



Ingeniería en Producción Multimedia

“Proyecto: Shaders en 2D”

Alumno: Paula Remy Ochoa

4^{to} Semestre

189054

Cd. Obregón, Sonora, 19 de febrero del 2020

Introducción

En este proyecto se puede conocer sobre los conceptos de los materiales utilizados en Unity, para la creación de escenas donde utilicemos los ejemplos vistos en clase con el propósito de conocer cómo funcionan de manera independiente y conjunta. La física esta presente en cualquier cosa que elaboremos en la vida y en los videojuegos también está presente por este motivo es importante conocer como la luz afecta a los cuerpos, los reflejos, las sombras, la reacciones entre dos elementos diferentes y que gracias a operaciones matemáticas podemos recrear.

Las definiciones como lambert, shader lambert, blinn, phong y normal, son muestras de como se aplica los efectos en Unity como el fresnel que funciona con luz o también el glow, las texturas dan un diferente aspecto a los modelos 3D y 2D, las diferentes técnicas se combinan con conocimientos previos ya aprendidos.

Que por lo que tenemos como objetivo es crear shaders combinando 2 elementos como el outline, el entintado, efectos de hologramas y el fresnel, también se utilizan sub-graphs para poder enlazarlos en un nuevo material otorgando el efecto deseado.

¿Qué es un shader?

Estos son scripts pequeños que contienen los cálculos de matemáticas y algoritmos para calcular el color de cada pixel renderizado, basándose en el input de iluminación y la configuración del material.

Los shaders otorgan efectos especiales a los sprites o modelos, para poder hacer que el resultado sea mejor del que era inicialmente, además que se toma en cuenta como la materia se comporta de acuerdo con la situación y al entorno y por este motivo no podemos dar la misma iluminación a un escenario que ocurre de día a uno que ocurre de noche.

El shader en los videojuegos nos informa de todas las sombras que se pueden tomar en cuenta para la creación del material o realizar efecto como neblina, fuego o toques de iluminación.

Con los materiales es lo que nos permite asignar las propiedades de los shaders ajustándolo para la escena dándonos una vista previa de una visualización previa de la cámara, nos permite conocer si el shader tiene algún problema al dar un resultado correcto o si muestra algún error.

También los materiales son definiciones acerca de cómo la superficie debería ser renderizada, incluyendo referencias a texturas utilizadas, información del tiling (suelo de baldosas), tines de color y más. Las opciones disponibles para un material dependen en qué shader del material está utilizando.

Ambos conceptos son indispensables al realizar un juego y como depende uno del otro para la creación y la implementación de este, las variables que se utilizan pueden determinar cómo será utilizado una textura en un proyecto y vas creando las herramientas necesarias para completarlo logrando el sprite con los efectos necesarios, un ejemplo en el del holograma se puede utilizar para simular un video, una transmisión de hace años o un error en una televisión.

El efecto de outline es muy utilizado en los videojuego ya que puede ayudar a resalta algo dentro de un fondo, indicar un aumento de poder, un escudo o una característica que cambia del personaje y que esto nos da diferentes

interpretaciones que le podemos dar al usuario para que sea reconocible, también con el entintado se podría dar a entender un cambio de estado como en Pokémon al ser envenenado puede colorearse ligeramente de un tono morado, o también estar congelado, estar bajo un efecto positivo o un cambio con la luz. El fresnel también nos otorga una nueva manera para iluminar un material, una iluminación que empieza en el centro de esta y se va expandiendo con un difuminado de luces, en el caso del 3D puede otorgar una cierta profundidad al modelo y que se debe revisar la dirección de esta para que quede lo más natural posible.

Las luces se encargan de hacer que un material se vea más realista o que de evidencia que sigue las leyes de la física, el personaje puede mostrar una animación si se decide elaborar, dando un movimiento a la secuencia de sprites, que al dar play al proyecto podremos ver la animación del personaje combinado con el efecto elegido. La suma de luces y sombras proporciona algo más pegado a la realidad, pero también puede dar un nuevo efecto de shader, la intensidad como el poder otorga más contenido o también puede quitar un poco del medio visual que se quiere dar a interpretar.

Al elaborar un elemento es una tarea donde se debe saber cómo están conectados los efectos del proyecto, los shaders se usan para colorear píxeles alterando el color original de su textura. Es decir, tenemos una pared blanca, le acercamos la luz roja de la sirena y se tiene que ver roja. Cuanto más lejos esté la fuente de luz, menos roja se verá esa parte de la pared. O al tener una luz reflejada en un objeto la sombra producida dependerá de la cercanía de la luz al objeto, esto produce que las luces y sombras sigan un orden de acuerdo con los objetos los cuales refleja la luz.

En los juegos, animaciones, películas y series los efectos de iluminación juegan un papel importante para reflejar alguna situación, al ser parte del ambiente o quieren hacer que algo sea más evidente, misterioso o que muestre diversas maneras de mostrar algo, ejemplo en una película de terror es conveniente que las luces sean lo más leves posible para resaltar la oscuridad al relacionar la noche con el terror, pero también al tener la iluminación necesaria podemos ver a los personajes de manera exacta a diferencia de que todo estuviera con poca luz, además se podría

realizar un efecto de neblina para agregar un misterio a la escena, en las animaciones y juegos los shaders logran que partiendo de un objeto ya sea 2D o 3D cobre más características reales, que sea más apegado a la realidad.

Las texturas proporcionan al albedo mayor aspectos como si es una superficie suave o rugosa, dando nuevas posibilidades a los materiales y sprites. Además en cuanto a las luces, los light probes vendrían siendo una red creada para el transporte de luz, fueron creadas para mantener la ilusión de luz dinámica sobre nuestro personaje, ya que el escenario fue precalculado sin él. Esta red de luz solo afectaría a nuestro personaje y enemigos, calculando el resto de efectos faltantes como la luz sobre nuestro personaje y la sombra proyectada por él. Para hacerlo efectivo hay que distribuir esta red sobre los caminos posibles que nuestro personaje pudiera transitar, esto vendría siendo algo así como para renderizar como se hacía en Biohazard, solo que la cámara sería móvil y no estaría prefijada. Esto ayuda en el desarrollo del juego ya que CAPCOM les pidió desarrollar un nuevo juego y como resultado crearon Resident Evil donde Shinji Mikami invirtió tiempo con las cámaras para ser estáticas y se enfocaron en las apariencias de los zombies lo que causo un impacto en el juego dando un nuevo representativo al género de terror.

Los shaders pueden ser Pixel Shader o Vertex Shader los cuales permiten a los programadores una mayor libertad a la hora de diseñar gráficos en tres dimensiones, ya que puede tratarse a cada píxel y cada vértice por separado. De esta manera, los efectos especiales y de iluminación puede crearse mucho más detalladamente, sucediendo lo mismo con la geometría de los objetos.

Un vertex shader es una función que recibe como parámetro un vértice. Sólo trabaja con un vértice a la vez, y no puede eliminarlo, sólo transformarlo. Para ello, modifica propiedades del mismo para que repercutan en la geometría del objeto al que pertenece.

Mientras que, un píxel shader no interviene en el proceso de la definición del “esqueleto” de la escena (Wireframe), sino que forma parte de la segunda etapa: la rasterización (Rendering). Allí es donde se aplican las texturas y se tratan los pixeles

que forman parte de ellas. Básicamente, un píxel shader especifica el color de un píxel.

La particularidad de los píxel shaders es que, a diferencia de los vertex shaders, requieren de un soporte de hardware compatible. En otras palabras, un juego programado para hacer uso de píxel shaders requiere si o si de una tarjeta de video con capacidad para manipularlos. Los lenguajes son necesarios para comprender los shaders y para establecer los comandos de las funciones de píxel y vertex shaders se utiliza el lenguaje HLSL (High Level Shader Languages), que vendría a ser un lenguaje de alto nivel para trabajar con estos programas. Existe uno de bajo nivel, bastante más difícil de programar, que se denomina ensamblador.

Conclusión:

Al realizar los shaders pudimos darnos cuenta que podemos hacer diferentes manera para realizar un efecto y que cuando combinemos efectos se usa un Add, para dividir un Split, también usamos el Subtract, o un Multiply, que nos permite juntar o dividir características del Main Texture, también los colores y el Albedo, también de como creamos las carpentas para dividir por Textures, Shaders y Animations, creamos los shader y los materiales necesarios para mostrar, cree los shader algunos son de prueba y otros los use de base para los Subgraph para poder combinarlos y lograr el resultado deseado.

Referencias:

- Alapont Granero, S. 2015. Scares for sale: diseño y desarrollo de un videojuego en Unity 3D. Audio e introducción a Leap Motion (Doctoral dissertation).
- Blackman, S. 2013. Beginning 3D Game Development with Unity 4: All-in-one, multi-platform game development. Apress.
- Calabrese, D. (2014). Unity 2D Game Development. Packt Publishing Ltd.
- Card, D., & Mitchell, J. L. 2002. Non-photorealistic rendering with pixel and vertex shaders. ShaderX: Vertex and Pixel Shaders Tips and Tricks, 319-333.
- Caurin J. 2019. Shader. Recuperado de: <https://www.geekno.com/glosario/shader>.
- Dean, J. 2016. Mastering Unity Shaders and Effects. Packt Publishing Ltd.
- Doppioslash, C. 2018. Your First Unity Lighting Shader. In Physically Based Shader Development for Unity 2017 (pp. 51-64). Apress, Berkeley, CA.
- Ebner, M., Reinhardt, M., & Albert, J. 2005, March. Evolution of vertex and pixel shaders. In European Conference on Genetic Programming (pp. 261-270). Springer, Berlin, Heidelberg.
- Editorwp_6720, 2018, 28 marzo. Pixel Shaders y Vertex Shaders. Recuperado de: <https://www.neoteo.com/pixel-shaders-y-vertex-shaders>
- Goldstone, W. 2009. Unity game development essentials. Packt Publishing Ltd.
- Gonzales P. 2015. ¿Qué es un Shader?. Recuperado de: <https://thebookofshaders.com/01/?lan=es>
- Hergaarden, M. 2011. Graphics shaders. Graphics Shaders, 1.
- Lammers, K. 2015. Unity shaders and effects cookbook. Packt Publishing Ltd.
- Lewis, R. R. 1994, May. Making shaders more physically plausible. In Computer Graphics Forum (Vol. 13, No. 2, pp. 109-120). Edinburgh, UK: Blackwell Science Ltd.

Nilsson, M., & Lundmark, A. 2017. 2D Aesthetics with a 3D Pipeline: Achieving a 2D Aesthetic with 3D Geometry.

Paszkiet, S. 2020. Computer Game in UNITY Environment for BCI Technology. In *Analysis and Classification of EEG Signals for Brain–Computer Interfaces* (pp. 101-110). Springer, Cham.

Stam, J. 1999, July. Diffraction shaders. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques (pp. 101-110).

Thorn, A. 2013. Learn unity for 2d game development. Apress.

Thorn, A. (2015). Unity Animation Essentials. Packt Publishing Ltd.

Zucconi, A., & Lammers, K. (2016). Unity 5. x Shaders and Effects Cookbook. Packt Publishing Ltd