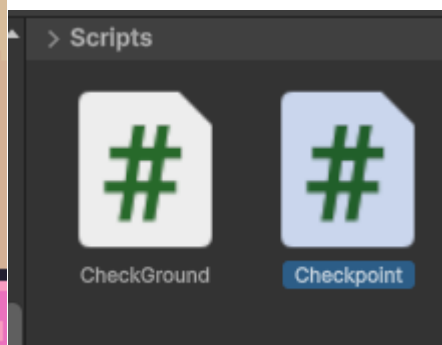
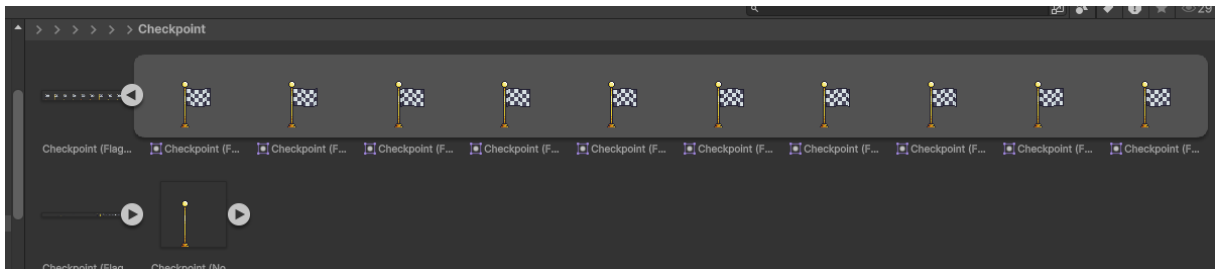


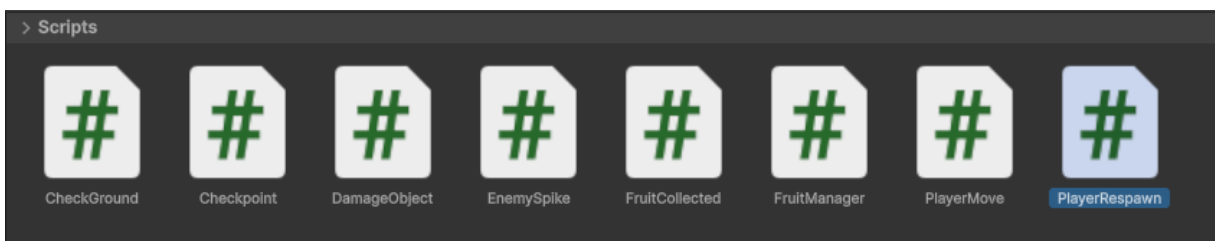
Tutoriales

Tutorial 7	2
Tutorial 8	5
Tutorial 9	7
Tutorial 10	13
Tutorial 11	16
Tutorial 12	19

Tutorial 7



```
Assets > Scripts > C# Checkpoint.cs > Checkpoint > OnTriggerEnter2D
1  using UnityEngine;
2
3  // Oscar Ulises Ramirez Cruz
4
5  0 références | Script Unity
6  public class Checkpoint : MonoBehaviour
7  {
8      0 références | Message Unity
9      private void OnTriggerEnter2D(Collider2D collision)
10     {
11         if (collision.CompareTag("Player"))
12         {
13             Debug.Log("Checkpoint reached!");
14         }
15     }
16 }
```



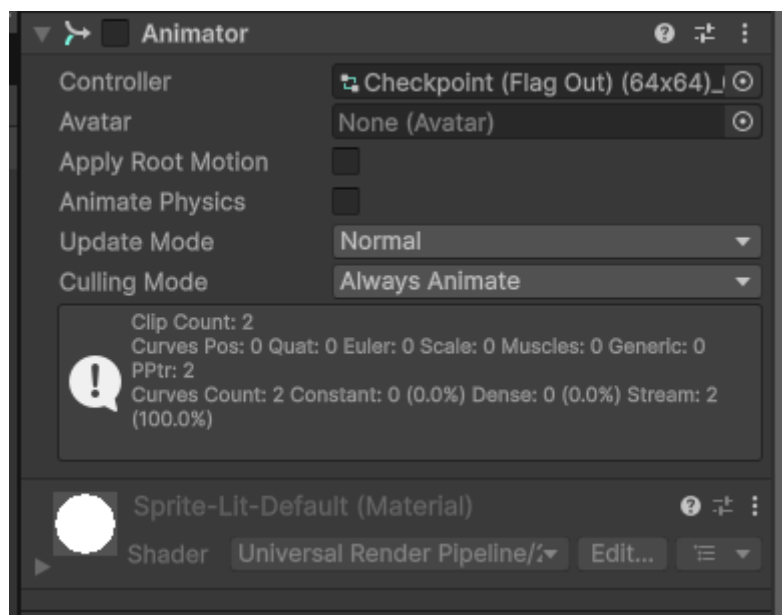
Assets > Scripts > C# PlayerRespawn.cs > PlayerRespawn > Start

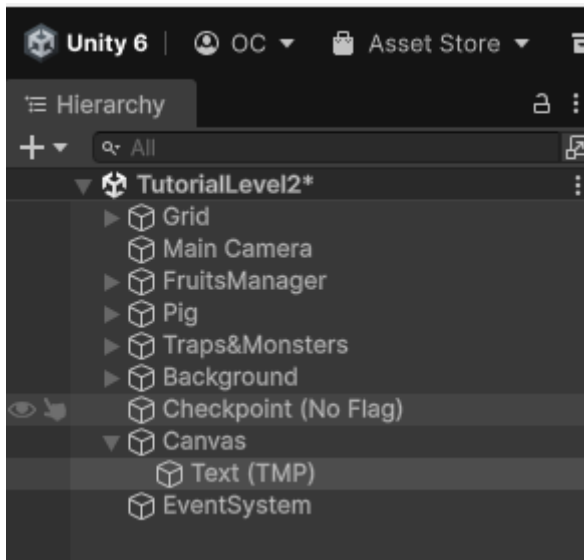
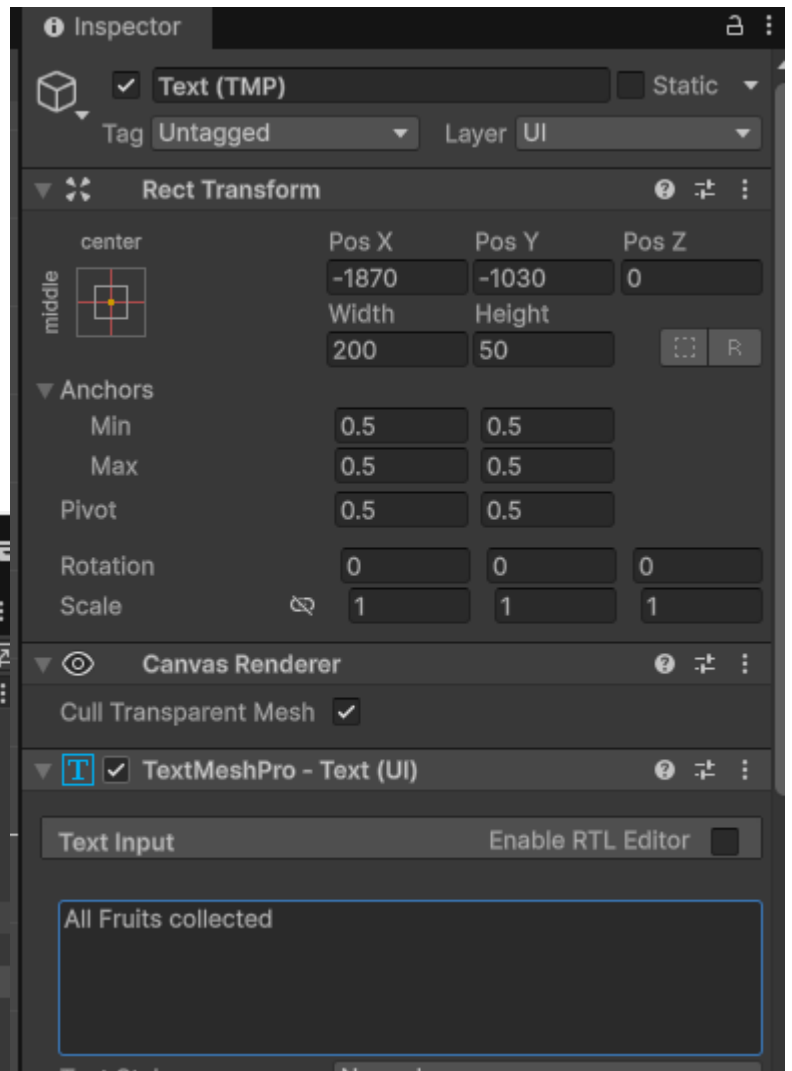
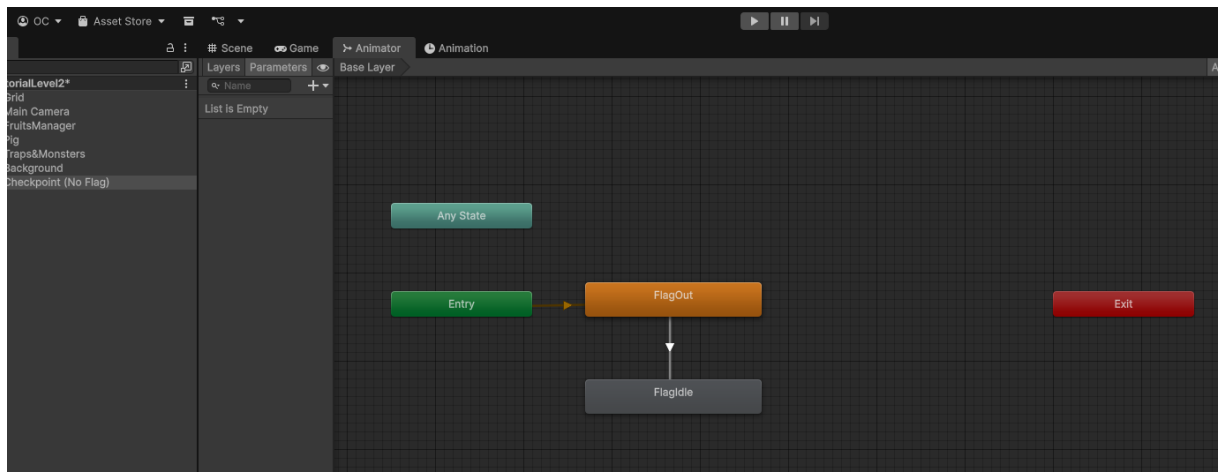
```
1 using UnityEngine;
2
3 // Oscar Ulises Ramirez Cruz
4
5 1 référence | Script Unity
6 public class PlayerRespawn : MonoBehaviour
7 {
8     2 références | 2 références
9     private float chPntX, chPntY;
10
11     0 références | Message Unity
12     void Start()
13     {
14         if (PlayerPrefs.GetFloat("chPntX") != 0 && PlayerPrefs.GetFloat("chPntY") != 0)
15         {
16             chPntX = PlayerPrefs.GetFloat("chPntX");
17             chPntY = PlayerPrefs.GetFloat("chPntY");
18
19             transform.position = new Vector2(chPntX, chPntY);
20         }
21
22     1 référence
23     public void ReachedCheckPoint(float x, float y)
24     {
25         PlayerPrefs.SetFloat("chPntX", transform.position.x);
26         PlayerPrefs.SetFloat("chPntY", transform.position.y);
27     }
28 }
```

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;
```





Tutorial 8

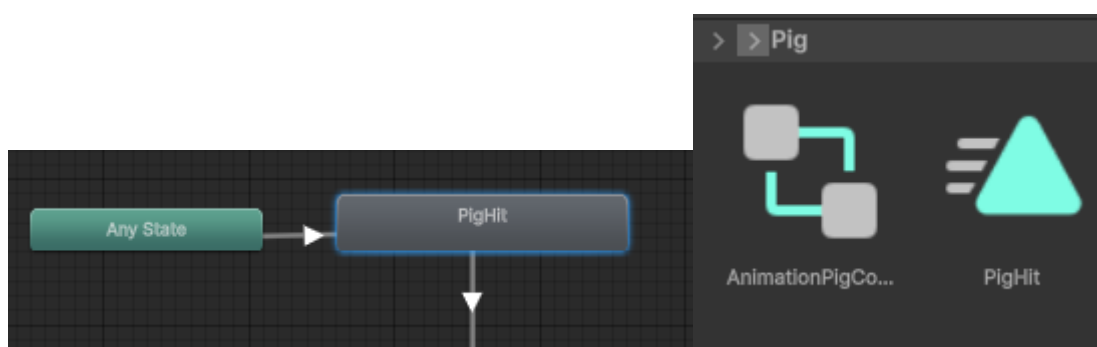
```
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;

// Oscar Ulises Ramirez Cruz

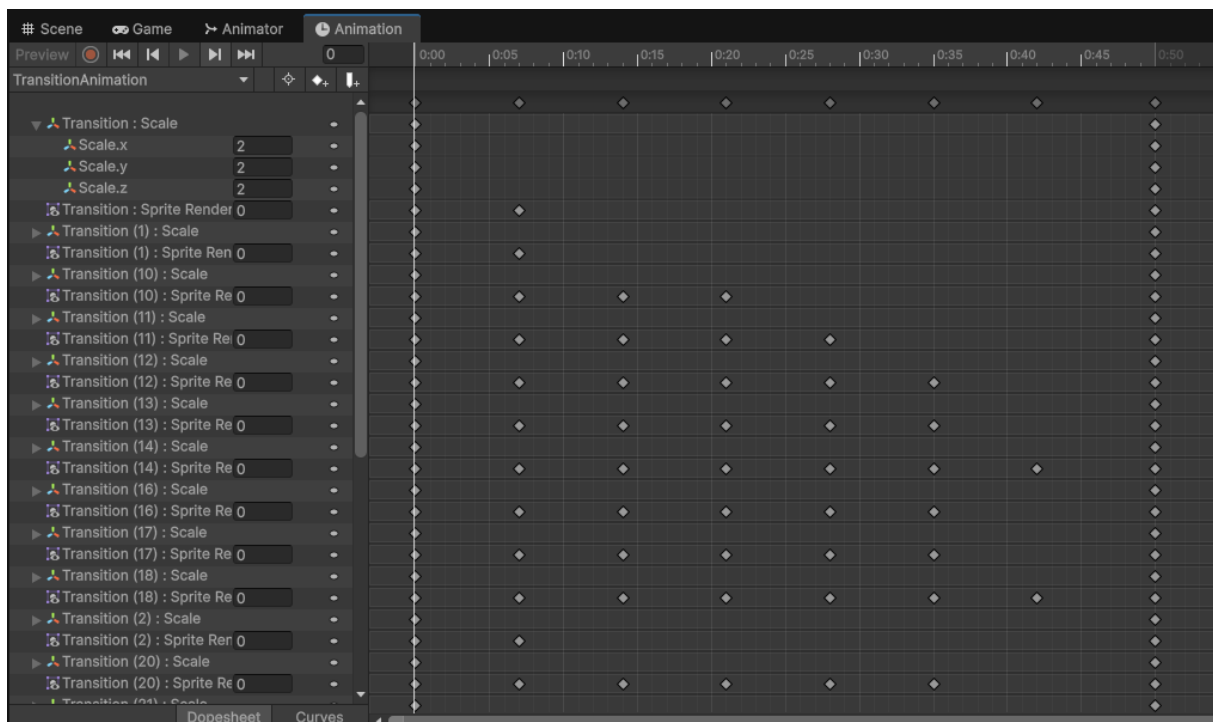
0 références | Script Unity (3 références de ressources)
public class FruitManager : MonoBehaviour
{
    1 référence | Champ Unity sérialisé
    public TMP_Text txtFruitsWin;

    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);
    }
}
```



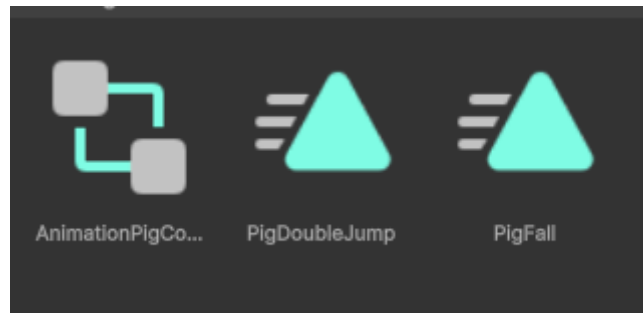
```
0 références
public void PlayerDamaged()
{
    anim.Play("PigHit");
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}
```



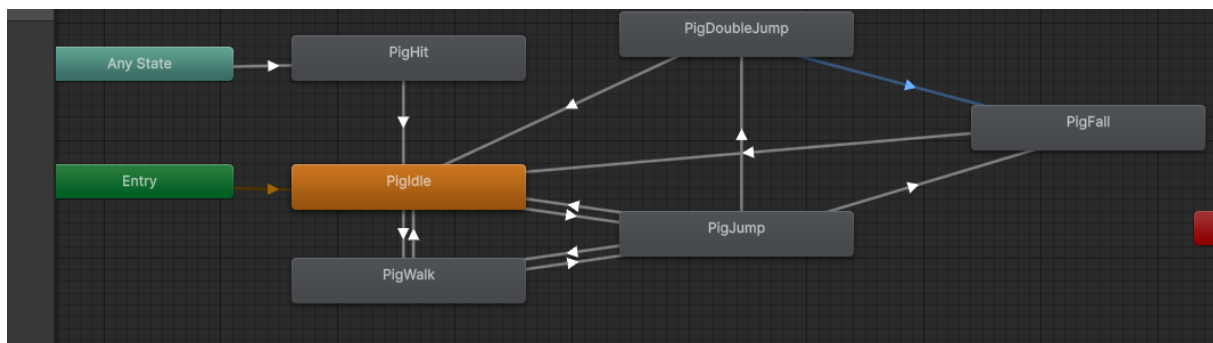
1 référence | Champ Unity sérialisé
 public GameObject transition;

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

Tutorial 9



```
if (Input.GetKey(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;
            }
        }
    }
}
```



Inspector

PigDoubleJump -> PigFall

1 AnimatorTransitionBase

Transitions

Solo Mute

PigDoubleJump -> PigFall

-

PigDoubleJump -> PigFall

Has Exit Time

Settings

Exit Time

0.1666667

Fixed Duration

Transition Duration...

0

Transition Offset

0

Interruption Source

None

Ordered Interruption

0:00

0:05

0:10

0:15

PigDoubleJump

PigFall

Conditions

=

Falling

true

+

-

Inspector

PigJump -> PigFall

1 AnimatorTransitionBase

Transitions

Solo Mute

PigJump -> PigFall

-

PigJump -> PigFall

Has Exit Time

Settings

Exit Time

0.5

Fixed Duration

☒

Transition Duration...

0

Transition Offset

0

Interruption Source

None

Ordered Interruption

☒

0:00

0:05

0:10

0:15

0:20

0:25

PigJump

PigFall

Conditions

=

Falling

true

+

-

Inspector

PigFall -> PigIdle

1 AnimatorTransitionBase

Solo

Mute

PigFall -> PigIdle

-

PigFall -> PigIdle

Has Exit Time

☐

▼ Settings

Exit Time

0.25

Fixed Duration

☒

Transition Duration...

d

Transition Offset

0

Interruption Source

None

Ordered Interruption

☒

0:00

0:05

0:10

0:15

0:20

0:25

1:00

PigFall

PigIdle

Conditions

=

Falling

false

+

-

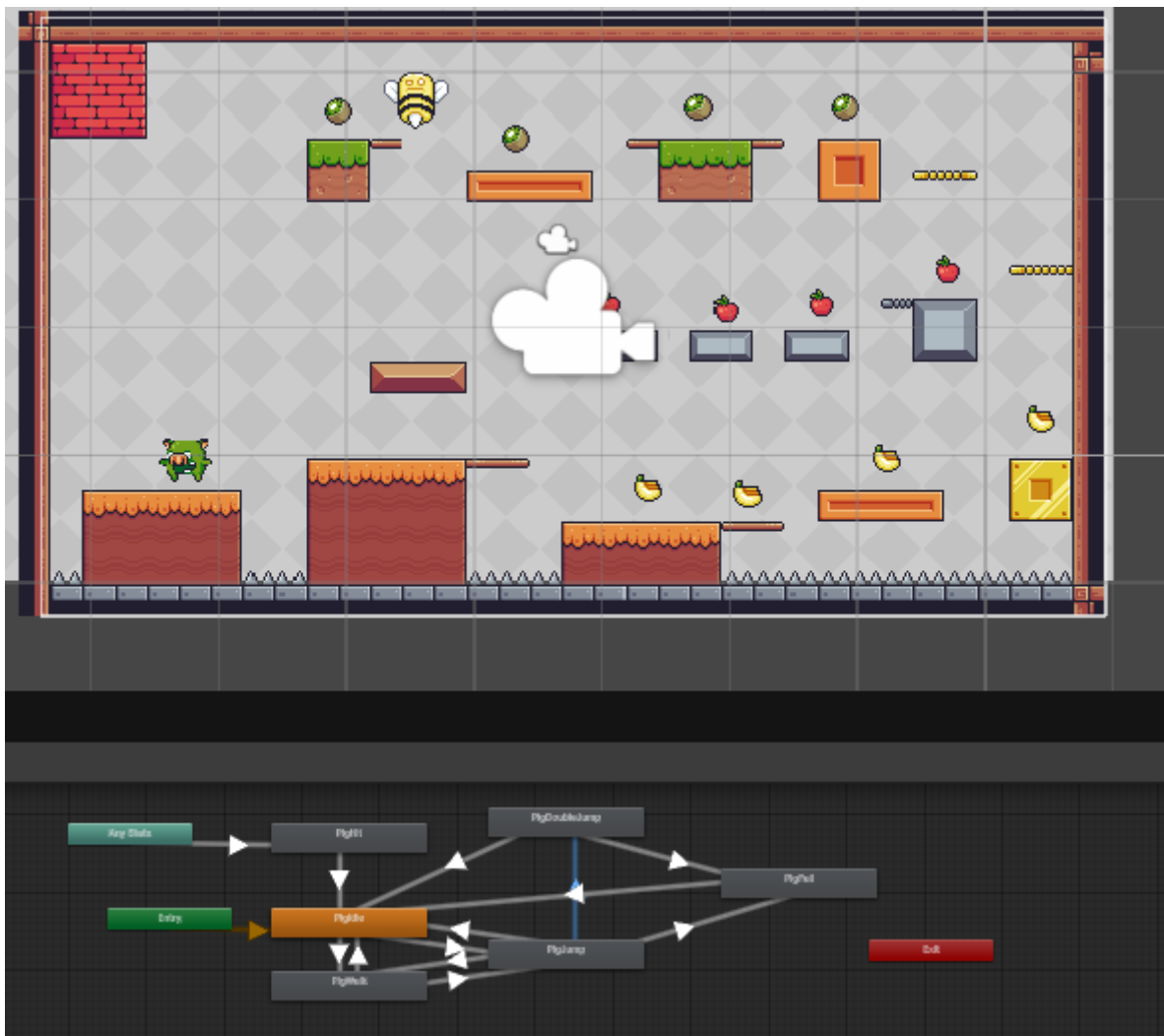
```

if (Input.GetKey(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;
            }
        }
    }
}

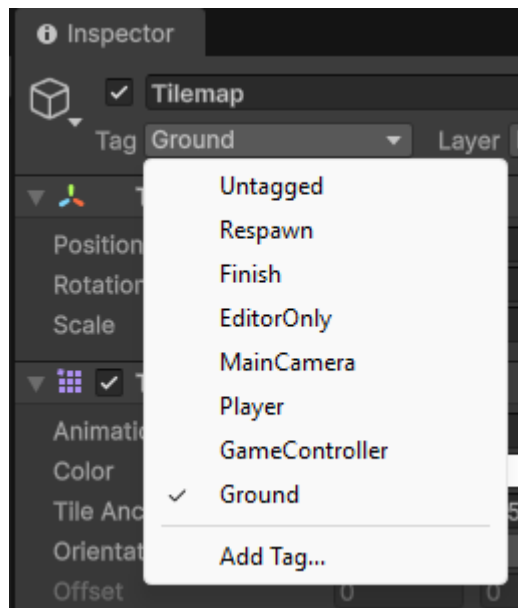
if (CheckGround.isGrounded)
{
    anim.SetBool("Jump", false);
    anim.SetBool("DoubleJump", false);
    anim.SetBool("Falling", false);
}
else
{
    anim.SetBool("Jump", true);
    anim.SetBool("Run", true);
}

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

```



Tutorial 10

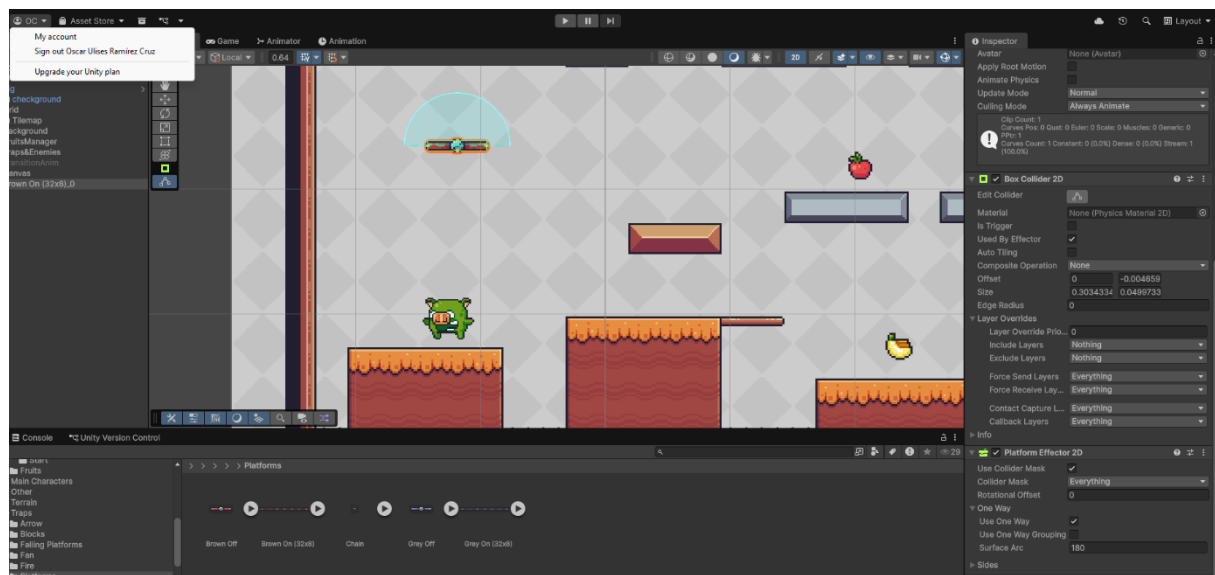


```
using UnityEngine;

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;
        }
    }

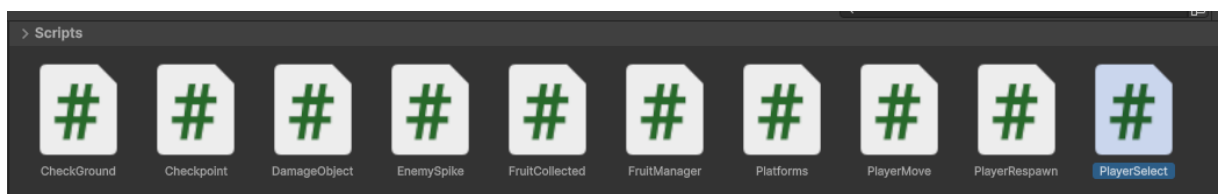
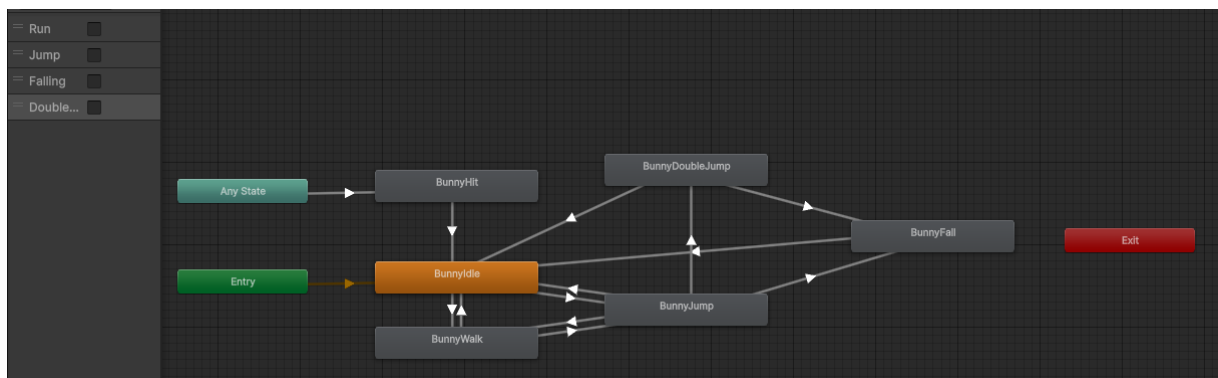
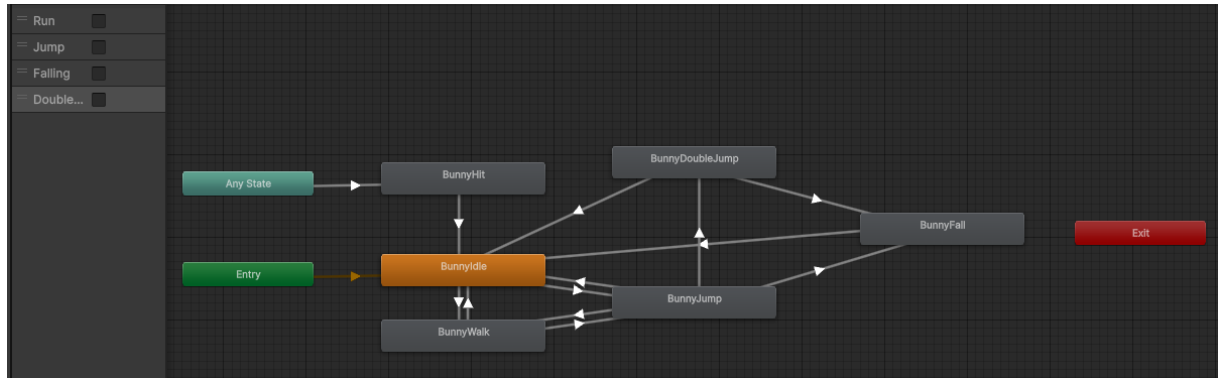
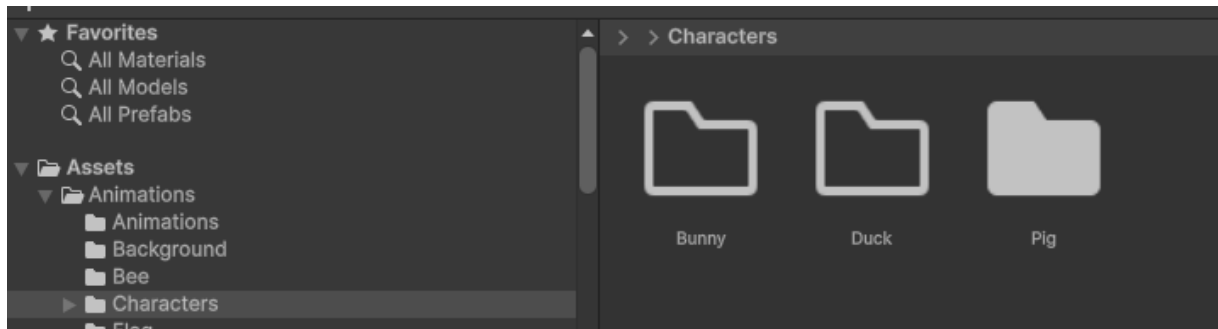
    0 références | Message Unity
    private void OnTriggerExit2D(Collider2D collision)
    {
        if (collision.CompareTag("Ground"))
        {
            isGrounded = false;
        }
    }
}
```



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

```
2 // Oscar Ulises Ramirez Cruz
3
4 0 références | Script Unity
5 public class Platforms : MonoBehaviour
6 {
7     3 références
8     private PlatformEffector2D platEff;
9     2 références | Champ Unity sérialisé
10    public float waitTime = 0.5f;
11    4 références
12    private float waitedTime;
13
14    0 références | Message Unity
15    void Start()
16    {
17        platEff = GetComponent<PlatformEffector2D>();
18    }
19
20    0 références | Message Unity
21    void Update()
22    {
23        if (Input.GetKeyUp(KeyCode.S) || Input.GetKeyUp(KeyCode.DownArrow))
24        {
25            waitedTime = waitTime;
26        }
27
28        if (Input.GetKey(KeyCode.S) || Input.GetKey(KeyCode.DownArrow))
29        {
30            if (waitedTime <= 0)
31            {
32                platEff.rotationalOffset = 180f;
33                waitedTime = waitTime;
34                Invoke("ResetPlatform", 0.5f);
35            }
36            else
37            {
38                waitedTime -= Time.deltaTime;
39            }
40        }
41
42        if (Input.GetKeyDown(KeyCode.Space))
43        {
44            ResetPlatform();
45        }
46    }
47
48    1 référence
49    private void ResetPlatform()
50    {
51    }
```

Tutorial 11

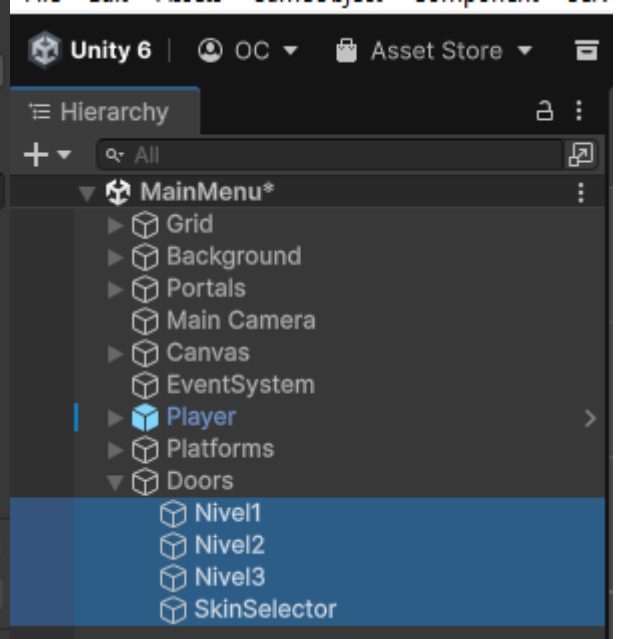
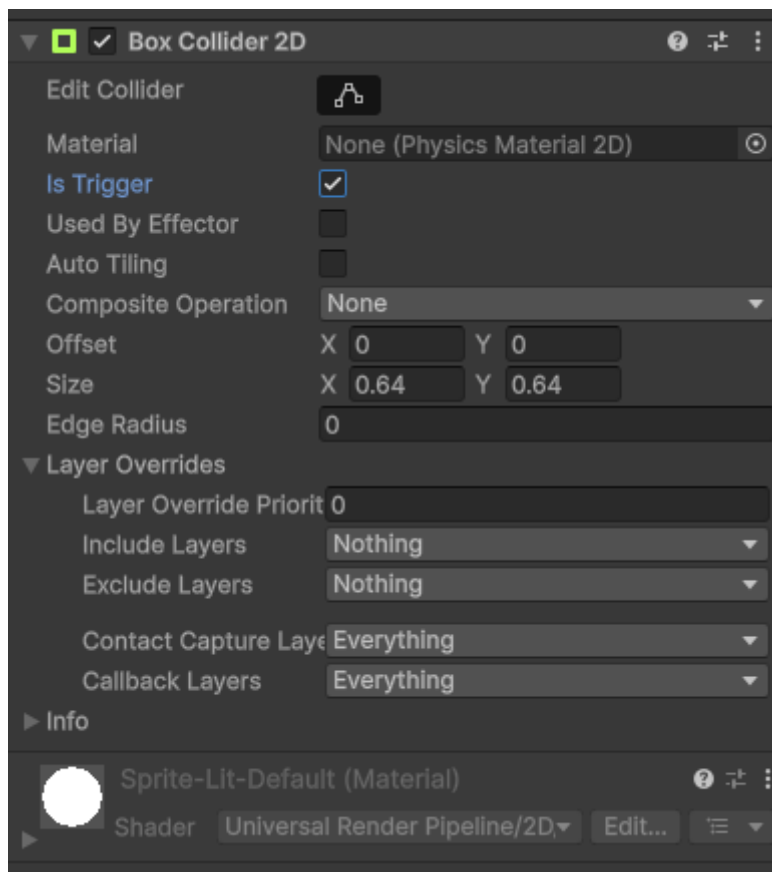
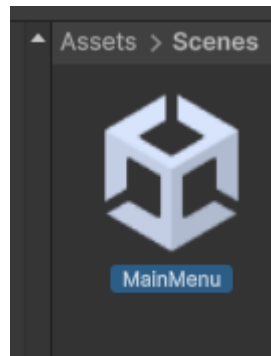




Assets > Scripts > C# PlayerSelect.cs > PlayerSelect > Start

```
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      public enum Player { Bunny, Chamaleon, Pig }
8      1 référence | Champ Unity sérialisé
9      public Player playerSelected;
10     3 références | Champ Unity sérialisé
11     public Animator anim;
12     3 références | Champ Unity sérialisé
13     public SpriteRenderer spriteRenderer;
14
15     3 références | Champ Unity sérialisé
16     public RuntimeAnimatorController[] playerControllers;
17     3 références | Champ Unity sérialisé
18     public Sprite[] playerSprites;
19
20     0 références | Message Unity
21     void Start()
22     {
23         switch (playerSelected)
24         {
25             case Player.Bunny:
26                 anim.runtimeAnimatorController = playerControllers[0];
27                 spriteRenderer.sprite = playerSprites[0];
28                 break;
29             case Player.Chamaleon:
30                 anim.runtimeAnimatorController = playerControllers[1];
31                 spriteRenderer.sprite = playerSprites[1];
32                 break;
33             case Player.Pig:
34                 anim.runtimeAnimatorController = playerControllers[2];
35                 spriteRenderer.sprite = playerSprites[2];
36                 break;
37         }
38     }
39 }
```

Tutorial 12





```
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  0 références
8  public class OpenDoors : MonoBehaviour
9  {
10     3 références
11     public TextMeshProUGUI portalText;
12     2 références
13     public string levelName;
14     3 références
15     public bool isInDoor = false;
16
17     0 références
18     private void OnTriggerEnter2D(Collider2D collision)
19     {
20         if (collision.CompareTag("Player"))
21         {
22             portalText.gameObject.SetActive(true);
23             portalText.text = "Presiona E para entrar a " + levelName;
24             isInDoor = true;
25         }
26     }
27
28     0 références
29     private void OnTriggerExit2D(Collider2D collision)
30     {
31         if (collision.CompareTag("Player"))
32         {
33             portalText.text = "";
34             isInDoor = false;
35         }
36     }
37
38     0 références
39     private void Update()
40     {
41         if (isInDoor && Input.GetKeyDown(KeyCode.E))
42         {
43             SceneManager.LoadScene(levelName);
44         }
45     }
46 }
```

