - ☐ Üs hasarı: 10 can
- ☐ **Uçan Düşman** sınıfını oluştur
 - ☐ Can: 50
 - ☐ Hız: 75 (50'nin %150'si)
 - ☐ Zırh: 0
 - ☐ Uçma: Evet
 - ☐ Ödül: 15 para
 - ☐ Üs hasarı: 5 can

Kule Sınıfları (3 adet)

- ☐ **Okçu Kulesi** sınıfını oluştur
 - ☐ Hasar: 10 (zırhlıya %50 daha az)
 - ☐ Ateş hızı: 1 saniye
 - ☐ Hedefleme: Üsse en yakın tek hedef
 - ☐ Maliyet: 50 para
 - ☐ **Topçu Kulesi** sınıfını oluştur
 - ☐ Hasar: 20 (alan hasarı)
 - ☐ Ateş hızı: 3 saniye
 - ☐ Hedefleme: Üsse en yakın (uçanlar hariç)
 - ☐ Alan yarıçapı: 50 piksel
 - ☐ Maliyet: 75 para
 - ☐ **Buz Kulesi** sınıfını oluştur
 - ☐ Hasar: 15
 - ☐ Ateş hızı: 2 saniye
 - ☐ Yavaşlatma: %50, 3 saniye
 - ☐ Hedefleme: Üsse en yakın tek hedef
 - ☐ Maliyet: 70 para
-

FAZA 3: OYUN MEKANİĞİ

Oyun Başlangıcı

- ☐ Oyuncu sınıfı oluştur
 - ☐ Can: 100
 - ☐ Para: 200
- ☐ Dalga yönetim sistemi oluştur
 - ☐ Toplam 2 dalga olacak
- ☐ **Dalga 1** düşmanlarını tanımla
 - ☐ 2 Standart Düşman
 - ☐ 1 Zırhlı Düşman
 - ☐ 1 Uçan Düşman
 - ☐ Toplam: 4 düşman
- ☐ **Dalga 2** düşmanlarını tanımla
 - ☐ En az 5, en fazla 10 düşman
 - ☐ Her türden en az 1 tane

Harita ve Yol Sistemi

- ☐ Harita sınıfı oluştur
- ☐ Yol noktalarını tanımla (başlangıç → üs)
- ☐ Kule inşa alanlarını belirle
- ☐ Yol takip algoritması yaz

Düşman Hareketi

- ☐ Düşman hareket metodunu kodla
- ☐ Yol takip sistemi geliştir
- ☐ Hız değişimi sistemi (yavaşlatma için)
- ☐ Üsse ulaşma kontrolü
- ☐ Üsse ulaşıncı silme ve hasar verme

Kule Ateş Sistemi

- ☐ Menzil kontrolü algoritması
- ☐ Hedef bulma algoritması (üsse en yakın)
- ☐ Ateş hızı zamanlayıcısı (Timer)
- ☐ Hasar hesaplama formülü: $Net_Hasar = Kule_Hasarı * (1 - (Zırh / (Zırh + 100.0)))$
- ☐ Topçu kulesi alan hasarı sistemi
- ☐ Buz kulesi yavaşlatma efekti (3 saniye zamanlayıcı)

Oyun Döngüsü

- ☐ Ana oyun döngüsü (game loop) oluştur
- ☐ Zamanlayıcı sistemi kur (her frame güncelleme)
- ☐ Düşman pozisyon güncelleme
- ☐ Kule ateş kontrolleri

- ☐ Can/Para güncelleme
 - ☐ Kazanma kontrolü (tüm dalgalar bitti + haritada düşman yok)
 - ☐ Kaybetme kontrolü (oyuncu canı ≤ 0)
-



FAZA 4: GRAFİKSEL KULLANICI ARAYÜZÜ

Ana Menü

- ☐ Ana menü ekranı tasarla
- ☐ "Oyunu Başlat" butonu
- ☐ "Çıkış" butonu

Oyun Ekranı

- ☐ Harita çizim sistemi
 - ☐ Yolu çiz (başlangıç → bitiş)
 - ☐ Kule inşa alanlarını işaretle
 - ☐ Arka planı yerleştir
- ☐ Kullanıcı Bilgi Paneli
 - ☐ Oyuncu canı label/text
 - ☐ Para miktarı label/text
 - ☐ Dalga numarası label/text
- ☐ Kule İnşa Butonları
 - ☐ Okçu Kulesi butonu (maliyet göster)
 - ☐ Topçu Kulesi butonu (maliyet göster)
 - ☐ Buz Kulesi butonu (maliyet göster)
 - ☐ Buton aktif/pasif durumu (para kontrolü)

Dinamik Görselleştirme

- ☐ Düşman sağlık barı sistemi
 - ☐ Her düşmanın üstünde bar
 - ☐ Can/MaxCan oranını göster
 - ☐ Hasar alınca anlık güncelleme
- ☐ Düşman görselleri
 - ☐ Her düşman türü için farklı ikon/sprite
 - ☐ Hareket animasyonu
- ☐ Kule görselleri
 - ☐ Her kule türü için farklı ikon/sprite
 - ☐ Haritaya yerleştirme
- ☐ Efekt görselleri
 - ☐ Buz yavaşlatma efekti (mavi renk)
 - ☐ Saldırı çizgileri/mermi animasyonu

- ☐ Hasar göstergesi

Kullanıcı Etkileşimi

- ☐ Kule seçme sistemi (buton tıklama)
 - ☐ Haritada kule yerleştirme sistemi (mouse tıklama)
 - ☐ Para yeterliliği kontrolü
 - ☐ Boş alan kontrolü
 - ☐ Para düşürme ve kule oluşturma
-



FAZA 5: SİMÜLASYON GÜNLÜK DOSYASI

Log Sistemi

- ☐ Log dosyası oluşturma (savunma_gunlugu.txt)
- ☐ Zaman damgası sistemi

Kayıt Edilecek Olaylar

- ☐ Simülasyon başlangıcı (Can, Para)
- ☐ Kule inşa edilmesi (konum, ID, kalan para)
- ☐ Dalga başlangıcı (dalga no, düşman türü, sayı)
- ☐ Düşman haritaya girmesi (ID, Can, Zırh)
- ☐ Kule hedefleme (kule ID, düşman ID, hedefleme nedeni)
- ☐ Ateş etme ve hasar (taban hasar, zırh cezası, net hasar, kalan can)
- ☐ Yavaşlatma efekti (düşman ID, süre, oran)
- ☐ Alan hasarı detayları (merkez hedef, etkilenen düşmanlar)
- ☐ Düşman ölümü (ID, ödül, toplam para)
- ☐ Üsse ulaşma (ID, oyuncu canı azalması)
- ☐ Oyun sonu (kazanma/kaybetme, kalan can, toplam para)

Log Formatı

- ☐ Konsept diline uygun isimlendirme kullan
 - ☐ Her olay detaylı açıklama içermeli
 - ☐ Sayısal değerler net görünmeli
-



FAZA 6: TEST VE HATA AYIKLAMA

Birim Testleri

- ☐ Düşman hareket testleri
- ☐ Kule ateş testleri
- ☐ Hasar hesaplama testleri
- ☐ Zırh formülü testleri
- ☐ Yavaşlatma efekti testleri

- ☐ Alan hasarı testleri

Entegrasyon Testleri

- ☐ Tam oyun senaryosu testi
- ☐ Dalga geçişleri testi
- ☐ Kazanma senaryosu testi
- ☐ Kaybetme senaryosu testi
- ☐ GUI-simülasyon senkronizasyonu testi

Hata Kontrolü

- ☐ Null pointer/reference kontrolleri
- ☐ Sınır değer testleri
- ☐ Para yeterliliği hataları
- ☐ Zamanlayıcı senkronizasyon hataları



FAZA 7: PROJE RAPORU (IEEE FORMAT)

Rapor Bölümleri

- ☐ **Özet** (Abstract)
 - ☐ Projenin amacı
 - ☐ Kullanılan teknolojiler
 - ☐ Sonuçlar
- ☐ **Giriş** (Introduction)
 - ☐ Kule savunma oyunları hakkında
 - ☐ Proje hedefleri
 - ☐ NYP ilkelerinin önemi
- ☐ **Yöntem** (Methodology)
 - ☐ UML sınıf diyagramları (tüm sınıflar)
 - ☐ GUI-simülasyon motor haberleşmesi
 - ☐ Tasarım desenleri
 - ☐ Algoritma açıklamaları
- ☐ **Oyun Konsepti** (Game Concept)
 - ☐ Seçilen temanın gerekçesi
 - ☐ Oyun mekaniklerine yansımaları
 - ☐ İsimlendirme stratejisi tablosu
 - ☐ Görsel tasarım kararları
- ☐ **Simülasyon** (Simulation)
 - ☐ savunma_gunlugu.txt örnek çıktıları
 - ☐ Ekran görüntüleri (en az 5-6 adet)
 - ☐ Oyun akışı açıklaması
- ☐ **Sonuç** (Conclusion)

- ☐ Elde edilen sonuçlar
- ☐ Karşılaşılan zorluklar
- ☐ Gelecek geliřtirmeler
- ☐ **Yazar Katkıları** (Author Contributions)
 - ☐ Her üyenin yaptığı işler
- ☐ **Kaynakça** (References)
 - ☐ Kullanılan kaynaklar (IEEE format)

Rapor Formatı

- ☐ LaTeX veya Word kullan
 - ☐ PDF formatında kaydet
 - ☐ En az 4 sayfa
 - ☐ IEEE formatına uy
-

FAZA 8: SUNUM HAZIRLIĞI

Sunum İçeriğı

- ☐ Demo videosu hazırla
- ☐ Kod açıklama senaryosu yaz
- ☐ NYP ilkeleri örnekleri hazırla
- ☐ Sorulabilecek soruları tahmin et

Teknik Hazırlık

- ☐ Kodun her satırını açıklayabilir durumda ol
- ☐ Kalıtım örnekleri göster
- ☐ Polimorfizm kullanımını açıkla
- ☐ Kapsülleme örnekleri göster
- ☐ Soyutlama mantığını açıkla

Canlı Test

- ☐ Sunumda çalışacak bilgisayarı test et
 - ☐ Kod değıřtirip çalıştırma pratiğı yap
 - ☐ Yedek dosyalar hazırla
-

FAZA 9: TESLİM

Teslim Dosyaları

- ☐ Kaynak kod dosyaları (.cs, .java vb.)
- ☐ Proje raporu (PDF)
- ☐ savunma_gunlugu.txt örnek çıktısı

- ☐ UML diyagramları (ayrı dosya)
- ☐ README dosyası (kurulum talimatları)
- ☐ Görsel dosyalar (asset'ler)

Sistem Kontrolü

- ☐ edestek2.kocaeli.edu.tr sistemine erişim
 - ☐ Teslim tarihini kontrol et: **30.11.2025**
 - ☐ 3 gün öncesine kadar soru sor
-



ÖNEMLİ HATIRLATMALAR

NYP İlkeleri Kontrol Listesi

- ☐ **Kalıtım:** Enemy ve Tower base sınıfları doğru kullanılmış mı?
- ☐ **Polimorfizm:** Farklı düşman/kule türleri aynı arayüzle çağrılıyor mu?
- ☐ **Kapsülleme:** Private/protected değişkenler, getter/setter kullanımı
- ☐ **Soyutlama:** Abstract/interface kullanımı uygun mu?

Puan Kriterler Kontrol

- ☐ NYP ilkeleri doğru uygulanmış
- ☐ GUI canlı ve akıcı çalışıyor
- ☐ Log dosyası eksiksiz
- ☐ Rapor profesyonel ve detaylı
- ☐ Konsept özgün ve tutarlı
- ☐ Kod okunabilir ve düzenli

Son Kontroller

- ☐ Tüm isterleri karşılıyor mu?
 - ☐ Oyun baştan sona çalışıyor mu?
 - ☐ Hiç hata/crash olmuyor mu?
 - ☐ Arayüz tüm bilgileri gösteriyor mu?
 - ☐ Log dosyası doğru yazılıyor mu?
-



ÖNERİLEN ZAMAN ÇİZELGESİ

- **Hafta 1:** Konsept seçimi, sınıf mimarisi tasarımı
 - **Hafta 2:** Temel sınıfları kodlama, oyun mekaniği
 - **Hafta 3:** GUI geliştirme, entegrasyon
 - **Hafta 4:** Test, hata ayıklama, log sistemi
 - **Hafta 5:** Rapor yazımı, sunum hazırlığı
 - **Teslim:** 30.11.2025
-

NOT: Bu liste projenin tüm gereksinimlerini en temel seviyede karşılayacak şekilde hazırlanmıştır. Her maddeyi tamamladıkça işaretleyin ve takım arkadaşınızla düzenli olarak ilerlemeyi kontrol edin. Başarılar! 🚀