

Room Escape

20211024 조성진

목 차

01. 게임 소개

02. 게임 구성

03. 핵심 코드

04. 시현 영상

01.

게임 소개

✓ Room Escape

단서들을 종합하여 문 옆에 존재하는 키패드를 입력하여 클리어하는 방탈출 게임으로 관찰력을 중요시 하는 게임입니다.

방탈출 게임인 만큼 시간제한 또한 존재하여 시간 안에 키패드에 정답을 입력하지 못하거나 시간이 초과되어도 그대로 게임이 종료됩니다.

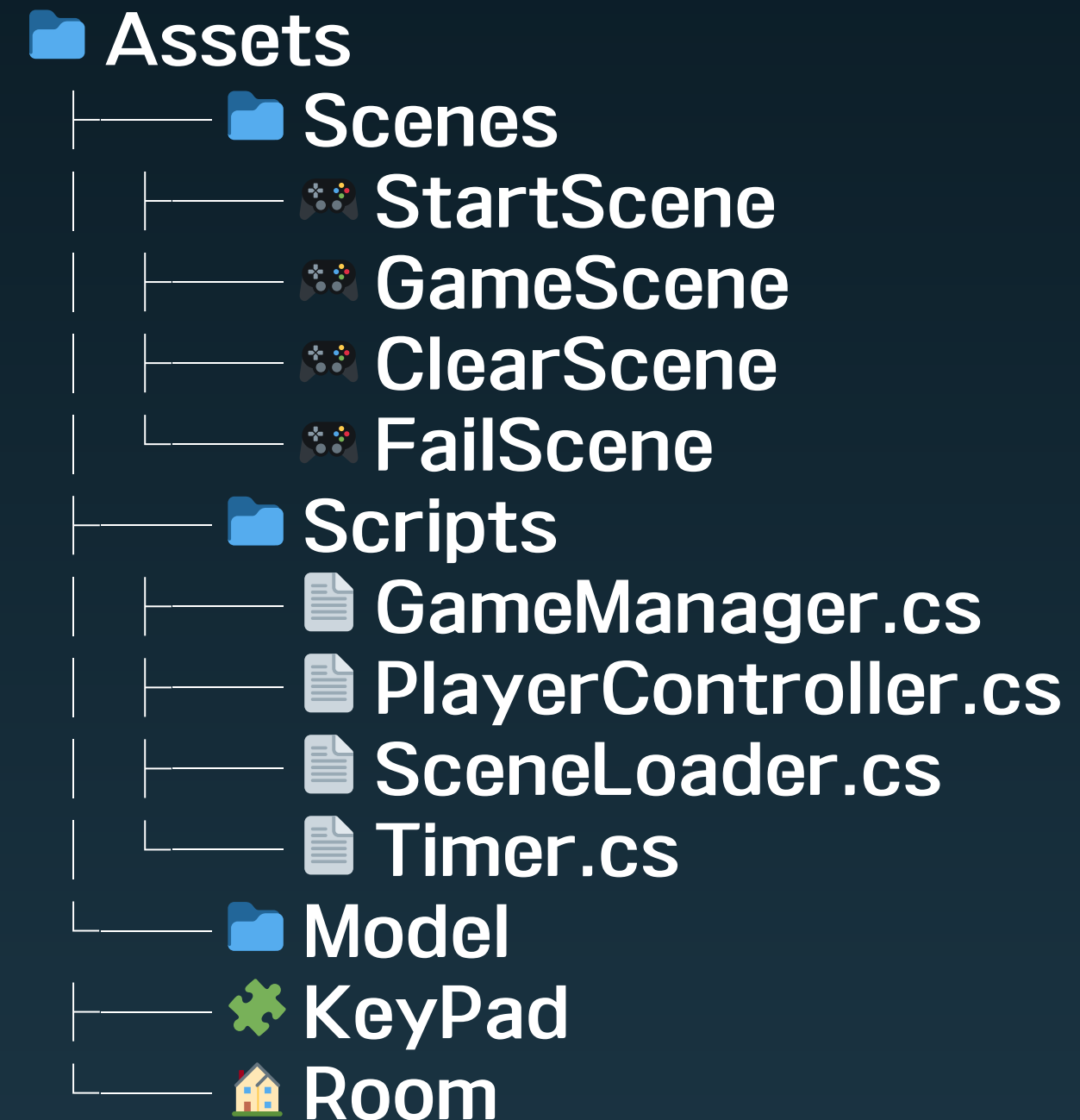
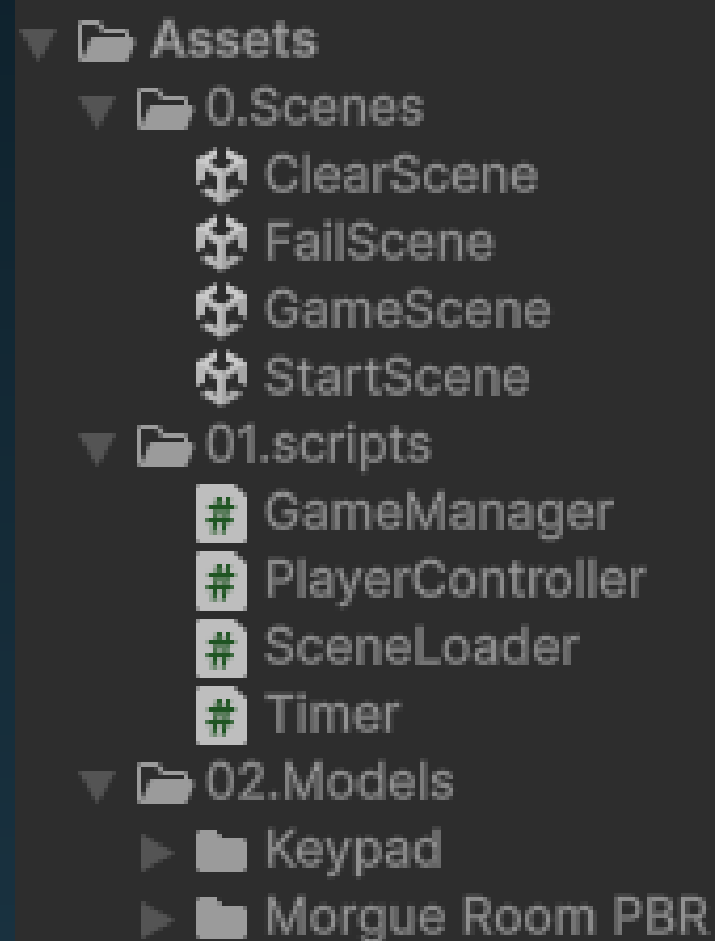
남은 시간: 180



02.

게임 구성

전체적인 폴더 구조



03. 핵심 코드



GameManager.cs

```
1 using UnityEngine;
2 using UnityEngine.SceneManagement; // 이 줄을 추가합니다.
3
4
5 // Unity 스크립트(자산 참조 1개)|참조 2개
6 public class GameManager : MonoBehaviour
7 {
8     public float timeLimit = 180f; // 5분
9     public Timer timer;
10
11     // Unity 메시지|참조 0개
12     void Start()
13     {
14         timer.StartTimer(timeLimit);
15     }
16
17     참조 0개
18     public void GameClear()
19     {
20         timer.StopTimer();
21         Debug.Log("탈출 성공!");
22         SceneManager.LoadScene("ClearScene");
23     }
24
25     참조 1개
26     public void GameOver()
27     {
28         Debug.Log("시간 초과! 탈출 실패.");
29         SceneManager.LoadScene("FailScene");
30     }
31 }
```



PlayerController.cs

```
1 using UnityEngine;
2 // using NavKeyPad; // 이제 Keypad, KeypadButton을 직접 사용하지 않으므로 이 네임스페이스는 필요 없을 수 있습니다.
3 // using System; // 이제 StringComparison.OrdinalIgnoreCase가 필요 없으므로 제거해도 됩니다.
4
5 [RequireComponent(typeof(CharacterController))]
6 // Unity 스크립트(자산 참조 1개)|참조 0개
7 public class PlayerController : MonoBehaviour
8 {
9     public float speed = 5f;
10     public float mouseSensitivity = 2f;
11     public Camera playerCamera; // 플레이어 카메라 연결 (이동/회전에는 여전히 필요)
12
13     // --- 상호작용 관련 변수 제거 또는 주석 처리 ---
14     // public float interactionDistance = 3f; // 이 변수는 더 이상 PlayerController에서 사용되지 않습니다.
15
16     private CharacterController controller;
17     private float xRotation = 0f;
18
19     // Unity 메시지|참조 0개
20     void Start()
21     {
22         controller = GetComponent<CharacterController>();
23         // 마우스 커서 상태는 이제 상호작용 스크립트나 다른 곳에서 관리할 수 있습니다.
24         // 여기서는 기본적으로 보이게 두거나, 게임 시작 시 어떻게 할지 결정합니다.
25         // Cursor.lockState = CursorLockMode.Locked; // 만약 마우스 커서를 보이게 하고 싶다면 주석 유지 또는 삭제
26         Cursor.lockState = CursorLockMode.None; // 마우스 커서를 보이게 설정
27         Cursor.visible = true; // 마우스 커서 보이도록 설정
28     }
29
30     // Unity 메시지|참조 0개
31     void Update()
32     {
33         // 카메라 회전 (이동 모드일 때만 적용되도록 나중에 수정 가능)
34         float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity;
35         float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity;
36
37         xRotation -= mouseY;
38         xRotation = Mathf.Clamp(xRotation, -90f, 90f);
39         playerCamera.transform.localRotation = Quaternion.Euler(xRotation, 0, 0);
40         transform.Rotate(Vector3.up * mouseX);
41
42         // 이동 (이동 모드일 때만 적용되도록 나중에 수정 가능)
43         float h = Input.GetAxis("Horizontal");
44         float v = Input.GetAxis("Vertical");
45
46         Vector3 move = transform.right * h + transform.forward * v;
47         controller.Move(move * speed * Time.deltaTime);
48     }
49 }
```

03. 핵심 코드



SceneLoader.cs

```
1 using UnityEngine;
2 using UnityEngine.SceneManagement; // 씬 관리를 위해
3
4 // Unity 스크립트(자산 참조 4개) | 참조 0개
5 public class SceneLoader : MonoBehaviour
6 {
7     // 원하는 씬의 이름을 Inspector에서 설정할 수 있음
8     // 시작 씬의 이름은 Unity Build Settings에서 확인
9     public string sceneNameToLoad = "GameScene"; // <
10
11     // 이 메소드는 버튼의 OnClick() 이벤트에 연결하여
12     // 참조 0개
13     public void LoadTargetScene()
14     {
15         // 지정된 이름의 씬을 로드합니다.
16         SceneManager.LoadScene(sceneNameToLoad);
17     }
18 }
```



Timer.cs

```
2 using UnityEngine.UI;
3
4 // Unity 스크립트(자산 참조 1개) | 참조 1개
5 public class Timer : MonoBehaviour
6 {
7     public Text timerText;
8     private float remainingTime;
9     private bool isRunning = false;
10     private GameManager gm;
11
12     // Unity 메시지 | 참조 0개
13     void Start()
14     {
15         gm = FindObjectOfType<GameManager>();
16     }
17
18     참조 1개
19     public void StartTimer(float seconds)
20     {
21         remainingTime = seconds;
22         isRunning = true;
23     }
24
25     참조 1개
26     public void StopTimer()
27     {
28         isRunning = false;
29     }
30
31     // Unity 메시지 | 참조 0개
32     void Update()
33     {
34         if (!isRunning) return;
35
36         remainingTime -= Time.deltaTime;
37
38         // --- 이 부분을 수정합니다 ---
39         // 남은 시간 값 뒤에 "초"를 붙입니다.
40         timerText.text = "남은 시간: " + Mathf.CeilToInt(remainingTime).ToString();
41         // -----
42
43         if (remainingTime <= 0)
44         {
45             isRunning = false;
46             gm.GameOver();
47         }
48     }
49 }
```

04.

시현 영상

Room Escape

Tap to start