



UNIVERSITY OF ORADEA

Multimedia Technologies Project

Get There!

Coordinator:

Assoc. dr.ing. Ovidiu Constantin Novac

Student:

Mario Vázquez Pérez

Requirements.....	2
Project Description.....	2
Used technologies.....	4
- Unity.....	4
- Visual Studio 2022.....	4
- C# (CSharp).....	5
Implementation.....	6
Scripts.....	7
GameManager.....	7
AudioManager.....	10
ButtonManager.....	12
Ball.....	13
Coins.....	13
Sound.....	14
VolumeSettings.....	14
MainMenu.....	16
PauseMenu.....	17
CameraControl.....	17
Conclusions.....	18
Bibliography.....	19

Requirements

To carry out this project, a video game created on the Unity platform was required, which had a free theme, so each student was given imaginative freedom to carry out the project that they most wanted.

Project Description

This 3D game is called “Get There!” and it has been developed using the Unity platform and C# programming language. The game consists of achieving the 15 coins sprayed on the map before the time is up while you are controlling a ball.

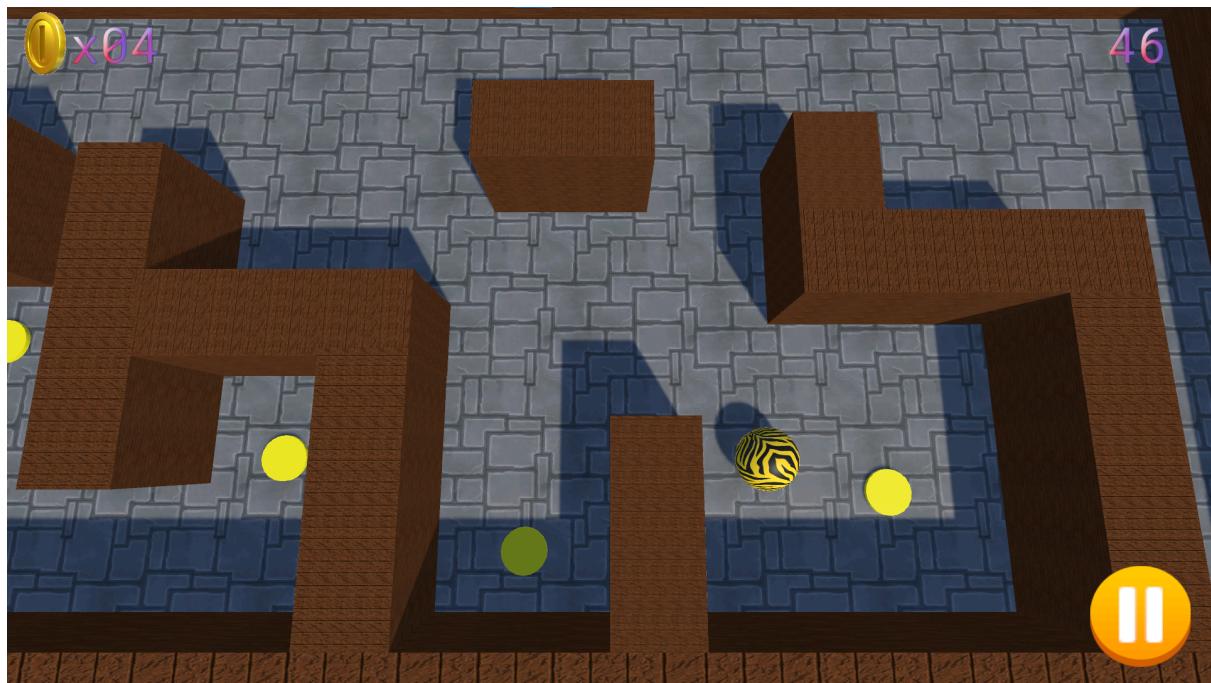


Fig 1. View of the game

In the upper part of the screen, the count of coins and the timer are visible. In the right bottom corner, also the pause button is visible, so the game can be stopped to make some quick adjustment in the settings or to exit to the main menu.

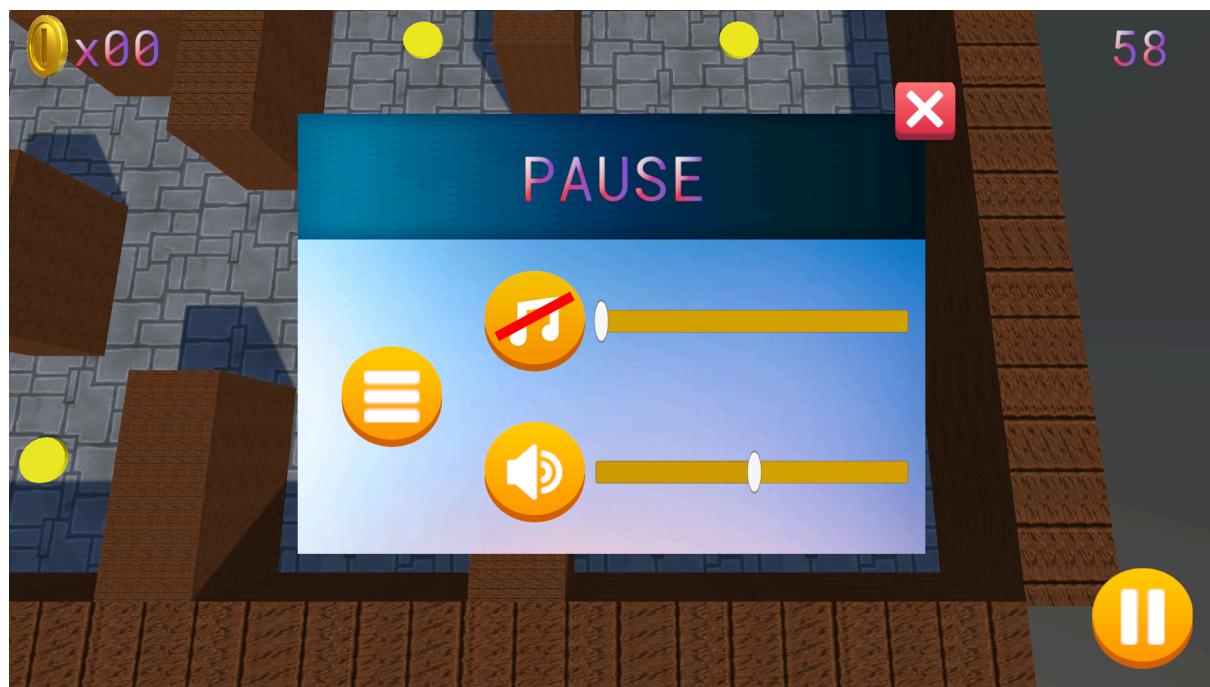


Fig 2. Pause menu

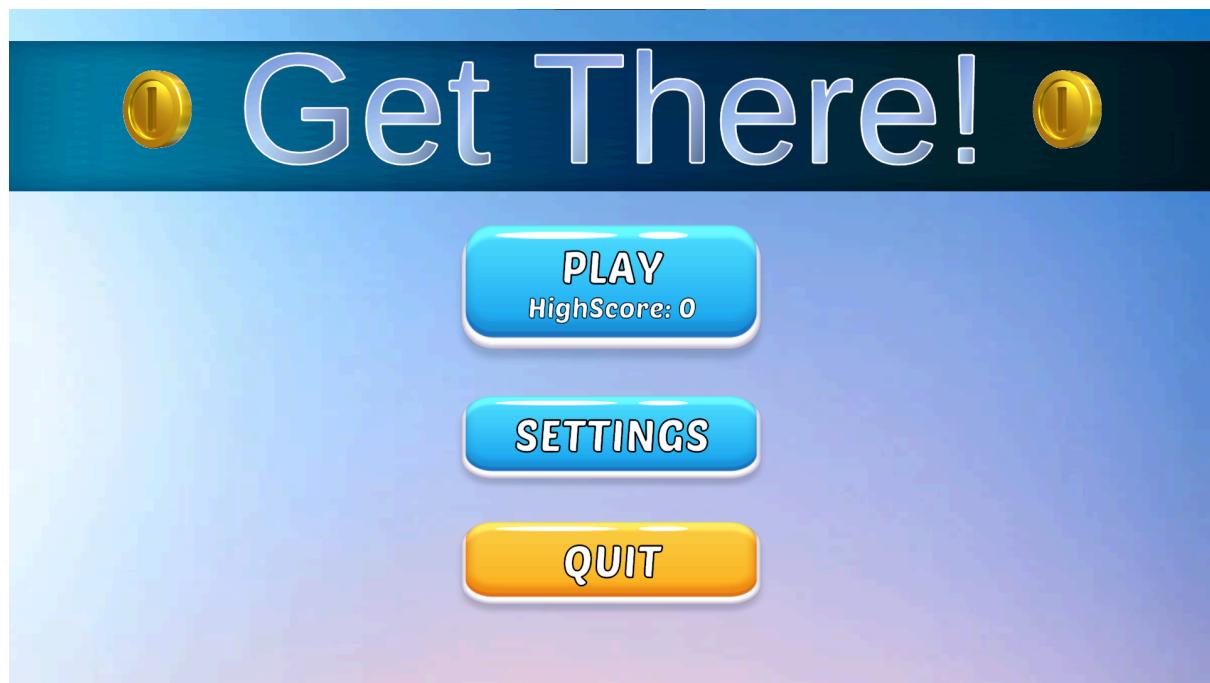


Fig 3. Main menu

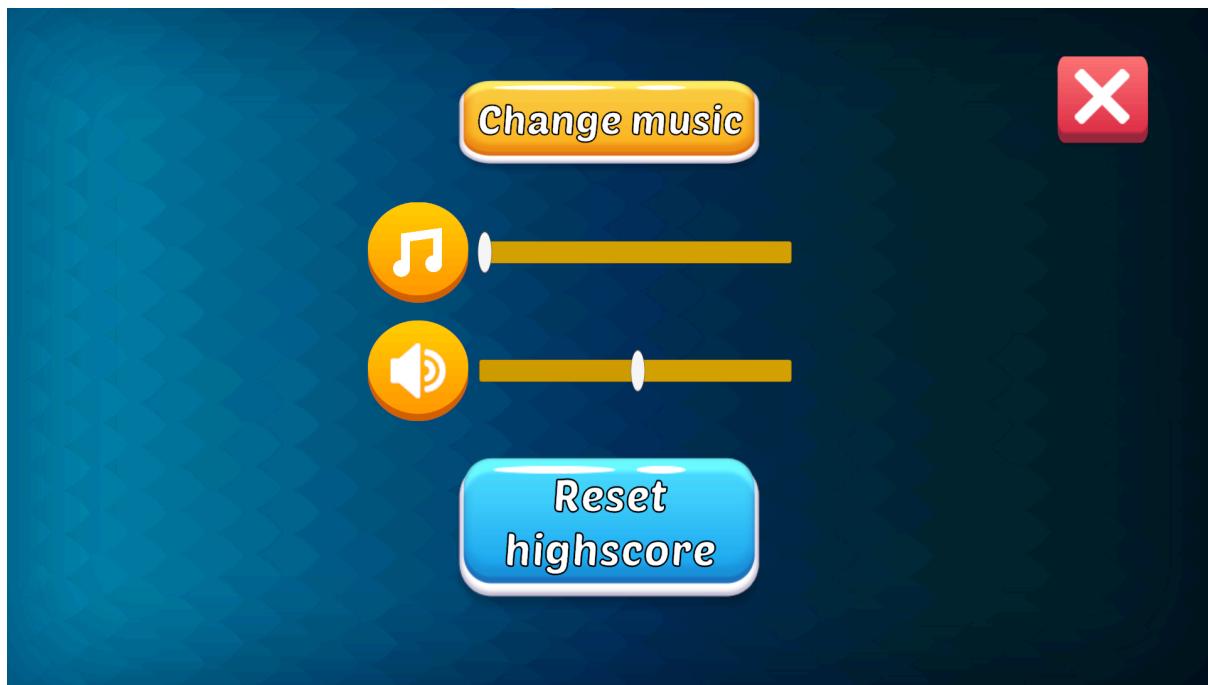


Fig 4. Settings menu

Used technologies

For the realization of the project, the following technologies were used:

- Unity

Unity is a cross-platform game engine developed by Unity Technologies, released in 2005. It supports several desktop, mobile and console platforms and it can be used to create 2D and 3D games, making this tool very useful for video games development. [1]



Fig 5. Unity logo [2]

- Visual Studio 2022

Visual Studio is an IDE developed by Microsoft. It's used to write scripts in some programming languages for developing websites, apps and so on.

It includes a code editor supporting IntelliSense, which is a code completion component, a very helpful tool for programming. The integrated debugger works as both a source-level debugger and as a machine-level debugger. It accepts plugins enabling users to reach new functionalities. [3]

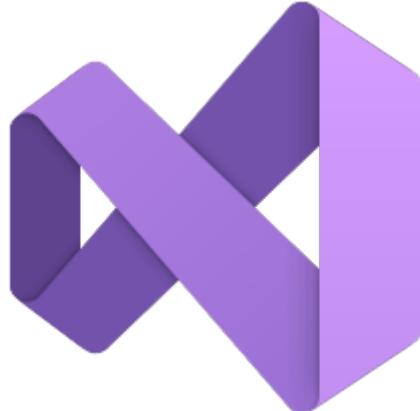


Fig 6. Visual Studio 2022 logo [4]

- C# (CSharp)

C Sharp is a high level programming language that appeared in 2000. Its biggest advantages are that it includes static typing, strong typing, imperative, declarative, object oriented (class-based) and so on. Windows introduced it with .NET Framework and Visual Studio.

This language is very versatile and allows us to manage every object and event occurring in the game. [5]

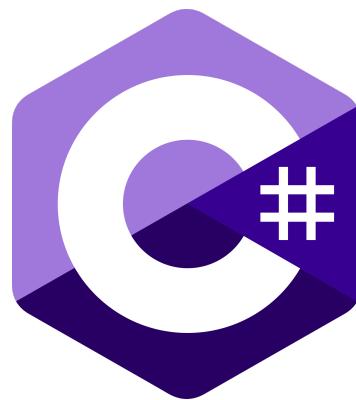


Fig 7. C# logo [6]

Some other websites were also accessed during the development of the game, such as Youtube to search for some video tutorials [7], Jamendo to download some music for the game [8], the Unity Asset Store to use some assets in the project [9] and some images extracted from Internet also [10].

Implementation

In Unity, C# scripts can be easily created and assigned to objects in order to invoke functions or events handling them. For the implementation of this game, eleven scripts were created and used.

- GameManager: This script is the one in charge of managing the events happening during the game, such as updating the timer or showing the score.
- Audio Manager: This is similar to the GameManager, but it is only handling the audio events, such as the volume of the music or the coin sounds.
- ButtonManager: This script was created to manage the “shuffle track” option button, which was very problematic due to the change of scenes.
- Ball: In this script, the movements of the ball are controlled using the predefined axis (“Vertical” and “Horizontal”) and the Rigidbody component to manage the collisions with the walls.



Fig 8. Ball script assigned to the Sphere object

- Coins: In order, to destroy the coin objects when the ball touches them, the sphere has a “Player” tag. In this script, when the tag is detected in the coin, it is destroyed and the coin sound is played, as well as the coin count is updated.
- Sound: This script is very simple and is Serializable, which means that it is only created in order to assign an object of this type in the inspector of a game object.
- VolumeSettings: In this script, there are several functions related with the play, pause, mute and unmute of the music and SFX.

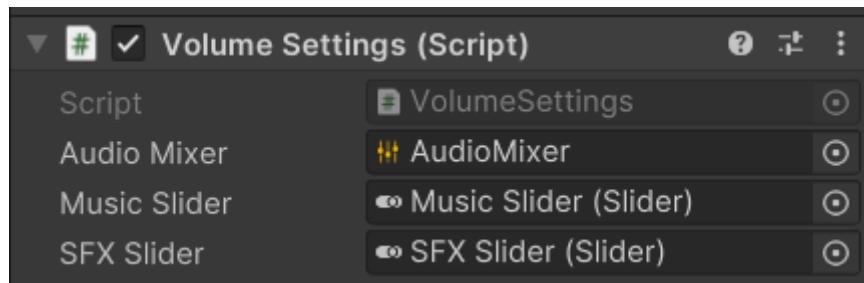


Fig 9. VolumeSettings script assigned to the Pause Menu

- MainMenu: This script has some functions to change between scenes or just quit the game.

- PauseMenu: As well as the previous script, it has some functions related with the options selection, but in the pause menu, meaning that the game has to stop, which is a more complex behavior.
- CameraControl: This script makes the main camera follow the target player when it is moving all around the map.

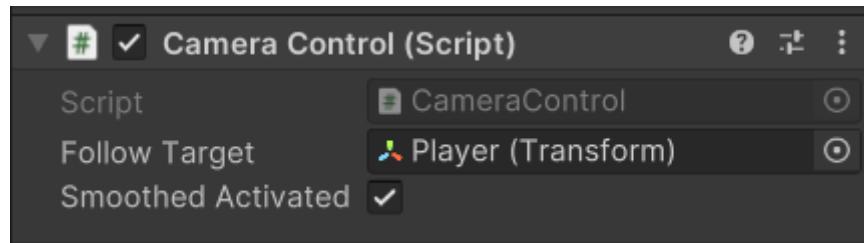


Fig 10. CameraControl script with the Player as Follow Target

Scripts

GameManager

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using UnityEngine.UI;

public class GameManager : MonoBehaviour
{
    [SerializeField]
    private TextMeshProUGUI coinsText;
    [SerializeField]
    private TextMeshProUGUI timeText;
    [SerializeField]
    private TextMeshProUGUI finalText;
    [SerializeField]
    private TextMeshProUGUI scoreText;
    [SerializeField]
    private TextMeshProUGUI highScoreText;
    [SerializeField]
    private Button button;

    private float time = 60.99f;
    private int coins = 0;
    private int score = 0;
```

```

private void Start()
{
    highScoreText.text = "HighScore: " + PlayerPrefs.GetInt("HighScore", 0).ToString(); //First HighScore = 0
}

private void Update()
{
    if (time > 0f)
    {
        time -= Time.deltaTime;
        timeText.text = "" + (int)time;
    }
    else
    {
        GameOver();
    }
}

public void GivePoints()
{
    coins++;

    if (coins < 10)
    {
        coinsText.text = "x0" + coins;
    }
    else
    {
        coinsText.text = "x" + coins;
    }

    if (coins >= 15)
    {
        WinGame();
    }

    if (time <= 0f)
    {
        GameOver();
    }
}

public void WinGame()
{
    if (coins >= 15)
    {
        score = (int)(coins * time);
    }
}

```

```

        }

finalText.gameObject.SetActive(true);
finalText.text = "YOU WIN!";
scoreText.gameObject.SetActive(true);

if (score > PlayerPrefs.GetInt("HighScore", 0))
{
    PlayerPrefs.SetInt("HighScore", score);
    scoreText.text = "New HighScore: " + score;
    highScoreText.text = "HighScore: " + score;
}
else
{
    scoreText.text = "Your Score: " + score;
}

Pause();
button.gameObject.SetActive(true);
}

public void GameOver()
{
    score = 0;

finalText.gameObject.SetActive(true);
finalText.text = "YOU LOSE!";
scoreText.gameObject.SetActive(true);
scoreText.text = "Your Score: " + score;

Pause();
button.gameObject.SetActive(true);
}

public void Pause()
{
    Time.timeScale = 0;
}

public void Resume()
{
    Time.timeScale = 1;
}
}

```

AudioManager

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AudioManager : MonoBehaviour
{
    public static AudioManager instance;

    public Sound[] musicSounds, sfxSounds;
    public AudioSource musicSource, sfxSource;

    private int currentTrackIndex = 0;

    private void Awake()
    {
        if (instance == null)
        {
            instance = this;
            DontDestroyOnLoad(gameObject);
        }
        else
        {
            Destroy(gameObject);
            return; // Singleton
        }
    }

    private void Start()
    {
        PlayMusicTrack(currentTrackIndex);
    }

    public void PlayMusicTrack(int index)
    {
        if (index >= 0 && index < musicSounds.Length)
        {
            Sound s = musicSounds[index];
            if (s == null)
            {
                Debug.Log("Sound not found");
            }
            else
            {
                musicSource.clip = s.sound;
                musicSource.Play();
            }
        }
    }
}
```

```

        musicSource.loop = false;
        StartCoroutine(WaitForTrackEnd(s.sound.length));
    }
}
}

private IEnumerator WaitForTrackEnd(float duration)
{
    yield return new WaitForSeconds(duration);
    PlayNextTrack();
}

public void PlayNextTrack()
{
    currentTrackIndex = (currentTrackIndex + 1) % musicSounds.Length;
    PlayMusicTrack(currentTrackIndex);
}

public void OnButtonClickNextTrack()
{
    StopCoroutine("WaitForTrackEnd");
    PlayNextTrack();
}

public void PlayMusic(string name)
{
    Sound s = Array.Find(musicSounds, x => x.soundName == name);

    if (s == null)
    {
        Debug.Log("Sound not found");
    }
    else
    {
        musicSource.clip = s.sound;
        musicSource.Play();
    }
}

public void StopMusic()
{
    AudioManager.instance.musicSource.Stop();
}

public void PlaySFX(string name)
{
    Sound s = Array.Find(sfxSounds, x => x.soundName == name);
}

```

```

        if (s == null)
    {
        Debug.Log("SFX not found");
    }
    else
    {
        sfxSource.PlayOneShot(s.sound);
    }
}
}

```

ButtonManager

```

using UnityEngine;
using UnityEngine.UI;

public class ButtonManager : MonoBehaviour
{
    public Button nextTrackButton;

    private void Start()
    {
        if (nextTrackButton != null)
        {
            nextTrackButton.onClick.AddListener(() =>
            {
                if (AudioManager.instance != null)
                {
                    AudioManager.instance.OnButtonClickNextTrack();
                }
                else
                {
                    Debug.LogError("AudioManager instance is null.");
                }
            });
        }
    }
}

```

Ball

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Ball : MonoBehaviour
{
    private Rigidbody rig;

    [SerializeField]
    private float speed;

    private void Awake()
    {
        rig = GetComponent<Rigidbody>();
    }

    void Update()
    {
        float horizontal = Input.GetAxis("Horizontal");
        float vertical = Input.GetAxis("Vertical");
        rig.AddForce (new Vector3(horizontal,0,vertical) * speed * Time.deltaTime);
    }
}
```

Coins

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Coins : MonoBehaviour
{
    GameManager gameManager;

    private void Awake()
    {
        gameManager = FindObjectOfType<GameManager>();
    }

    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            gameManager.GivePoints(); // Give points according to the collected coins
            Destroy(this.gameObject); // Destroy the coin
        }
    }
}
```

```
        AudioManager.instance.PlaySFX("Coin");
    }
}
}
```

Sound

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[System.Serializable]
public class Sound
{
    public string soundName;
    public AudioClip sound;
}
```

VolumeSettings

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Audio;
using UnityEngine.UI;

public class VolumeSettings : MonoBehaviour
{
    [SerializeField]
    private AudioMixer audioMixer;
    [SerializeField]
    private Slider musicSlider;
    [SerializeField]
    private Slider SFXSlider;

    private void Start()
    {
        if (PlayerPrefs.HasKey("musicVolume"))
        {
```

```

        LoadVolume();
    }
    else
    {
        SetMusicVolume();
        SetSFXVolume();
    }
}

public void SetMusicVolume()
{
    float volume = musicSlider.value;
    audioMixer.SetFloat("music", Mathf.Log10(volume)*20);
    PlayerPrefs.SetFloat("musicVolume", volume);
}

public void SetSFXVolume()
{
    float volume = SFXSlider.value;
    audioMixer.SetFloat("SFX", Mathf.Log10(volume) * 20);
    PlayerPrefs.SetFloat("SFXVolume", volume);
}

private void LoadVolume()
{
    musicSlider.value = PlayerPrefs.GetFloat("musicVolume");
    SFXSlider.value = PlayerPrefs.GetFloat("SFXVolume");

    SetMusicVolume();
    SetSFXVolume();
}

public void MuteMusic()
{
    musicSlider.value = 0;
    PlayerPrefs.SetFloat("musicVolume", musicSlider.value);
}

public void MuteSFX()
{
    SFXSlider.value = 0;
    PlayerPrefs.SetFloat("SFXVolume", SFXSlider.value);
}

public void UnmuteMusic()
{
    musicSlider.value = 1;
    PlayerPrefs.SetFloat("musicVolume", musicSlider.value);
}

```

```
    }

    public void UnmuteSFX()
    {
        SFXSlider.value = 1;
        PlayerPrefs.SetFloat("SFXVolume", SFXSlider.value);
    }
}
```

MainMenu

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class MainMenu : MonoBehaviour
{
    public void PlayGame()
    {
        Time.timeScale = 1;
        SceneManager.LoadSceneAsync(1);
    }

    public void QuitGame()
    {
        Application.Quit();
    }

    public void ShowMainMenu()
    {
        Time.timeScale = 1;
        SceneManager.LoadSceneAsync(0);
    }
}
```

PauseMenu

```
using System.Collections;
using System.Collections.Generic;
```

```

using UnityEngine;
using UnityEngine.SceneManagement;

public class PauseMenu : MonoBehaviour
{
    GameManager gameManager;
    public void PauseGame()
    {
        gameManager.Pause();
    }

    public void ResumeGame()
    {
        gameManager.Resume();
    }

    public void MainMenu()
    {
        SceneManager.LoadSceneAsync(0);
    }
}

```

CameraControl

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraControl : MonoBehaviour
{
    public Transform followTarget;
    private float distanceTargetX = 10f;
    private float camSpeed = 15f;
    public bool smoothedActivated = false;
    private Vector3 newPos;

    void Update()
    {
        newPos = this.transform.position;
        newPos.x = followTarget.transform.position.x - distanceTargetX;
        newPos.z = followTarget.transform.position.z;

        if (smoothedActivated)
        {
            this.transform.position = Vector3.Lerp(this.transform.position, newPos, camSpeed * Time.deltaTime);
        }
    }
}

```

```
    }
    else
    {
        this.transform.position = newPos;
    }
}
```

Conclusions

By completing this project I have learned a lot. As my first experience with Unity, and creating a video game also, I realized that the initial project I had proposed was too difficult to finish before the deadline, so I changed my mind and tried an easier game, resulting in a funny and enriching experience. I really enjoyed during the realization of this project, so probably, I will modify this game and I will end the other also in my free time

Bibliography

- [1] [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)) (accessed on 27.05.2024)
- [2] [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)) (accessed on 27.05.2024)
- [3] https://en.wikipedia.org/wiki/Visual_Studio (accessed on 27.05.2024)
- [4]
https://upload.wikimedia.org/wikipedia/commons/thumb/2/2c/Visual_Studio_Icon_2022.svg/25px-Visual_Studio_Icon_2022.svg.png (accessed on 27.05.2024)
- [5] [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)) (accessed on 27.05.2024)
- [6]
https://upload.wikimedia.org/wikipedia/commons/thumb/b/bd/Logo_C_sharp.svg/1820px-Logo_C_sharp.svg.png (accessed on 27.05.2024)
- [7] Youtube Tutorial - <https://www.youtube.com/watch?v=bysQuBFe22w> (accessed on 28.05.2024)
- [8] <https://www.jamendo.com/start> (accessed on 28.05.2024)
 - <https://www.jamendo.com/track/1784811/waiting-world> (Waiting World - Anitek)
 - <https://www.jamendo.com/track/1465314/moments> (Moments - THE.MADPIX.PROJECT)
 - <https://www.jamendo.com/track/1900662/mineral> (Mineral - HRCRX)
 - <https://www.jamendo.com/track/1784818/demure> (Demure - Anitek)
 - <https://www.jamendo.com/track/1884527/the-deep> (The Deep - Anitek)
- [9] <https://assetstore.unity.com/> (accessed on 27.05.2024)
 - <https://assetstore.unity.com/packages/2d/gui/icons/ux-flat-icons-free-202525> (Icons)
 - <https://assetstore.unity.com/packages/audio/sound-fx/free-sfx-package-5178> (Coin sound)
 - <https://assetstore.unity.com/packages/2d/gui/buttons-set-211824> (Buttons)
 - <https://assetstore.unity.com/packages/2d/textures-materials/wood/hand-painted-seamless-wood-texture-vol-6-162145> (Walls texture)
 - <https://assetstore.unity.com/packages/2d/textures-materials/stone/stone-floor-texture-20336> (Floor texture)
- [10]
Tiger print-<https://i.pinimg.com/474x/a8/5d/c0/a85dc04f742acc7b866a91ccc9f63d5b.jpg>
(accessed on 27.05.2024)
Background-https://st3.depositphotos.com/2370557/19141/i/450/depositphotos_191418712-stock-photo-abstract-refocused-hologram-gradient-background.jpg (accessed on 27.05.2024)