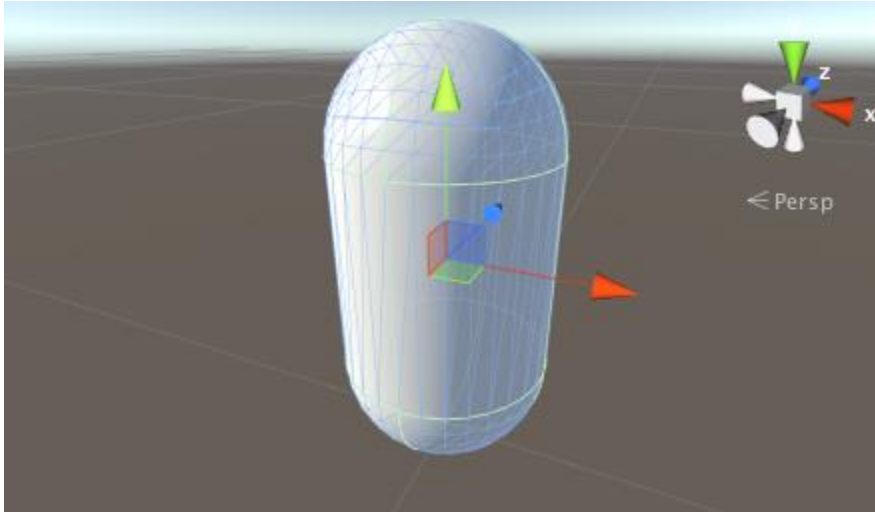


| | | |
|---------------------------------------------------------------------|-----------------------------------------------------------|------------------------------|
| Nombre: Mauricio Bennett Alcantar Domínguez | | Matrícula: 2938988 |
| Nombre del curso: Programación y prototipo de videojuegos | Nombre de la profesora: Claudia Lorenzo Muradas | |
| Tema 5: GameObjects y Prefabs | Actividad 5: Creando prefabs | |
| Fecha: 20 de febrero de 2022 | | |

Referencias

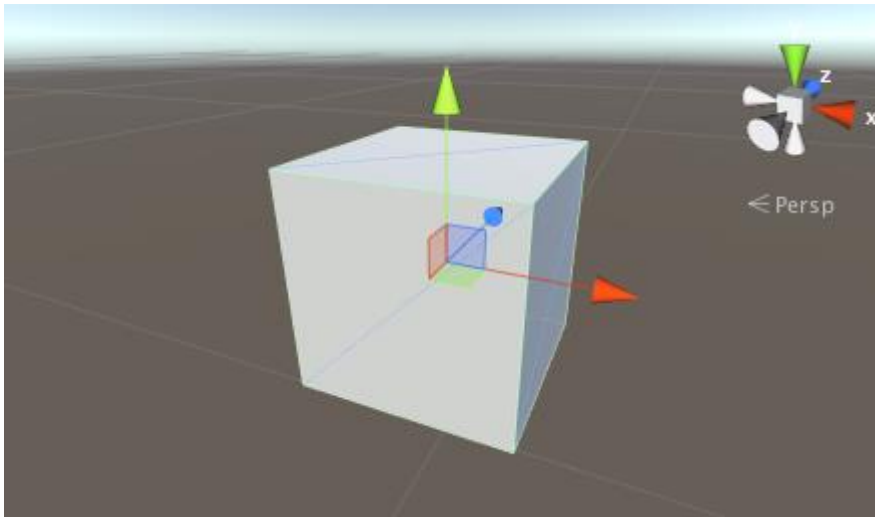
Unity. (2016). Usando Components. 2022, de Unity Sitio web:
<https://docs.unity3d.com/es/530/Manual/UsingComponents.html>

Componentes en Unity



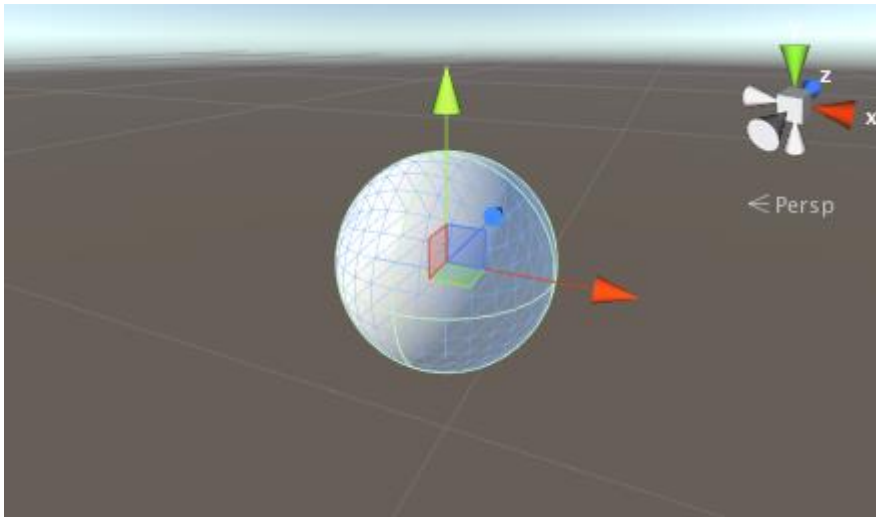
Componentes utilizados en la capsula.

1. **Audio Source (Fuente de Audio):** 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).
2. **Audio Chorus Filter:** El Audio Chorus Filter toma un Audio Clip y lo procesa creando un efecto de coro.
3. **Audio Distortion Filter:** El Audio Distortion Filter distorsiona el sonido de un AudioSource o de sonidos alcanzando el AudioListener.
4. **Audio Echo Filter:** El Audio Echo Filter repite un sonido después de un Delay dado, atenuando las repeticiones basándose en el Decay Ratio.
5. **Audio High Pass Filter:** El Audio High Pass Filter pasa altas frecuencias de un AudioSource y corta las señales de las frecuencias menores del Cutoff Frequency.



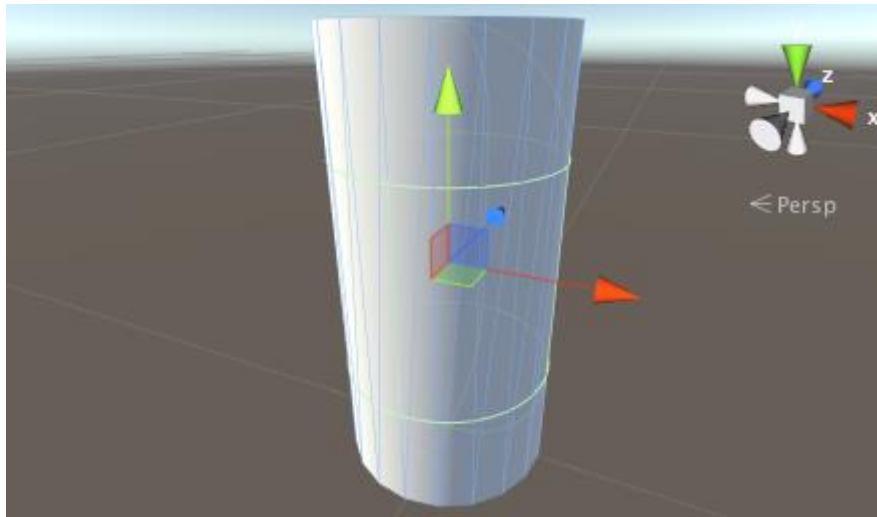
Componentes utilizados en el cubo.

1. **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 altavoces del computador.
2. **Audio Low Pass Filter:** El Audio Low Pass Filter pasa frecuencias bajas de un AudioSource o todo el sonido alcanzando un AudioListener mientras se quitan las frecuencias mayores que el Cutoff Frequency.
3. **Audio Reverb Filter:** El Audio Reverb Filter coge un Audio Clip y lo distorsiona para crear un efecto reverb personalizado.
4. **Reverb Zones:** Las Reverb Zones toman un Audio Clip y lo distorsionan dependiendo de dónde el audio listener esté ubicado dentro de la zona de reverberación(reverb zone).
5. **Halo:** Los Halos son áreas claras alrededor de fuentes de luz, usadas para dar la impresión de pequeñas partículas de polvo en el aire.



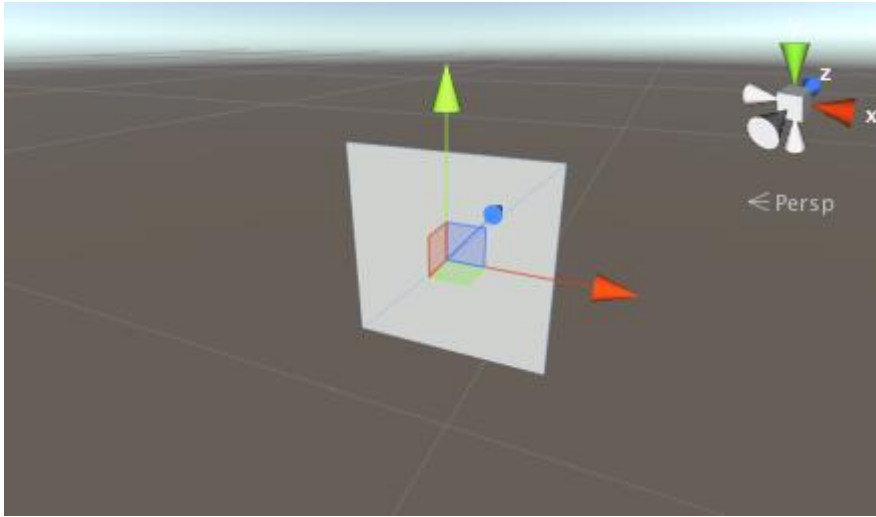
Componentes utilizados en la esfera.

1. **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.
2. **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 utilizar un renderizador de líneas para dibujar cualquier cosa, desde una simple línea recta hasta una espiral compleja.
3. **Particle System:** El componente Particle System simula entidades fluidas como líquidos, nubes y llamas generando y animando un gran número de pequeñas imágenes 2D en la escena.
4. **Projector:** Un proyector le permite proyectar un material en todos los objetos que cruzan su frustum.
5. **Trail Renderer:** El componente Trail Renderer renderiza 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 trayectoria o la posición de los objetos en movimiento. Un rastro detrás de un proyectil agrega claridad visual a su trayectoria; Las estelas de la punta de las alas de un avión son un ejemplo de un efecto de rastro que ocurre en la vida real.



Componentes utilizados en el cilindro.

1. **Visual Effect:** El componente Visual Effect contiene controles básicos para configurar una instancia de un efecto visual en una escena.
2. **Event System:** Se encarga de controlar todos los demás elementos que componen el eventing. Coordina qué módulo de entrada está actualmente activo, qué GameObject se considera actualmente 'seleccionado' y una serie de otros conceptos de sistema de eventos de alto nivel.
3. **Event Trigger:** Recibe eventos del sistema de eventos y llama a las funciones registradas para cada evento.
4. **Aspect Ratio Fitter:** Funciona como un controlador de diseño que controla el tamaño de su propio elemento de diseño. Puede ajustar la altura para que se ajuste al ancho o viceversa, o puede hacer que el elemento quepa dentro de su padre o envolver a su padre. El Aspect Ratio Fitter no tiene en cuenta la información de diseño, como el tamaño mínimo y el tamaño preferido.
5. **Grid Layout Group:** Coloca sus elementos de diseño secundarios en una cuadrícula.



Componentes utilizados en el Quad.

1. **Mesh Filter:** El Mesh Filter toma un mesh de sus assets y los pasa al Mesh Renderer para que sea renderizado en la pantalla.
2. **Mesh Renderer:** El Mesh Renderer toma la geometría del filtro de malla y la representa en la posición definida por el componente Transformar del GameObject.
3. **Mesh Collider:** El Mesh Collider toma un Mesh Asset y construye su Collider basado en esa Mesh. Es más preciso para la detección de colisiones que el uso de primitivas para mallas complicadas. Los Colisionadores de Malla que están marcados como Convexos pueden colisionar con otros Colisionadores de Malla.
4. **Animation:** Este componente era usado con fines de animación antes de la introducción del sistema de animación actual de Unity.
5. **Terrain:** El Editor de Unity incluye un conjunto integrado de características de Terrain que te permiten agregar paisajes a tu juego. En el Editor, puede crear varias fichas de terreno, ajustar la altura o la apariencia de su paisaje y agregarle árboles o césped. En tiempo de ejecución, Unity optimiza el renderizado Terrain integrado para mayor eficiencia.