

GAMEOBJECTS Y PREFABS

Actividad 5: Creando prefabs

“Redacta un reporte de una cuartilla sobre los componentes que agregaste a tus *GameObjects* y menciona para qué funcionan.”

Al momento de utilizar los *GameObjects* es esencial agregar los componentes que quieras utilizar para que cada uno de esos objetos y así se diferencien unos de otros al momento de ejecutar tu proyecto. Los principales componentes que se utilizan en estos objetos son los *collider*, los principales son: *box collider*, *capsule collider*, entre otros que tienen la forma primitiva del objeto que debes incluir.

También hay otra gran cantidad de componentes que funcionan para la iluminación como lo son *light* y *halo*, los cuales tienen distintas formas de proyección de luz. Otros componentes pueden funcionar como cámaras que te permitan ver desde la vista del objeto que tenga configurado este elemento.

Si en algún momento necesitas que alguno de los objetos que estén en tu proyecto tenga un audio por separado o cuenten con un elemento musical puedes utilizar los componentes de audio en los cuales podrás configurarlos como tú lo necesites. Para agregar elementos de fondo en tu proyecto como decoraciones es tan sencillo como utilizar el componente de *Raw image*.

1. Capsule Collider:

El *Capsule Collider* está hecho de dos semi-esferas juntas por un cilindro. Tiene la misma forma que la primitiva *Capsule*.

2. Particle System:

Un componente del sistema de partículas simula entidades fluidas como líquidos, nubes y llamas al generar y animar un gran número de pequeñas imágenes 2D en la escena.

3. Rigidbody:

Los Rigidbodies le permite a sus GameObjects actuar bajo el control de la física. El Rigidbody puede recibir fuerza y torque para hacer que sus objetos se muevan de una manera realista. Cualquier GameObject debe contener un Rigidbody para ser influenciado por gravedad, actué debajo fuerzas agregadas vía scripting, o interactuar con otros objetos a través del motor de física NVIDIA Physx.

4. Audio Listener:

El Audio Listener actúa como un dispositivo parecido a un micrófono. Este recibe input de cualquier Audio Source dado en la escena y reproduce el sonido a través de los alta-voces del computador.

5. Video player:

Use el componente Video Player para adjuntar archivos de video a GameObjects y reproducíalos en GameObject's Texture en tiempo de ejecución.

6. Box Collider:

The Box Collider is a basic cuboid-shaped collision primitive.

7. Trail renderer:

El componente Trail Renderer representa un rastro de polígonos detrás de un GameObject en movimiento. Esto se puede usar para dar una sensación de movimiento enfatizada a un objeto en movimiento, o para resaltar la ruta o la posición de los objetos en movimiento.

8. Character Controller:

El personaje en un juego de primera o tercera persona va a necesitar físicas basadas en colisiones para que no caiga a través del suelo o camine a través las paredes. Sin embargo, usualmente la aceleración del personaje y el movimiento no serán físicamente reales, por lo que podría acelerar, frenar y cambiar de dirección casi instantaneamente sin ser afectado por el momento.

9. Audio Source:

El Audio Source reproduce un Audio Clip en la escena. El clip se puede reproducir en un audio listener o a través de un audio mixer. El audio source puede reproducir cualquier tipo de Audio Clip y puede ser configurado que reproduzca estos como 2D, 3D, o como una mezcla (SpatialBlend). El audio se puede esparcir entre los altavoces (estéreo a 7.1)(Spread) y transformado entre 3D y 2D (SpatialBlend).

10. Camera:

Las Cameras son los dispositivos que capturan y muestran el mundo al jugador. Al personalizar y manipular las cámaras, usted hace la presentación de su juego de verdad única. Usted puede tener un número ilimitado de cámaras en la escena. Ellas pueden ser establecidas para renderizar en cualquier orden, en cualquier lugar de la pantalla, o en solo ciertas partes de la pantalla.

11. Streaming Controller:

El componente Streaming Controller es parte del sistema Texture Streaming. Agregue un controlador de transmisión a GameObjects si desea usar Texture Streaming para cargar sus mapas mip en Unity en tiempo de ejecución.

12. Rect Transform:

El componente Rect Transform es la contraparte del diseño 2D del componente Transform. Donde el Transform representa un solo punto, un Rect Transform representa un rectángulo donde un elemento UI puede ser colocado en su interior. Si el padre del Rect Transform también es un Rect Transform, el Rect Transform hijo puede también especificar cómo va a ser posicionado y va a cambiar de tamaño relativo al rectángulo padre.

13. Line Renderer:

El componente Line Renderer toma una matriz de dos o más puntos en el espacio 3D y dibuja una línea recta entre cada uno. Puede usar un renderizador de líneas para dibujar cualquier cosa, desde una simple línea recta hasta una espiral compleja.

14. Audio Reverb Zone:

El Audio Reverb Filter coge un Audio Clip y lo distorsiona para crear un efecto reverb personalizado.

15. Tilemap:

El componente Tilemap es un sistema que almacena y maneja Tile Assets para crear niveles 2D. Transfiere la información requerida de los mosaicos colocados en él a otros componentes relacionados, como Tilemap Renderer y Tilemap Collider 2D.

16. Skybox:

Un skybox es un cubo con una textura diferente en cada cara. Cuando usa un skybox para renderizar un cielo, Unity esencialmente coloca su escena dentro del cubo del skybox. Unity renderiza primero el skybox, por lo que el cielo siempre se renderiza en la parte posterior.

17. Sphere Collider:

El Sphere Collider es una primitiva de colisión básica con la forma de una esfera.

18. Lens Flare:

Los Lens Flares simulan el efecto de las luces que se refractan dentro de la lente de la cámara. Se utilizan para representar luces realmente brillantes o, más sutilmente sólo para añadir un poco más de ambiente a su escena.

19. Light:

Lightmapping es el proceso de calcular previamente el brillo de las superficies en una escena y almacenar el resultado en una textura llamada mapa de luz para su uso posterior.

20. Rotation constrain:

Un componente de restricción de rotación rota un GameObject para que coincida con la rotación de sus GameObjects de origen.

21. Mesh Collider:

The Mesh Collider takes a Mesh Asset and builds its Collider based on that Mesh. It is more accurate for collision detection than using primitives for complicated Meshes. Mesh Colliders that are marked as Convex can collide with other Mesh Colliders.

22. Halo:

Los Halos son áreas claras alrededor de fuentes de luz, usadas para dar la impresión de pequeñas partículas de polvo en el aire.

23. Layout element:

Si desea anular el tamaño mínimo, preferido o flexible de un elemento de diseño, puede hacerlo agregando un componente Elemento de diseño al GameObject.

24. Canvas renderer:

El componente Canvas Renderer renderiza un objeto UI gráfico contenido dentro de un Canvas.

25. Raw image:

El control Raw Image muestra una imagen no interactiva al usuario. Puede usar esto para fines tales como decoraciones o íconos, y puede cambiar la imagen de una secuencia de comandos para reflejar cambios en otros controles. El control es similar al control Imagen, pero ofrece más opciones para animar la imagen y llenar con precisión el rectángulo de control.

Technologies, U. (2019). Unity User Manual (2019.4 LTS) - Unity Manual.
Unity. <https://docs.unity3d.com/es/2019.4/Manual/index.html>