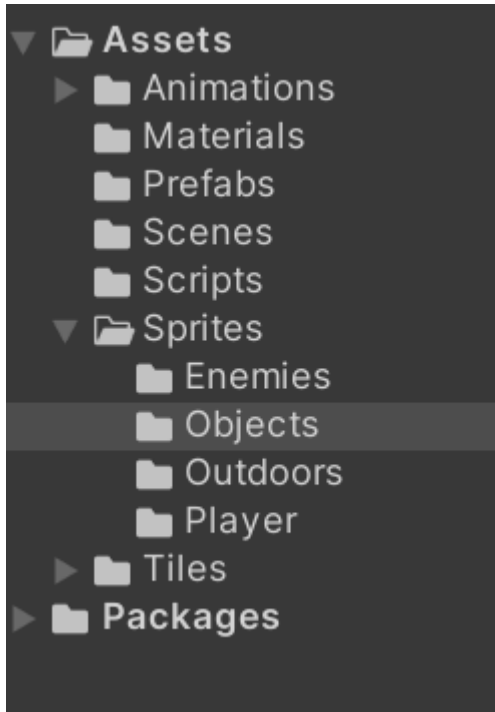
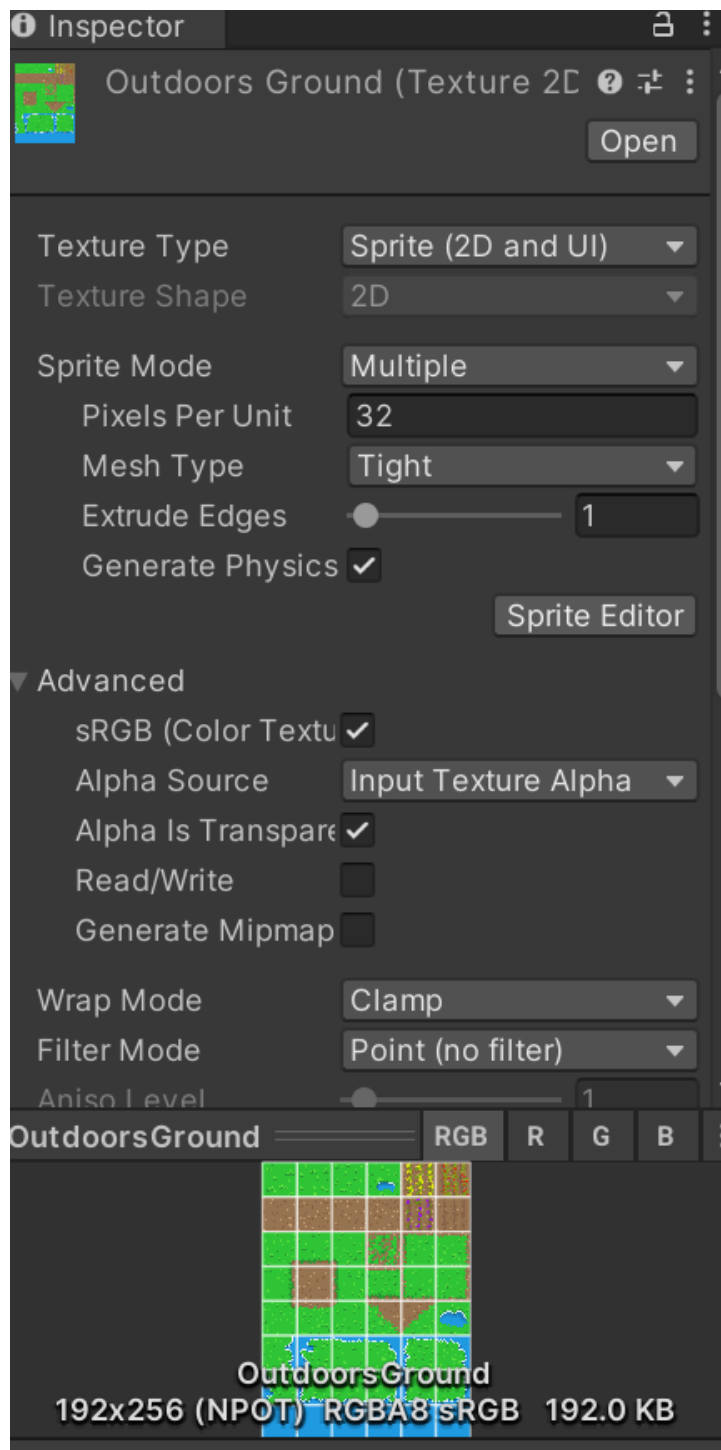


Tutorial 3

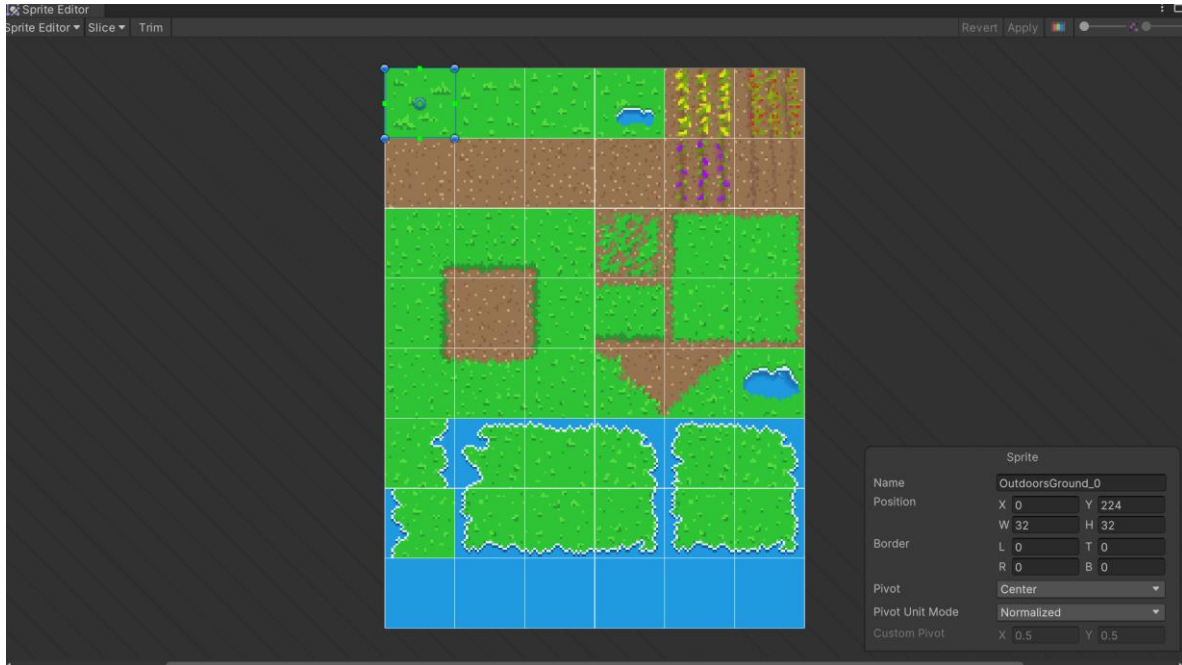
Configuración inicial de carpetas para Tilemaps. Creamos la estructura de carpetas: "Objects" y "Outdoors" dentro de Sprites, y carpetas similares en Tiles. Importamos la hoja de sprites "OutdoorsGround.png" y la configuramos como Multiple con PPU 32.



Configuración inicial de carpetas para Tilemaps. Creamos la estructura de carpetas: "Objects" y "Outdoors" dentro de Sprites, y carpetas similares en Tiles. Importamos la hoja de sprites "OutdoorsGround.png" y la configuramos como Multiple con PPU 32.



División de la hoja de sprites en tiles individuales. Usamos el Sprite Editor con Grid By Cell Size de 32x32 píxeles para cortar la hoja de sprites en tiles individuales. Aplicamos los cambios y verificamos que las divisiones sean correctas.

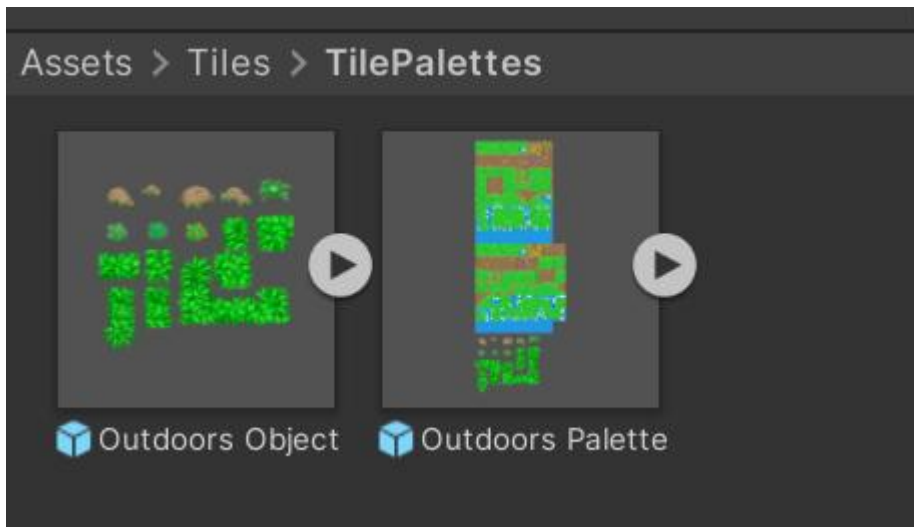


Creación del primer Tilemap. En Hierarchy creamos un Tilemap rectangular mediante 2D Object → Tilemap → Rectangular. Aparece un objeto Grid que contiene el Tilemap, estableciendo la base para construir nuestro entorno.

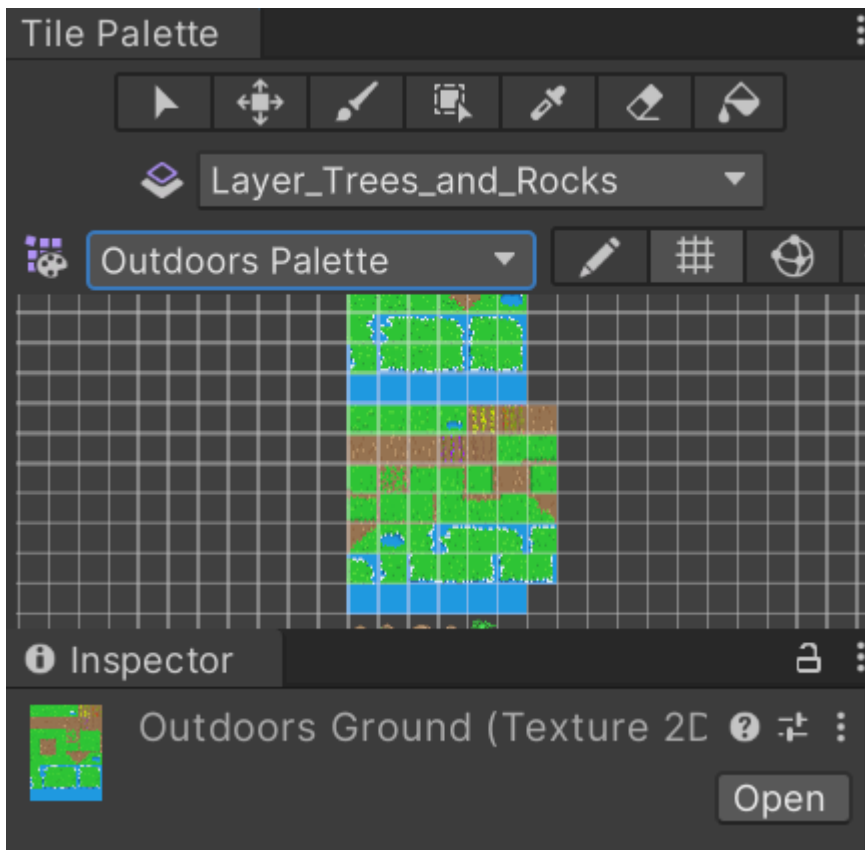


Configuración de la Tile Palette. Abrimos Window → 2D → Tile Palette y creamos una nueva paleta llamada "Outdoors Palette". La guardamos en la carpeta TilePalettes y comenzamos a organizar nuestro conjunto de herramientas de dibujo.

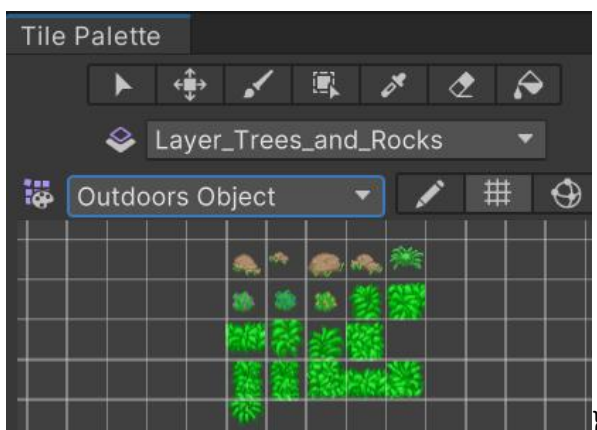
Creación de paleta para objetos. Generamos una segunda Tile Palette llamada "Outdoors Object" específicamente para los elementos decorativos. Mantenemos las paletas separadas para mejor organización del flujo de trabajo.



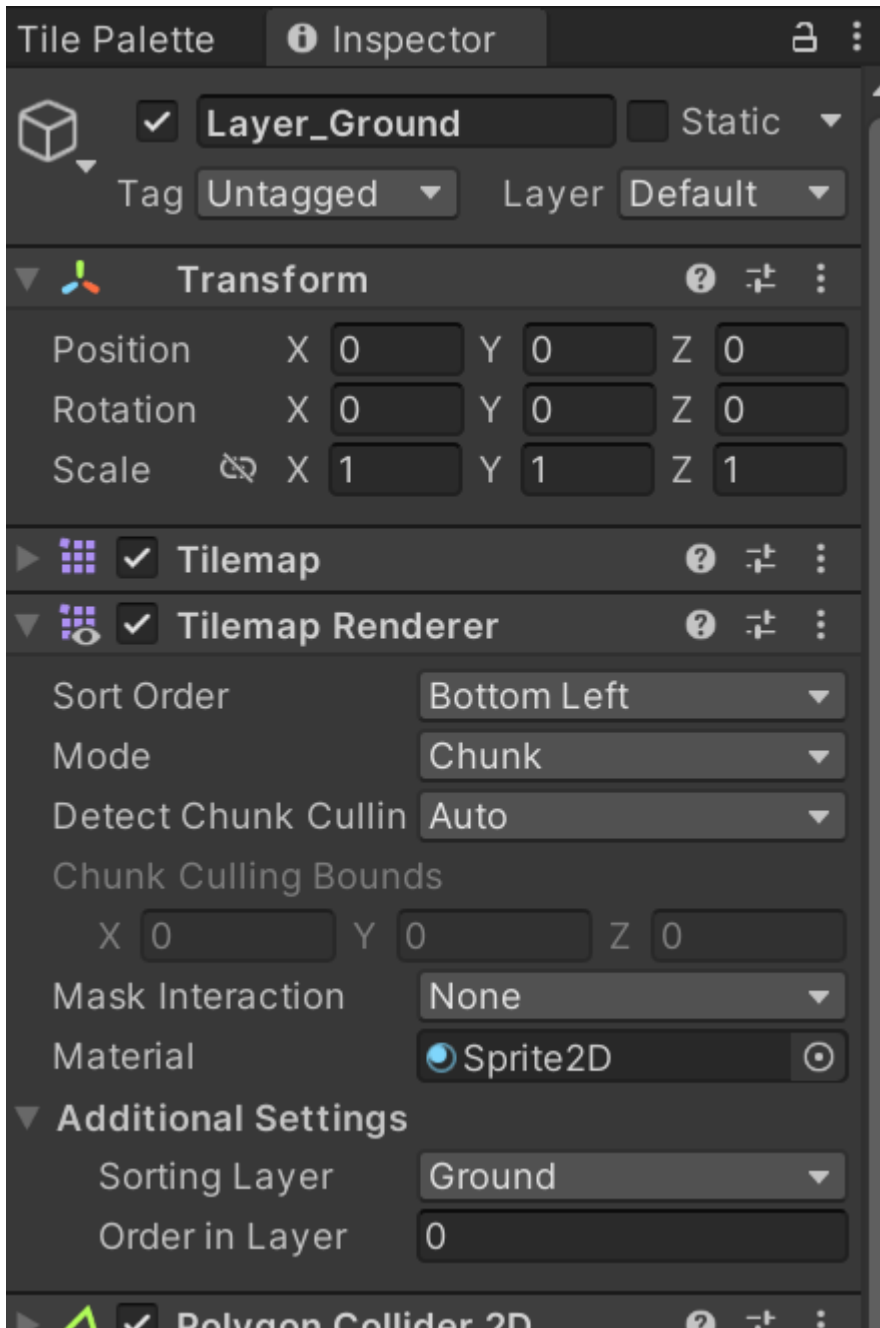
Llenado de la Tile Palette con sprites. Arrastramos la hoja de sprites OutdoorsGround a la Tile Palette. Unity genera automáticamente los tiles en la carpeta designada, creando nuestro conjunto de herramientas para dibujar el terreno.

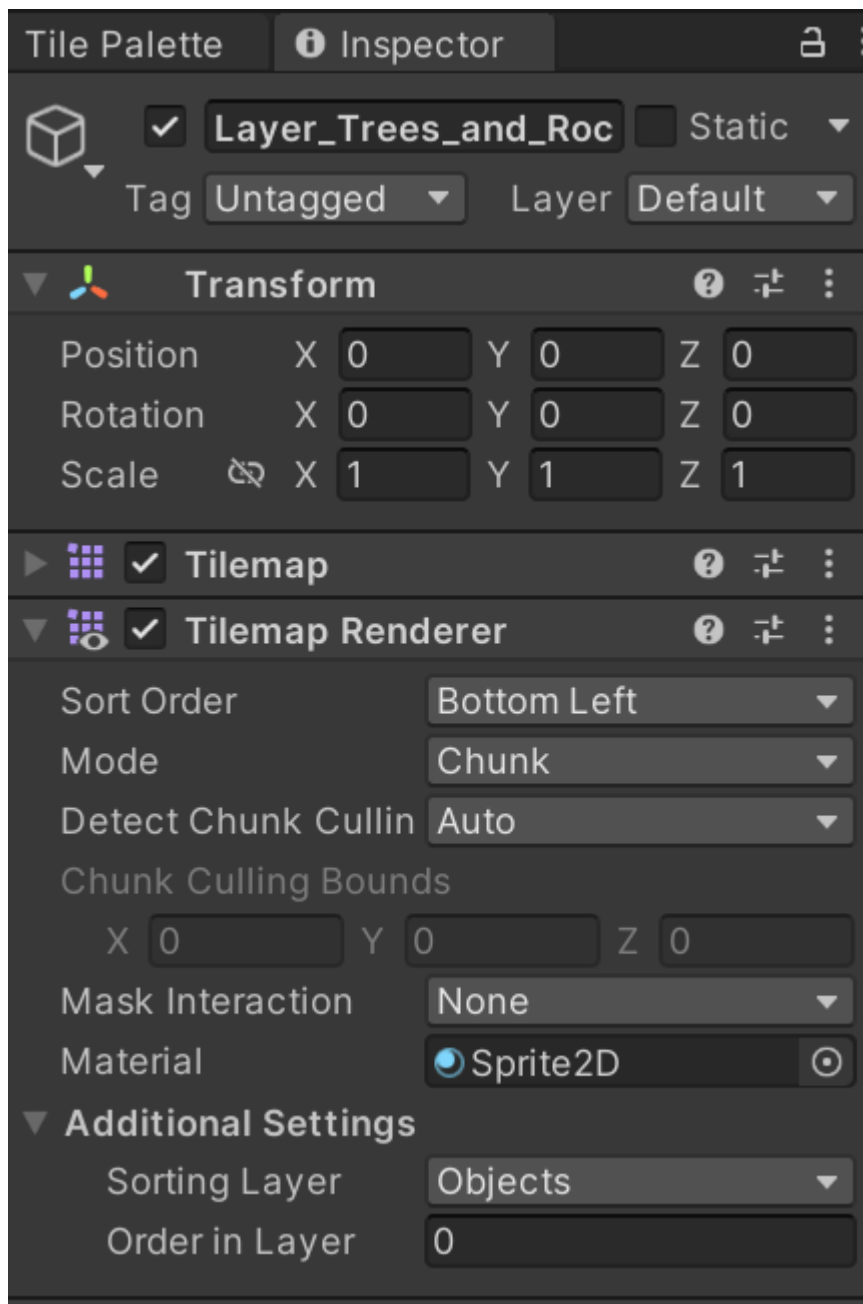


Implementación de múltiples Tilemaps. Creamos un segundo Tilemap llamado "Layer_Trees_and_Rocks" para los objetos decorativos. Utilizamos el menú Active Tilemap en la Tile Palette para cambiar entre capas al dibujar.



Configuración de Sorting Layers para Tilemaps. Establecemos las Sorting Layers "Ground" y "Objects", asignando cada Tilemap a su respectiva capa. Esto controla el orden de renderizado entre suelo y objetos.







Adaptive Performance
Audio
Burst AOT Settings
Editor
Graphics
Input Manager
Memory Settings
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
Tags and Layers
TextMesh Pro
Time
Timeline
UI Builder
▼ Version Control
 Unity Version Control Set
Visual Scripting
XR Plugin Management

Quality

Levels			
Very Low	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Low	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Medium	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
High	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Very High	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Ultra	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Default ▼ ▼

Add Quality Level

Current Active Quality Level

Name Ultra

Rendering

Render Pipeline Asset None (Render Pipeline Asset)

Pixel Light Count 4

Anti Aliasing Disabled

Realtime Reflection Probes ☒

Resolution Scaling Fixed DPI Factor 1

VSync Count Every V Blank

Textures

Global Mipmap Limit 0: Full Resolution

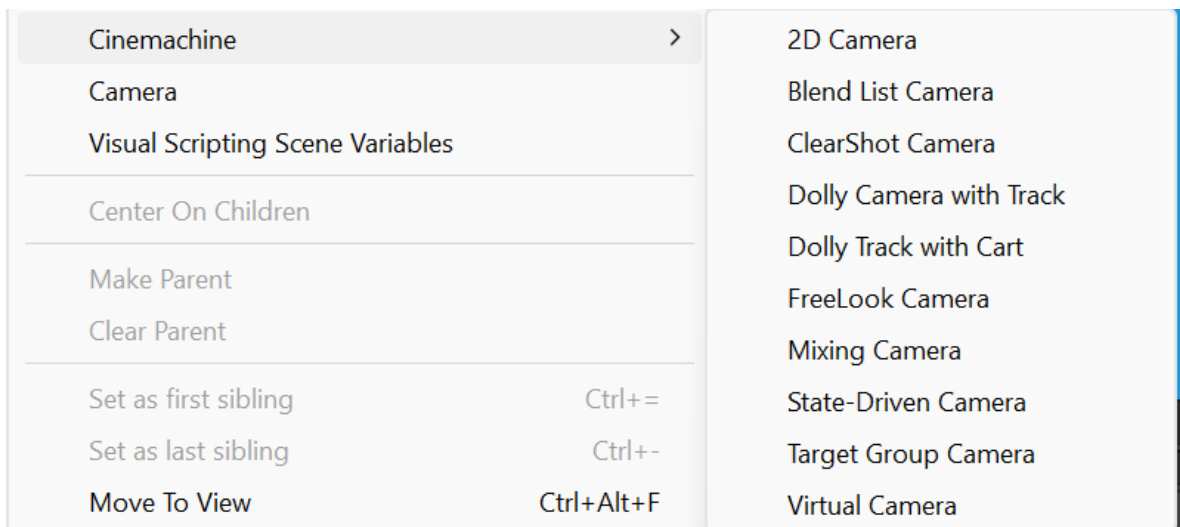
Mipmap Limit Groups

List is Empty

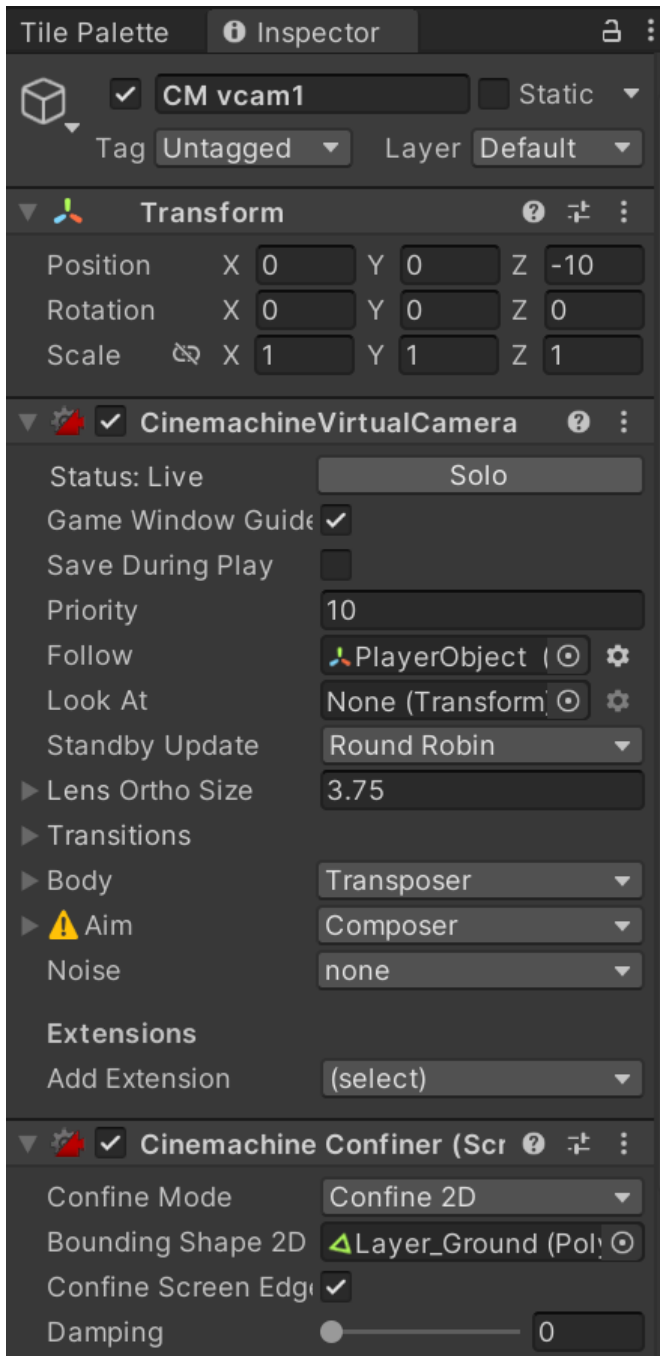


Instalación e implementación de Cinemachine. Usamos Package Manager para instalar Cinemachine y creamos una Virtual Camera que sigue automáticamente al PlayerObject, proporcionando un seguimiento suave y profesional.

Configuración de Cinemachine Confiner. Agregamos un Polygon Collider 2D al Layer_Ground y lo usamos como Bounding Shape para el Cinemachine Confiner, evitando que la cámara se salga de los límites del mapa.



Configuración de Cinemachine Confiner. Agregamos un Polygon Collider 2D al Layer_Ground y lo usamos como Bounding Shape para el Cinemachine Confiner, evitando que la cámara se salga de los límites del mapa.



Assets > Scripts



MovementControl...




RoundCameraPos

```
using UnityEngine;
using Cinemachine;

public class RoundCameraPos : CinemachineExtension
{
    public float PixelsPerUnit = 32;

    protected override void PostPipelineStageCallback(
        CinemachineVirtualCameraBase vcam,
        CinemachineCore.Stage stage, ref CameraState state,
        float deltaTime)
    {
        if (stage == CinemachineCore.Stage.Body)
        {
            Vector3 pos = state.FinalPosition;
            Vector3 pos2 = new Vector3(Round(pos.x), Round(pos.y), pos.z);
            state.PositionCorrection += pos2 - pos;
        }
    }

    float Round(float x)
    {
        return Mathf.Round(x * PixelsPerUnit) / PixelsPerUnit;
    }
}
```



☒ **Layer_Ground**

☐ 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


▶




☒ **Tilemap**

?







▼



☒ **Tilemap Renderer**

?





Sort Order

Bottom Left

Mode

Chunk

Detect Chunk Cullin

Auto

Chunk Culling Bounds

X

0

Y

0

Z


0

Mask Interaction

None

Material

☒ Sprite2D



▼

Additional Settings

Sorting Layer

Ground

Order in Layer

0

