

«Talento Tech»

Desarrollo de Videojuegos

Unity 3D

Clase 14



Clase N° 14 | Particulas

Temario:

- Sistema de partículas.
 - Propiedades básicas del Sistema de Partículas.
 - Aplicaciones en VJ.
 - Ejemplos de uso.
-

Objetivos de la clase

En esta clase, los estudiantes explorarán el **Sistema de Partículas** en Unity, comprendiendo sus propiedades básicas y su aplicación en el desarrollo de videojuegos. A través de ejemplos prácticos, aprenderán a crear efectos visuales dinámicos y analizarán sus diversas implementaciones para mejorar la estética y la interacción en entornos virtuales.

¿Qué son los sistemas de partículas?

Los sistemas de partículas en Unity permiten simular efectos visuales dinámicos y realistas, como humo, fuego, lluvia, chispas y explosiones. A diferencia de los modelos 3D tradicionales, que son estáticos, las partículas están diseñadas para ser pequeñas imágenes (sprites) o meshes que se generan, se mueven y desaparecen con el tiempo.

Unity proporciona el **Particle System**, una herramienta poderosa y versátil que permite configurar y personalizar efectos de partículas con gran detalle.

Usos en videojuegos

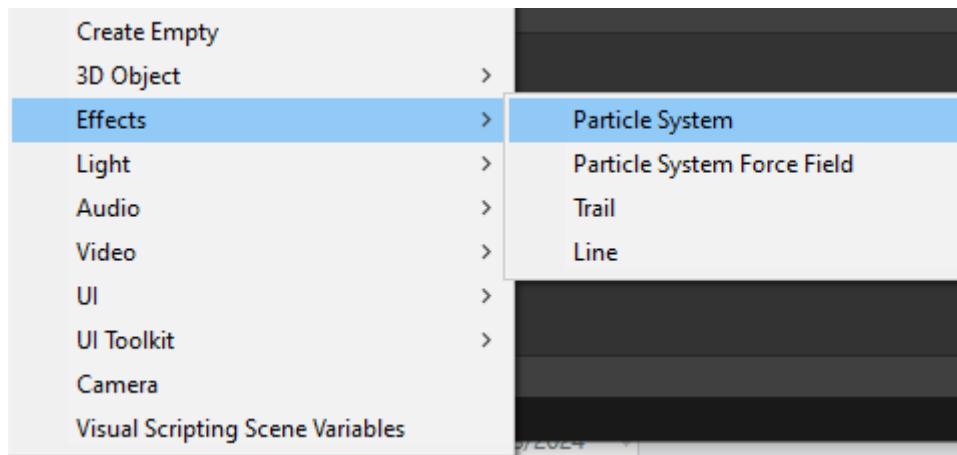
Los sistemas de partículas se utilizan en una variedad de situaciones para mejorar la inmersión y la estética de un juego. Algunos ejemplos comunes incluyen:

- **Fuego** 🔥 → Simulación de antorchas, fogatas, explosiones.
- **Humo** ☁️ → Efectos de combustión, daño en vehículos, escapes de vapor.
- **Explosiones** 💣 → Impactos de proyectiles, granadas, destrucción de objetos.
- **Lluvia/Nieve** 🌧️ ❄️ → Clima dinámico, ambientación en escenas.
- **Efectos mágicos** ✨ → Hechizos, auras, escudos de energía, partículas flotantes.
- **Impactos de proyectiles** 💥 → Balas golpeando superficies, chispas en metales.
- **Portales o teletransportación** 🌀 → Apariciones y desapariciones de personajes u objetos.

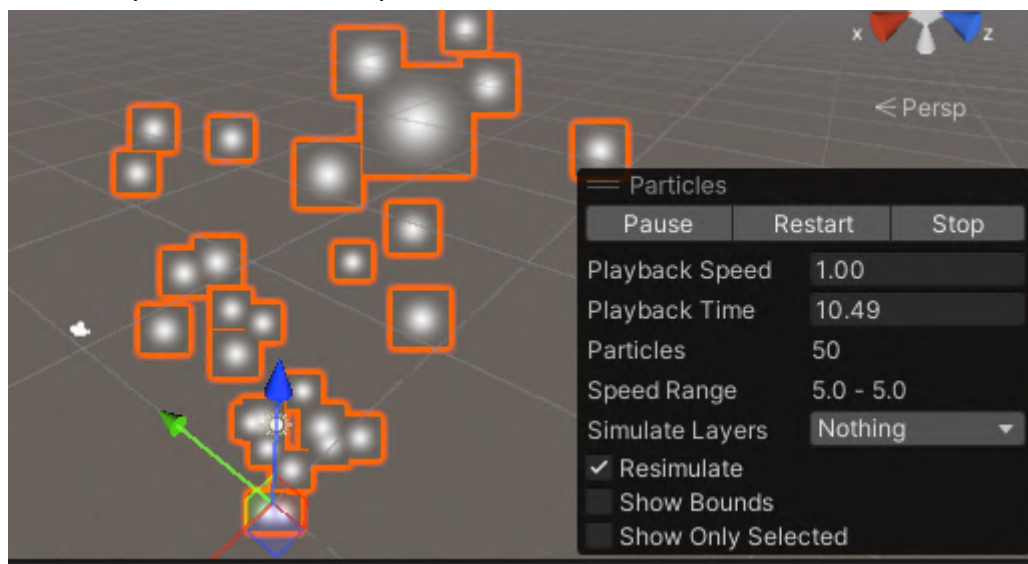
Estos efectos contribuyen a mejorar la experiencia del jugador al agregar realismo, impacto visual y feedback a las acciones dentro del juego.

Bases del Sistema de Partículas

Primero iremos a la jerarquía -> Click Derecho -> Effects -> Particle System



Esto creará unas partículas básicas que se producirán para arriba y en un cuadro a la derecha nos empezará a mostrar opciones sobre su ciclo de vida



1. Playback Speed

- Controla la velocidad de reproducción de la simulación.
- Un valor mayor a **1** acelera las partículas, y un valor menor a **1** las ralentiza.
- Útil para ver cómo se comporta el efecto a diferentes velocidades.

2. Playback Time

- Muestra el tiempo actual de la simulación en segundos.
- Puedes modificar este valor para avanzar o retroceder la simulación manualmente.

3. Particles

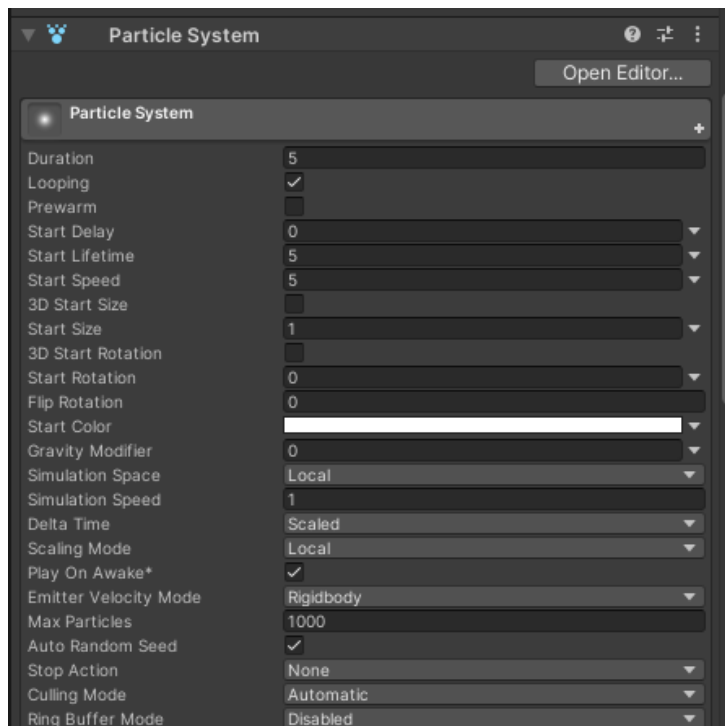
- Indica la cantidad actual de partículas activas en la escena.
- Sirve para monitorear el rendimiento y asegurarte de que el sistema de partículas no genera demasiadas partículas simultáneamente.

4. **Speed Range**
 - Muestra el rango de velocidades de las partículas activas en el sistema.
 - Permite verificar la variabilidad de la velocidad entre partículas.
5. **Resimulate (Checkbox)**
 - Si está activado, cualquier cambio en los parámetros del sistema de partículas se aplicará en tiempo real sin necesidad de reiniciar la simulación.
 - Es útil para ajustar efectos sin tener que reiniciar el sistema constantemente.
6. **Playback Controls (Botones de Play, Pause, Stop)**
 - **Play** ▶ → Inicia la simulación del sistema de partículas.
 - **Pause** ⏸ → Pausa la simulación en el estado actual.
 - **Stop** ■ → Detiene la simulación y reinicia el sistema.

Inspector y variables

Si vemos el Inspector al seleccionar el Sistema de Partículas, veremos lo siguiente:

Main



Opciones a modificar:



Main

Dentro del main nos encontramos con las variables principales del sistema.

1. Duration

- Define la duración total del sistema de partículas en segundos.
- Si el sistema no está en modo "Looping", se detendrá después de este tiempo.
- Ejemplo: Si **Duration = 5**, las partículas se reproducirán por 5 segundos antes de detenerse.

2. Looping

- Activa o desactiva la repetición automática del sistema de partículas.
- Si está activado, el sistema de partículas se reiniciará automáticamente al finalizar la duración.

3. Prewarm

- Solo funciona si **Looping** está activado.
- Permite que el sistema de partículas comience con un estado avanzado, simulando que ha estado funcionando desde antes de que aparezca en pantalla.
- Útil para efectos como fuego o lluvia que deberían estar activos desde el inicio del juego.

4. Start Delay

- Especifica un tiempo de espera antes de que comience la emisión de partículas.
- Puede ser un valor fijo o un rango aleatorio.
- Ejemplo: Si **Start Delay = 2**, las partículas comenzarán a generarse 2 segundos después de activarse el sistema.

5. Start Lifetime

- Define cuánto tiempo vive cada partícula antes de desaparecer.
- Puede ser un valor fijo o un rango aleatorio.
- Ejemplo: Si **Start Lifetime = 3**, cada partícula se extinguirá después de 3 segundos.

6. Start Speed

- Define la velocidad inicial de las partículas cuando se generan.
- Puede ser un valor fijo o un rango aleatorio.
- Ejemplo: **Start Speed = 5** hará que las partículas se muevan rápidamente al nacer.

7. 3D Start Size

- Si está desactivado, todas las partículas tendrán el mismo tamaño en X, Y y Z.
- Si está activado, puedes establecer diferentes tamaños en cada eje.

8. Start Size

- Define el tamaño inicial de las partículas.
- Puede ser un valor fijo o un rango aleatorio.
- Ejemplo: Si **Start Size = 2**, las partículas serán el doble de su tamaño normal.

9. 3D Start Rotation

- Si está activado, permite rotaciones diferentes en los ejes X, Y y Z.

10. Start Rotation

- Define la rotación inicial de las partículas en grados.
- Puede ser un valor fijo o aleatorio.

11. Flip Rotation

- Invierte la rotación de las partículas en un porcentaje determinado (0 a 1).
- Útil para variar la apariencia cuando se usan sprites o texturas.

12. Start Color

- Define el color inicial de las partículas.
- Puede ser un color fijo o un rango de colores.

13. Gravity Modifier

- Ajusta la influencia de la gravedad sobre las partículas.
- **0** significa que no hay efecto de gravedad.
- **1** usa la gravedad estándar de Unity.
- Valores negativos hacen que las partículas floten hacia arriba.

14. Simulation Space

- Define el espacio de simulación de las partículas:
 - **Local** → Las partículas se mueven con el sistema (si el objeto se mueve, las partículas se mueven con él).
 - **World** → Las partículas se mueven libremente en el mundo, sin importar el movimiento del sistema.
 - **Custom** → Permite usar otro objeto como referencia para la simulación.

15. Simulation Speed

- Modifica la velocidad global del sistema de partículas.
- Un valor mayor a **1** acelera las partículas, y un valor menor a **1** las ralentiza.

16. Delta Time

- Define cómo se calculan los pasos de tiempo de la simulación:
 - **Scaled** (por defecto) → Usa el tiempo normal del juego (afectado por `Time.timeScale`).
 - **Unscaled** → No se ve afectado por cambios en `Time.timeScale` (útil para partículas que deben seguir activas en pausas).

17. Scaling Mode

- Define cómo se escala el sistema de partículas:
 - **Hierarchy** → Se escala con el objeto padre.
 - **Local** → Solo afecta el tamaño de las partículas sin modificar la velocidad.
 - **Shape** → Se escala en base a la forma del emisor.

18. Play On Awake

- Si está activado, el sistema de partículas se reproduce automáticamente al iniciar la escena.
- Si está desactivado, necesitas activarlo manualmente desde un script o con el método `Play()`.

19. Emitter Velocity Mode

- Controla cómo se comportan las partículas en relación con el movimiento del objeto que las emite:
 - **Off** → Las partículas se generan sin tomar en cuenta la velocidad del emisor.
 - **Rigidbody** → Se ven afectadas por la velocidad del Rigidbody del emisor.
 - **Transform** → Se ven afectadas por el movimiento del objeto transform del emisor.

20. Max Particles

- Define el número máximo de partículas que el sistema puede generar al mismo tiempo.
- Limitar este valor ayuda a optimizar el rendimiento.

21. Auto Random Seed

- Si está activado, Unity genera un valor aleatorio para la simulación cada vez que se ejecuta.
- Si está desactivado, puedes fijar un **Random Seed** para que el comportamiento sea siempre el mismo.

Creando nuestras Partículas:

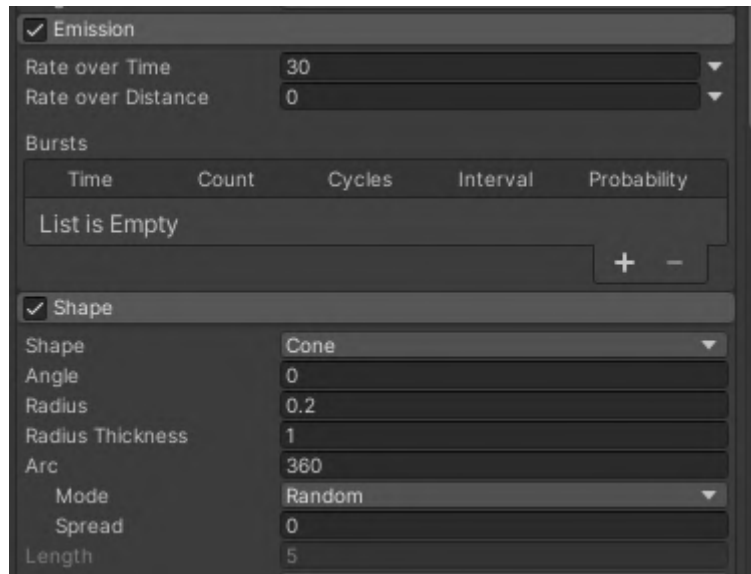
🔥 Fuego:

1. **Crear el sistema de partículas:**
 - En el **Hierarchy**, haz clic derecho → *Effects* → *Particle System*.
 - Renómbralo a **"Fire"**.
2. **Configurar el Main Module:**
 - **Duration:** 2
 - **Looping:** ☒ (Activado)
 - **Start Lifetime:** 1 - 2 (Rango aleatorio)
 - **Start Speed:** 0.5 - 1
 - **Start Size:** 0.5 - 1
 - **Start Color:** Gradiente de naranja a amarillo 🔥
 - **Gravity Modifier:** 0



3. Modificar la Emisión y Forma:

- **Emission:**
 - Rate over Time: 30
- **Shape:**
 - Tipo: *Cone*
 - Angle: 0
 - Radius: 0.2



4. Añadir Color Over Lifetime:

- Activa el módulo **Color over Lifetime**.
- Usa un gradiente: **naranja** → **amarillo** → **transparente**.

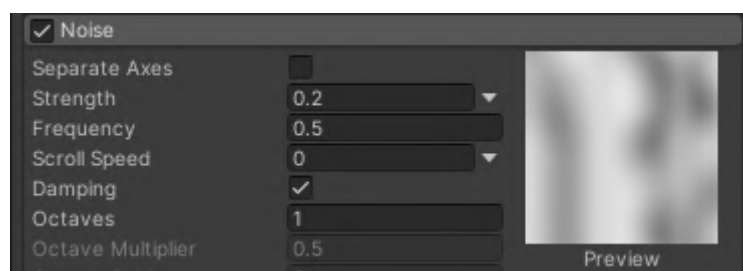
5. Añadir Size Over Lifetime:

- Activa el módulo **Size over Lifetime**.



6. Añadir Noise (Opcional):

- Activa el módulo **Noise** para darle más dinamismo.
- Strength: 0.2



7. Textura y Material:

- Crea un **material** con un shader de partículas y usa una textura de fuego.
- Pueden encontrar la textura en este link:

<https://drive.google.com/drive/folders/19TMho3Nn1q-Z4PWzmx07XzU06bbL-R1q?usp=sharing>

Editar la textura

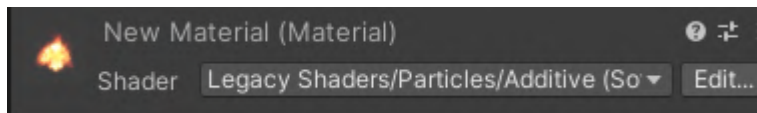
Importar la Textura en Unity

1. Arrastra la imagen de la textura a la carpeta **Assets** en Unity.
2. Selecciona la textura y, en el **Inspector**, cambia estas configuraciones:
 - **Texture Type:** Default
 - **Alpha Is Transparency:** ☒ (Activado)
 - **Wrap Mode:** Repeat
 - **Filter Mode:** Bilinear
 - **Compression:** None

Esto asegurará que la textura tenga buena calidad y un fondo transparente.

Crear un Material para las Partículas

1. Ve a la carpeta **Assets**, haz clic derecho → *Create* → *Material*.
2. En el **Inspector**, cambia el **Shader** a **Additive(Soft)**



Aplicar la Textura al Material

1. En el **Inspector** del material, busca la opción **Albedo / Main Texture**.
2. Arrastra la textura de fuego a este campo.
3. Ajusta los valores de **Tint Color** para que el fuego tenga el color deseado.

Asignar el Material al Sistema de Partículas

1. Selecciona el sistema de partículas que creaste.
 2. En el **Inspector**, busca el módulo **Renderer**.
 3. En la opción **Material**, arrastra **"Fire_Material"**.
-

Render

Renderizar es el proceso de generar la imagen final de un objeto 3D o 2D en la pantalla. La computadora toma la información del objeto (forma, textura, color, iluminación) y la convierte en **píxeles visibles** en la escena del juego. Este proceso lo realiza la **GPU (Tarjeta Gráfica)**, y puede ser más rápido o más lento dependiendo de la cantidad de detalles a mostrar.

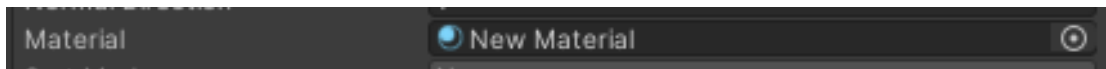
Entonces un **Renderer** es un componente en Unity que le **permite a un objeto aparecer en la pantalla**.

El Renderer le dice a Unity **cómo dibujar un objeto**, tomando en cuenta su forma (malla), materiales, texturas y efectos visuales.

Que nuestro sistema de partículas tenga una opción de "Render" es vital. Esta nos dará la capacidad de convertir casi cualquier cosa en partículas. Veamos sus opciones a destacar:

1. Material

Define la **textura o shader** que se aplicará a las partículas.



Opciones Claves:

- Puedes asignar un **Material con una textura** para cambiar el aspecto de las partículas.
- Se recomienda usar shaders específicos de partículas como:
 - **Particles/Unlit** → No se ve afectado por la luz.
 - **Particles/Additive** → Hace que las partículas brillen y se mezclen con el fondo (ideal para fuego o magia).
 - **Particles/Alpha Blended** → Usa transparencia para suavizar los bordes.


Ejemplo: Para hacer fuego, asigna un material con una textura de llamas y usa el shader **Particles/Additive**.

2. Render Mode

Cambia la forma en que se representan las partículas en la escena.



Opciones Claves:

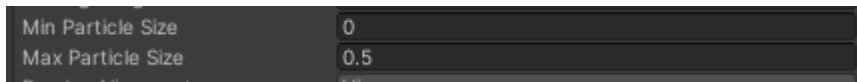
1. **Billboard** (Por defecto) → Las partículas siempre miran a la cámara. 
2. **Horizontal Billboard** → Las partículas siempre están planas en el suelo.
3. **Vertical Billboard** → Las partículas siempre son verticales pero giran hacia la cámara.
4. **Mesh** → En lugar de usar un sprite, se usa un **modelo 3D** para cada partícula.

Ejemplo:

- **Billboard** es ideal para humo, fuego y efectos mágicos.
- **Mesh** es útil para proyectiles 3D como flechas o esferas de energía.

3. Min Particle Size / Max Particle Size

Controla el tamaño mínimo y máximo que pueden tener las partículas en pantalla.



Opciones Claves:

- Si **Max Particle Size** es **muy grande**, las partículas podrían cubrir demasiado espacio en la pantalla.
- Si **Min Particle Size** es **muy pequeño**, podrían no ser visibles.

Ejemplo:

Si creas partículas de polvo en un juego 3D, puedes limitar su tamaño para que no crezcan demasiado en la distancia.

4. Flip

Invierte las partículas en **X** o **Y**, dándoles variedad visual sin necesidad de nuevas texturas.



Opciones Claves:

- **Flip X:** Invierte la partícula horizontalmente.
- **Flip Y:** Invierte la partícula verticalmente.

Ejemplo:

Si tienes partículas de hojas cayendo, activar **Flip X** y **Flip Y** puede hacer que cada hoja tenga una orientación diferente, creando un efecto más natural. 🍂

Conclusión

El sistema de partículas es bastante complejo, su gran número de herramientas y versatilidad para mezclarlas permite generar millones de posibilidades distintas. En la clase pudimos ver algunas de las más importantes. Les sugerimos jugar con ellas y crear sus propias ideas.



Efectos Especiales:

*Después de semanas de arduo trabajo en **Nexus**, el equipo de **Talentolab** ha demostrado ser una pieza clave en el desarrollo del proyecto. Las mecánicas básicas, el sistema de enemigos y la*

optimización del rendimiento están avanzando a paso firme.

Sin embargo, los diseñadores y artistas han señalado algo importante: el mundo de Nexus se siente vacío.

"El motor gráfico ya es eficiente, pero nos falta algo más... ¡Necesitamos vida, dinamismo! Fuego ardiendo en las antorchas, humo saliendo de los respiraderos, chispas en los impactos, y energía fluyendo por los portales. Todo esto hará que el mundo cobre vida y se sienta realmente inmersivo."

Para lograr esto, la **nueva misión** es dominar el **Sistema de Partículas de Unity**.

Ejercicios prácticos:

Creando Efectos Únicos para Nexus

El equipo de **TalentoLab** ha dado los primeros pasos en la implementación del **Sistema de Partículas**, logrando efectos básicos como fuego y humo. Durante una reunión con el equipo de dirección artística de **Nexus**, surge una nueva solicitud:

"Necesitamos efectos visuales que sean únicos y memorables. Algo que haga que nuestro mundo se sienta especial y diferente de cualquier otro juego."



Luigi cree que es el momento ideal para que aportes tu creatividad y dejes un sello en el proyecto.

Objetivo: Experimentar con el **Particle System** de Unity y crear efectos únicos para el mundo de Nexus.

1. **Explorar las configuraciones del sistema de partículas**, modificando color, tamaño, velocidad, colisiones y efectos especiales.
2. **Diseñar un efecto original**, ya sea un aura mágica, una explosión distinta, una neblina sobrenatural o cualquier otra idea innovadora.
3. **Justificar cómo este efecto podría encajar en el mundo del juego**, pensando en qué mecánica, entorno o narrativa lo integraría mejor.

Materiales y recursos adicionales.

Introducción a Sistema de Partículas

<https://learn.unity.com/tutorial/introduction-to-particle-systems>

Documentación

<https://docs.unity3d.com/2022.3/Documentation/Manual/ParticleSystems.html>

Preguntas para reflexionar.

1. ¿Qué partículas sería interesante crear?
2. ¿Podríamos usarlas para indicar la naturaleza de la situación u objeto?

Próximos pasos.

En la próxima clase veremos una introducción a juegos Mobile. Pasaremos a adaptar nuestros controles básicos para poder usarlos en Celular y crearemos una Build para jugarlo



Buenos Aires
aprende
Agencia de Habilidades para el Futuro

BA Buenos
Aires
Ciudad