

CUBE ESCAPE :Death game

3주차 전달사항

목차

1. 2주차 목표 점검 및 3주차 목표 설정.
2. SOUND 에 관하여..
3. 코딩방식 정리
4. 그래픽 확인

2주차 목표(7.12~7.17, 총6일)

☑ STAGE 1~4 최종마무리 (디버깅,효과음추가, 부가기능추가 등등)

☑ STAGE 5~7 서브기획 및 최종기획서 제출

☐ STAGE 5~7 그래픽 일부 제작.

3주차 목표(7.18~7.24, 총7일)

- STAGE 5~8 코딩
- STAGE 5~7 그래픽 제작.

SOUND 관련..

1) STAGE별로 사용한 모든 BGM/효과음 표기해놓기

무료 BGM/효과음의 경우에는 꼭 저작자 표시를
명시할 것을 부탁하는 경우가 많더라고요...! 그래
서 저작자분들 한페이지에 다 기록해놨다가, 게임
마지막부분 or 게임 배포할 페이지에 적어두어야
할 것 같아요..

진짜 흑~시라도 게임 최종 빌드 후 배포시에 저
작권 컴플레인이 들어오게 되면 정말 최악의 상황
이 발생할 수도 있으니.. 미리미리 주의합시다!

The screenshot shows the 'sound' application interface. The top navigation bar includes a search icon, 'sound' logo, and tabs for 'List', 'Board', 'STAGE BGM 2', 'Form', 'Table', and '+ View'. There are also buttons for 'Automate (0)' and 'Share'. The left sidebar shows a list of 'PAGES' from 'STAGE 1' to 'STAGE 8', with 'STAGE 3' selected. The main content area is titled 'STAGE 3' and contains the following text:

Heading 1 ▾ | **B** | *I* | U | S | [List Icon] | [Checkmark Icon] | [Code Icon] | [Link Icon] | [Image Icon] | **A** ▾ | [Color Picker Icon] | [List Icon] | [Plus Icon] | [More Icon]

-fireplace

Thanks to, ezwa

From: <http://soundbible.com/1543-Fireplace.html>

Distributor: 저작권 걱정없는 유튜브용 음원 & 무료 효과음 서비스 '뮤팟'

<https://www.mewpot.com>

프로젝트 페이지
SOUND->STAGE BGM 에서 확인 가능!

+ 저작자 표시를 할 필요가 없는 BGM/효과음
의 경우는 안적어도 되는게 아니라 **STAGE에 사
용한 모든 BGM/효과음**을 적어주세요!

Why? 지금은 초반이라 뭐 사용했는지 찾아보는게 그렇
게 번거롭지는 않은데 stage가 점점 많아질수록 찾기가 힘들
어질거같아서..

코딩방식통일

1. Interaction controller /StageGameManager ->필수 스크립트!

:Hierarchy 창에서 아래와 같이 빈 오브젝트로 만들어 관리하기.



InteractionController

-> interaction controller 스크립트 적용.



GameManager

->stage game manager 스크립트 적용.

-Interaction controller

```
Unity 스크립트 | 참조 0개
public class InteractionController2 : MonoBehaviour
{
    [SerializeField]
    private Camera cam;

    RaycastHit hitInfo;

    [SerializeField]
    private int sizeofLazer;

    [SerializeField]
    private GameObject[] interactionUI;

    [SerializeField]
    private Transform arm;

    [SerializeField]
    private GameObject ElectricBoxUI;

    [SerializeField]
    private GameObject EnglishKeypadUI;

    [SerializeField]
    private GameObject[] PaintsUI;

    [SerializeField]
    private GameObject[] spotLights;

    [SerializeField]
    private GameObject KeypadUI;

    [SerializeField]
    private Stage2GameManager gameManager;

    [SerializeField]
    private SceneManagement sceneManagement;

    private int state;
    private bool is_electricbox_open = false;

    // Update is called once per frame
    Unity 메시지 | 참조 0개
```

```
else if(hitInfo.transform.name == "Death")
{
    if (state == 4)
    {
        interactionUI[0].GetComponent<Text>().text = "확대한다.";

        if (Input.GetMouseButtonDown(0))
        {
            GameObject.Find("Player").GetComponent<MovePlayer2>().enabled = false;
            if (spotLights[0].GetComponent<ChangeSpotLight>().GetCount() == 2)
            {
                PaintsUI[1].SetActive(true);
            }
            else
            {
                PaintsUI[0].SetActive(true);
            }
        }
    }
}
```

:클릭했을때 띄워야하는 ui가 있는 사물들 (ex.일기장,키패드,자물쇠,키패드,전선함 등)를 일단 몽땅 [SerializeField]로 모두 선언해놓고 클릭하면 SetActive(true)시키는 방식이용하기!
+ 되도록이면 ui띄울때는 플레이어/화면 움직임 멈추기.

<-> 상호작용이 있는 각각의 사물에 대해 스크립트를 만들고 voidOnMouseDown을 이용하면 스크립트 수가 많아져서 관리하기 힘들거같음!ㅠㅠ

Q. 그럼 SetActive(false)는
어떻게 처리하죠?

A. 오른쪽처럼 마우스 오른쪽 클릭시
SetActive(false)를 수행하는
Quit라는 클래스를 만들어서
꺼지는 기능이 필요한 UI(일기장, 그림,
책 등등)에 적용시키기.
스크립트를 공용으로 적용시켜주시면
굳이 사물마다 script 만들필요X.

+아 물론 자물쇠,키패드,전선편 같이 복잡하게 동작하는
UI들은 Quit 클래스 적용시키지 않고 그 사물에 관한 클래스
만들어서 그 안에서 꺼지도록 구현해도 전혀 상관없습니다!!
이건 그냥 벵들이 코딩을 더 편하게 했으면 좋겠어서..

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

public class Quit : MonoBehaviour
{
    [SerializeField]
    private GameObject player;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetMouseButtonDown(1))
        {
            Cursor.lockState = CursorLockMode.Locked;
            player.GetComponent<MovePlayer2>().enabled = true;
            gameObject.SetActive(false);
        }
    }
}
```

-StageGameManager

:여러가지 게임 탈출 조건 변수 선언(Public static bool 타입)

->사실 저는 이방식이 가장 직관적이고 편할거라고 생각해서 그런건데 혹시라도 다른 방식 (int타입 state활용) 이 더 편하시면 편하신 대로 해주셔도 전혀 상관없습니당!!!!

Unity 스크립트 | 참조 18개

```
public class Stage3GameManager : MonoBehaviour
```

```
{
```

```
    public static bool isSecretBoxUnlocked; //비밀상자 해제시 true로 바꿔줄것.  
    public static bool isClockFired;  
    public static bool isDecorateFired;  
    public static bool isFamilyPictureFired;
```

```
    //3개의 물건을 벽난로에 다 태우면 clear
```

```
    public static bool isStageCleared;
```

Unity 스크립트 | 참조 13개

```
public class Stage1GameManager : MonoBehaviour
```

```
{
```

```
    public static bool doesPlayerhavekey = false;
```

```
    public static int does1stSolved = -1; //첫번째 문제(큐브 문제) 해결 여부 - CodePanel과 연결 / -1이면 1번 문제 해결 x 0이면 해결후 변화 실행
```

```
    public static int does2ndSolved = -1; //두번째 문제(큐브 숫자) 해결 여부 - CodeLock과 연결
```

```
    public static bool doesDialog1End = false; //dialog manager와 연결
```

```
    public static bool doesDialog2End = false; //dialog manager와 연결
```

```
    public static bool issafeboxunlocked = false; //그림금고 비밀번호 해제했는지 여부.
```

```
    public static bool isWhalepadUnlocked = false; //고래비밀번호 입력기 해제했는지 여부
```

```
    public static bool isliarQuizSolved = false; // 진실을 말하는 피노키오한테 꽃을 줘서 문제를 풀었는지 여부.
```

그리고 update함수에서는 선언한 변수를 활용하여
그 조건 충족시 특정 이벤트가 실행되도록!

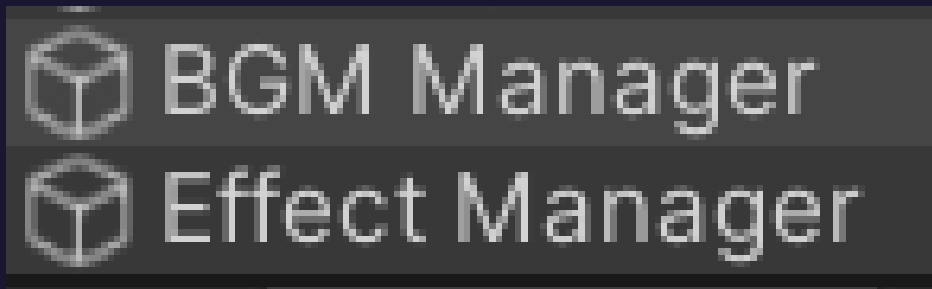
```
}  
  
if(doesIntroEnd1 == true)  
{  
    doesDialogEnd = false;  
    StartCoroutine("StageIntro2"); //다시 페이드 아웃 - 플레이어 2로 변경 - 페이드 인  
}  
  
if (doesIntroEnd2 == true)  
{  
    doesIntroEnd1 = false;  
    fadeInOutPanel.SetActive(false);  
    fadeInOutPanel2.SetActive(false);  
  
    if(gameControlFlag == 0)
```

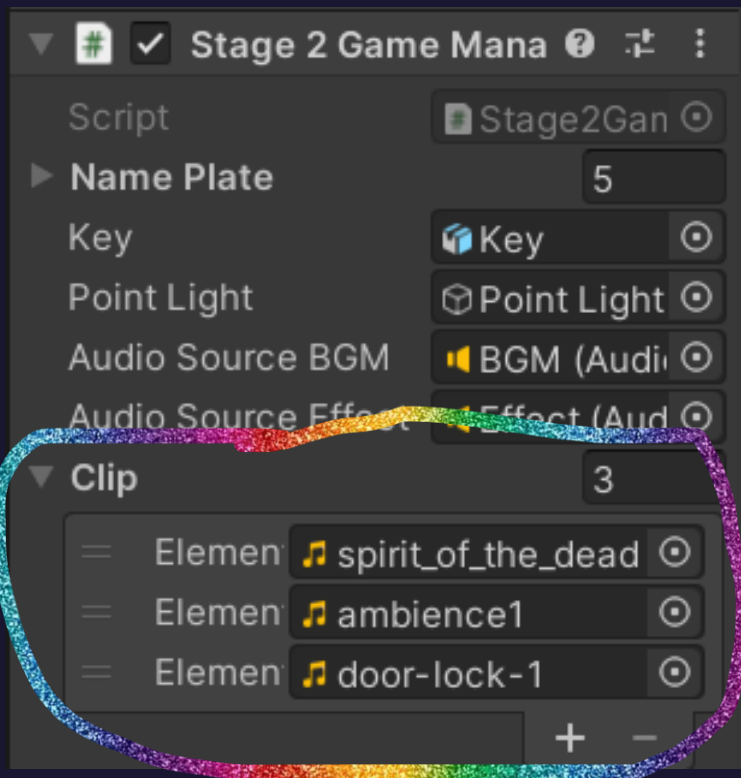
```
if (!clock.activeInHierarchy)  
    isClockFired = true;  
  
if (!decorate.activeInHierarchy)  
    isDecorateFired = true;  
  
if (!familypicture.activeInHierarchy)  
    isFamilyPictureFired = true;  
  
if (isClockFired && isDecorateFired && isFamilyPictureFired)  
{  
    isStageCleared = true;  
    // 엔딩후 다음스테이지로.  
  
    SceneManager.completedStage = 3;  
    sm3.movetoNextStage();  
}  
  
if (isNewsviewed && b)  
{  
    StartCoroutine("secretBoxOpen");  
  
    b = false;
```

2. AudioSource는 BGM/Effect 두개만 두기!

(아마 세개의 음향이 중첩되어야 할 일은 거의 없을 것이라고 생각하는데 혹시나 있으면 그땐 어쩔수 없이 AudioSource 한대 더 두셔도 무방...)

:각각의 사물에 AudioSource를 적용시키면 나중에 효과음/BGM 변경할 일 생길때 위치찾기가 넘 번거로울거 같아서..?



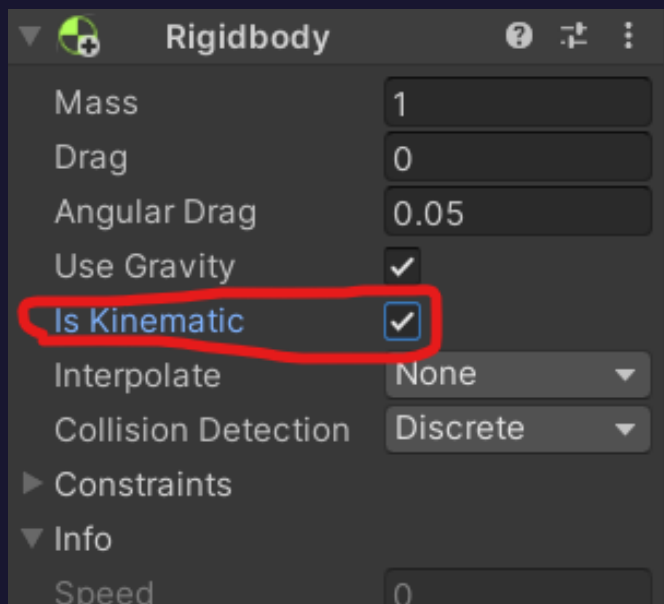


```
[SerializeField]  
private AudioClip[] clip;
```

```
else if(state == 4 && b1)  
{  
    pointLight.GetComponent<Light>().enabled = false;  
    audioSourceBGM.clip = clip[0];  
    audioSourceBGM.Play();  
    audioSourceEffect.clip = clip[1];  
    audioSourceEffect.Play();  
    b1 = false;  
}  
  
else if(state == 5 && b2)  
{  
    audioSourceEffect.clip = clip[2];  
    audioSourceEffect.Play();  
    b2 = false;  
}
```

그리고 BGM/ 효과음을 바꿔야 할일이 생기면 Audio Source를 한대 더 만드 는게 아니라 Clip배열을 선언하시고, 기 존 Audio Source에서 Clip(음원)을 바꿔주는 방법으로 해주세요! 사진은 주이님 코드 참고 ..ㅎㅎ ><

3. 게임내의 모든 사물들은 Rigidbody-> isKinematic 체크하기!



Otherwise, 그냥 플레이어와의 물리적 상호작용(충돌 등)으로 사물들이 움직여버림.. 그래서 위치 고정 필요한 사물의 경우 체크해놓는게 좋을듯!

IsKinematic = script에 의해서만 움직일수 있도록 하겠다.

그래픽 확인

이
:혹시 빠진것이 있다면 프로젝트 페이지 Graphic 칸에 task 추가해주
세요 제가 확인할게요!

▼ ✓ [STAGE 5] THE RIVER OF OBLIVION 그래픽 2 | +

✓ 보물지도

✓ 일기장

김

김

▼ ✓ [STAGE 6] THE NIGHTMARE 그래픽 1 | +

✓ 드림캐쳐도안

황

▼ ✓ [STAGE 7] THE REVERSE 그래픽 1 | +

✓ 숫자 4개

김

▼ ✓ [STAGE 8] THE UNCONCIOUS 그래픽 3 | +

✓ 아동학대 관련 사진

✓ 장례식 관련 사진

✓ 유치원 왕따 관련 사진