



Ad: Hilal

Soyad: Kaya

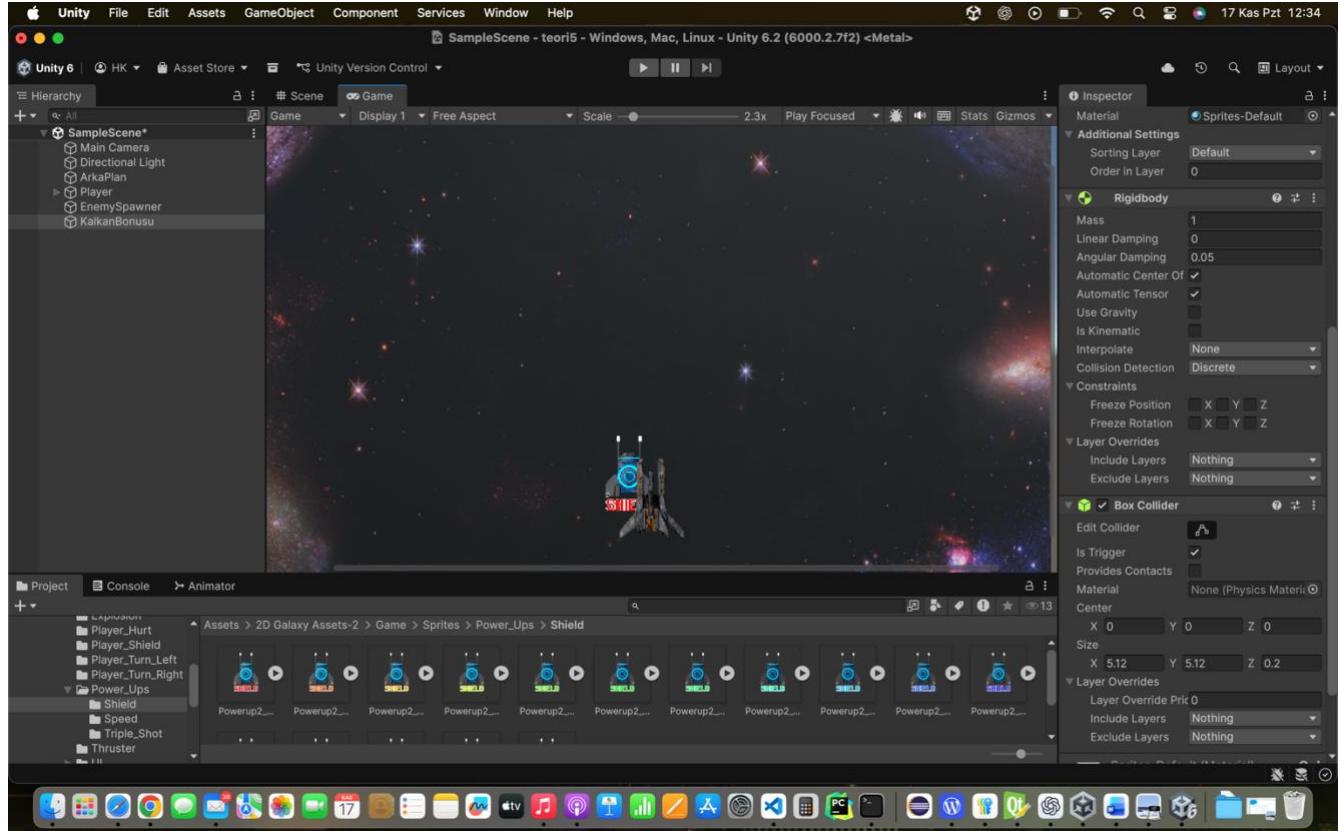
Numara: 22360859046

Ders: Oyun Programlama

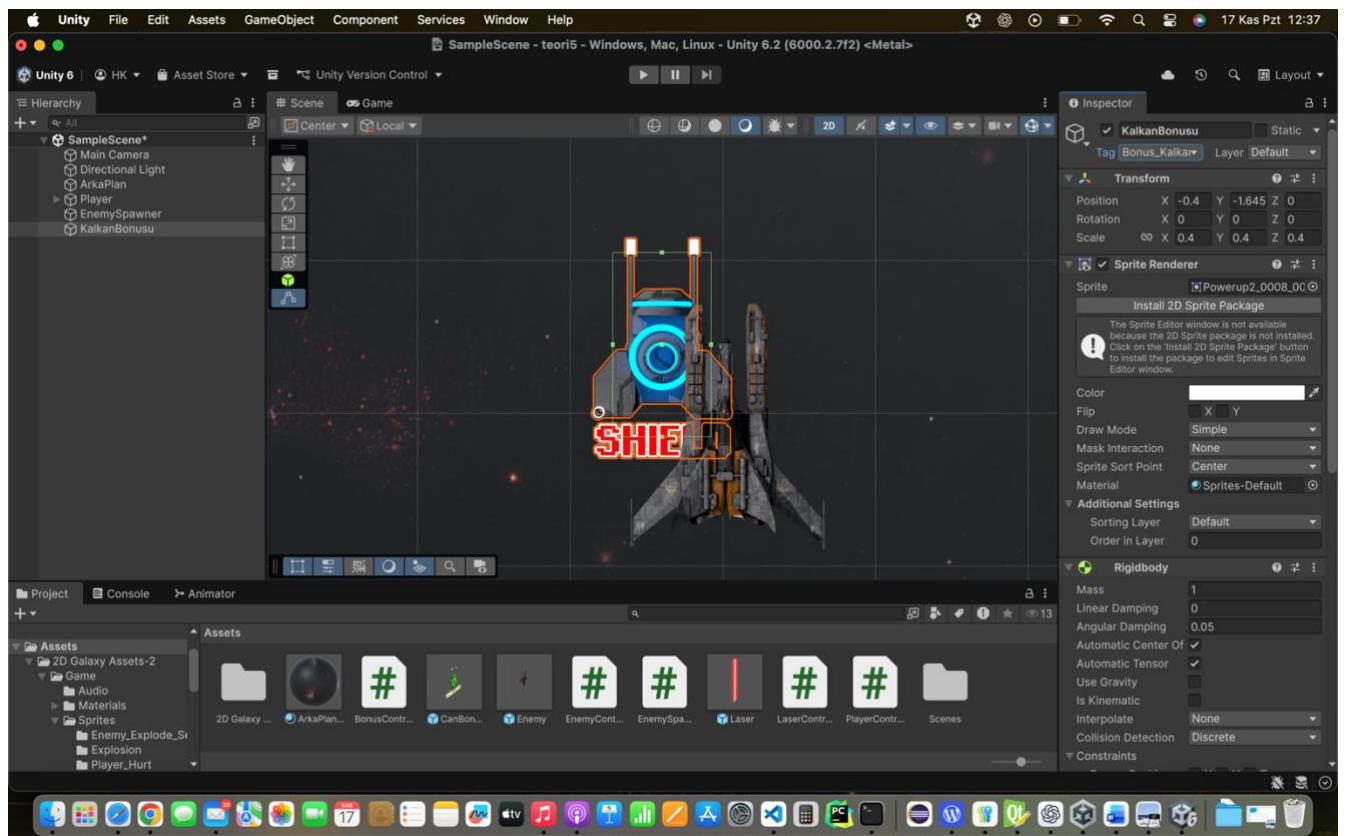
Ödev adı: Hafta 6

Geçen hafta bir uzay oyunu yapmıştık. Şimdi bu oyunu biraz daha geliştirip bazı ek özellikler ekleyeceğiz.

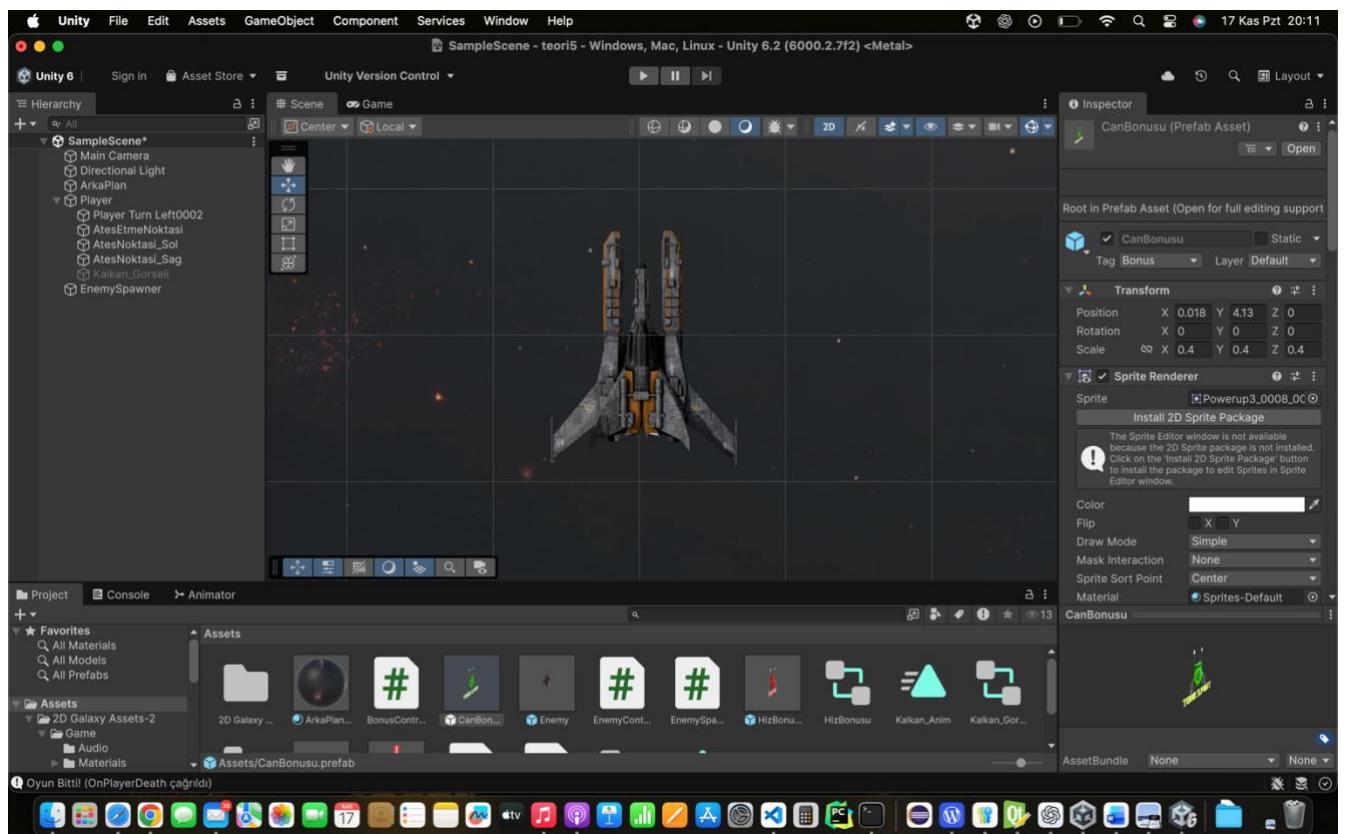
Bonus can nesneleri gibi yakalanabilen bonus kalkan nesneleri ve bonus hız nesnelerini ekleyeceğiz.



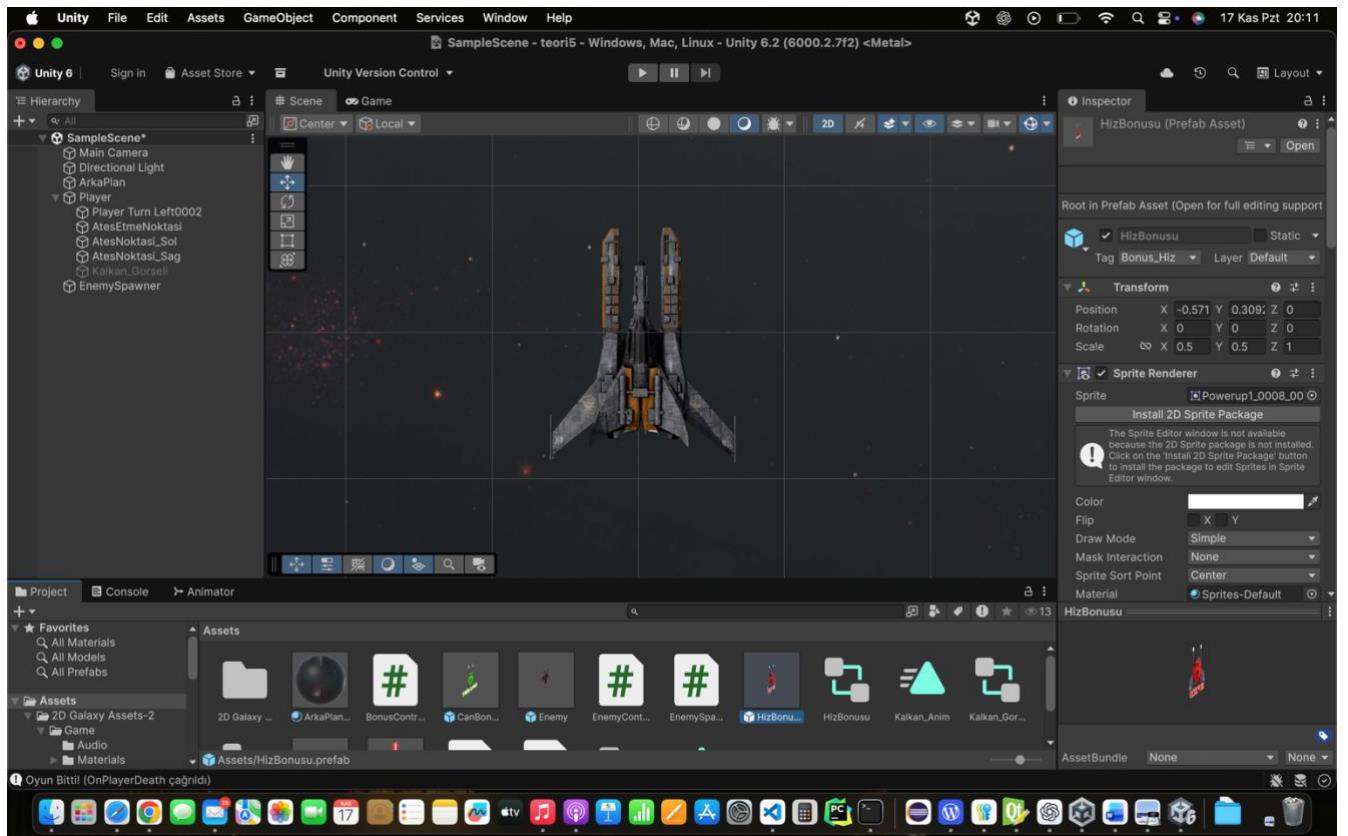
Kalkan bonusu nesnesi için istediğimiz görselleri assets klasöründen animasyona ekleyerek zaman kısa zaman aralıklarıyla farklı renklerle görünmesini sağlarız.



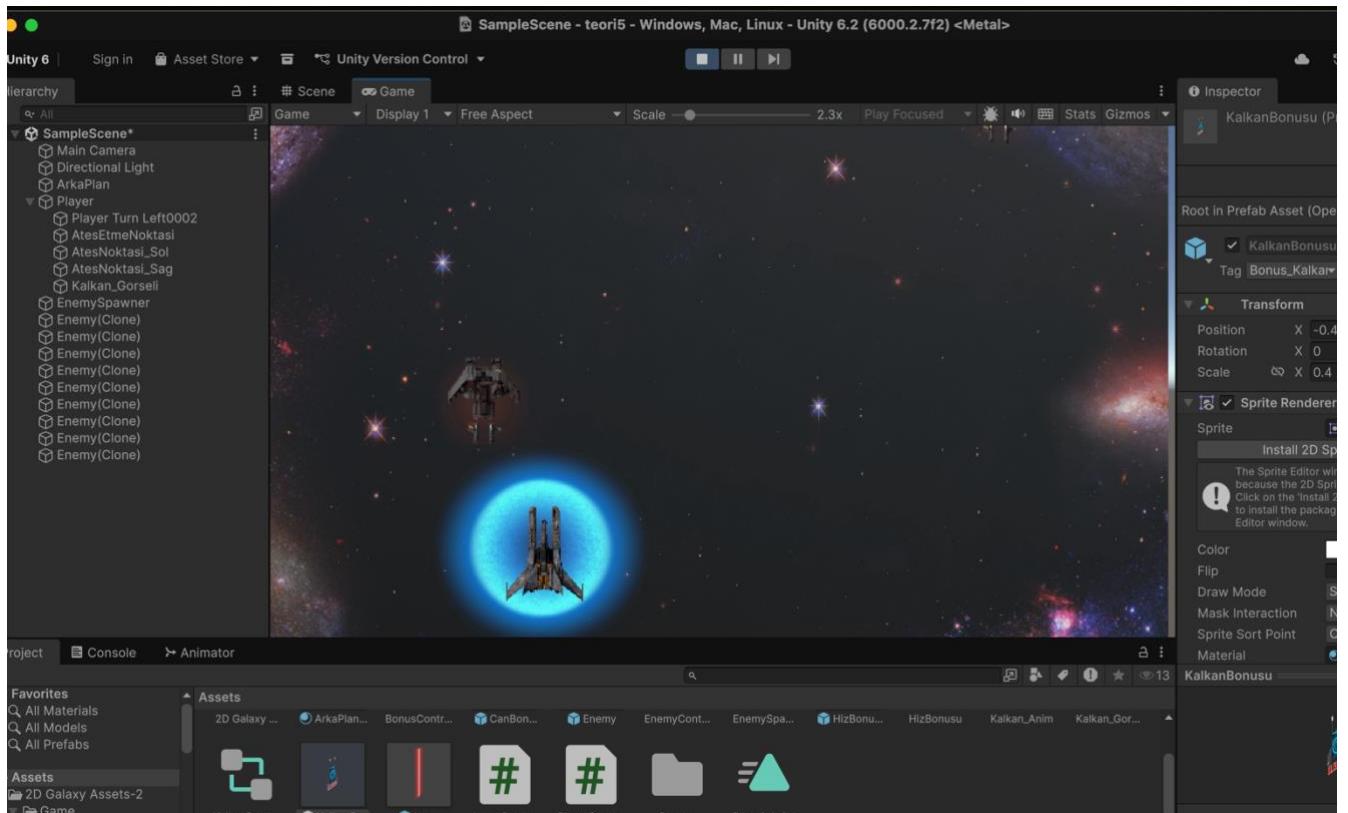
Sornasında diğer bonus nesnesine yaptığımız ayarlamaları yaparız. Rigidbody, Sprite Renderer ve Box Collider gibi komponentleri ekleriz. Sorna bonusu prefab haline getiririz.

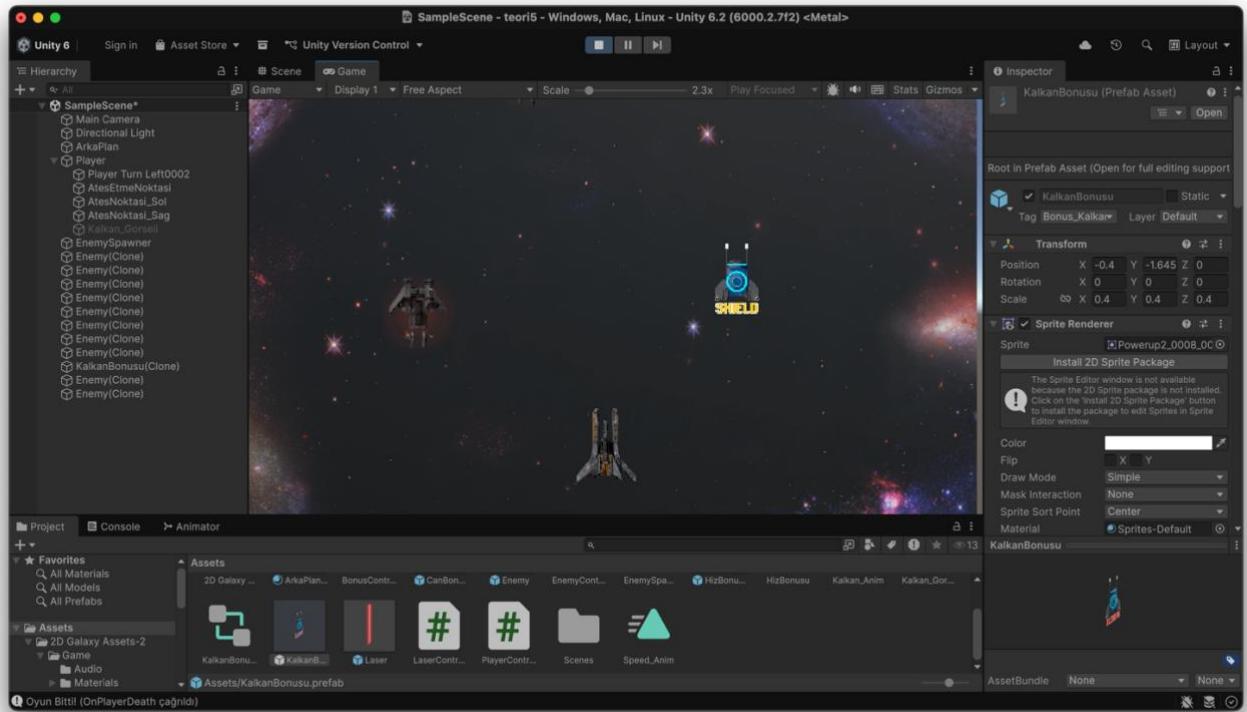


Daha sorna, kalkanı aldığında player nesnesinin etrafını sarsın diye koruma kalkanı ekleriz.



Diğer bonuslara yaptığımız tüm işlemleri hız bonusu adında oluşturduğumuz boş bir nesne olan gameObject'e de yaparız. Ve onu da diğerleri gibi prefab haline getiririz.





Oyunu başlattığımızda bunlar gibi anlarla karşılaşırız. Şimdi sırada bunun arka planında çalışan kodları inceleyeceğiz.

Bir önceki hafta kullandığımız kodların çoğunu değiştirmedigimiz için onları kısaca anlatacağım ama playerController scriptini daha yakından inceleyeceğiz.

LaserController : Lazerin yukarı yönlü hareketini ve yaşam süresini belirledik.

EnemyController : Düşman nesnelerinin aşağı yönlü hareketini sağlamak için oluşturulmuştur. Lazerle temasını da algılar.

BonusController : Bonusları da aynı düşmanlar gibi aşağı yönlü hareketi için yazılmıştır.

EnemySpawner : Oyunu sonsuz bir döngüde tutar. Ve rastgele konumlardan nesnelerin oluşmasını sağlar.

PlayerController :

The screenshot shows the Tabnine AI IDE interface. On the left is a sidebar with various icons and a "Welcome to Tabnine" section. The main area displays a code editor with the following C# code:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerController : MonoBehaviour
6 {
7     public float hareketHizi = 7f;
8     public float ekranKenarBoslugu = 1f;
9     private float minX, maxX;
10    public int can = 3;
11    public GameObject laserPrefab;
12    public Transform[] atesEtmeNoktalari;
13    public float atesEtmeAraligi = 0.5f;
14    private float atesEtmeZamani;
15
16    public GameObject kalkanGorseli;
17    private bool kalkanAktif = false;
18
19    public float hizArtisMiktari = 5f;
20    public float hizBonusSuresi = 10f;
21    private float orjinalHareketHizi;
22    private float hizBonusBitmeZamani;
23
24    void Start()
25    {
26        EkranSinirlariniHesapla();
27        Debug.Log("Oyuncu Canı: " + can);
28
29        orjinalHareketHizi = hareketHizi;
30        if (kalkanGorseli != null)
31        {
32            kalkanGorseli.SetActive(false);
33        }
34        hizBonusBitmeZamani = Time.time;
35    }
36    void EkranSinirlariniHesapla()
37    {
38        Camera kamera = Camera.main;
39        Vector3 solAlt = kamera.ViewportToWorldPoint(new Vector3(0, 0, 10));
40        Vector3 sagUst = kamera.ViewportToWorldPoint(new Vector3(1, 0, 10));
41        minX = solAlt.x + ekranKenarBoslugu;
42        maxX = sagUst.x - ekranKenarBoslugu;
43    }
44    void Update()
45    {
46        if (hareketHizi > orjinalHareketHizi && Time.time > hizBonusBitmeZamani)
47        {
48            hareketHizi = orjinalHareketHizi;
49        }
50
51        if (can <= 0) return;
52        HareketiYonet();
53        AtesEtmeiYonet();
54
55        void HareketiYonet()
56        {
57            float yatayGiris = Input.GetAxis("Horizontal");
58            Vector3 pozisyon = transform.position;
59            pozisyon.x += yatayGiris * hareketHizi * Time.deltaTime;
60            pozisyon.x = Mathf.Clamp(pozisyon.x, minX, maxX);
61            transform.position = pozisyon;
62        }
63
64        void AtesEtmeiYonet()
65        {
66            if (Input.GetKeyDown(KeyCode.Space) && Time.time > atesEtmeZamani)
67            {
68                atesEtmeZamani = Time.time + atesEtmeAraligi;
69                AtesEt();
70            }
71        }
72
73        void AtesEt()
74        {
75            foreach (Transform nokta in atesEtmeNoktalari)
76            {
77                Instantiate(laserPrefab, nokta.position, Quaternion.identity);
78            }
79        }
80
81        private void OnTriggerEnter(Collider other)
82        {
83            if (other.CompareTag("Enemy"))
84            {
85                Destroy(other.gameObject);
86            }
87        }
88    }
89}
```

At the bottom of the code editor, status bars show ".NET Ready", "Watch Off", file counts (0△0), and "BLACKBOX Agent Open Website". The status bar also indicates "Ln 42, Col 5 Spaces: 4 UTF-8 LF C#" and "BLACKBOX Autocomplete: DISABLED".

This screenshot shows the same Tabnine AI IDE interface as the first one, but with more code added to the PlayerController.cs file. The code now includes the Update() method and its associated logic for movement and shooting.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerController : MonoBehaviour
6 {
7     public float hareketHizi = 7f;
8     public float ekranKenarBoslugu = 1f;
9     private float minX, maxX;
10    public int can = 3;
11    public GameObject laserPrefab;
12    public Transform[] atesEtmeNoktalari;
13    public float atesEtmeAraligi = 0.5f;
14    private float atesEtmeZamani;
15
16    public GameObject kalkanGorseli;
17    private bool kalkanAktif = false;
18
19    public float hizArtisMiktari = 5f;
20    public float hizBonusSuresi = 10f;
21    private float orjinalHareketHizi;
22    private float hizBonusBitmeZamani;
23
24    void Start()
25    {
26        EkranSinirlariniHesapla();
27        Debug.Log("Oyuncu Canı: " + can);
28
29        orjinalHareketHizi = hareketHizi;
30        if (kalkanGorseli != null)
31        {
32            kalkanGorseli.SetActive(false);
33        }
34        hizBonusBitmeZamani = Time.time;
35    }
36    void EkranSinirlariniHesapla()
37    {
38        Camera kamera = Camera.main;
39        Vector3 solAlt = kamera.ViewportToWorldPoint(new Vector3(0, 0, 10));
40        Vector3 sagUst = kamera.ViewportToWorldPoint(new Vector3(1, 0, 10));
41        minX = solAlt.x + ekranKenarBoslugu;
42        maxX = sagUst.x - ekranKenarBoslugu;
43    }
44    void Update()
45    {
46        if (hareketHizi > orjinalHareketHizi && Time.time > hizBonusBitmeZamani)
47        {
48            hareketHizi = orjinalHareketHizi;
49        }
50
51        if (can <= 0) return;
52        HareketiYonet();
53        AtesEtmeiYonet();
54
55        void HareketiYonet()
56        {
57            float yatayGiris = Input.GetAxis("Horizontal");
58            Vector3 pozisyon = transform.position;
59            pozisyon.x += yatayGiris * hareketHizi * Time.deltaTime;
60            pozisyon.x = Mathf.Clamp(pozisyon.x, minX, maxX);
61            transform.position = pozisyon;
62        }
63
64        void AtesEtmeiYonet()
65        {
66            if (Input.GetKeyDown(KeyCode.Space) && Time.time > atesEtmeZamani)
67            {
68                atesEtmeZamani = Time.time + atesEtmeAraligi;
69                AtesEt();
70            }
71        }
72
73        void AtesEt()
74        {
75            foreach (Transform nokta in atesEtmeNoktalari)
76            {
77                Instantiate(laserPrefab, nokta.position, Quaternion.identity);
78            }
79        }
80
81        private void OnTriggerEnter(Collider other)
82        {
83            if (other.CompareTag("Enemy"))
84            {
85                Destroy(other.gameObject);
86            }
87        }
88    }
89}
```

The status bar at the bottom shows ".NET Ready", "Watch Off", file counts (0△0), and "BLACKBOX Agent Open Website". The status bar also indicates "Ln 42, Col 5 Spaces: 4 UTF-8 LF C#" and "BLACKBOX Autocomplete: DISABLED".

The screenshot shows the Tabnine AI IDE interface. On the left, there's a sidebar with various icons and sections about Tabnine's AI capabilities. The main area displays a C# script named 'PlayerController.cs' with line numbers from 92 to 134. The code handles player death logic based on different enemy types and bonus items. A preview window on the right shows a 3D game environment with a player character and some UI elements.

```
92     can--;
93     Debug.Log("Düşman çarptı! Kalan Can: " + can);
94     if (can <= 0)
95     {
96         OnPlayerDeath();
97     }
98 }
99 else if (other.CompareTag("Bonus"))
100 {
101     can++;
102     Debug.Log("Bonus alındı! Kalan Can: " + can);
103     Destroy(other.gameObject);
104 }
105 else if (other.CompareTag("Bonus_Kalkan"))
106 {
107     kalkanAktif = true;
108     if (kalkanGorseli != null)
109     {
110         kalkanGorseli.SetActive(true);
111     }
112     Destroy(other.gameObject);
113     Debug.Log("Kalkan alındı!");
114 }
115 else if (other.CompareTag("Bonus_Hiz"))
116 {
117     hareketHizi = orjinalHareketHizi + hizArtisMiktari;
118     hizBonusBitmeZamani = Time.time + hizBonusSuresi;
119     Destroy(other.gameObject);
120     Debug.Log("Hız bonusu alındı!");
121 }
122 }
123 private void OnPlayerDeath()
124 {
125     Debug.Log("Oyun Bitti! (OnPlayerDeath çağrıldı)");
126     SpriteRenderer sr = GetComponent<SpriteRenderer>();
127     if (sr != null)
128     {
129         sr.enabled = false;
130     }
131     Collider col = GetComponent<Collider>();
132     if (col != null)
133     {
134         col.enabled = false;
135     }
136 }
```

Oyuncunun yatay hareketini (Input.GetAxis("Horizontal")), ekranda kalma sınırlarını (Mathf.Clamp), atış etmesini (Input.GetKeyDown(KeyCode.Space)), canını (OnPlayerDeath() Fonksiyonu), bonuslarla etkileşimini (other.CompareTag(...)) ile etiketini kontrol ederiz) yönetir.

Github Linki : <https://github.com/hilal-kaya/OyunProgramlama/blob/main/Teo6.pdf>