

DESARROLLO DE VIDEOJUEGOS Y REALIDAD VIRTUAL CON UNITY 3D

Tema 3: Gráficos 3D y 2D

1

TEMA 3. GRÁFICOS 3D Y 2D

- Modelos 3D
- **Sprites 2D**
- Shaders, materiales y texturas
- Sistemas de partículas

SPRITES 2D

- Importar arrastrando.
 - Unity asigna el tipo “Sprite (2D and UI)” automáticamente.
- Sprite size (proporción del Sprite):



- Evitar difuminados en los sprites:



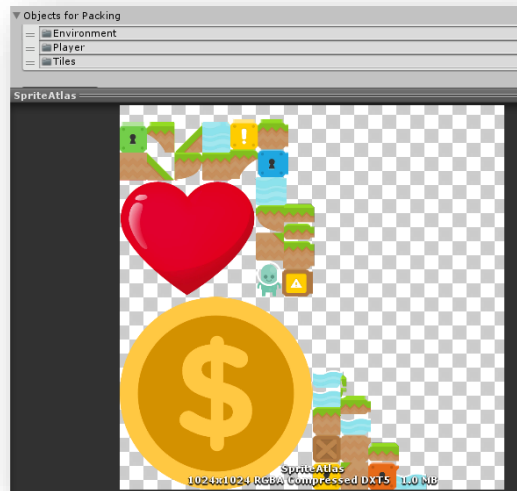
- La propiedad “Order in Layer” permite determinar en qué orden (profundidad) van a dibujarse los sprites en la pantalla.

SPRITES 2D

- Crear Atlas.
 - En Edit->Project Settings->Editor verificar que está activo el módulo “Sprite Packer”
 - Agregar al proyecto un componente SpriteAtlas
 - Agregar los sprites o las carpetas de sprites
 - Configurar según la siguiente configuración:
 - Include in Build: incluir en las compilaciones
 - Allow rotation: permitir rotar los sprites
 - Tight packing: permite que los sprites estén más apretados en el mapa.
 - Read/Write enabled: permite leer las texturas desde los scripts.
 - Generate Mip Maps: generar versiones reducidas de los sprites para utilizar cuando la textura sea pequeña.
 - Filter mode: modo de filtrado de la textura.

SPRITES 2D

- Crear Atlas.

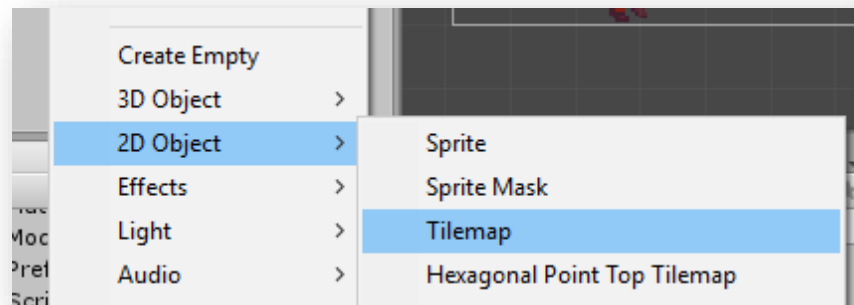


Reducción de “draw calls”



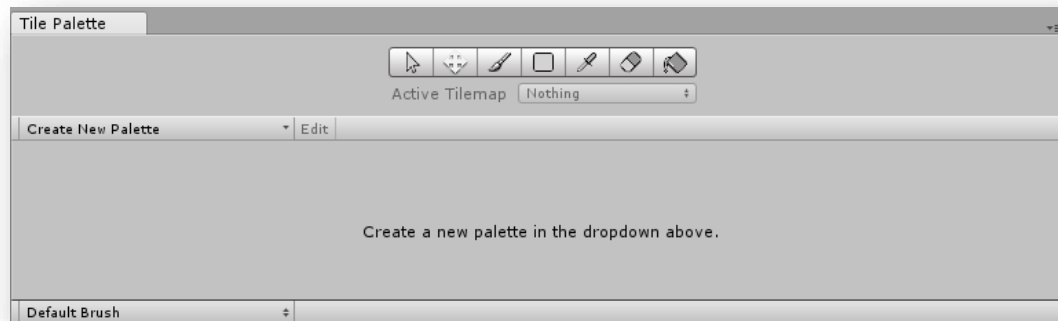
SPRITES 2D

- **TILE MAPS:**
 - Permite la creación de niveles 2D usando tiles.
 - Unity proporciona 2D extras a través de GitHub.
 - <https://github.com/Unity-Technologies/2d-extras>
 - Descomprimir y copiar la carpeta Tilemap en Assets
 - La creación de un Tilemap provoca la creación de un Grid y un Tilemap.

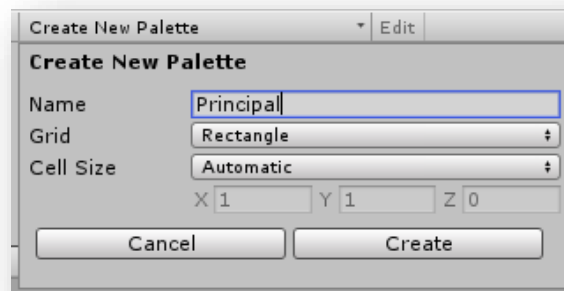


SPRITES 2D

- **TILE PALETTE:**
 - Permite crear paletas de Tiles.
 - Windows -> Tile Palette.



- Crear paleta.



SPRITES 2D

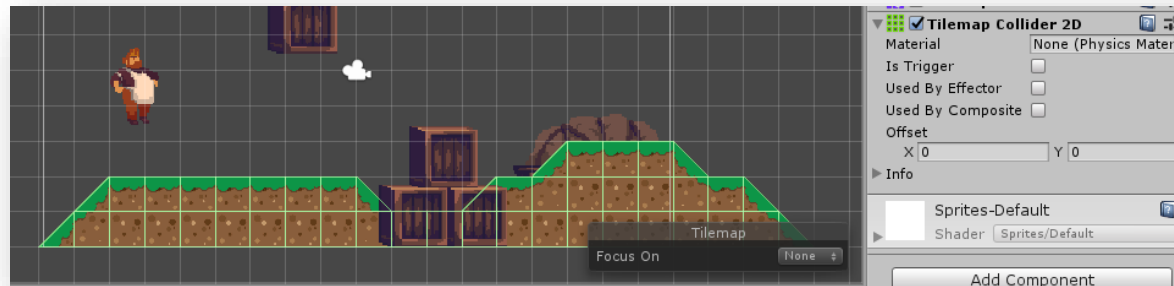
- Después de crear el TilePalette:
- Incorporar los sprites de los tiles al proyecto.
- Ajustar los tamaños mediante el atributo de los sprites “Pixel Per Unit” (en función del tamaño de los sprites).
- Arrastrar los sprites al TilePalette.
 - Guardar los tiles en una carpeta.
- Agregar un Tilemap al juego.

SPRITES 2D

- Edición (muchacha atención al “Active Tilemap” de la ventana “Tile Palette”:
 - Seleccionar tile y pintar en el juego.
 - Borrado:
 - Seleccionar el componente rectángulo del “Tile Palette”
 - Pulsar la tecla “Ctrl” y marcar un área en la escena (una o más cuadrículas).
 - Seleccionar el borrador del “Tile Palette” y borrar.
 - Para trabajar con varios tiles simultáneamente: seleccionar el selector (el cuentagotas del Tile Palette) y con la tecla Ctrl pulsada delimitar el conjunto de tiles que forman el área.
- Copiado:
 - Seleccionar el componente cuentagotas del “Tile Palette”
 - Pulsar la tecla “Ctrl” y marcar un área en la escena (una o más cuadrículas).
 - Se activa automáticamente una copia del área seleccionada.

SPRITES 2D

- Agregar un “Tilemap Collider 2D” al Tilemap. Se creará el collider en función de la transparencia.



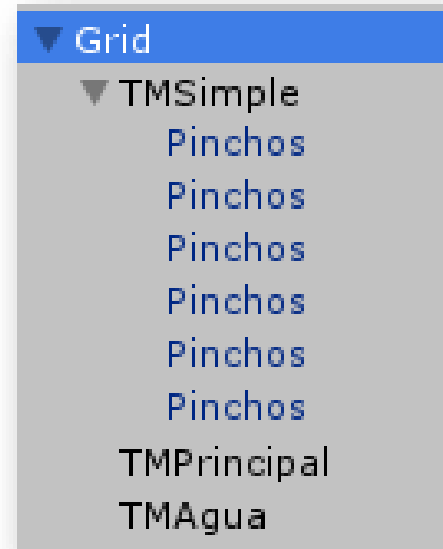
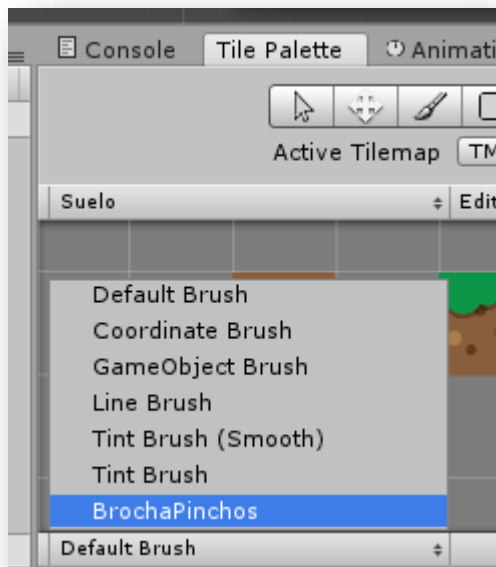
- Para optimizar:
 - Se agrega al tilemap un componente “Composite Collider 2D”. Esto provoca que se agregue un “Rigidbody 2D” y un “Composite Collider 2D”.
 - Se pone el componente “Rigidbody 2D” como estático.
 - Se marca en el componente “Tilemap Collider 2D” la propiedad “Used By Composite”. De esta manera todos los tiles tendrán un único collider.

SPRITES 2D

- Uso de un PREFAB como Brush
 - Crear una carpeta Brushes
 - Crear una brocha: Create->Brushes->Prefab brush
 - En el campo Size se le indica 1 (más valores para generación aleatoria).
 - Se arrastra el Prefab.

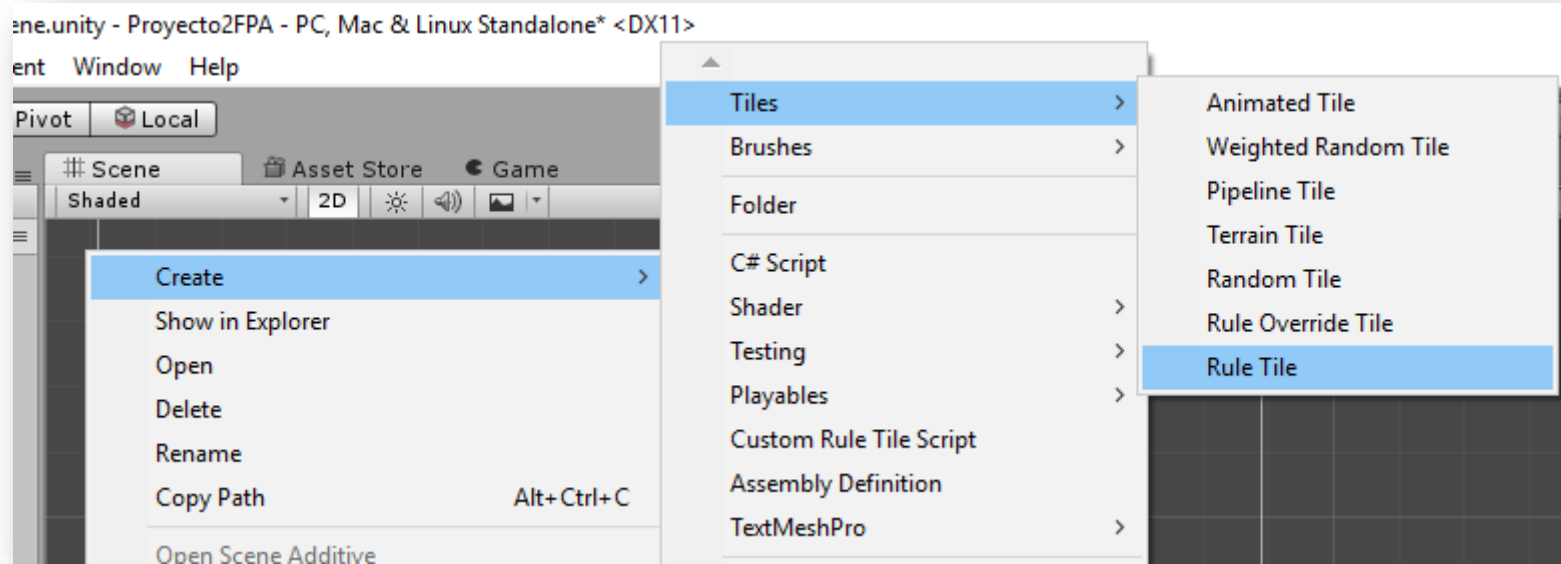
SPRITES 2D

- Uso de un PREFAB como Brush
 - Ahora en Tile Palette aparece una nueva brocha.
 - Se selecciona la brocha y se utiliza.
 - En el TileMap aparecerán los objetos correspondientes.



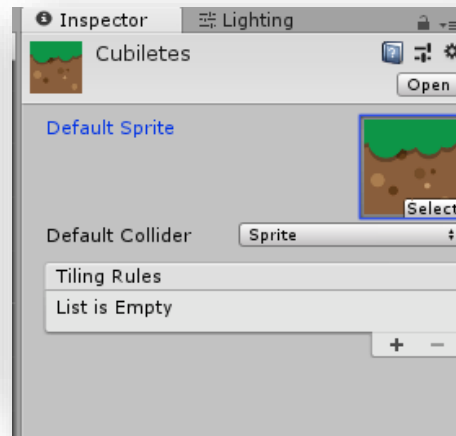
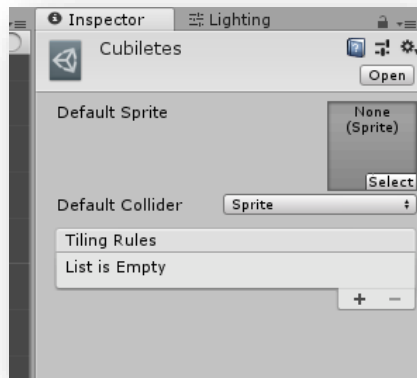
SPRITES 2D

- Repositorio 2d-extras (Rule Tile):
 - Crear dentro de la carpeta Tiles un Rule Tile:

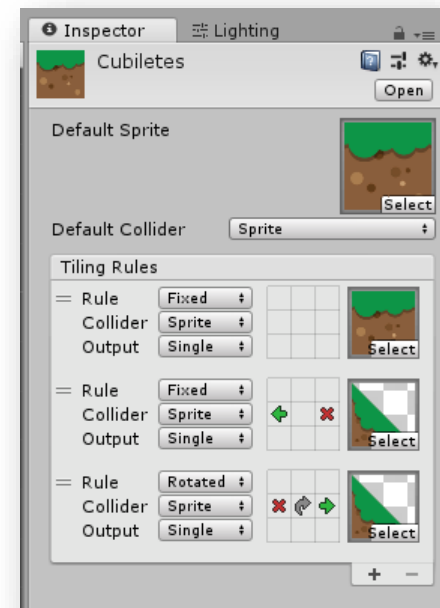


SPRITES 2D

- Repositorio 2d-extras (Rule Tile):
 - Marcar el Sprite por defecto:



- Crear las reglas:



SPRITES 2D

- **Constant Force 2D**
 - Componente que permite aplicar una aceleración constante a objetos con Rigidbody.
- **Effectors 2D:**
 - Colección de componentes que provocan fuerzas físicas sobre objetos con el componente Collider2D cuando entran en contacto.
 - PlatformEffector2D. Proporciona a los objetos la capacidad de ser atravesados desde abajo pero no desde arriba (plataforma).
 - SurfaceEffector2D. Desplaza los objetos situados sobre su superficie, simulando el comportamiento de una cinta transportadora (cuidado de no estar aplicando una aceleración constante a los objetos o no tendrá efecto).
 - PointEffector2D. Atrae o repele los objetos que entran en contacto con un punto.
 - BouyancyEffector2D. Hace flotar a los objetos.
 - AreaEffector2D. Aplica fuerzas en cualquier dirección sobre el objeto (con collider2D) con el que colisiona.
- **NOTA:** cuidado de no estar aplicando una aceleración constante a los objetos o no tendrán efecto en algunos casos.

SPRITES 2D

- 2D Joints
 - DistanceJoint2D. Mantiene dos objetos unidos a cierta distancia máxima.
 - FixedJoint2D. Mantiene dos objetos unidos a cierta distancia fija.
 - FrictionJoint2D. Determina una fricción entre objetos.
 - HidgeJoint2D. Fija un eje de rotación.
 - RelativeJoint2D. Permite determinar la posición de un objeto en función de la posición de otro.
 - SliderJoint2D. Permite deslizar un objeto a través de una línea.
 - SpringJoint2D. Une dos objetos con rigidbody con un resorte.
 - TargetJoint2D. Similar a SprintJoint pero no necesita rigidbody.
 - WheelJoint2D. Simula ruedas y suspensión.

SPRITES 2D - ROTACIÓN

- Rotar un Sprite desde teclado:
 - En función de la pulsación, se cambia la escala del Sprite en el eje x:

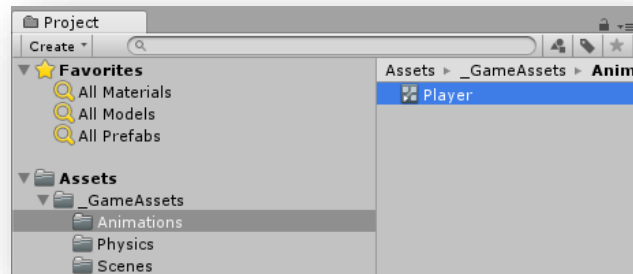
```
float xInput = Input.GetAxis("Horizontal");  
if (xInput > 0)  
{  
    transform.localScale = new Vector3(1, 1, 1);  
}  
else if (xInput < 0)  
{  
    transform.localScale = new Vector3(-1, 1, 1);  
}
```

SPRITES 2D - MOVIMIENTO

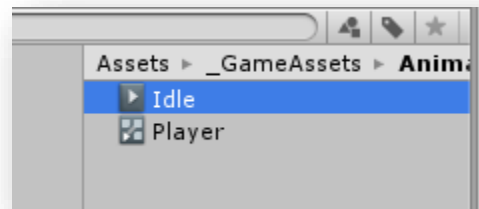
- Mover un Sprite desde teclado:
 - El script debe contener una referencia (se obtendrá en el Awake) a Rigidbody2D:
 - `Rigidbody2D miRigidbody;`
 - `miRigidbody = GetComponent<Rigidbody2D>();`
 - Se obtiene la componente horizontal de movimiento en el FixedUpdate ya que vamos a modificar el Rigidbody2D:
 - `float xInput = Input.GetAxis("Horizontal");`
 - Se modifica la velocidad en el eje x del Rigidbody:
 - `Vector2 velocidad = miRigidbody.velocity;`
 - `velocidad.x = xInput * velocidadX;`
 - `miRigidbody.velocity = velocidad;`

SPRITES 2D - ANIMACIONES

- Animación 2D:
 - Crear un Animator Controller en la carpeta Animations

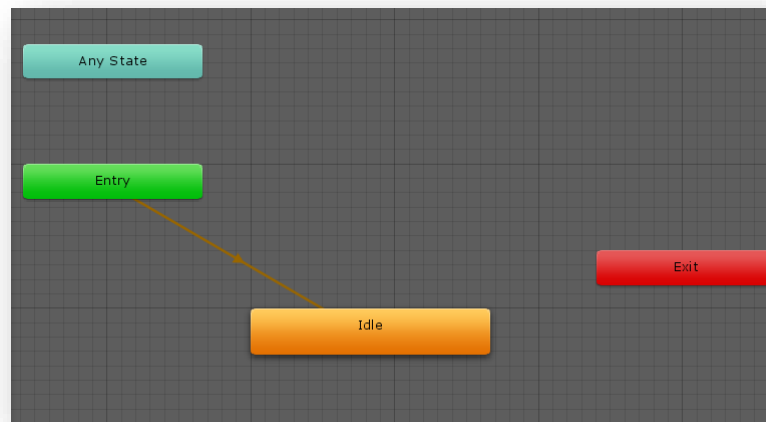


- Crear un Animation en la carpeta Animations.



SPRITES 2D - ANIMACIONES

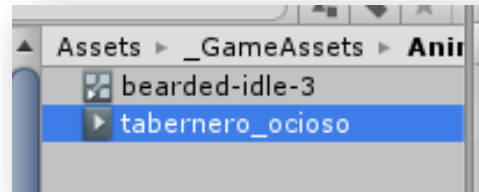
- Animación 2D:
 - Arrastrar la animación al Animator (Window Animator).



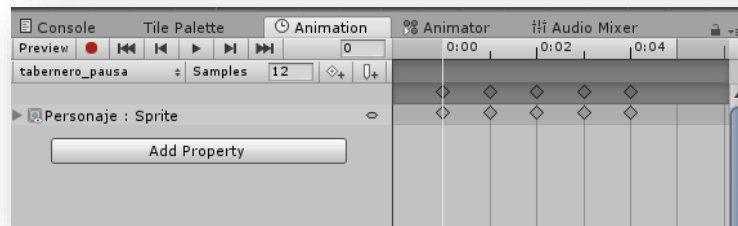
- Arrastrar el Sprite “Idle” a las propiedades de Animation.
- Arrastrar el Animator al GameObject como componente.
 - Si se añade el Animator desde el botón “Add component”, se debe arrastrar el objeto Animator de la animación al Controller del Animator del GameObject.

SPRITES 2D - ANIMACIONES

- Animación 2D:
 - Seleccionamos todos los sprites de la animación y se la asignamos a la jerarquía.
 - Almacenamos la animación (fichero .anim) en la carpeta Animations.
 - Se crean dos elementos: Animación (tabernero_ocioso) y Animator (bearded-idle-3).

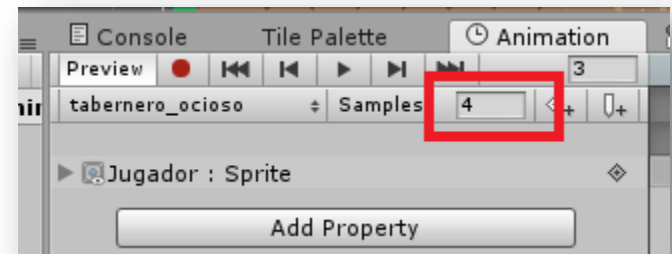
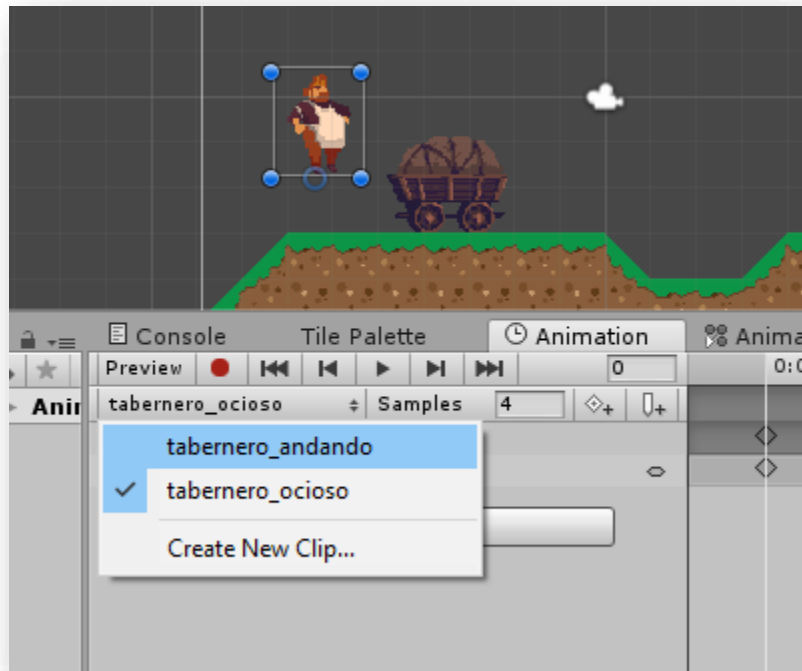


- Modificación de la velocidad. “Samples” indica el número de muestras por frame. A mayor valor, mayor velocidad.



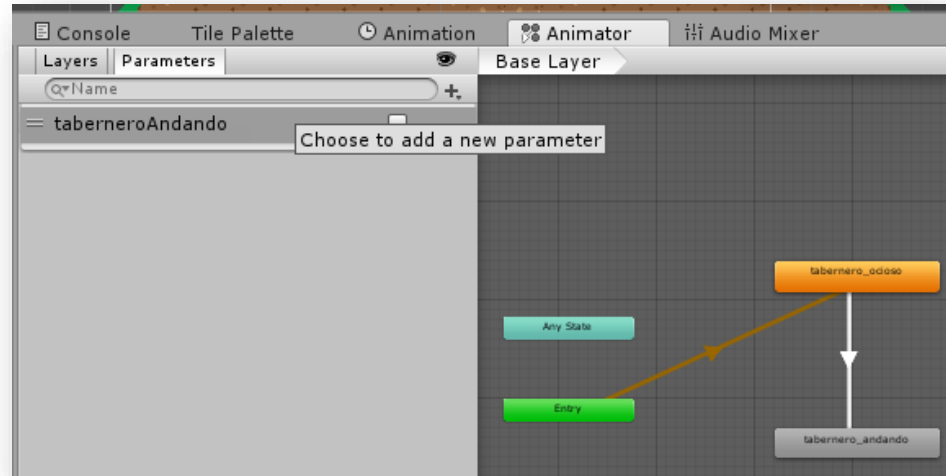
SPRITES 2D - ANIMACIONES

- Animación 2D:
 - Configurar la animación (Window-Animation-Animation –Ctrl+6).



SPRITES 2D - ANIMACIONES

- Animación 2D - Transiciones:
 - Se arrastran a la ventana Animator (Window-Animation-Animator), arrastrando en primer lugar el estado inicial.
 - Se crean los parámetros:



- Se programan los cambios de valor de los parámetros en el script:
 - `miAnimator.SetBool("taberneroAndando", true);`