

A blue parallelogram and a light green parallelogram are positioned on the left side of the slide, overlapping each other and the dark background. The blue shape is on the left, and the green shape is to its right, partially overlapping it.

Curso introductorio de Shaders en Unity URP

Pablo Moreno - @AstrophelMoore



Profesor

Pablo Moreno - @AstrophelMoore



- Fundador de **Croxel Studios**.
- **Director Creativo y Productor** de *Tormenture*.
- **10+ años de experiencia** en la industria.
- **20+ desarrollos de videojuegos** a sus espaldas.



Audiencia Objetivo

Interesados en aprender **creación de Shaders** para la realización de **arte técnico** para videojuegos **a partir de 15 años**.

Prerrequisitos mínimos

- Conocimientos básicos de **Unity**; importar assets, asignar materiales, escena y cámara.
- Nociones básicas de **vectores** y **matemáticas**.



Objetivo del curso

Al final del curso los alumnos **entenderán el funcionamiento de la pipeline de renderizado** y los conceptos generales necesarios para la **creación de shaders mediante Shader Graph** y tendrán las herramientas necesarias para que puedan **desarrollar sus propios shaders** más adelante y para que sigan aprendiendo.

Además, los alumnos acabarán el curso con proyectos de **shaders de calidad profesional** hechos en clase que podrán **mostrar en su portafolio**, tales como:

- Efecto *dissolve* simple.
- Vertex-displacement shader que simula olas de agua.
- Shader de escala de grises para sprites.
- Shader de iluminación Cartoon.
- Shader de escala de grises para cualquier textura.
- Shader de detección de bordes en pantalla completa.



Contenido

Qué incluye

- Entendimiento de los **fundamentos de la Render pipeline**.
- Desarrollo de habilidades básicas para **construir shaders de media complejidad** en Shader Graph aplicables a proyectos profesionales.
- Conocimientos básicos de **sistemas de partículas en CPU** en Unity URP.

Qué no incluye

- Conocimientos avanzados de **código HLSL**.
- Conocimientos de **Compute Shaders**.
- Modificaciones y uso avanzado de **Custom Render Pipelines** o **Scriptable Renderer Features**.
- Conocimientos avanzados de **efectos visuales** o **partículas en GPU**.

Clases (8 clases de 2.5h => 20h)

Módulo 1: Fundamentos

1. Introducción. La pipeline de renderizado.

- **Presentación del curso.**
 - Preguntas a alumnos para enfocar mejor las clases: Nivel de conocimiento de programación, Objetivos personales o profesionales, Expectativas, etc.
- **Fases en el pipeline de renderizado.**
- Explicación de conceptos relevantes: **UVs, normales, sampling, vertex displacement, blending.**
- Introducción a **Shader Graph** y a **Shaders de fragmento de superficie** + ejemplo **Dissolve Shader**.
 - **Ejercicio:** Modificar Dissolve Shader para que el efecto ocurra de abajo a arriba usando el nodo de posición.

3. Shaders en código - Ligero resumen para entender funcionamiento interno de Shader Graph.

- Estructura básica de un **Shader en HLSL**.
- **Estructuras de datos.**
- Funcionamiento del **Vertex** y el **Fragment shader** en HLSL.
- Ejemplo de **Sprite Shader**.
 - **Ejercicio:** Modificar un Sprite Shader en HLSL para modificar los colores del sprite.

2. Vertex shaders.

- **Diferencias entre Vertex y Fragment shaders + ejemplos.**
- **Displacement Shaders.**
 - Espacios y matrices de transformación (object, world, view, clip).
- **Ejemplos de Vertex Shaders útiles.**
- Implementación de un **Displacement Shader** sencillo con ondas senoidales.
 - **Ejercicio:** Modificar Displacement Shader para que utilice ruido y cambie con el tiempo según una velocidad => Shader de oleaje aplicable a agua.

4. Shaders modulares con Subgraphs y custom nodes.

- Aplicar **rotación y transformación de UVs** a un **Subgraph**.
 - Aprovecho para explicar funcionamiento de UVs en detalle si no dio tiempo en la clase 1.
- **Introducción a Custom Function nodes.**
 - Ejemplo con un Custom Function sencillo.
 - Introducción a Custom Functions en documentos externos.
 - **Ejercicio:** Construir nuevo efecto utilizando el Subgraph de Custom Function que ya tenemos.



Clases (8 clases de 2.5h => 20h)

Módulo 2: Conceptos avanzados

5. Tipos de grafo y sistemas de partículas.

- Diferencias entre Opaque y Transparent. Orden de render.
- Unlit, Lit, Canvas, Fullscreen. Usos de cada uno.
- Introducción a sistemas de partículas de CPU.
- Ejemplo de sistema de partículas de brillo.
 - **Ejercicio:** Construir shader Unlit para sistema de partículas.
- Custom Textures
 - **Ejercicio:** Construir shader de Custom Texture para cambiar los colores de cualquier textura.

7. Efectos de outline.

- Diferentes acercamientos al renderizado de outlines.
- Inverted Hull.
- Depth Texture, Normals Texture.
- Fullscreen shader de detección de bordes
 - **Ejercicio:** Border detection shader sencillo en fullscreen.

6. Custom Lighting.

- Modelos de iluminación (Lambert, phong, blinn-phong; PBR básico conceptualmente).
- Tipos de toon shading.
 - Espacios y matrices de transformación (object, world, view, clip).
- Funcionamiento de las luces en la pipeline de renderizado.
- Shader Graph básico de Custom Lighting aplicado en la directional light.
 - Ligera introducción al Custom Lighting aplicado en point lights. Explicar for loops en shaders. Explicar operaciones de blend.
 - **Ejercicio:** Modificar Shader para obtener otro tipo de toon shading.

8. Optimización, Debugging, Keywords y Shader Variants.

- Proceso de debuggear Shaders.
- Profiling y Frame Debugger.
- SRP Batchter y Draw Calls.
- Propiedades tipo Keyword.
 - Funcionamiento y razón de uso de Shader Keywords.
 - Problemas de Shader Variant Explosion.
 - Opciones de Keywords + Opciones extra en HLSL.
 - **Ejercicio:** Construir un shader combinando lo aprendido y que funcione con Keywords
- Pequeña charla sobre cosas que se quedan en el tintero.