**KAYNAK KOD**

**Player\_Script.cs :** Dokunmatik mekanizma, puanlama sistemi, can geri sayım mekanizması bulunduran script dosyamız.

private Rigidbody2D Maymun;

public float move\_Speed = 2f;

public float normal\_Itme = 10f;

public float extra\_Itme = 14f;

private bool initial\_Push;

private int Itme\_Sayisi;

private bool player\_Oldu;

public GameObject kazandin\_pnl;

public GameObject kaybettin\_pnl;

public AudioSource bg\_music;

public AudioSource kazandin\_music\_pnl;

public AudioSource kaybettin\_music\_pnl;

public TextMeshProUGUI hatali\_txt;

public int can = 10;

public TextMeshProUGUI Puan\_txt;

public int puan = 0;

void Awake()//ilk çalışacak kod

{

Maymun = GetComponent<Rigidbody2D>();//fizik işlemleri için

can = 10;

hatali\_txt.text = "X : " + can;

puan = 0;

Puan\_txt.text = "Puan : " + puan + " /100";

}

// Update is called once per frame

void FixedUpdate()

{

Move();

}

void Move()// hareket methodu

{

if (player\_Oldu)

return;

if (SimpleInput.GetAxisRaw("Horizontal") > 0)//hareket sıfırdan büyükse

{

Maymun.velocity = new Vector2(move\_Speed, Maymun.velocity.y);

}

else if (SimpleInput.GetAxisRaw("Horizontal") < 0)//hareket sıfırdan küçük ise

{

Maymun.velocity = new Vector2(-move\_Speed, Maymun.velocity.y);

}

} // player hareketi

void OnTriggerEnter2D(Collider2D target)

{

if (player\_Oldu)// maymun kaybederse return ile method ignore et

return;

if (target.tag == "extra\_Itme")

{

if (!initial\_Push)

{

initial\_Push = true;

Maymun.velocity = new Vector2(Maymun.velocity.x, 18f);

target.gameObject.SetActive(false);

Ses\_Yonetim.instance.JumpSoundFX();

// ilk itme nedeniyle trigger girişinden çıkış

return;

} // ilk itme

// ilk itme blok sonu

}

if (target.tag == "NormalItme")

{

puan += 3;

Puan\_txt.text = "Puan : " + puan + " /100";

Maymun.velocity = new Vector2(Maymun.velocity.x, normal\_Itme);

target.gameObject.SetActive(false);

Itme\_Sayisi++;

Ses\_Yonetim.instance.JumpSoundFX();

}

if (target.tag == "extra\_Itme")

{

puan += 5;

Puan\_txt.text = "Puan : " + puan + " /100";

Maymun.velocity = new Vector2(Maymun.velocity.x, extra\_Itme);

target.gameObject.SetActive(false);

Itme\_Sayisi++;

Ses\_Yonetim.instance.JumpSoundFX();//zıplama sesi

}

if (Itme\_Sayisi == 2)

{

Itme\_Sayisi = 0;

PlatformSpawner.instance.SpawnPlatforms();

}

//can konrolü

if (target.tag == "Yanlis\_Kutu")//

{

can--;

hatali\_txt.text = "X : " + can;

}

if ( target.tag == "Yanlis\_Kutu")

{

Ses\_Yonetim.instance.GameOverSoundFX();

if (can==0)

{

player\_Oldu = true;//maymun düştü

kaybettin\_pnl.SetActive(true);

bg\_music.Stop();

kaybettin\_music\_pnl.Play();

//instance.RestartGame();

}

}

if (target.tag == "MaymunDustu" )// maymun düşerse hemen ölsün

{

Ses\_Yonetim.instance.GameOverSoundFX();

player\_Oldu = true;//maymun düştü

kaybettin\_pnl.SetActive(true);

bg\_music.Stop();

kaybettin\_music\_pnl.Play();

//Yonetim.instance.RestartGame();

}

if (puan>=100)//(target.tag == "Kazandın" || target.tag == "Puan kontrol")

{

kazandin\_pnl.SetActive(true);

Time.timeScale = 0;

bg\_music.Stop();

kazandin\_music\_pnl.Play();

}

} // on trigger enter

**Camera\_Takip.cs:** maymun karakterini takip işlemini sağlayan script dosyamız.

private Transform hedef;

private bool TakipPlayer;

public float min\_Y\_Esik = -2.6f;

void Awake()

{

hedef = GameObject.FindGameObjectWithTag("Player").transform;

}

// Update is called once per frame

void Update()

{

Follow();

}

void Follow()

{

if (hedef.position.y < (transform.position.y - min\_Y\_Esik))

{

TakipPlayer = false;

}

if (hedef.position.y > transform.position.y)

{

TakipPlayer = true;

}

if (TakipPlayer)

{

Vector3 temp = transform.position;

temp.y = hedef.position.y;

transform.position = temp;

}

}

**Ses\_Yonetim.cs :**  sahnedeki ses dosyalarının yönetim işlemini yapan script dosyamız.

public static Ses\_Yonetim instance;

[SerializeField]

private AudioSource soundFX;

[SerializeField]

private AudioClip jumpClip, gameOverClip;

void Awake()

{

if (instance == null)

instance = this;

}

public void JumpSoundFX()

{

soundFX.clip = jumpClip;

soundFX.Play();

}

public void GameOverSoundFX()

{

soundFX.clip = gameOverClip;

soundFX.Play();

}

**Yonetim.cs :** sahne içerisinde genel yönetimi, ses, animasyon, panel işemleri, geri sayım, puanlama gibi işlemleri bu script dosyası üzerinden kontrol ettik.

public static Yonetim instance;

public GameObject Bilgi\_panel;

public AudioSource bg\_music;

public AudioSource bg\_music\_pnl;

public AudioSource kazandin\_music\_pnl;

public AudioSource kaybettin\_music\_pnl;

//sahne geçişleri bu scriptten kontrol edilecek

void Awake()

{

if (instance == null)

instance = this;

if (SceneManager.GetActiveScene().buildIndex!=0)

{

Time.timeScale = 0;

}

else if (SceneManager.GetActiveScene().buildIndex == 0)

{

Time.timeScale = 1;

}

}

public void RestartGame()

{

Invoke("LoadGameplay", 0.0f);

}

void LoadGameplay()

{

UnityEngine.SceneManagement.SceneManager.LoadScene(UnityEngine.SceneManagement.SceneManager.GetActiveScene().buildIndex );

}

public void Next\_Game()

{

Invoke("NextGame", 0.0f);

}

void NextGame()

{

UnityEngine.SceneManagement.SceneManager.LoadScene(UnityEngine.SceneManagement.SceneManager.GetActiveScene().buildIndex + 1);//aktif sahne indexine +1 ekle

}

public void Exit\_Game()

{

Invoke("Exit", 0.0f);

}

void Exit()

{

Application.Quit();//oyunu sonlandır.

}

public void Main\_Menu()// olmasada olur

{

Invoke("MainMenu",0.0f);

}

void MainMenu()

{

UnityEngine.SceneManagement.SceneManager.LoadScene(0);//Ana menuye geri dön

Time.timeScale = 1;

}

public void Oyna()

{

bg\_music\_pnl.Stop();

Invoke("Play", 0.0f);

bg\_music.Play();

}

void Play()

{

Time.timeScale = 1;

Bilgi\_panel.SetActive (false);

}

**BGSpawner.cs :** arka plan resminin sürekliliğini sağlayan script dosyamız.

private GameObject[] bgs;//sürekli arka plan için

private float height;

private float highest\_Y\_Pos;

void Awake()

{

bgs = GameObject.FindGameObjectsWithTag("BG");//BG tagına sahip abjeleri alır

}

void Start()

{

height = bgs[0].GetComponent<BoxCollider2D>().bounds.size.y;//arka plan resminin yüksekliğini ata

highest\_Y\_Pos = bgs[0].transform.position.y;

for (int i = 1; i < bgs.Length; i++)

{

if (bgs[i].transform.position.y > highest\_Y\_Pos)

highest\_Y\_Pos = bgs[i].transform.position.y;

}

}

void OnTriggerEnter2D(Collider2D target)

{

if (target.tag == "BG")

{

// gidebileceği en yüksek noktayı belirle

if (target.transform.position.y >= highest\_Y\_Pos)

{

Vector3 temp = target.transform.position;

for (int i = 0; i < bgs.Length; i++)

{

if (!bgs[i].activeInHierarchy)

{

temp.y += height;

bgs[i].transform.position = temp;

bgs[i].gameObject.SetActive(true);//kamera yukarı çıktıktça setactivi true yap

highest\_Y\_Pos = temp.y;

}

}

}

}

} // on trigger enter

**Spawn\_collector.cs :** yeni sahne oluştuğunda belleği yormamak için eski sahne ve içindeki objeleri pasif yapan script dosyamız.

void OnTriggerEnter2D(Collider2D target)

{

if (target.tag == "BG" || target.tag == "Platform" ||

target.tag == "NormalItme" || target.tag == "ExtraItme" ||

target.tag == "Yanlis\_Kutu")

{

target.gameObject.SetActive(false);

}

} // on trigger

**Uretec.cs :** sayı oluşturma ve yuvarlama işlemini yapan script dosyamız.

int sayi;

// Start is called before the first frame update

void Awake ()

{

// Time.timeScale = 1;

sayi = Uret();

Debug.Log(sayi);

Debug.Log((sayi/100)\*100);

Debug.Log(onluk\_Yuvarla(sayi));

Debug.Log(yuzluk\_Yuvarla(sayi));

Debug.Log(binlik\_Yuvarla(sayi));

//yuzluk\_txt.text = yuzluk\_Yuvarla(sayi).ToString();

}

// Update is called once per frame

void Update()

{

}

public int Uret()

{

System.Random random = new System.Random();

int randomValue = random.Next (100, 5000);//random.Next(1000) \* (max - min) + min;

return randomValue;

}

static int yuzluk\_Yuvarla(int sayi)// enyakın yüzlüğe yuvarla

{

return ((int)Math.Round(sayi / 100.0)) \* 100;

}

static int onluk\_Yuvarla(int sayi)// enyakın onluğa yuvarla

{

return ((int)Math.Round(sayi / 10.0)) \* 10;

}

static int binlik\_Yuvarla(int sayi)// enyakın binliğe yuvarla

{

return ((int)Math.Round(sayi / 1000.0)) \* 1000;

}

**PlatformScript.cs :** platform üzerinde rastgele sayı ve meyve spawn işlemini yapan script dosyamız.

[SerializeField]

private GameObject one\_Banana, bananas;

[SerializeField]

private Transform spawn\_Point;

void Start()

{

GameObject newBanana = null;

if (Random.Range(0, 10) > 3)

{

newBanana = Instantiate(one\_Banana, spawn\_Point.position, Quaternion.identity);

}

else

{

newBanana = Instantiate(bananas, spawn\_Point.position, Quaternion.identity);

}

newBanana.transform.parent = transform;

}

**PlatformSpawner.cs :** belli konumlar arasında rastgele olarak platform ve kuş prefablarını spawn eden script dosyamız.

public static PlatformSpawner instance;

[SerializeField]

private GameObject left\_Platform, right\_Platform;

private float sag\_X\_Min = -4.26f, sag\_X\_Max = -2.89f, sol\_X\_Min = 4.26f, sol\_X\_Max = 2.78f;//platformun x spawn koordinatları

private float y\_Treshold = 2.6f;//platformun y spawn koordinatları

private float last\_Y;

public int spawn\_Count = 16;

private int platform\_Spawned;

[SerializeField]

private Transform platform\_Parent;

// more variables to spawn bird enemy

[SerializeField]

private GameObject bird;

public float bird\_Y = 5f;

private float bird\_X\_Min = -2.3f, bird\_X\_Max = 2.3f;

void Awake()

{

if (instance == null)

instance = this;

}

void Start()

{

last\_Y = transform.position.y;

SpawnPlatforms();

}

public void SpawnPlatforms()

{

Vector2 temp = Vector2.zero;

GameObject newPlatform = null;

for (int i = 0; i <spawn\_Count; i++)

{

temp.y = last\_Y;

// we have even number

if ((platform\_Spawned % 2) == 0)

{

temp.x = Random.Range(sag\_X\_Min, sag\_X\_Max);//sağ spawn aralıkları rastgele

newPlatform = Instantiate(right\_Platform, temp, Quaternion.identity);

}

else

{

// if we have odd number

temp.x = Random.Range(sol\_X\_Min, sol\_X\_Max);// sol spawn aralıkları rastgele

newPlatform = Instantiate(left\_Platform, temp, Quaternion.identity);

}

newPlatform.transform.parent = platform\_Parent;

last\_Y += y\_Treshold;

platform\_Spawned++;

}

if (Random.Range(0, 2) > 0)//%50 ihitmalle hatalı kutu açılacak

{

SpawnBird();

}

} // spawn platforms

void SpawnBird()

{

Vector2 temp = transform.position;

temp.x = Random.Range(bird\_X\_Min, bird\_X\_Max);

temp.y += bird\_Y;

GameObject newBird = Instantiate(bird, temp, Quaternion.identity);

newBird.transform.parent = platform\_Parent;

}