

T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ
YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ



BULUT BİLİŞİM ONLINE OYUN PROJESİ

Kadir Ozan ATEŞ

DANIŞMAN

Dr. Hüseyin PARMAKSIZ

BİLECİK 2025

Özet

“Online Toy Tanks” bulut biliřim dersi iin yapılan online tabanlı  boyutlu bir oyundur. Oyunda kullanılan oyun motoru Unity Engine’dir ve C# dili ile geliřtirildi. Oyunun temel amacı, yerel aėdan online servere baėlanan oyuncular birbirileri ile savařarak mcadele etmesidir. Oyuncak tank kontrol eden oyuncular ateř ederek savařabilir. En son hayatta kalan oyuncu ise kazanan olarak belirlenir.

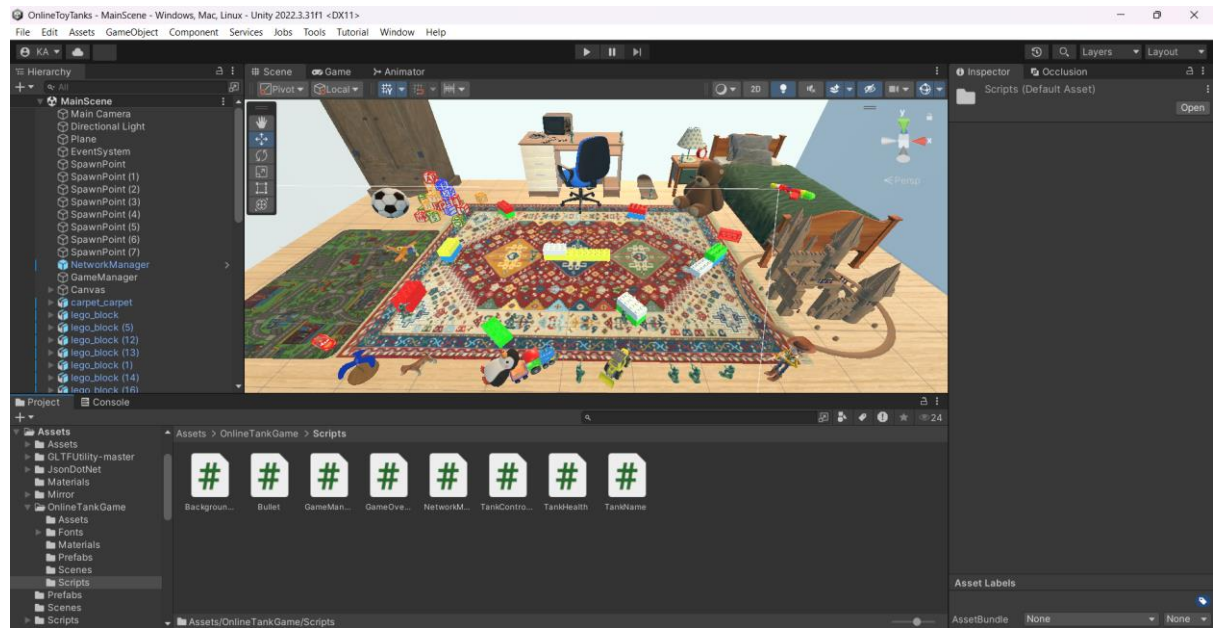
Giriř

Oyunun online sistemi temelde ip adresi üzerinden yerel ağ ile online bir şekilde oynanmaktadır. Oyun kurucu “Host” olup odayı kurabilir ve diğer oyuncular aynı internet üzerinden bağlanarak kurulan odaya katılabilir.

Oyunun Geliştirilme Süreci

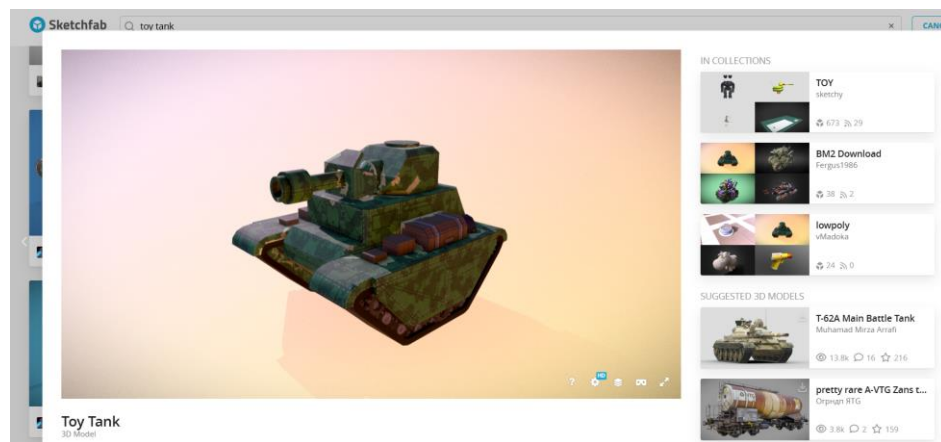
Kullanılan Oyun Motoru:

Oyun C# dilinde Unity Engine kullanılıp 2022.3.31f1 sürüm versiyonu ile geliştirilmiştir.



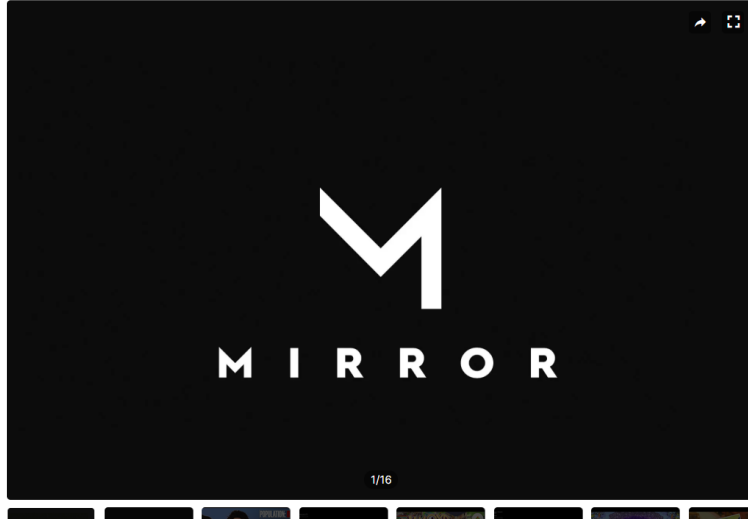
3 Boyutlu Modeller:

Oyundaki 3 boyutlu objeler hazır olarak “Sketchfab” sitesinden indirildi.



Online Sistem:

Oyunun online yapısı Unity Asset Store'daki “Mirror” paketi ile geliştirildi.



Mirror

Mirror Networking

★★★★★ (309) | ♥ (6425)

FREE

1540 views in the past week

Add to My Assets



License agreement [Standard Unity Asset Store EULA](#)

License type [Extension Asset](#)

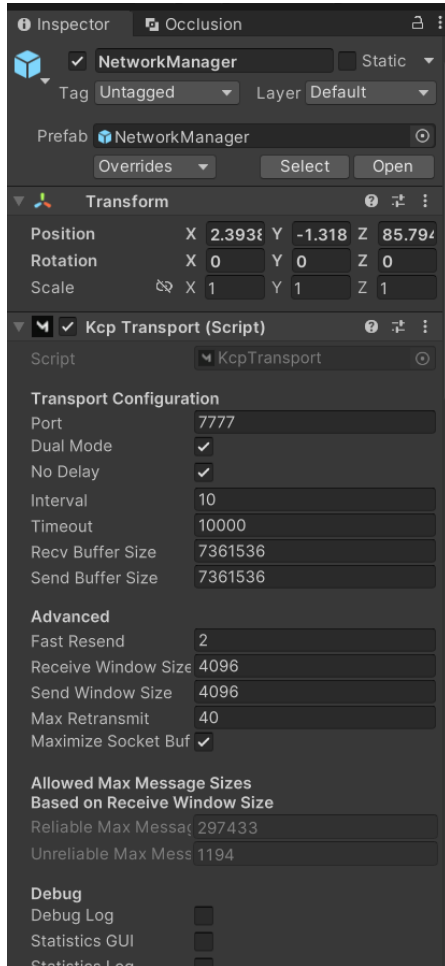
File size 13.6 MB

Latest version 96.0.1

Latest release date Feb 27, 2025

Original Unity version 2021.3.35

Support [Visit site](#)



Network Manager, online sistemde gerekli ayarlamalar yapmamıza yarayan unity objesidir. Burada ip adresi ayarlamasını, sahne geçişlerini, online oyuncu tanımlaması gibi ayarlamalar yapabiliriz.

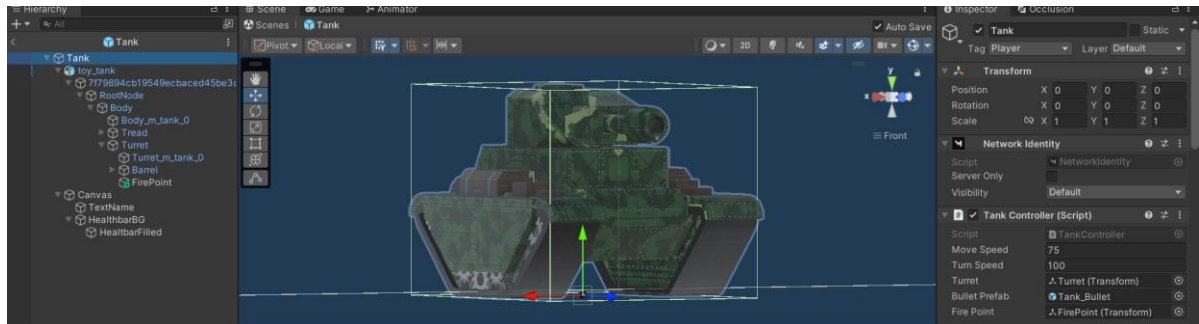
Online Oyuncu Kontrolü:

Her oyuncu kendi tankını kontrol edebilmesi için kod yazıldı.

```
TankController.cs
Assets > OnlineTankGame > Scripts > TankController.cs > ...
1 using UnityEngine;
2 using Mirror;
3
4 public class TankController : NetworkBehaviour
5 {
6     public float moveSpeed = 5f;
7     public float turnSpeed = 100f;
8
9     public Transform turret;
10    public GameObject bulletPrefab;
11    public Transform firePoint;
12
13    private float fireCooldown = 1f;
14    private float lastFireTime = -999f;
15
16    void Update()
17    {
18        if (!isLocalPlayer) return; // Sadece local player kendi girişlerini işlesin
19
20        HandleMovement();
21        HandleTurret();
22        HandleFire();
23    }
24
25    void HandleMovement()
26    {
27        float moveInput = Input.GetAxis("Vertical");
28        float turnInput = Input.GetAxis("Horizontal");
```

Oyuncu W, A, S, D den tankı hareket ettirebilir ve Boşluk tuşundan da ateş edebilir. Nişan almak için tankın üst kısmını O ve P tuşları ile sağa, sola döndürebilir. Ateş etmenin belli bir bekleme süresi vardır. Yani saniyede 1 kurşun fırlatabilir her tank.

Kodlamada offline oyunlardan farklı olarak “MonoBehaviour” yerine “NetworkBehaviour” kullanıyoruz. Çünkü internet üzerinden oynanıyor ve diğer oyuncuların bilgisayarında da yaptığımız hareketler gözükmelidir.



Her oyuncunun tankı aynı olduğu için tank objesi Prefab olarak ayarlandı. Network Transform ve Network Identity gibi bileşenler online sistemde önemlidir. Ayrıca oyuncuların birbirileri ile etkileşime girebilmesi için tanka Collider bileşeni eklendi. Bunu “hitbox” olarak da düşünebilirsiniz.

Merminin Kodu:

Tankların fırlattığı mermiye de birkaç özellik eklemek için kod yazıldı.

```
Bullet.cs
Assets > OnlineTankGame > Scripts > Bullet.cs > ...
1 using UnityEngine;
2 using Mirror;
3
4 0 references
5 public class Bullet : NetworkBehaviour
6 {
7     1 reference
8     public float speed = 10f;
9     1 reference
10    public float lifeTime = 5f;
11    1 reference
12    public int damage = 25;
13
14    2 references
15    public GameObject impactEffectPrefab; // <-- efekt prefabı
16
17    0 references
18    public override void OnStartServer()
19    {
20        Invoke(nameof(DestroySelf), lifeTime);
21    }
22
23    0 references
24    void Update()
25    {
26        if (isServer)
27        {
28            transform.Translate(Vector3.forward * speed * Time.deltaTime);
29        }
30    }
31
32    [ServerCallback]
33    0 references
34    private void OnTriggerEnter(Collider other)
35    {
36        if (!isServer) return;
37
38        // Eğer oyuncuya çarptıysa hasar ver
```

Bu kod mermiye belli bir hızda hareket etmesi, belli bir mesafeyi geçince yok olması, oyuncuya çarpınca etkilediği hasar ve bir yere temas edince patlama efekti oluşması gibi özellikler ekliyor.

Mermi objesi de Prefab olarak ayarlandı. Çünkü mermiler tankın “Firepoint” dediğimiz yerinden oluşup hareket ediyor.

Tank Can Barı Sistemi:



Tanklar mermilerden hasar aldığı zaman canlarının azaldığını gösteren can barı sistemi yapıldı. Her tank 4 kurşun isabet aldıktan sonra patlamaktadır. Patlama efekti "Particle effect" ile yapılmaktadır.

```
TankHealth.cs X
Assets > OnlineTankGame > Scripts > TankHealth.cs > ...
1  using UnityEngine;
2  using UnityEngine.UI;
3  using TMPro;
4  using Mirror;
5
6  0 references
7  public class TankHealth : NetworkBehaviour
8  {
9      [SyncVar(hook = nameof(OnHealthChanged))]
10     4 references
11     public int currentHealth = 100;
12     1 reference
13     public int maxHealth = 100;
14
15     2 references
16     public GameObject explosionEffect;
17     1 reference
18     public GameOverUI gameOverUI;
19
20     [Header("UI Elements")]
21     3 references
22     public TextMeshProUGUI nameText;
23     2 references
24     public Image healthBarFill; // ← Can barı için Image (Fill Amount kullanan)
25
26     [SyncVar(hook = nameof(OnNameChanged))]
27     2 references
28     public string playerName;
29
30     0 references
31     public override void OnStartClient()
32     {
33         base.OnStartClient();
34         gameOverUI = FindObjectOfType<GameOverUI>();
35         UpdateHealthBar();
36         nameText.text = playerName;
37     }
38
39 
```

Can sistemi için de kod yazıldı.

Rastgele Oyuncu İsimleri:

Servere katılan her oyuncuya “Player123” gibi farklı sayılardan oluşan rastgele oyuncu isimleri atanmasını sağlayan sistem yapıldı.

```
TankName.cs X
Assets > OnlineTankGame > Scripts > TankName.cs > ...
1 using UnityEngine;
2 using TMPro;
3 using Mirror;
Click to add a breakpoint
0 references
5 public class TankName : NetworkBehaviour
6 {
7     2 references
8     public TextMeshProUGUI nameText;
9
10    [SyncVar(hook = nameof(OnNameChanged))]
11    1 reference
12    public string playerName;
13
14    0 references
15    public override void OnStartLocalPlayer()
16    {
17        // Lokal oyuncu adını ayarla (isteğe bağlı olarak rastgele ya da sırayla da verebiliriz)
18        string newName = $"Player{Random.Range(1, 1000)}";
19        CmdSetName(newName);
20    }
21
22    [Command]
23    1 reference
24    void CmdSetName(string newName)
25    {
26        playerName = newName;
27    }
28
29    1 reference
30    void OnNameChanged(string oldName, string newName)
31    {
```

Kazanan Oyuncuyu Belirleme:

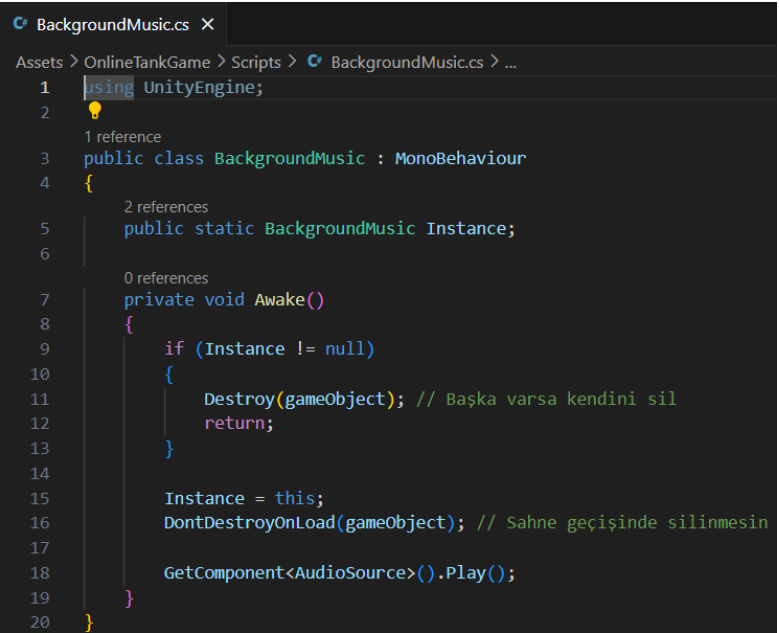
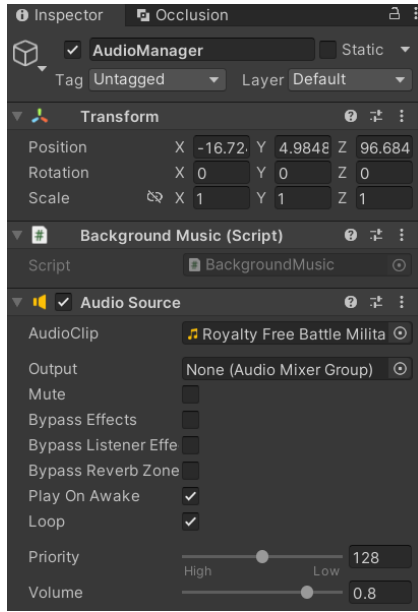


Kazanan oyuncuyu belirleyebilmek için “Game Over” sistemi yapıldı. Online oyuncuların en son hayatta kalan kazanan oyuncu olarak belirlenir ve çıkan panelde kazanan oyuncunun adı yazılır.

```
Assets > OnlineTankGame > Scripts > GameOverUI.cs > ...
1 using UnityEngine;
2 using UnityEngine.UI;
3 using TMPro;
4 using Mirror;
5
6 0 references
7 public class GameOverUI : MonoBehaviour
8 {
9     2 references
10    public GameObject panel;
11
12    1 reference
13    public TextMeshProUGUI winnerText;
14
15    4 references
16    public Button restartButton;
17
18    0 references
19    void Start()
20    {
21        panel.SetActive(false);
22        restartButton.gameObject.SetActive(false);
23        restartButton.onClick.AddListener(OnRestartClicked);
24    }
25
26    0 references
27    public void ShowWinner(string winnerName)
28    {
29        panel.SetActive(true); // Paneli aktif et
30        winnerText.text = $"Kazanan: {winnerName}";
31        restartButton.gameObject.SetActive(NetworkServer.active); // sadece host görsün
32    }
33
34    0 references
35    public void ShowRestartButton()
36    {
37        restartButton.gameObject.SetActive(true);
38    }
39 }
```

Oyunu yeniden başlatma butonu sadece Host oyuncuda gözüktür.

Oyunun Arka Fon Müziği:



Savaş atmosferini hissettirmesi için oyuna arka fon müziği eklendi. Ayrıca sahne geçişlerinde de müziğin hala çalmaya devam etmesi için kod yazıldı.

Oyunun Menüsü:



Oyunun menüsünde üst köşeye oyun kurma ve bağlanma arayüzü eklendi. Doğru ip adresini girerek oyuncular oyuna buradan başlayabilir.

