****

Módulo Profesional 08: Programación para dispositivos móviles y multimedia **M08UF2 - Act1**

**ShaderGraph**

**CICLO FORMATIVO DE GRADO SUPERIOR EN**

**Videojuegos y ocio digital**

**MODALIDAD PRESENCIAL**

**Nombre y apellidos del alumno**



**Introducción**

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Prepara una serie de materiales interesantes en Unity. Utiliza ShaderGraph para diseñar el Shader con el que esos Shaders tiene funcionamiento

|  |
| --- |
| Objetivos |
| Entender el proceso de dibujado que ocurre en la GPU.  Entender el concepto de shader y aprender a crear shaders a través de la herramienta ShaderGraph.  Crear materiales utilizando dichos shaders, pudiendo diferenciar un material de otro a través de los parámetros aceptados por dichos shaders.  Crear shaders que se comportan de diferente manera según el tiempo, la posición de mundo, etc.  Crear shaders que determinen la posición de los vértices (vertex shaders).  Crear shaders para el dibujado de superficies (pixel/fragment shaders).  Crear shaders para UI.  Crear shaders a pantalla completa. |
| **Metodología de evaluación** |
| 15% Ejercicio 1.  15% Ejercicio 2.  15% Ejercicio 3.  15% Ejercicio 4.  15% Ejercicio 5.  5% Ejercicio 6.  20% Opcionales  Recuerda que todos los ejercicios obligatorios tienen que estar correctamente terminados para poder aprobar la actividad. |
| **Entrega** |
| A través del campus con nombre M08UF2-ACT1-NombreApellidos.zip incluyendo enunciado, proyecto Unity, APK y vídeo de la prueba.  Importante:   * Si falta cualquier entregable la nota es 0 * Por cada día de retraso en la entrega se resta un punto |
| **Documentos de referencia** |
| Documentación de Unity  [Basics of Shader Graph - Unity Tutorial (youtube.com)](https://www.youtube.com/watch?v=Ar9eIn4z6XE) |

**Situación**

Acabas de entrar a trabajar de prácticas en un pequeño estudio independiente. Es tu primer día, estás sentado en tu sitio, y te llega un olor como de carne a la brasa. Siguiendo el olor, te das cuenta de que viene del artista gráfico, que tiene que preparar una serie de efectos especiales – pero como no conoce las capacidades de los shaders, lleva días cabreado preparando una serie de efectos a mano y está que arde (de ahí el olor).

Como sabes que a ti te va a costar mucho menos gracias al poder de los Shaders y de ShaderGraph, te ofreces voluntario a preparar dichos efectos.

**Descripción de la Aplicación**

Se trata de una aplicación que tiene que funcionar para Android, tanto en modo horizontal como en modo vertical.

La aplicación se realizará en Unity. El pipeline gráfico a utilizar será URP.

La aplicación consiste en la preparación de una timeline que demuestre el funcionamiento de los efectos que se van a implementar:

* Inflate
* Glow
* Dissolve
* Heat Haze
* UI Scroll

**Ejercicio1: Inflate**

Prepara un Shader que infle y desinfle un modelo en ciclos de una duración determinada utilizando un “easing” de seno. Llámalo **InflateShader**.

Pantalla de juego de computadora

Descripción generada automáticamente con confianza media

**Parámetros**

* Distancia de Inflación Máxima
* Tiempo total de ciclo

**Cómo hacer la inflación**

* Debes realizar la siguiente operación para cada vértice y pasarla a la VertexPosition del MasterNode
  + nuevaPosicion(local) = posiciónVertice(local) + (normalVertice(local) \* distanciaInflaciónSegunElTiempo)

**Cómo calcular la Distancia de Inflación según el Tiempo**

* Debes utilizar el nodo Time para leer el tiempo actual, dividirlo entre 2 \* PI \* TiempoTotalDeCiclo, y utilizar esto como parámetro para el nodo Sin
* El resultado es un valor entre -1 y 1, que normalizaremos sumando 1 (ahora el valor está entre 0 y 2) y, posteriormente, dividiendo entre 2 (ahora el valor está entre 0 y 1)
* Este valor, multiplicado por la Distancia de Inflación Máxima, es la Distancia de Inflación según el Tiempo

**Demostración del funcionamiento**

Crea un material partiendo del InflateShader y aplicalo a tres modelos: una esfera, una capsula, y un cubo.

**Ejercicio 1:** Incluye a continuación un enlace a un vídeo en el que demuestres el funcionamiento.

|  |
| --- |
|  |

**Ejercicio2: Glow**

Prepara un Shader que, aplicado a un modelo, module el canal de emisión en ciclos de una duración determinada utilizando un “easing” de seno. Llámalo **GlowShader**.

Pantalla de un computador

Descripción generada automáticamente con confianza media

**Parámetros**

* Textura de Color
* Textura de Normal
* Textura de Emisión
* Color de Emisión
* Emisión Mínima
* Emisión Máxima
* Fuerza del Fresnel
* Tiempo Total de Ciclo

**Crea los Inputs Normales**

* Prepara nodos de tipo SampleTexture para leer las texturas de Color y Normal y hacerlas llegar a los Inputs adecuados del MasterNode
  + Para la textura de color, estos son los inputs de Color y Alpha
  + Para la de Normal, es el input de Normal

**Crea el Input de Emisión**

* Prepara el input para la emisión. Este Input recibirá el resultado de la siguiente operación:
  + Suma:
    - Valor de salida del un Nodo Fresnel que tienes que crear
    - Sampling de la textura de Emisión
  + Multiplica el resultado por el Color de Emisión
  + Multiplica el resultado por la Emisión en el momento actual:
    - Debes utilizar el nodo Time para leer el tiempo actual, dividirlo entre 2 \* PI \* TiempoTotalDeCiclo, y utilizar esto como parámetro para el nodo Sin
    - El resultado es un valor entre -1 y 1, que normalizaremos sumando 1 (ahora el valor está entre 0 y 2) y, posteriormente, dividiendo entre 2 (ahora el valor está entre 0 y 1)
    - Utilizaremos este valor en un Nodo Remap, de modo que obtengamos un valor entre Emisión Mínima y Emisión Máxima
  + Envía este valor al Input de Emisión del MasterNode

**Demostración del funcionamiento**

Crea un material partiendo del GlowShader y aplicalo a tres modelos: una esfera, una capsula, y un cubo.

**Ejercicio 2:** Incluye a continuación un enlace a un vídeo en el que demuestres el funcionamiento.

|  |
| --- |
|  |

**Ejercicio3: Dissolve**

Prepara un Shader que haga aparecer y desaparecer un modelo en ciclos de una duración determinada utilizando un “easing” de seno. Llámalo **DissolveShader**.

Pantalla de una computadora

Descripción generada automáticamente

**Antes de Empezar**

* Tendrás que modificar la propiedades del Shader para que acepte Alpha Clipping

**Parámetros**

* Textura de Color
* Textura de Normal
* Textura de Emisión
* Intensidad de Emisión
* Tiempo Total de Ciclo
* Escala del Ruido

**Crea los Inputs Normales**

* Prepara nodos de tipo SampleTexture para leer las texturas de Color, Normal y Emisión y hacerlas llegar a los Inputs adecuados del MasterNode
  + Para la textura de color, estos son los inputs de Color y Alpha
    - En la próxima sección explicaremos cómo modificar el Color de Alpha
  + Para la de Normal, es el input de Normal
  + Para la de Emisión, es el input de Emisión
    - Multiplícalo por la Intensidad de Emisión antes de hacerlo llegar al Input

**Cómo hacer la Disolución**

* Debes realizar la siguiente operación y hacerla llegar al valor de Alpha
  + Crea un nodo de tipo ruido, pasa el parámetro de escala de ruido a dicho nodo
    - Los valores de salida de este nodo están entre el 0 y el 1
  + Súmale una valor de seno ajustado al Tiempo Total de Ciclo
    - Debes utilizar el nodo Time para leer el tiempo actual, dividirlo entre 2 \* PI \* TiempoTotalDeCiclo, y utilizar esto como parámetro para el nodo Sin
    - El resultado está entre -1 y 1
    - Dado que el valor de ruido está entre 0 y 1:
      * Cuando el Seno devuelve -1, la suma de ambos está entre -1 y 0
      * Cuando el Seno devuelve 1, la suma de ambos está entre 1 y 2
  + Crea un nodo Clamp para recortar entre 0 y 1
  + Multiplica el Alpha que obtienes de la textura de color con este valor

**Demostración del funcionamiento**

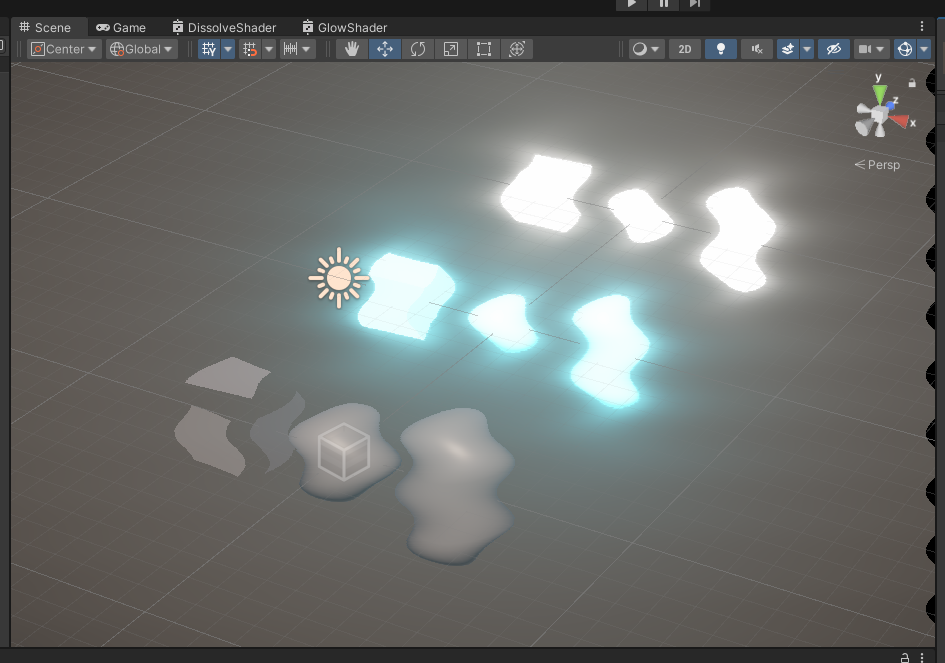
Crea un material partiendo del DissolveShader y aplícalo a tres modelos: una esfera, una capsula, y un cubo.

**Ejercicio 3:** Incluye a continuación un enlace a un vídeo en el que demuestres el funcionamiento.

|  |
| --- |
|  |

**Ejercicio4: Heat Haze**

Prepara un Shader que haga un efecto de “distorsión por calor” a pantalla completa. Llámalo **HeatHazeShader**.



**Parámetros**

* Número de Filas
* Desplazamiento
* Tiempo Total de Ciclo

**Antes de Empezar**

* Selecciona la cámara de la escena
  + En Inspector 🡪 Componente Camera 🡪 Apartado Rendering 🡪 Propiedad Renderer selecciona el **Renderer 0:URP-HighFidelity-Renderer**
* Selecciona el asset en Project 🡪 Assets/Settings selecciona el URP-HighFidelity-Renderer
  + Pulsa el botón Add Renderer Feature y elige la Full Screen Pass Renderer Feature
    - Esto te permitirá asignar el material de Full Screen al Renderer

**Crea un Shader Full Screen que “no haga nada”**

* Crea un nuevo Shader, esta vez de tipo Full Screen Shader. Llámalo **HeatHazeShader**.
* En shadergraph:
  + Añade el nodo URP Sample Buffer
  + Selecciona Source Buffer 🡪 BlitSource
  + Conecto el output con los dos inputs del Fragment Node (Base Color, Alpha)
* Crea un nuevo Material que utilice este Full Screen Shader
* Asígnalo a la Full Screen Pass Renderer Feature
* Comprueba que la imagen que sale es la normal (es decir, es igual que si no estuviera asignado dicho material)

**Modifica el Shader para que haga distorsión por calor**

* Haremos la distorsión pasando al URP Sample Buffer una versión ligeramente modificada de la Screen Position
* Para ello:
  + Crea el Nodo Screen Position
  + Con el Nodo Split, separa en 4 valores el Output de dicho nodo
  + Multiplica el valor G (sería la Y) \* 2 \* PI \* Número de Filas
  + Suma al resultado el Nodo Tiempo 🡪 Time Dividido entre el Tiempo de Ciclo Total
  + Obtén el seno de dicha suma y multiplícalo por el Desplazamiento
  + Pasa el Resultado a la X de un Nodo Vector2, y deja la Y en 0
  + Suma este Vector2 al Nodo Screen Position
  + Utiliza el resultadocomo input para el URP SampleBuffer

**Demostración del funcionamiento**

En estos momentos ya debería haber un efecto a pantalla completa que esté realizando la distorsión por calor.

**Ejercicio 4:** Incluye a continuación un enlace a un vídeo en el que demuestres el funcionamiento.

|  |
| --- |
|  |

**Ejercicio5: UI Scroll**

Prepara un Shader de tipo Sprite Unlit para utilizarlo en un elemento de UI. Llámalo **UIShader**.

Imagen que contiene luz, tráfico, computadora, firmar

Descripción generada automáticamente

**Parámetros**

* Escala
* Textura 2D llamada específicamente MainTex
  + Este nombre está reservado a la textura principal que proporciona el componente Image

**Antes de Empezar**

* Marca la casilla Alpha Clipping en las propiedades del Shader

**Haciendo el Shader**

* La idea es hacer un shader sencillo que muestre un movimiento sobre el objeto de UI
* Realiza los siguientes pasos:
  + Crea un Nodo Screen Position
    - Multiplícalo por el parámetro Escala
  + Crea un Nodo Time
    - En un Nodo Vector2 pasa el SineTime del Nodo Time, y en la Y el CosineTime
  + Suma el resultado de la Multiplicación y el Vector2
  + Crea un nodo Voronoi que reciba en el Input UV el resultado anterior
  + Crea un nodo Vertex Color y multiplícalo por el Out del Voronoi
  + Crea un nodo Sample Texture 2D y pon como input la MainTex
    - Envía el Output de Alpha al Input del Fragment Shader
    - Multiplica el Output RGBA con el resultado de la multiplicación del Voronoi, y pasalo al Base Color del Fragment Shader

**Demostración del funcionamiento**

Crea un material partiendo del UIShader y aplícalo a un botón que quede centrado en pantalla

**Ejercicio 5:** Incluye a continuación un enlace a un vídeo en el que demuestres el funcionamiento.

|  |
| --- |
|  |

**Ejercicio6: Timeline Demostrativa**

Prepara una timeline en bucle que muestre los diferentes modelos con los diferentes efectos en marcha.

Prepara un botón para resetear la escena, y otro para cerrar la aplicación.

**Ejercicio 6:** Incluye a continuación un enlace a un vídeo en el que demuestres el funcionamiento.

|  |
| --- |
|  |

**Extra (1/3 cuenta como actitud): Características Opcionales**

Puedes realizar por tu propia al menos dos de las siguientes características:

* **Realiza un shader de superficie de agua (10%) –** realiza un shader que combine dos mapas de normales para hacer un efecto de agua. Puedes utilizar también un mapa de distorsión.
* **Realiza un shader libre (10%) –** realiza un shader que no se parezca a ninguno de los shaders descritos en el ejercicio

**Opcionales:** Incluye a continuación un enlace a un vídeo en el que demuestres el funcionamiento.

|  |
| --- |
|  |