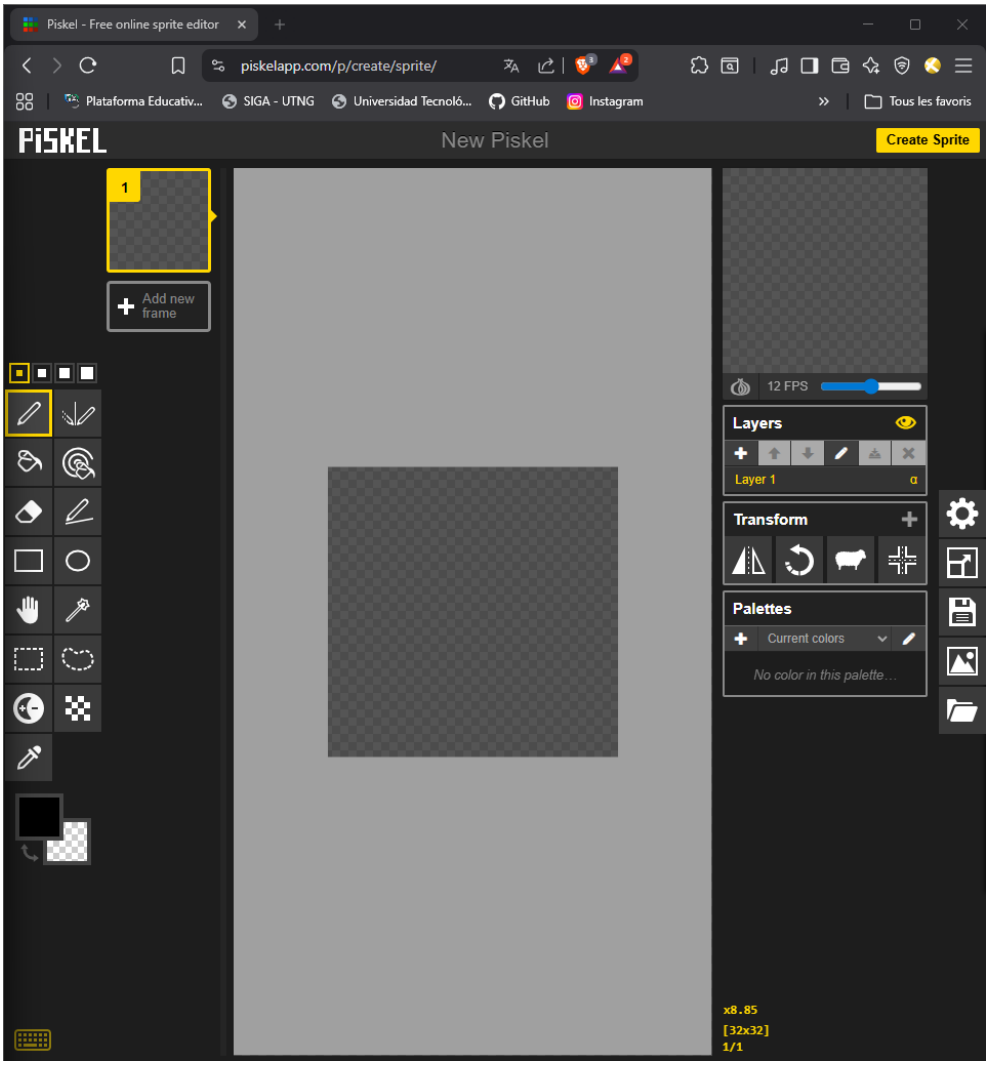
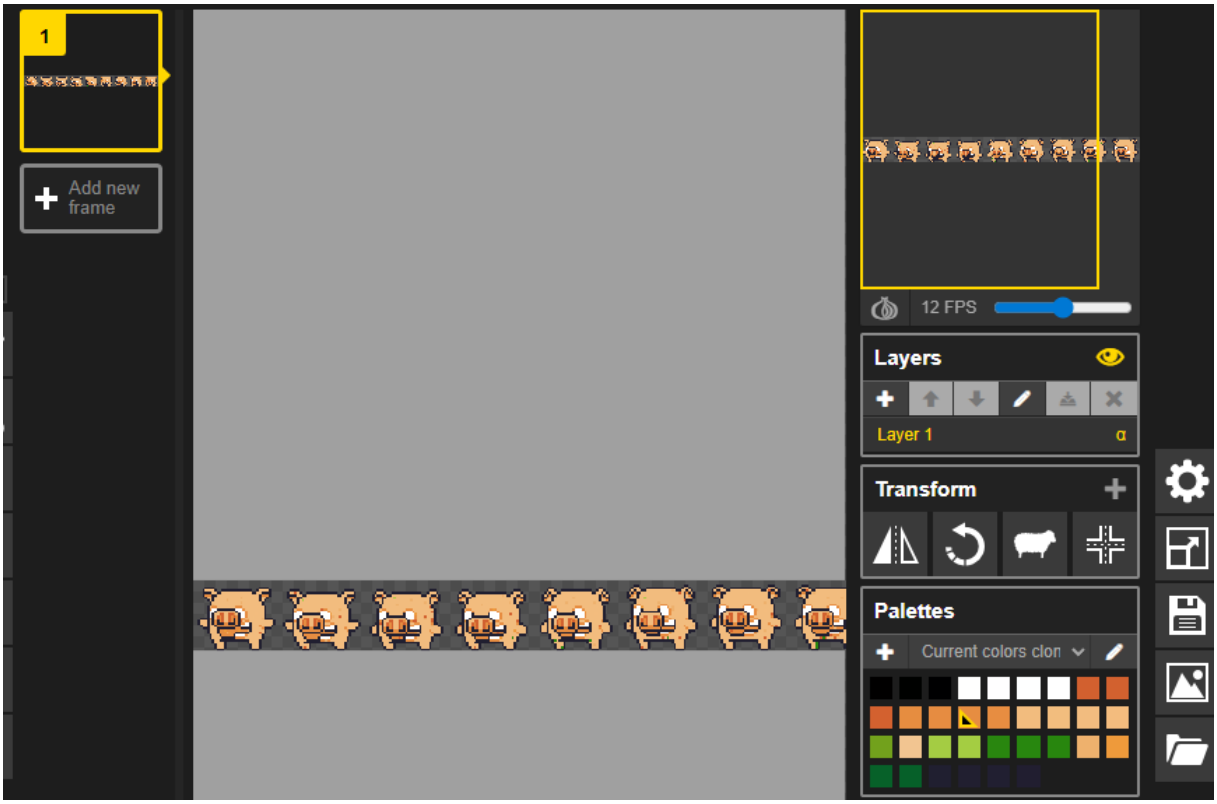
Tutoriales

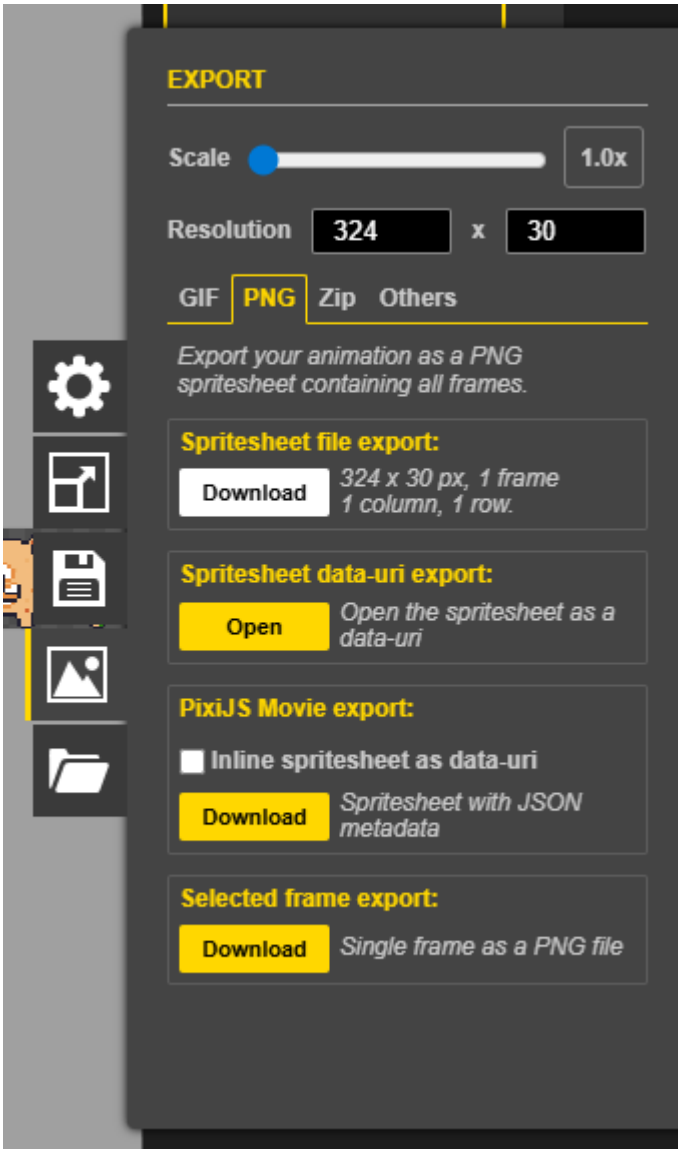
Tutorial 25

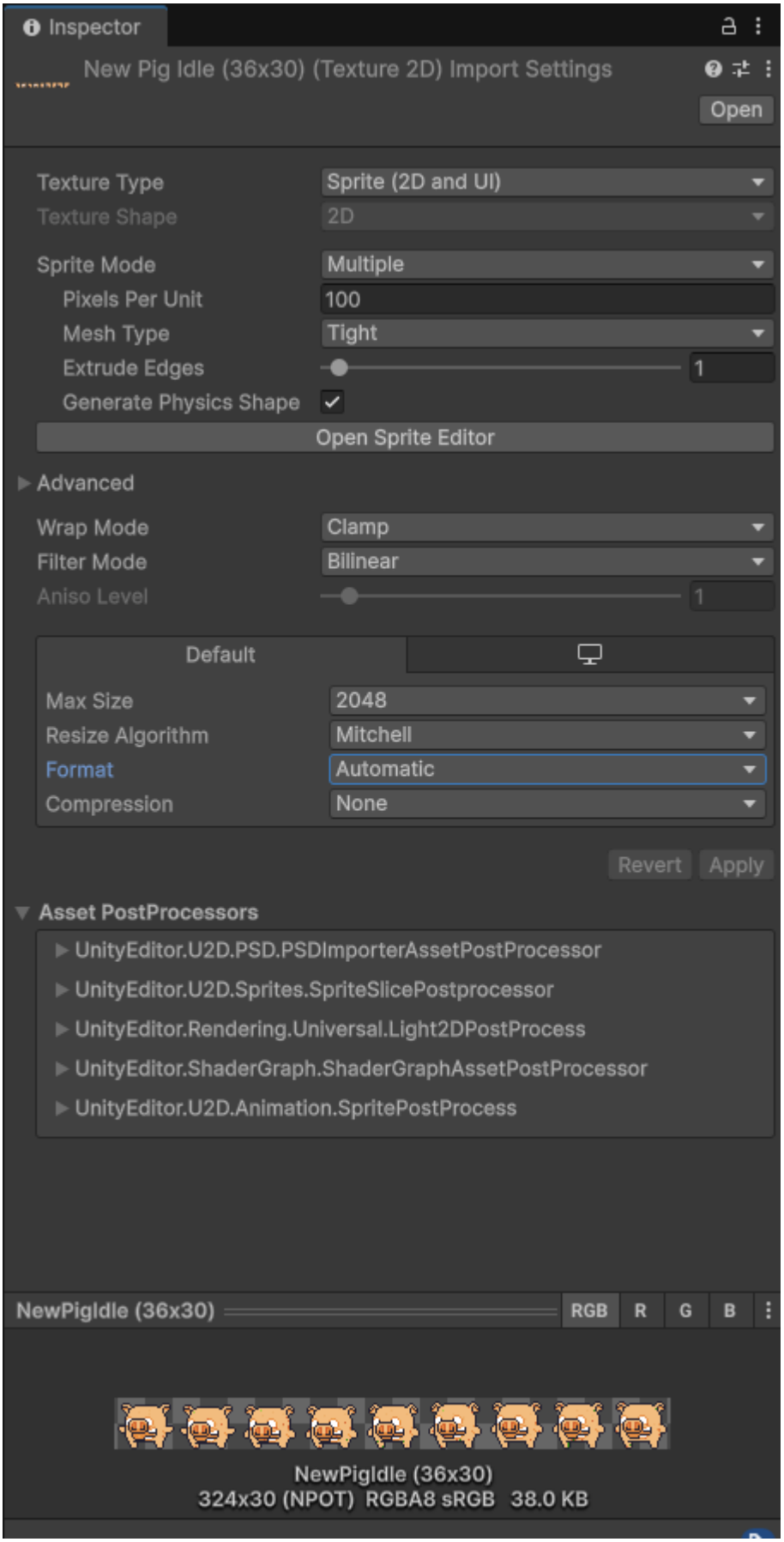

Piskel - Free online sprite editor
piskelapp.com/p/create/sprite/
Plataforma Educativ...
SIGA - UTNG
Universidad Tecnoló...
GitHub
Instagram
Tous les favoris
PiSKEL
New Piskel
Create Sprite
1
Add new frame
12 FPS
Layers
Layer 1
Transform
Palettes
Current colors
No color in this palette...
x8.85
[32x32]
1/1


PiSKEL
Idle (36x30) *
Create Sprite
1
Add new frame
12 FPS
Layers
Layer 1
Transform
Palettes
Current colors

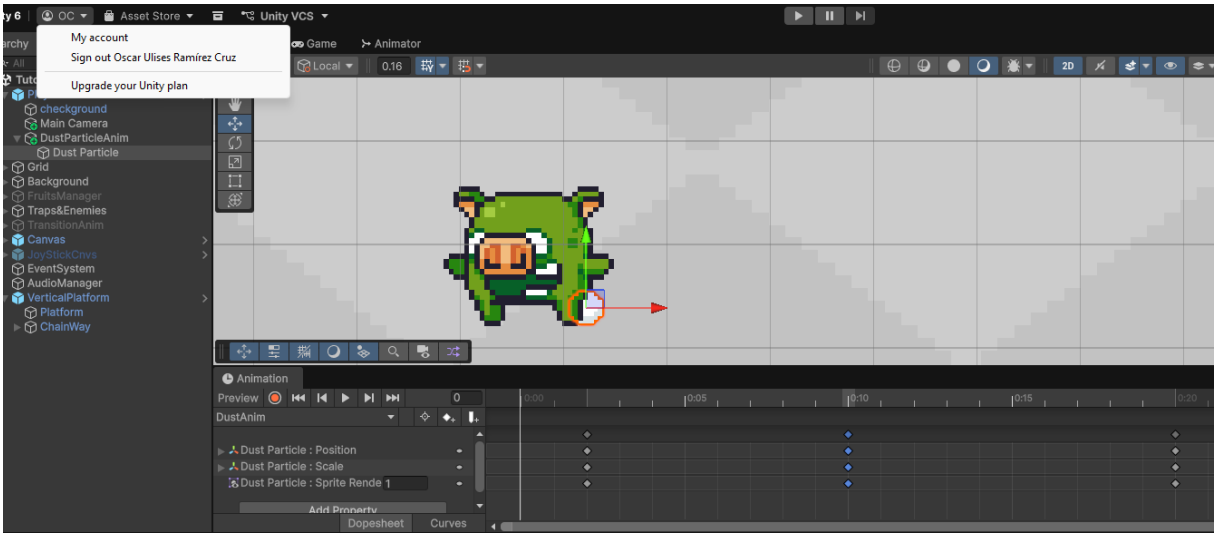

Create Palette
X
Name
Current colors clone
Import from file
#010000
H
0
S
100
V
0
R
1
G
0
B
0


1
Add new frame
12 FPS
Layers
Layer 1
α
Transform
Palettes
Current colors clon


EXPORT
Scale
1.0x
Resolution
324
x
30
GIF
PNG
Zip
Others
Export your animation as a PNG spritesheet containing all frames.
Spritesheet file export:
Download
324 x 30 px, 1 frame
1 column, 1 row.
Spritesheet data-uri export:
Open
Open the spritesheet as a data-uri
PixiJS Movie export:
Inline spritesheet as data-uri
Download
Spritesheet with JSON metadata
Selected frame export:
Download
Single frame as a PNG file


Inspector
New Pig Idle (36x30) (Texture 2D) Import Settings
Open
Texture Type
Sprite (2D and UI)
Texture Shape
2D
Sprite Mode
Multiple
Pixels Per Unit
100
Mesh Type
Tight
Extrude Edges
1
Generate Physics Shape
Open Sprite Editor
Advanced
Wrap Mode
Clamp
Filter Mode
Bilinear
Aniso Level
1
Default
Max Size
2048
Resize Algorithm
Mitchell
Format
Automatic
Compression
None
Revert
Apply
Asset PostProcessors
UnityEditor.U2D.PSD.PSDImporterAssetPostProcessor
UnityEditor.U2D.Sprites.SpriteSlicePostprocessor
UnityEditor.Rendering.Universal.Light2DPostProcess
UnityEditor.ShaderGraph.ShaderGraphAssetPostProcessor
UnityEditor.U2D.Animation.SpritePostProcess
NewPigIdle (36x30)
RGB
R
G
B
NewPigIdle (36x30)
324x30 (NPOT) RGBA8 sRGB 38.0 KB

Tutorial 26



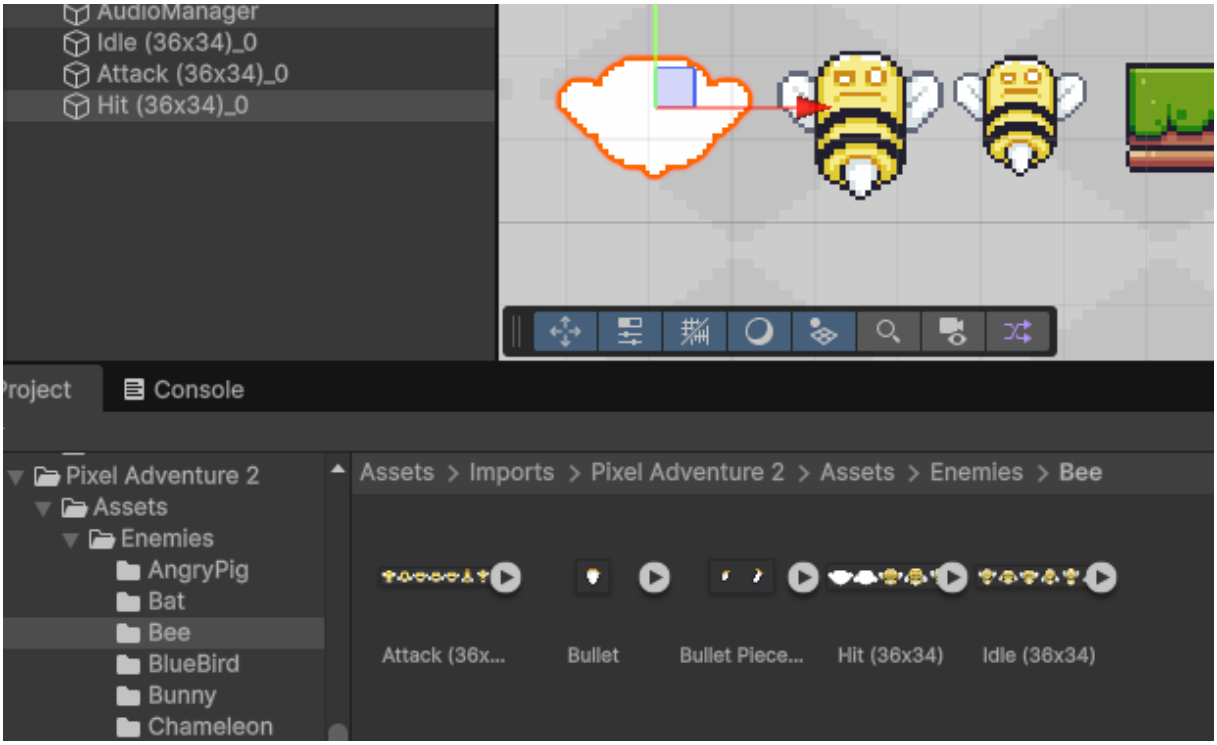

```
0 références | Message Unity
void Update()
{
    float moveInput = Input.GetAxis("Horizontal");
    rb.linearVelocity = new Vector2(moveInput * moveSpeed, rb.linearVelocityY);

    if (moveInput > 0) {
        //sprRnd.flipX = true;
        transform.localScale = new Vector3(-0.8f, 0.8f, 1);
        anim.SetBool("Run", true);
    }
    else if (moveInput < 0)
    {
        //sprRnd.flipX = false;
        transform.localScale = new Vector3(0.8f, 0.8f, 1);
        anim.SetBool("Run", true);
    }
    else
    {
        anim.SetBool("Run", false);
    }
```

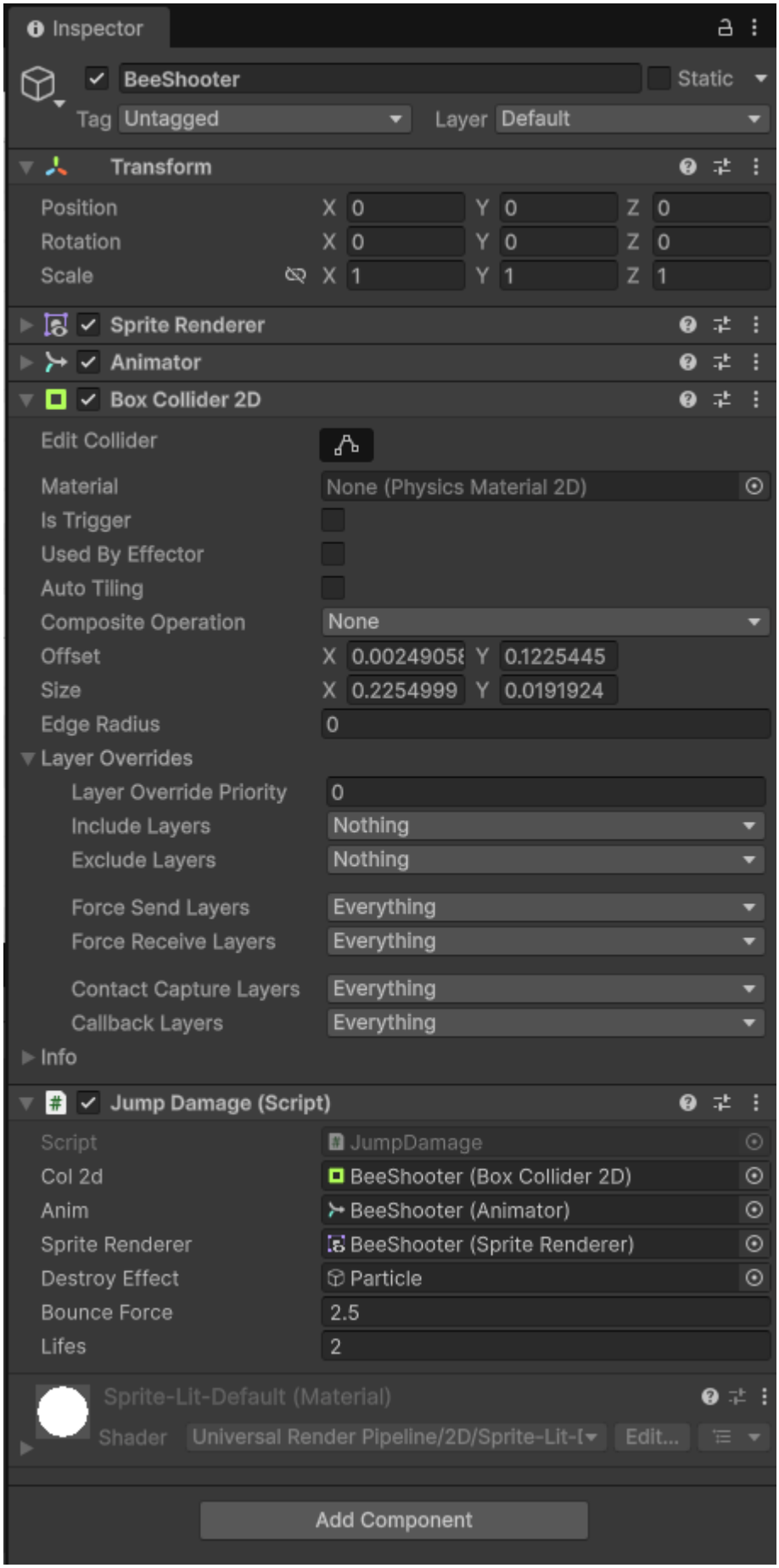
```
if (moveInput > 0) {
    //sprRnd.flipX = true;
    transform.localScale = new Vector3(-0.8f, 0.8f, 1);
    anim.SetBool("Run", true);
    if (CheckGround.isGrounded) dustAnimation.SetActive(true);
}
else if (moveInput < 0)
{
    //sprRnd.flipX = false;
    transform.localScale = new Vector3(0.8f, 0.8f, 1);
    anim.SetBool("Run", true);
    if (CheckGround.isGrounded) dustAnimation.SetActive(true);
}
else
{
    anim.SetBool("Run", false);
    dustAnimation.SetActive(false);
}
```

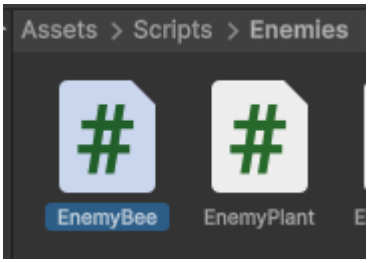
Assets > Scripts > Player > PlayerJoystick.cs > PlayerJoystick > dustAnimation

```
4   public class PlayerJoystick : MonoBehaviour
18      public SpriteRenderer sprRnd;
        11 références | Champ Unity sérialisé
19      public Animator anim;
20
        3 références | Champ Unity sérialisé
21      public GameObject dustAnimation;
22
        0 références | Message Unity
23      void Start()
24      {
25          rb = GetComponent<Rigidbody2D>();
26      }
27
        0 références | Message Unity
28      void Update()
29      {
30          horizontalmove = joystick.Horizontal * moveSpeedHorizontal;
31          transform.position += new Vector3(horizontalmove, 0, 0) * Time.deltaTime * moveSpeed;
32
33          float moveInput = Input.GetAxis("Horizontal");
34          //rb.linearVelocity = new Vector2(moveInput * moveSpeed, rb.linearVelocityY);
35
36          if (horizontalmove > 0) {
37              transform.localScale = new Vector3(-0.8f, 0.8f, 1);
38              anim.SetBool("Run", true);
39              if (CheckGround.isGrounded) dustAnimation.SetActive(true);
40          }
41          else if (horizontalmove < 0)
42          {
43              //sprRnd.flipX = false;
44              transform.localScale = new Vector3(0.8f, 0.8f, 1);
45              anim.SetBool("Run", true);
46              if (CheckGround.isGrounded) dustAnimation.SetActive(true);
47          }
48          else
49          {
50              anim.SetBool("Run", false);
51              dustAnimation.SetActive(false);
52          }
53
```

Tutorial 27

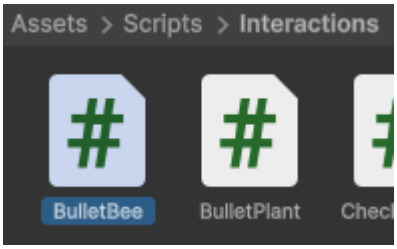

AudioManager
Idle (36x34)_0
Attack (36x34)_0
Hit (36x34)_0
Project
Console
Pixel Adventure 2
Assets
Enemies
AngryPig
Bat
Bee
BlueBird
Bunny
Chameleon
Assets > Imports > Pixel Adventure 2 > Assets > Enemies > Bee
Attack (36x...
Bullet
Bullet Piece...
Hit (36x34)
Idle (36x34)


Inspector
BeeShooter
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Sprite Renderer
Animator
Box Collider 2D
Edit Collider
Material None (Physics Material 2D)
Is Trigger
Used By Effector
Auto Tiling
Composite Operation None
Offset X 0.0024905 Y 0.1225445
Size X 0.2254999 Y 0.0191924
Edge Radius 0
Layer Overrides
Layer Override Priority 0
Include Layers Nothing
Exclude Layers Nothing
Force Send Layers Everything
Force Receive Layers Everything
Contact Capture Layers Everything
Callback Layers Everything
Info
Jump Damage (Script)
Script JumpDamage
Col 2d BeeShooter (Box Collider 2D)
Anim BeeShooter (Animator)
Sprite Renderer BeeShooter (Sprite Renderer)
Destroy Effect Particle
Bounce Force 2.5
Lifes 2
Sprite-Lit-Default (Material)
Shader Universal Render Pipeline/2D/Sprite-Lit-D
Edit...
Add Component



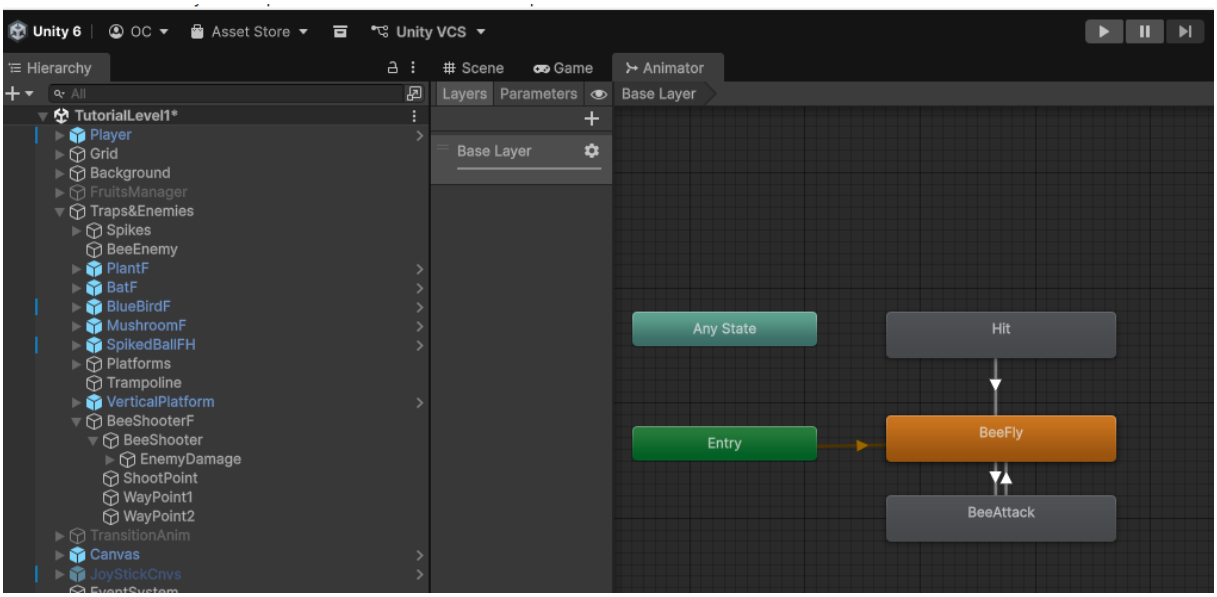
Assets > Scripts > Enemies > EnemyBee.cs > EnemyBee

```
3   public class EnemyBee : MonoBehaviour
    {
        0 références | Champ Unity sérialisé
6       public SpriteRenderer spriteRenderer;
        1 référence | Champ Unity sérialisé
7       public float speed = 0.5f;
        4 références
8       private float waitTime;
        5 références | Champ Unity sérialisé
9       public Transform[] moveSpots;
        2 références | Champ Unity sérialisé
10      public float startWaitTime = 2f;
        6 références
11      private int direction = 0;
        0 références
12      private Vector2 currentPosition;
13
        0 références | Message Unity
14      void Start()
15      {
16          waitTime = startWaitTime;
17      }
18
        0 références | Message Unity
19      void Update()
20      {
21          if (moveSpots.Length == 0 || moveSpots[direction] == null) return;
22
23          transform.position = Vector2.MoveTowards(transform.position, moveSpots[direction].position, speed * Time.deltaTime);
24
25          if (Vector2.Distance(transform.position, moveSpots[direction].position) < 0.1f)
26          {
27              if (waitTime <= 0)
28              {
29                  if (direction < moveSpots.Length - 1)
30                  {
31                      direction++;
32                  }
33                  else
34                  {
35                      direction = 0;
36                  }
37                  waitTime = startWaitTime;
38              }
39              else
40              {
41                  waitTime -= Time.deltaTime;
42              }
43          }
44      }
```



Assets > Scripts > Interactions > BulletBee.cs > BulletBee > Update

```
1   using UnityEngine;
2
    0 références | Script Unity
3   public class BulletBee : MonoBehaviour
4   {
        1 référence | Champ Unity sérialisé
5       public float speed = 2f;
        1 référence | Champ Unity sérialisé
6       public float lifeTime = 2f;
7
        0 références | Message Unity
8       private void Start()
9       {
10          Destroy(gameObject, lifeTime);
11      }
12
        0 références | Message Unity
13      private void Update()
14      {
15          transform.Translate(Vector2.down * speed * Time.deltaTime);
16      }
17  }
18
```



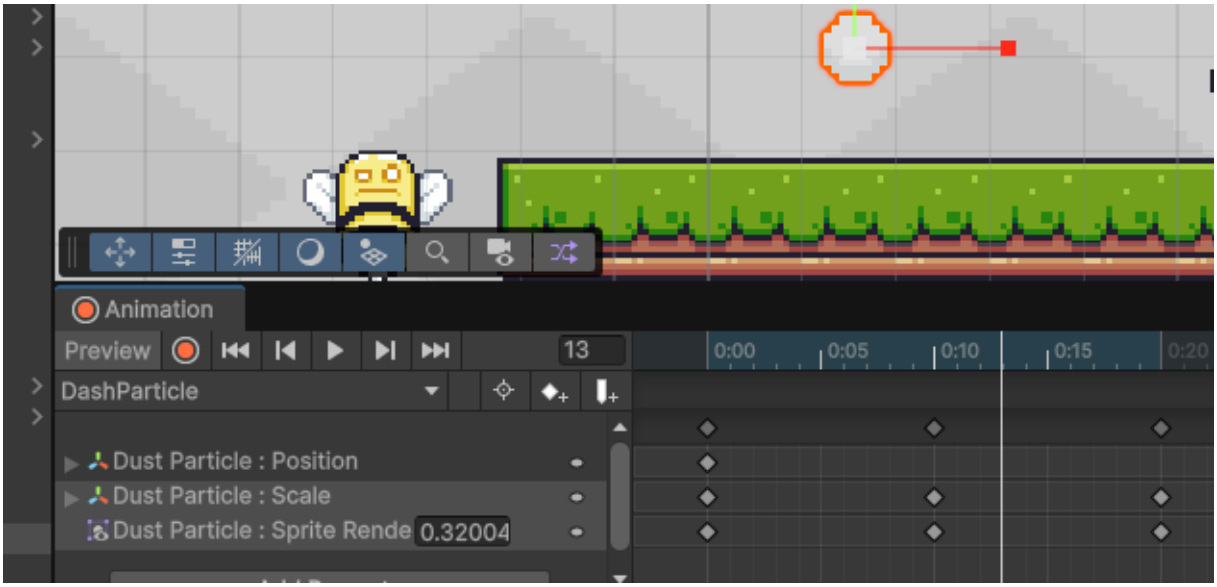
Assets > Scripts > Enemies > BeeAtack.cs > BeeAtack

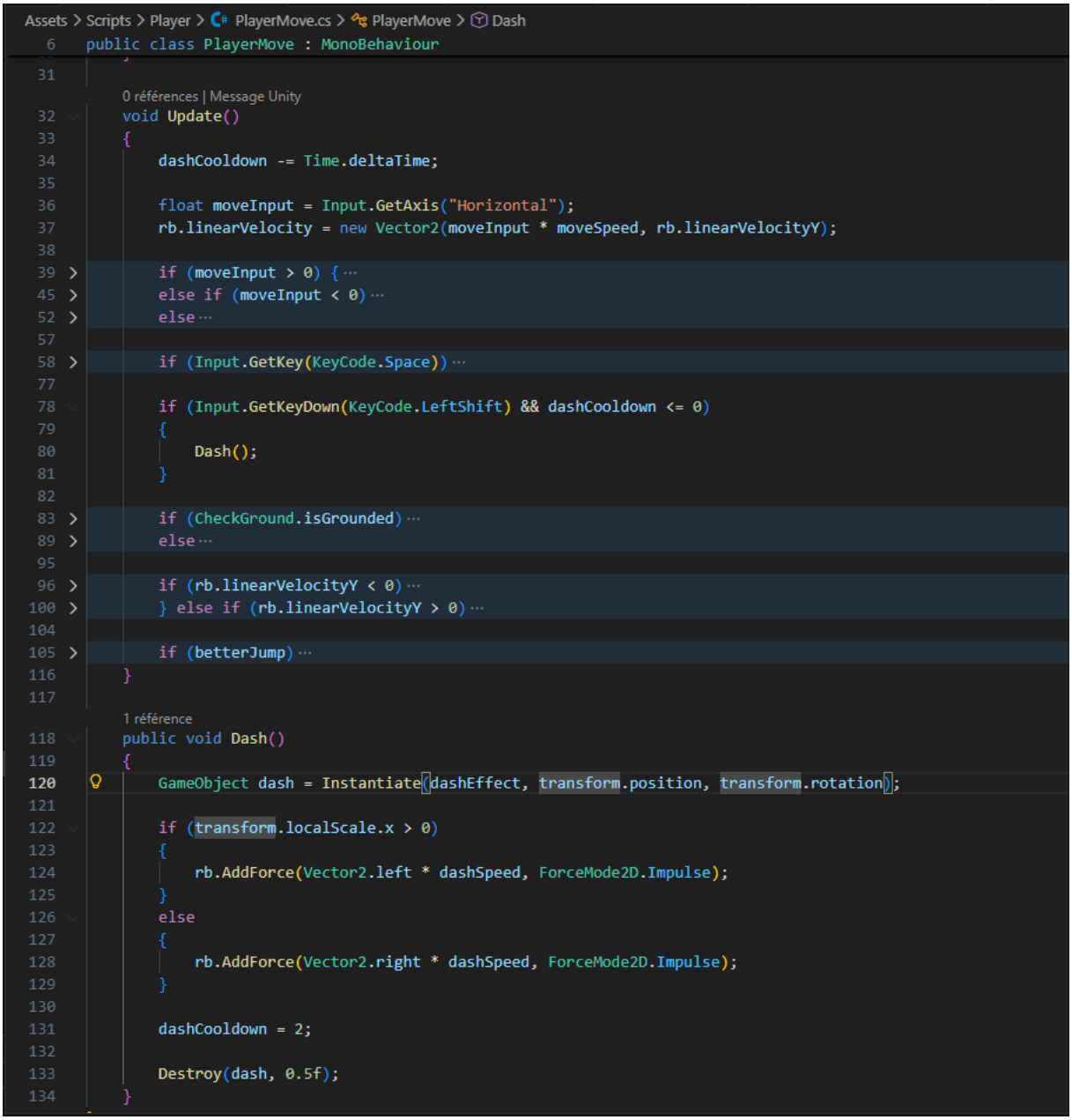
```
using UnityEngine;

0 références | Script Unity
public class BeeAtack : MonoBehaviour
{
    1 référence | Champ Unity sérialisé
    public Animator anim;
    1 référence | Champ Unity sérialisé
    public float distRaycast = 0.5f;
    1 référence
    private float cooldownAttack = 1.5f;
    4 références
    private float currentCooldownAttack;
    1 référence | Champ Unity sérialisé
    public GameObject bullet;

    0 références | Message Unity
    private void Start()
    {
        currentCooldownAttack = 0;
    }

    0 références | Message Unity
    private void Update()
    {
        currentCooldownAttack -= Time.deltaTime;
    }

    0 références
    private void FxedUpdate()
    {
        RaycastHit2D hit = Physics2D.Raycast(transform.position, Vector2.down, distRaycast);

        if (hit.collider != null && hit.collider.CompareTag("Player") && currentCooldownAttack <= 0)
        {
            anim.Play("BeeAttack");
            Invoke("ShootBullet", 0.5f);
            currentCooldownAttack = cooldownAttack;
        }
    }

    0 références
    public void ShootBullet()
    {
        GameObject newBullet = Instantiate(bullet, transform.position, transform.rotation);
    }
}
```

Assets > Scripts > Enemies > BeeAtack.cs > BeeAtack > FixedUpdate

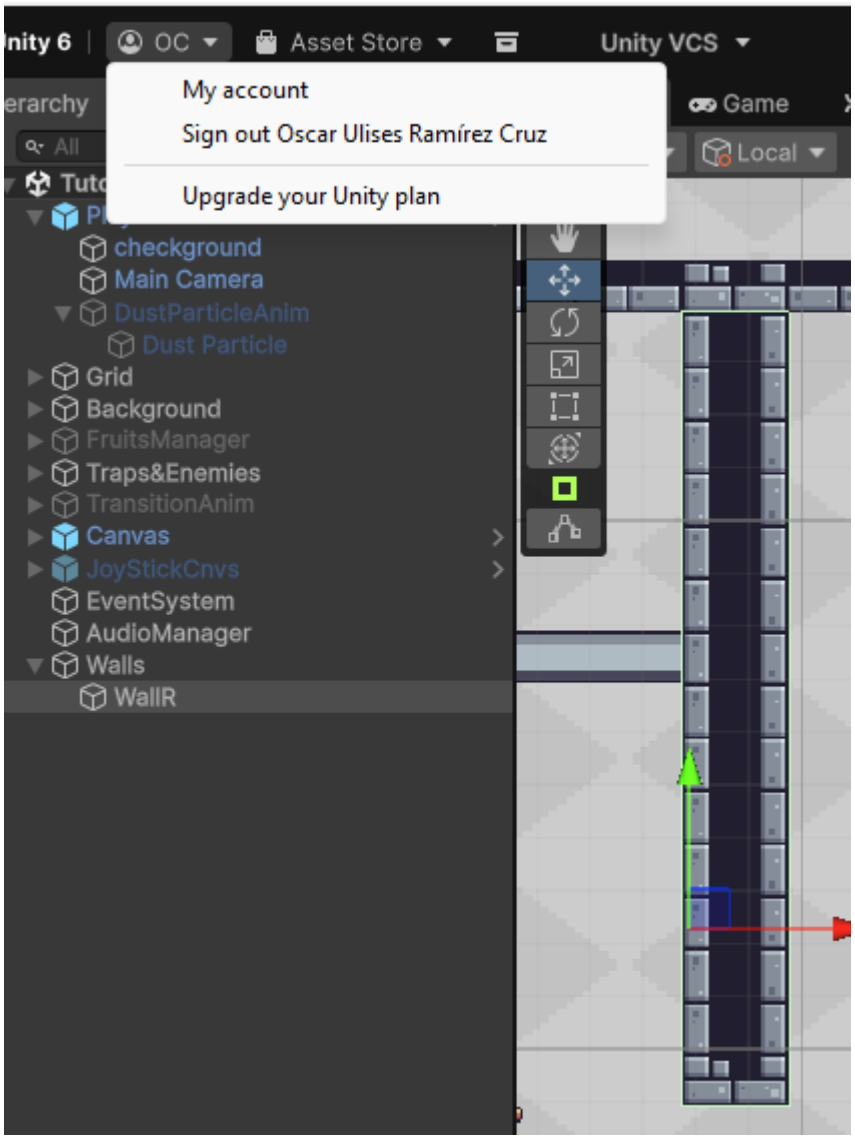
```
3   public class BeeAtack : MonoBehaviour
        2 références | Champ Unity sérialisé
6       public float distRaycast = 0.5f;
        1 référence
7       private float cooldownAttack = 1.5f;
        4 références
8       private float currentCooldownAttack;
        1 référence | Champ Unity sérialisé
9       public GameObject bullet;
10
        0 références | Message Unity
11      private void Start()
12      {
13          currentCooldownAttack = 0;
14      }
15
        0 références | Message Unity
16      private void Update()
17      {
18          currentCooldownAttack -= Time.deltaTime;
19
20          Debug.DrawRay(transform.position, Vector2.down * distRaycast, Color.red);
21      }
22
        0 références | Message Unity
23      private void FixedUpdate()
24      {
25          RaycastHit2D hit = Physics2D.Raycast(transform.position, Vector2.down, distRaycast);
26
27          if (hit.collider != null && hit.collider.CompareTag("Player") && currentCooldownAttack <= 0)
28          {
29              anim.Play("BeeAttack");
30              Invoke("ShootBullet", 0.5f);
31              currentCooldownAttack = cooldownAttack;
32          }
33      }
34
        0 références
35      public void ShootBullet()
36      {
37          GameObject newBullet = Instantiate(bullet, transform.position, transform.rotation);
38      }
```

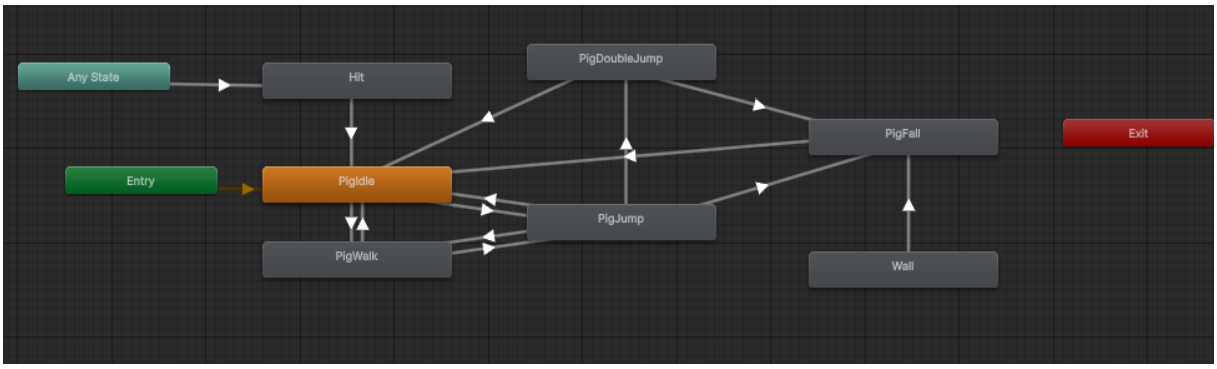
Tutorial 28


Animation
Preview
13
0:00
0:05
0:10
0:15
0:20
DashParticle
Dust Particle : Position
Dust Particle : Scale
Dust Particle : Sprite Rende
0.32004


Assets > Scripts > Player > PlayerMove.cs > PlayerMove > Dash
public class PlayerMove : MonoBehaviour
0 références | Message Unity
void Update()
{
dashCooldown -= Time.deltaTime;
float moveInput = Input.GetAxis("Horizontal");
rb.linearVelocity = new Vector2(moveInput * moveSpeed, rb.linearVelocityY);
if (moveInput > 0) {
else if (moveInput < 0)
else
if (Input.GetKey(KeyCode.Space))
if (Input.GetKeyDown(KeyCode.LeftShift) && dashCooldown <= 0)
{
Dash();
}
if (CheckGround.isGrounded)
else
if (rb.linearVelocityY < 0)
} else if (rb.linearVelocityY > 0)
if (betterJump)
}
1 référence
public void Dash()
{
GameObject dash = Instantiate(dashEffect, transform.position, transform.rotation);
if (transform.localScale.x > 0)
{
rb.AddForce(Vector2.left * dashSpeed, ForceMode2D.Impulse);
}
else
{
rb.AddForce(Vector2.right * dashSpeed, ForceMode2D.Impulse);
}
dashCooldown = 2;
Destroy(dash, 0.5f);
}

Tutorial 29


nity 6
OC
Asset Store
Unity VCS
My account
Sign out Oscar Ulises Ramírez Cruz
Upgrade your Unity plan
erarchy
Game
All
Local
checkground
Main Camera
DustParticleAnim
Dust Particle
Grid
Background
FruitsManager
Traps&Enemies
TransitionAnim
Canvas
JoyStickCnvs
EventSystem
AudioManager
Walls
WallR


Any State
Hit
PigDoubleJump
PigFall
Exit
Entry
PigIdle
PigJump
PigWalk
Wall

Assets > Scripts > Player > PlayerMove.cs > PlayerMove > isTouching

```
6   public class PlayerMove : MonoBehaviour
26  {
        0 références
27      bool isTouchingFront = false;
        0 références
28      bool wallSliding = false;
29
        0 références | Champ Unity sérialisé
30      public float wallSlidingSpeed = 0.75f;
31
        0 références
32      bool isTouching;
33
```

```
if (moveInput > 0 && isTouching == false) {
    //sprRnd.flipX = true;
    transform.localScale = new Vector3( 0.8f, 0.8f, 1);
    anim.SetBool("Ru
    (Champ) static bool CheckGround.isGrounded
    if (CheckGround.isGrounded) dustAnimation.SetActive(true);

    if (Input.GetKeyDown(KeyCode.LeftShift) && dashCooldown <= 0) Dash();
}
else if (moveInput < 0 && isTouching == false)
{
    //sprRnd.flipX = false;
    transform.localScale = new Vector3(0.8f, 0.8f, 1);
    anim.SetBool("Run", true);
    if (CheckGround.isGrounded) dustAnimation.SetActive(true);

    if (Input.GetKeyDown(KeyCode.LeftShift) && dashCooldown <= 0) Dash();
}
else...

if (Input.GetKey(KeyCode.Space) && !wallSliding)
{
    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 (isTouchingFront && !CheckGround.isGrounded)
{
    wallSliding = true;
}
else
{
    wallSliding = false;
}

if (wallSliding)
{
    anim.Play("Wall");
    rb.linearVelocity = new Vector2(rb.linearVelocityX, Mathf.Clamp(rb.linearVelocityY, -wallSlidingSpeed, float.MaxValue));
}
```

```
0 références | Message Unity
private void OnCollisionStay2D(Collision2D collision)
{
    if (collision.gameObject.CompareTag("wall"))
    {
        isTouchingFront = true;
        isTouching = true;
    }
}

0 références | Message Unity
private void OnCollisionExit2D(Collision2D collision)
{
    isTouchingFront = false;
    isTouching = false;
}
```

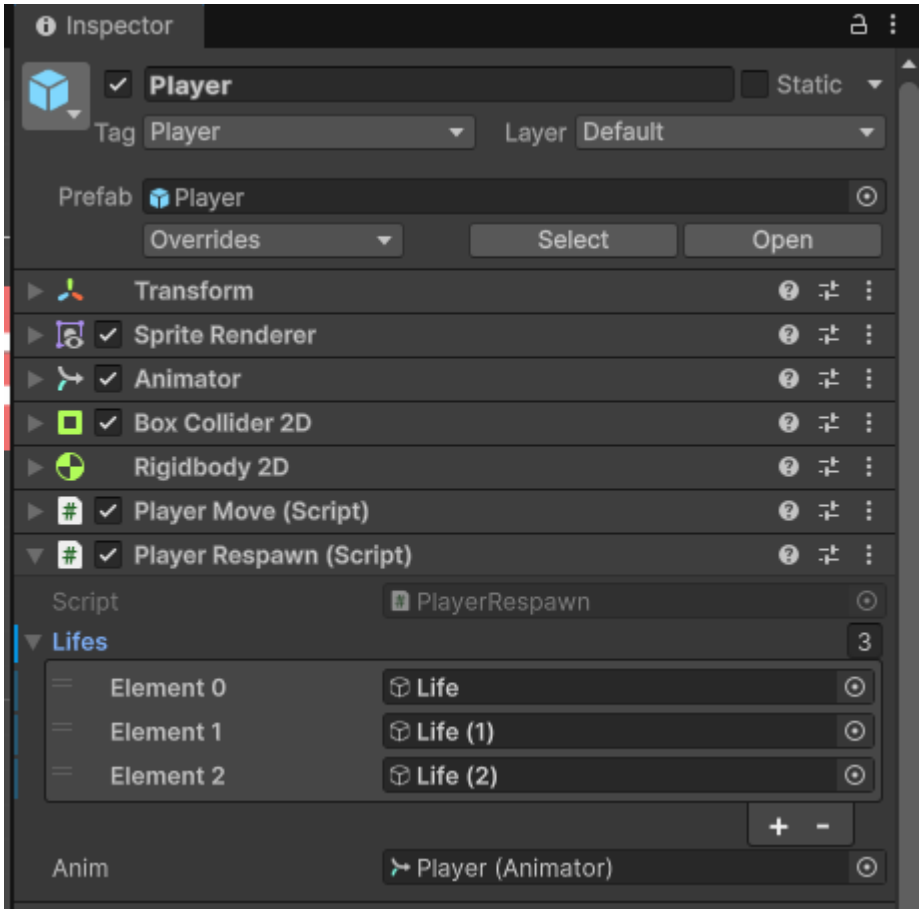
Tutorial 30


Unity 6
OC
Asset Store
Unity VCS
Hierarchy
Scene
Game
Animator
Animation
Pivot
Local
0.16
All
TutorialLevel1*
Player
checkground
Main Camera
DustParticleAnim
Dust Particle
Grid
Background
FruitsManager
Traps&Enemies
TransitionAnim
Canvas
OptionsBtn
Text (TMP)
FruitsScore
FruitsCount
TotalFruits
Slash
SettingsPanel
Lifes
Life
Life (1)
Life (2)
Frutas: 0 / 0

Assets > Scripts > Player > PlayerRespawn.cs > PlayerRespawn > Start

```
6   public class PlayerRespawn : MonoBehaviour
14
        0 références | Message Unity
15      void Start()
16      {
17          life = lifes.Length;
18          if (PlayerPrefs.GetFloat("chPntX") != 0 && PlayerPrefs.GetFloat("chPntY") != 0)
19          {
20              chPntX = PlayerPrefs.GetFloat("chPntX");
21              chPntY = PlayerPrefs.GetFloat("chPntY");
22
23              transform.position = new Vector2(chPntX, chPntY);
24          }
25      }
26
        0 références | Message Unity
27      private void Update()
28      {
29          if (life < 1)
30          {
31              anim.Play("Hit");
32              Destroy(lifes[0]);
33              SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
34          }
35          else if (life < 2) {
36              anim.Play("Hit");
37              Destroy(lifes[1]);
38          }
39          else if (life < 3) {
40              anim.Play("Hit");
41              Destroy(lifes[2]);
42          }
43      }
44
        1 référence
45      public void ReachedCheckPoint(float x, float y)
46      {
47          PlayerPrefs.SetFloat("chPntX", transform.position.x);
48          PlayerPrefs.SetFloat("chPntY", transform.position.y);
49      }
50
        2 références
51      public void PlayerDamaged()
52      {
53          life--;
54      }
55  }
56
```

```
        1 référence
27      private void CheckLifes()
28      {
29          if (life < 1)
30          {
31              anim.Play("Hit");
32              Destroy(lifes[0]);
33              SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
34          }
35          else if (life < 2) {
36              anim.Play("Hit");
37              Destroy(lifes[1]);
38          }
39          else if (life < 3) {
40              anim.Play("Hit");
41              Destroy(lifes[2]);
42          }
43      }
44
        1 référence
45      public void ReachedCheckPoint(float x, float y)
46      {
47          PlayerPrefs.SetFloat("chPntX", transform.position.x);
48          PlayerPrefs.SetFloat("chPntY", transform.position.y);
49      }
50
        2 références
51      public void PlayerDamaged()
52      {
53          life--;
54          CheckLifes();
55      }
56  }
57
```


Inspector
Player
Static
Tag Player
Layer Default
Prefab Player
Overrides
Select
Open
Transform
Sprite Renderer
Animator
Box Collider 2D
Rigidbody 2D
Player Move (Script)
Player Respawn (Script)
Script PlayerRespawn
Lifes 3
Element 0 Life
Element 1 Life (1)
Element 2 Life (2)
Anim Player (Animator)