Universidad La Salle Noroeste

Tópicos de física

Proyecto del primer parcial

Jennifer Elizabeth Yáñez Durán

Ingeniería en Producción Multimedia

Cuarto semestre

Profesor Jesús Francisco Caro

19 de febrero de 2020

Propósito:

Realizar cuatro efectos en un solo sprite, éstos deberán de ser, el entintado, outline, fresnel y glitch, y que puedan ser modificados, ya sea por un cambio de color, por su grosor o por su velocidad y que cada uno funcione como debe de ser.

Introducción:

En este documento se podrá observar el proceso por el que fue realizado mi proyecto del primer parcial de la clase de Tópicos de Física, así como sus complicaciones y soluciones, este está realizado en su totalidad en Unity.

Desarrollo:

Primeramente, se creó un proyecto LWPR en unity, LWPR es Lightweight Pipeline; el objetivo de LWPR es proporcionar un rendimiento optimizado en tiempo real en plataformas de rendimiento limitado al hacer algunas compensaciones con respecto a la iluminación y sombra, es por eso que elegimos este tipo, ya que lo que se hará será en tiempo real, con el uso de iluminación y sombras, esto es posible debido a que el LWRP realiza un renderizado directo de una sola pasada con una luz de sombra en tiempo real y una selección de luz por objeto, con la ventaja de que todas las luces están sombreadas en una sola pasada y dado a que esto es compatible con la herramienta de shader graph, la cual será la base de todo el proyecto.

Seguido de esto, dentro creé una carpeta, ésta se encuentra en la carpeta de assets, la nueva carpeta llevará el nombre de Shader; a continuación, seleccioné un sprite, que es un mapa de bits, el cual es gestionado por un hardware especializado y permite utilizarlo de manera independiente a los fondos, en videojuegos; es la máscara o imagen de un objeto que tiene la capacidad de colisión.

El sprite elegido se posiciona dentro de la carpeta creada recientemente, para poder tener acceso a ella y usar el sprite como tal, le tuve que cambiar el texture type, de un default a un sprite (2D and UI) para que éste funcionara correctamente, para así poder aplicarle los distintos materiales que contendrán los shaders que serán creados.

Posteriormente, creé un shader, que es un tipo especializado de programa gráfico que determina cómo la información y la textura son combinadas para generar los pixeles del objeto renderizado en la pantalla; existen diversos tipos de shaders dependiendo del objetivo que quieras lograr, en este caso se usó un shader 2D

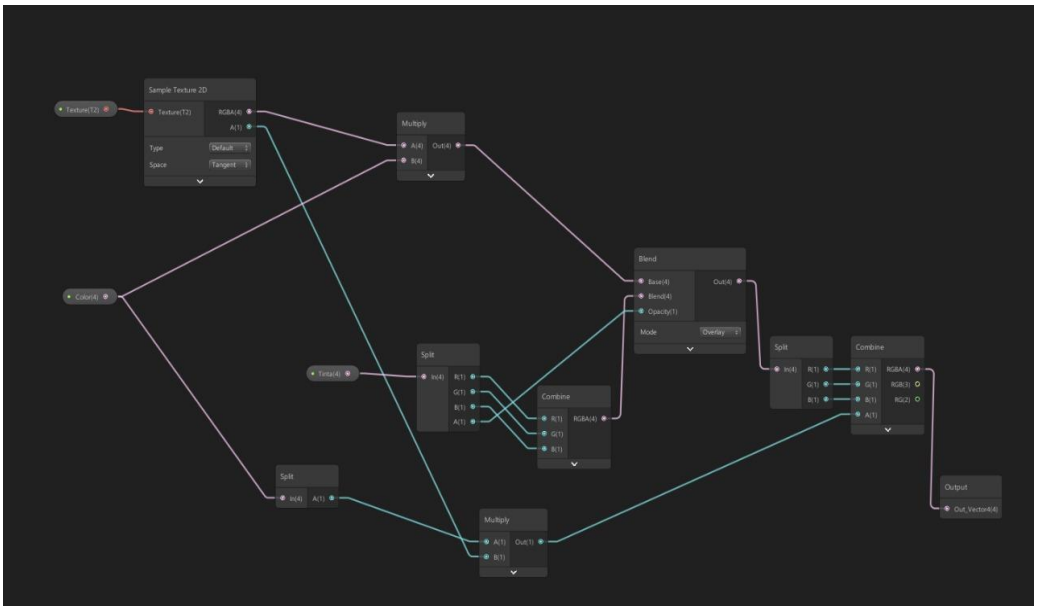
render, sprite unlit graph el cual lleva el nombre de unión, y dado que ya tenía tanto el shader como el sprite, se tenía que crear un material, los materiales son una parte esencial de cómo los objetos son mostrados, ya que su inspector muestra características determinadas por el shader con el cual se encuentra ligado, ese shader será el que creamos posteriormente.

El primer efecto que se le realizó al sprite, fue el entintado y, debido a que serían varios efectos para evitar confundirme o inclusive para tener un mejor orden en el proyecto, para el efecto de entintado, cree un subgrafo para que de esta manera funcionara conforme el shader.

Para dejar todo listo, coloqué el sprite dentro de la escena, arrastrándole el material, o bien, seleccionándolo desde el material; los shaders, que son pequeños scripts que contienen los cálculos matemáticos y algoritmos para calcular el Color de cada píxel procesado, en función de la entrada de iluminación y la configuración del Material.

Usando el primer subgrafo para el entintado; el entintado es pintar el sprite por encima del mismo, para este efecto se creó una textura y dos colores, el color base y la tinta; teniendo lo necesario, se colocó primeramente la textura en la mesa de trabajo, de la cual el alfa se multiplicó con el split de un color, lo que hace que divida la matriz del color en subcadenas (o vectores), y como es color, el split lo divide en R,G,B,A. Lo mismo se hace con la tinta (el cual es otro color, pero es el que utilizaremos para entintarlo), éste se combinó con un blend que venía de la textura y del color; el blend sirve para mezclar una base (en este caso el color y la textura, ya que de ahí parte el funcionamiento de la tinta), con la opacidad (en este caso es dado por el alfa de la tinta) y por último, lo que se va a mezclar (que es la combinación de todos los canales de la tinta, sin el alfa, ya que el alfa fue mandado como opacidad), los canales de esta mezcla se vuelven a combinar con el alfa del color y la textura, para poder crear el efecto de entintado y que sea posible bajarle todo el alfa al entintado, sin que el sprite desaparezca.

La salida para este subgrafo y para todas las demás fue de un vector 4, ya que se trataba de texturas mezcladas con colores, y algunos efectos, y como todos sabemos, un color es un vector 4.



Para cerciorarme de su funcionalidad, este subgrafo fue llamado en el shader Union, llamado por el nombre que se le dio, se volvió a crear la textura, el color y la tinta, que eran los elementos que el vector del subgrafo pedía y se conectó al color.

Habiendo comprobado su buen funcionamiento, lo siguiente fue crear otro subgrafo para el outline, este efecto agrega un contorno a componentes gráficos, este contorno debe de rodearlo al 100 %. Primeramente, se creó una textura con el sprite en uso, un vector 1 que sirvió para el grosor del outline y un color.

Con todo esto, lo primero fue colocar la textura en la mesa de trabajo y substraerle el alfa, dejando la substracción de lado por un momento; se multiplicó el vector 1 ya creado, con el nodo de vector 1, para así combinarla y mandar el último canal resultante a un nodo, el cual sirve para detallar mapas y texturas, ya que

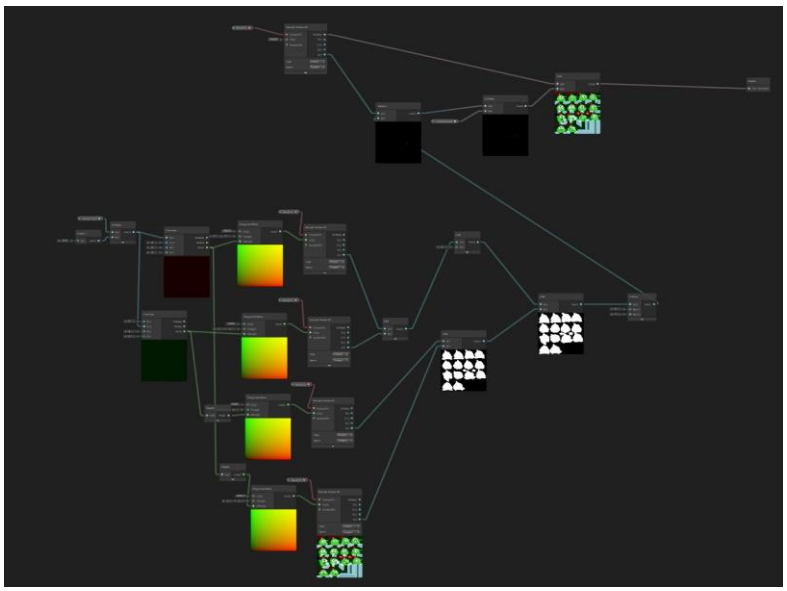
debemos de detallar el sprite para poder marcar los límites y poder realizar el outline, este nodo es el tiling and offset.

Este es mandado a la textura, con él se crea uno de los lados de sprite, pero dado que son cuatro, este proceso se deba de realizar cuatro veces; pero a diferencia de éste, dos de ellos serán ligados con un negate antes del tiling and offset, ya que esto invertirá el lugar de posicionamiento del outline.

Los que están unidos al negate, saldrán del último canal del combine del vector 1 (el creado) y del vector 1 (el nodo), para que así sea el lado opuesto del que se le está indicando, ya que, si esto no se hace, unity no sabría de dónde tomar lo que se va a hacer negativo.

Ya hecho esto, se suman los positivos y los negativos con sus respectivos, y ambos son unidos en un clamp, que devuelve un valor dentro del rango, en otras palabras, va a unir lo antes hecho y va a dar un resultante que estará limitado por los otros cuatro.

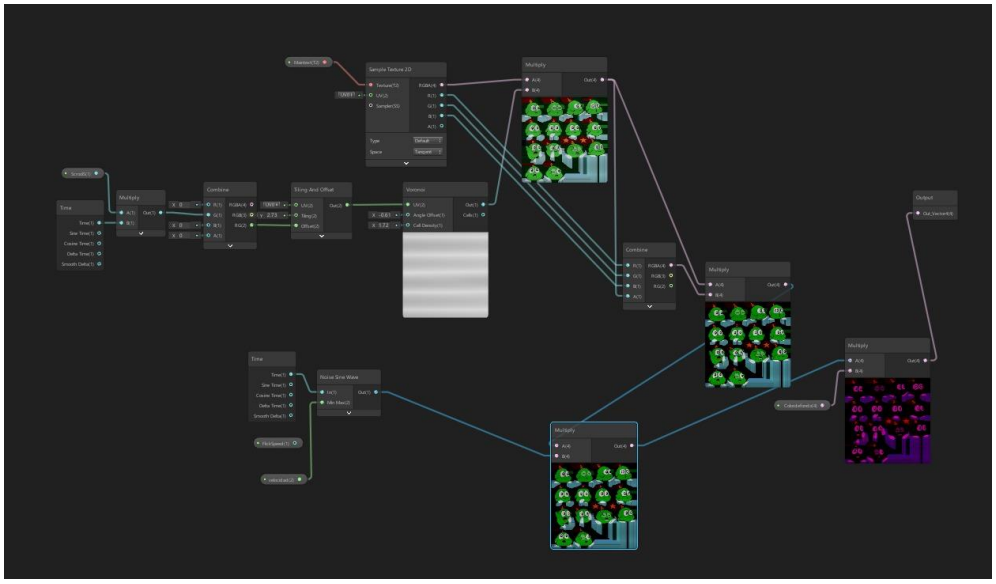
Retomando la substracción del alfa de la textura, se conectará con todo lo acabado de hacer y se multiplicará por el color que tendrá el outline y sumarlo con la textura.



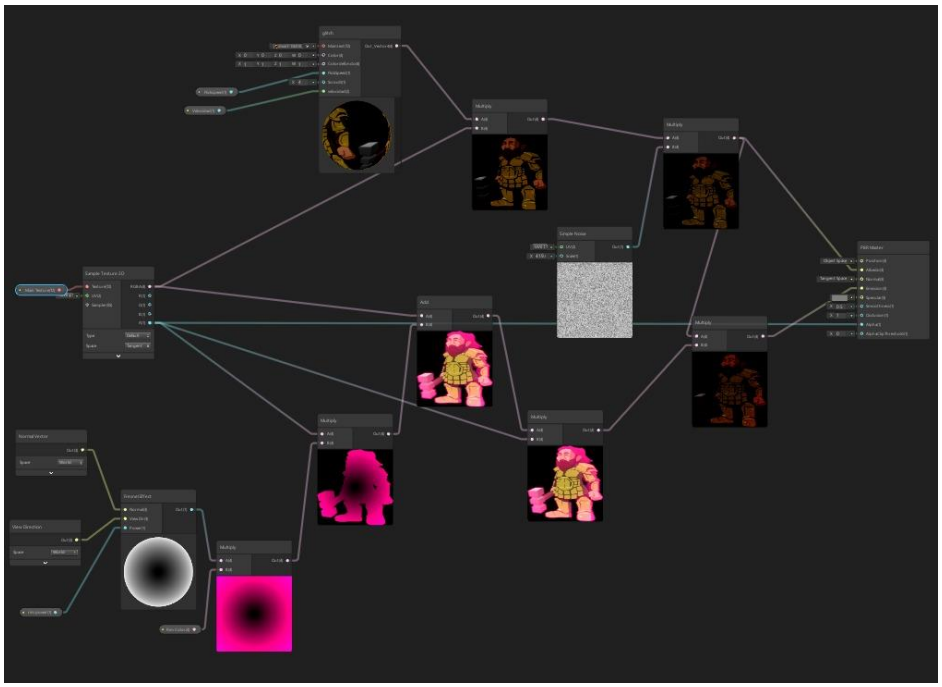
Este al igual que el subgrafo del entintado, se coloca en el shader UNION y se suma con el pasado, para colocarlos en el mismo material.

Para colocar el ruido, simplemente se buscó el nodo de simple noise que, como su nombre lo dice, le agrega ruido al sprite.

Dado que el sprite original era una hoja de sprite, para realizar el glitch y el fresnel, tuve que elegir un sprite solo, ya que el siguiente efecto, que es el de fresnel, fue en un PBR; el fresnel tiene que ver con los ángulos reflexivo de la luz de un objeto, para esto se usa el nodo de Fresnel effect, se multiplica por el color que este tendrá y por el alfa de la textura, para que solo iluminé la silueta del sprite y no el cuadro completo; pero para que esto funcione con la textura, se suma con el RGBA de la textura principal, y se volvió a multiplicar por el alfa de la textura para que iluminara solo dentro del Sprite, y esto fue conectado a la emisión del PBR Master, ya que es luz.



Dado que el glitch tenía que ser unido al fresnel, se hizo en un subgrafo, en el cual la textura se combinó con un tiling and offset y se dirigió a un voronio que era lo que le daba la textura de holograma, ya que es la intención del glitch, que se vea como si estuviera fallando y, para hacerlo sobre la textura, se multiplicó por ella y se combinó con todos los canales de ésta excepto el alfa, el cual provino de la multiplicación; por último, para dar el efecto de parpadeo, se le colocó un ruido conectado a un tiempo, para que el ruido se activara cada cierto tiempo y velocidad (para la cual se necesitó un vector 1).



Para terminar, este subgrafo se multiplicó con la textura en el PBR, se le colocó un ruido, un simple noise, multiplicándolo de igual manera y se conectó al albedo, ya que el albedo es el color puro.

Con el nuevo sprite se hizo lo mismo que con la hoja de sprites, se arrastró a la escena y se creó un material que se enlazó con el PBR.

Debido a que a una hoja de sprites no se le puede colocar de la manera correcta un fresnel hecho en PBR, tuve que elegir uno nuevo que no fuera una hoja de sprites, y dado que el fresnel y el glitch tenían que ir en un solo shader, no logré colocar los cuatro efectos en un solo sprite, por lo que la hoja de sprites tiene tres efectos más el ruido y el sprite solo tiene dos efectos más el ruido.

Conclusión:

Por tanto, se llega a la conclusión de que es posible colocar cuatro o incluso más efectos en un solo sprite, siempre y cuando se tenga en cuenta el tipo de shader que se usará y el tipo de sprite, así como el uso de consecutivo de sumas y multiplicaciones durante el proceso de unión de los distintos efectos.

Bibliografía:

Unity . (2019). Albedo color. 18 de febrero de 2020, Sitio web: https://docs.unity3d.com/Manual/StandardShaderMaterialParameterAlbedoColor.html

Unity . (2019). The fresnel effect. 18 de febrero de 2020, Sitio web: https://docs.unity3d.com/Manual/StandardShaderFresnel.html

Unity . (2019). Mathf.Clamp. 18 de febrero de 2020, Sitio web: https://docs.unity3d.com/ScriptReference/Mathf.Clamp.html

Unity . (2019). Tiling And Offset Node. 18 de febrero de 2020, Sitio web: https://docs.unity3d.com/Packages/com.unity.shadergraph@6.9/manual/Tiling-And-Offset-Node.html

Unity . (2018). Outline (Contorno). 18 de febrero de 2020, Sitio web:https://docs.unity3d.com/es/2018.4/Manual/script-Outline.html

Unity . (2019). ShadeLab:Blending. 18 de febrero de 2020, Sitio web:https://docs.unity3d.com/Manual/SL-Blend.html

Unity . (2018). Material. 18 de febrero de 2020, Sitio web:https://docs.unity3d.com/es/2018.4/Manual/class-Material.html

Unity . (2016).Creando y utilizando Materiales. 19 de febrero de 2020, Sitio web:https://docs.unity3d.com/es/530/Manual/Materials.html

Gamerdic. (2017). Sprite. 18 de febrero de 2020, Sitio web: http://www.gamerdic.es/termino/sprite/

Tim Copper. (2018). the lightweight render pipeline optimizing real time performance. 18 de febrero de 2020, de Unity Sitio web: https://blogs.unity3d.com/es/2018/02/21/the-lightweight-render-pipeline-optimizing-real-time-performance/

Unity . (2015). Experimental 2D Lights and Shader Graph support in LWRP. 18 de febrero de 2020, Sitio web: https://forum.unity.com/threads/experimental-2d-lights-and-shader-graph-support-in-lwrp.683623/

Unity . (2019).Physically Based Rendering Material Validator. 18 de febrero de 2020, Sitio web: https://docs.unity3d.com/Manual/MaterialValidator.html

Unity . (2018).Texturas 2D. 18 de febrero de 2020, Sitio web: https://docs.unity3d.com/es/2018.4/Manual/Textures.html

Unity . (2016).Editor del Sprite (Sprite Editor). 18 de febrero de 2020, Sitio web:https://docs.unity3d.com/es/530/Manual/SpriteEditor.html

Unity . (2016).Experiencia de Juego en 2D. 18 de febrero de 2020, Sitio web:https://docs.unity3d.com/es/530/Manual/Overview2D.html

Unity . (2018).Materiales, texturas y shaders. 18 de febrero de 2020, Sitio web https://docs.unity3d.com/es/2018.4/Manual/Shaders.html

Yone Moreno Jiménez. (2017). Curso Unity 2d primera parte: sprites y animaciones. 18 de febrero de 2020, de Medium Sitio web: https://medium.com/@yonem9/curso-unity-2d-primera-parte-sprites-y-animaciones-a7fbaca3638e

EVILNAPSIS. (2016). Animacion de Personajes, Elementos o Sprites en Unity. 18 de febrero de 2020, Sitio web: https://evilnapsis.com/2016/12/15/animacion-de-personajes-elementos-o-sprites-en-unity/

Unity . (2020).Sub Graph. 18 de febrero de 2020, Sitio web: https://docs.unity3d.com/Packages/com.unity.shadergraph@7.1/manual/Sub-graph.html

Unity . (2015).Nodes. 18 de febrero de 2020, Sitio web: https://docs.unity3d.com/Packages/com.unity.shadergraph@5.10/manual/Node.html

Unity . (2020).Simple Noise Node. 18 de febrero de 2020, Sitio web:https://docs.unity3d.com/Packages/com.unity.shadergraph@7.1/manual/Simple-Noise-Node.html