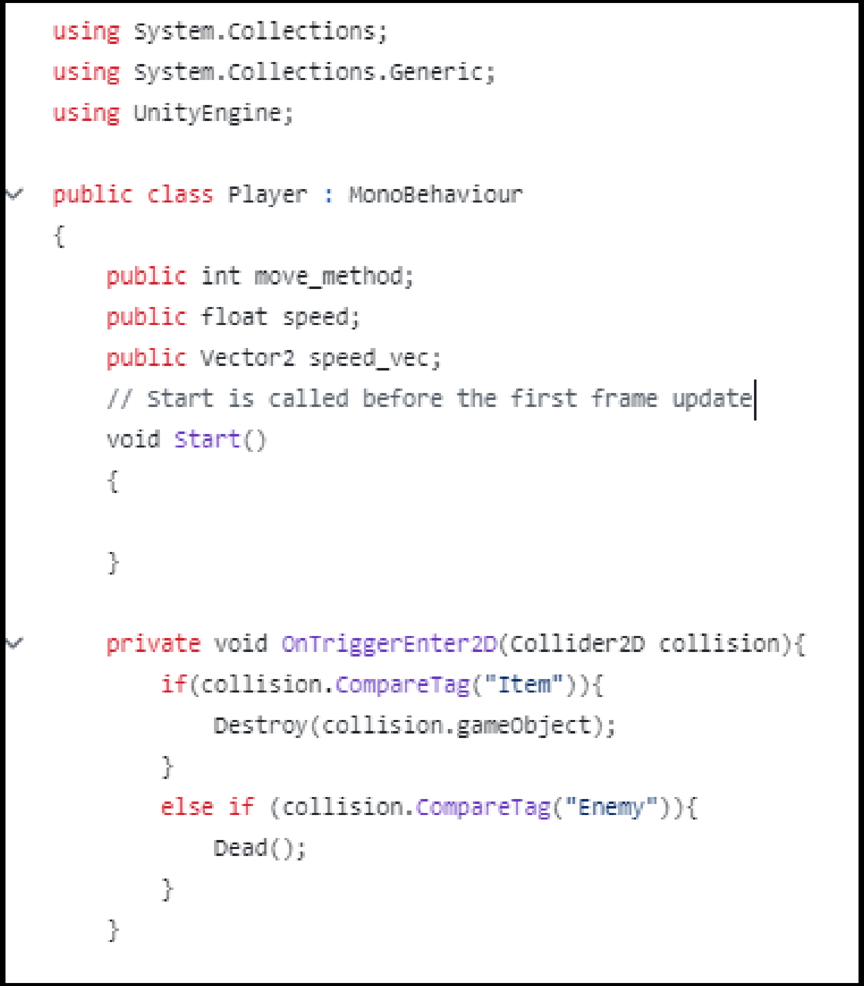
Unity Project

소프트웨어 학과
게임 프로그래밍
2021863030 박세희
2023.12.13

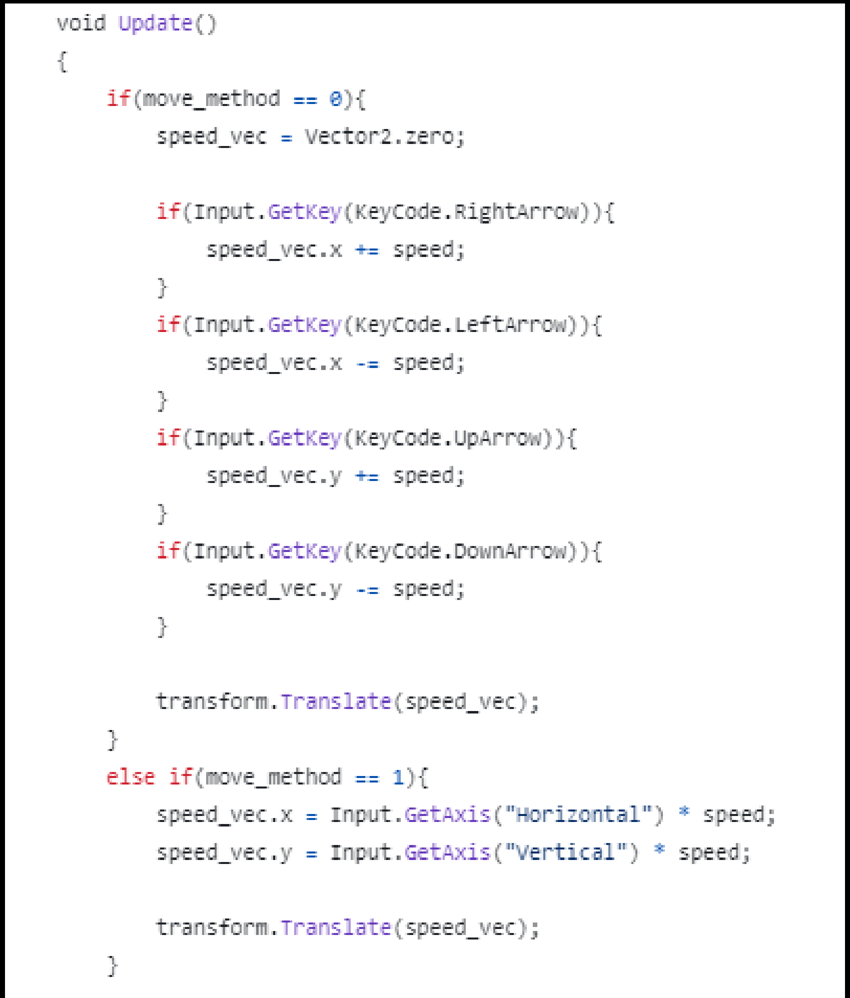
https://github.com/27sh

원본 소스

https://doggie-development.tistory.com/category/%EC%9C%A0%EB%8B%88%ED%8B%B0%20%EA%B0%95%EC%A2%8C/%EC%9C%A0%EB%8B%88%ED%8B%B0%20%EA%B8%B0%EC%B4%88%20%EA%B0%95%EC%A2%8C%20%282D%29



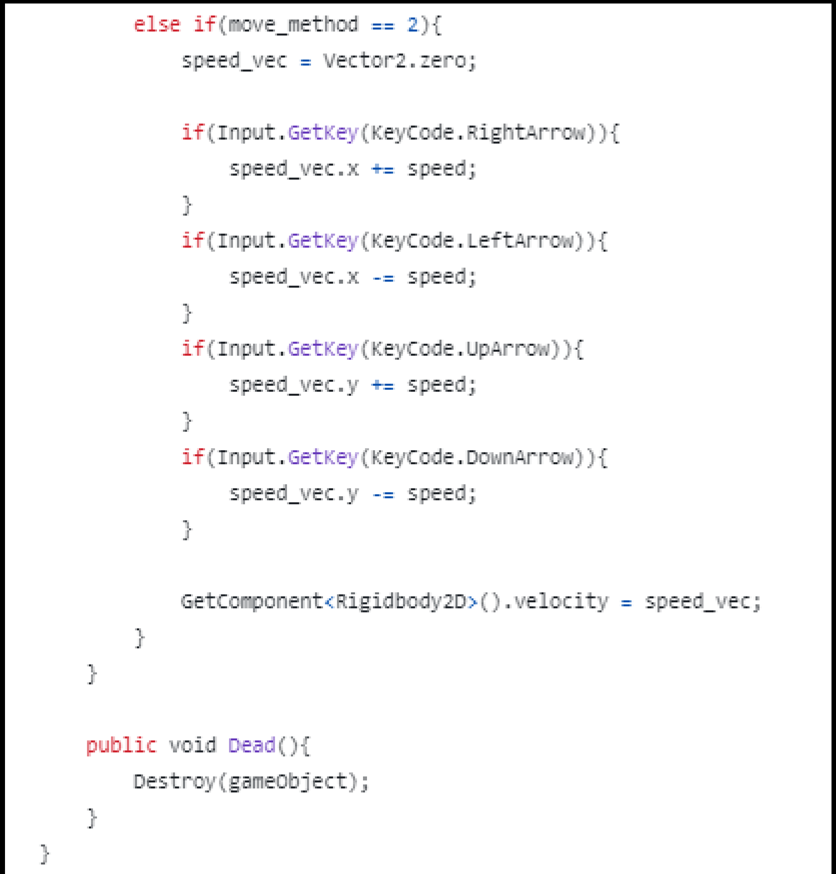
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
    public int move_method;
    public float speed;
    public Vector2 speed_vec;
    // Start is called before the first frame update
    void Start()
    {

    }

    private void OnTriggerEnter2D(Collider2D collision){
        if(collision.CompareTag("Item")){
            Destroy(collision.gameObject);
        }
        else if (collision.CompareTag("Enemy")){
            Dead();
        }
    }
```



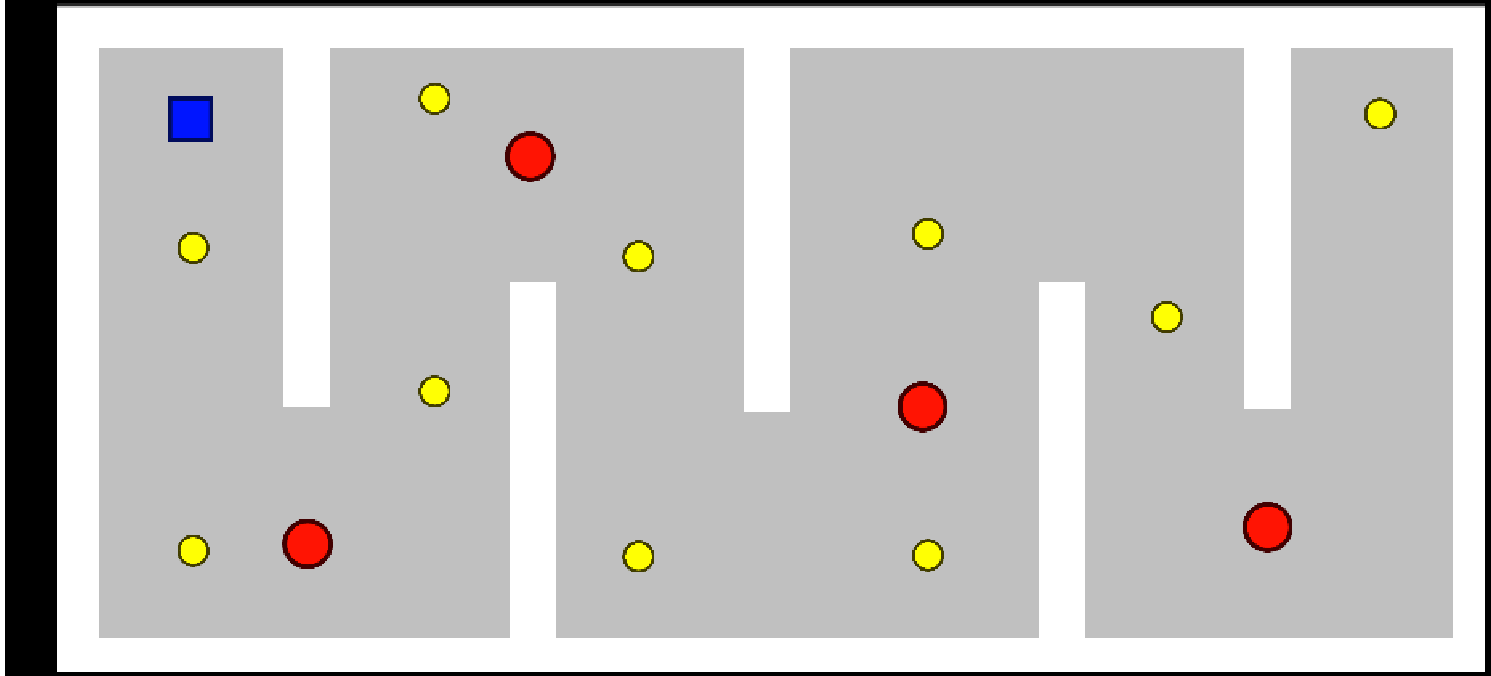
```
void Update()
{
    if(move_method == 0){
        speed_vec = Vector2.zero;

        if(Input.GetKey(KeyCode.RightArrow)){
            speed_vec.x += speed;
        }
        if(Input.GetKey(KeyCode.LeftArrow)){
            speed_vec.x -= speed;
        }
        if(Input.GetKey(KeyCode.UpArrow)){
            speed_vec.y += speed;
        }
        if(Input.GetKey(KeyCode.DownArrow)){
            speed_vec.y -= speed;
        }

        transform.Translate(speed_vec);
    }
    else if(move_method == 1){
        speed_vec.x = Input.GetAxis("Horizontal") * speed;
        speed_vec.y = Input.GetAxis("Vertical") * speed;

        transform.Translate(speed_vec);
    }
```

원본 소스

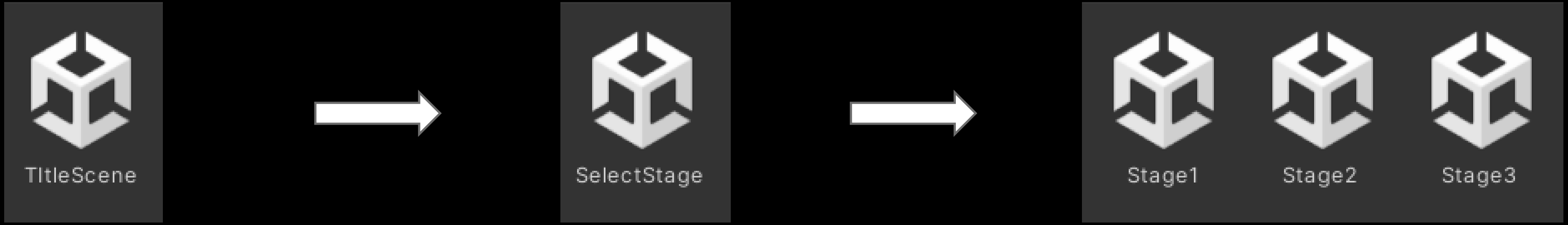
https://doggie-development.tistory.com/category/%EC%9C%A0%EB%8B%88%ED%8B%B0%20%EA%B0%95%EC%A2%8C/%EC%9C%A0%EB%8B%88%ED%8B%B0%20%EA%B8%B0%EC%B4%88%20%EA%B0%95%EC%A2%8C%20%282D%29



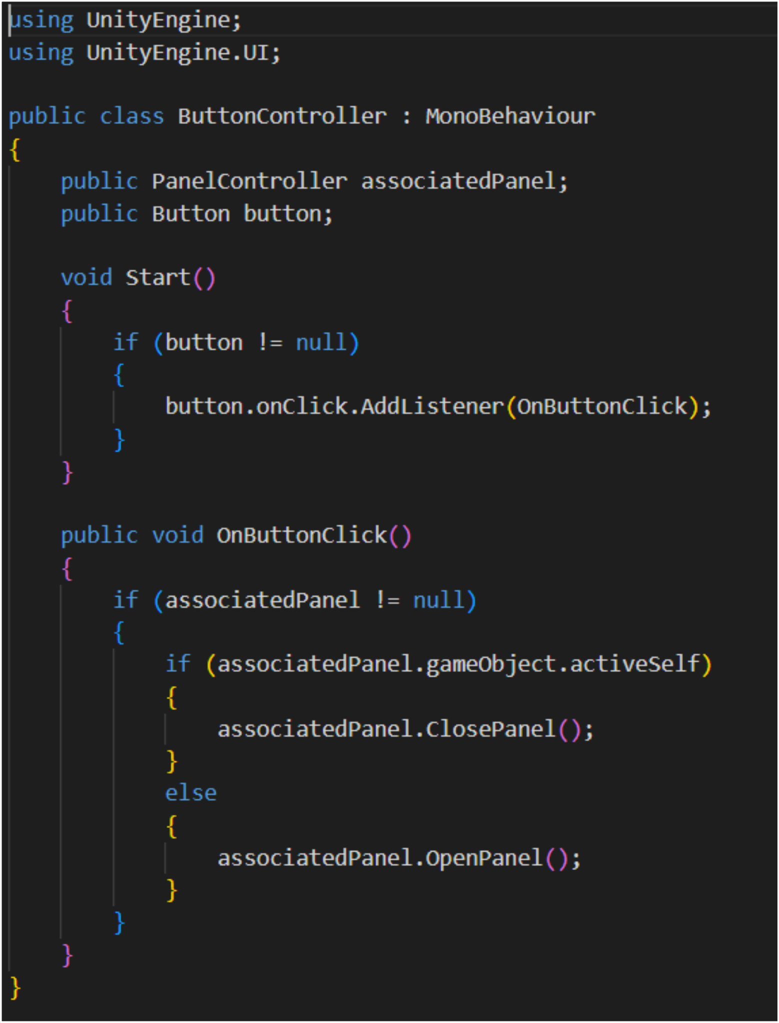
```
        else if(move_method == 2){
            speed_vec = Vector2.zero;

            if(Input.GetKey(KeyCode.RightArrow)){
                speed_vec.x += speed;
            }
            if(Input.GetKey(KeyCode.LeftArrow)){
                speed_vec.x -= speed;
            }
            if(Input.GetKey(KeyCode.UpArrow)){
                speed_vec.y += speed;
            }
            if(Input.GetKey(KeyCode.DownArrow)){
                speed_vec.y -= speed;
            }

            GetComponent<Rigidbody2D>().velocity = speed_vec;
        }
    }

    public void Dead(){
        Destroy(gameObject);
    }
}
```



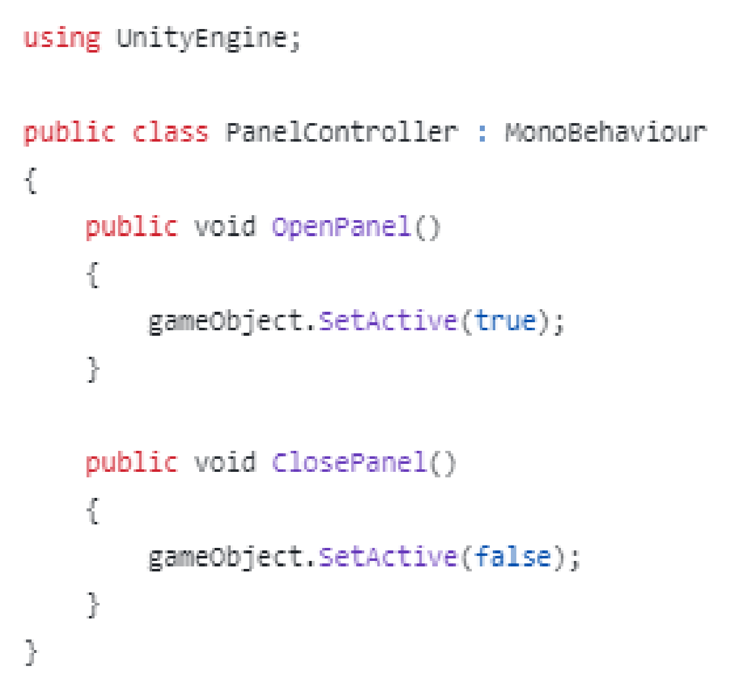
응용 구현



Button Controller / Panel Controller

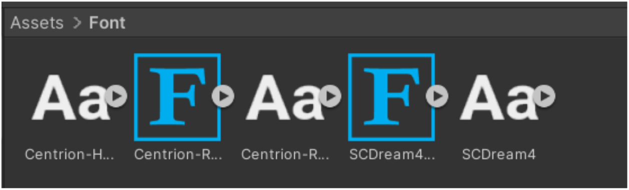


```
using UnityEngine;
using UnityEngine.UI;

public class ButtonController : MonoBehaviour
{
    public PanelController associatedPanel;
    public Button button;

    void Start()
    {
        if (button != null)
        {
            button.onClick.AddListener(OnButtonClick);
        }
    }

    public void OnButtonClick()
    {
        if (associatedPanel != null)
        {
            if (associatedPanel.gameObject.activeSelf)
            {
                associatedPanel.ClosePanel();
            }
            else
            {
                associatedPanel.OpenPanel();
            }
        }
    }
}
```



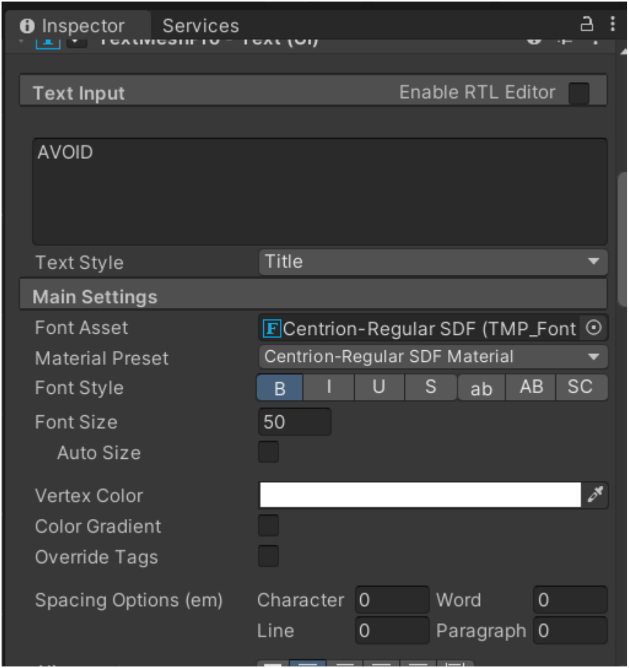
```
using UnityEngine;

public class PanelController : MonoBehaviour
{
    public void OpenPanel()
    {
        gameObject.SetActive(true);
    }

    public void ClosePanel()
    {
        gameObject.SetActive(false);
    }
}
```

TitleScene – main_menu



Assets > Font

Centrion-H... Centrion-R... Centrion-R... SCDream4... SCDream4



Inspector Services

Text Input — Enable RTL Editor □

AVOID

Text Style: Title

Main Settings

Font Asset: Centrion-Regular SDF (TMP_Font

Material Preset: Centrion-Regular SDF Material

Font Style: B I U S ab AB SC

Font Size: 50

Auto Size: □

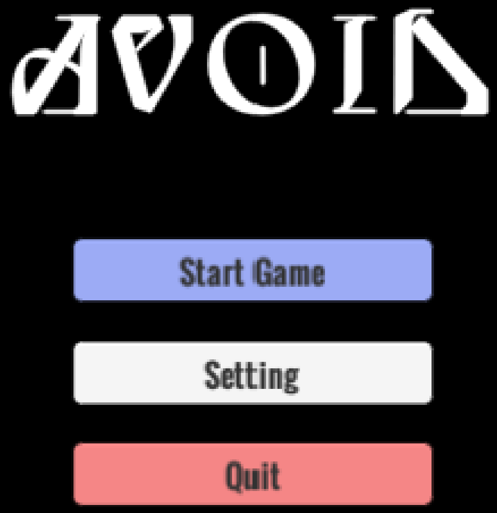
Vertex Color

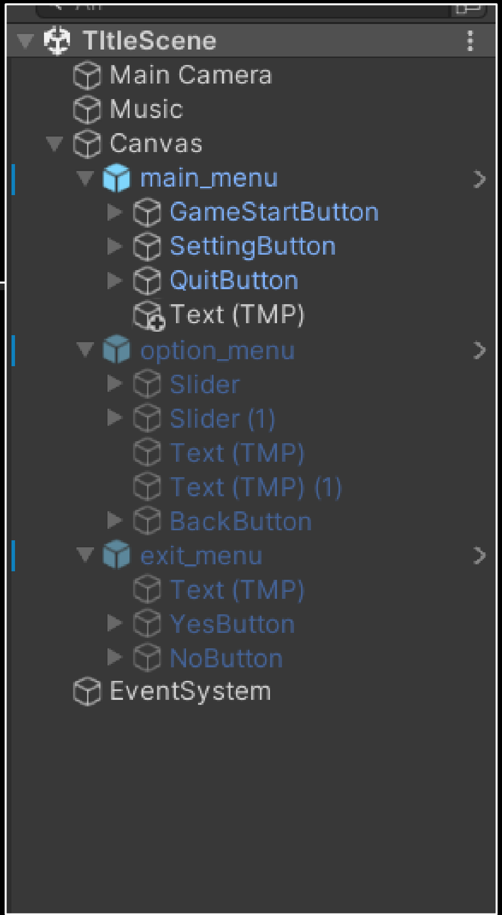
Color Gradient: □

Override Tags: □

Spacing Options (em): Character 0 Word 0

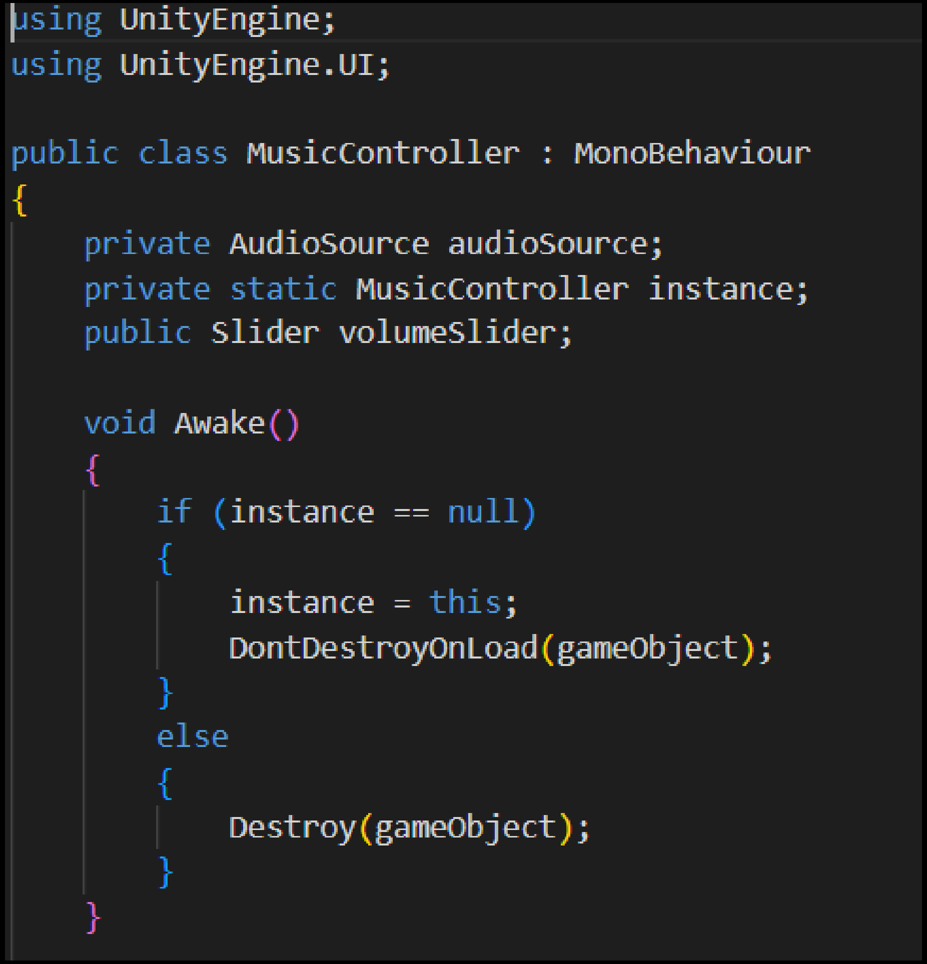
Line 0 Paragraph 0



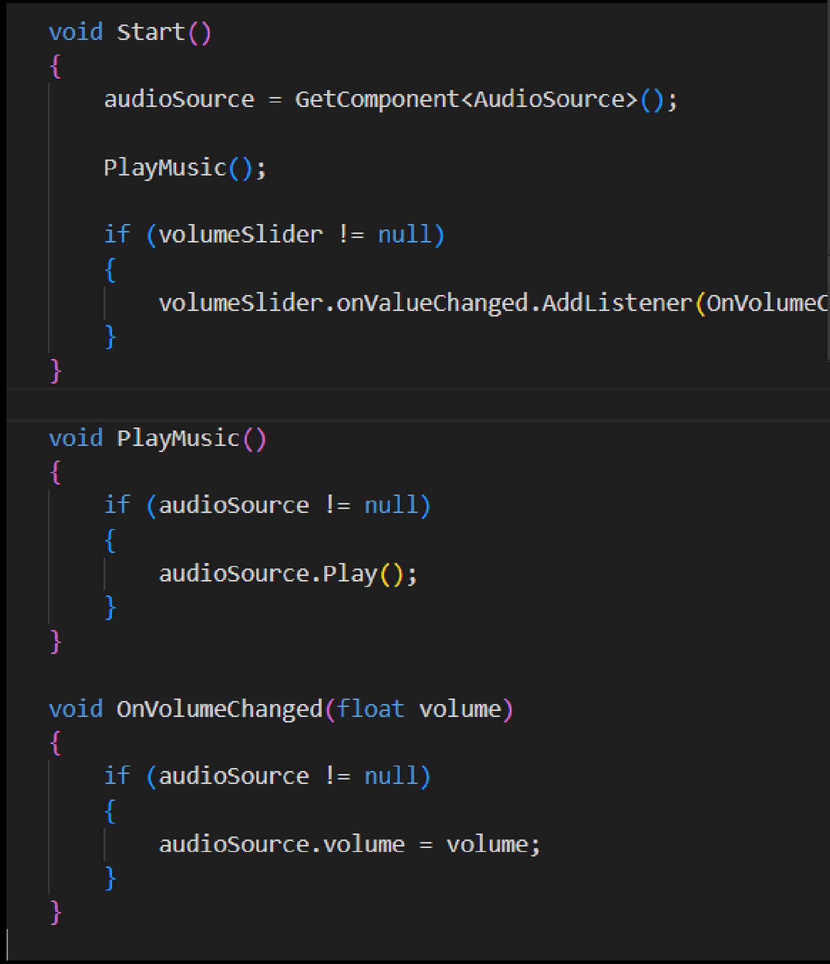


```
TitleScene
  Main Camera
  Music
  Canvas
    main_menu
      GameStartButton
      SettingButton
      QuitButton
      Text (TMP)
    option_menu
      Slider
      Slider (1)
      Text (TMP)
      Text (TMP) (1)
      BackButton
    exit_menu
      Text (TMP)
      YesButton
      NoButton
  EventSystem
```

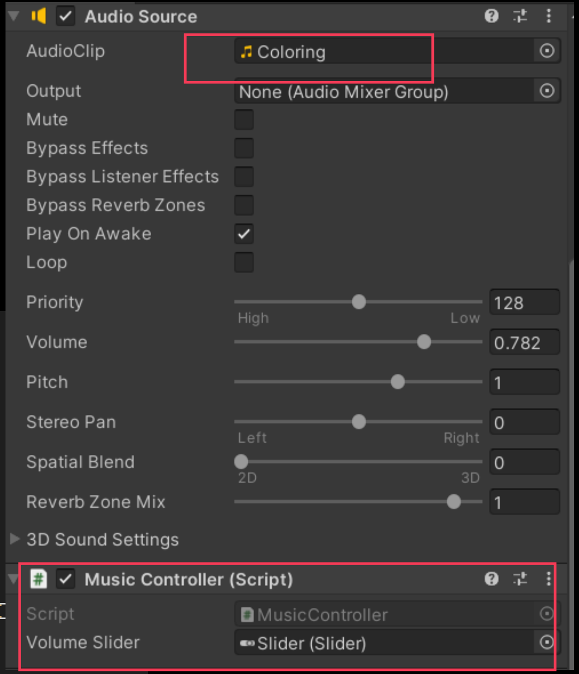
TitleScene – Music



```
using UnityEngine;
using UnityEngine.UI;

public class MusicController : MonoBehaviour
{
    private AudioSource audioSource;
    private static MusicController instance;
    public Slider volumeSlider;

    void Awake()
    {
        if (instance == null)
        {
            instance = this;
            DontDestroyOnLoad(gameObject);
        }
        else
        {
            Destroy(gameObject);
        }
    }
```



```
void Start()
{
    audioSource = GetComponent<AudioSource>();

    PlayMusic();

    if (volumeSlider != null)
    {
        volumeSlider.onValueChanged.AddListener(OnVolumeC
    }
}

void PlayMusic()
{
    if (audioSource != null)
    {
        audioSource.Play();
    }
}

void OnVolumeChanged(float volume)
{
    if (audioSource != null)
    {
        audioSource.volume = volume;
    }
}
```



☑ Audio Source

AudioClip: Coloring

Output: None (Audio Mixer Group)

Mute: □

Bypass Effects: □

Bypass Listener Effects: □

Bypass Reverb Zones: □

Play On Awake: ☑

Loop: □

Priority: 128 (High – Low)

Volume: 0.782

Pitch: 1

Stereo Pan: 0 (Left – Right)

Spatial Blend: 0 (2D – 3D)

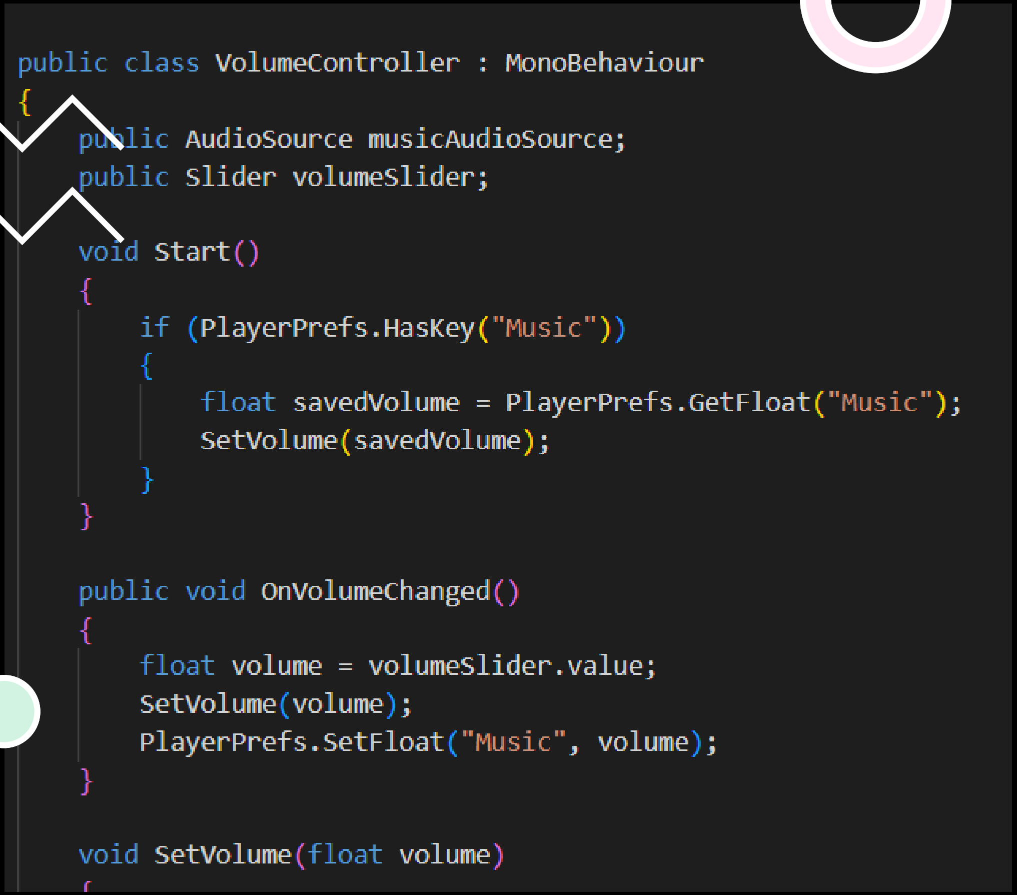
Reverb Zone Mix: 1

3D Sound Settings

☑ Music Controller (Script)

Script: MusicController

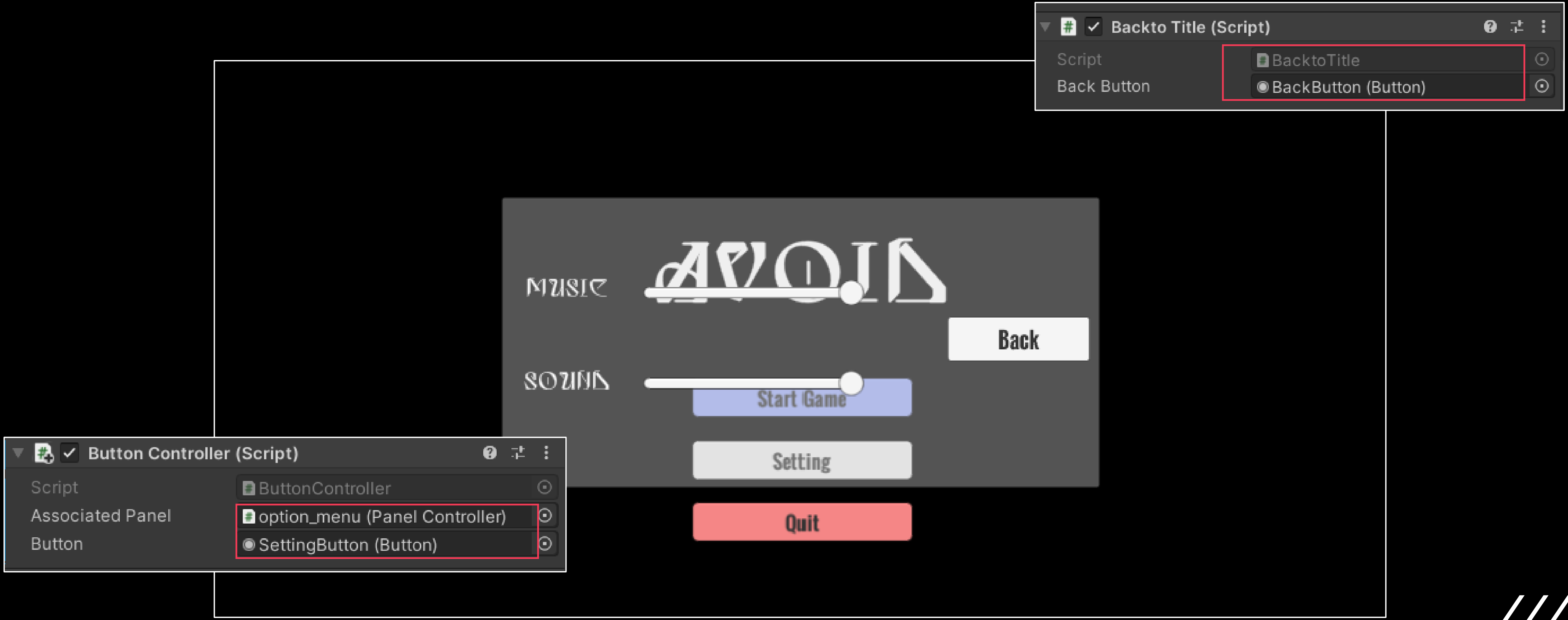
Volume Slider: Slider (Slider)



```
public class VolumeController : MonoBehaviour
{
    public AudioSource musicAudioSource;
    public Slider volumeSlider;

    void Start()
    {
        if (PlayerPrefs.HasKey("Music"))
        {
            float savedVolume = PlayerPrefs.GetFloat("Music");
            SetVolume(savedVolume);
        }
    }

    public void OnVolumeChanged()
    {
        float volume = volumeSlider.value;
        SetVolume(volume);
        PlayerPrefs.SetFloat("Music", volume);
    }

    void SetVolume(float volume)
    {
```

TITLESCENE – MUSIC

TitleScene – option_menu



Backto Title (Script)

Script: BacktoTitle

Back Button: BackButton (Button)

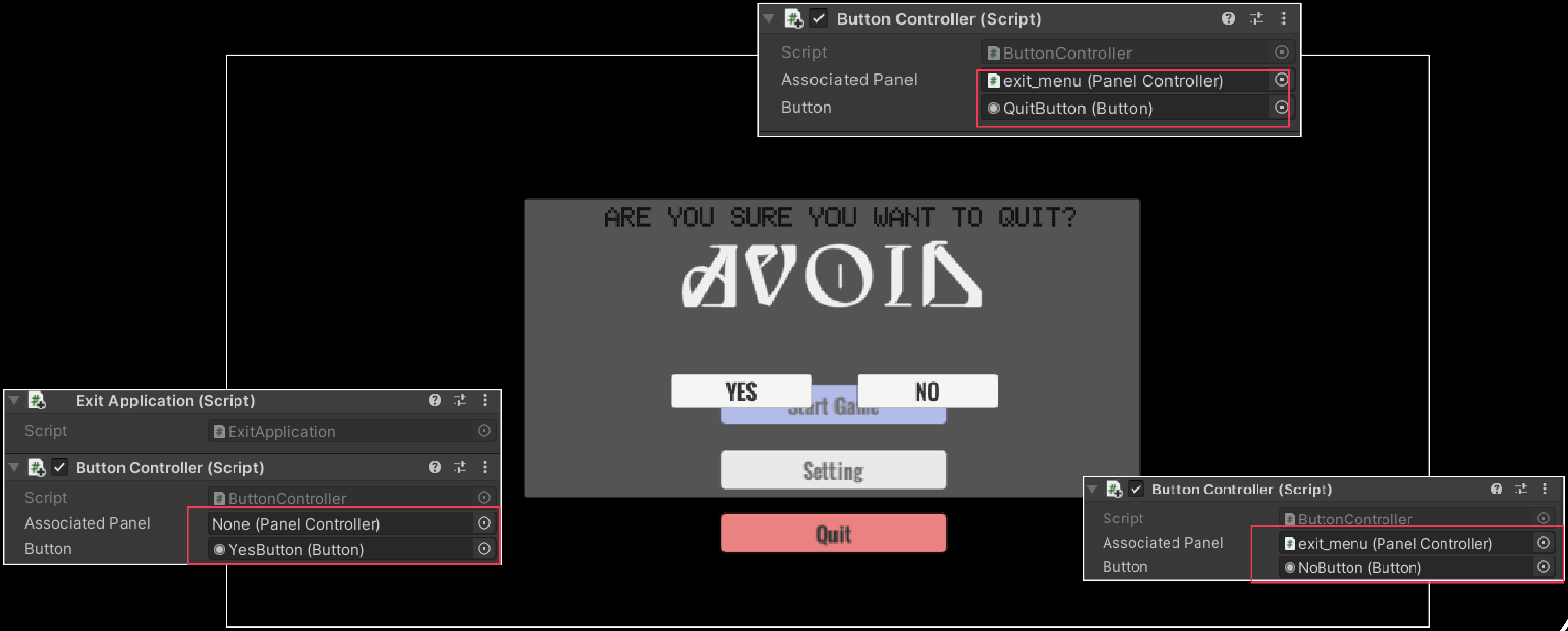
Button Controller (Script)

Script: ButtonController

Associated Panel: option_menu (Panel Controller)

Button: SettingButton (Button)

TitleScene – exit_menu



Button Controller (Script)

Script	ButtonController
Associated Panel	exit_menu (Panel Controller)
Button	QuitButton (Button)

Exit Application (Script)

Script	ExitApplication

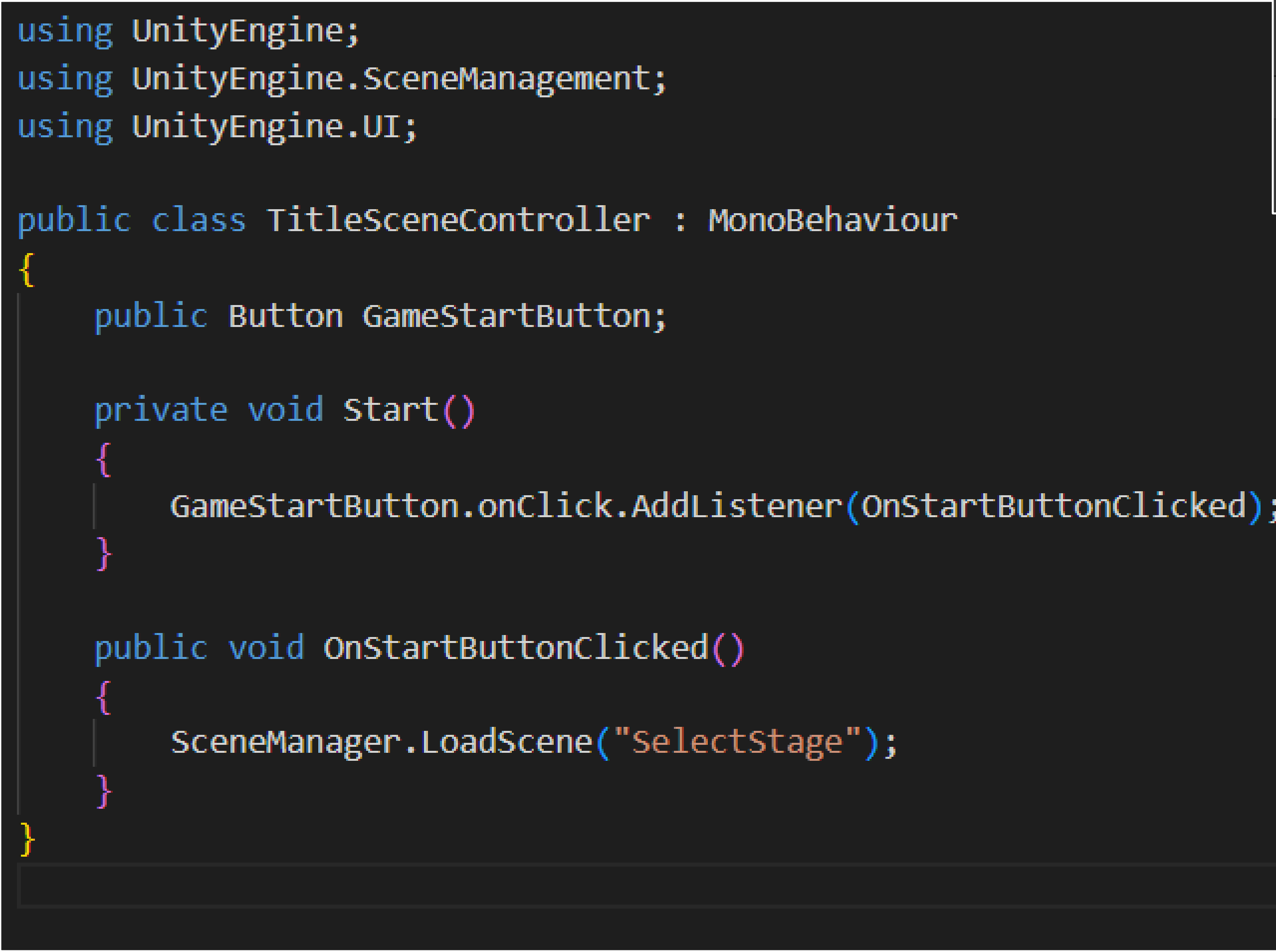
Button Controller (Script)

Script	ButtonController
Associated Panel	None (Panel Controller)
Button	YesButton (Button)

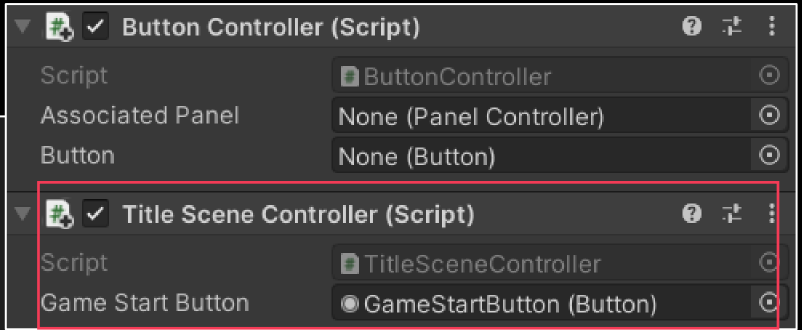
Button Controller (Script)

Script	ButtonController
Associated Panel	exit_menu (Panel Controller)
Button	NoButton (Button)

Game Start Button



```csharp
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class TitleSceneController : MonoBehaviour
{
    public Button GameStartButton;

    private void Start()
    {
        GameStartButton.onClick.AddListener(OnStartButtonClicked);
    }

    public void OnStartButtonClicked()
    {
        SceneManager.LoadScene("SelectStage");
    }
}
```



Button Controller (Script)

Script: ButtonController

Associated Panel: None (Panel Controller)

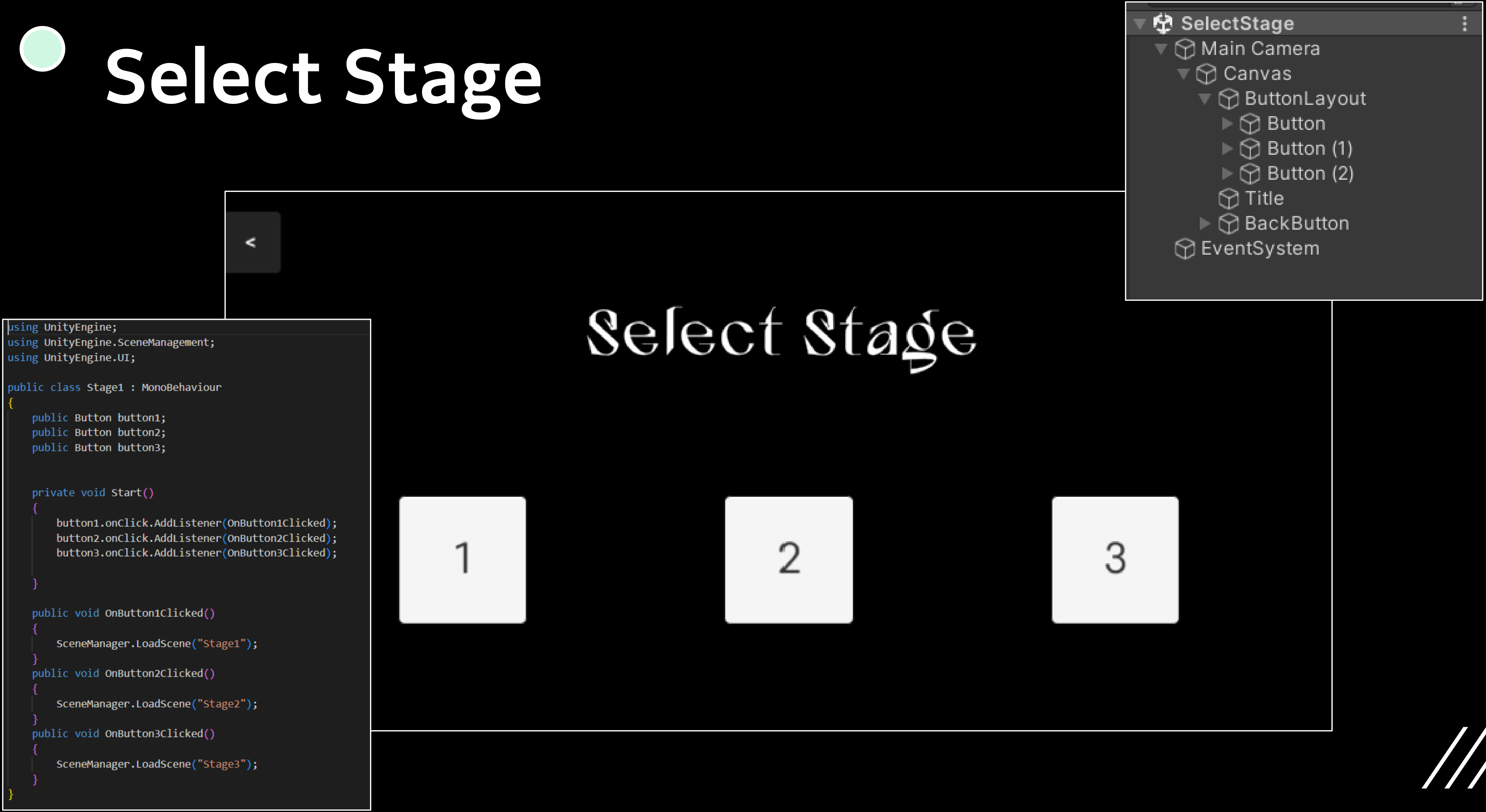
Button: None (Button)

Title Scene Controller (Script)

Script: TitleSceneController

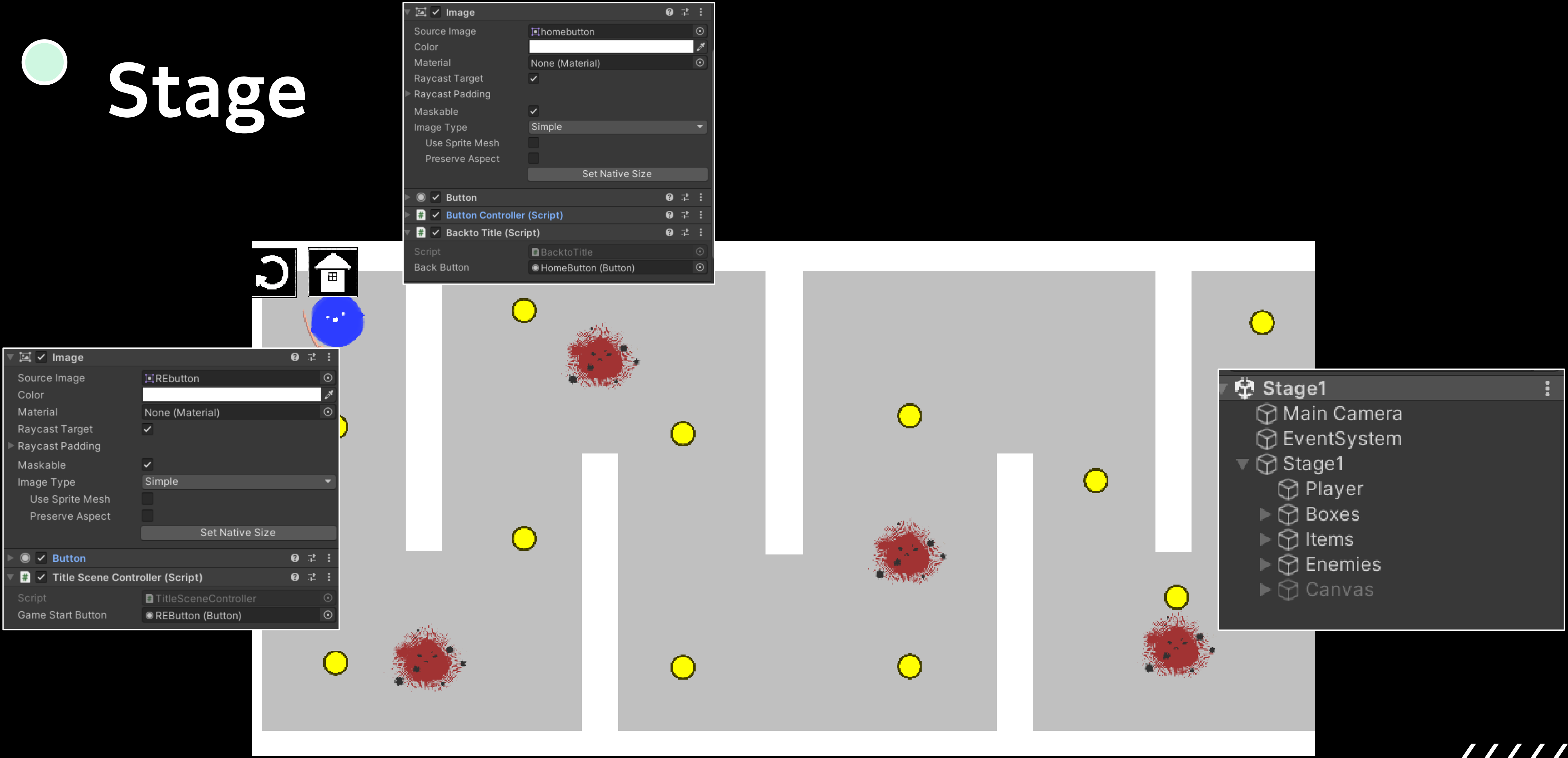
Game Start Button: GameStartButton (Button)

Select Stage



```
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class Stage1 : MonoBehaviour
{
    public Button button1;
    public Button button2;
    public Button button3;


    private void Start()
    {
        button1.onClick.AddListener(OnButton1Clicked);
        button2.onClick.AddListener(OnButton2Clicked);
        button3.onClick.AddListener(OnButton3Clicked);

    }

    public void OnButton1Clicked()
    {
        SceneManager.LoadScene("Stage1");
    }
    public void OnButton2Clicked()
    {
        SceneManager.LoadScene("Stage2");
    }
    public void OnButton3Clicked()
    {
        SceneManager.LoadScene("Stage3");
    }
}
```

Stage



☑ Image

Source Image: homebutton

Color:

Material: None (Material)

Raycast Target: ☑

Raycast Padding

Maskable: ☑

Image Type: Simple

Use Sprite Mesh: ☐

Preserve Aspect: ☐

Set Native Size

☑ Button

☑ Button Controller (Script)

☑ Backto Title (Script)

Script: BacktoTitle

Back Button: HomeButton (Button)

☑ Image

Source Image: REbutton

Color:

Material: None (Material)

Raycast Target: ☑

Raycast Padding

Maskable: ☑

Image Type: Simple

Use Sprite Mesh: ☐

Preserve Aspect: ☐

Set Native Size

☑ Button

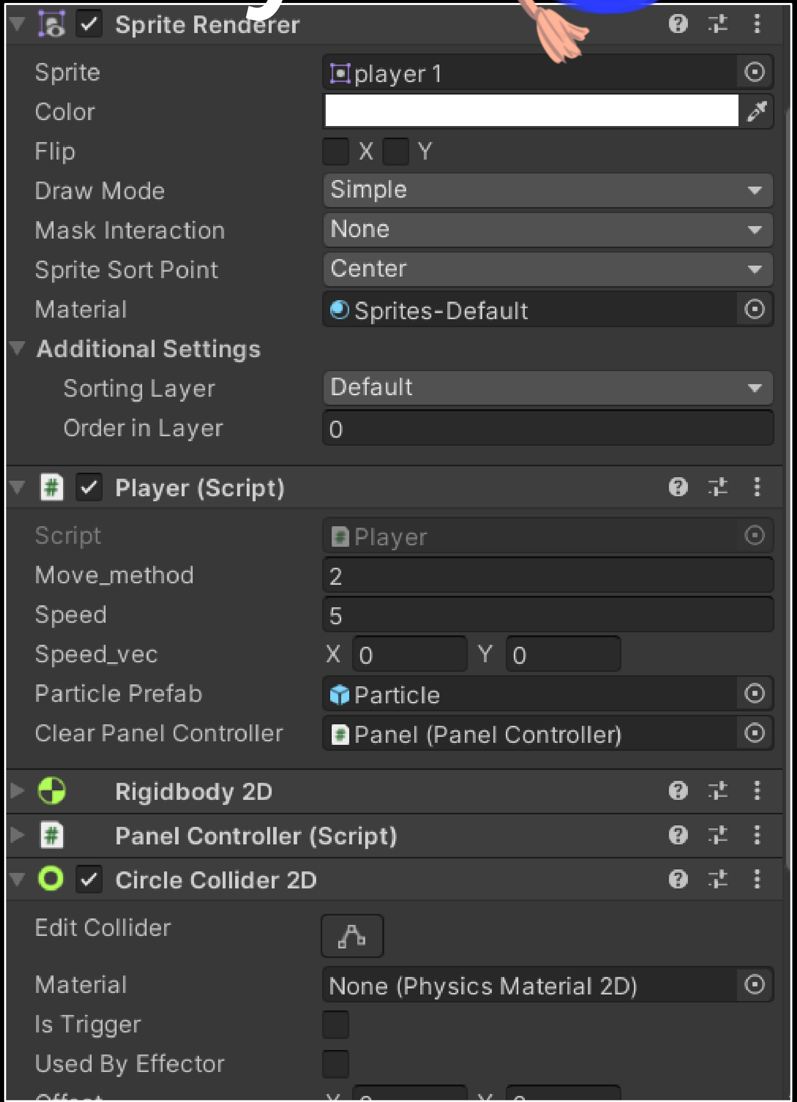
☑ Title Scene Controller (Script)

Script: TitleSceneController

Game Start Button: REButton (Button)

```
Stage1
  Main Camera
  EventSystem
  Stage1
    Player
    Boxes
    Items
    Enemies
    Canvas
```

Player



▼ ☑ Sprite Renderer

Sprite	player 1
Color	
Flip	□ X □ Y
Draw Mode	Simple
Mask Interaction	None
Sprite Sort Point	Center
Material	Sprites-Default

▼ Additional Settings

Sorting Layer	Default
Order in Layer	0

▼ ☑ Player (Script)

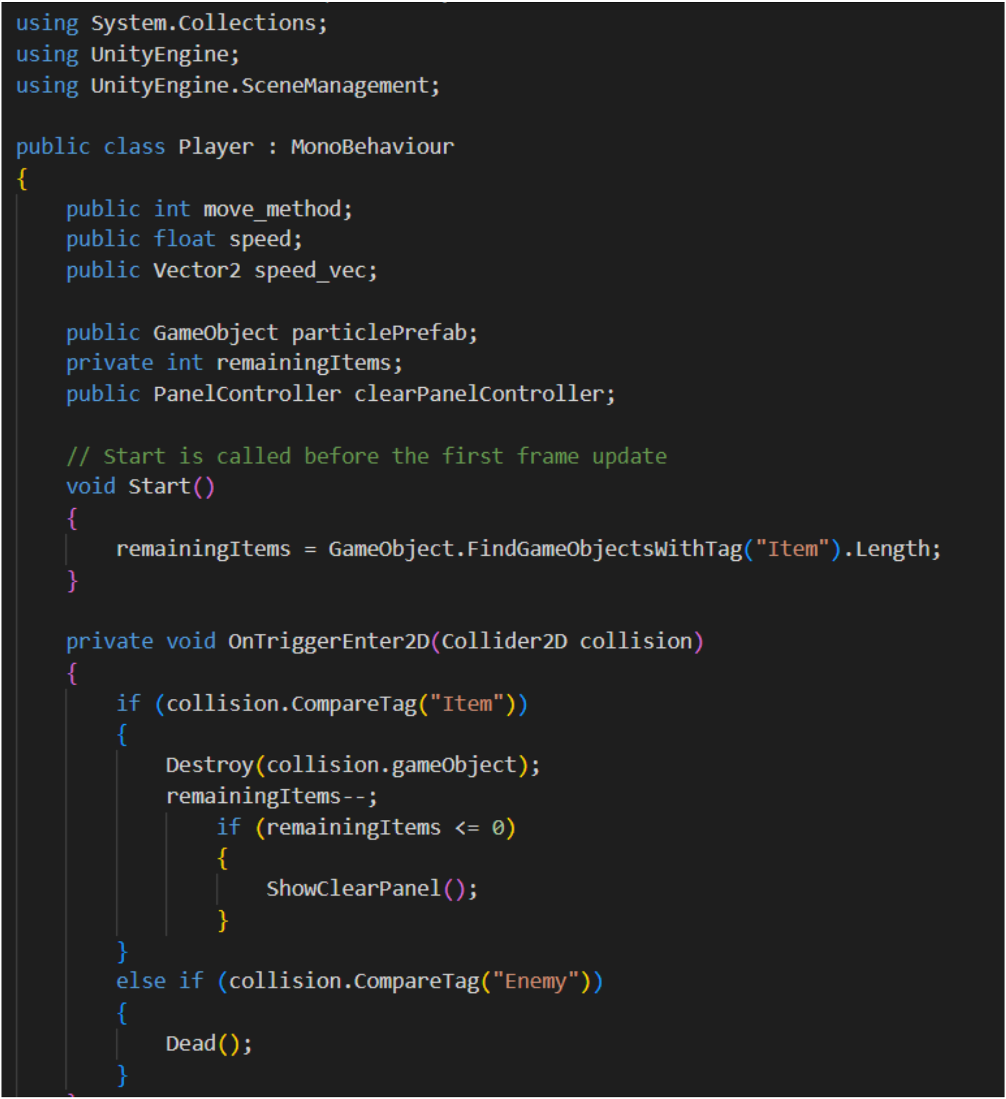
Script	Player
Move_method	2
Speed	5
Speed_vec	X 0 Y 0
Particle Prefab	Particle
Clear Panel Controller	Panel (Panel Controller)

▶ Rigidbody 2D

▶ Panel Controller (Script)

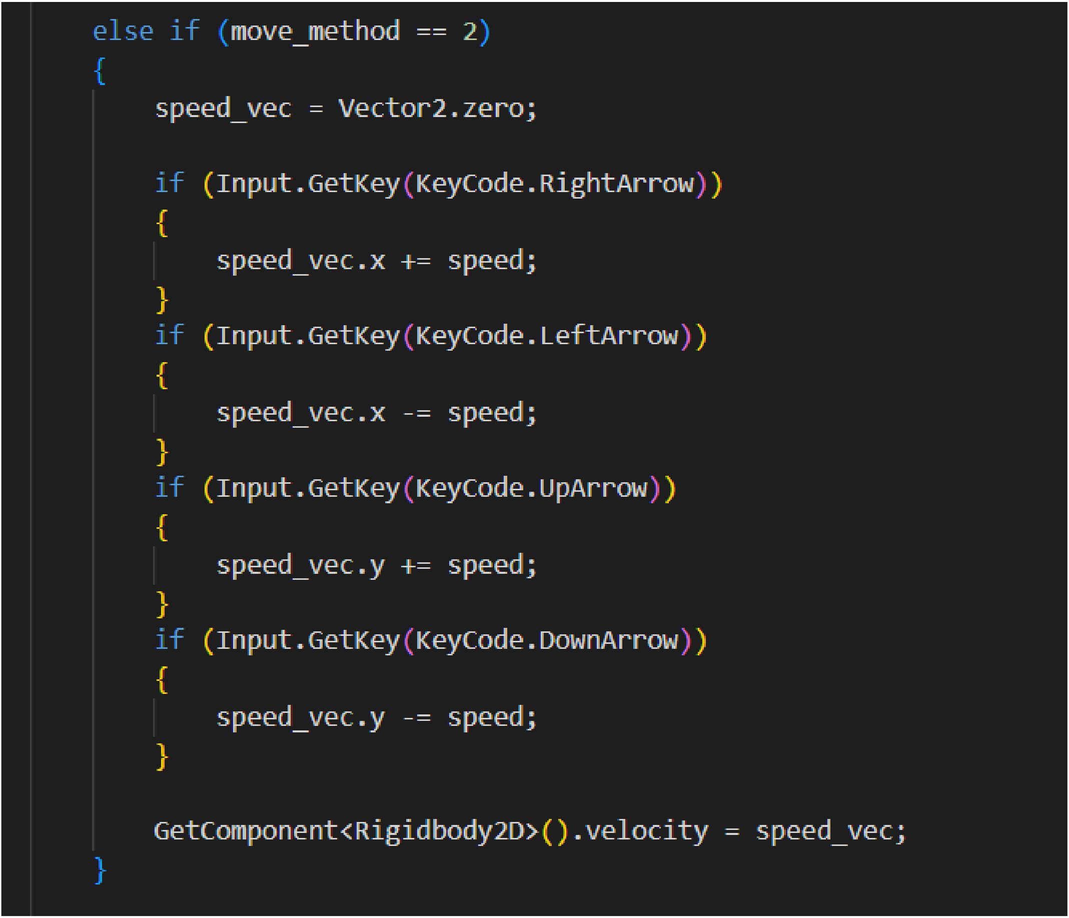
▼ ☑ Circle Collider 2D

Edit Collider	
Material	None (Physics Material 2D)
Is Trigger	□
Used By Effector	□
Offset	X 0 Y 0

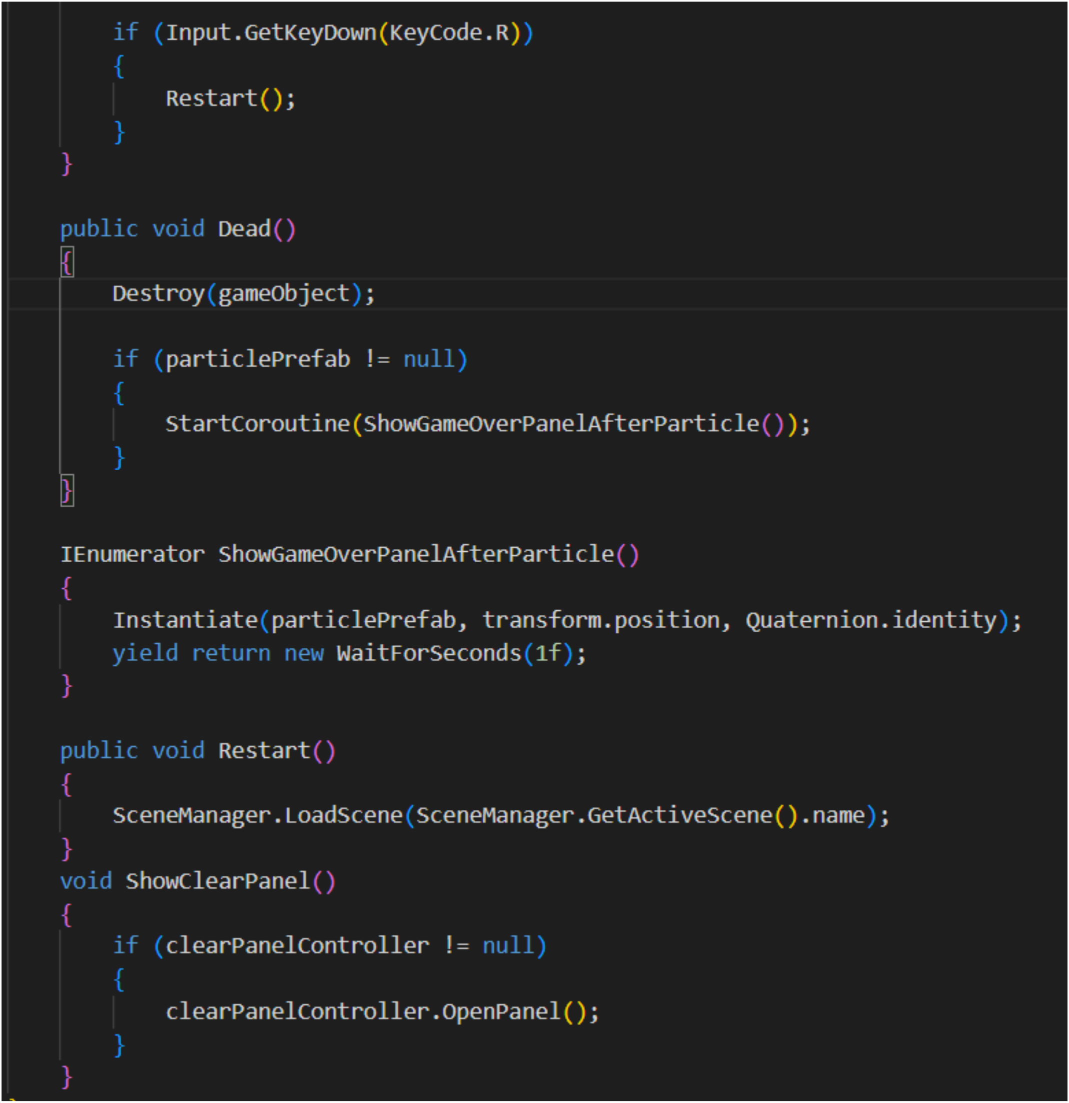


```
using System.Collections;
using UnityEngine;
using UnityEngine.SceneManagement;

public class Player : MonoBehaviour
{
    public int move_method;
    public float speed;
    public Vector2 speed_vec;

    public GameObject particlePrefab;
    private int remainingItems;
    public PanelController clearPanelController;

    // Start is called before the first frame update
    void Start()
    {
        remainingItems = GameObject.FindGameObjectsWithTag("Item").Length;
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Item"))
        {
            Destroy(collision.gameObject);
            remainingItems--;
            if (remainingItems <= 0)
            {
                ShowClearPanel();
            }
        }
        else if (collision.CompareTag("Enemy"))
        {
            Dead();
        }
```

Player

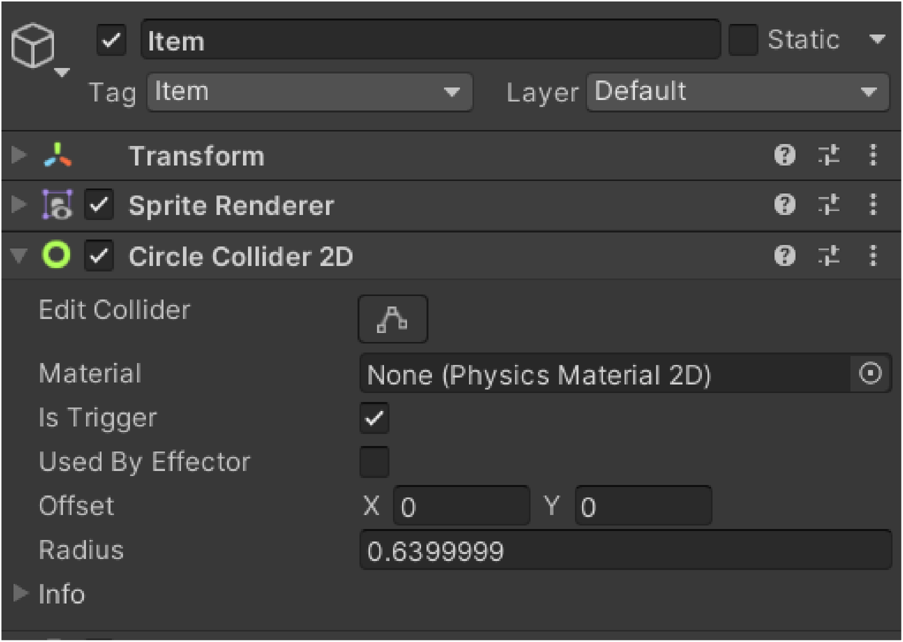


```
else if (move_method == 2)
{
    speed_vec = Vector2.zero;

    if (Input.GetKey(KeyCode.RightArrow))
    {
        speed_vec.x += speed;
    }
    if (Input.GetKey(KeyCode.LeftArrow))
    {
        speed_vec.x -= speed;
    }
    if (Input.GetKey(KeyCode.UpArrow))
    {
        speed_vec.y += speed;
    }
    if (Input.GetKey(KeyCode.DownArrow))
    {
        speed_vec.y -= speed;
    }

    GetComponent<Rigidbody2D>().velocity = speed_vec;
}
```



```
    if (Input.GetKeyDown(KeyCode.R))
    {
        Restart();
    }
}

public void Dead()
{
    Destroy(gameObject);

    if (particlePrefab != null)
    {
        StartCoroutine(ShowGameOverPanelAfterParticle());
    }
}

IEnumerator ShowGameOverPanelAfterParticle()
{
    Instantiate(particlePrefab, transform.position, Quaternion.identity);
    yield return new WaitForSeconds(1f);
}

public void Restart()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}
void ShowClearPanel()
{
    if (clearPanelController != null)
    {
        clearPanelController.OpenPanel();
    }
}
```

Item / Box



☑ Item □ Static

Tag Item Layer Default

Transform

☑ Sprite Renderer

☑ Circle Collider 2D

Edit Collider

Material None (Physics Material 2D)

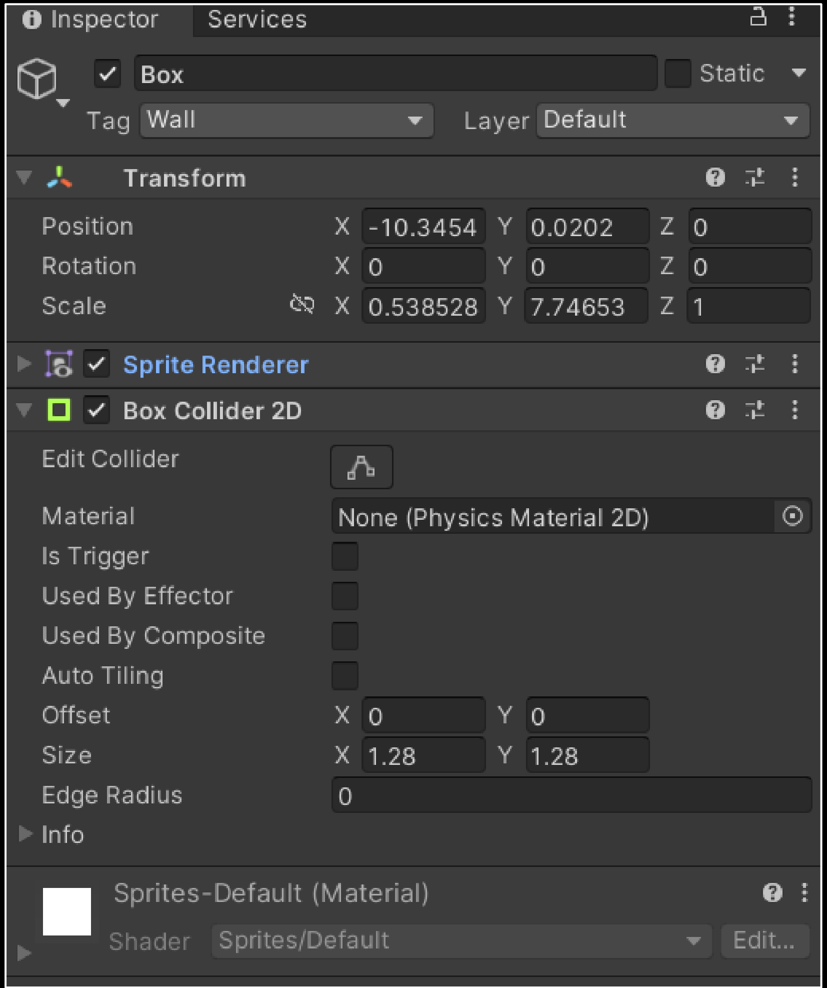
Is Trigger ☑

Used By Effector □

Offset X 0 Y 0

Radius 0.6399999

Info



☑ Box □ Static

Tag Wall Layer Default

Transform

Position X -10.3454 Y 0.0202 Z 0

Rotation X 0 Y 0 Z 0

Scale X 0.538528 Y 7.74653 Z 1

☑ Sprite Renderer

☑ Box Collider 2D

Edit Collider

Material None (Physics Material 2D)

Is Trigger □

Used By Effector □

Used By Composite □

Auto Tiling □

Offset X 0 Y 0

Size X 1.28 Y 1.28

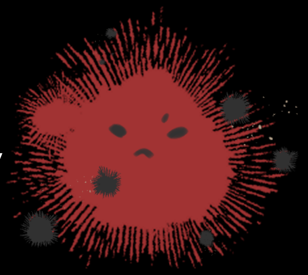
Edge Radius 0

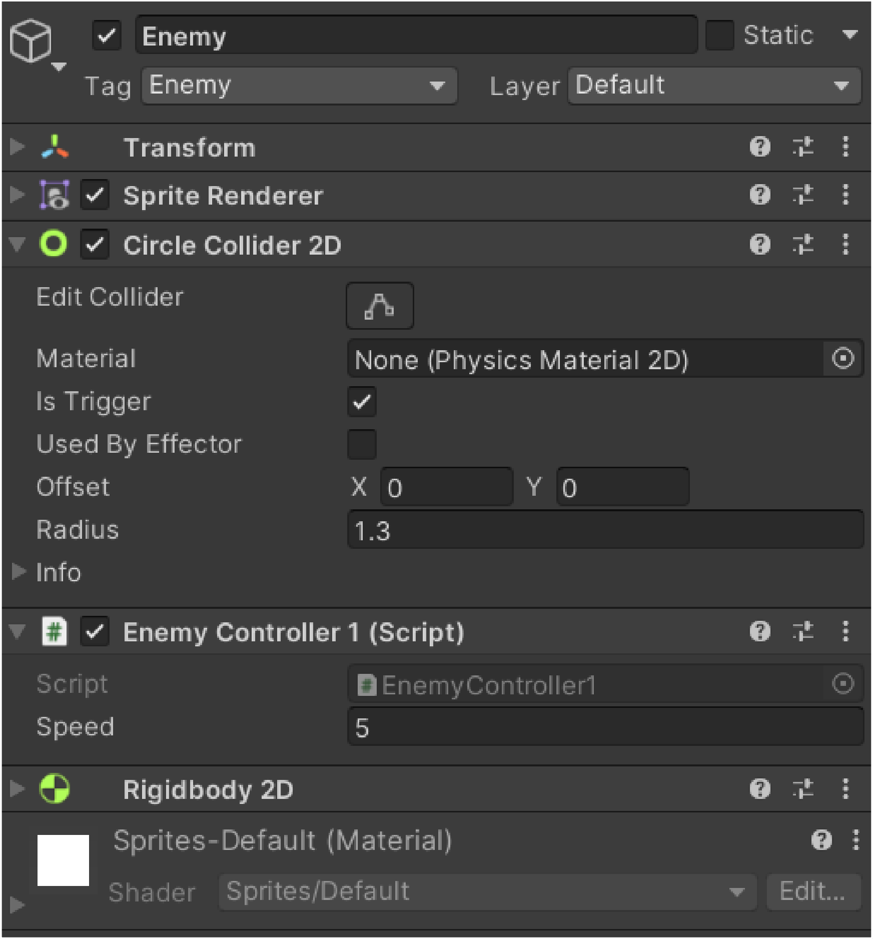
Info

Sprites-Default (Material)

Shader Sprites/Default Edit...

Enemy





☑ Enemy □ Static

Tag Enemy Layer Default

Transform

☑ Sprite Renderer

☑ Circle Collider 2D

Edit Collider

Material: None (Physics Material 2D)

Is Trigger: ☑

Used By Effector: □

Offset: X 0 Y 0

Radius: 1.3

Info

☑ Enemy Controller 1 (Script)

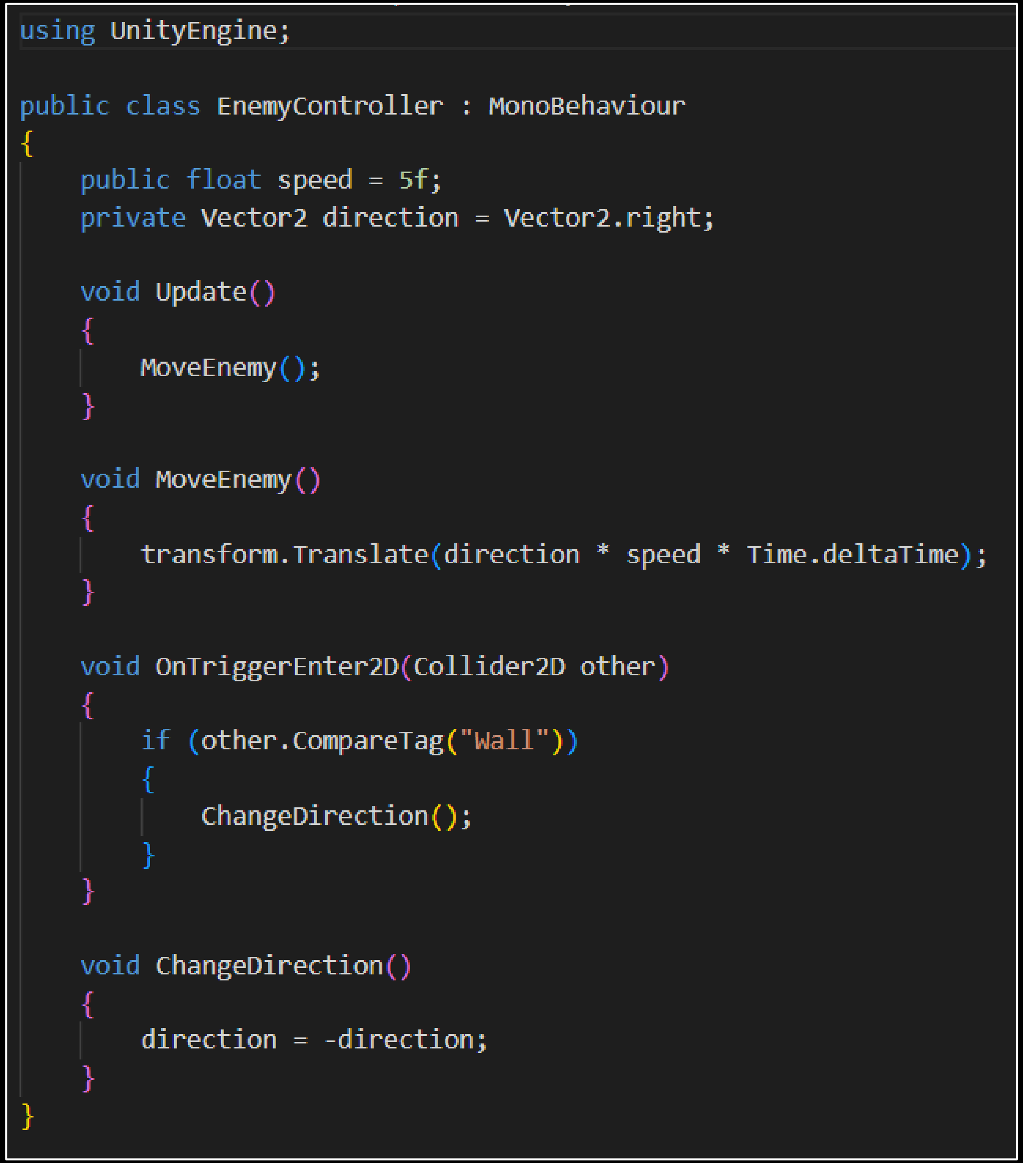
Script: EnemyController1

Speed: 5

Rigidbody 2D

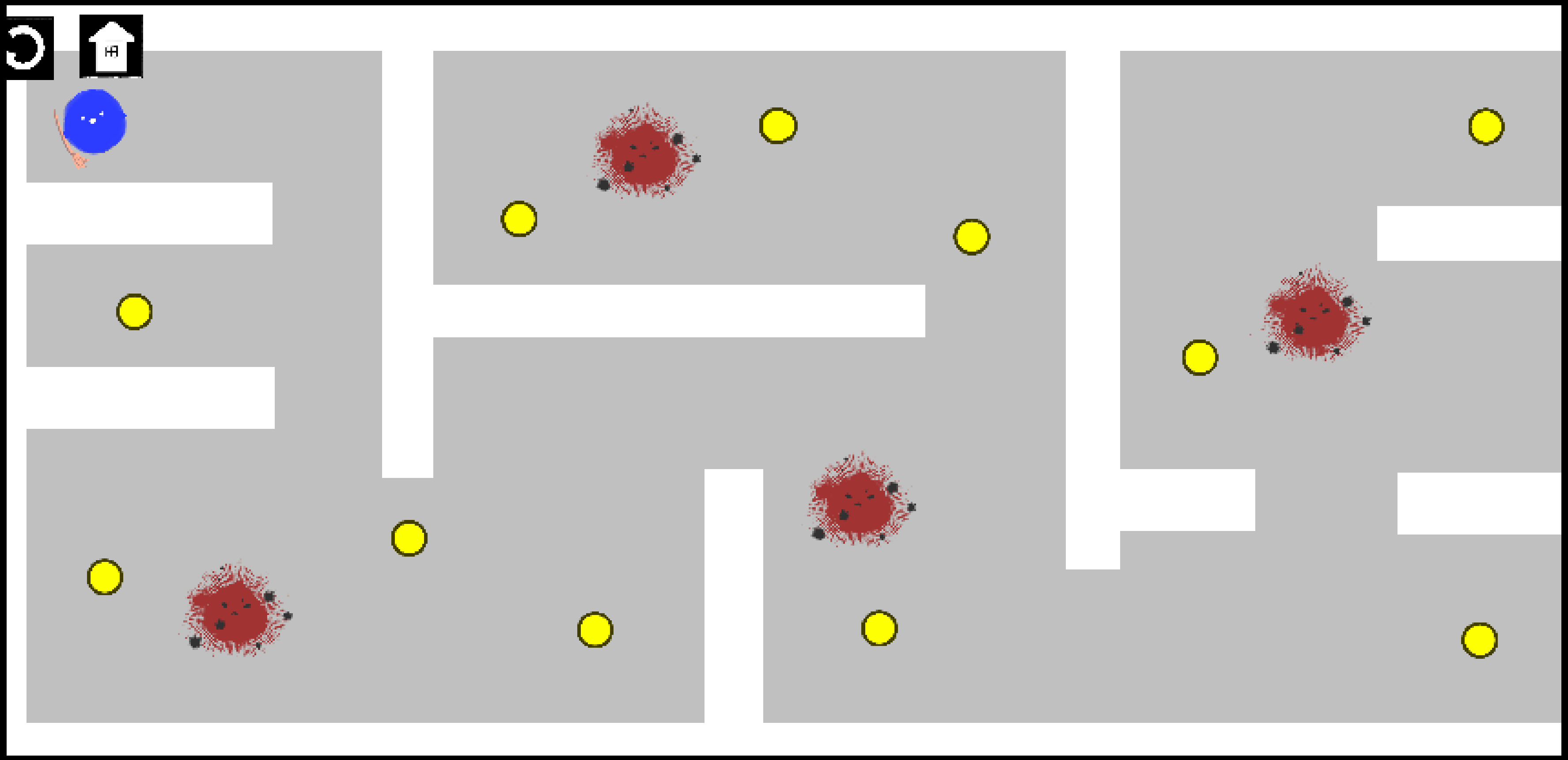
Sprites-Default (Material)

Shader: Sprites/Default Edit...

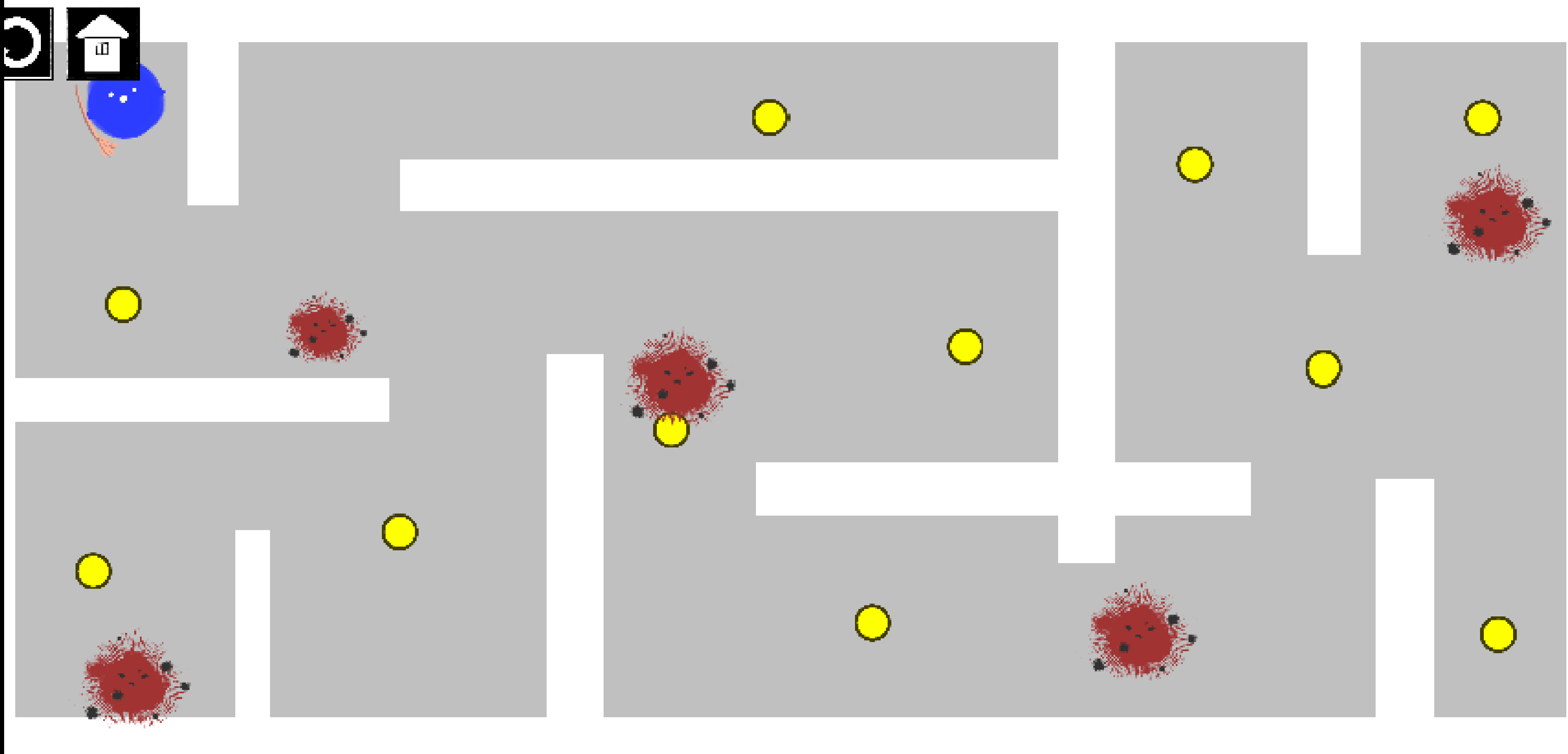


```
using UnityEngine;

public class EnemyController : MonoBehaviour
{
    public float speed = 5f;
    private Vector2 direction = Vector2.right;

    void Update()
    {
        MoveEnemy();
    }

    void MoveEnemy()
    {
        transform.Translate(direction * speed * Time.deltaTime);
    }

    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Wall"))
        {
            ChangeDirection();
        }
    }

    void ChangeDirection()
    {
        direction = -direction;
    }
}
```

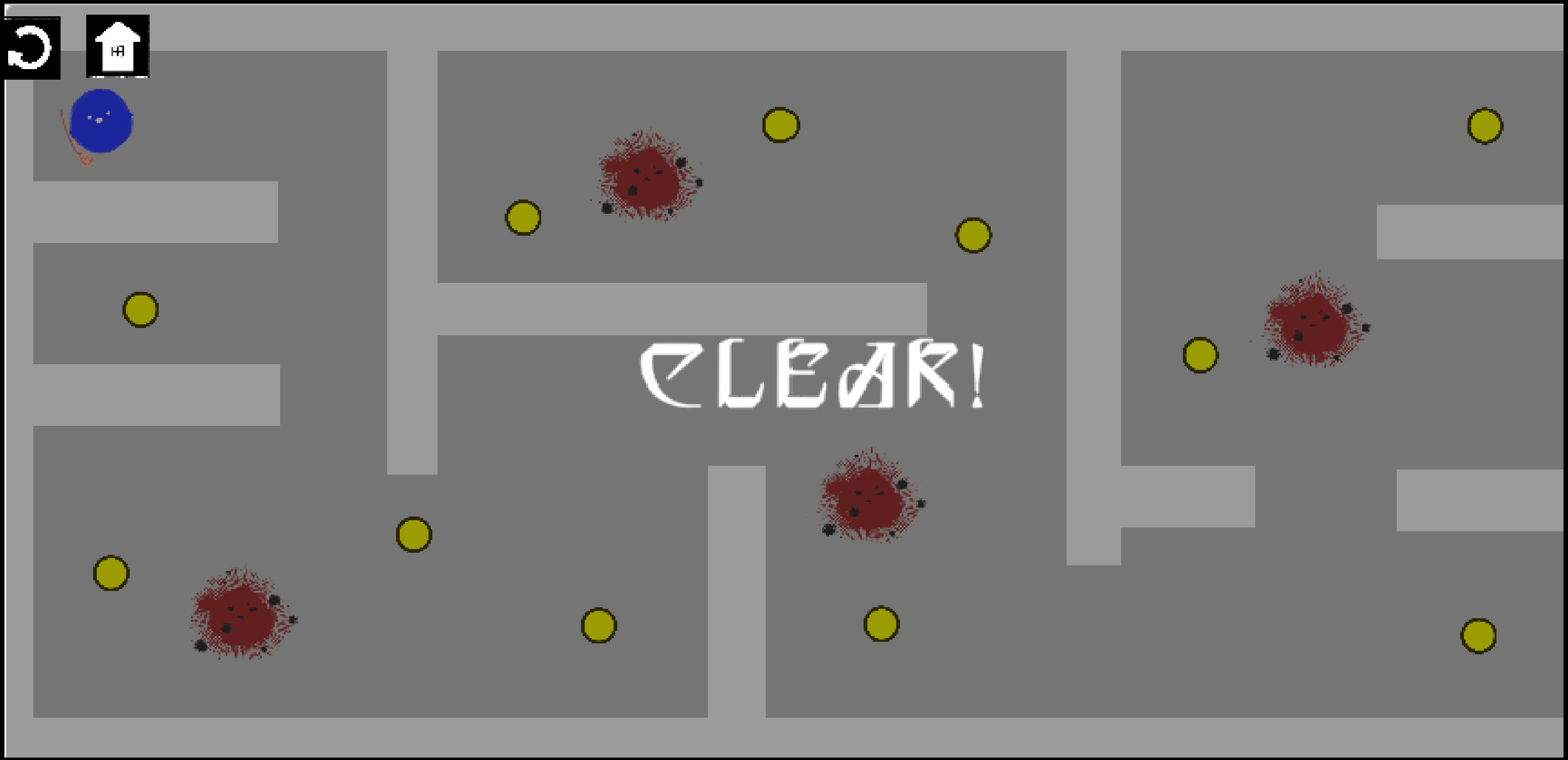
● Stage2



Stage3



Clear



Reference

- https://doggie-development.tistory.com/7
- https://fun-coding-study.tistory.com/10
- https://docs.unity3d.com/kr/560/Manual/class-Font.html
- https://chat.openai.com