

MARMARA ÜNİVERSİTESİ TEKNOLOJİ

FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ

2024-2025 DÖNEMİ

BİLGİSAYAR AĞLARINA GİRİŞ

PROJE FİNAL RAPORU

PROJE ADI: Yerel Ağda Gerçek Zamanlı Çok Oyunculu Oyun

GRUP NO: 14

DANIŞMAN: Merve Hazan İşcan

HAZIRLAYANLAR:

Muhammet Batuhan Karanfil

Muhammet Enes Yıldırır

Muhammet Vefa Yoksul

İçindekiler

[1. Giriş 3](#_Toc198577329)

[2. Yöntem 3](#_Toc198577330)

[2.1 Kullanılan Yazılımlar ve Teknolojiler 3](#_Toc198577331)

[2**.**2 Neler Yaptık 4](#_Toc198577332)

[3. Bulgular 5](#_Toc198577333)

[5. Github 7](#_Toc198577334)

# **Giriş**

Bu proje, bilgisayar ağları dersinde öğrenilen teorik bilgileri pratik bir yazılım projesiyle pekiştirmek amacıyla geliştirilmiştir. Proje kapsamında, TCP soketleri kullanılarak iki oyunculu bir Amiral Battı (Battleship) oyunu tasarlanmış ve uygulanmıştır.

Oyuncular, kendi bilgisayarlarında çalışan istemci (client) uygulamaları üzerinden, merkezi bir sunucuya (server) bağlanarak gerçek zamanlı şekilde etkileşimde bulunurlar. Her iki oyuncu sırayla hamle yaparak rakibin gemilerini vurmaya çalışır. Tüm iletişim, TCP/IP protokolü üzerinden sağlanır ve oyun ilerleyişi sunucu tarafından kontrol edilir.

Ayrıca, kullanıcı deneyimini artırmak amacıyla oyun arayüzü Python’un Pygame kütüphanesi ile grafiksel ve sesli olarak desteklenmiştir. Bu sayede proje, sadece ağ programlama bilgisini değil, aynı zamanda kullanıcı arayüzü tasarımı ve olay yönetimi gibi yazılım geliştirme konularını da kapsamaktadır.

# **Yöntem**

## 2.1 Kullanılan Yazılımlar ve Teknolojiler

* Python 3.13 – Oyun sunucusu ve istemci uygulamaları bu dil ile geliştirildi.
* Socket Kütüphanesi – TCP/IP protokolü üzerinden ağ bağlantıları sağlandı.
* JSON Modülü – İstemci ve sunucu arasındaki mesajlaşma yapısında kullanıldı.
* Pygame Kütüphanesi – Oyuncunun gemi yerleştirmesini ve hamle yapmasını sağlayan grafiksel arayüz geliştirildi.
* Threading Modülü – Sunucunun aynı anda birden fazla istemciyle iletişim kurmasını sağladı.

## 2**.**2 Neler Yaptık

Proje Python programlama dili ile geliştirilmiştir. Ağ haberleşmesi için Python’un `socket` modülü, çoklu istemci yönetimi için `threading` kütüphanesi kullanılmıştır. Görsel arayüz tasarımı ve kullanıcı etkileşimi için ise `pygame` kütüphanesinden yararlanılmıştır. Oyuncular arasındaki mesajlaşma yapısı ise JSON formatında yapılandırılmıştır.

Proje üç ana bileşenden oluşmaktadır:

1. Sunucu(server.py)

- Oyuncuları sırayla kabul eder.

- Gelen mesajları işler ve oyuncular arası veri akışını sağlar.

- Oyunun başlama, hamle yapma ve bitiş mantığını yönetir.

- Tüm oyun akışı sunucu tarafından kontrol edilir.

2. İstemci 1 (client1.py)

- Oyuncu 1’in arayüzüdür.

- Gemi yerleştirme, hamle yapma ve rakip hareketlerini görme gibi tüm işlemleri sağlar.

- Grafiksel olarak Pygame ile tasarlanmıştır.

- Server ile JSON mesajları üzerinden iletişim kurar.

3. İstemci 2 (client2.py)

- Oyuncu 2 için oluşturulmuş arayüzdür.

- client1.py ile aynı yapıya sahiptir, yalnızca oyuncu ismi farklıdır.

Başlıca Adımlar:

- Oyun ekranları oluşturulmuş (başlangıç, gemi yerleştirme, oyun, oyun sonu),

- Gemilerin sürüklenerek yerleştirilebileceği bir sistem tasarlanmış,

- Oyuncuların gemi pozisyonları sunucuya gönderilmiş,

- Her iki oyuncu hazır olduğunda oyun başlamış,

- Hamle sırası sunucu tarafından belirlenmiş ve oyuncular arasında sıra dönüşümlü olarak işlenmiş,

- Yapılan hamlelerin sonucu (miss, hit, sink) oyunculara bildirilmiş,

- Oyun sonunda kazanan belirlenmiş ve oyunculara bildirilmiştir.

# **Bulgular**

Geliştirilen Amiral Battı oyunu, dört farklı ekran yapısıyla kullanıcıya etkileşimli bir oyun deneyimi sunmaktadır. Tüm ekranlar Pygame kütüphanesi ile grafiksel olarak tasarlanmıştır ve kullanıcı etkileşimi fare ve klavye üzerinden sağlanmaktadır.

Aşağıda her ekranın işlevi ve örnek görselleri sunulmuştur:

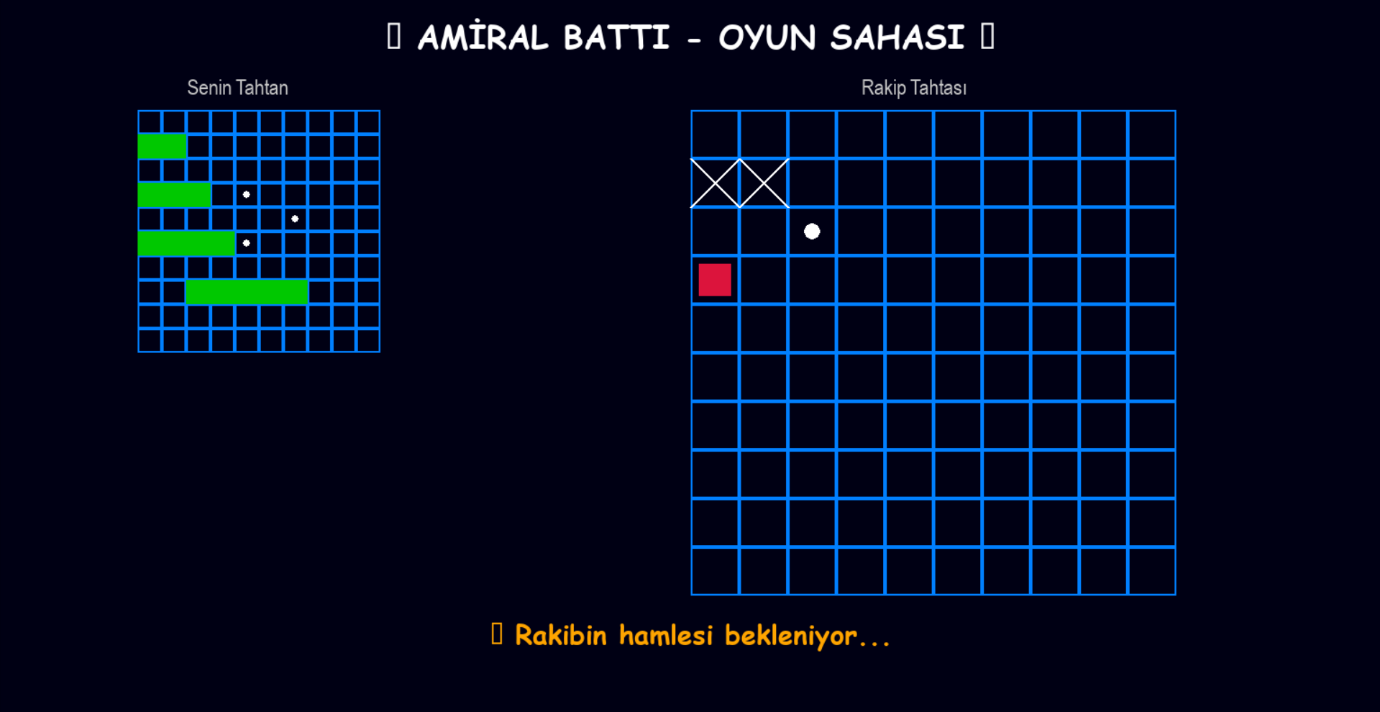
1.Başlangıç Ekranı (Start Screen)

Oyunun açılışında karşılaşılan ekrandır. Arka planda deniz görseli ve müzik yer alır. Kullanıcı bir tuşa bastığında veya tıkladığında oyun gemi yerleştirme ekranına geçer.

ekran görüntüsü, metin, tasarım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.2. Gemi Yerleştirme Ekranı (Placement Screen)

Oyuncu, gemilerini oyun tahtasına sürükleyerek yerleştirir. “R” tuşu ile geminin yönü yatay-dikey olarak değiştirilebilir. Tüm gemiler yerleştirildikten sonra “BAŞLA” butonuna tıklanarak oyun başlatılır.

3.Oyun Ekranı (Gameplay Screen)

İki oyuncu sırayla hamle yapar. Sol üstte küçük bir grid üzerinden kendi gemileri ve rakibin vuruşları görüntülenir. Sağda ise rakip oyuncunun tahtasına tıklayarak hamle yapılır. Her hamlede "miss", "hit" veya "sink" durumları animasyon ve ses ile gösterilir.

1. metin, ekran görüntüsü, yazı tipi, logo içeren bir resim

   Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Oyun Sonu Ekranı (Game Over Screen)

Bir oyuncunun tüm gemileri battığında oyun sona erer. Kazanan oyuncuya özel bir ekran gösterilir ve “Tekrar Oyna” veya “Çık” butonlarıyla seçim yapması istenir.

# **Github**

Proje, GitHub üzerinde herkese açık bir şekilde yayınlanmıştır. Tüm istemci ve sunucu dosyaları, görseller, ses dosyaları ve README açıklaması yüklenmiştir.

🔗 GitHub Linki:

https://github.com/mVefa/battleship-game

📸 Aşağıda GitHub sayfasına ait ekran görüntüsü yer almaktadır.

metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.