CSE 7021 TERM PROJECT

DENİZ BOZKAN - 20190808044

DAMLA HATİCE ŞİMŞEK - 20180808067

BÜŞRA SARIGEYİK - 2018080847

CONTENTS

a.Introduction

a.1 Inspiration

Our main goal was to make games like the ones we played as children. We could go to different worlds and experience different games. That is why the primary goal of our design and concept was to create a game that reminded us of our childhood. We wanted to design a game that would make us feel like children again and that we could sit and play for hours without getting bored.

We wanted a world that we can change the environment, our inspiration for that was the world. In real world we cannot see different environments easily we have to take some miles to get there. So we wanted to visit different places in our game. You can see different worlds in our game and have fun with the games.

a.2 What it does?

There are different universes and different games. Our character has to wander around the universe and find the playgrounds hidden there. We added doors to pass along the universes. Player has to find the doors in the map and with touching it, player can enter a different world. Player can play the games with touching the doors as well. We add two different game so some doors opens these games.

In our first game player has to pass the levels and earn the tokens. You have to jump on the tokens to earn them and you have to pass some barriers. In our second game alsı player has to pass the levels. In this game there is a ball and player have to move it without falling down.

With these games player will have fun and want to pass the levels. With our universe player will explore new things.

a.3 How you build it?

We used the Unity Hub. We have designed different games and terrains. Our character can move freely in the terrain. The character needs to find the doors that transfer he or she to the game. The character can experience different games when passing through the doors. We have added scenes from the builder settings and placed them in order to combine different terrains and games. We have written C# codes for transition points and added the codes to the component section.

We used different assets to design our terrains. To gain third person controller we used Unity Hub's starter assets package.

To build our game we got help from many different Unity tutorials and used YouTube to code the project. For designing the terrains we got inspiration from different YouTube videos. We got help from some other friends as well.

a.4 Challenges you ran into

It was difficult to add animation to the character. All the members of our group tried to add animation to the character. The transition between animations was difficult. We could not move the avatar properly. It wasn't using the running animation or standing animation. Also we had some camera problem. Our camera wasn't following the avatar and wasn't moving. We could not place the camera behind the avatar for long time. To add third person controller to the camera we spend too much time to trying it. After some time we solve that problem with adding a new asset to the project.

For some reason our terrain could not use the design assets that we download. So we could not design our terrain for some time. We find solution as reimporting our assets.

One of our frinds had some problem with unity versioning. Our project used different version but my friend used different version for her project. We didn't start with same projects. Everyone had openned different projects and implement their games on that project. After some time we tried to combine the projects but we had version issues. My friend tried to install that version but Unity Hub wouldn't open so we pass the project to our other member and make the differences with her computer.

We had little trouble with scene changing. But we quickly solve that problem with help of some YouTube video.

a.5 Accomplishments that you are proud of

We didn't know about c# that much but with this project we learn many more things about the language. Any of us hadn't use Unity Hub before and build a game but with this project we build a game for the first time. We have always wondered how do they make an universe like forests in the game but now er know how to make it so we are very proud of it.

Solving the the third person controller problem and being able to move the avatar with the animations were very satisfying because we work on that problem many hourse. Now we know how to add an avatar and camera that will follow the avatar.

Connecting the scenes with doors was very clever and fun to make. We are very proud of doing that.

a.6 What you learned

We learned that Unity Hub was very easy to learn and with additional assets it is very easy to code a game or many other things. With these assets we can design an universe that we want and add many additional things. Other thing that we learned is for the object movement we have to add scripts to determine position and rotation. And also we have to add scripts for the scene changing as well.

It is very important to use correct version to implement the game. With different versions it can cause problems. So we have to be carefull about it.

We have learned that if you want to build something you can build anything you want if you try it and use YouTube tutorials. YouTube was very helpful for our project.

b.Method and Code Explanation

We use this class to add constant velocity to the objects.

```
public class AddConstantVelocity : MonoBehaviour {
    [SerializeField]
    Vector3 v3Force;

    void FixedUpdate() {
        GetComponent<Rigidbody>().velocity += v3Force;
    }
}
```

With this class player can determine the velocity of the object. If it is key positive velocity will increase if otherwise it will decrease.

```
public class AddPlayerControlledVelocity : MonoBehaviour {
    [SerializeField]
    Vector3 v3Force;
    [SerializeField]
    KeyCode keyPositive;
    [SerializeField]
    KeyCode keyNegative;

    void FixedUpdate() {
        if(Input.GetKey(keyPositive))
         GetComponent<Rigidbody>().velocity += v3Force;
        if(Input.GetKey(keyNegative))
         GetComponent<Rigidbody>().velocity -= v3Force;
    }
}
```

We use this function to quit the games.

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

We use this method for Finish class. We use this method for transition between levels.

```
private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.name == "Player")
        {
            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +
1);
        }
    }
```

This class used for collecting coins. One of our game has a feature that we can collect coins, this class used for that.

```
public class ItemCollector : MonoBehaviour
{
    int coins = 0;

    [SerializeField] TextMeshProUGUI coinsText;

    [SerializeField] AudioSource collectionSound;

    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag("Coin"))
        {
            Destroy(other.gameObject);
            coins++;
            coinsText.text = "Coins: " + coins;
            collectionSound.Play();
        }
    }
}
```

We use this class for loading the levels.

```
public class Menuu : MonoBehaviour
{

    public static bool GameIsPaused = false;

    public GameObject pauseMenuUI;

    public void Startt()
    {
        Time.timeScale = 1f;
        SceneManager.LoadScene("Start Screen 1");
    }
    public void Level01()
    {
        Time.timeScale = 1f;
        SceneManager.LoadScene("Level001");
    }
    public void Level02()
    {
        Time.timeScale = 1f;
```

```
        SceneManager.LoadScene("Level002");
    }
    public void Level03()
    {
        Time.timeScale = 1f;
        SceneManager.LoadScene("Level003");
    }
    public void QuitGame()
    {
        Time.timeScale = 1f;
        SceneManager.LoadScene("Start Screen 1");
    }
}
```

We use this class for pausing in the game and returning to menu.

```
public class PauseMenu : MonoBehaviour{

    public static bool GameIsPaused= false;

    public GameObject pauseMenuUI;


    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Escape))
        {
            if(GameIsPaused)
            {
                Resume();
            }
            else
            {
                Pause();
            }
        }
    }

    public void Resume()
    {
        pauseMenuUI.SetActive(false);
        Time.timeScale = 1f;
        GameIsPaused= false;
```

```
    }
    void Pause()
    {
        pauseMenuUI.SetActive(true);
        Time.timeScale = 0f;
        GameIsPaused= true;
    }

    public void LoadMenu()
    {
        Time.timeScale = 1f;
        SceneManager.LoadScene("Menuu");
    }
    public void QuitGame()
    {
        Time.timeScale = 1f;
        SceneManager.LoadScene("Start Screen 1");
    }
}
```

In this class we determine the conditions that player is dead.

```
public class PlayerLife : MonoBehaviour
{
    [SerializeField] AudioSource deathSound;

    bool dead = false;

    private void Update()
    {
        if (transform.position.y < -1f && !dead)
        {
            Die();
        }
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Enemy Body"))
        {
            GetComponent<MeshRenderer>().enabled = false;
            GetComponent<Rigidbody>().isKinematic = true;
            GetComponent<PlayerM>().enabled = false;
            Die();
        }
    }
```

```
    void Die()
    {
        Invoke(nameof(ReloadLevel), 1.3f);
        dead = true;
        deathSound.Play();
    }

    void ReloadLevel()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }
}
```

With this class we can pass through the door and change to the rolling game scene.

```
public class rollinggamekapısı : MonoBehaviour
{
    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.name == "PlayerArmature")
        {
            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +
1);
        }
    }
}
```

This class used for reseting the game. When playes falls down from platform it goes to starting point of the game so it resets the game.

```
public class Reset : MonoBehaviour {

    [SerializeField]
    private Transform player;
    [SerializeField]
    private Transform respawnPoint;

    void OnTriggerEnter(Collider other){
        player.transform.position = respawnPoint.transform.position;
    }

}
```

c.Conclusion

In conclusion we have learned so many things from this project and one of the member of this group decided to advance in this profession. For the future we can add many more futures to this project. We definitly will continue to build more projects like this.