**Proiect TM**

**Flappy Bird**

**Studenti: Muresan Marius**

**Pirlac Sebastian**

**Grupa: 2531, anul III**

**Prof: Ana Maria Ruscanu**

**Tema aleasa:**

Flappy Bird este un joc video simplu și captivant dezvoltat de programatorul vietnamez Dong Nguyen. Jocul a fost lansat în mai 2013 pentru platformele iOS și Android si a devenit extrem de popular intr-un timp foarte scurt.

Scopul jocului este sa controlezi o mica pasare si sa o ajuți sa zboare printre niste conducte verzi. Pasarea zboara in mod automat, iar jucatorul trebuie sa apese un buton sau sa faca click pentru a-l menține in aer. Fiecare trecere printre conducte valoreaza un punct si obiectivul este sa obtii cat mai multe puncte posibile.

Flappy Bird a devenit extrem de popular datorita dificultatii sale ridicate și a gameplay-ului simplu, dar captivant. Jocul a devenit viral pe retelele sociale si mulți jucatori au fost captivati de provocarea de a obține un scor cat mai mare. Cu toate acestea, popularitatea rapida a jocului a dus si la unele controverse si critici legate de nivelul de dificultate si de impactul negativ asupra sanatatii mintale a jucatorilor.

**Scripturi:**

**Player.cs**

using UnityEngine;

public class Player : MonoBehaviour

{

    private SpriteRenderer spriteRenderer;

    public Sprite[] sprites;

    private int spriteIndex;

    public float strength = 5f;

    public float gravity = -9.81f;

    public float tilt = 5f;

    private Vector3 direction;

    private void Awake()

    {

        spriteRenderer = GetComponent<SpriteRenderer>();

    }

    private void Start()

    {

        InvokeRepeating(nameof(AnimateSprite), 0.15f, 0.15f);

    }

    private void OnEnable()

    {

        Vector3 position = transform.position;

        position.y = 0f;

        transform.position = position;

        direction = Vector3.zero;

    }

    private void Update()

    {

*if* (Input.GetKeyDown(KeyCode.Space) || Input.GetMouseButtonDown(0)) {

            direction = Vector3.up \* strength;

        }

*// Apply gravity and update the position*

        direction.y += gravity \* Time.deltaTime;

        transform.position += direction \* Time.deltaTime;

*// Tilt the bird based on the direction*

        Vector3 rotation = transform.eulerAngles;

        rotation.z = direction.y \* tilt;

        transform.eulerAngles = rotation;

    }

    private void AnimateSprite()

    {

        spriteIndex++;

*if* (spriteIndex >= sprites.Length) {

            spriteIndex = 0;

        }

*if* (spriteIndex < sprites.Length && spriteIndex >= 0) {

            spriteRenderer.sprite = sprites[spriteIndex];

        }

    }

    private void OnTriggerEnter2D(Collider2D other)

    {

*if* (other.gameObject.CompareTag("Obstacle")) {

            FindObjectOfType<GameManager>().GameOver();

        } *else* *if* (other.gameObject.CompareTag("Scoring")) {

            FindObjectOfType<GameManager>().IncreaseScore();

        }

**Player.cs :**

* Pune pasarea in miscare
* Seteaza gravitatia si miscarea pasarii
* La apasarea butonului SPACE sau click Stanga pasarea o sa inceapa sa “zboare”, dupa care va trebui sa fie mentinuta prin apasarea acelorasi butoane.
* Face animatia pasarii mai fluida

**Parallax.cs:**

using UnityEngine;

public class Parallax : MonoBehaviour

{

    private MeshRenderer meshRenderer;

    public float animationSpeed = 1f;

    private void Awake()

    {

        meshRenderer = GetComponent<MeshRenderer>();

    }

    private void Update()

    {

        meshRenderer.material.mainTextureOffset += new Vector2(animationSpeed \* Time.deltaTime, 0);

    }

}

**Parallax:**

* Folosit pentru animatia vitezei Background-ului si a Ground-ului
* Background-ul se misca mai incet
* Ground-ul se misca mai repede

**Spawner.cs:**

using UnityEngine;

public class Spawner : MonoBehaviour

{

    public GameObject prefab;

    public float spawnRate = 1f;

    public float minHeight = -1f;

    public float maxHeight = 2f;

    private void OnEnable()

    {

        InvokeRepeating(nameof(Spawn), spawnRate, spawnRate);

    }

    private void OnDisable()

    {

        CancelInvoke(nameof(Spawn));

    }

    private void Spawn()

    {

        GameObject pipes = Instantiate(prefab, transform.position, Quaternion.identity);

        pipes.transform.position += Vector3.up \* Random.Range(minHeight, maxHeight);

    }

}

**Spawner:**

* Folosit pentru spawnarea random a conductelor
* Valorile pentru fiecare conducta spawnata se aleg random dintr-un interval (minim, maxim)

**Pipes.cs:**

using UnityEngine;

public class Pipes : MonoBehaviour

{

    public Transform top;

    public Transform bottom;

    public float speed = 5f;

    private float leftEdge;

    private void Start()

    {

        leftEdge = Camera.main.ScreenToWorldPoint(Vector3.zero).x - 1f;

    }

    private void Update()

    {

        transform.position += Vector3.left \* speed \* Time.deltaTime;

*if* (transform.position.x < leftEdge) {

            Destroy(gameObject);

        }

    }

}

**Pipes:**

* Folosit pentru miscarea si pozitionarea conductelor
* Folosit pentru distrugerea conductelor dupa ce pasarea a trecut de ele (raman in spate)

**GameManager.cs**

using UnityEngine;

using UnityEngine.UI;

public class GameManager : MonoBehaviour

{

    private Player player;

    private Spawner spawner;

    public Text scoreText;

    public GameObject playButton;

    public GameObject gameOver;

    public int score { get; private set; }

    private void Awake()

    {

        Application.targetFrameRate = 60;

        player = FindObjectOfType<Player>();

        spawner = FindObjectOfType<Spawner>();

        Pause();

    }

    public void Play()

    {

        score = 0;

        scoreText.text = score.ToString();

        playButton.SetActive(false);

        gameOver.SetActive(false);

        Time.timeScale = 1f;

        player.enabled = true;

        Pipes[] pipes = FindObjectsOfType<Pipes>();

*for* (int i = 0; i < pipes.Length; i++) {

            Destroy(pipes[i].gameObject);

        }

    }

    public void GameOver()

    {

        playButton.SetActive(true);

        gameOver.SetActive(true);

        Pause();

    }

    public void Pause()

    {

        Time.timeScale = 0f;

        player.enabled = false;

    }

    public void IncreaseScore()

    {

        score++;

        scoreText.text = score.ToString();

    }

}

**GameManager:**

* Folosit pentru inregistrarea scorului
* Folosit pentru terminarea jocului (Game Over)
* Folosit pentru apasarea butonului Play
* Folosit pentru manageriatul jocului

**Scena:**

A screenshot of a video game

Description automatically generated

A screenshot of a video game

Description automatically generated

**Referinte:**

<https://www.youtube.com/watch?v=ihvBiJ1oC9U&list=WL&index=146>

<https://softuni.org/project-tutorials/how-to-make-our-own-flappy-bird-in-unity/>

<https://www.youtube.com/watch?v=WBlJs02sTYk>

<https://nakov.com/blog/2022/02/09/how-to-make-flappy-bird-game-in-unity-and-csharp/>