Programación de Videojuegos  
Construyendo el mundo

Tercera Parte

Línea corta

# 

# Objetivo

* Usaremos la vista Tile Palette para dibujar nuestro entorno.
* Crearemos mapas de tiles o tiles maps qué son arreglos de sprites e imágenes para crear el entorno.
* Usaremos el componente Cinemachine para seguir al jugador.

# Introducción

Es hora de crear un mundo para los personajes a habitar. Los mundos bidimensionales (2D) a menudo son creados colocando una serie de tiles juntos para dibujar un fondo, luego colocando otros tiles en la parte superior de ese fondo para crear la ilusión de profundidad. Estos tiles en realidad son solo sprites que se han segmentado o "cortado" en dimensiones convenientes y generalmente se colocan usando una paleta de tiles. El diseñador o desarrollador puede crear varias capas de estos **Tilemaps** para crear efectos de árboles, pájaros volando por encima o incluso montañas en la distancia.

## Índice

[**Objetivo 2**](#_1wxowfp5rmhu)

[**Introducción 2**](#_fnjk6vnqzbqk)

[Índice 3](#_ju0ji1rgatg0)

[**Tilemaps & paleta Tiles 3**](#_iuhsksg1fa6g)

[**Creando Tiles Palettes 7**](#_mphhcssxltjc)

[**Dibujando con Tiles Palette 11**](#_vwo9ex9p53qq)

[**Trabajando con múltiples Tilemaps 16**](#_krylalgpkdhf)

[**Configuración de gráficos 23**](#_8ljzscvljr53)

[**La cámara 23**](#_li3tbotu3e9t)

[Usando Cinemachine 28](#_tb4sjv47uqro)

[Instalando Cinemachine en Unity 28](#_ild7f916i4mm)

[Después de Instalar Cinemachine 29](#_7vsif4vn1glg)

[**Cámaras Virtuales 29**](#_r1o6v1dzk2ro)

[**Cinemachine Confinador 34**](#_b58i20m5cpwf)

[Construyamos un Cinemachine Confiner 36](#_lkiirvr8po2j)

[**Estabilización 39**](#_1c1e4g3nq3bz)

[Materiales 40](#_m97utdww8yj)

[**Colliders y Tilemaps 41**](#_sqfxrr9fq9nc)

[Tilemap Collider 2D 41](#_5sw0xcrhieck)

[**Composite Colliders 44**](#_qbw8503gvk88)

## 

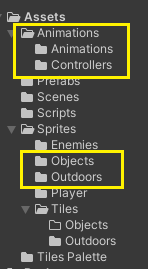
# Tilemaps & paleta Tiles

Con la introducción de la función **Tilemap**, Unity dio un paso significativo adelante en su cadena de herramientas de flujo de trabajo 2D. Unity **Tilemaps** hace que sea fácil crear niveles de forma nativa dentro del Editor de Unity, en lugar de depender de herramientas externas.

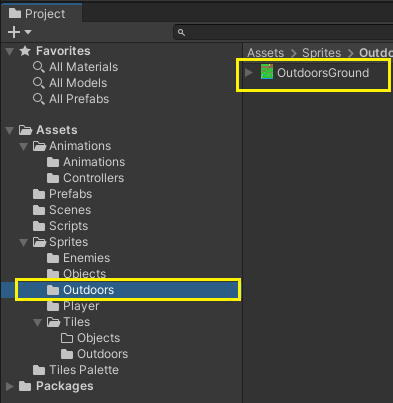
**Tilemaps**.- Son estructuras de datos que almacenan sprites en un arreglo. Para comenzar, necesitaremos importar los activos de **Tilemap**, al igual qué importamos los activos de sprites utilizados para nuestro jugador y enemigo en sesiones pasadas.

Instrucciones

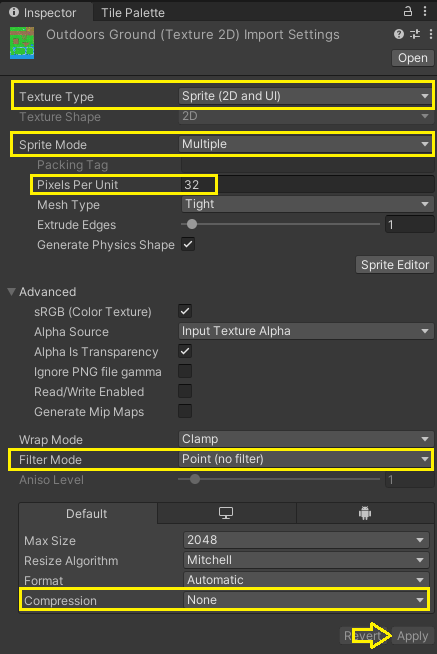
1. Crear nuevas carpetas en la carpeta Sprites tituladas: "Objects" y "Outdoors". Usaremos estas carpetas para almacenar las hojas de sprites, en parte utilizadas para nuestros **Tilemaps** externos y varios objetos que colocaremos en nuestro mundo.



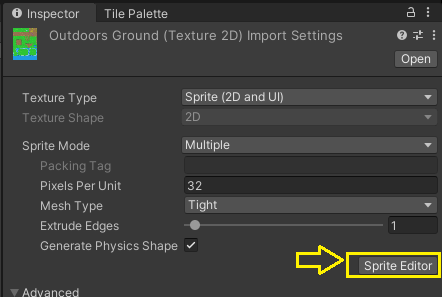
1. Encuentra la hoja de sprites titulada "*OutdoorsGround.png*" qué viene como recurso de éste tutorial. Arrastre la hoja de sprites a la carpeta Sprites ➤ Outdoors



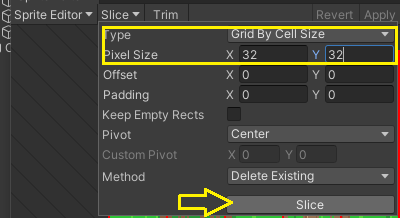
1. Establece la configuración de la imagen en la vista **Inspector** donde debe establecerse lo siguiente, una vez hecho como lo muestra la siguiente figura da clic en el botón Apply



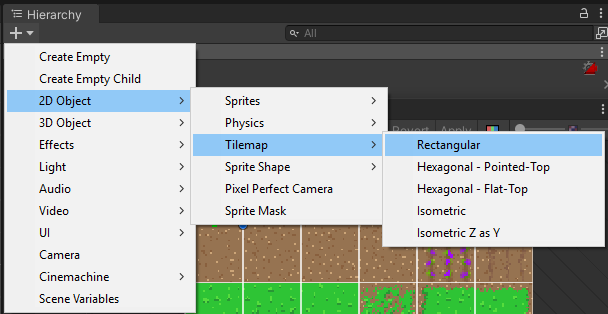
1. Ahora queremos cortar la hoja de sprites que acabamos de importar. Entrar en el Sprite Editor haciendo clic en su botón respectivo en la vista Inspector.

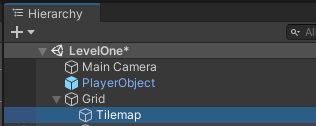


1. Presione el botón de Slice en la parte superior izquierda y luego en la opción Grid By Cell Size. Utilice 32 × 32 para el tamaño de píxel X e Y. Presione el botón Slice.



1. Verifique que las líneas de corte resultantes se vean bien y luego presione el botón Apply en la esquina superior derecha del Sprite Editor. Ahora tenemos nuestro conjunto de tiles externos.
2. A continuación, queremos crear nuestro Tilemaps. En la vista **Hierarchy**, haga clic en el botón **+** y seleccione la opción Objeto 2D ➤ Tilemap ➤ Rectangular para crear un GameObject de Tilemaps.

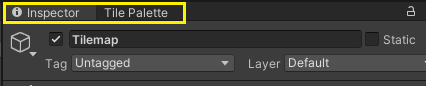




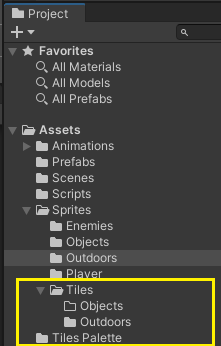
## Creando Tiles Palettes

Antes que podamos dibujar, necesitamos crear **Tilemaps**, que están hechas de tiles individuales.

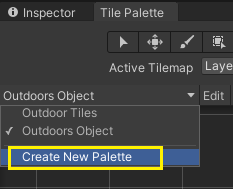
1. Vaya al menú Window ➤ 2D ➤ Tile Palette para mostrar la vista Tile Palette. Acople el panel en la misma área que la vista Inspector tal y como lo muestra la siguiente imagen.



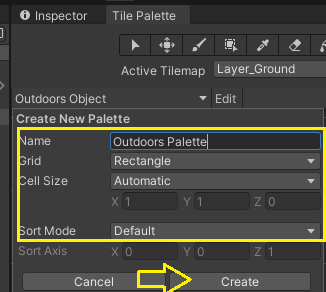
1. Queremos que el proyecto se mantenga organizado, así que crear una carpeta dentro del proyecto, dentro de la carpeta **Assets** llamada **TilePalettes**, luego cree otra carpeta llamada **Tiles** en la carpeta **Sprites**. En la carpeta **Tiles**, crea dos carpetas llamadas Outdoors y Objects. Véase la siguiente figura.



1. Seleccione el botón **Create New palette** en la ventana Tile Palette, nombrar la paleta como Outdoors Palette y deja la configuración de Cuadrícula y Tamaño de celda qué asigna por default



1. Presione el botón **Create** y guárdelo en la recién creada carpeta **TilePalette**. Esto creará un GameObject de tipo **TilePalette**.



1. Ahora, seleccione la carpeta Sprites ➤ Outdoors en la vista Project, enseguida seleccione la Vista de Tile Palette desde donde lo haya acoplado.
2. Seleccione la hoja de sprites **Outdoors** y arrástralo al área de Tile Palette hacia donde dice A*rrastra los elementos de Tile, Sprite o Sprite Texture aquí*.



1. Cuando se le solicite "Generar TILES en la carpeta", navegue hasta la carpeta Sprites ➤ Tiles ➤ Outdoor Tiles que creamos anteriormente, y presione el botón **Seleccionar carpeta**.

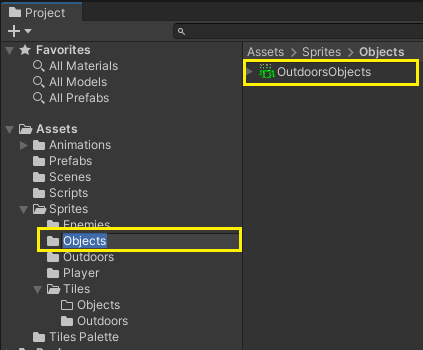


# Dibujando con Tiles Palette

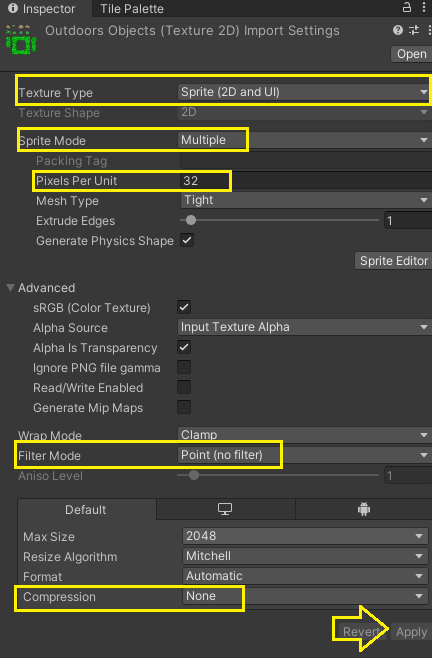
1. Seleccione la herramienta Pincel de la vista Tile Palette, luego seleccione un Tile de la vista Tile palette. Utilice el pincel para dibujar en el Tilemap en la Vista Scene.



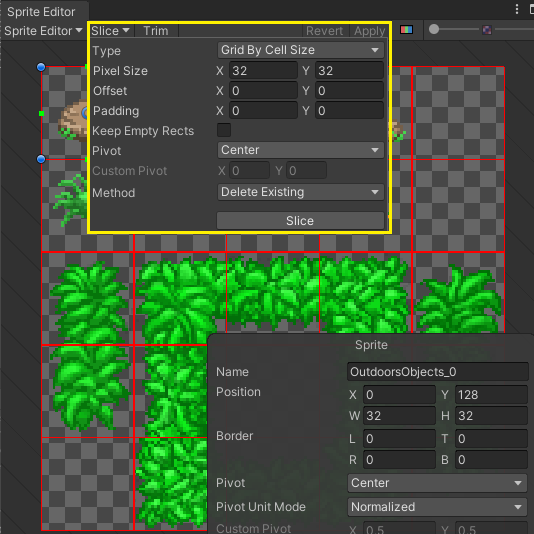
1. De los recursos que descargó para este tutorial, arrastre el archivo titulado "OutdoorsObjects.png" en la carpeta Sprites ➤ Objects.



1. Establezca las propiedades del objeto en la vista Inspector y al final presiona el botón Apply. Véase la siguiente imagen.



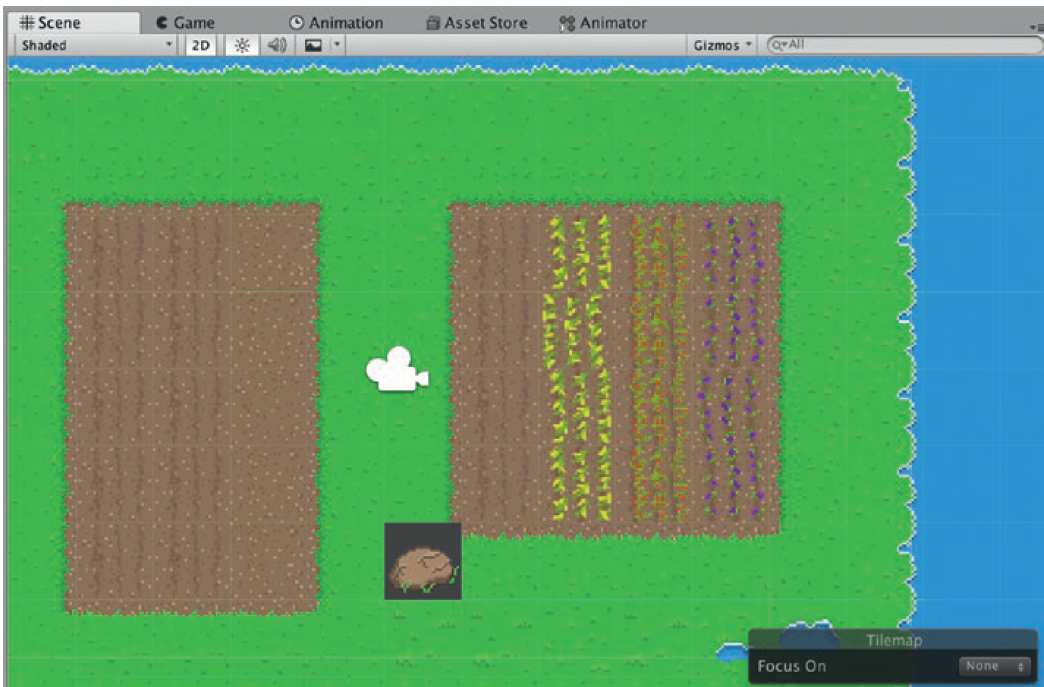
1. Ahora ingrese al Sprite Editor haciendo clic en su botón respectivo en la vista Inspector. Presione el botón Slice en la esquina superior izquierda y luego Grid by Cell Size de la opción Tipo. Utilice 32 × 32 para el tamaño de píxel X e Y.



1. Presione el botón Slice y verifique que las líneas blancas de corte resultantes se vean como si estuvieran dividiendo la hoja de sprites en las posiciones correctas. Presione el botón Apply en la esquina superior derecha del Editor de Sprite. Ahora tenemos un conjunto de sprites de objetos con temática externa para colocar en nuestra escena.
2. Ahora vamos a crear una paleta de mosaicos para dibujar con estos objetos sprites. Vuelva a nuestra Paleta **TilePalette** y seleccione Create New Palette en el menú desplegable. Nombra la nueva paleta, "Outdoors Object" y presiona el botón Create. Cuando se le solicite, guarde esta paleta en la carpeta **TilePalettes** donde guardamos nuestra Paleta de tiles para Outdoors antes.
3. Ahora haremos lo mismo que hicimos con los Outdoors Tiles: seleccionar la hoja Spritesheet y arrastrarla al área de **Tile Palette** donde dice: "Arrastra los elementos de Tile, Sprite o Sprite Texture aquí". Véase el resultado como lo muestra la siguiente imagen.



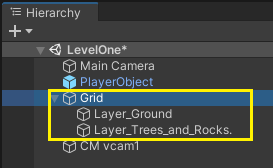
1. Cuando se le solicite "Generate Tiles into folder", navegue hasta la carpeta Sprites ➤ Tiles ➤ Objects que creamos y presiona el botón **Seleccionar carpeta**. Unity ahora generará los mosaicos en la vista Tile Palette a partir de los sprites cortados individualmente.
2. Seleccione una de las rocas de la vista Tile Palette de Outdoors Object haciendo clic y arrastrando un rectángulo alrededor de los cuatro mosaicos. Usa el pincel para colocar una sola piedra en tu Tilemaps.



Cuando pintamos los tiles de roca en el mismo **Tilemaps** como en el suelo de mosaicos, en realidad no pintamos sobre los mosaicos existentes. En cambio, nosotros reemplazamos los mosaicos existentes con mosaicos nuevos. Porque los sprites de roca qué dibujamos algunos son píxeles transparentes, podemos ver el fondo de la vista Scene. Para evitar esto, usaremos varios Timemaps y capas ordenación.

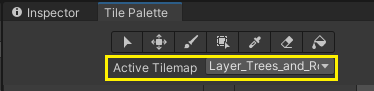
# Trabajando con múltiples Tilemaps

1. Organicemos nuestros Tilemaps. Haga clic en el objeto Tilemap en el Vista Hierarchy y cámbiale el nombre: **Layer\_Ground**.
2. Vamos a crear varios Tilemaps y apilarlos encima de cada uno sobre las capas. Haga clic con el botón derecho en el objeto Grid en la vista Hierarchy y seleccione : 2D Object ➤ Tilemap ➤ Rectangular para crear un nuevo Tilemaps. Seleccione este nuevo Tilemap y cambiale de nombre a **Layer\_Trees\_and\_Rocks**.



En este punto, si comenzara a dibujar, notará que se ha topado con él mismo problema de transparencia nuevamente. Hay dos cosas que tenemos que hacer para arreglar este problema.

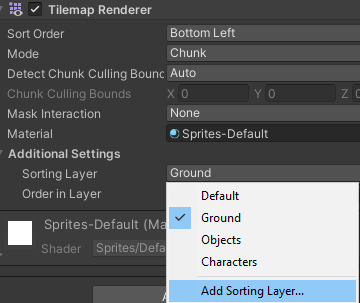
Para dibujar en un Tilemap específico, debe seleccionarse Active Tilemap en la vista Tile Palette. En la ventana Tile Palette, notará el menú desplegable de Active Tilemap. Úselo para seleccionar nuestra nueva capa **Layer\_Trees\_and\_Rocks**.

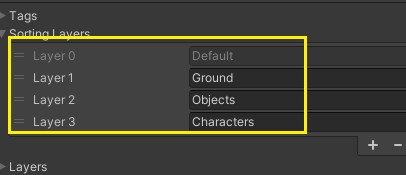


Si recuerda nuestra discusión anterior, Sprite Renderer usa Sorting Layers para determinar el orden de renderizar los sprites. Antes de que podamos dibujar en nuestro Tilemaps Layer\_Trees\_and\_Rocks, necesitamos configurar Sorting Layers para nuestros Tilemaps.

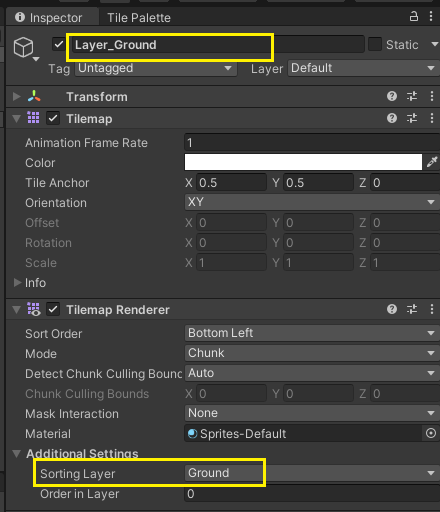
1. Seleccione Layer\_Ground y busque el componente Tilemap Renderer en la vista inspector.
2. Presione el botón Add Sorting Layer dentro del componente Tilemap Renderer y crea dos capas: llame a la primera capa "**Ground**" y a la segunda capa "**Objects**".

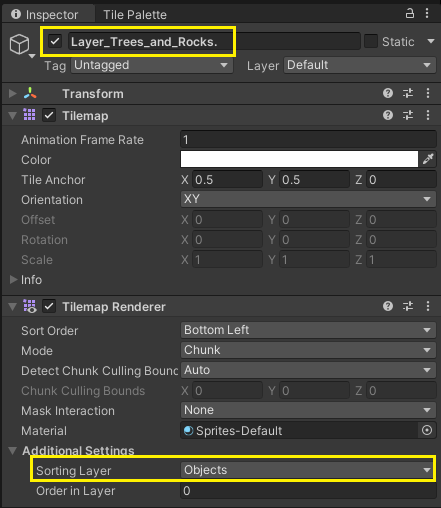
Reorganice estas Sorting Layers haciendo clic sobre ellas y arrastrarlas para qué la capa Ground esté por encima de los Objects



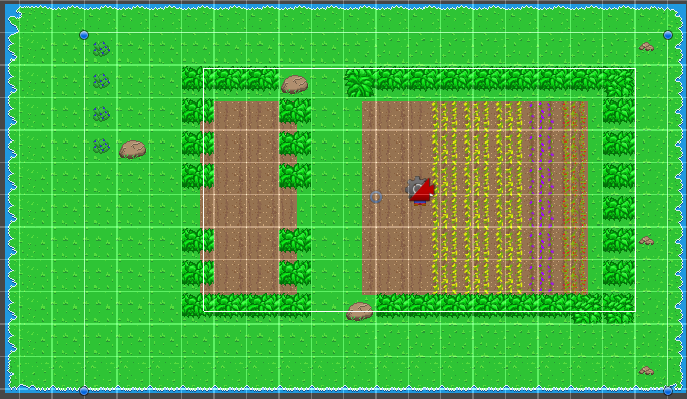


1. Seleccione el Tilemap Layer\_Ground en la vista Hierarchy nuevamente, para ver sus propiedades en la vista Inspector. En el componente Tilemap Renderer, cambie Sorting Layer a "**Ground**". Seleccione el Tilemap Layer\_Trees\_and\_Rocks y cambie Sorting Layer a "**Objects**".



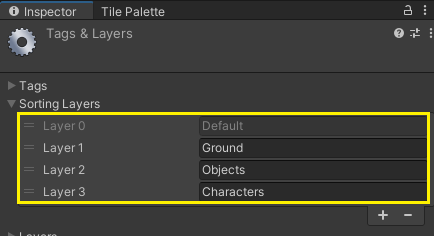


1. Elimine los tiles de roca que dibujamos anteriormente configurando la Capa activa a Layer\_Ground y, a continuación, seleccione la herramienta Borrar del conjunto de herramientas Tile Palette. También puede eliminar elementos con el pincel manteniendo presionada la tecla Shift y seleccionando cada cuadro. Rellene el lugar borrado con un poco de hierba o tile de la paleta Outdoors Objects.
2. Ahora estamos listos para dibujar, dibuja tú propio mundo. Cuando quiera dibujar tiles de tierra asegúrese de establecer el Active Tilemap qué esté configurado en Layer\_Ground y cuando desee pintar árboles, rocas y arbustos, asegúrese de que el Tilemap activo sea Layer\_Trees\_and\_Rocks.



Si queremos asegurarnos de que el Player se renderice en la parte superior del suelo y rocas. Establezcamos Sorting Layer del jugador.

1. Seleccione **PlayerObject**, luego busque la propiedad Sorting Layer en el componente Sprite Renderer Component y presione el botón Add Sorting Layer. Agregar una Sorting Layer llamada "**Caracteres**" y muévela hacia la parte inferior, después de la Capas Ground y Objects.



Ahora le hemos dicho al Sprite Renderer que renderice los objetos en orden desde la primera capa de clasificación, "Ground” hasta la última clasificación Capa "Characters".

1. Seleccione el PlayerObject y establezca su Sorting Layer a la capa **Characters** qué justo hemos creado.

Explicaremos cómo funciona la cámara más adelante, pero por ahora seleccione el objeto de la cámara y cambie la propiedad Size a 3,75.



1. Pulsa el botón Play y lleva a nuestro Player a pasear por la pequeña isla.

Notarás algunas cosas de inmediato:

* La cámara no sigue al jugador.
* El jugador puede atravesar objetos en el mapa.
* Es posible que vea algunas líneas de aspecto extraño en el Tilemap.

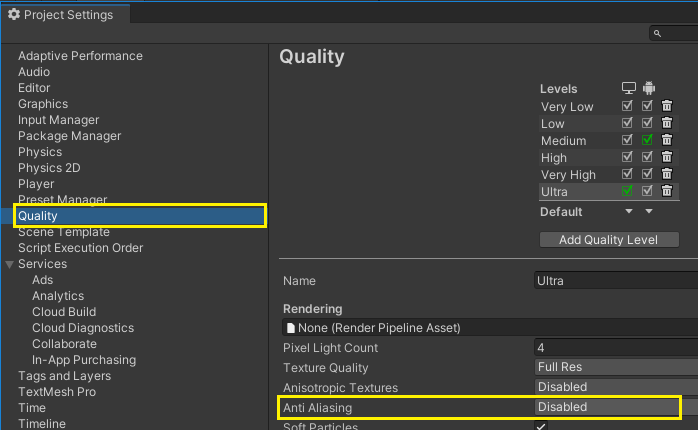
Aprenderemos a usar Colliders para evitar que el jugador camine a través de todo, y usaremos una herramienta llamada Cinemachine para hacer qué la cámara siga al jugador mientras camina.

# Configuración de gráficos

Unity usa un algoritmo llamado **Anti-Aliasing** cuando la salida de gráficos del dispositivo actual no es lo suficientemente potente para renderizar los bordes de los objetos en líneas perfectamente suaves. El algoritmo anti-aliasing corre sobre los bordes de un objeto y le da una suave apariencia para compensar la salida de gráficos irregulares.

El suavizado está activado de forma predeterminada en Unity Editor independientemente de la potencia del dispositivo que estás usando.

1. Para desactivar el suavizado, vaya a la opción Edit ➤ Projects Settings ➤ Quality ➤ Anti Aliasing como Desactivado.



1. Desde ese mismo menú Edit ➤ Project Settings ➤ Quality también desactiva Anisotropic Textures. El filtrado anisotrópico es una forma de mejorar calidad de imagen cuando se utiliza un tipo específico de perspectiva de cámara.

# 

# La cámara

Todos los proyectos 2D en Unity usan algo llamado cámara ortográfica. Las cámaras ortográficas renderizan objetos cercanos y lejanos, del mismo tamaño. Nosotros configuramos nuestro proyecto Unity para usar una cámara ortográfica en el mismo, cuando configuramos un proyecto 2D. Para obtener los mejores resultados al renderizar gráficos 2D, es importante comprender cómo funciona la cámara en un juego 2D.

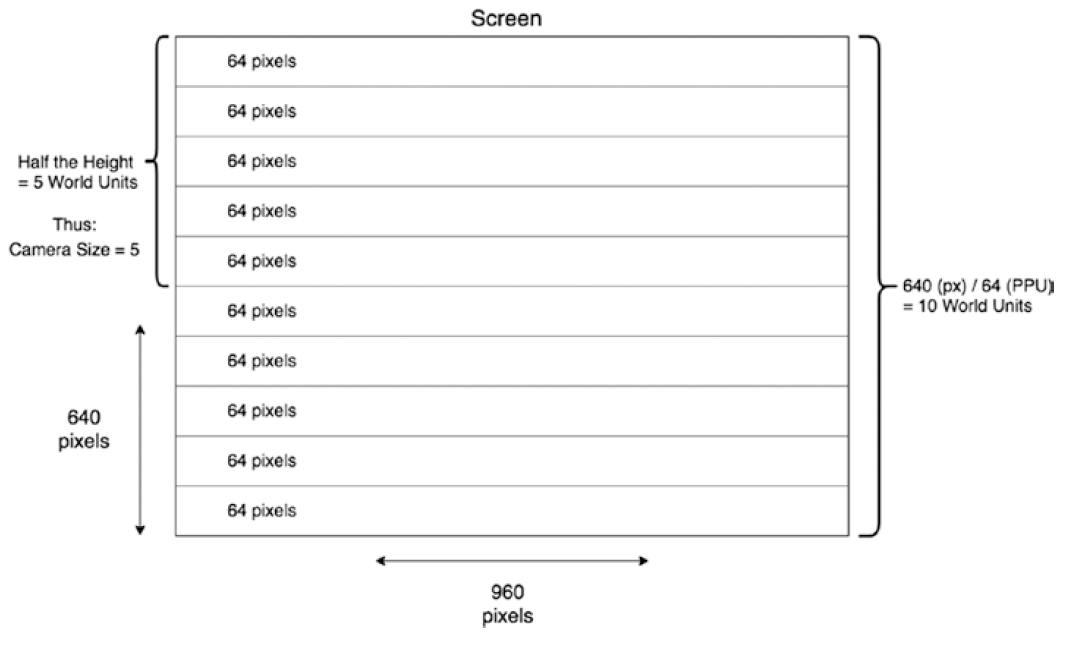
Las cámaras ortográficas tienen una propiedad llamada Size que determina cuántas "unidades de mundos verticales "pueden caber en la mitad de la altura de la pantalla. Las unidades de mundo están determinadas configurando **PPU** o Píxeles por Unidad en Unity.

La ecuación para el tamaño de la cámara es:

**(Resolución vertical / PPU) \* 0.5 = Tamaño de la cámara**

Ejemplo

Dada una resolución de pantalla de 960 × 640, la altura vertical de la pantalla es de 640 píxeles. Usemos una PPU de 64 para simplificar nuestros cálculos: 640 divididos por 64 es igual a 10. Eso significa 10 unidades de mundo apiladas una encima de la otra qué ocuparía toda la altura vertical de la pantalla y 5 unidades de mundo ocupa la mitad de la altura vertical de la pantalla. Por tanto, el tamaño de la cámara es 5.



El truco para conseguir un juego de pixel Art atractivo es prestar atención al tamaño de la cámara ortográfica con respecto a la resolución, y asegúrese de que la obra de arte se vea bien en un determinado PPU.

En nuestro juego, usaremos una resolución de 1280 × 720, pero usaremos un truco para escalar un poco el arte. Vamos a multiplicar el PPU por un factor de escala de 3.

**(Resolución vertical / (PPU \* Factor de escala)) \* 0.5 = Tamaño de la cámara**

Usando una resolución de 1280 × 720 y una PPU de 32:

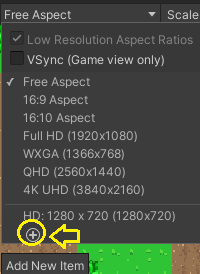
(720 / (32 PPU \* 3)) \* 0.5 = 3.75 Tamaño de la cámara

Es por eso que configuramos el tamaño de nuestra cámara en 3.75 antes.

1. Haga clic en la ventana Game y busque el menú desplegable Resolución de pantalla. De forma predeterminada, probablemente se establecerá en Aspecto libre.



1. En la parte inferior del menú desplegable, presione el signo **+** para abrir una ventana donde puede ingresar una nueva resolución. Cree una resolución personalizada de 1280 × 720.





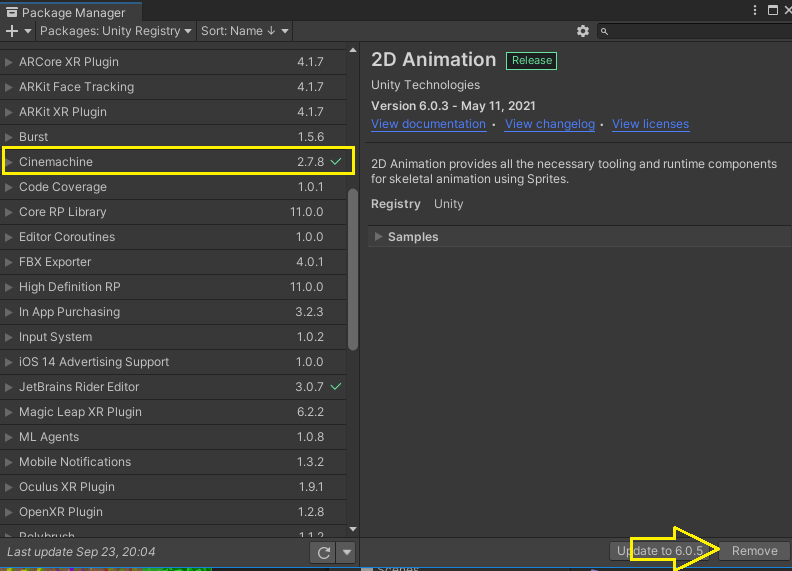
1. Presiona el botón Play y camina con el personaje por el mapa para ver nuestra nueva resolución y la cámara en acción.

## Usando Cinemachine

Cinemachine es un poderoso conjunto de herramientas de Unity para procedimientos en el juego cámaras, cinemáticas y escenas de corte. Usaremos Cinemachine para rastrear automáticamente al jugador mientras camina por el mapa.

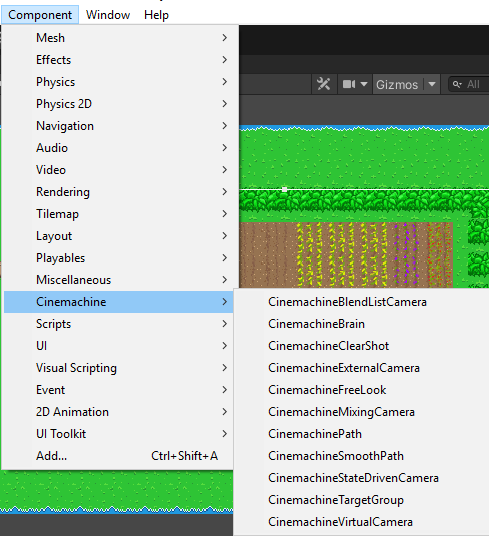
## Instalando Cinemachine en Unity

1. En los menús, seleccione Windows ➤ Package Manager aparece la ventana Unity Package Manager

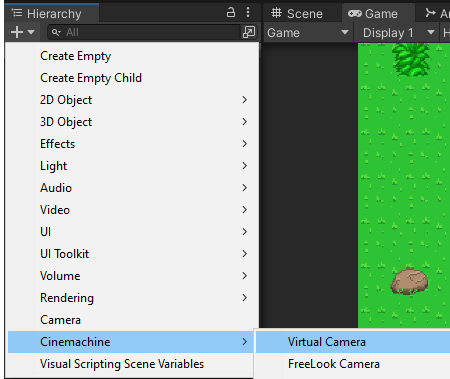


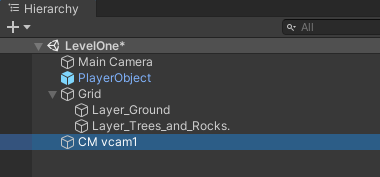
1. Haga clic en el botón Install en la parte inferior derecha para instalar Cinemachine. Una vez que Cinemachine haya terminado de instalarse, cierre la ventana de Administrador de paquetes. Debería ver una nueva carpeta de Paquetes en la vista Proyecto.

## Después de Instalar Cinemachine



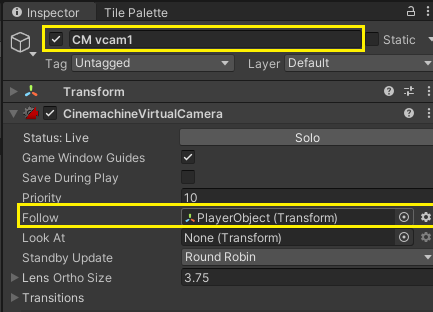
# Cámaras Virtuales

1. Seleccione de la vista Jerárquica y agregue **Cinemachine|Virtual Camera,** para ello ubique el signo **+**
2. 



Una cámara virtual puede considerarse un camarógrafo. Este camarógrafo controla la posición y la configuración de la lente de la pantalla principal de la cámara, pero en realidad no es una cámara. El Cinemachine Brain es el vínculo real entre la cámara principal y las cámaras virtuales en una escena.

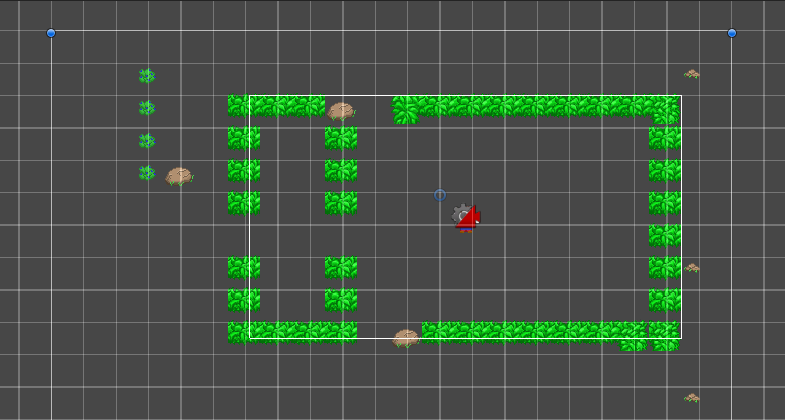
1. Seleccione la cámara virtual y arrastre el objeto **PlayerObject** a la propiedad llamado **Follow**



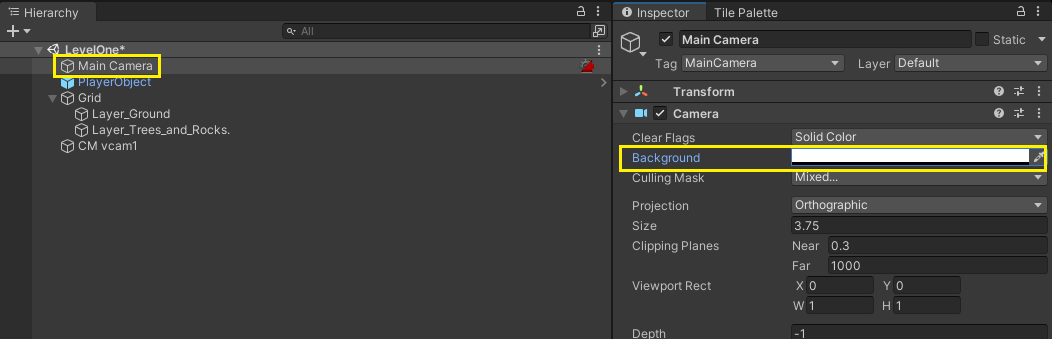
1. Pulsa Play y observa cómo la cámara sigue al jugador. ¡Con buena forma!

Para tener una mejor idea de los parámetros que rigen el movimiento de la cámara, ocultamos la capa Ground. Seleccione el objeto Layer\_Ground Tilemap de la vista Hierarchy. Desmarcar el cuadro junto al componente Tilemap Renderer de Tilemap para desactivarlo. Ahora Unity no renderiza el mapa de tiles Layer\_Ground.

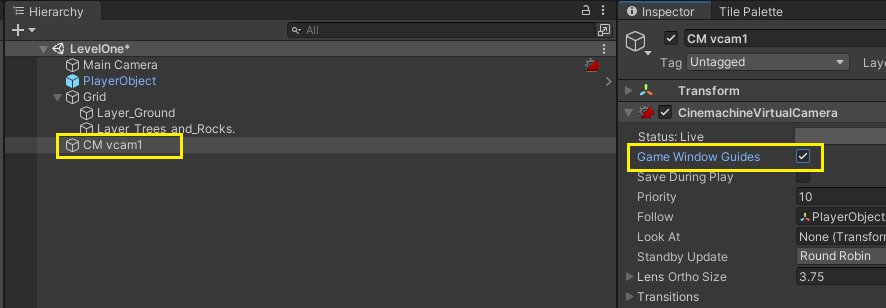




1. Ahora haga clic en el objeto **Main Camera** en la vista de Hierarchy y presione el cuadro de color que dice Fondo. Cambie el color de fondo a **blanco**.



1. Seleccione la cámara virtual y asegúrese de que la propiedad **Game Window Guides** está marcada. Experimente moviendo al jugador.



1. Detenga la reproducción y seleccione el objeto Layer\_Ground en la vista de Hierarchy. Marca la casilla a la izquierda de "**Tilemap Renderer**" para que la capa sea visible de nuevo.

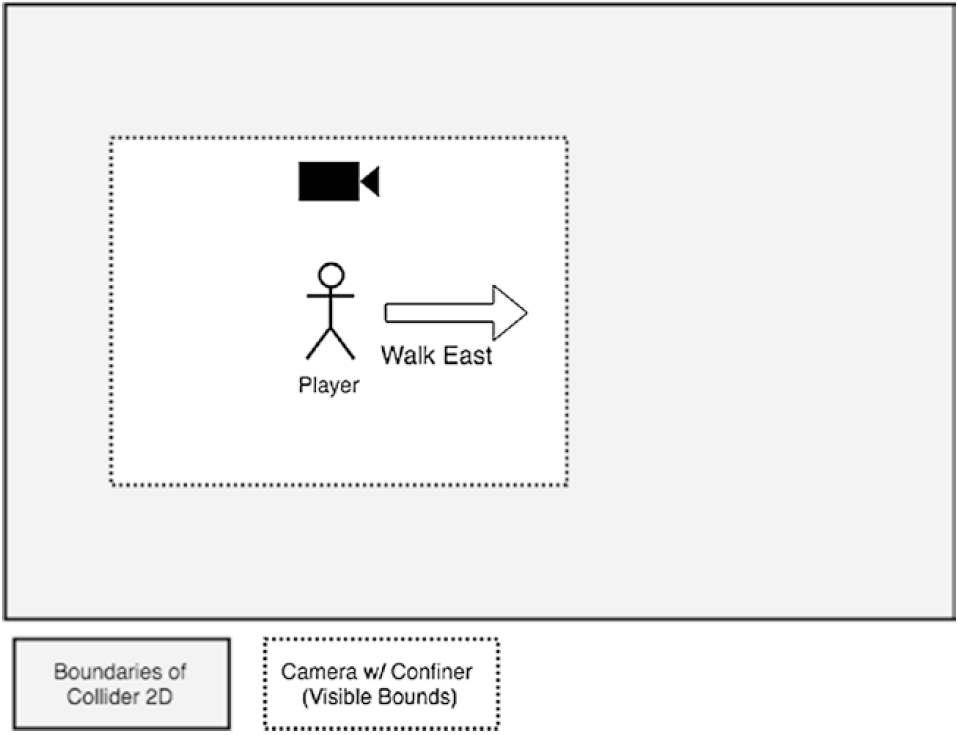
# Cinemachine Confinador

Ahora que sabemos cómo hacer que la cámara siga al jugador mientras camina alrededor, vamos a aprender cómo evitar que la cámara se mueva cuando el jugador se acerca al borde de la pantalla. Usaremos un componente llamado **Cinemachine Confiner** para confinar la cámara a un área determinada.

Tenga en cuenta que la cámara virtual en realidad está dirigiendo la cámara de escena activa, indicando dónde debe moverse y a qué velocidad.

Tenemos al jugador en una escena, a punto de caminar hacia el este.

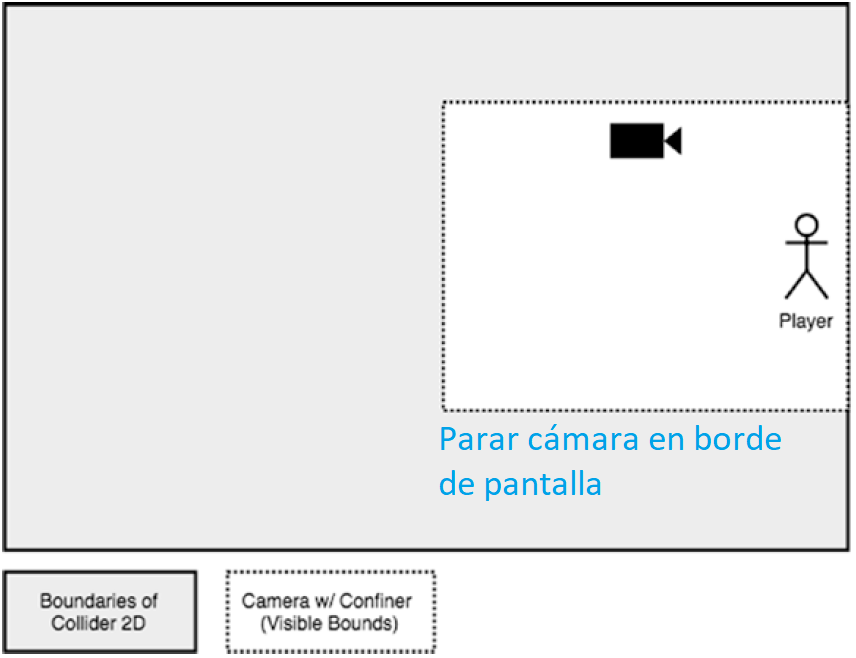
**ESCENA**



El área en blanco es la ventana de visualización visible de la cámara actualmente activa. El área en gris es el resto del mapa, fuera de la ventana gráfica de la cámara y no visible actualmente. El perímetro del área en gris está rodeado por un Collider 2D.

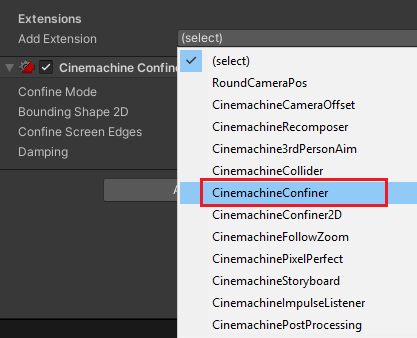
El movimiento de la cámara virtual tendrá en cuenta el movimiento del jugador, velocidad, el tamaño de la zona muerta y la cantidad de amortiguación aplicada al cuerpo de la cámara.

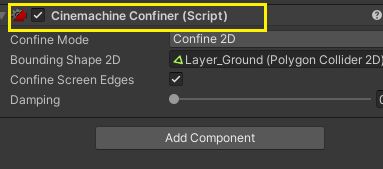
Cuando el borde de **Confiner** golpea el borde el jugador camina hacia el este de esa forma delimitadora, interactúa y le dirá a la cámara virtual que dirija la cámara activa para dejar de moverse.



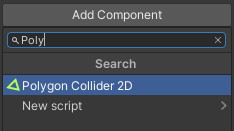
## Construyamos un Cinemachine Confiner

1. Selecciona la *cámara virtual* de la vista Hierarchy. En la ventana Inspector, junto a Add Extension, seleccione **CinemachineConfine**r en el menú desplegable menú. Esto agregará un componente Cinemachine Confiner a nuestro Cámara 2D Cinemachine

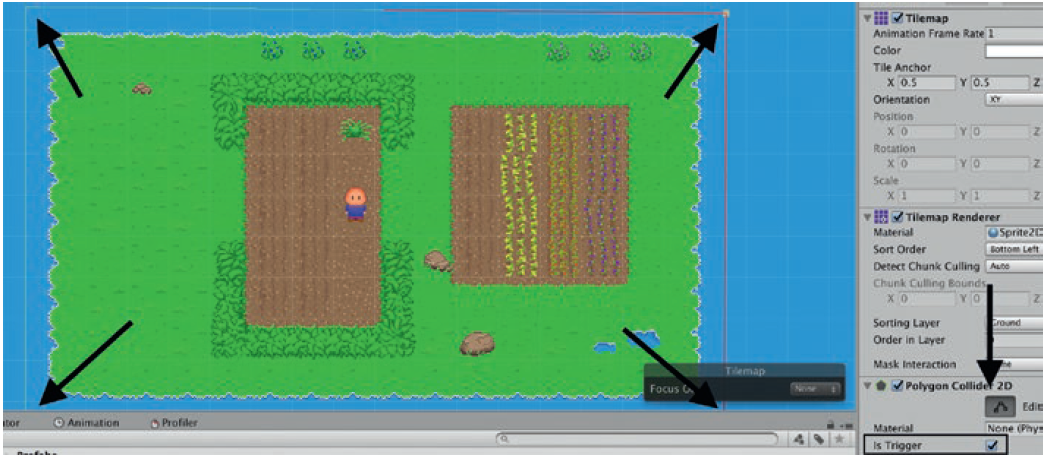




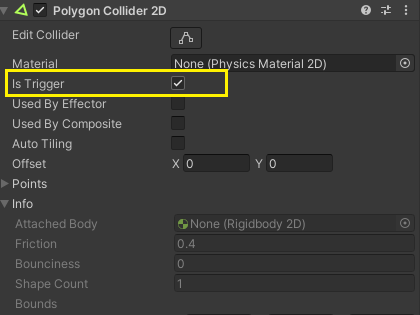
1. El **CinemachineConfiner** requiere un Composite Collider 2D, o un Polygon Collider 2D para determinar dónde están los bordes del confinamiento para empezar. Selecciona el objeto Layer\_Ground y agregue un Polygon Collider 2D.



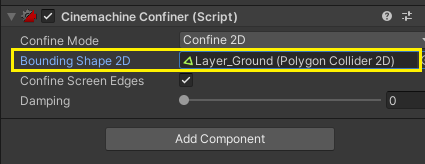
1. Haga clic en el botón Editar Collider en el componente y edite el colisionador para que delimita los bordes de nuestro nivel **Layer\_Ground**



1. Vuelva a presionar el botón Edit Collider cuando haya terminado de modificar.
2. Marque la propiedad **Is Trigger** en el componente colisionador y luego seleccione nuestra cámara **Cinemachine** nuevamente.



1. Seleccione y arrastre el objeto Layer\_Ground a la propiedad Bounding Shape 2D campo del **Cinemachine Confiner**. Asegúrate de marcar la casilla **Confine Screen Edge** para indicarle al confinador que se detenga en los bordes del polígono 2D.



1. Presiona el botón **Play** y camina hacia el borde de la pantalla. El Jugador ha caminado lejos de la zona muerta, y mientras el punto de seguimiento ha continuado moviéndose con el jugador, la cámara se ha detenido.

Para revisar, los tres pasos para configurar un Cinemachine Confiner:

1. Agregue una extensión CinemachineConfiner a la cámara virtual.

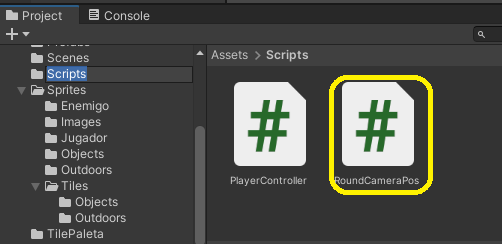
2. Cree un Polygon Collider 2D en un Tilemap edite su forma para determinar los bordes de confinamiento, y establecer la propiedad "Is Trigger".

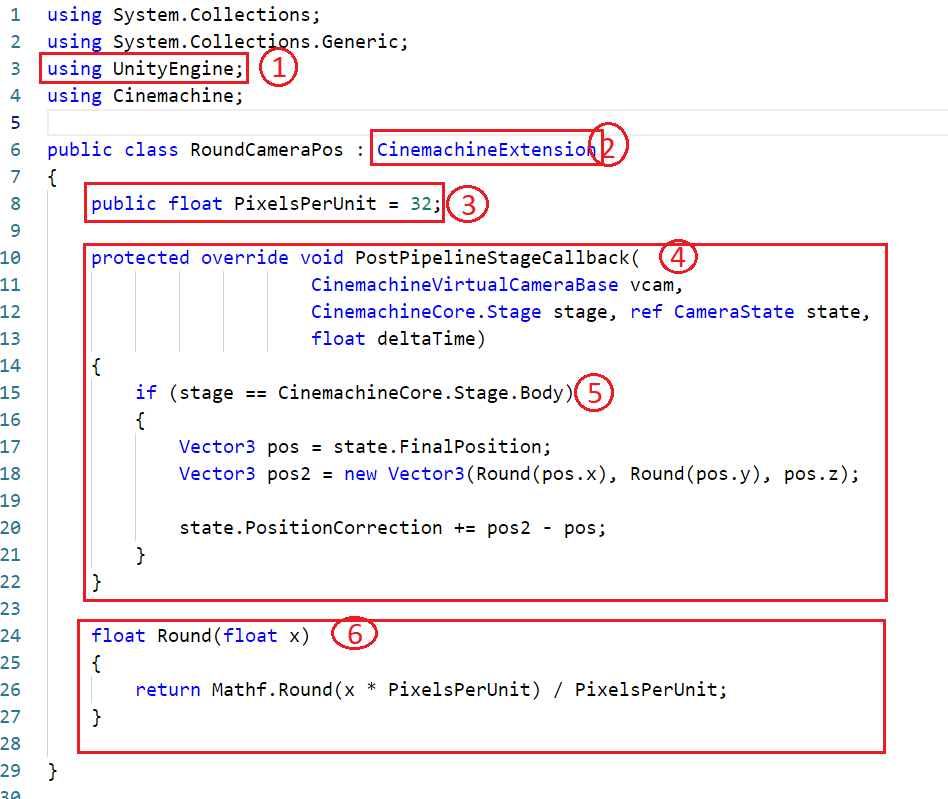
3. Utilice este Polygon Collider 2D en el campo Bounding Shape 2DCampo 2D del Cinemachine Confiner.

# Estabilización

Mientras camina con el jugador por el Tilemap, es posible que note un ligero efecto temblor . El temblor es especialmente pronunciado cuando deja de caminar y la amortiguación de la cámara virtual detiene lentamente el seguimiento.

Para solucionar este temblor, queremos forzar la posición final del Cinemachine Virtual a permanecer dentro de los límites de píxeles.

1. Checa el Script **RoundCamera.cs** de carpeta Elearning y descargalo.
2. Arrastrar el script a la carpeta **Scripts** del proyecto. 
3. Analizar el código.

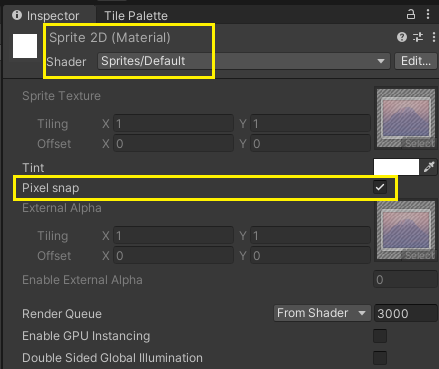


1. Selecciona la cámara virtual y agregarla como componente el script creado en el paso anterior

## Materiales

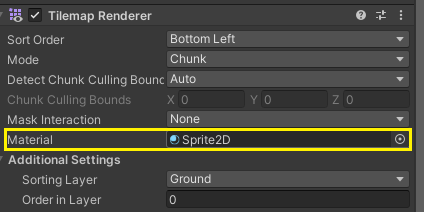
Mientras camina con el jugador por el Tilemap, es posible que observe algunas líneas o Lágrimas entre los tiles. Eso se debe a que no se están ajustando precisamente a una ubicación perfecta en píxeles. Para solucionar esto, usaremos algo llamado Material para decirle a Unity cómo queremos que se representen nuestros sprites.

1. Cree una nueva carpeta llamada **Materials**, luego haga clic con el botón derecho y Crear ➤ Material. Llame a este material, **Sprite2D**.
2. Establezca las propiedades del material véase la siguiente figura:



Queremos que el componente Renderer en nuestros GameObjects usen este material en lugar del Material predeterminado.

1. Seleccione el tilemap **Layer\_Ground** y cambie el material en el componente Tilemap Renderer haciendo clic en el punto junto a la propiedad Material.



1. Haga esto para todas nuestras capas de **Tilemap**, y luego presione el botón Play y los temblores deberían haber desaparecido.

.

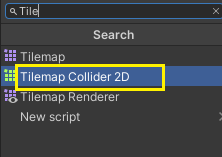
# Colliders y Tilemaps

## 

## Tilemap Collider 2D

Ahora vamos a resolver el problema en el que el jugador puede atravesar todo en el Tilemap. Hay un componente especialmente adaptado para Tilemaps llamado **Tilemap Collider 2D**. Cuando un Tilemap Collider 2D se agrega a un Tilemap, Unity detectará y agregará automáticamente un Collider 2D para cada Tile de sprite que detecta en ese Tilemap. Vamos a utilizar estos colisionadores de Tilemap para determinar cuándo el colisionador PlayerObject entra en contacto con un colisionador de Tiles y evita que el jugador camine a través de ellos.

1. Selecciona Layer\_Trees\_and\_Rocks en la vista **Hierarchy** y luego presione el botón Add Component en la vista Inspector. Busque y agregue un componente llamado "**Tilemap Collider 2D**".



Notarás que todos los sprites en tu Tilemap Layers\_Objects ahora tienen una delgada línea verde que los rodea, lo que indica un componente Collider



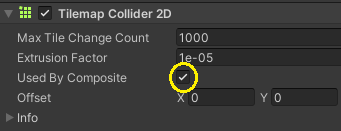
1. Ahora seleccione el Tilemap deseado: **Layer\_Trees\_and\_Rocks** en la vista **Hierarchy** y agregarle un componente Tilemap Collider 2D . Acabamos de agregar un Collider 2D a cada elemento de mosaico en Layer\_Objects. El problema con esto es que es bastante ineficaz para Unity para realizar un seguimiento de todos estos colisionadores.



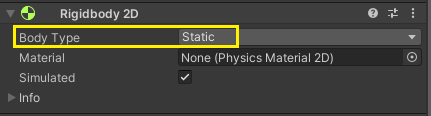
# Composite Colliders

Afortunadamente, Unity viene con una herramienta llamada Composite Collider qué combinará todos estos colisionadores separados en un colisionador grande, que es más eficiente.

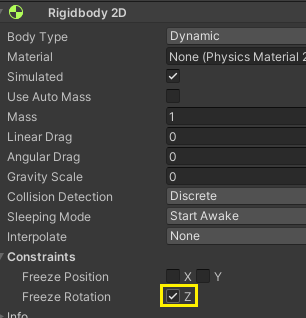
1. Mantener el Tilemap Layer\_Trees\_and\_Rocks seleccionada en la vista Hierarchy, seleccione Add Component y agregue un componente Composite Collider 2D a Layer\_Trees\_and\_Rocks. Usted puede dejar todos los ajustes predeterminados como están.
2. Ahora marque la casilla dentro del Tilemap Collider 2D que dice "Used by composite" y observe cómo todos los colisionadores de los arbustos se fusionan como por arte de magia.



1. Cuando agregamos un Composite Collider 2D a la capa Tilemap, Unity agregó automáticamente un componente Rigidbody 2D. Configurar este Rigidbody 2D Tipo de cuerpo del componente a Static porque no se moverá.



1. Antes de presionar Play, asegurémonos de que cuando el jugador choca con algo, no gira. Debido a que PlayerObject tiene un componente Dynamic Rigidbody 2D, está sujeto a fuerzas impuestas por el motor físico cuando interactúa con otros colisionadores. Seleccione PlayerObject y en el componente Rigidbody 2D adjunto, marque la casilla de verificación "Freeze Rotation Z"



1. Presiona el botón Play y lleva al jugador por el mapa.

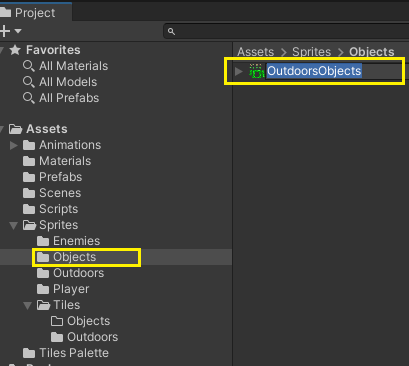
También notará que para algunos objetos hay un espacio notable entre el jugador donde se ha detenido y el objeto en el Tilemap. Para ver mejor los límites de cada colisionador, mantener el juego en funcionamiento y cambiar a la vista Scene seleccionando la pestaña Scene.

1. Acérquese al jugador con la rueda de desplazamiento del ratón o presionando la tecla F.
2. Selecciona el PlayerObject de la vista de Hierarchy para ver su Box Collider, mantén presionada la tecla **Ctrl** + selecciona TileMap Layer\_Trees\_and\_Rocks sin deseleccionar el PlayerObject.

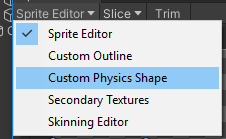


Editando la física de las figuras

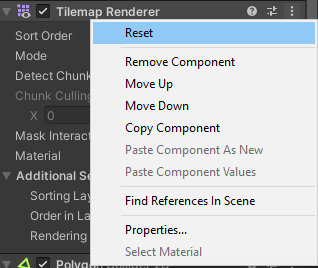
1. Para editar la forma física de los sprites en una hoja de sprites, seleccione hoja de Sprites de **Outdoors Object** en la vista Project y abra el Editor de Sprite en el Inspector.



1. Vaya al menú desplegable Sprite Editor en la parte superior izquierda y seleccione Edit Physics Shape.



Cuando estés satisfecho con la forma física, presiona el botón Apply y cierre el Editor de Sprites, presione el botón Restablecer desde el menú desplegable del icono de engranaje en el componente 2D de Tilemap Collider,



Ahora presione el botón Play y vea como los colisionadores están funcionando

Ajustar el Box Collider 2D en nuestro reproductor para que sea un poco más pequeño

