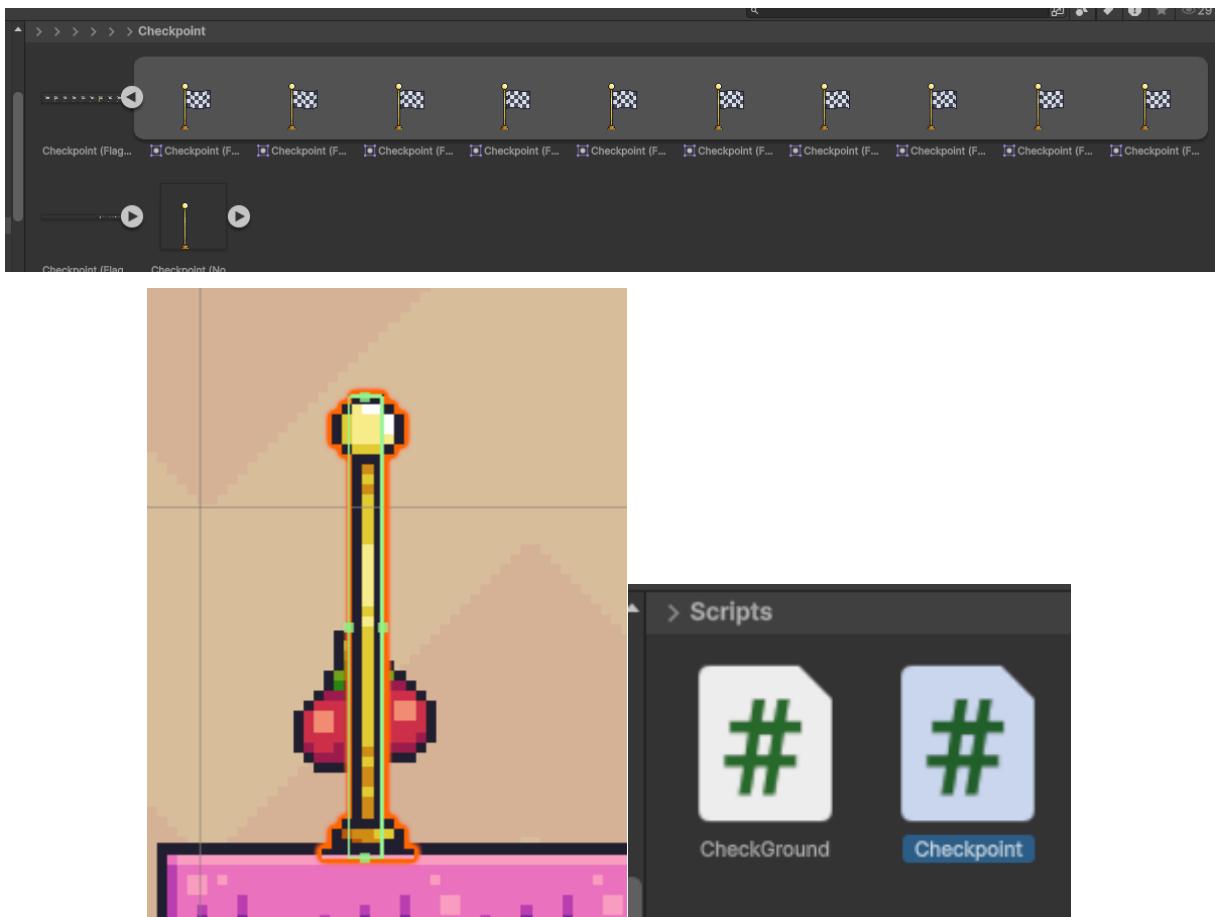


## **Tutoriales**

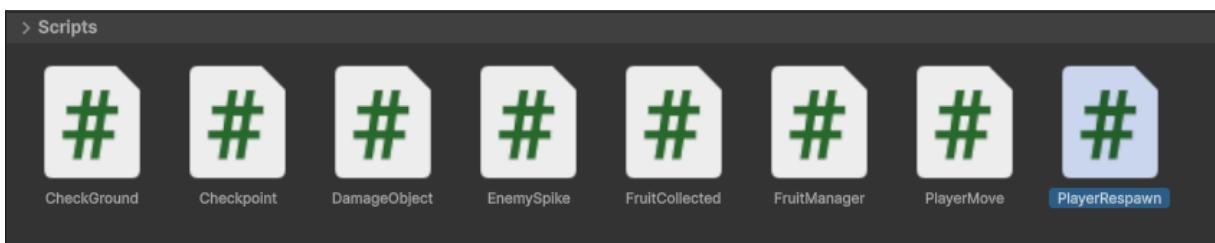
<b>Tutorial 7 .....</b>	<b>2</b>
<b>Tutorial 8.....</b>	<b>5</b>
<b>Tutorial 9.....</b>	<b>7</b>
<b>Tutorial 10 .....</b>	<b>13</b>
<b>Tutorial 11 .....</b>	<b>16</b>
<b>Tutorial 12 .....</b>	<b>19</b>

## Tutorial 7



Assets > Scripts > C# Checkpoint.cs > Checkpoint > OnTriggerEnter2D

```
1  using UnityEngine;
2
3  // Oscar Ulises Ramirez Cruz
4
5  public class Checkpoint : MonoBehaviour
6  {
7      private void OnTriggerEnter2D(Collider2D collision)
8      {
9          if (collision.CompareTag("Player"))
10         {
11             Debug.Log("Checkpoint reached!");
12         }
13     }
14 }
15 }
```



```

Assets > Scripts > C# PlayerRespawn.cs > PlayerRespawn > Start
1  using UnityEngine;
2
3  // Oscar Ulises Ramirez Cruz
4
5  1 référence | Script Unity
6  public class PlayerRespawn : MonoBehaviour
7  {
8
9      2 références | 2 références
10     private float chPntX, chPntY;
11
12
13     0 références | Message Unity
14     void Start()
15     {
16
17         if (PlayerPrefs.GetFloat("chPntX") != 0 && PlayerPrefs.GetFloat("chPntY") != 0)
18         {
19             chPntX = PlayerPrefs.GetFloat("chPntX");
20             chPntY = PlayerPrefs.GetFloat("chPntY");
21
22             transform.position = new Vector2(chPntX, chPntY);
23         }
24
25     }
26
27 }

```

```

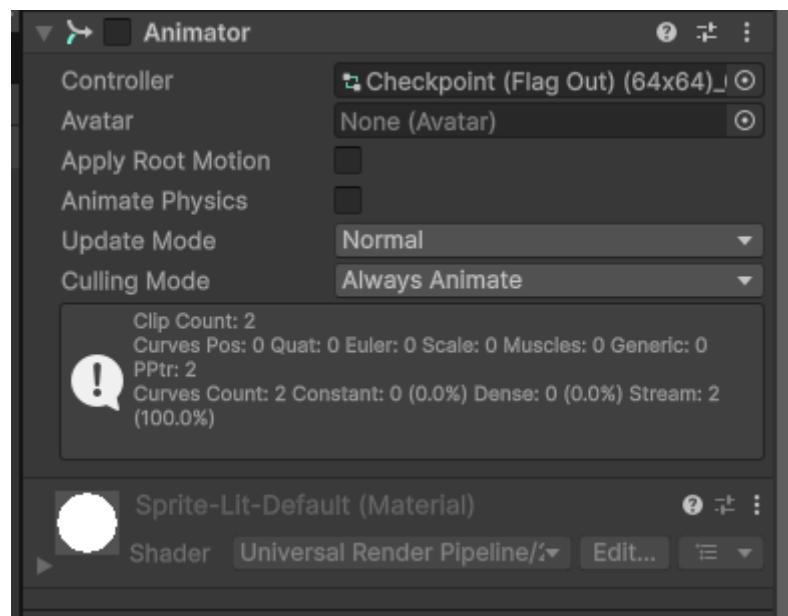
0 références | Message Unity
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.CompareTag("Player"))
    {
        collision.GetComponent<PlayerRespawn>().ReachedCheckPoint(transform.position.x, transform.position.y);
    }
}

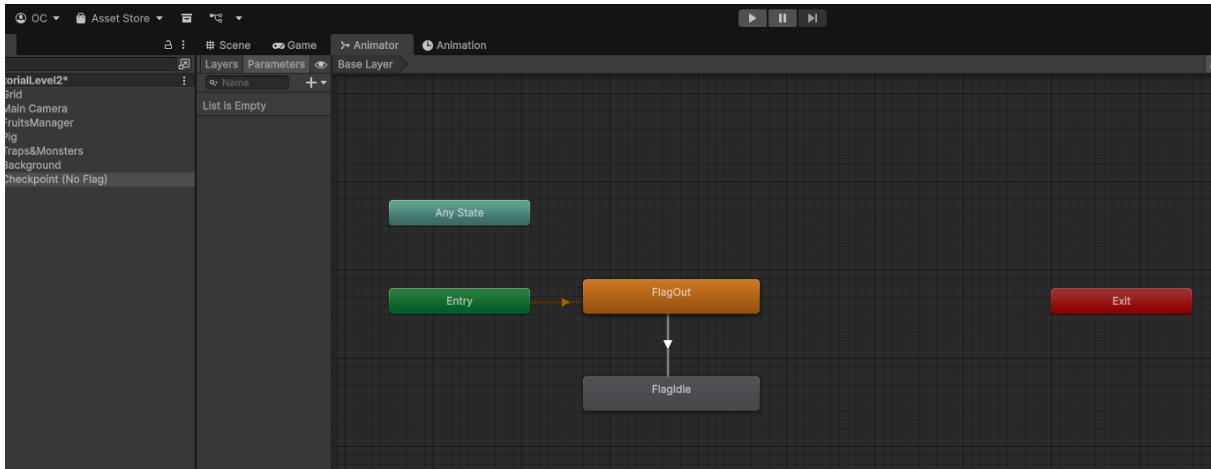
```

```

    GetComponent<Animator>().enabled = true;
}

```





**Inspector**

**Text (TMP)**

- Static:
- Tag: Untagged
- Layer: UI

**Rect Transform**

center	Pos X: -1870	Pos Y: -1030	Pos Z: 0
middle	Width: 200	Height: 50	<input type="button"/> R
Min	0.5	0.5	
Max	0.5	0.5	
Pivot	0.5	0.5	
Rotation	0	0	0
Scale	1	1	1

**Canvas Renderer**

Cull Transparent Mesh:

**TextMeshPro - Text (UI)**

Text Input: All Fruits collected

Enable RTL Editor:

**Hierarchy**

- TutorialLevel2\*
  - Grid
  - Main Camera
  - FruitsManager
  - Pig
  - Traps&Monsters
  - Background
  - Checkpoint (No Flag)
  - Canvas
    - Text (TMP)
  - EventSystem

## Tutorial 8

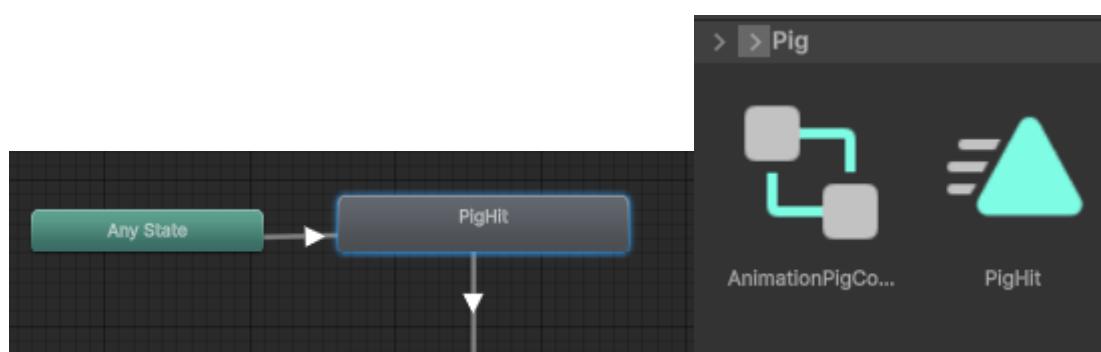
```
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;

// Oscar Ulises Ramirez Cruz

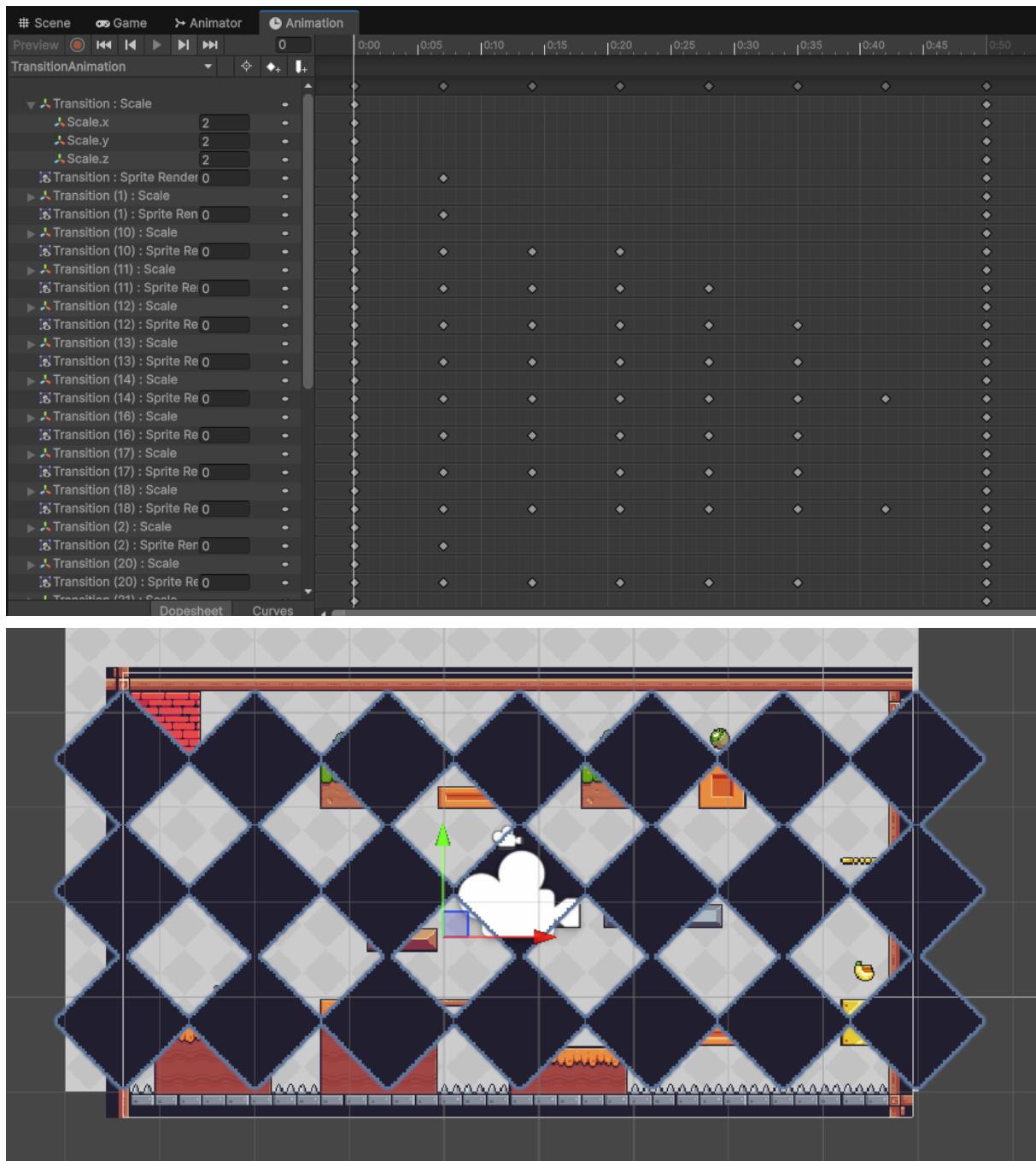
0 références | Script Unity (3 références de ressources)
public class FruitManager : MonoBehaviour
{
    1 référence | Champ Unity sérialisé
    public TMP_Text txtFruitsWin;

    0 références | Message Unity
    private void Update()
    {
        AllFruitCollected();
    }
    1 référence
    public void AllFruitCollected()
    {
        if (transform.childCount == 0)
        {
            Debug.Log("All fruits collected!");
            txtFruitsWin.enabled = true;
            Invoke("ChangeScene", 2f);
        }
    }

    0 références
    void ChangeScene()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }
}
```



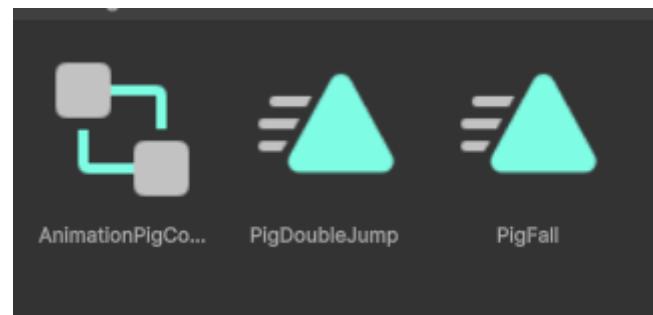
```
0 références
public void PlayerDamaged()
{
    anim.Play("PigHit");
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}
```



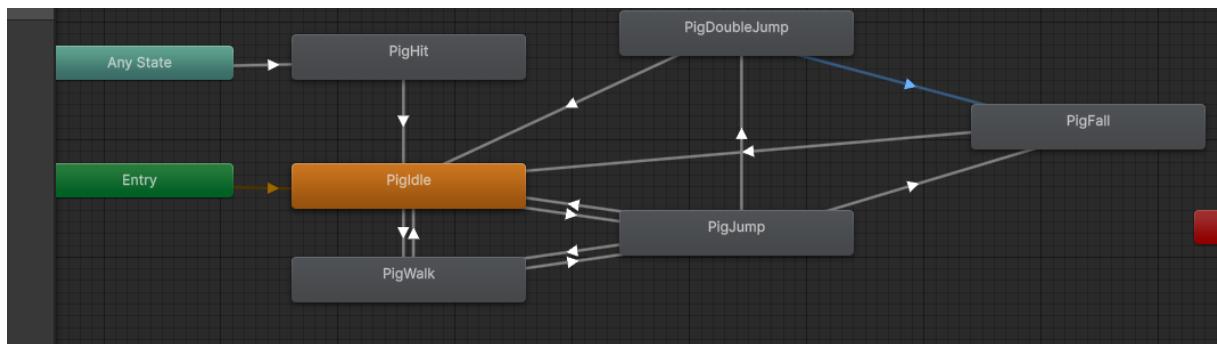
1 référence | Champ Unity sérialisé  
public GameObject transition;

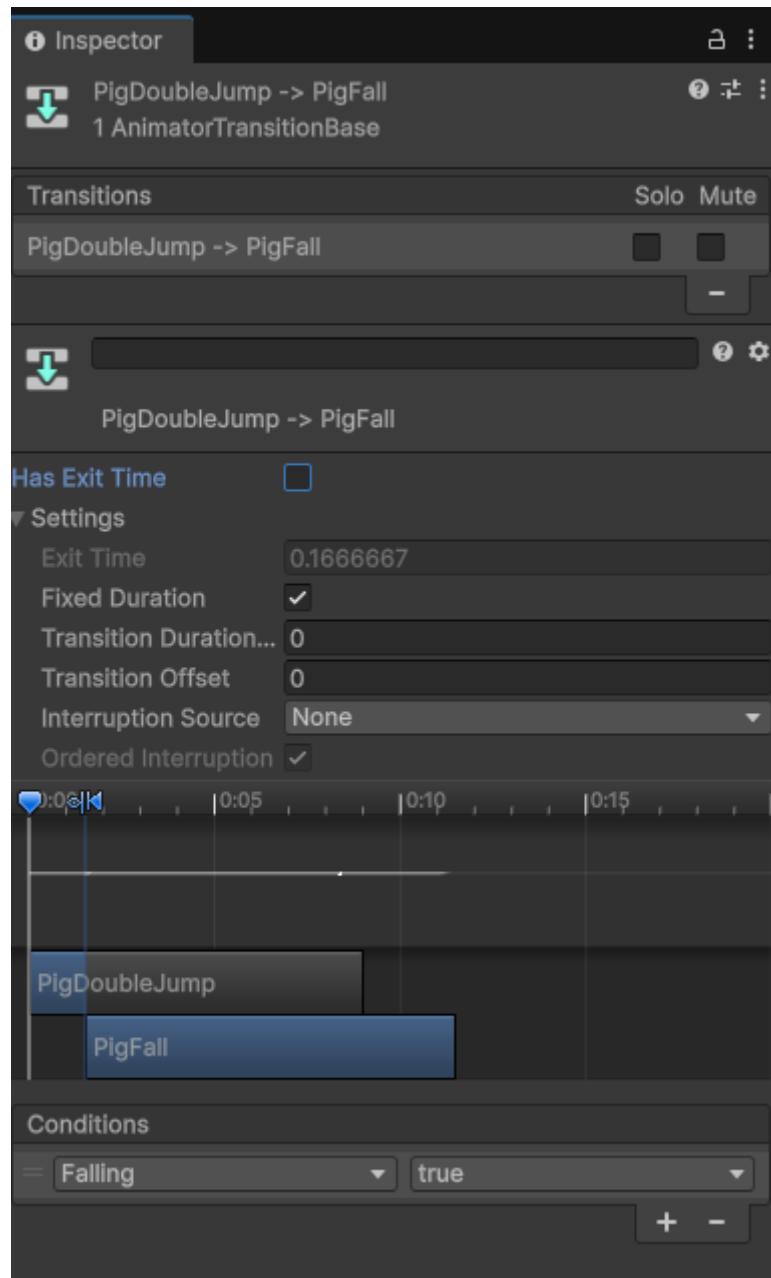
```
txtFRUITSWIN.gameObject.SetActive(true);
transition.SetActive(true);
Invoke("ChangeScene", 2f);
```

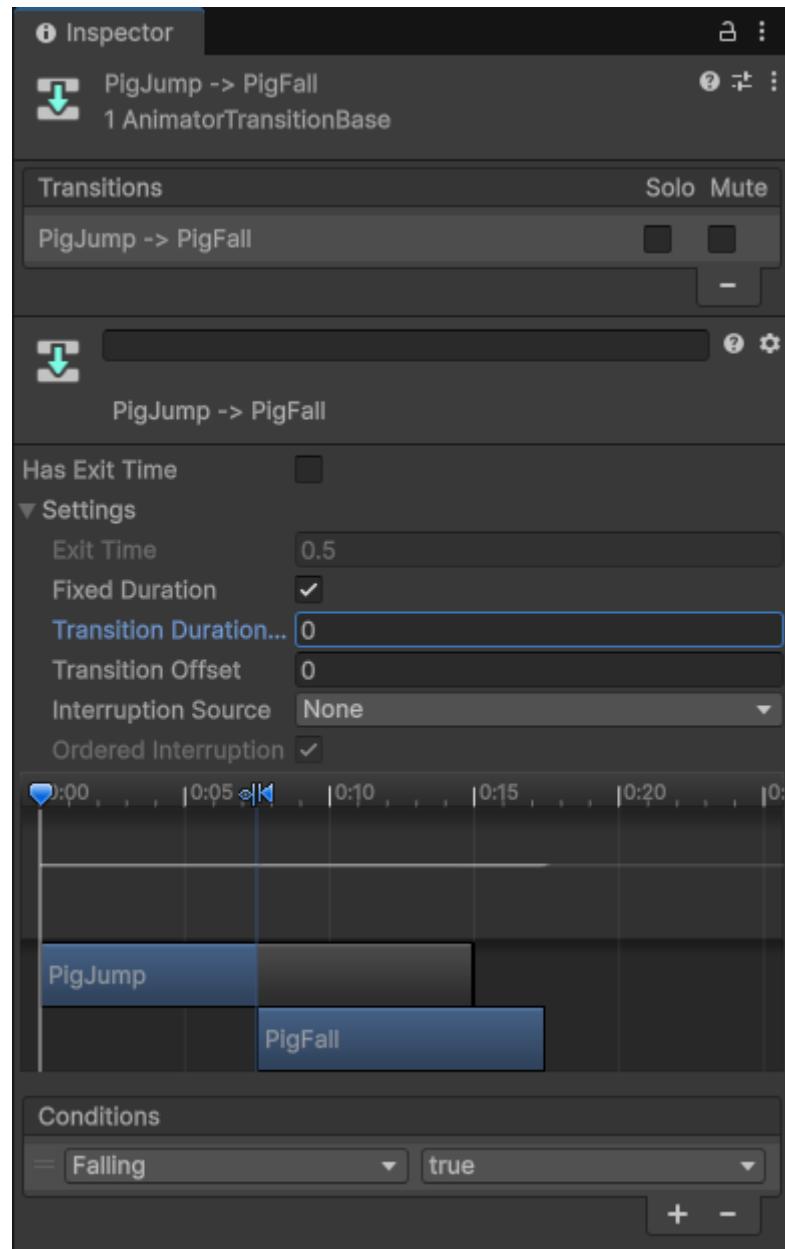
## Tutorial 9

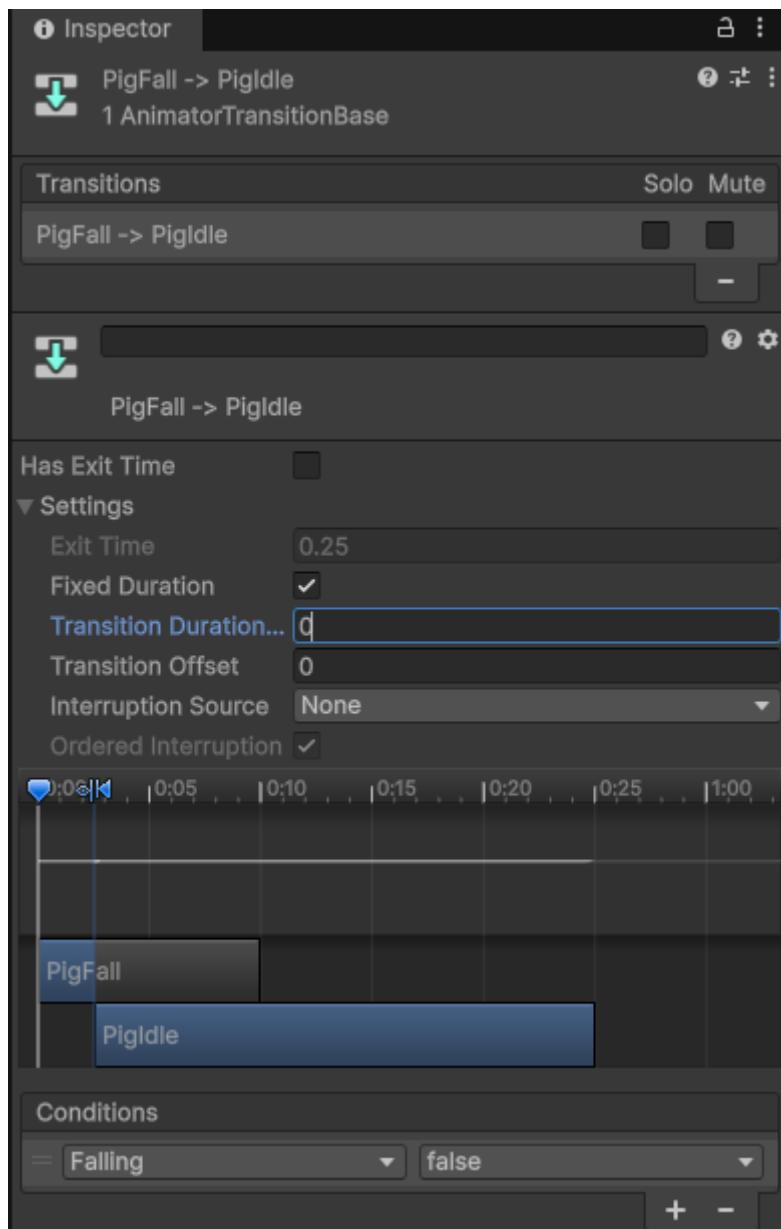


```
if (Input.GetKeyDown(KeyCode.Space))
{
    if (CheckGround.isGrounded)
    {
        canDoubleJump = true;
        rb.linearVelocity = new Vector2(rb.linearVelocityX, jumpForce);
    } else
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            if (canDoubleJump)
            [
                rb.linearVelocity = new Vector2(rb.linearVelocityX, doubleJumpForce);
                canDoubleJump = false;
            ]
        }
    }
}
```





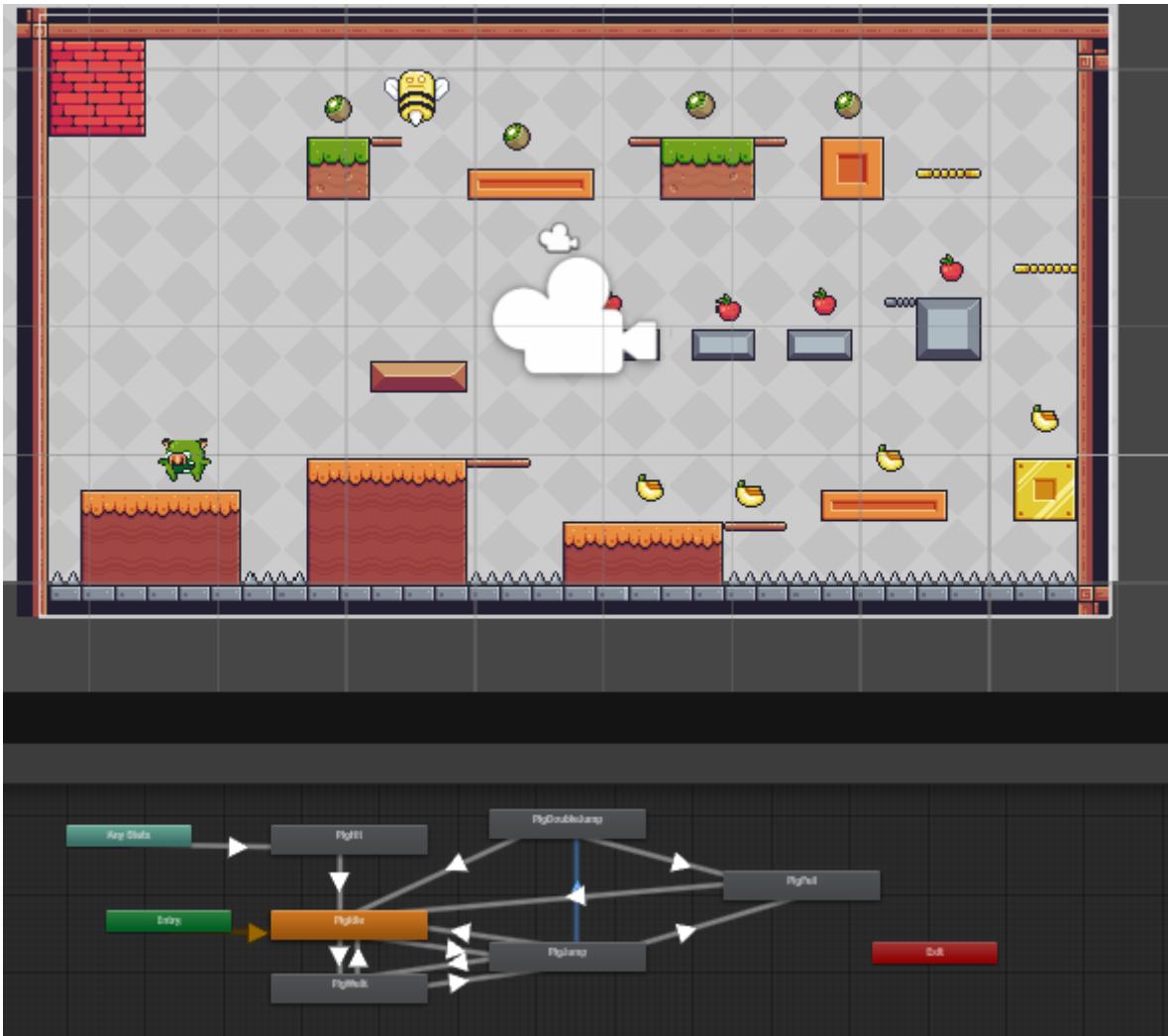




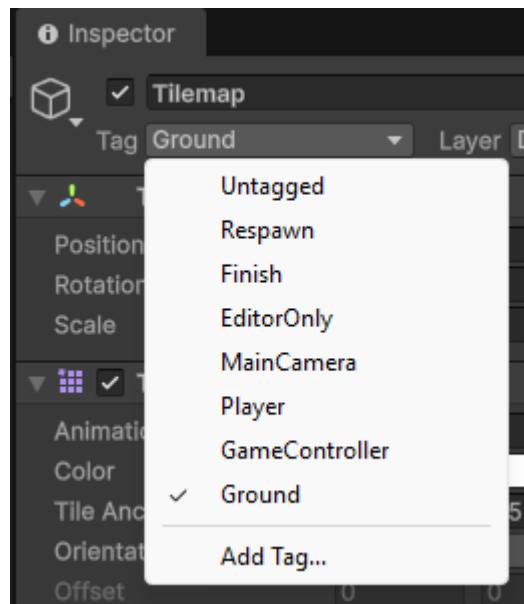
```
if (Input.GetKeyDown(KeyCode.Space))
{
    if (CheckGround.isGrounded)
    {
        canDoubleJump = true;
        rb.linearVelocity = new Vector2(rb.linearVelocityX, jumpForce);
    } else
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            if (canDoubleJump)
            {
                anim.SetBool("DoubleJump", true);
                rb.linearVelocity = new Vector2(rb.linearVelocityX, doubleJumpForce);
                canDoubleJump = false;
            }
        }
    }
}

if (CheckGround.isGrounded)
{
    anim.SetBool("Jump", false);
    anim.SetBool("DoubleJump", false);
    anim.SetBool("Falling", false);
}
else
{
    anim.SetBool("Jump", true);
    anim.SetBool("Run", true);
}

if (rb.linearVelocityY < 0)
{
    anim.SetBool("Falling", true);
} else if (rb.linearVelocityY > 0)
{
    anim.SetBool("Falling", false);
}
```



## Tutorial 10

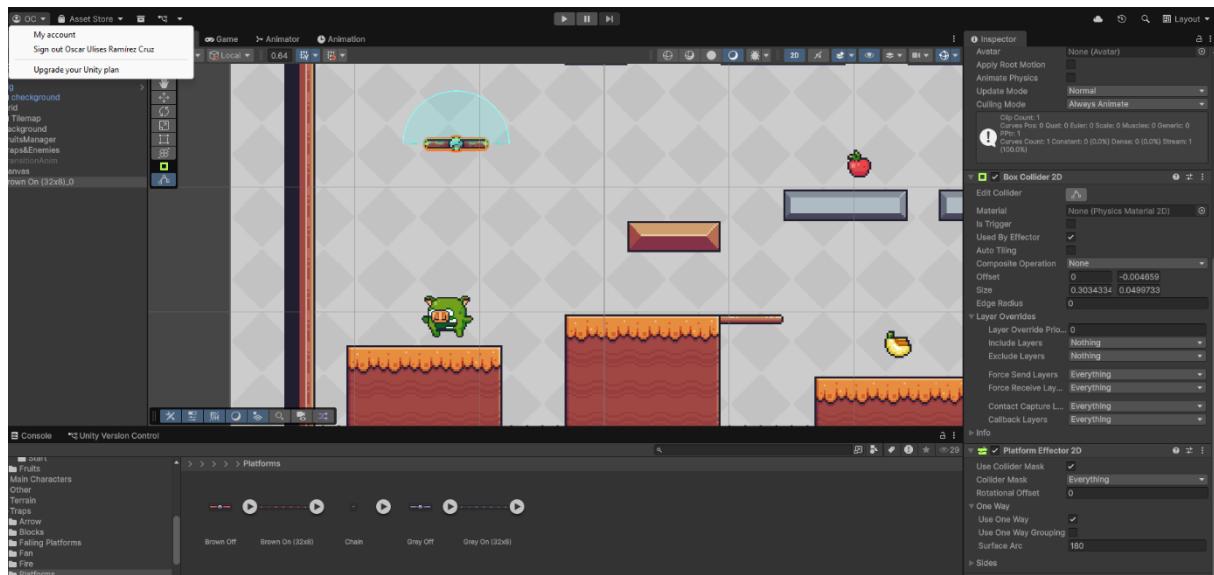


```
using UnityEngine;

2 références | Script Unity (2 références de ressources)
public class CheckGround : MonoBehaviour
{
    4 références | Champ Unity sérialisé
    public static bool isGrounded;

    0 références | Message Unity
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Ground"))
        {
            isGrounded = true;
        }
    }

    0 références | Message Unity
    private void OnTriggerExit2D(Collider2D collision)
    {
        if (collision.CompareTag("Ground"))
        {
            isGrounded = false;
        }
    }
}
```



Assets > Scripts > C# Platforms.cs > ...

```
2 // Oscar Ulises Ramirez Cruz
3
4 public class Platforms : MonoBehaviour
5 {
6     private PlatformEffector2D platEff;
7     public float waitTime = 0.5f;
8     private float waitedTime;
9
10    void Start()
11    {
12        platEff = GetComponent<PlatformEffector2D>();
13    }
14
15    void Update()
16    {
17        if (Input.GetKeyUp(KeyCode.S) || Input.GetKeyUp(KeyCode.DownArrow))
18        {
19            waitedTime = waitTime;
20        }
21
22        if (Input.GetKeyDown(KeyCode.S) || Input.GetKeyDown(KeyCode.DownArrow))
23        {
24            if (waitedTime <= 0)
25            {
26                platEff.rotationalOffset = 180f;
27                waitedTime = waitTime;
28                Invoke("ResetPlatform", 0.5f);
29            }
30            else
31            {
32                waitedTime -= Time.deltaTime;
33            }
34        }
35
36        if (Input.GetKeyDown(KeyCode.Space))
37        {
38            ResetPlatform();
39        }
40    }
41
42    private void ResetPlatform()
43    {
```

## Tutorial 11

★ Favorites

- All Materials
- All Models
- All Prefabs

Assets

- Animations
  - Animations
  - Background
  - Bee
- Characters
  - Bunny
  - Duck
  - Pig
- Flag

Character State Machine (Bunny)

```
graph LR; AnyState[Any State] --> BunnyHit[BunnyHit]; Entry[Entry] --> BunnyIdle[BunnyIdle]; BunnyHit --> BunnyIdle; BunnyIdle --> BunnyWalk[BunnyWalk]; BunnyIdle --> BunnyJump[BunnyJump]; BunnyIdle --> BunnyDoubleJump[BunnyDoubleJump]; BunnyJump --> BunnyWalk; BunnyJump --> BunnyFall[BunnyFall]; BunnyDoubleJump --> BunnyFall; BunnyFall --> Exit[Exit];
```

Character State Machine (Duck)

```
graph LR; AnyState[Any State] --> BunnyHit[BunnyHit]; Entry[Entry] --> BunnyIdle[BunnyIdle]; BunnyHit --> BunnyIdle; BunnyIdle --> BunnyWalk[BunnyWalk]; BunnyIdle --> BunnyJump[BunnyJump]; BunnyIdle --> BunnyDoubleJump[BunnyDoubleJump]; BunnyJump --> BunnyWalk; BunnyJump --> BunnyFall[BunnyFall]; BunnyDoubleJump --> BunnyFall; BunnyFall --> Exit[Exit];
```

Scripts

- CheckGround
- Checkpoint
- DamageObject
- EnemySpike
- FruitCollected
- FruitManager
- Platforms
- PlayerMove
- PlayerRespawn
- PlayerSelect



Assets > Scripts > C# PlayerSelect.cs > PlayerSelect > Start

```
1  using UnityEngine;
2
3  0 références | Script Unity (1 référence de ressource)
4  public class PlayerSelect : MonoBehaviour
5  {
6      4 références | 1 référence | 1 référence | 1 référence
7      1 référence | Champ Unity sérialisé
8      public enum Player { Bunny, Chamaleon, Pig }
9
10     3 références | Champ Unity sérialisé
11     public Player playerSelected;
12
13     3 références | Champ Unity sérialisé
14     public Animator anim;
15
16     3 références | Champ Unity sérialisé
17     public SpriteRenderer spriteRenderer;
18
19     0 références | Message Unity
20     void Start()
21     {
22         switch (playerSelected)
23         {
24             case Player.Bunny:
25                 anim.runtimeAnimatorController = playerControllers[0];
26                 spriteRenderer.sprite = playerSprites[0];
27                 break;
28             case Player.Chamaleon:
29                 anim.runtimeAnimatorController = playerControllers[1];
30                 spriteRenderer.sprite = playerSprites[1];
31                 break;
32             case Player.Pig:
33                 anim.runtimeAnimatorController = playerControllers[2];
34                 spriteRenderer.sprite = playerSprites[2];
35                 break;
36         }
37     }
38 }
```

## Tutorial 12

The image shows a Unity 2D game development interface. At the top, the title "Tutorial 12" is displayed. Below it, the "Assets > Scenes" panel shows a scene titled "MainMenu" with a 3D cube icon.

The main workspace displays a 2D level editor with a green diamond-tiled background. Three rectangular platforms are labeled "Nivel 1", "Nivel 2", and "Nivel 3". A central area is labeled "Seleccionar skin". A green frog-like character is positioned near Nivel 2. The bottom of the screen features a pink and orange layered floor.

The bottom half of the image shows the Unity Editor's interface:

- Inspector Panel (Left):** Shows a "Box Collider 2D" component with settings:
  - Material: None (Physics Material 2D)
  - Is Trigger: Checked
  - Used By Effector: Unchecked
  - Auto Tiling: Unchecked
  - Composite Operation: None
  - Offset: X: 0, Y: 0
  - Size: X: 0.64, Y: 0.64
  - Edge Radius: 0
- Hierarchy Panel (Right):** Shows the project structure:
  - MainMenu\*
  - Grid
  - Background
  - Portals
  - Main Camera
  - Canvas
  - EventSystem
  - Player
  - Platforms
  - Doors
  - Nivel1
  - Nivel2
  - Nivel3
  - SkinSelector



Assets > Scripts > C# OpenDoors.cs > OpenDoors > Update

```
1  using UnityEngine;
2  using TMPro;
3  using UnityEngine.SceneManagement;
4
5  // Oscar Ulises Ramirez Cruz
6
7  public class OpenDoors : MonoBehaviour
8  {
9      public TextMeshProUGUI portalText;
10     public string levelName;
11     public bool isInDoor = false;
12
13     private void OnTriggerEnter2D(Collider2D collision)
14     {
15         if (collision.CompareTag("Player"))
16         {
17             portalText.gameObject.SetActive(true);
18             portalText.text = "Presiona E para entrar a " + levelName;
19             isInDoor = true;
20         }
21     }
22
23     private void OnTriggerExit2D(Collider2D collision)
24     {
25         if (collision.CompareTag("Player"))
26         {
27             portalText.text = "";
28             isInDoor = false;
29         }
30     }
31
32     private void Update()
33     {
34         if (isInDoor && Input.GetKeyDown(KeyCode.E))
35         {
36             SceneManager.LoadScene(levelName);
37         }
38     }
39 }
40
```

