

*Estate
Escape*



1688004 김도겸

목차

게임 기획

1. 게임 개요
2. 제작의도
3. 게임 컨셉
4. 게임 특징
5. 시놉시스
6. 게임 조작법
7. 게임 규칙

게임 구현

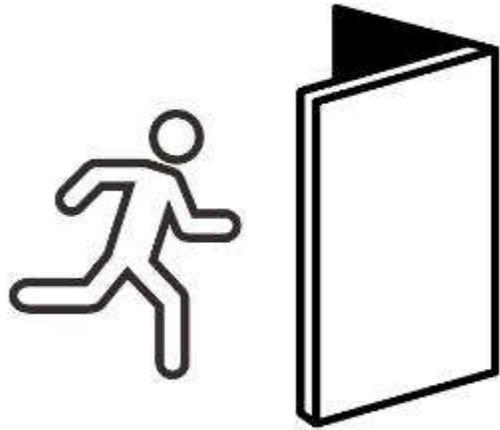
1. 디자인
2. 주요 스크립트

마무리

1. 소감
2. QnA

게임 개요

제목	Estate Escape
플랫폼	PC (Windows)
개발도구 및 언어	Unity3D / C#
장르	퍼즐(방 탈출)
시점	1인칭



제작 의도

많지 않은 단일 엔딩이 아닌 퍼즐 게임을 만들고 싶었으며,

그러면서도 간단하고 쉽게 즐길 수 있는 방 탈출 식 게임을

만들고 싶어 만들게 되었습니다.

게임 컨셉

- ▶ 메인 컨셉: 탈출

정해진 스테이지속 문제를 해결 하고, 탈출 하는 것이 게임의 주 목적입니다.

- ▶ 서브 컨셉1 : 오래된 저택

플레이어의 주무대가 되는 공간입니다. 플레이어는 이곳에서 탈출을 하게 됩니다.

- ▶ 서브 컨셉2 : 퍼즐

플레이어가 탈출하기 위해서는 주어진 문제. 즉, 퍼즐을 풀어야 합니다.

- ▶ 서브 컨셉3 : 추리

플레이어는 주어진 단서를 통해 퍼즐을 풀기 위한 추리를 해야만합니다.

게임특징

-멀티 엔딩



★ 여러가지 엔딩:
플레이어가 어떤 특수한 기믹을 찾는가에 따라
여러 엔딩을 볼 수 있습니다.

게임특징

-간결한 UI

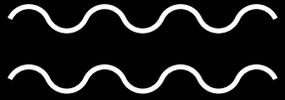


★ UI의 간결함:
별도의 UI가 따로 존재 하지 않으며, 필요한 때에만
UI가 가끔 나옵니다.

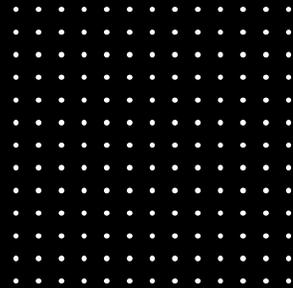
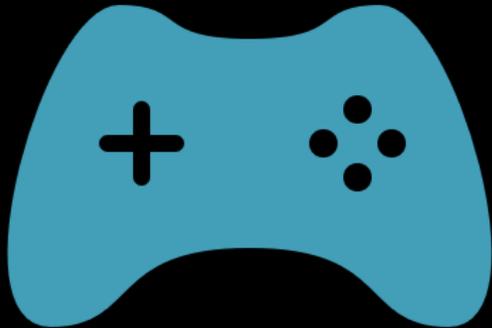
게임 특징

- 가볍게 플레이

★ 가볍게 즐길 수 있다:
플레이 타임이 짧기 때문에,
적은 시간을 활용하여 쉽고 가볍게 즐길 수 있습니다.

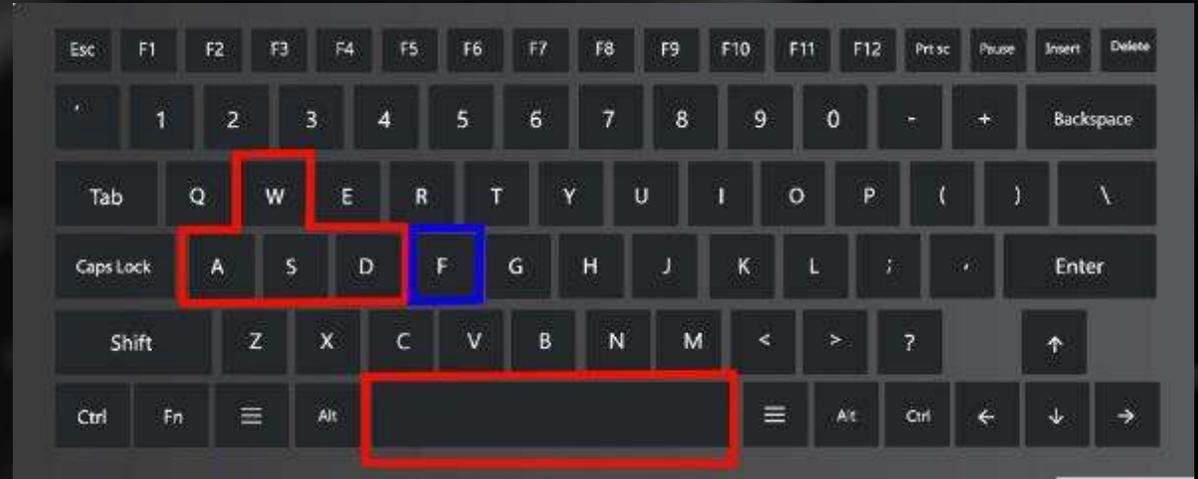


시/놉시/스



- 시대적 배경 : 빅토리안 시대
- 공간적 배경 : 기묘한 터널, 저택
- 내용:
옛날 어느 마을, 시골의 작은 귀족의 한 소녀가 있었다.
그 소녀 엘마는 마을에서 호기심이 많기로 소문이 자자했다.
엘마는 어느 날 산책을 나서던 중 어떤 신비한 동굴에
떨어졌다.
정신을 차리고 보니, 익숙하지만 무언가 이상한 방에 있었다.
소녀는 집에 늦게 들어가면 혼날 것 을 염려해 그 곳을
나가고자
수수께끼의 문제 그 자체인 이 곳을 탈출 하고자 한다.

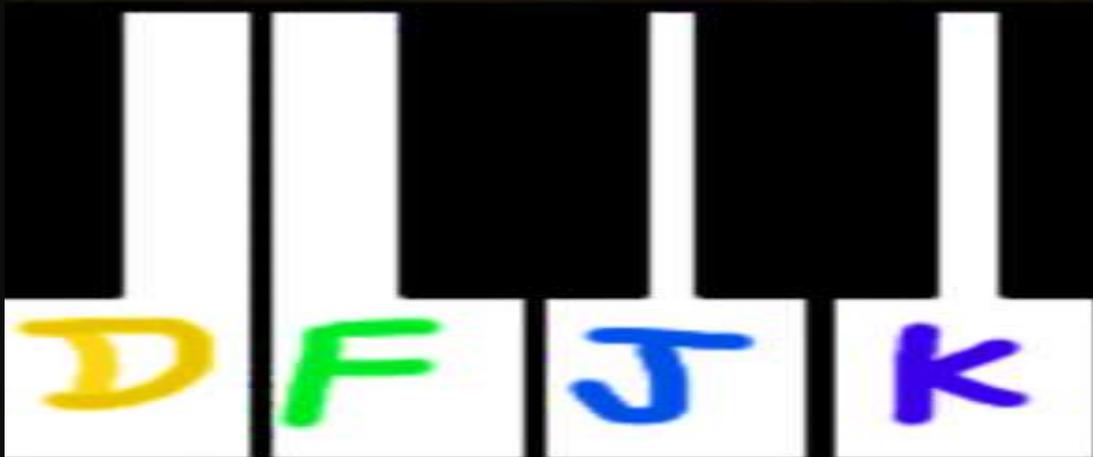
게임 조작법

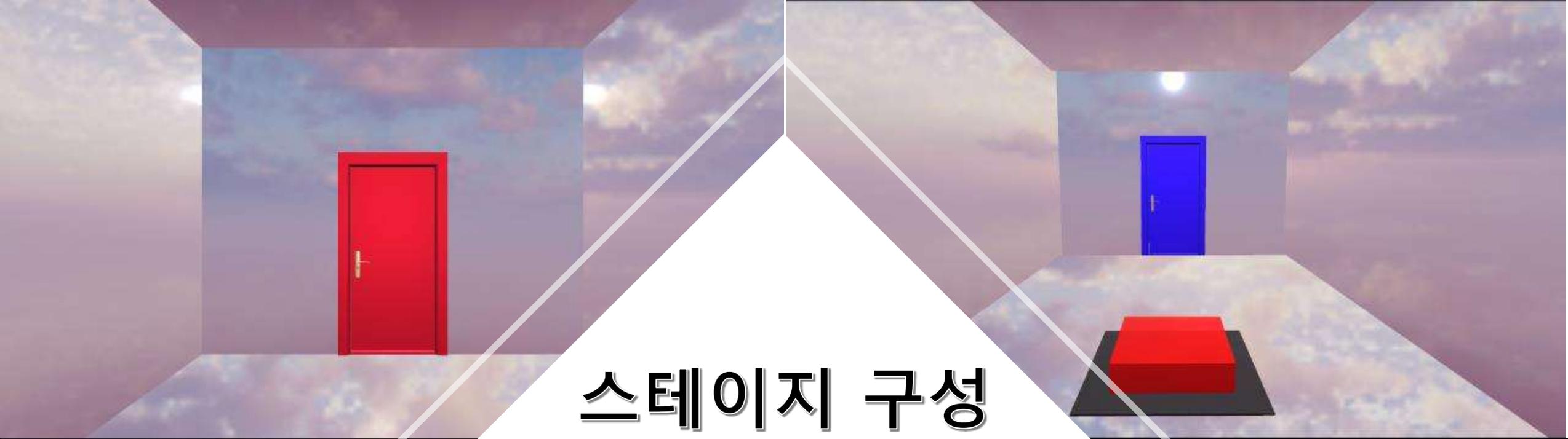


- 키보드_WASD - 이동키
- 키보드_Left Shift - 달리기
- 키보드 Space Bar - 점프
- 키보드 F key - 상호작용
- 마우스 커서 - 카메라 회전

게임 규칙

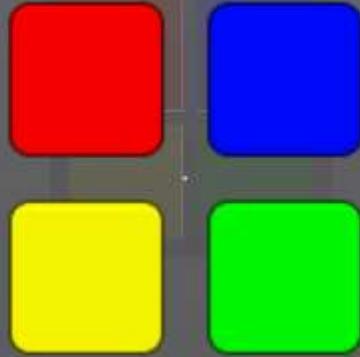
- 플레이어는 각 스테이지 속, 단서를 유추해내어
 - 각 스테이지의 방의 문을 찾아 내어 문을 열어 클리어 하는 것이 목적입니다.
 - 통상 단서 외 숨겨진 요소와 단서들을 찾아내어
 - 특정 기믹을 성공 시켜, 각 스테이지마다 있는 히든 스테이지 혹은 히든 기믹을 열어서, 다른 엔딩을 볼 수도 있습니다.
-





스태이지 구성

【튜토리얼】



Ⓢ Enter
PassWord



A S D F J K L ;

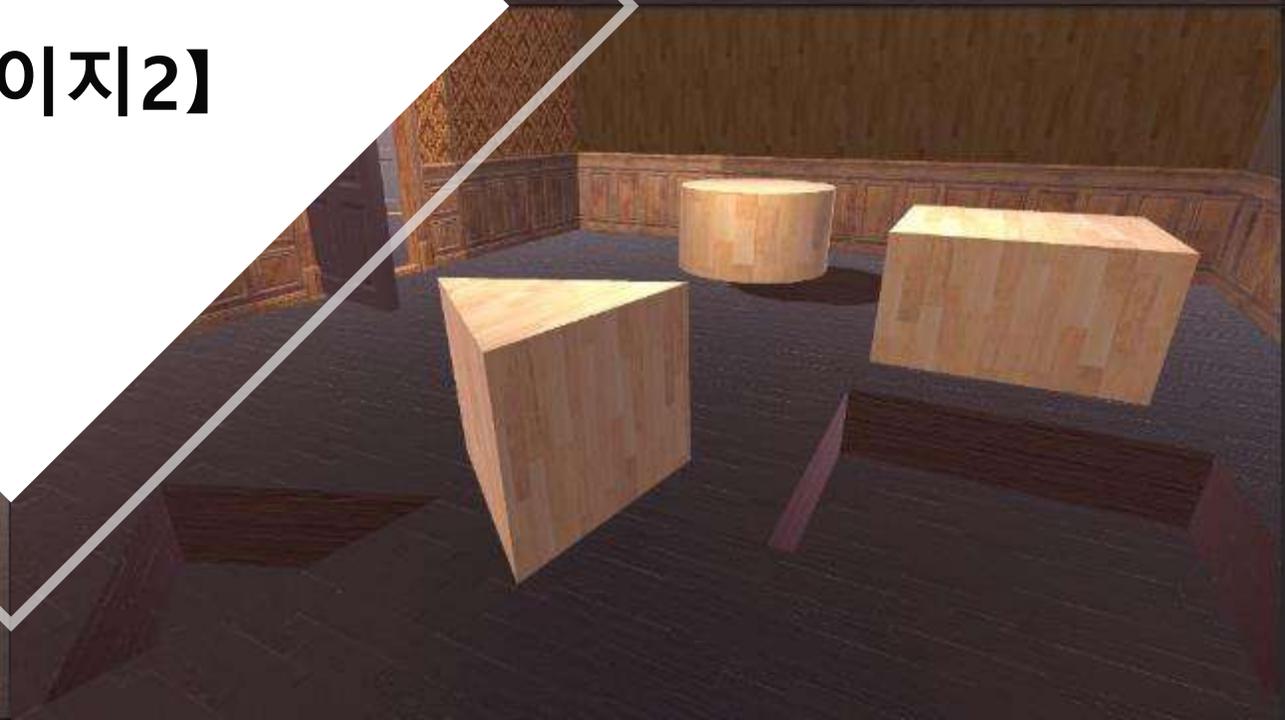
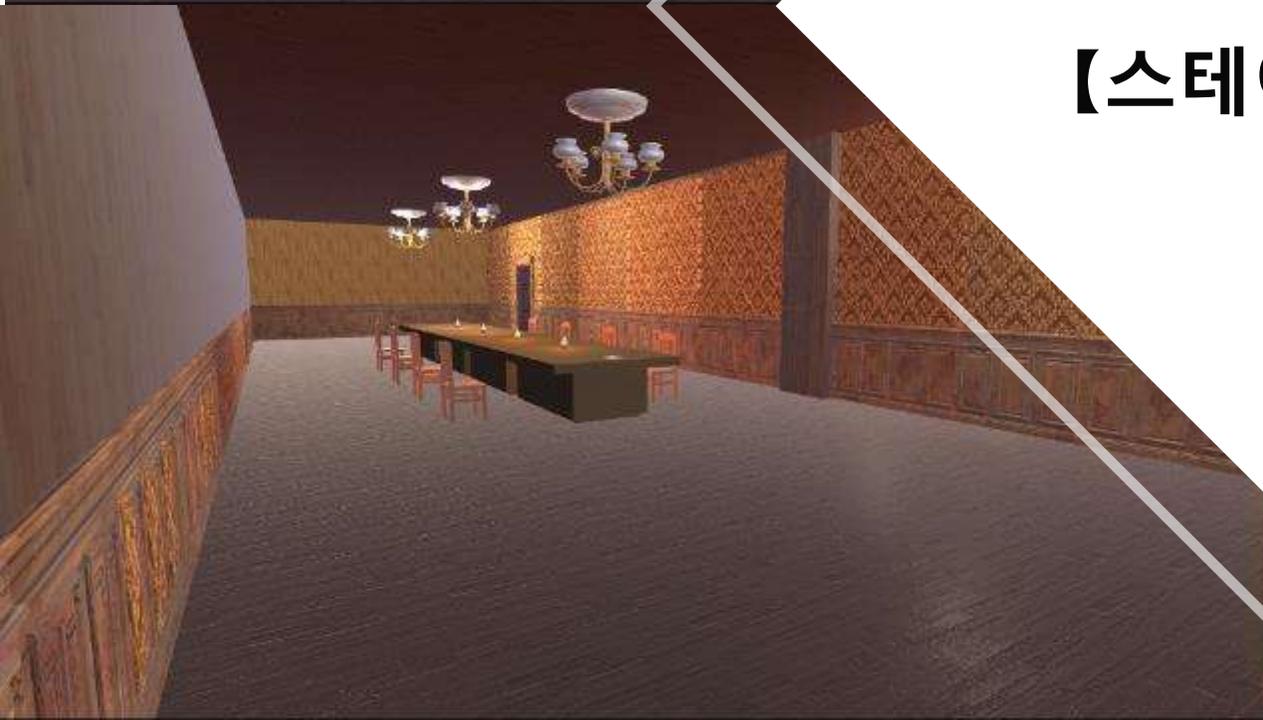


스테이지 구성
【스테이지1】



스태이지 구성

【스태이지2】



주요 스크립트 – StageManager.cs

```
if (Stage1_Clear == true)
    Hint_trigger.SetActive(true);

if (HiddenClear_Counter == 3)
{
    Stage2_Hidden = true;
}

if (Clear_Counter == 3 && HiddenClear_Counter < 3)
{
    FinStageDoor.SetBool("Open", true);
}

if (Stage1_Hidden == true && Stage2_Hidden == true && ClearYet == true)
{
    AnotherFinDoor.SetBool("DoorOpen", true);
    Talk_Pannel.SetActive(true);
    ClearYet = false;
    player_Text.text = "윗층 방의 문의 잠금이 열렸다." + "\n" + "[ESC를 눌러 대화창 닫기]";

    if (Input.GetKeyDown(KeyCode.Escape))
    {
        Talk_Pannel.SetActive(false);
    }
}
```

```
참조 1개
void Stage1()
{
    //사계카운터가 카운트가 13일시 1로 초기화
    if (WinterCounter >= 13)
        WinterCounter = 1;

    if (SummerCounter >= 13)
        SummerCounter = 1;

    if (FallCounter >= 13)
        FallCounter = 1;

    if (WinterCounter == WinterClearCount && SummerCounter == SummerClearCount && FallCounter == FallClearCount)
    {
        StageDoorTrigger[0].GetComponent<Animator>().SetBool("MoveOn", true);
        Stage1_Clear = true;
    }

    //히든 클리어 조건
    if (WinterCounter == 5 && SummerCounter == 3 && FallCounter == 6 && SpringCounter == 12)
    {
        StageDoorTrigger[0].GetComponent<Animator>().SetBool("MoveOn", true);
        Stage1_Clear = true;
        Hidden_Open1 = true;
    }
}
```

업데이트 – 엔딩 부분

스태이지 1

주요 스크립트 – StageManager.cs

참조 1개

```
void Stage2()
{
    //1번방
    if(Mission1_Clear_Counter < 0)
    {
        for(int i = 0; i < Figure_Objects.Length; i++)
        {
            Figure_Objects[i].transform.position = Figure_Objects_Position[i];
        }

        Mission1_Clear_Counter = 0;
    }

    if (Mission1_Clear_Counter == 3)
    {
        Clear_Counter++;
        Mission1_Clear_Counter++; //3일때 계속 올라가기 때문에 한번만 올라게 하기 위해 숫자를 올려, 브레이크를 줌.

        if (HiddenClear_Counter == 2)
        {
            HiddenClear_Counter++;
        }
        else
        {
            HiddenClear_Counter = 0;
        }
    }

    //2번방
    if(Mission2_Clear_Counter == 4)
    {
        Clear_Counter++;
        Mission2_Clear_Counter++;

        if (HiddenClear_Counter == 0)
        {
            HiddenClear_Counter++;
        }
        else
        {
            HiddenClear_Counter = 0;
        }
    }

    //3번방
    if (SpringCounter >= 13)
        SpringCounter = 1;

    if (SpringCounter == SpringClearCount && WinterCounter == 4 && SummerCounter == 8 && FallCounter == 6)
    {
        Clear_Counter++;

        SpringCounter = 0;
        WinterCounter = 0;
        SummerCounter = 0;
        FallCounter = 0;

        if(HiddenClear_Counter == 1)
        {
            HiddenClear_Counter++;
        }
        else
        {
            HiddenClear_Counter = 0;
        }
    }
}
```

스태이지 2

주요 스크립트 - Button_Object.cs

☞ Unity 메시지 | 참조 0개

```
private void OnCollisionStay(Collision other)
{
    if (other.gameObject == player)
    {
        if (Push_UI.isActiveInHierarchy == false)
        {
            Push_UI.SetActive(true);
        }

        if (Input.GetKeyDown(KeyCode.F))
        {
            AudioManager.instance.SFXPlay("버튼 소리", clip[0]);
            Stage01_ButtonTag_Event();
        }
    }
}
```

☞ Unity 메시지 | 참조 0개

```
private void OnCollisionExit(Collision collision)
{
    Push_UI.SetActive(false);
}
```

참조 1개

```
void Stage01_ButtonTag_Event()
{
    if (this.gameObject.tag == "WinterBook")
    {
        stageManager.WinterCounter++;
    }
    else if (Object_Button.tag == "SummerBook")
    {
        stageManager.SummerCounter++;
    }
    else if (Object_Button.tag == "AutumnBook")
    {
        stageManager.FallCounter++;
    }
    else if (Object_Button.tag == "SpringBook")
    {
        stageManager.SpringCounter++;
    }
    else
    {
        print(this.gameObject.tag);
    }
}
```

주요 스크립트 - Clock.cs

```
Unity 스크립트(사실 참조 5개, 참조 0개)
public class Clock : MonoBehaviour
{
    [Range(0, 3)]
    public int seseason_Number = 0; // 0 = 봄, 1 = 여름, 2 = 가을, 3 = 겨울

    public Animator Clock_animator;

    public StageManager Stage_manager;

    // Update is called once per frame
    @ Unity 메시지 참조 0개
    void Update()
    {
        if (seseason_Number == 0)
            Clock_animator.SetInteger("ClockCount", Stage_manager.SpringCounter);

        if (seseason_Number == 1)
            Clock_animator.SetInteger("ClockCount", Stage_manager.SummerCounter);

        if (seseason_Number == 2)
            Clock_animator.SetInteger("ClockCount", Stage_manager.FallCounter);

        if (seseason_Number == 3)
            Clock_animator.SetInteger("ClockCount", Stage_manager.WinterCounter);
    }
}
```



스태이지 1

주요 스크립트 - Chairs.cs / Stage2_Collide_Mission.cs

```
© Unity 메시지 참조 0개
private void Update()
{
    if (stg_Manager.Mission2_Clear_Counter < 0)
    {
        stg_Manager.Mission2_Clear_Counter = 0;
    }

    if(stg_Manager.Mission2_Clear_Counter >= 4)
    {
        Chair.SetBool("Active", false);
    }
}
```

```
© Unity 메시지 참조 0개
private void OnCollisionStay(Collision other)
{
    if (other.gameObject.CompareTag("Player"))
    {
        if (Input.GetKeyDown(KeyCode.F))
        {
            Active_Chair();

            if (Chair_number == 1 || Chair_number == 5 || Chair_number == 6 || Chair_number == 3) //활성화시
            {
                if (Chair.GetBool("Active") == true) // 활성화
                {
                    stg_Manager.Mission2_Clear_Counter++;
                }
                else if (Chair.GetBool("Active") == false) //비활성화
                {
                    stg_Manager.Mission2_Clear_Counter--;
                }
            }
        }
    }
}
```

```
참조 1개
private void Active_Chair()
{
    if(Chair.GetBool("Active") == false)
    {
        Chair.SetBool("Active", true);
    }
    else if(Chair.GetBool("Active") == true)
    {
        Chair.SetBool("Active", false);
    }
}
```

```
if(My_Number == 1)
{
    if (collision.gameObject.name == "Cylinder")
    {
        if(stg_Manager.Mission1_Clear_Counter == 0)
        {
            stg_Manager.Mission1_Clear_Counter++;
            stg_Manager.Figure_Objects[0].transform.position = new Vector3(-100, -100, -100);
        }
        else
        {
            stg_Manager.Mission1_Clear_Counter = -1;
        }
    }
}

if(My_Number == 2)
{
    if(collision.gameObject.name == "Cube")
    {
        if (stg_Manager.Mission1_Clear_Counter == 2)
        {
            stg_Manager.Mission1_Clear_Counter++;
            stg_Manager.Figure_Objects[1].transform.position = new Vector3(-100, -100, -100);
        }
        else
        {
            stg_Manager.Mission1_Clear_Counter = -1;
        }
    }
}

if(My_Number == 3)
{
    if (collision.gameObject.name == "TriBox")
    {
        if (stg_Manager.Mission1_Clear_Counter == 1)
        {
            stg_Manager.Mission1_Clear_Counter++;
            stg_Manager.Figure_Objects[2].transform.position = new Vector3(-100, -100, -100);
        }
        else
        {
            stg_Manager.Mission1_Clear_Counter = -1;
        }
    }
}
```

주요 스크립트 - Chairs.cs / Stage2_Collide_Mission.cs



스테이지 2

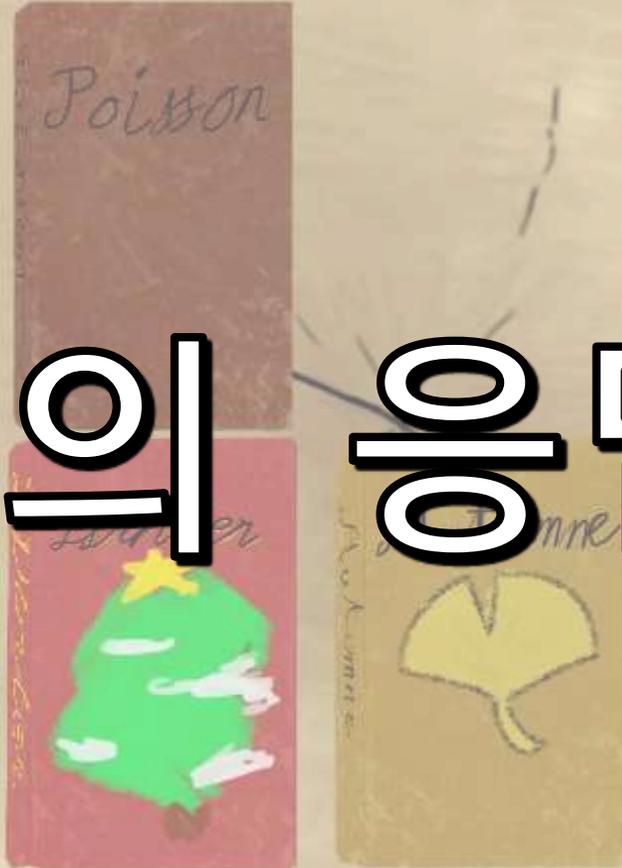
소감

일단 끝내고 나니, 후련한 기분이 들면서 한편으로는 퀄리티적인 부분에서 생각보다 구현이 덜 되었고, 버그도 다 고치지 못하여 많이 아쉽습니다.

처음 기획하고 구연 했던 것의 거의 모든 것을 다 뜯어 고쳤기에, 생각 보다 시간이 부족했던 것 같습니다.

교수님의 도움, 많은 지인들의 조언을 통해 많은 영감을 받아, 겨우겨우 완성 할 수 있었던 것 같습니다.

질의 응답





Thank you