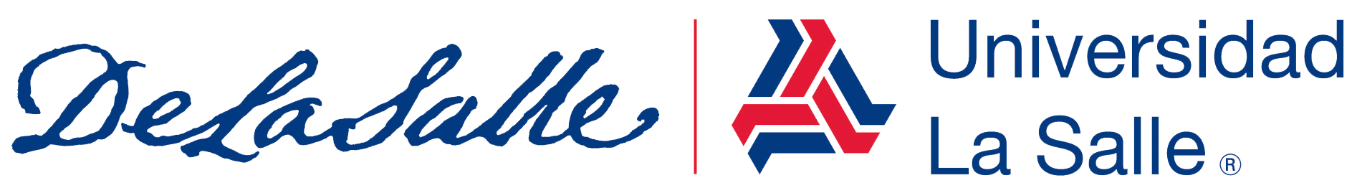
**–INGENIERIA Producción Multimedia**

***“Proyecto Primer Parial”***

Andrés Ramón Reyes Galván

189236

Ingeniería Producción Multimedia

Materia: Estadistica

Evaluación de la Asignación por el Mtro. Jesus Caro

**Ciudad Obregón, Sonora. de Febrero de 2020.**

Introducción

En este formato se hablará acerca del proyecto que se estuvo realizando en el Primer parcial de La materia Tópicos de Física, de la misma forma, se mostrara en la forma y como fue realizado, desarrollando los temas o tópicos que se estuvieron viendo en el parcial, finalizando con el proyecto dando la propia conclusión del tema.

Objetivo

El Objetivo del proyecto es añadir Shaders en un sprite el cual debe tener: Entintado, Outline, Glitch y Rimlight con Fresneit, los cuales tanto como el Outline y el Tinte estarán funcionados en un Shader para visualizarlo en un sprite; el Glitch y el Rimlight juntos en otro, para formar 2 conjuntos, los cuales tienen 2 shaders en cada uno de los conjuntos.

Preparativos

Antes de empezar a desarrollar el proyecto, tendremos que saber algunos conceptos básicos, los cuales son fundamentales para realizarlo. A continuación, se hablaran de estos mismo:

**Shader**

Como ya se sabrá, un programa o un juego, se compone de funciones. Así, podemos encontrar una función de disparo que llama a la función que diseñada por un arma, manda reproducir un sonido de disparo y genera una bala que con su propia función se irá moviendo.  
  
En el caso de los shaders, también hablamos de funciones (algoritmos si nos ponemos técnicos). Un shader es un algoritmo que calcula cómo responde un punto de un objeto ante la luz. De este modo, si le da una luz roja, de una sirena por ejemplo, el shader calculará que un punto de la pared debe enrojecerse dependiendo de la distancia con la fuente de luz, su intensidad, otras fuentes de luz, la oclusión ambiental, etc.

**Material**

Mediante el proyecto han sido utilizados varios materiales, los cuales son utilizados en conjunto con Mesh Renderers, Particle Systems y otros componentes de rendering utilizados en Unity. Estos juegan una parte esencial en la definición de cómo los objetos son mostrados.

**Sprites**

Los Sprites son objetos gráficos 2D. Si usted está acostumbrado a trabajar en 3D, los Sprites son esencialmente texturas estándar, pero hay técnicas especiales para combinar y manejar texturas sprites por eficiencia y conveniencia durante el desarrollo.

**Tinte**

O conocido mayormente como Tint, es el tratamiento de colores que se le puede colocar al material mediante la sincronización del Shader.

**Outline**

También conocido como el contorno, es el componente Outline agrega un efecto simple de contorno a componentes gráficos tal como Texto o Imagen. Debe estar en el mismo GameObject que el componente gráfico.

**Holograma(Glitch)**

Se denomina holograma a una especie de fenómeno visual donde una imagen parece tridimensional por efectos de la luz.

**Rimlight(Fresneit)**

Da un tipo de aura o brillo

Procedimiento

Ahora que ya sabemos sobre algunos de los conceptos generales, podemos desarrollar el proyecto. A continuación, se mostrarán los pasos a seguir para el proyecto y la experiencia del mismo. Ptt: Sabiendo que ya se creó un proyecto de unity.

Primeramente, se creará un shader el cual se le colocara el nombre que dese, adentro de este mismo se creara lo siguiente; una estructura base, la cual se llama MainText el cual será anclado con un simple 2D para el cual se dé el color. Después se creará un color llamado “Albedo” del color preferido, el cual se le conectará a un multply ya creado. El alfa ya pudiera estar pintando, sin embargo, esta enfrente del entintado, por lo tanto, lo que se tendrá que hacer, es hacer otro fragmento de color llamado “Entintado” el cual servirá el entintado de color que tendrá el estructurado del material, para que el alfa no desaparezca al personaje, es por esto que se coloca el albedo color; ahora se tomara el Albedo se colocara el color en un Split para separar todos los colores, el cual se unirá con un multiply; después se usara el color con otro multiply; ahora el tinte creado con un “color” será conectado con un Split hacia un combine; enseguida se conectara el color base con el blend y el combine creado desde el intentado se agarrara el RGB; el intentado solo se quitara el alfa del Split y se colocara en un blend. El Blend será dirigido en un splt que se dirigirá en combine, suprimiendo el alfa y el color dirigido directamente hacia el mismo combine, por último, se conectará el combine directamente hacia la base de todo. Finalmente, ya Habiendo guardado el shader y si gustas de probarlo, crear el material junto con el sprite que desees y únelos, para poder usar el shader creado.

Después, se creara otro shader el cual puede ser combinado con el mismo material y el mismo sprite del anterior, ingresamos en el shader de outline que crearemos; se relaizara un maintext, para un reder del sprite, la base será el sprite que ya será utilizado antes o cualquier otro; crearemos unos estructura 2d el cual será conectado para asegurarnos que se pueda pre visualizar en la pantalla; señalaremos un color el cual le pondremos de nombre “Outline color” y le añadiremos el HDR, para moverle al color; también se necesitara un Hamler para el tamaño del Outline, se recomienda que se coloque un nombre con el que se le identifique, este será un tipo slider e inicara de 1 y el máximo será 10 y el mínimo de 0, una vez que se tienen la estructura principal se le necesita hacer un clon(ctrl+d) para darle un deniliado de un estrucutrado original para hacer un estampado en la estrctura original; en este caso se necesitara un TIling para el outline, el cual se dejara default; se creara otro vector para unidades en selección al multiply; todo esto será conectado a un combine junto a la conexión del combine; a todo esto se recolectara en substract el cual se multiplcara con el alfa del color para darle el contorno de la estructura original, hasta aquí estaría listo, sin embargo, queremos que se contorno en todo el sprite; para esto quitaremos el multiply del cotrolador 2d y se clonara el controlador 2d para dividirlos; se hará realización de un add adiriendolo con otro add; así mismo se clonaran los anteriores para así conectarlos a ellos a un negative en los nodos; ahora desde la realización del mismo combina se conectaran a cada negativo; una vez realizado todo esto, crearemos un add el cual conectara los combine para mezclarlos con ultimo add para hacerle la realización de un clamb; por último se conectara al subtract, podrás añadirle el color y ancho que desees .

Y asi, con los seguiumientos para crear el Glitsh y el Fresnate, creamos lo que ques el brillo el lineado y el color del sprite.

Conclusion

Es facil crear Outlines, Freslight, tinte y Glitsh, sin emabrgo es algo tedioso para poder contruirlos y conectarlos para cuando queremos hacer la realizacion de la misma, en el proyecto pude aprender varias cosas al respecto de este mismo y espor ello que me gustaria seguir avazando o aprendiendo mas sobre este mismo, me ha llamado tanto la atencion jugar con materiales, sprites y shaders, que se que seguire usandolos en un futuro.

# Referencias

Anonimo. (s.f.). *2D Joints .* Obtenido de Unity : https://docs.unity3d.com/es/2018.4/Manual/Joints2D.html

Anonimo. (s.f.). *Animacion*. Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/AnimationSection.html

Anonimo. (s.f.). *asdf*. Obtenido de asdf: https://docs.unity3d.com/es/2018.4/Manual/2DSorting.html

Anonimo. (s.f.). *Collirder 2D.* Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/Collider2D.html

Anonimo. (s.f.). *Constat Force 2D.* Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/class-ConstantForce2D.html

Anonimo. (s.f.). *Effectors 2D.* Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/Effectors2D.html

Anonimo. (s.f.). *Experiencia de juego 2D*. Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/Overview2D.html

Anonimo. (s.f.). *Physics Material 2D.* Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/class-PhysicsMaterial2D.html

Anonimo. (s.f.). *Physics Reference 2D*. Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/class-Tilemap.html

Anonimo. (s.f.). *Rigidbody 2D.* Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/class-Rigidbody2D.html

Anonimo. (s.f.). *Sprite Creator*. Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/SpriteCreator.html

Anonimo. (s.f.). *Sprite Editor*. Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/SpriteOutlineEditor.html

Anonimo. (s.f.). *Sprite Editor*. Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/CustomPhysicsShape.html

Anonimo. (s.f.). *Sprite Packer.* Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/SpritePacker.html

Anonimo. (s.f.). *Sprite Renderere*. Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/class-SpriteRenderer.html

Anonimo. (s.f.). *Spriters*. Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/Sprites.html

Anonimo. (s.f.). *Tilemap*. Obtenido de Unity: https://docs.unity3d.com/es/2018.4/Manual/class-Tilemap.html

Anonimo. (s.f.). *Unity User Manual*. Obtenido de Unity Documentacion: https://docs.unity3d.com/es/2018.4/Manual/UnityManual.html

Anonimo. (s.f.). *UnityAssetStore*. Obtenido de https://assetstore.unity.com/packages/vfx/shaders/2d-shader-pack-62930

Anonimo. (s.f.). *UnityAssetStore*. Obtenido de https://assetstore.unity.com/packages/tools/visual-scripting/shadero-sprite-2d-shader-editor-97406

Anonimo. (s.f.). *UnnityAssentStore*. Obtenido de https://assetstore.unity.com/packages/vfx/shaders/2d-shader-collection-133046