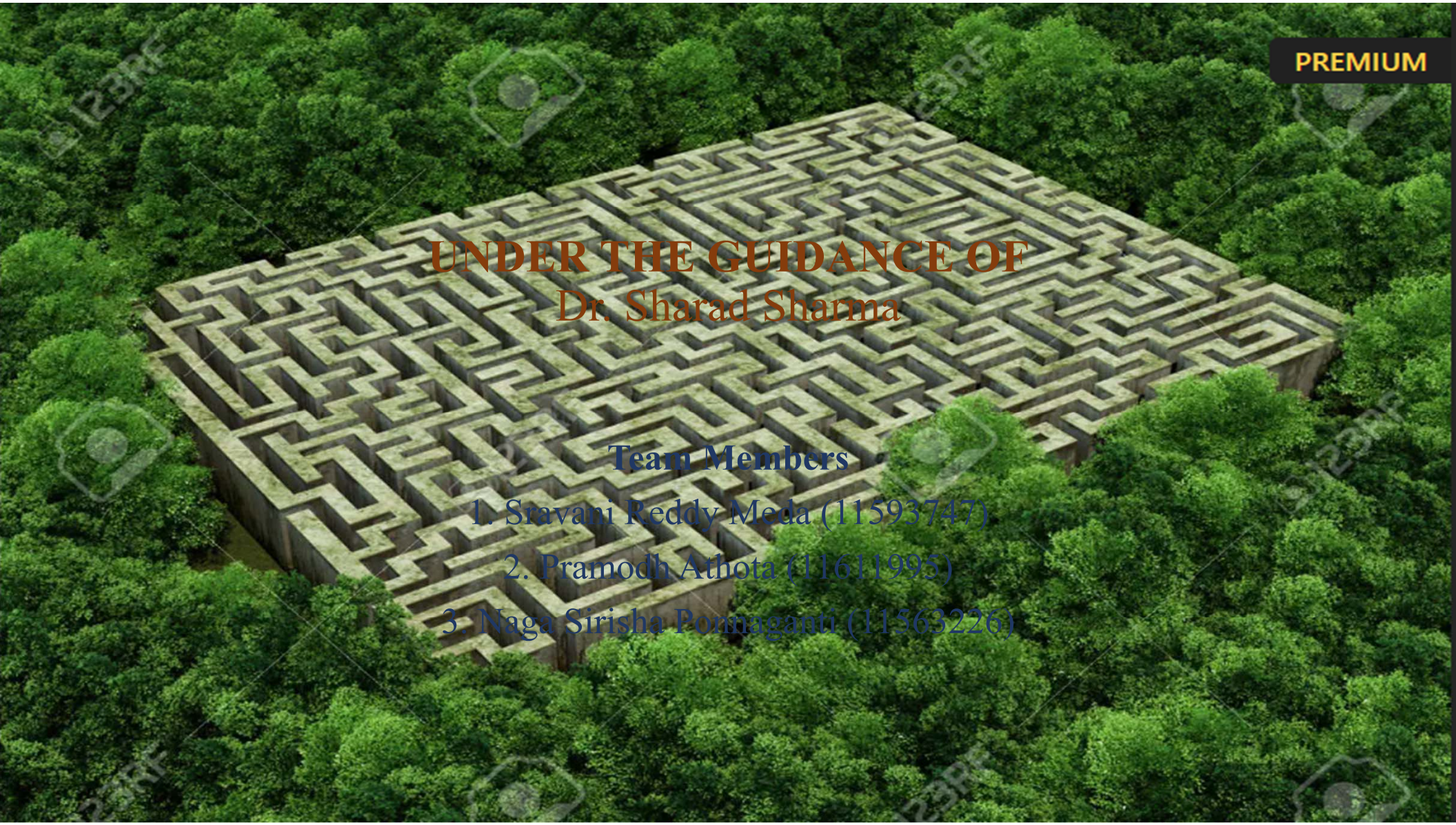
VIRTUAL REALITY AND IT'S APPLICATIONS
FINAL PROJECT

THE MAZE MASTER



UNDER THE GUIDANCE OF

Dr. Sharad Sharma

Team Members

1. Sravani Reddy Meda (11593747)
2. Pramodh Athota (11611995)
3. Naga Sirisha Ponnaganti (11563226)

CONTENTS

ABSTRACT

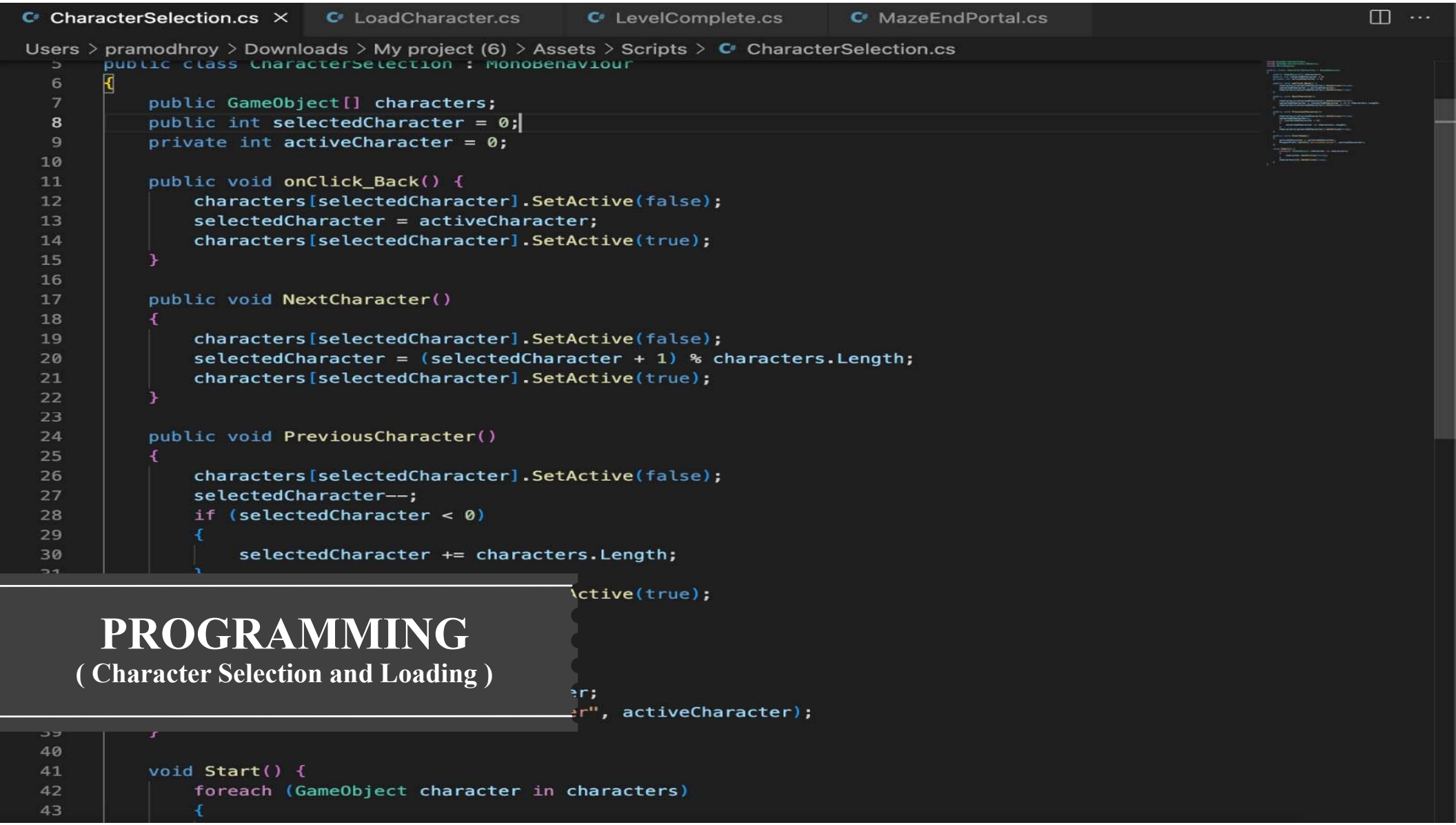
- A VR experience that is both immersive and interactive was developed using Unity for the Maze Master project. The goal of this project is to give players a fun and interesting approach to go into a maze-like environment and solve puzzles.
- The target audience for Maze Master game are primarily individuals who enjoy puzzle-solving and are interested in exploring new and innovative virtual environments.
- The application is useful because it provides a fun and interactive way for individuals to improve their problem-solving skills while also engaging with cutting-edge technology.


My project (6)
Time - 00:19
MODELLING
(Easy Level)
DESERT THEME
➢ WALLS & GROUND – SAND STONE TEXTURE
➢ 4K TEXTURES FROM BLENDER & POLY HAVEN
➢ PRO-BUILDER TOOL (To construct Maze)

My project (6)


Time - 00:19
MODELLING
(Medium Level)
FOREST THEME
➢ WALLS & GROUND –GREEN FOREST TEXTURE
➢ VOLUMETRIC FOG – USING PARTICLE GENERATOR PLUGIN
➢ PRO-BUILDER TOOL (To construct Maze)


My project (6)
Time - 00:23
MODELLING
(Difficult Level)
CITY THEME
➢ WHITE CITY PACKAGE FROM UNITY ASSET STORE
➢ 6000 BUILDINGS
➢ PRO-BUILDER TOOL (To construct Maze)



PROGRAMMING

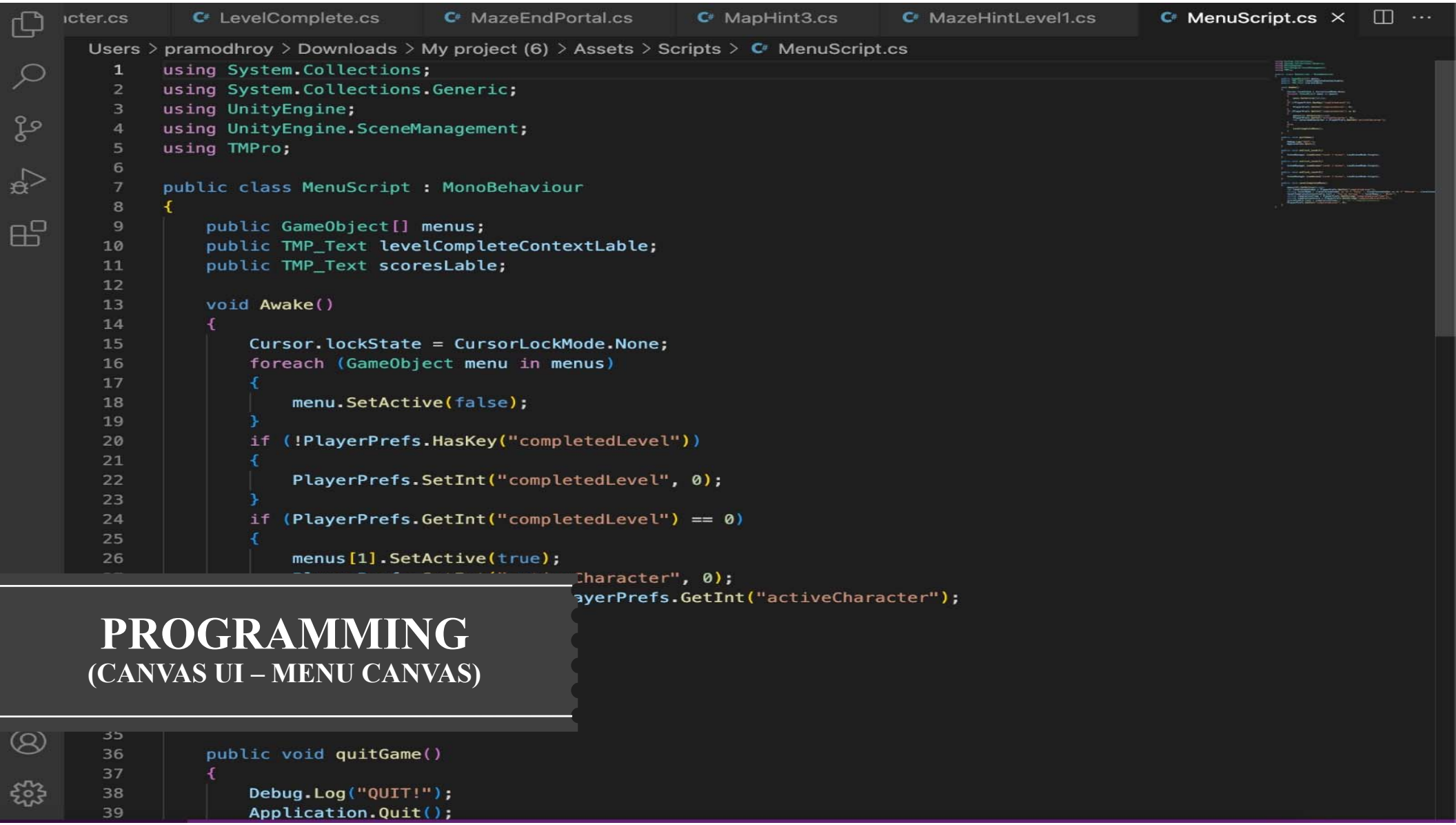
(Character Selection and Loading)

CharacterSelection.cs | LoadCharacter.cs | LevelComplete.cs | MazeEndPortal.cs

Users > pramodhroy > Downloads > My project (6) > Assets > Scripts > LevelComplete.cs

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

public class LevelComplete : MonoBehaviour //Unused Script
{
    public TMP_Text contextLable;
    public TMP_Text scoresLable;

    void StartNot()
    {
        string completionTime = "0" + PlayerPrefs.GetString("completedLevelTime") + ":00";
        string completionScore = PlayerPrefs.GetString("completedLevelScore") + "00";
        scoresLable.text = completionTime + "\n" + completionScore;
        PlayerPrefs.SetInt("completedLevel", 0);
    }

    // Update is called once per frame
    void Update()
    {

    }
}

```

PROGRAMMING
(CANVAS UI – LEVEL COMPLETION)

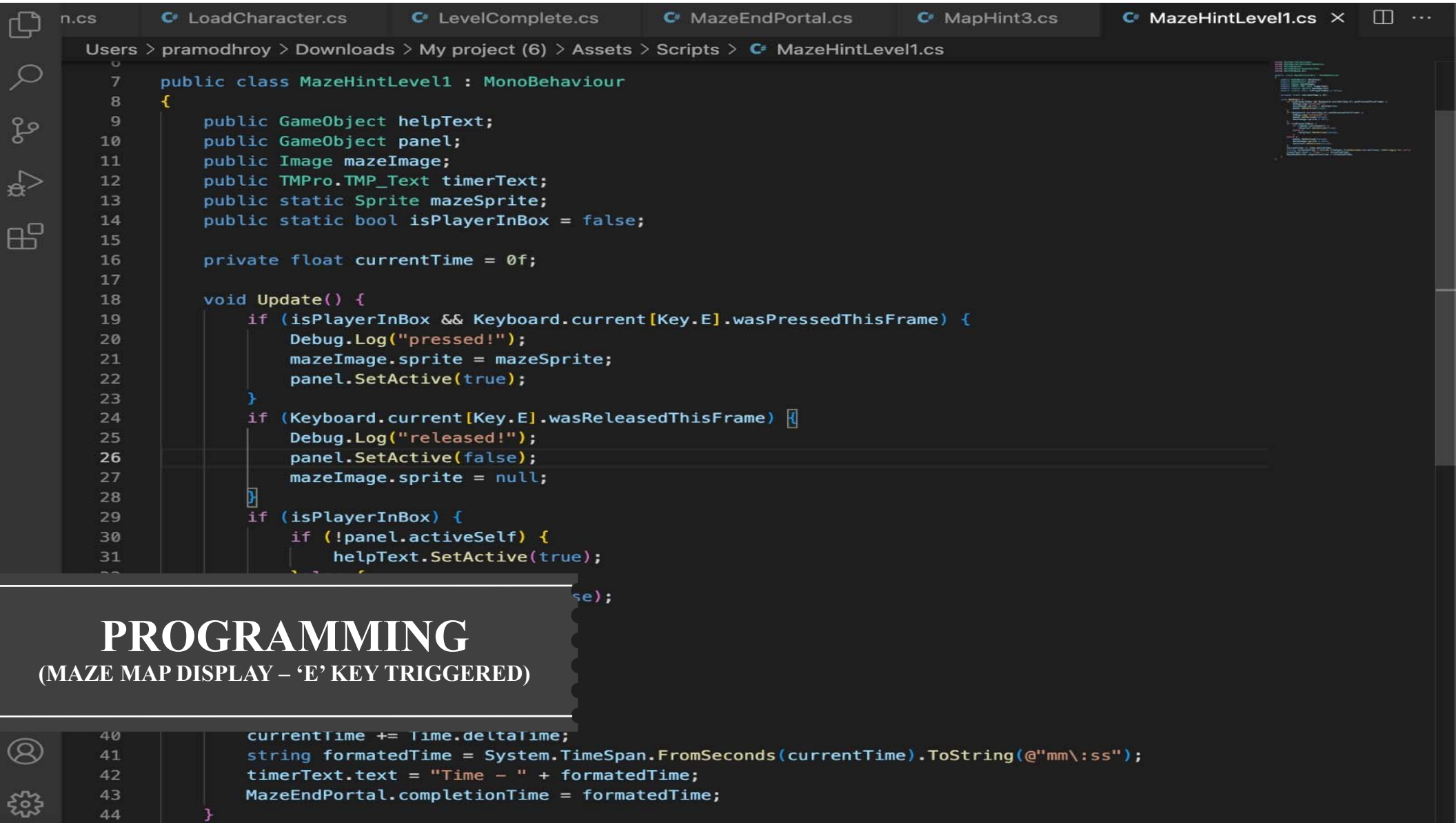

acter.cs
LevelComplete.cs
MazeEndPortal.cs
MapHint3.cs
MazeHintLevel1.cs
MenuScript.cs
Users > pramodhroy > Downloads > My project (6) > Assets > Scripts > MenuScript.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;
public class MenuScript : MonoBehaviour
{
public GameObject[] menus;
public TMP_Text levelCompleteContextLable;
public TMP_Text scoresLable;
void Awake()
{
Cursor.lockState = CursorLockMode.None;
foreach (GameObject menu in menus)
{
menu.SetActive(false);
}
if (!PlayerPrefs.HasKey("completedLevel"))
{
PlayerPrefs.SetInt("completedLevel", 0);
}
if (PlayerPrefs.GetInt("completedLevel") == 0)
{
menus[1].SetActive(true);
Character", 0);
ayerPrefs.GetInt("activeCharacter");
PROGRAMMING
(CANVAS UI – MENU CANVAS)
public void quitGame()
{
Debug.Log("QUIT!");
Application.Quit();

CharacterSelection.cs | LoadCharacter.cs | LevelComplete.cs | MazeEndPortal.cs | MapHint3.cs

Users > pramodhroy > Downloads > My project (6) > Assets > Scripts > MapHint3.cs

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class MapHint3 : MonoBehaviour
6   {
7       public Sprite mazeImage;
8
9       void OnTriggerEnter(Collider other)
10      {
11          Debug.Log("Entered box");
12          MazeHintLevel3.isPlayerInBox = true;
13          MazeHintLevel3.mazeSprite = mazeImage;
14
15      }
16
17      void OnTriggerExit(Collider other)
18      {
19          Debug.Log("Exited box");
20          MazeHintLevel3.isPlayerInBox = false;
21          MazeHintLevel3.mazeSprite = null;
22      }
23  }
24
```

PROGRAMMING
(HINT DISPLAY)



PROGRAMMING

(MAZE MAP DISPLAY – 'E' KEY TRIGGERED)

FUNCTIONALITY

VISION:

- EASY LEVEL – 10*10 MAZE , DESERT THEME
- MEDIUM LEVEL – 15*15 MAZE , FOREST THEME
- DIFFICULT LEVEL – 20*20 MAZE , CITY THEME

FUNCTIONALITY

SOUND:

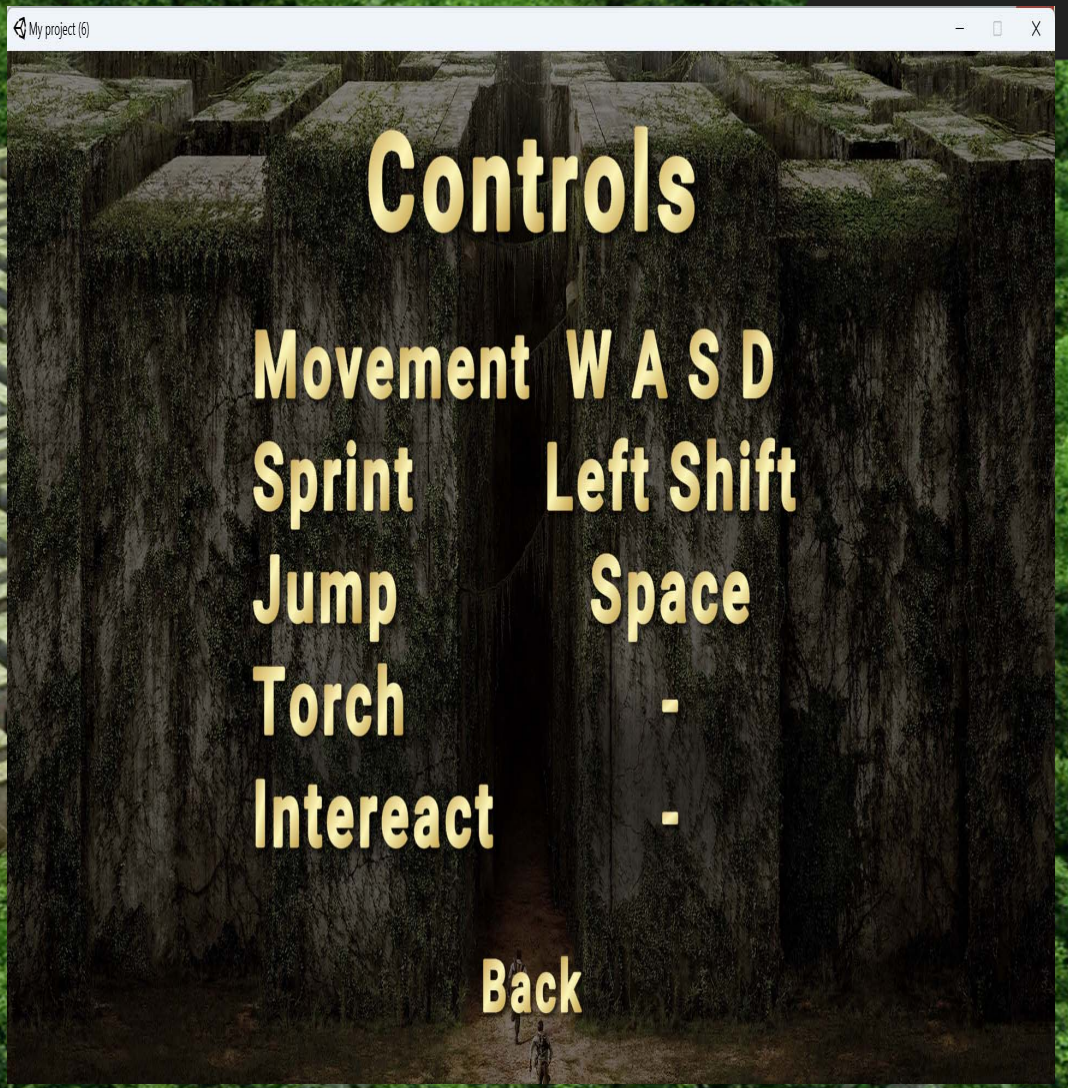
- FOOTSTEPS OF THE PLAYER ACCORDING TO MOVEMENT
- SOUND EFFECTS WHEN PLAYER REACHES A POINT IN MAZE.
- BACKGROUND MUSIC THROUGHOUT THE GAME

FUNCTIONALITY

ANIMATIONS:

- **PLAYER ANIMATIONS –**

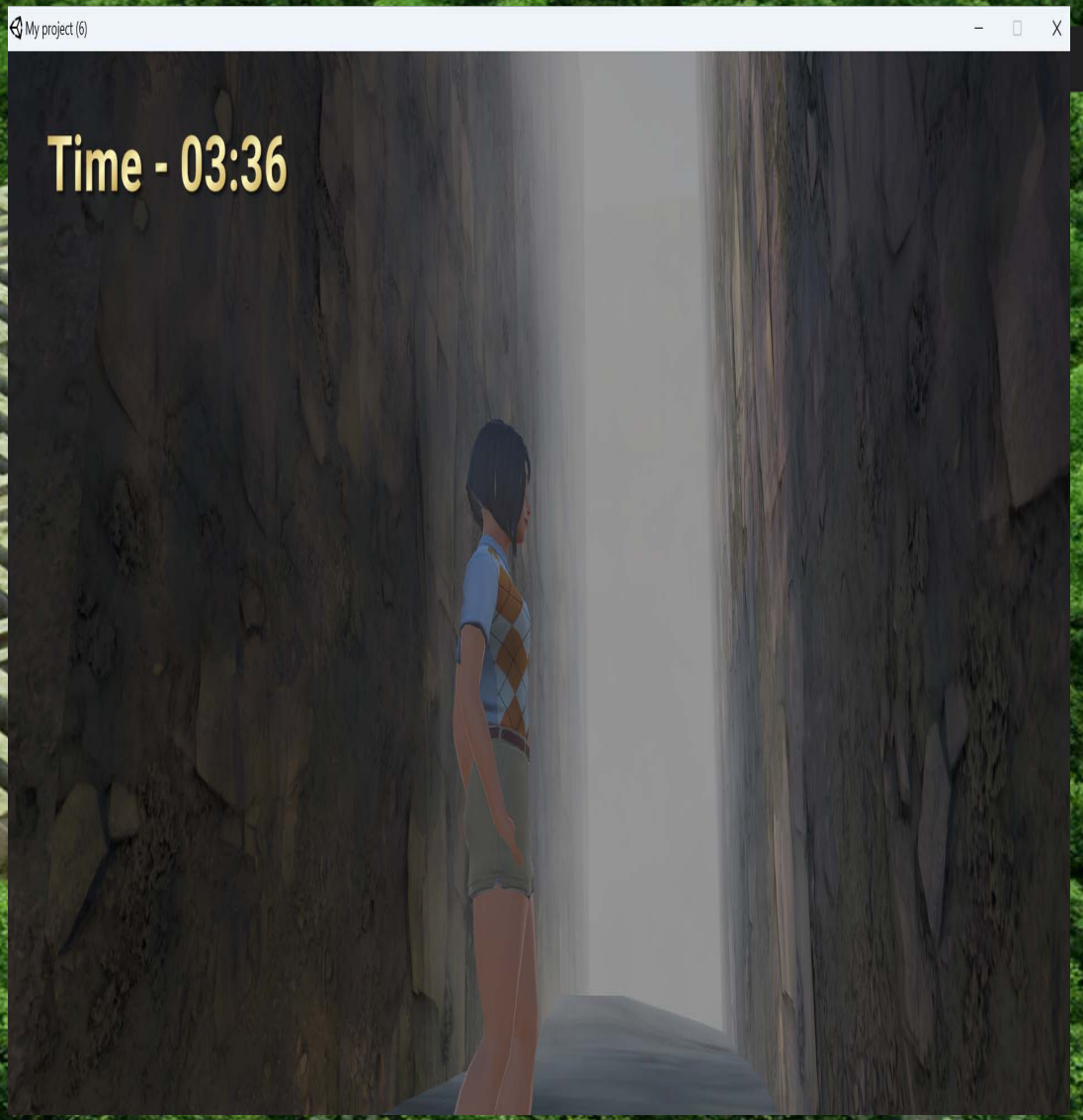
Player movement in all the directions, Jumping , Running and Climbing



FUNCTIONALITY

ANIMATIONS:

- **FOG ANIMATIONS –**
  Volumetric Fog is created using particle generator Plugin



FUNCTIONALITY

INTERACTIVITY (AVATAR SELECTION)

FUNCTIONALITY

- INTERACTIVITY (LEVEL SELECTION)

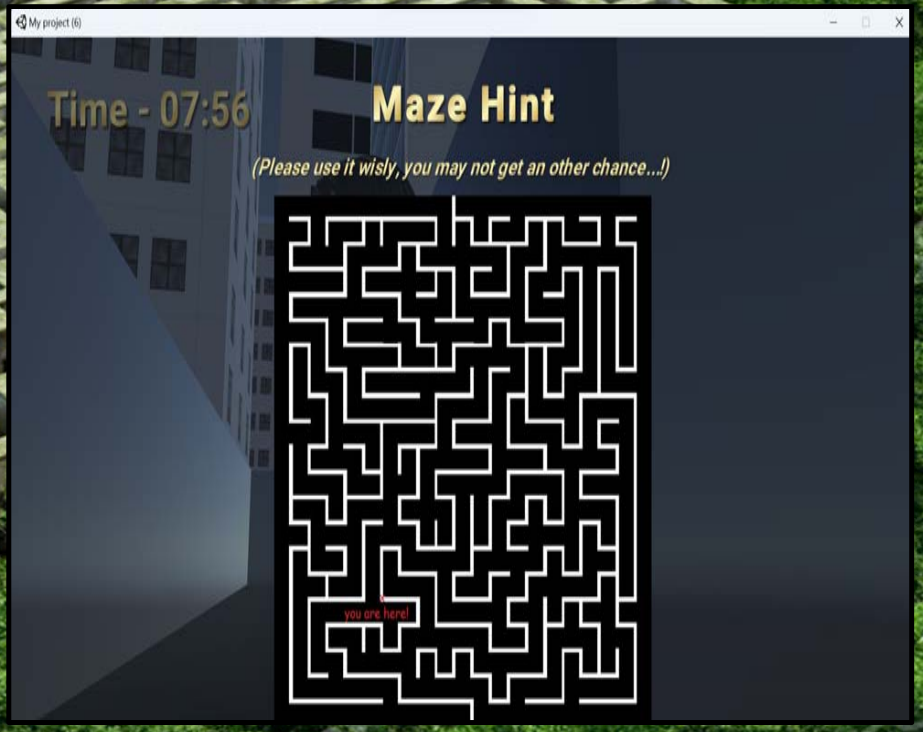
PREMIUM

FUNCTIONALITY

INTERACTIVITY (HELP NOTIFICATION)

FUNCTIONALITY

INTERACTIVITY (MAZE MAP DISPLAY)



FUNCTIONALITY

INTERACTIVITY (TELEPORTATION)

FUNCTIONALITY

CHARACTERS/AVTARS

- FEMININE AVTAR
- MASCULINE AVTAR
- W, A, S, D and LEFT SHIFT KEY , SPACE BAR for movement.
- Interactive with environment
- If the player collides with walls/buildings – Proximity Sensor

FUNCTIONALITY

- SENSORS (PROXIMITY SENSOR)

FUNCTIONALITY

- SENSORS (TIME SENSOR)

FUNCTIONALITY

- SENSORS (KEYBOARD INPUT - TOUCH SENSOR)

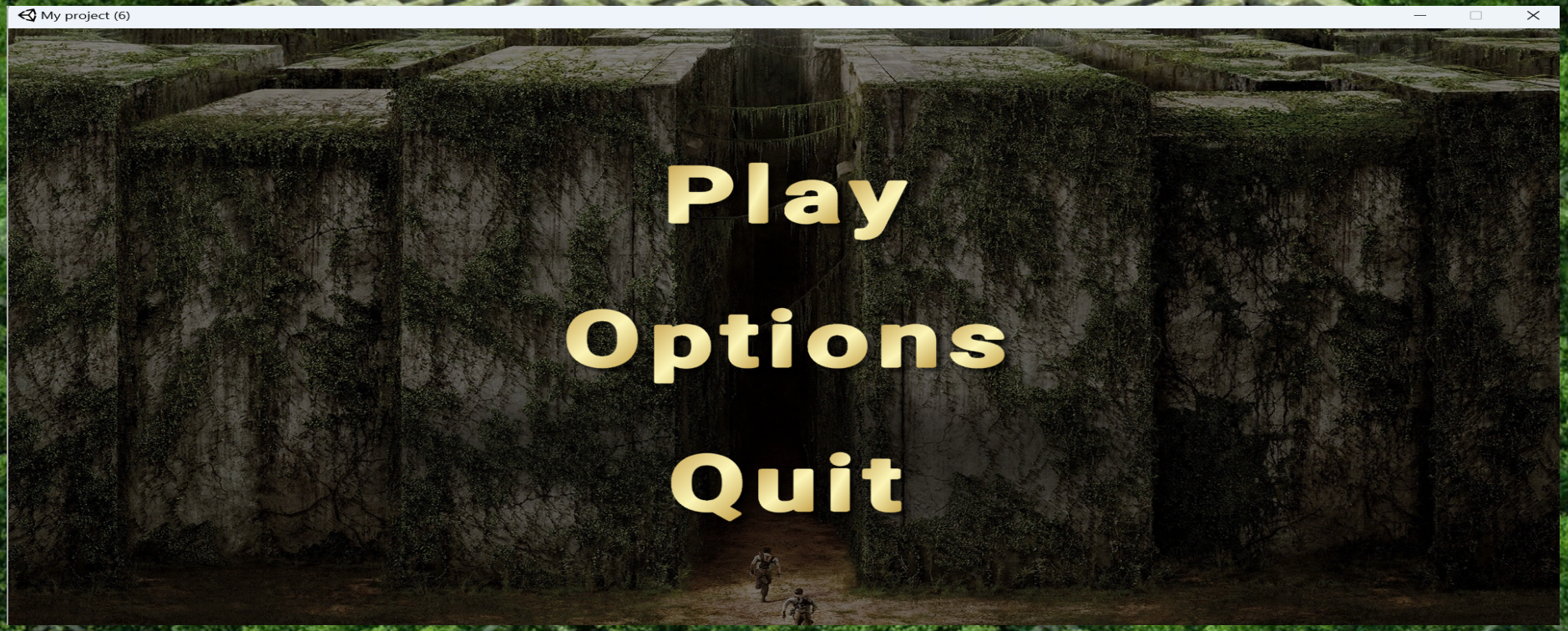
FUNCTIONALITY

PLAYER

Third Person Controller

FUNCTIONALITY

INTERFACE ELEMETS - MAIN MENU



FUNCTIONALITY

INTERFACE ELEMETS – CHARACTER SELECTION

FUNCTIONALITY

INTERFACE ELEMETS - OPTIONS

PREMIUM

FUNCTIONALITY

INTERFACE ELEMETS – HELP NOTIFICATION

FUNCTIONALITY

INTERFACE ELEMETS – MAZE MAP

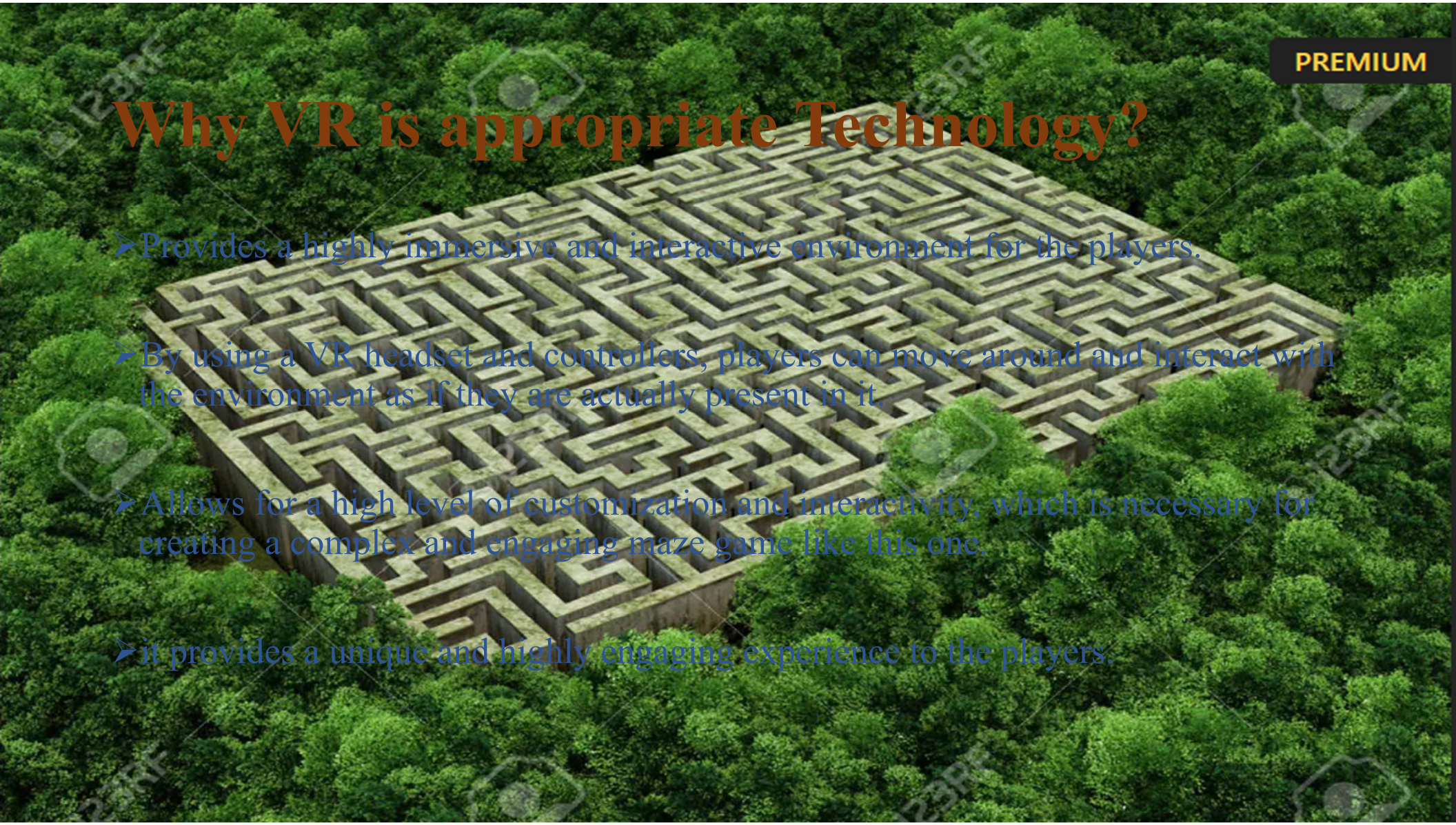
FUNCTIONALITY

INTERFACE ELEMETS – TELEPORTATION

FUNCTIONALITY

INTEREACE ELEMETS – TIMER

A timer runs for each level separately to calculate the time taken by the player to solve the maze

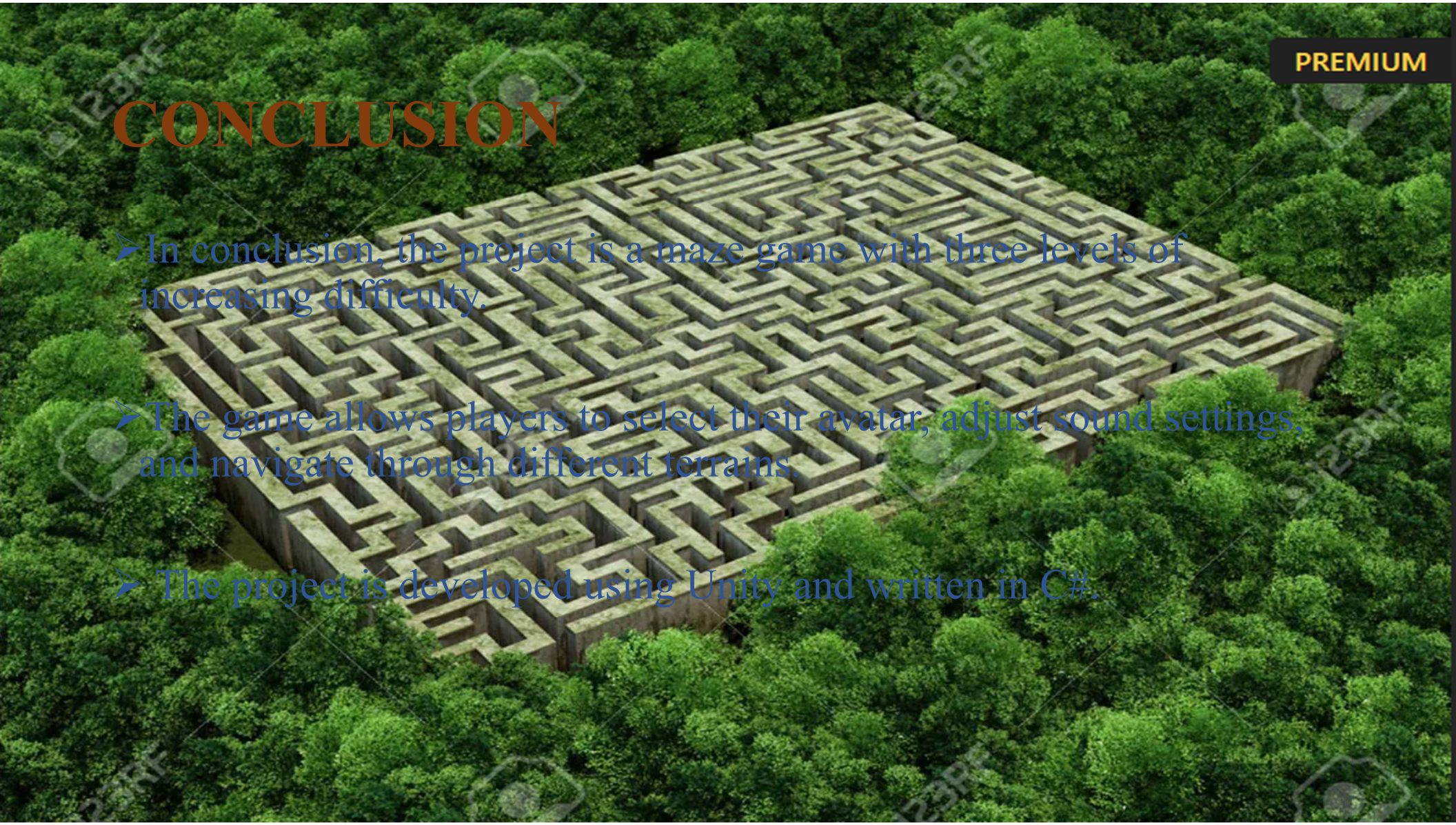


Why VR is appropriate Technology?

- Provides a highly immersive and interactive environment for the players.
- By using a VR headset and controllers, players can move around and interact with the environment as if they are actually present in it.
- Allows for a high level of customization and interactivity, which is necessary for creating a complex and engaging maze game like this one.
- it provides a unique and highly engaging experience to the players.

FUTURE WORK

- Multiplayer Mode
- More Levels
- Adding Opponent
- Achievements and Rewards



CONCLUSION

- In conclusion, the project is a maze game with three levels of increasing difficulty.
- The game allows players to select their avatar, adjust sound settings, and navigate through different terrains.
- The project is developed using Unity and written in C#.


PREMIUM
THANK YOU !!