Escape the Maze

Introduction:

The Escape the Maze game involves a character stranded inside a maze and there is a hidden block leading to the exit. Player needs to look through the maze using keyboard and mouse input to escape the levels to complete the game.

Features:

The game involves 2 levels with different difficulty levels.

Level 1: Player needs to find the hidden exit block amidst fake walls. Time given to complete the quest is 180 seconds.

Level 2: Player needs to find the hidden exit amidst fake walls, magic blocks, and time constraint of 90 seconds.

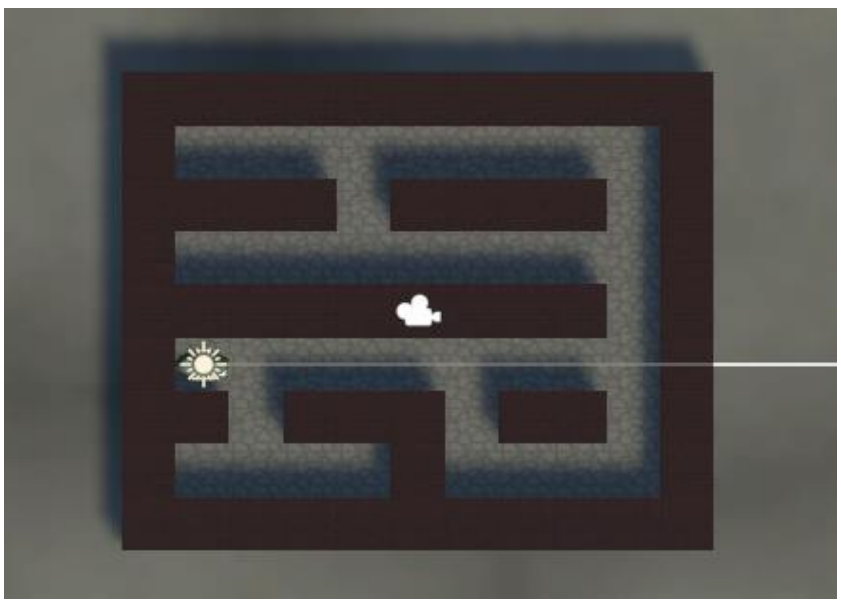
Also, map of the gameplay is rendered on to the gameplay to assist the player using render texture.

Note: Magic blocks lead the players near the exit block.

Snapshots:

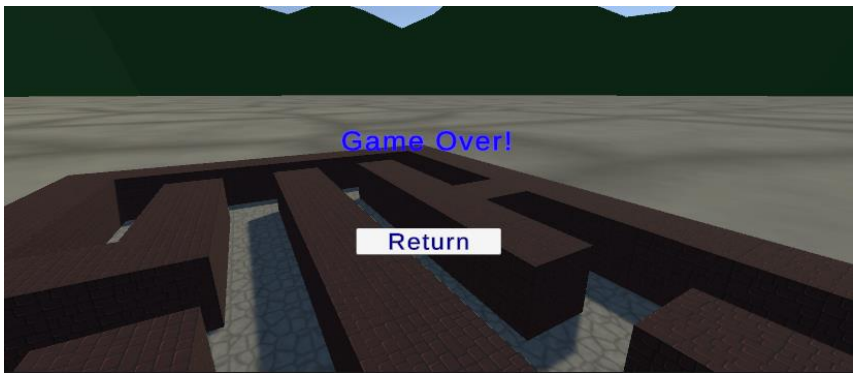
Main menu:

Level 1:



Level 2:

Game Over:



Implementation:

CanvasScript.cs

```
public void PressPlay() // Method for play button
    {
        SceneManager.LoadScene("Level 1"); // Load Level 1 scene
    }

    public void PressQuit() // Method for quit Button
    {
        Application.Quit(); // Quits the game
    }

    public void PressReturn() // Method for quit Button
    {
        SceneManager.LoadScene("Menu"); // Load Level 1 scene
    }
```

Loads the scenes based on button press.

PlayerMove.cs

```
        Vector3 move = transform.right * x + transform.forward * z

        charController.Move(move * speed * Time.deltaTime

        if (Input.GetButtonDown("Jump") && grounded && canJump)
        {
            velocity.y = Mathf.Sqrt(jumpHeight * -2f * gravity
        }

        velocity.y += gravity * Time.deltaTime;

        charController.Move(velocity * Time.deltaTime);
```

The above code block helps the character to move across the maze

```
    private void OnTriggerEnter(Collider collisionCube)
    {
        if (collisionCube.tag == "End Cube") {
            Scene currentScene = SceneManager.GetActiveScene();

            if (currentScene.name != "Level 2")
            {
                SceneManager.LoadScene(SceneManager.GetActiveScene().buildInde
x + 1);
            }
            else
            {
                SceneManager.LoadScene("Menu");
            }
        }

        if (collisionCube.tag == "Magic Cube") {

            charController.enabled = false;
            transform.position = new Vector3(9, 0.5f, 0);
            charController.enabled = true;
        }


    }
```

Will be called when the player enters the End Cube trigger and magic cube trigger.

TimerController.cs

```
private void Start()
    {
        timeCounter.text = "Time: 01:00.00";
        timerGoing = false;
        BeginTimer();
    }

    public void BeginTimer()
    {
```

```
        timerGoing = true;
        elapsedTime = 90f;
        if(SceneManager.GetActiveScene().name == "Level 1") {
            elapsedTime = 180f;
        }
        coroutine = UpdateTimer();
        StartCoroutine(coroutine);
    }

    public void EndTimer()
    {
        timerGoing = false;
        timeCounter.text = "Time: 00:00.00";
        StopCoroutine(coroutine);
        SceneManager.LoadScene("Game Over");
    }

    private IEnumerator UpdateTimer()
    {
        while (timerGoing)
        {
            elapsedTime -= Time.deltaTime;
            if (elapsedTime <= 0f) {
                EndTimer();
                yield return null;
            }
            timePlaying = TimeSpan.FromSeconds(elapsedTime);
            string timePlayingStr = "Time: " +
timePlaying.ToString("mm':'ss'.'ff");
            timeCounter.text = timePlayingStr;

            yield return null;
        }
    }
```

Used to update timer and take action accordingly.

Further Enhancements:

Add more levels
Add scoring
Improve Graphical User Interface

References:

https://forum.unity.com/threads/render-a-camera-to-a-cube.26493/

https://docs.unity3d.com/Manual/class-MeshFilter.html

https://docs.unity3d.com/Manual/class-AnimationClip.html

https://learn.unity.com/