



MARMARA
ÜNİVERSİTESİ

MARMARA ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİLGİSAYAR AĞLARINA GİRİŞ FİNAL RAPORU

Proje Adı:Python ile GUI Tabanlı Çok Oyunculu Tic Tac Toe Oyunu

Hazırlayanlar: Büşra Ecem Özbek, Ahmet Eren Başalı

Teslim Tarihi: 19 Mayıs 2025

1. Giriş
2. Yöntem
 - 2.1. Kullanılan Teknolojiler
 - 2.2. Ağ Mimarisi ve OSI Katmanları
 - 2.3. Python Socket API Detayları
 - 2.4. Blocking vs Non-blocking & Threading
 - 2.5. Veri Protokolü Tasarımı
 - 2.6. Geliştirme Ortamı
 - 2.7. Grafik Arayüz (GUI)
3. Bulgular
4. GitHub Deposu
5. Uygulama Görselleri
6. Sonuç ve Değerlendirme

1. Giriş

Bu proje, Bilgisayar Ağlarına Giriş dersi kapsamında istemci-sunucu mimarisi kullanılarak geliştirilen, gerçek zamanlı iki oyunculu *Tic Tac Toe – Neon Edition* oyununu konu almaktadır. TCP/IP temelli bağlantı kurulumu, veri iletimi, eşzamanlı işlem yapısı ve uygulama katmanı protokolü gibi temel kavramların uygulamalı olarak pekiştirilmesi hedeflenmiştir. Proje, hem teorik bilgilerin pratiğe dökülmesini hem de modern grafiksel kullanıcı arayüzüyle etkileşimli bir yazılım geliştirme deneyimi sunmuştur.

2. Yöntem

2.1. Kullanılan Teknolojiler

- Python 3.12: Ana geliştirme dili olarak tercih edilmiştir.
- socket: TCP/IP tabanlı bağlantı kurmak ve veri iletmek için kullanılmıştır.
- threading: Çoklu bağlantı ve arka planda veri dinleme işlemleri için kullanılmıştır.
- queue.Queue: Thread-safe mesaj iletimi sağlamak amacıyla tercih edilmiştir.
- customtkinter: Neon temalı gelişmiş bir kullanıcı arayüzü oluşturmak için kullanılmıştır.

2.2. Ağ Mimarisi ve OSI Katmanları

- Fiziksel & Veri Bağlantı Katmanı: Sanal makineler bridged adapter ile yerel ağa dahil edilmiştir.
- Ağ Katmanı: IP adresleme ve yönlendirme işlemleri işletim sistemi tarafından gerçekleştirilmiştir.
- Taşıma Katmanı: socket.AF_INET ve socket.SOCK_STREAM ile TCP bağlantısı kurulmuştur.
- Oturum Katmanı: connect() ve accept() metodları ile bağlantı oturumu başlatılmıştır.
- Sunum Katmanı: UTF-8 ile veri kodlama/çözme işlemleri yapılmıştır.
- Uygulama Katmanı: Satır tabanlı özel protokol ile istemciler arasında haberleşme sağlanmıştır.

2.3. Python Socket API Detayları

Sunucu tarafında: `socket.socket()` ile TCP soketi oluşturulur, `bind()` ile IP ve port atanır, `listen(2)` ile bağlantı beklenir, `accept()` ile istemci kabul edilir. İstemci tarafında: `socket.socket().connect()` ile sunucuya bağlanılır, `sendall()` ile veri gönderilir, `recv()` ile veri alınır.

2.4. Blocking vs Non-blocking & Threading

`recv()` fonksiyonu bloklayıcıdır; veri gelene kadar bekler. `conn.setblocking(False)` ile non-blocking hale getirilerek sistemin donması engellenebilir. Sunucu tarafında her istemci bağlantısı ayrı bir thread ile yönetilir. İstemcide mesaj dinleme işlemi arka planda yapılır. GUI thread'i ise `after()` fonksiyonu ile güncellemeleri işler.

2.5. Veri Protokolü Tasarımı

İstemciler ve sunucu arasında aşağıdaki satır tabanlı komutlar kullanılmıştır:

- `ROLE;X`: Oyuncuya X veya O rolü atanır.
- `BOARD;XXX; 0 ; 0X0`: Oyun tahtasının güncel durumu.
- `PROMPT;Your turn`: Sıra bildirimi.
- `CHAT;X;Merhaba`: Sohbet mesajı.
- `RESULT;X wins`: Oyun sonucu bildirimi.
- `REMATCH;REQUEST`, `REMATCH;ACCEPT`, `REMATCH;DECLINE`, `REMATCH;START`: Yeni oyun başlatma süreci.

2.6. Geliştirme Ortamı

Sunucu ve istemci uygulamaları, Windows işletim sistemi üzerinde çalışan Oracle VM VirtualBox sanal makinelerinde test edilmiştir. Bridged ağ yapısı kullanılarak iki sanal makine aynı yerel ağa dahil edilmiştir. Geliştirme ortamı olarak PyCharm Professional tercih edilmiştir. Tüm bağlantılar aynı fiziksel ağ üzerinden gerçekleştirilmiştir.

2.7. Grafik Arayüz (GUI)

`customtkinter` kütüphanesi kullanılarak neon temalı, modern bir kullanıcı arayüzü geliştirilmiştir. GUI ve ağ iletişimi ayrı thread'lerde yönetilmiş, böylece arayüzde donma veya gecikme yaşanmamıştır. Oyunun bitişinde kazanan animasyonu, renkli kare vurguları ve yeniden oyun isteği gibi etkileşimli öğeler yerleştirilmiştir. Butonlar üzerinden sohbet, oyun sırası ve sonuç gibi bilgiler dinamik olarak kullanıcıya sunulmuştur.

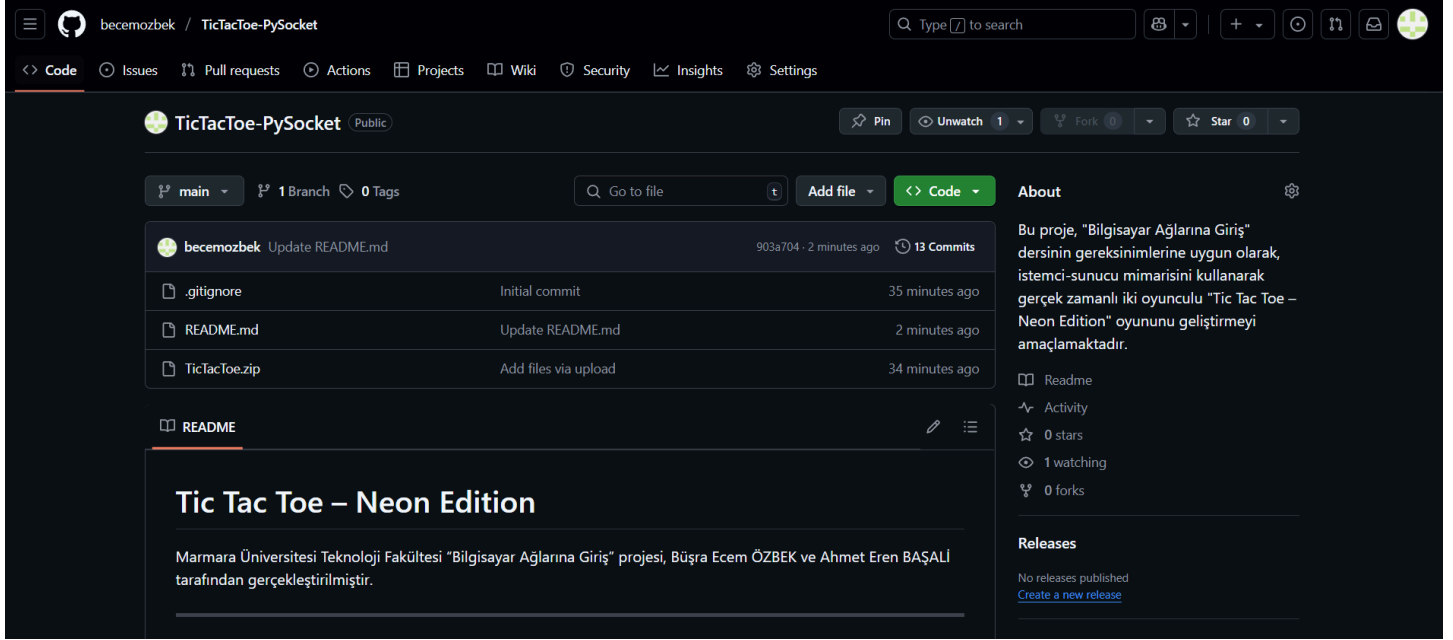
3. Bulgular

- İki istemci arasında gerçek zamanlı ve kesintisiz TCP bağlantısı kurulmuştur.
- GUI, arka planda çalışan ağ trafiğinden etkilenmeden stabil şekilde çalışmıştır.
- Kullanıcı arayüzünde neon renk teması, görsel geçişler ve kazanan animasyonu başarıyla uygulanmıştır.
- Oyun sonunda iki taraf arasında yeniden rekabet süreci etkili şekilde yönetilmiştir.

4. GitHub Deposu

Projenin tüm kaynak kodları aşağıdaki GitHub bağlantısından erişilebilir durumdadır:

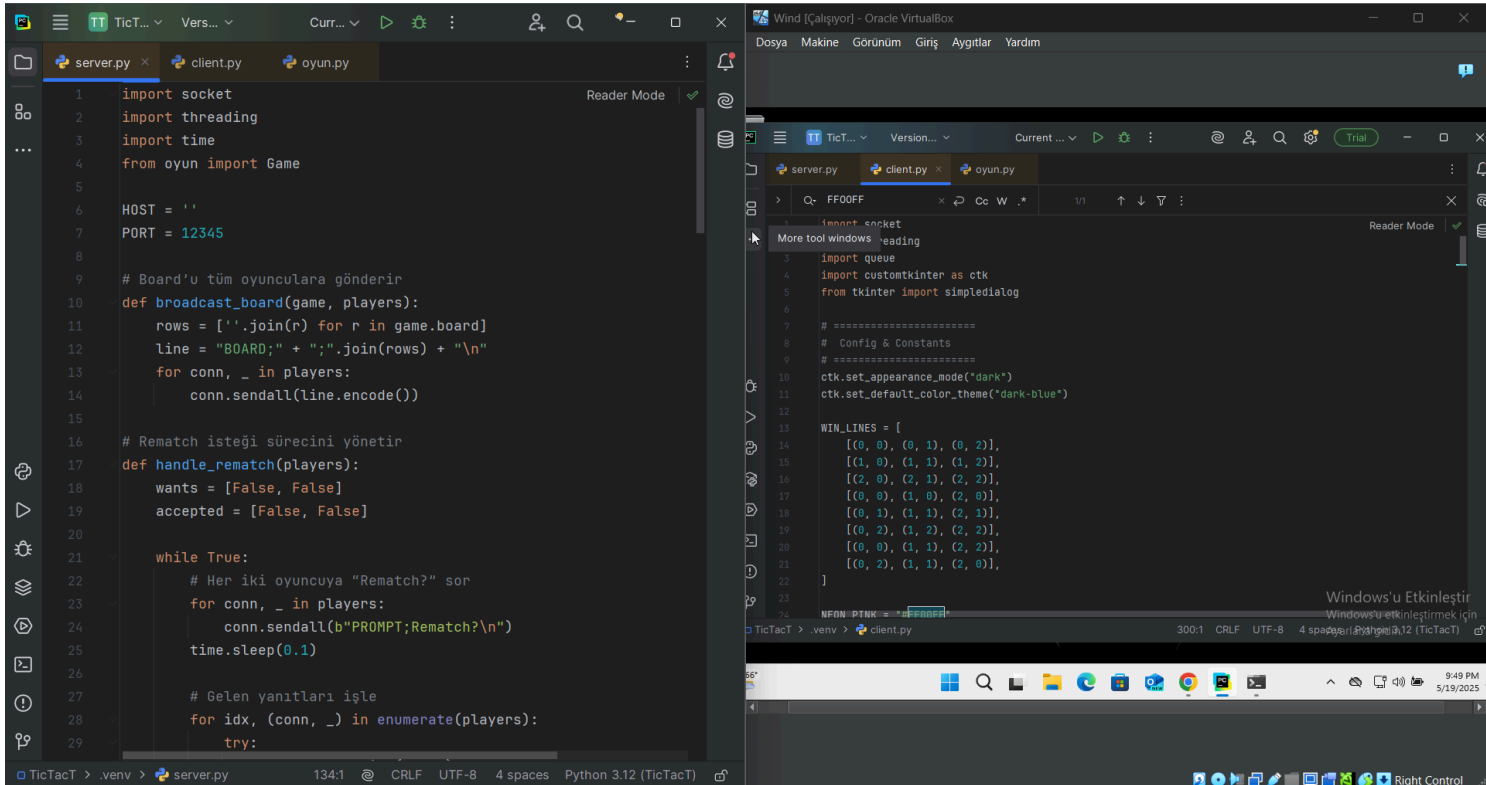
<https://github.com/becemozbek/TicTacToe-PySocket>



The screenshot shows the GitHub repository page for "TicTacToe-PySocket" by user "becemozbek". The repository is public and has 1 branch and 0 tags. The commit history shows three recent commits: ".gitignore" (Initial commit, 35 minutes ago), "README.md" (Update README.md, 2 minutes ago), and "TicTacToe.zip" (Add files via upload, 34 minutes ago). The README content is titled "Tic Tac Toe – Neon Edition" and describes the project as a 3x3 grid game implemented in Python using sockets. The repository statistics show 0 stars, 1 watching, and 0 forks.

5. Uygulama Görselleri

Uygulama ekranı, neon renkli butonlarla donatılmış 3x3 karelik bir oyun alanı içerir. Sağ alt köşede kullanıcı sohbeti, üstte oyun sırası ve sonuç bilgisi, ekran sonunda ise yeniden rekabet isteği gösterilmektedir. Görsellerde GUI ile birlikte ağ bağlantı mesajları da anlık şekilde izlenebilmektedir.



The screenshot shows the application running in a virtual machine. The left window displays the Python code for the server and client. The right window shows the GUI with a 3x3 grid and a chat area. The server code includes imports for socket, threading, and time, and defines functions for broadcasting board state and handling rematch requests. The client code includes imports for tkinter and tkinter.simpledialog, and defines a list of win lines for the 3x3 grid.

```
1 import socket
2 import threading
3 import queue
4 import customtkinter as ctk
5 from tkinter import simpledialog
6
7 # =====
8 # Config & Constants
9 # =====
10 ctk.set_appearance_mode("dark")
11 ctk.set_default_color_theme("dark-blue")
12
13 WIN_LINES = [
14     [(0, 0), (0, 1), (0, 2)],
15     [(1, 0), (1, 1), (1, 2)],
16     [(2, 0), (2, 1), (2, 2)],
17     [(0, 0), (1, 0), (2, 0)],
18 ]
19
20 NEON_GREEN = "#39FF14"
21 WIN_GREEN = "#39FF14"
22 LOSE_RED = "#FF0000"
```

Sunucu IP adresini gir:

192.168.1.9

OK Cancel

Komut İstemi

Link-local IPv6 Address : fe80::1da1:3ae7:5d7b:8dda%8

IPv4 Address. : 192.168.1.9

Subnet Mask : 255.255.255.0

Default Gateway : 192.168.1.1

```
1 import socket
2 import threading
3 import queue
4 import customtkinter as ctk
5 from tkinter import simpledialog
6
7 # =====
8 # Config & Constants
9 # =====
10 ctk.set_appearance_mode("dark")
11 ctk.set_default_color_theme("dark-blue")
12
13 WIN_LINES = [
14     [(0, 0), (0, 1), (0, 2)],
15     [(1, 0), (1, 1), (1, 2)],
16     [(2, 0), (2, 1), (2, 2)],
17     [(0, 0), (1, 0), (2, 0)],
18     [(0, 1), (1, 1), (2, 1)],
19     [(0, 2), (1, 2), (2, 2)],
20     [(0, 0), (1, 1), (2, 2)],
21     [(0, 2), (1, 1), (2, 0)],
22 ]
23
24 NEON_GREEN = "#39FF14"
25 WIN_GREEN = "#39FF14"
26 LOSE_RED = "#FF0000"
```

Sunucu IP adresini gir:

192.168.1.9

OK Cancel

```
1 import socket
2
3 Tic Tac Toe — Neon Edition
4
5 Sıra sende (X)
6
7 O X
8
9 X O
10
11 Ben: SELAMLAR
12 Rakip: Selamlar
13
14 Gönder
```

```
1 import socket
2
3 Tic Tac Toe — Neon Edition
4
5 Rakip bekleniyor (O)
6
7 O X
8
9 X O
10
11 Rakip: SELAMLAR
12 Ben: Selamlar
13
14 Gönder
```

```
1 import socket
2
3 Tic Tac Toe — Neon Edition
4
5 Kazandın
6
7 O X
8
9 X O
10
11 Ben: SELAMLAR
12 Rakip: Selamlar
13 === Oyun bitti: Kazandın ===
14
15 Gönder
```

```
1 import socket
2
3 Tic Tac Toe — Neon Edition
4
5 Rakip kazandı
6
7 O X
8
9 X O
10
11 Rakip: SELAMLAR
12 Ben: Selamlar
13 === Oyun bitti: Rakip kazandı ===
14
15 Gönder
```

6. Sonu ve Deęerlendirme

Bu proje, Bilgisayar Aęlarına Giriř dersi kapsamında aę programlama konularının pratikte nasıl uygulanabileceğini etkili řekilde göstermiştir. TCP/IP baęlantısı, socket API kullanımı, oklu iř paracıęı, veri protokolü tasarımı ve GUI geliştirme gibi birçok temel beceri entegre edilerek iřlevsel bir yazılım ürünü ortaya ıkarılmıştır. Öğrencinin hem teorik bilgileri kavraması hem de bu bilgileri gerek zamanlı bir uygulamaya dönüřtürebilmesi açısından son derece faydalı bir süreç gerekleşmiştir. Ayrıca GUI ile aę yapısının doęru ayrıştırılması, kullanıcı deneyimini sorunsuz hale getirmiştir. Proje, dersin hedeflerine uygun olarak başarılı řekilde tamamlanmıştır.

Bu projede edindiğim bilgiler ve geliştirme süreci boyunca sağladıkları rehberlik için **Bilgisayar Aęlarına Giriř** dersini veren Merve Hocamıza teşekkür ederiz.