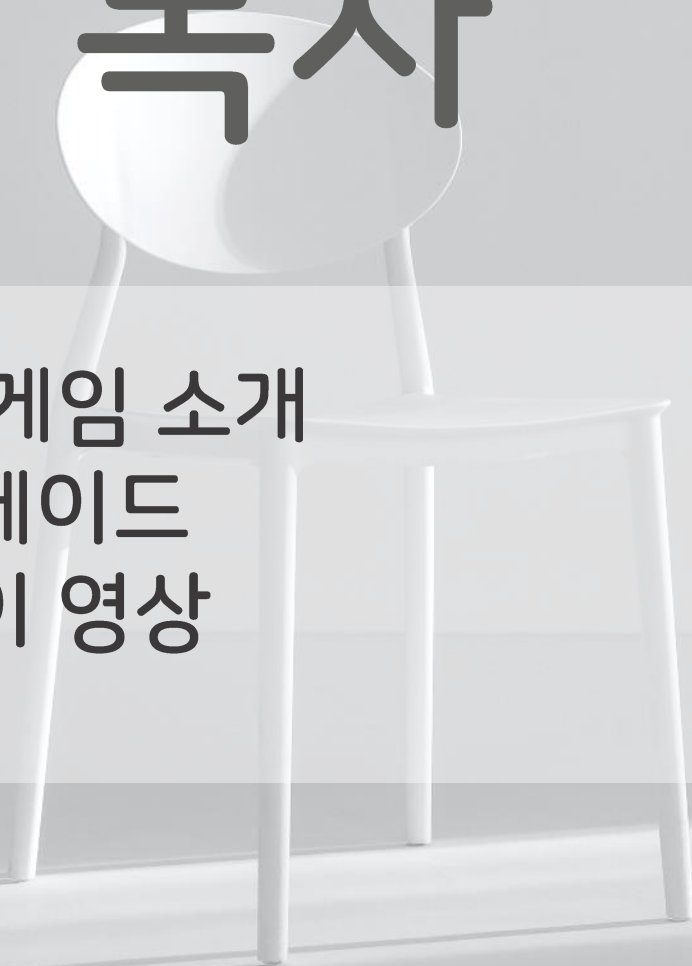


# 유니티 게임발표



---

# 목차

- 
- 1. 기존 게임 소개
  - 2. 업그레이드
  - 3. 플레이 영상

# 초밥 액션 게임

## 게임 규칙

캐릭터가 왼쪽, 오른쪽 키로 좌우로 이동하고,  
스페이스바로 점프합니다.

초밥을 먹으면 점수가 1 증가하고,  
모든 초밥을 먹으면 게임 클리어입니다.  
유령에 충돌하면 게임 오버입니다.



# 플레이 영상

0

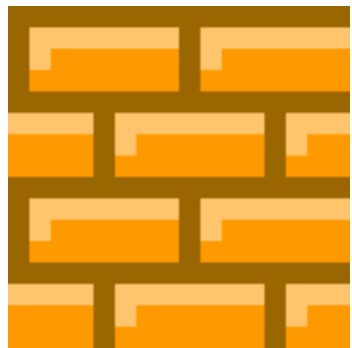


# 게임오버

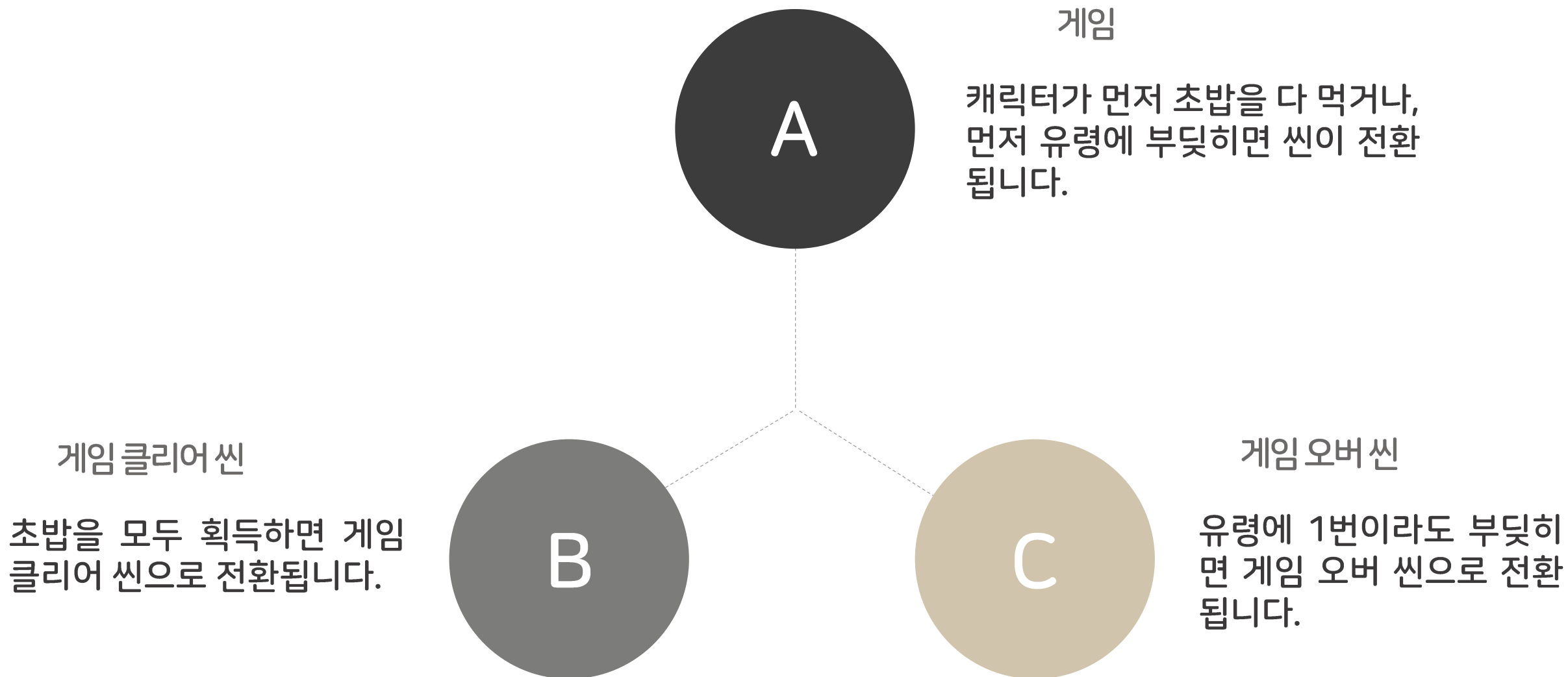
0



# 소스 소개

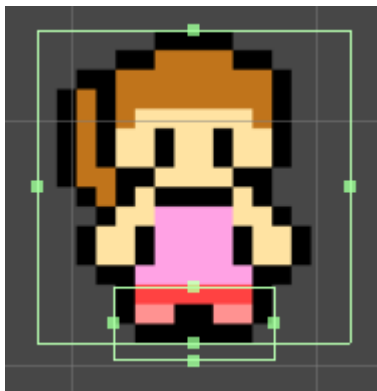


# Scene



## Part 1

# 플레이어



애니메이션 스크립트

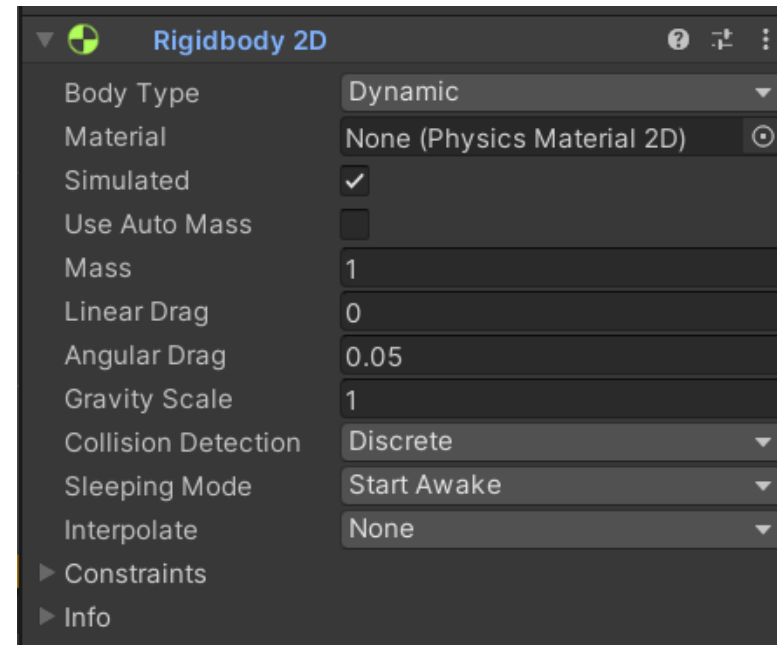
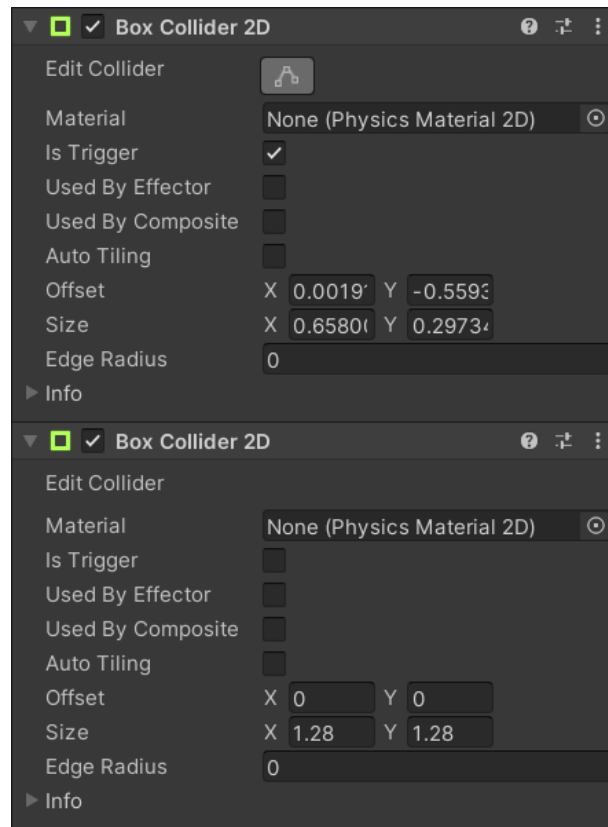
**On key Press\_Change Anime**

이동 스크립트

**On key press\_Move Gravity**

카메라 이동 스크립트

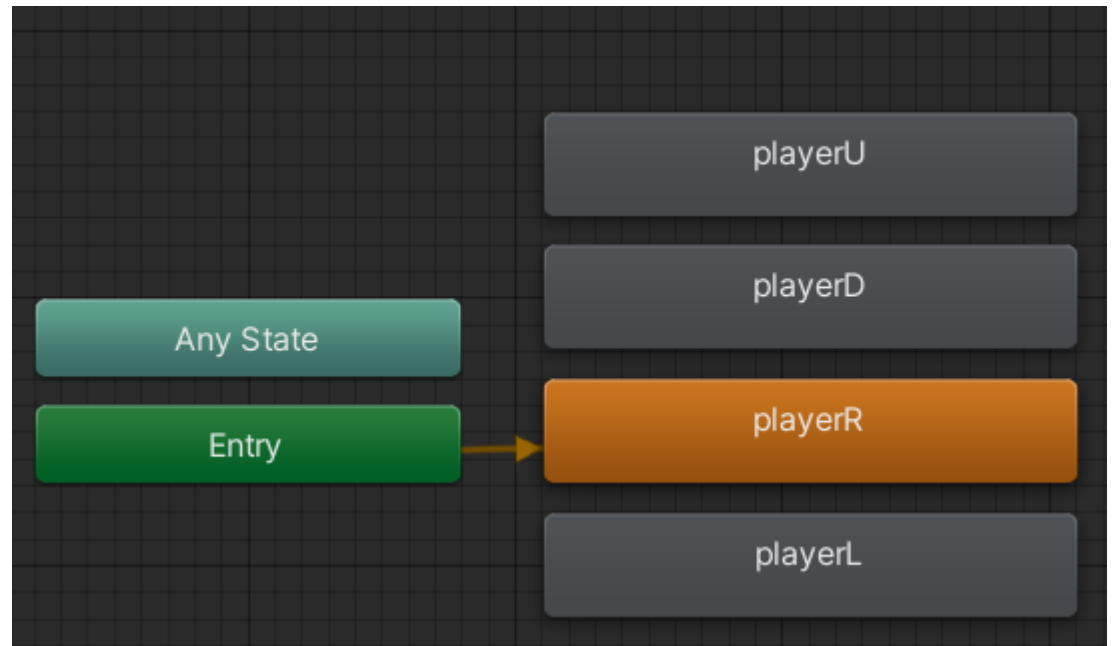
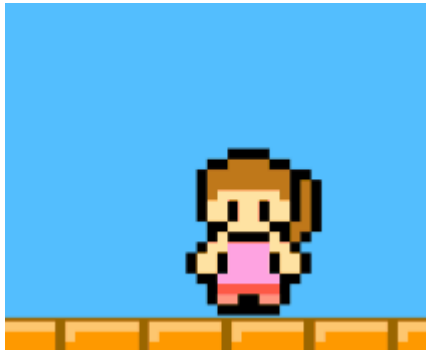
**Forever\_Chase Camera H**



**Box Collider 2D**  
**Rigidbody 2D**



# 플레이어 애니메이션



# 플레이어 애니메이션

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

// 키를 누르면 애니메이션을 전환한다
public class OnKeyPress_ChangeAnime : MonoBehaviour
{
    public string upAnime = ""; // 위 방향 : Insp
    public string downAnime = ""; // 아래 방향 : Insp
    public string rightAnime = ""; // 오른쪽 방향 : Insp
    public string leftAnime = ""; // 왼쪽 방향 : Insp

    string nowMode = "";
    string oldMode = "";

    void Start()// 처음에 시행한다
    {
        nowMode = downAnime;
        oldMode = "";
    }
}
```

```
void Update()// 계속 시행한다
{
    if (Input.GetKey("up"))// 위 키면
    {
        nowMode = upAnime;
    }
    if (Input.GetKey("down"))// 아래 키면
    {
        nowMode = downAnime;
    }
    if (Input.GetKey("right"))// 오른쪽 키면
    {
        nowMode = rightAnime;
    }
    if (Input.GetKey("left"))// 왼쪽 키면
    {
        nowMode = leftAnime;
    }
}

void FixedUpdate() // 계속 시행한다(일정 시간마다)
{
    // 만약 다른 키가 눌리면
    if (nowMode != oldMode)
    {
        oldMode = nowMode;
        // 애니메이션을 전환한다
        Animator animator = this.GetComponent<Animator>();
        animator.Play(nowMode);
    }
}
```

# 플레이어 이동

```

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

// 키를 누르면 이동하는 플레이어
public class OnKeyF : MonoBehaviour
{
    public float speed = 5f;
    public float jumpPower = 5f;

    float vx = 0;
    bool leftFlag = false;
    bool pushFlag = false;
    bool jumpFlag = false;
    bool groundFlag = false;
    Rigidbody2D rbody;

    void Start ()
    {
        rbody = GetComponent<Rigidbody2D>();
        rbody.constraints = RigidbodyConstraints2D.FreezeRotation;
    }

    void FixedUpdate() // 계속 시행한다(일정 시간마다)
    {
        // 이동한다(중력을 건 채)
        rbody.velocity = new Vector2(vx, rbody.velocity.y);
        // 왼쪽 오른쪽으로 방향을 바꾼다
        this.GetComponent<SpriteRenderer>().flipX = leftFlag;
        // 만약 점프할 때
        if (jumpFlag) {
            jumpFlag = false;
            rbody.AddForce(new Vector2(0, jumpPower), ForceMode2D.Impulse);
        }

        void OnTriggerStay2D(Collider2D collision)
        { // 발이 무언가에 닿으면
            groundFlag = true;
        }

        void OnTriggerExit2D(Collider2D collision)
        { // 발에 아무 것도 닿지 않으면
            groundFlag = false;
        }
    }
}

```

오른쪽 키가 눌리면  
이동량을 늘린다

왼쪽 키가 눌리면  
이동량을 늘린다

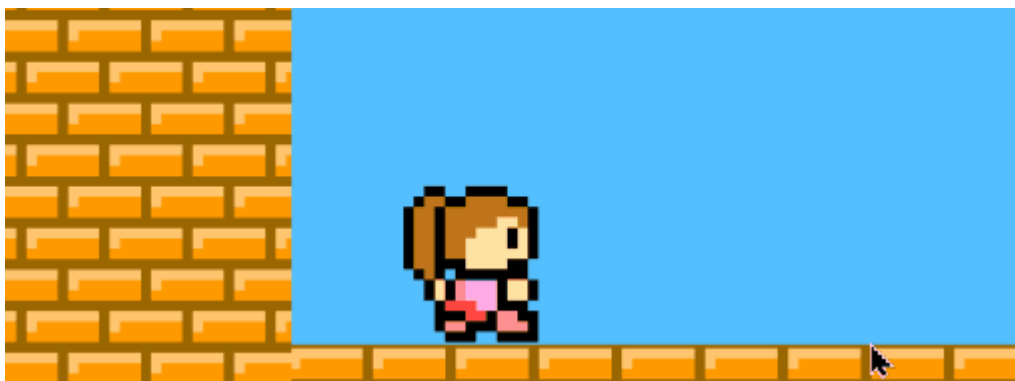
바닥에 닿았다면  
groundFlag

바닥에서 벗어나고 나가지 않으면  
groundFlag

바닥 상태  
groundFlag

바닥 해제  
groundFlag

# 카메라 이동



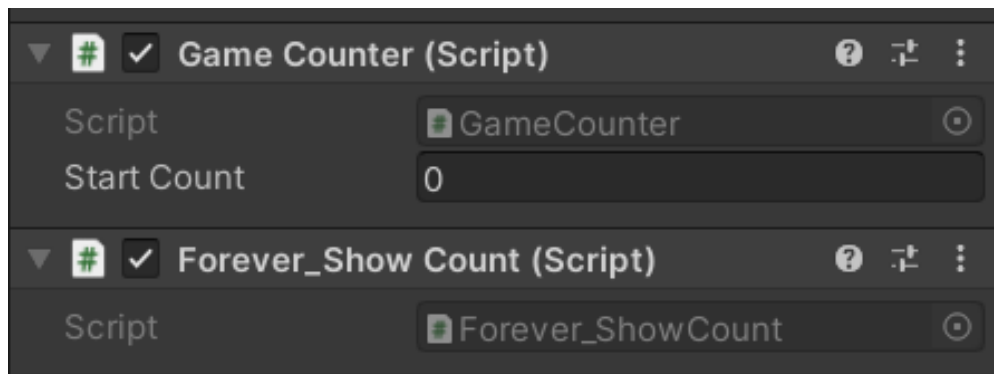
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

// 계속 카메라가 쫓아간다(수평으로)
public class Forever_ChaseCameraH : MonoBehaviour
{
    Vector3 base_pos;

    void Start() // 처음에 시행한
    {
        // 카메라의 원래 위치를 기억해 둔다
        base_pos = Camera.main.gameObject.transform.position;
    }

    void LateUpdate() // 계속 시행한다(여러 가지 처리의 마지막에)
    {
        Vector3 pos = this.transform.position; // 자신의 위치
        pos.z = -10; // 카메라이므로 앞으로 이동시킨다
        pos.y = base_pos.y; // 카메라 원래의 높이를 사용한다
        Camera.main.gameObject.transform.position = pos;
    }
}
```

# 점수 UI



Text UI를 게임 카운터로 지정하여  
시작 시 초기화

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

// 카운터 본체
public class GameCounter : MonoBehaviour
{
    public static int value; // 공유하는 카운터의 값

    public int startCount = 0; // 카운터 초기값 : Inspector

    void Start() // 처음에 시행한다
    {
        value = startCount; // 카운터를 리셋
    }
}
```

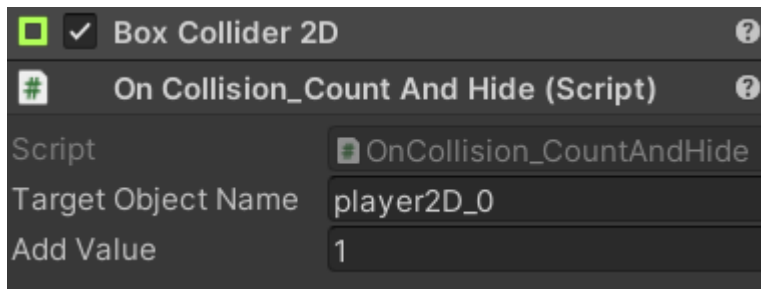
# 점수 UI



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

// 계속 카운터 값을 표시한다
public class Forever_ShowCount : MonoBehaviour
{
    void Update()// 계속
    {
        // 카운터 값을 표시한다
        GetComponent<Text>().text = GameCounter.value.ToString();
    }
}
```

# 아이템

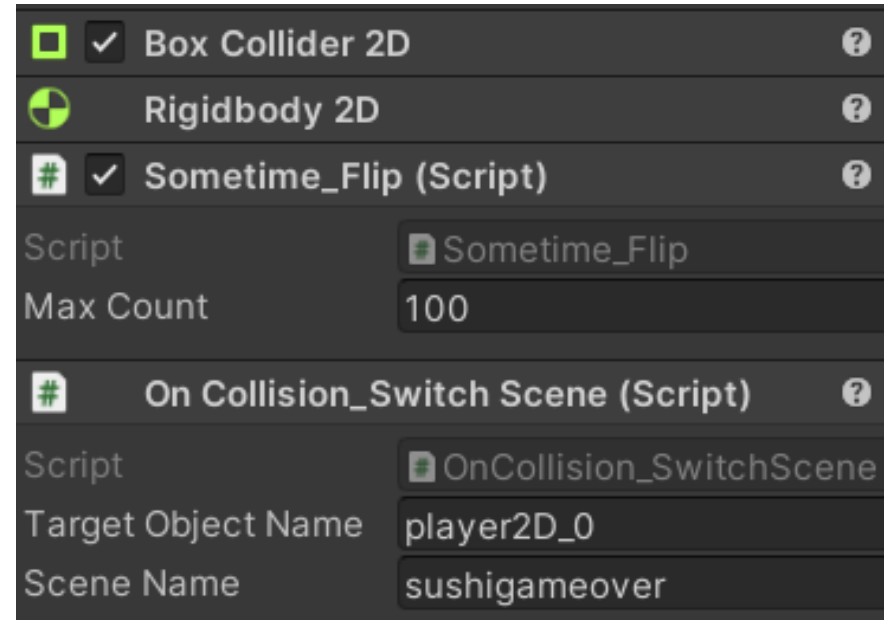


캐릭터와 충돌 시 삭제하고,  
1점을 추가

```
// 충돌하면 카운터를 증가하고 자신을 지운다
public class OnCollision_CountAndHide : MonoBehaviour
{
    public string targetObjectName; // 목표 오브젝트 이름 : Inspector에
    public int addValue = 1;      // 증가량 : Inspector에 지정

    void OnCollisionEnter2D(Collision2D collision) // 충돌했을 때
    {
        // 만약 충돌한 것의 이름이 목표 오브젝트면
        if (collision.gameObject.name == targetObjectName)
        {
            // 카운터 값을 증가한다
            GameCounter.value = GameCounter.value + addValue;
            // 자기자신을 지운다
            this.gameObject.SetActive(false);
        }
    }
}
```

# 몬스터



Box 범위 지정, 중력 지정  
이동 스크립트  
캐릭터와 충돌 시 scene 전환



# 몬스터 제자리 이동 스크립트



2초에 한 번씩 움직임

```
public class Sometime_Flip : MonoBehaviour
{
    public int maxCount = 50; // 빈도 : Inspector에 지정

    int count = 0; // 카운터 용
    bool flipFlag = false;

    void Start ()
    { // 처음에 시행한다
        count = 0; // 카운터를 리셋
    }

    void FixedUpdate()// 계속 시행한다(일정 시간마다)
    {
        count = count + 1; // 카운터에 1을 더해서
        if (count >= maxCount)
        { // 만약, maxCount가 되면
            this.transform.Rotate(0, 0, 180); // 180도 회전해서 돈다
            count = 0; // 카운터를 리셋
            // 그 때 그림이 180도 회전하므로 위 아래를 반전시킨다
            flipFlag = !flipFlag;
            this.GetComponent().flipY = flipFlag;
        }
    }
}
```

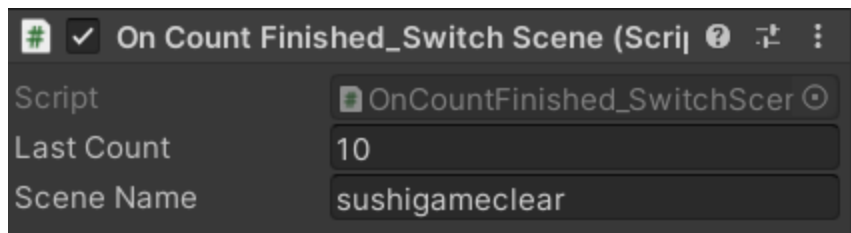
# 게임 오버 씬 전환



```
// 충돌하면 씬을 전환한다
public class OnCollision_SwitchScene : MonoBehaviour
{
    public string targetObjectName; // 목표 오브젝트 이름 : Inspector
    public string sceneName; // 씬 이름 : Inspector에 지정

    void OnCollisionEnter2D(Collision2D collision) // 충돌했을 때
    {
        // 만약 충돌한 것의 이름이 목표 오브젝트였다면
        if (collision.gameObject.name == targetObjectName)
        {
            // 씬을 전환한다
            SceneManager.LoadScene(sceneName);
        }
    }
}
```

# 게임 클리어

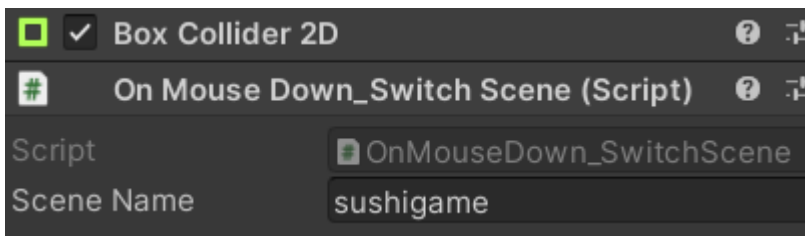
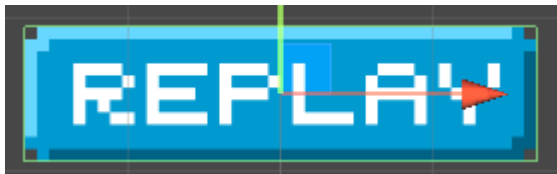


```
// 카운터가 최종값이면 씬을 전환한다
public class OnCountFinished_SwitchScene : MonoBehaviour
{
    public int lastCount = 3; // 카운터의 최종값 : Inspector
    public string sceneName = ""; // 씬 이름 : Inspector에 지

    void FixedUpdate() // 계속 시행한다
    {
        // 카운터가 최종값이 되면
        if (GameCounter.value == lastCount)
        {
            // 씬을 전환한다
            SceneManager.LoadScene (sceneName);
        }
    }
}
```

카운터가 10이면 게임 클리어 씬으로 전환

# 리플레이



Box Collider 지정 후 클릭 시  
게임 씬으로 전환

```
// 터치하면 씬을 전환한다
public class OnMouseDown_SwitchScene : MonoBehaviour
{
    public string sceneName; // 씬 이름 : Inspector에 지정

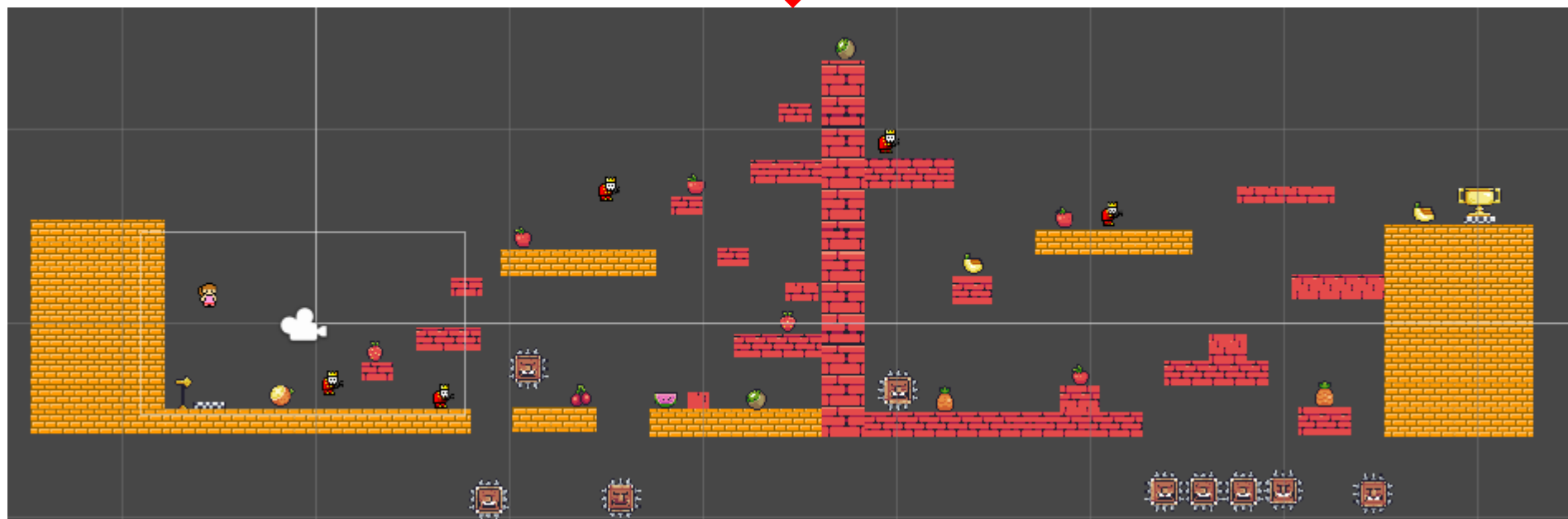
    void OnMouseDown()
    { // 터치하면
        // 씬을 전환한다
        SceneManager.LoadScene (sceneName);
    }
}
```

# Part 2

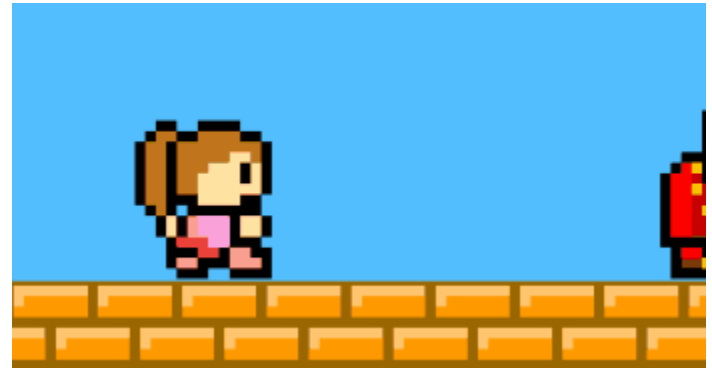
## 업그레이드



# 맵 변경



# 캐릭터 스피드, 점프력 변경

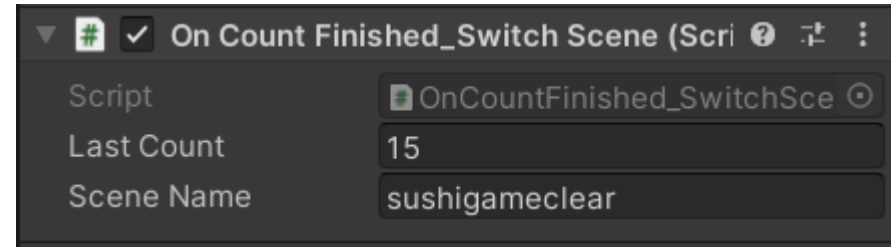


기존 스피드 2, 점프력 5에서  
스피드 5, 점프력 8로 변경

# 에셋 사용



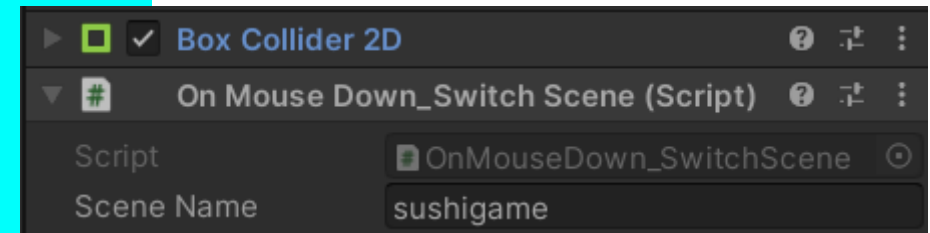
초밥 대신 다양한 과일로 변경  
벽의 일부분을 변경



클리어 점수를 15로 변경

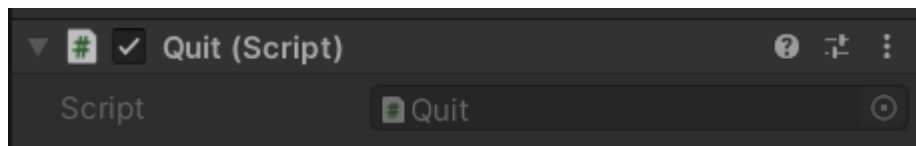


# 메뉴 추가



에셋 사용

# 게임 종료 추가



메뉴 화면에서 ESC를 누르면 게임 종료

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

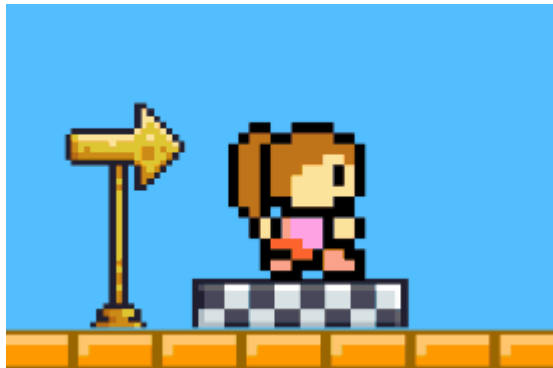
public class Quit : MonoBehaviour
{
    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Escape))
        {
            GameQuit();
        }

        public void GameQuit()
        {
            Application.Quit();
        }
    }
}
```

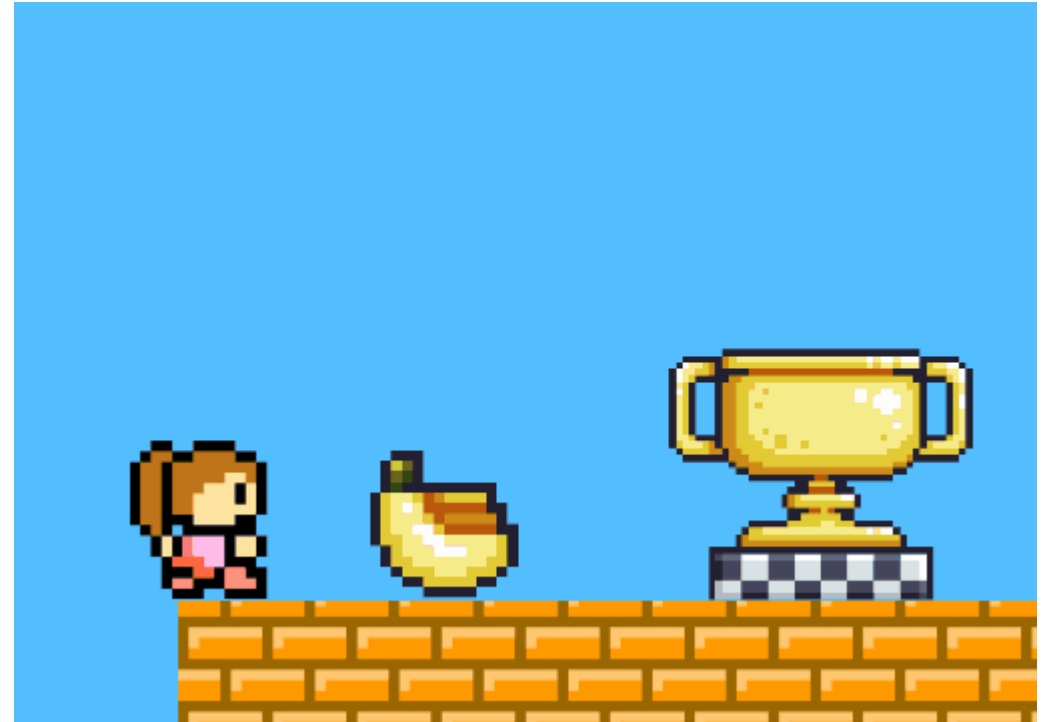
# 클리어 화면 메뉴 추가



# 시작점/ 도착점 표시

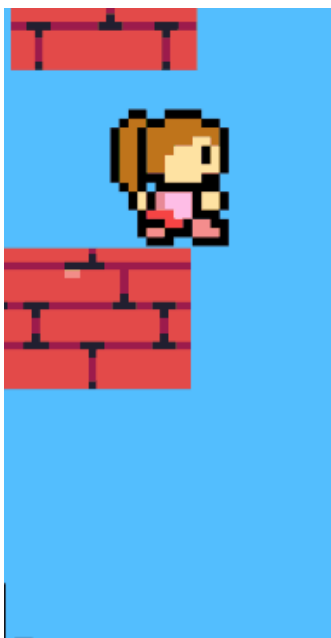


캐릭터가 떨어지면  
시작 포인트에 착지



마지막 과일 위치에 트로피

# 함정 추가



닿으면 게임

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement; // 씬 전환에 필요

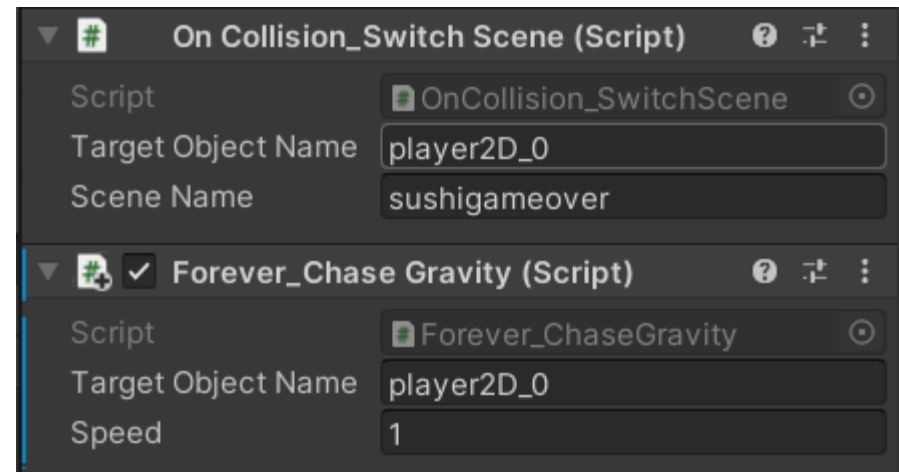
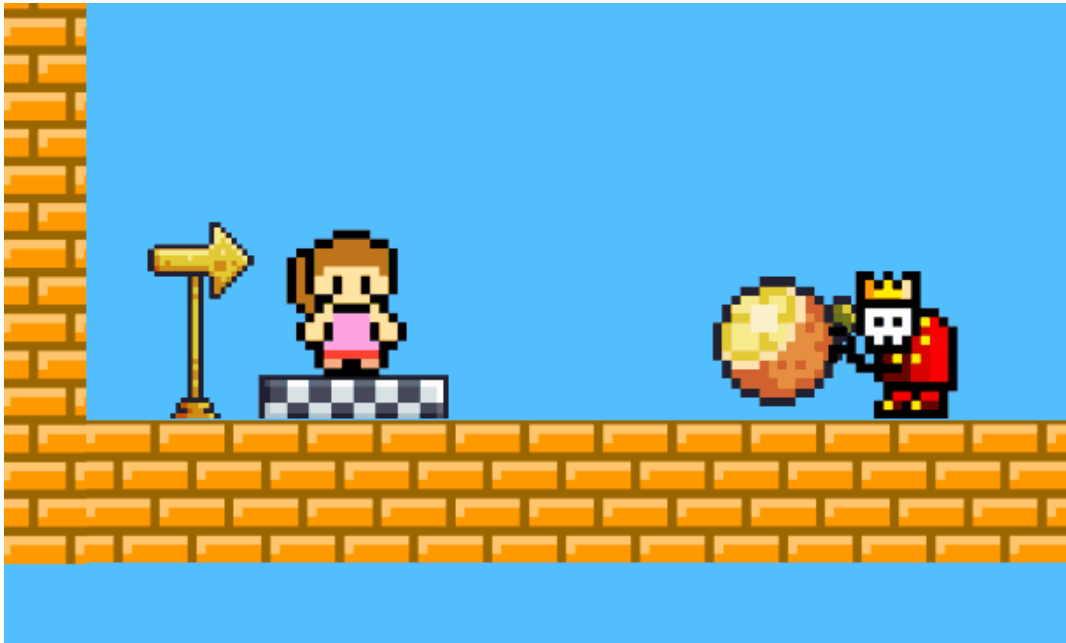
// 충돌하면 씬을 전환한다
public class OnCollision_SwitchScene : MonoBehaviour
{
    public string targetObjectName; // 목표 오브젝트 이름 : Inspector
    public string sceneName; // 씬 이름 : Inspector에 지정

    void OnCollisionEnter2D(Collision2D collision) // 충돌했을 때
    {
        // 만약 충돌한 것의 이름이 목표 오브젝트였다면
        if (collision.gameObject.name == targetObjectName)
        {
            // 씬을 전환한다
            SceneManager.LoadScene(sceneName);
        }
    }
}
```



는 함정

# 따라오는 적 추가



플레이어를 추적하는 유령 추가

# 따라오는 적 추가

```
// 계속 쫓아간다(중력 대응판)
```

```
public class Forever_ChaseGravity : MonoBehaviour
```

```
{
```

```
    public string targetObjectName; // 목표 오브젝트 이름 : Inspector
```

```
    public float speed = 1; // 속도 : Inspector에 지정
```

```
    GameObject targetObject;
```

```
    Rigidbody2D rbody;
```

```
    void Start () // 처음에 시행한다
```

```
{
```

```
    // 목표 오브젝트를 찾아둔다
```

```
    targetObject = GameObject.Find(targetObjectName);
```

```
    // 충돌 시에 회전시키지 않는다
```

```
    rbody = GetComponent<Rigidbody2D>();
```

```
    rbody.constraints = RigidbodyConstraints2D.FreezeRotation
```

```
}
```

```
    void FixedUpdate() // 계속 시행한다(일정 시간마다)
```

```
{
```

```
    // 목표 오브젝트의 방향을 조사해서
```

```
    Vector3 dir = (targetObject.transform.position  
- this.transform.position).normalized;
```

```
    // 그 방향으로 지정한 양으로 진행한다 (중력을 건 채로)
```

```
    float vx = dir.x * speed;
```

```
    rbody.velocity = new Vector2(vx, rbody.velocity.y);
```

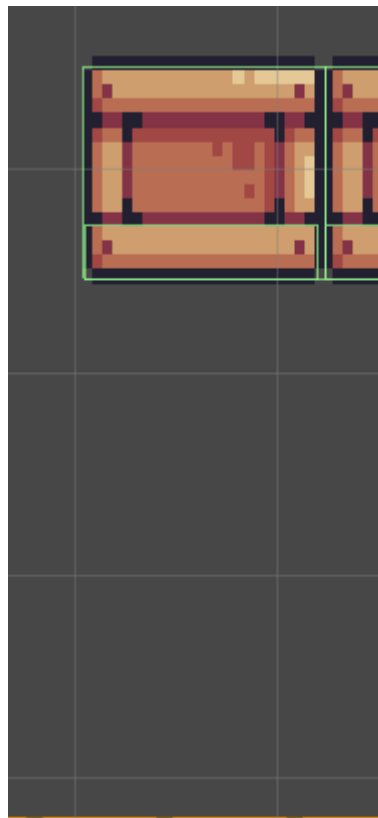
```
    // 진행하는 방향에 왼쪽 오른쪽의 방향을 바꾼다
```

```
    SpriteRenderer sprite = this.GetComponent<SpriteRenderer>();
```

```
    sprite.flipX = (vx < 0);
```

```
}
```

# 사라지는 박스 추가



```
// 충돌하면 지운다
public class OnCollision_Hide : MonoBehaviour
{
    public string targetObjectName; // 목표 오브젝트 이름 : Inspector에
    public string hideObjectName;   // 지울 오브젝트 이름 : Inspector에

    void Start()
    { // 처음은 아무 것도 하지 않는다
    }

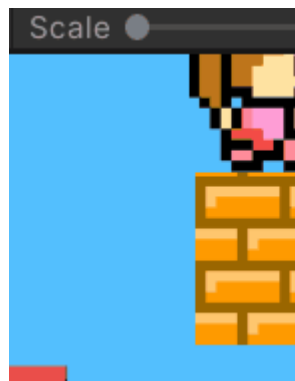
    void OnCollisionEnter2D(Collision2D collision)
    { // 충돌했을 때
        // 만약 충돌한 것의 이름이 목표 오브젝트였다면
        if (collision.gameObject.name == targetObjectName)
        {
            // 지울 오브젝트를 찾아서
            GameObject hideObject = GameObject.Find(hideObjectName);
            hideObject.SetActive(false); // 지운다
        }
    }
}
```



일부 박스에만 On Collision\_Hide 적용



# 카메라 이동 변경



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

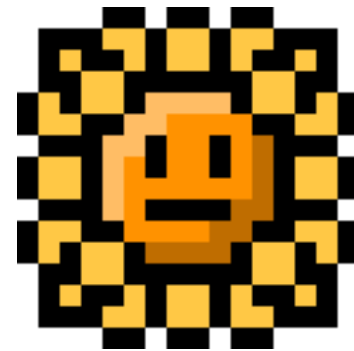
// 계속 카메라가 쫓아간다
public class Forever_ChaseCamera : MonoBehaviour
{
    void LateUpdate() // 계속 시행한다(여러 가지 처리의 마지막에)
    {
        Vector3 pos = this.transform.position; // 자신의 위치
        pos.z = -10; // 카메라이므로 앞으로 이동시킨다
        Camera.main.gameObject.transform.position = pos;
    }
}
```



좌우 수평으로 따라다녔던 카메라를  
상하좌우 모두 따라다니도록 변경

# 플레이 영상





감사합니다.

출처:

Ppt: [https://yusaebyeol.blogspot.com/2021/10/powerpoint-template-free-download-ppt\\_31.html](https://yusaebyeol.blogspot.com/2021/10/powerpoint-template-free-download-ppt_31.html)

도서 : 유니티 2D 게임 제작

<https://ncube-studio.tistory.com/31>

<https://assetstore.unity.com/packages/2d/characters/pixel-adventure-1-155360>

[https://www.youngjin.com/reader/pds/pds.asp?cate\\_cd=4](https://www.youngjin.com/reader/pds/pds.asp?cate_cd=4)

