

# 게임 기획서

# 목차

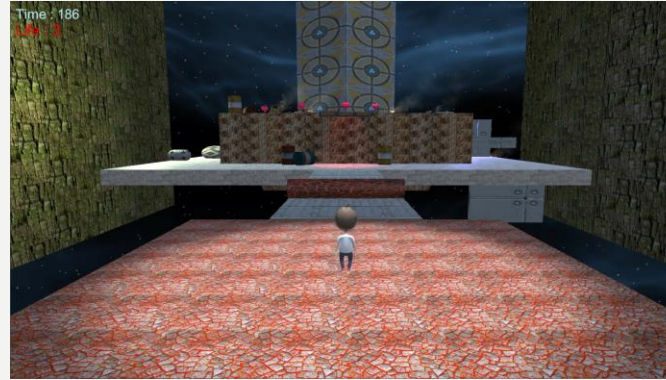
1. 씬 추가
2. 스테이지 추가
3. 기능 추가
4. 배경 추가

# 1. 썸 추가

# SPEED RUNNER'S SCENE



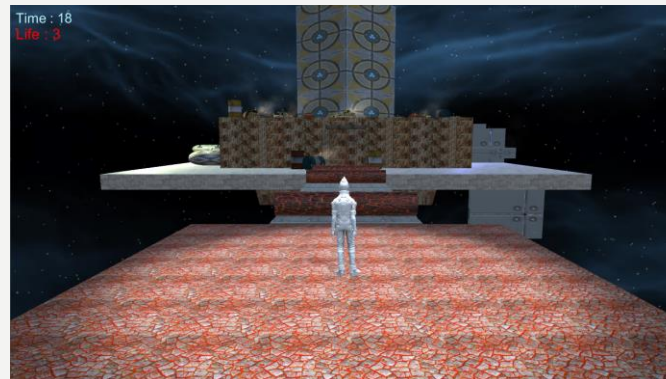
Start scene



Main scene



Ending scene



Hard scene



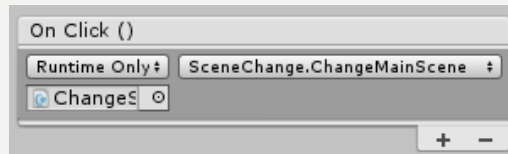
Ending2 scene

# SPEED RUNNER'S SCENE

## 씬 전환 방법

1) Application.LoadLevel("Ending");  
// 씬 이름

2) 버튼의 On Click() 이벤트에 함수 이름 등록



## SceneChange.cs 주요 코드

```
using UnityEngine.SceneManagement;

참조 0개
public class SceneChange : MonoBehaviour {
    참조 0개
    public void ChangeStartScene()
    {
        SceneManager.LoadScene("Start");
    }
    참조 0개
    public void ChangeMainScene()
    {
        SceneManager.LoadScene("Main");
    }
    참조 0개
    public void ChangeHardScene()
    {
        SceneManager.LoadScene("Hard");
    }
    참조 0개
    public void ChangeEndingScene()
    {
        SceneManager.LoadScene("Ending");
    }
    참조 0개
    public void ChangeEnding2Scene()
    {
        SceneManager.LoadScene("Ending2");
    }
}
```

## 2. 스테이지 추가

# 1) EASY MODE



Character



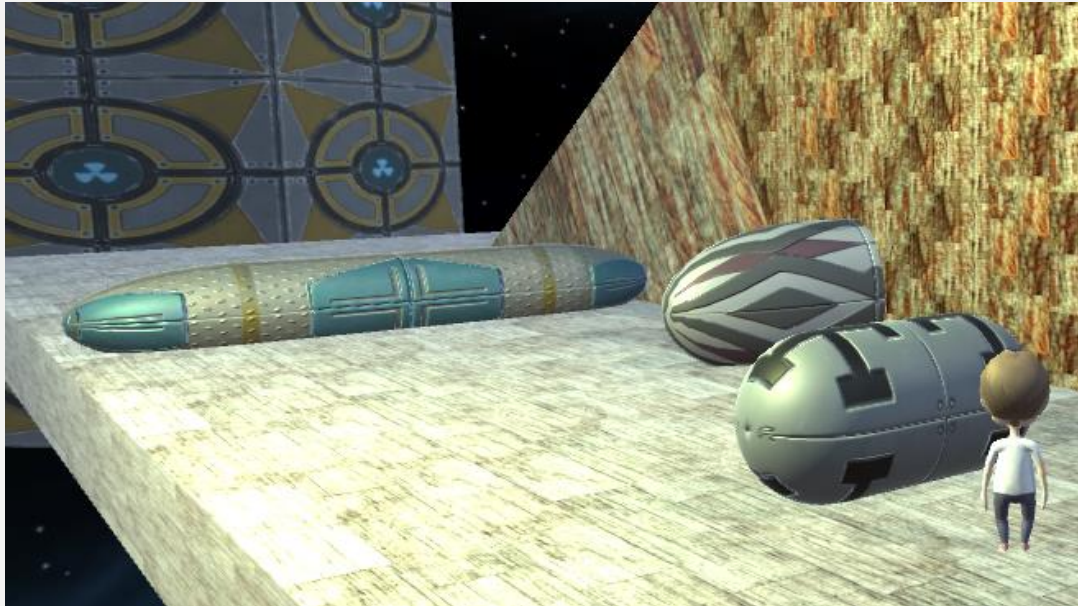
Bomb Zone



Lava Zone



# 1) EASY MODE



Capsule Zone



Wall Zone



## 2) HARD MODE



Character



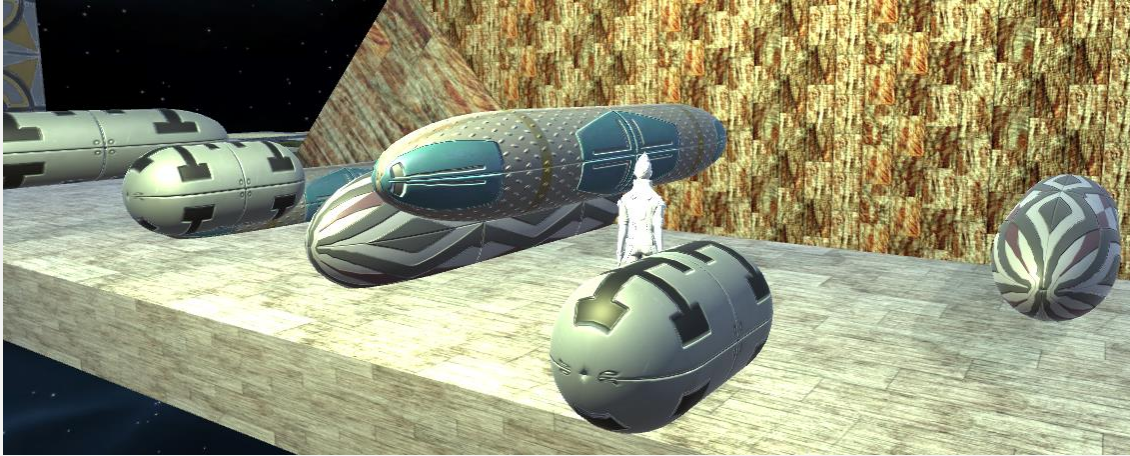
Bomb Zone



Lava Zone



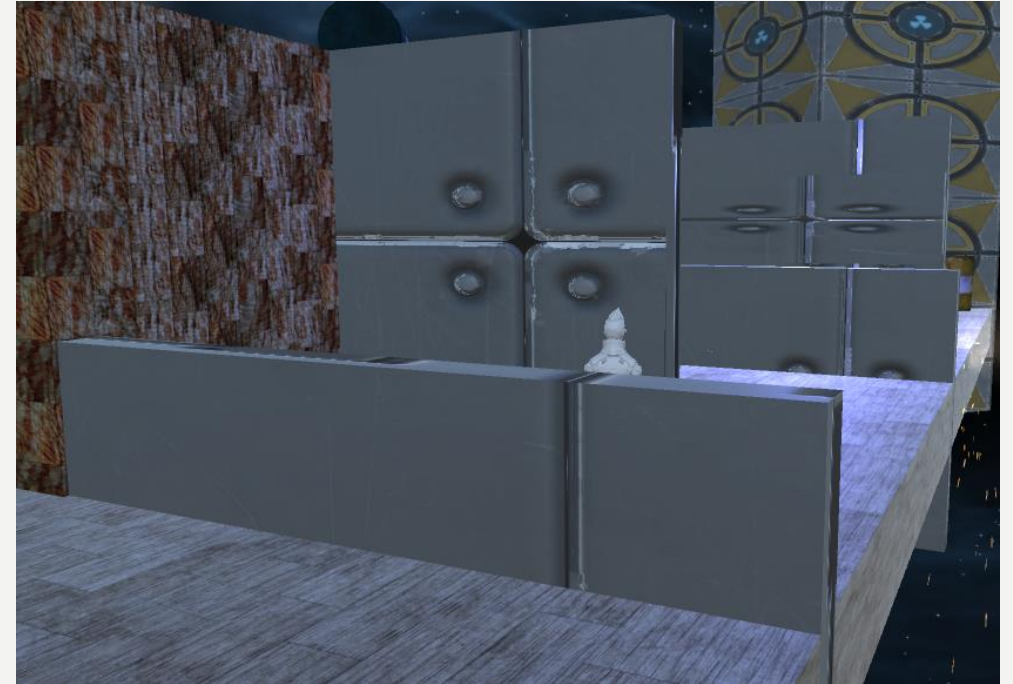
## 2) HARD MODE



Capsule Zone



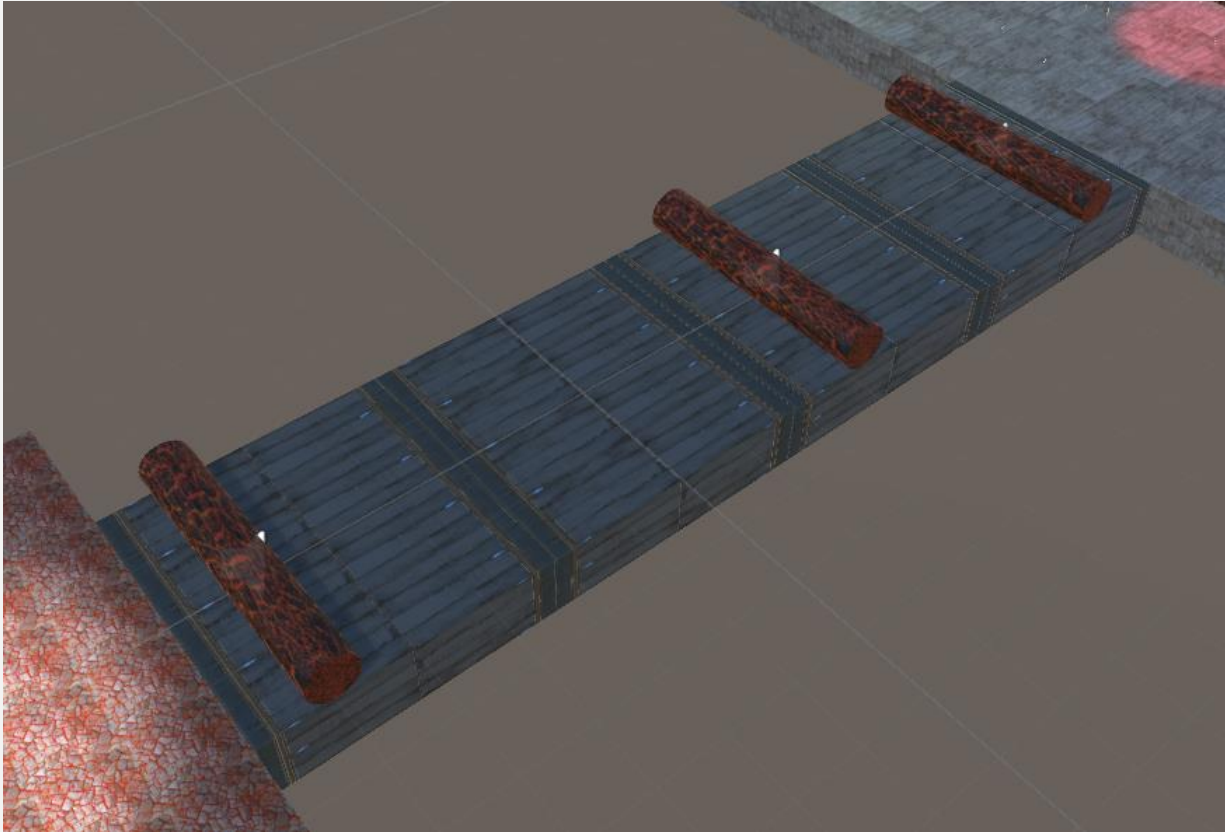
Last Mission Zone



Wall Zone

### 3. 기능 추가

# 1-1) 움직이는 불기둥 장애물



LavaObstacle

## MoveZ.cs 주요 코드

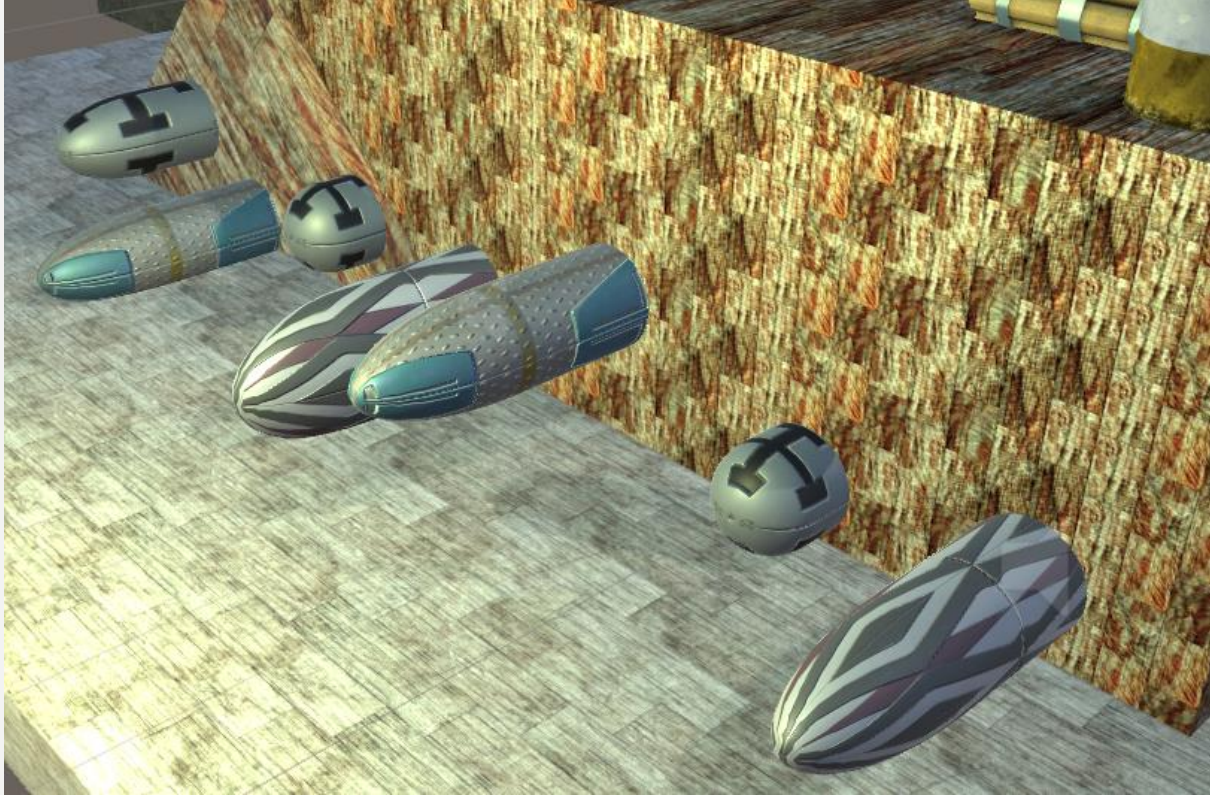
```
void Start()
{
    nowPst = transform.position; // 현재 위치
}
참조 0개
void Update()
{
    Vector3 vt = nowPst;
    vt.z += delta * Mathf.Sin(Time.time * speed);
    transform.position = vt;
    if (isRotating)
    {
        transform.Rotate(rotationAngle * rotationSpeed * Time.deltaTime);
    }
}
```

불기둥을 앞뒤로 이동시키기 위해  
이동 가능한 범위(delta)와 속도(speed) 값을 정하여  
Position의 Z값을 바꿔가며 움직임을 구현

회전시키기 위해 Y값에 회전 각도(rotationAngle)와  
회전 속도(rotationSpeed) 값을 정하여  
Rotate() 함수를 이용하여 회전을 구현



# 1-2) 움직이는 캡슐 장애물



CapsuleObstacle

MoveX.cs 주요 코드

```
void Start()
{
    nowPst = transform.position; // 현재 위치
}
참조 0개
void Update()
{
    Vector3 vt = nowPst;
    vt.x += delta * Mathf.Sin(Time.time * speed);
    transform.position = vt;
}
```

캡슐을 좌우로 이동시키기 위해  
이동 가능한 범위(delta)와 속도(speed) 값을 정하여  
Position의 X값을 바꿔가며 움직임을 구현

# 1-3) 움직이는 벽 장애물



WallObstacle

## MoveX.cs 주요 코드

```
void Start()
{
    nowPst = transform.position; // 현재 위치
}
참조 0개
void Update()
{
    Vector3 vt = nowPst;
    vt.x += delta * Mathf.Sin(Time.time * speed);
    transform.position = vt;
}
```

## MoveY.cs 코드 일부

```
vt.y += delta * Mathf.Sin(Time.time * speed);
```

벽을 상하 or 좌우로 이동시키기 위해  
이동 가능한 범위(delta)와 속도(speed) 값을 정하여  
Position의 X or Y값을 바꿔가며 움직임을 구현



## 2-1) 폭탄 함정

Tag: Bomb – 공통



Old-timer bomb prefab



Plastic bomb prefab



Magnetic bomb prefab

TouchBomb.cs 주요 코드

```
public class TouchBomb : MonoBehaviour {  
  
    참조 0개  
    void OnCollisionEnter(Collision target)  
    {  
        if (target.collider.CompareTag("Bomb"))  
        {  
            SoundManager.instance.ExplosionSound();  
            LifeCtrl.life--;  
        }  
    }  
}
```

(캐릭터에 위의 스크립트 붙여 넣음)

충돌 판정을 하기위해

onCollisionEnter() 메소드 사용하여

캐릭터와 태그 값이 Bomb인 오브젝트가 충돌하면  
폭발 효과음이 나고, 목숨이 1 줄어듦

# 3-1) 목숨 아이템



Heart

Tag: Item

GetItem.cs 주요 코드

```
public class GetItem : MonoBehaviour {  
    참조 0개  
    void OnCollisionEnter(Collision target)  
    {  
        if (target.collider.CompareTag("Item"))  
        {  
            SoundManager.instance.GetSound();  
            LifeCtrl.life++;  
            Destroy(target.gameObject);  
        }  
    }  
}
```

(캐릭터에 위의 스크립트 붙여 넣음)

충돌 판정을 하기위해

onCollisionEnter() 메소드 사용하여

캐릭터와 태그 값이 Item인 오브젝트가 충돌하면  
get 효과가음 나고, 목숨이 1 늘어남

## 4-1) 캐릭터 변경



Supercyan Character

Easy Mode용 캐릭터



Player

Hard Mode용 캐릭터

# 5-1) 목숨 추가

Time : 233  
Life : 3

게임의 완성도를 높이기 위해 목숨 기능 추가



◀ 목숨 +1 아이템



▲ 목숨 -1 폭탄 함정

## LifeCtrl.cs 주요 코드

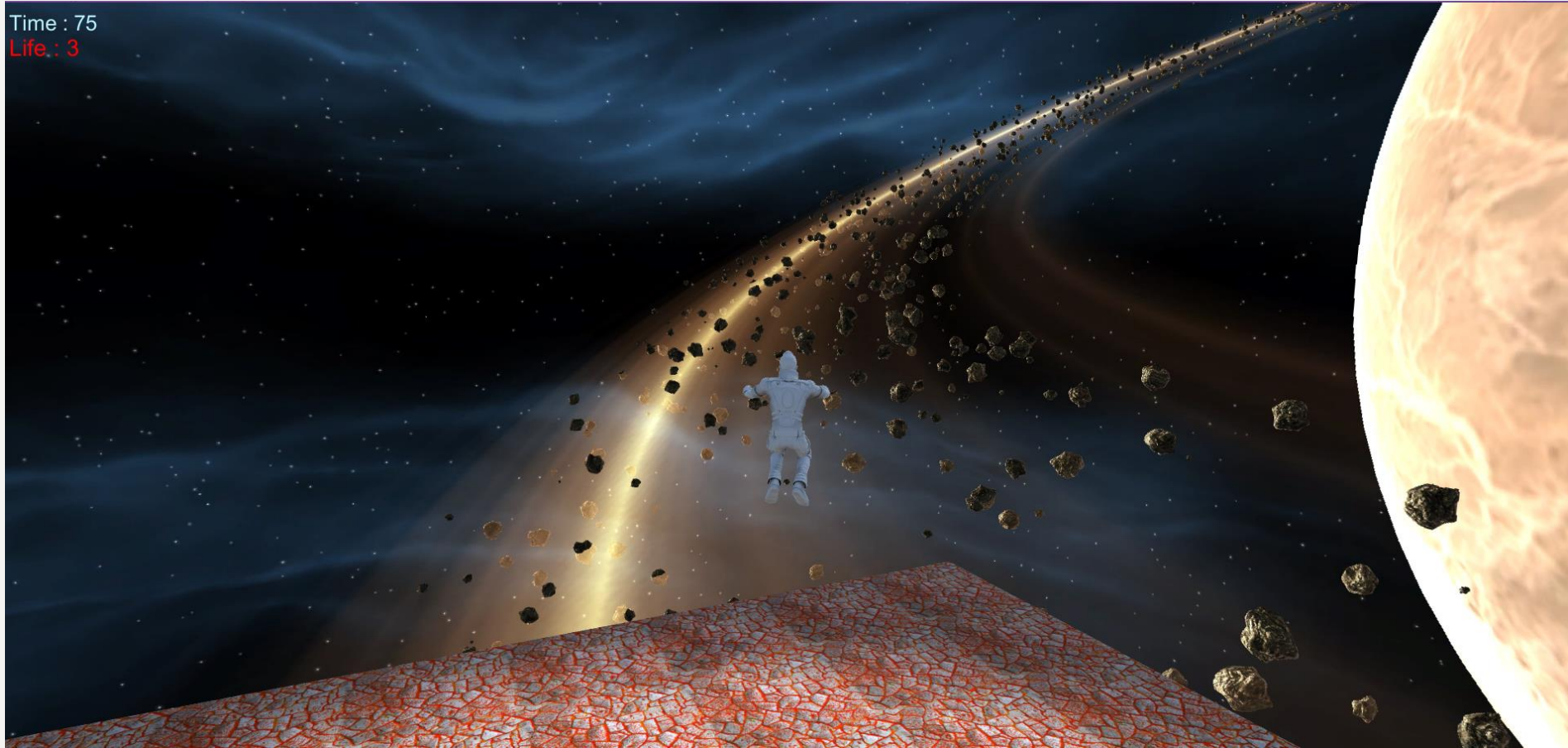
```
void Start () {  
    life = 3;  
}  
참조 0개  
void Update () {  
    Text uiText = GetComponent<Text>();  
    uiText.text = "Life : " + life.ToString();  
    if(life == 1)  
    {  
        time += Time.deltaTime;  
        if (time >= interval)  
        {  
            time = 0f;  
            SoundManager.instance.Hear tSound();  
        }  
    }  
    if (life == 0)  
        Application.LoadLevel("Ending");  
}
```

목숨 초기 값을 3으로 설정하고  
폭탄을 밟으면 목숨 -1, 아이템을 먹으면 목숨 +1  
목숨이 1 남으면 심장 소리 효과음이 들리고,  
0 남으면 Ending 씬으로 이동함

## 4. 배경 추가



# 1-1) 그래픽 배경



MainCamera에 **Skybox** 컴포넌트 추가 후 **Custom Skybox**에 DSBWP(배경) 추가



## 1-2) 텍스처



Wooden floor



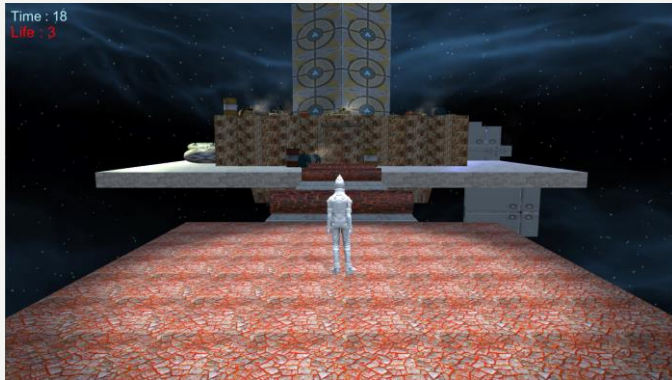
Lavabrick

각각의 3D 오브젝트의 표면에 매핑

## 2) 배경음



Main scene



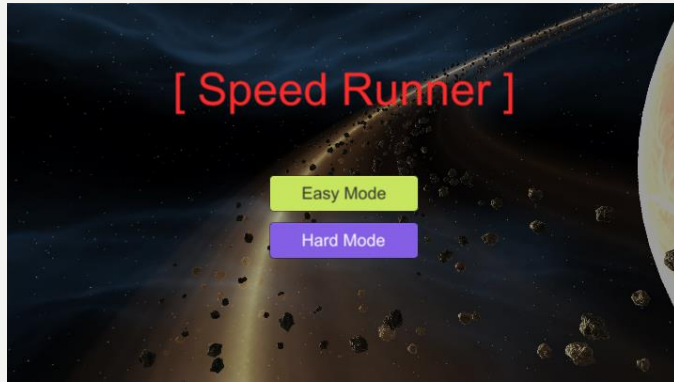
Hard scene



Pillar

Pillar에 **Audio Source** 컴포넌트 추가 후  
**AudioClip**에 StreetLove(음악) 추가하고  
Spatial Blend 수정 및 Loop 체크

## 2) 배경음



Start scene

MainCamera에 **Audio Source** 컴포넌트 추가 후  
**AudioClip**에 **Chunky\_Monkey**(음악) 추가하고 Loop 체크



Ending scene



Ending2 scene

MainCamera에 **Audio Source** 컴포넌트 추가 후  
**AudioClip**에 **SunnyDay**(음악) 추가하고 Loop 체크



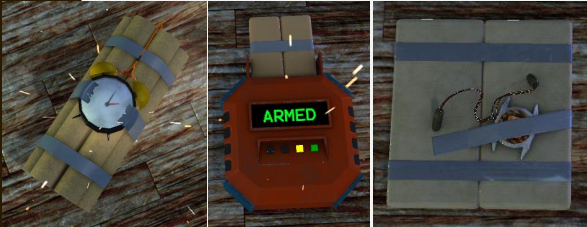
### 3) 효과음



GetItem.cs 일부 코드

```
void OnCollisionEnter(Collision target)
{
    if (target.collider.CompareTag("Item"))
    {
        SoundManager.instance.GetSound();
    }
}
```

목숨 아이템 획득 시 get 효과음 재생



TouchBomb.cs 일부 코드

```
void OnCollisionEnter(Collision target)
{
    if (target.collider.CompareTag("Bomb"))
    {
        SoundManager.instance.ExplosionSound();
    }
}
```

폭탄 밟을 시 bakuhatu13 효과음 재생



LifeCtrl.cs 일부 코드

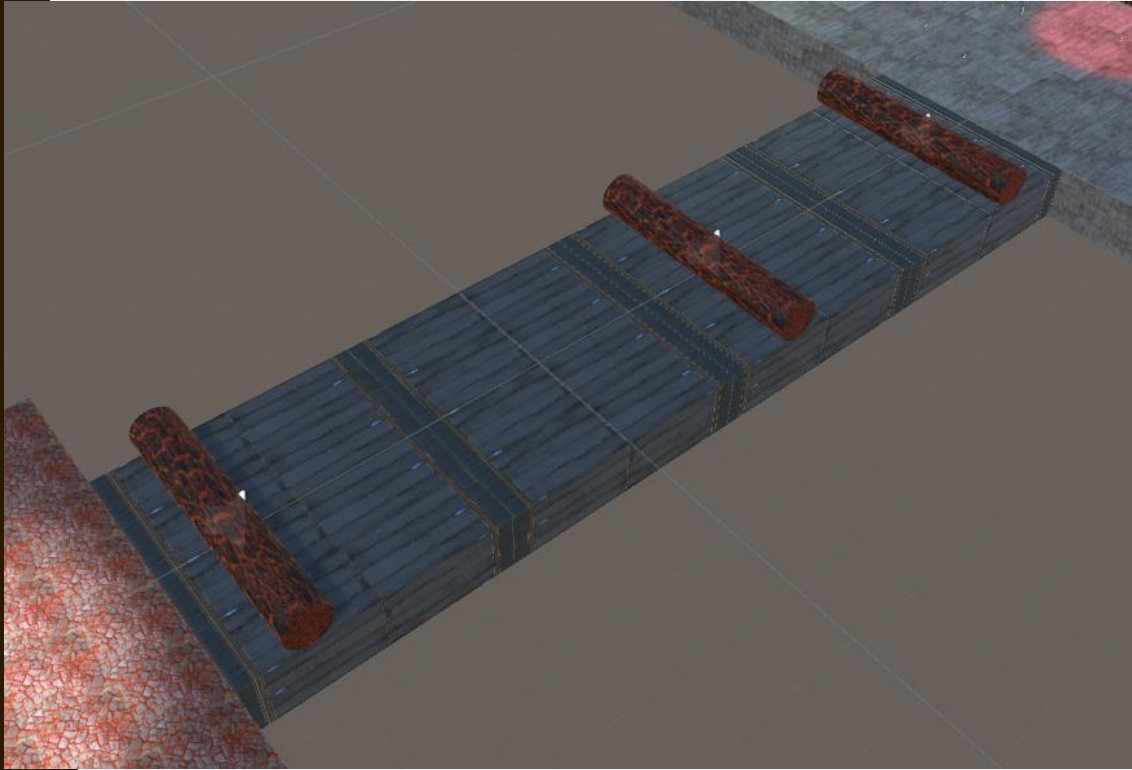
```
if(life == 1)
{
    time += Time.deltaTime;
    if (time >= interval)
    {
        time = 0f;
        SoundManager.instance.HeartSound();
    }
}
```

목숨이 1 남을 시 Heartbeat 효과음 재생

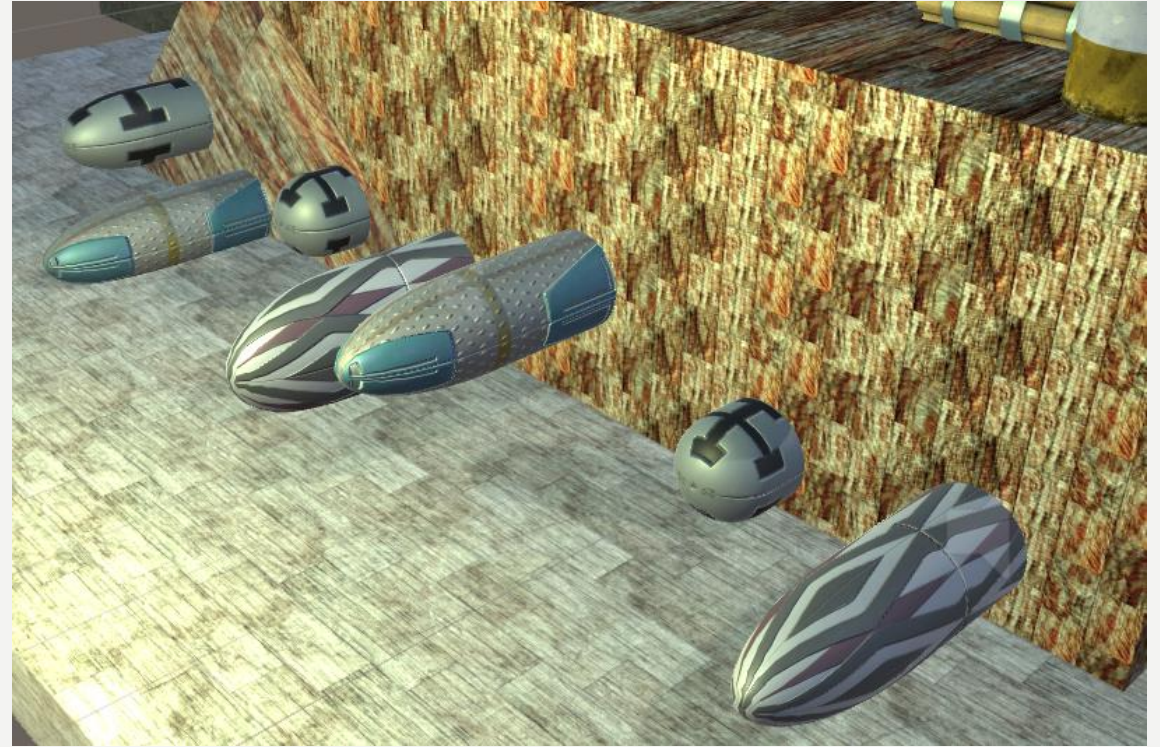
SoundManager.cs 주요 코드

```
public class SoundManager : MonoBehaviour {
    public AudioClip soundExplosion, soundGet, soundHeart;
    AudioSource myAudio;
    public static SoundManager instance;
    참조 0개
    void Awake()
    {
        if (SoundManager.instance == null)
            SoundManager.instance = this;
    }
    참조 0개
    void Start () {
        myAudio = GetComponent<AudioSource>();
    }
    참조 1개
    public void ExplosionSound()
    {
        myAudio.PlayOneShot(soundExplosion);
    }
    참조 1개
    public void GetSound()
    {
        myAudio.PlayOneShot(soundGet);
    }
    참조 1개
    public void HeartSound()
    {
        myAudio.PlayOneShot(soundHeart);
    }
}
```

### 3) 효과음



LavaObstacle



CapsuleObstacle

Audio Source 컴포넌트 추가 후 AudioClip에 Drawer(효과음) 추가하고  
Priority, Volume, Spatial Blend 값 수정 후 Loop 체크

### 3) 효과음



WallObstacle

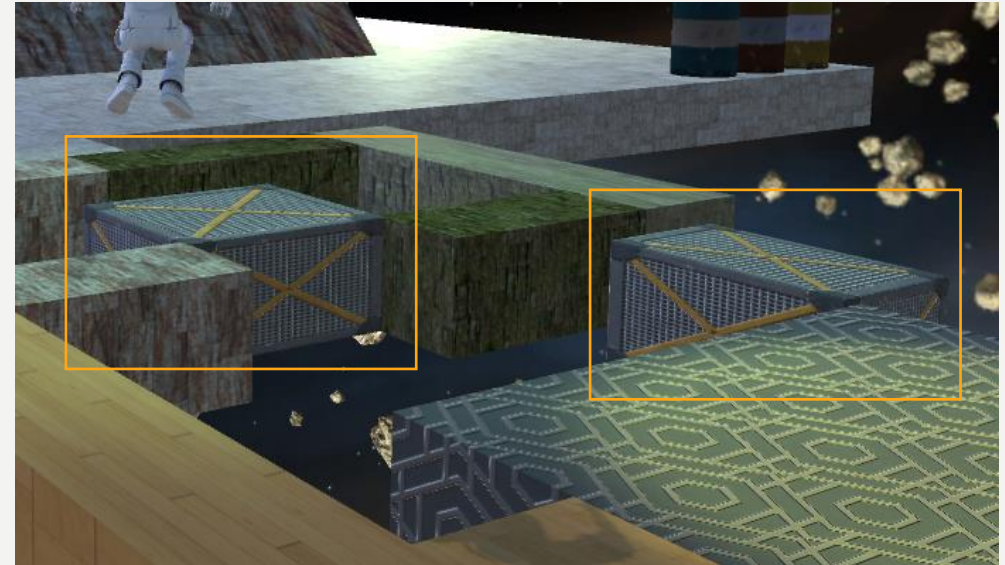
**Audio Source** 컴포넌트 추가 후 **AudioClip**에 movingwallsound-1(효과음) 추가하고  
Priority, Volume, Spatial Blend 값 수정 후 Loop 체크



### 3) 효과음



Lift



**Audio Source** 컴포넌트 추가 후 **AudioClip**에 liftsound(효과음) 추가하고  
Priority, Volume, Spatial Blend 값 수정 후 Loop 체크