

Implementación de lenguaje de Shaders en threeJs

Kevin Badrán, Arturo Pinzon , Hector Gomez
{6000258, 6000335, 6000277}@unimilitar.edu.co
Profesor: Gabriel Avila Buitrago

Aplicación de shaders a nuestro lenguaje de programación con Three.js, para simulación de comportamiento de luz.

Palabras clave: *sombreador, efecto, ShaderMaterial, vértices, fragmentos.*

I. INTRODUCCIÓN

Para nuestro proyecto se va a emplear shaders, los cuales con materiales con sombreado personalizados, donde un sombreado es un pequeño programa escrito en GLSL que se ejecuta en la GPU.

A. Marco teórico

Puede especificar dos tipos diferentes de sombreadores para cada material:

- El sombreador de vértices se ejecuta primero; recibe atributos , calcula / manipula la posición de cada vértice individual y pasa datos adicionales (que varían) al sombreador de fragmentos.
- El sombreador de fragmentos (o píxeles) se ejecuta en segundo lugar; establece el color de cada "fragmento" (píxel) individual que se muestra en la pantalla.

Hay tres tipos de variables en los sombreadores: uniformes, atributos y variaciones:

- Los uniformes son variables que tienen el mismo valor para todos los vértices; los mapas de iluminación, niebla y sombras son ejemplos de datos que se almacenarán en uniformes. Se puede acceder a los uniformes mediante el sombreador de vértices y el sombreador de fragmentos.
- Los atributos son variables asociadas con cada vértice; por ejemplo, la posición del vértice, la cara normal y el color del vértice son ejemplos de datos que se almacenarán en atributos. Solo se puede acceder a los atributos dentro del sombreador de vértices.
- Las variaciones son variables que se pasan del sombreador de vértices al sombreador de fragmentos. Para cada fragmento, el valor de cada variable se interpolara suavemente a partir de los valores de los vértices adyacentes.
- Tenga en cuenta que dentro del propio sombreador, los uniformes y los atributos actúan como constantes; solo puede modificar sus valores pasando valores diferentes a los búferes desde su código JavaScript.



Imagen1. Ejemplo de shaders

II. COMPETENCIAS A DESARROLLAR

- Determinar los materiales a modificar
- Determinar geometrías a crear
- Determinar las cámaras a usar
- Aplicar shaders

III. TRABAJO PREVIO

Para la realización de este proyecto, debemos tener bien claro los conocimientos previos, códigos referentes y materiales a utilizar. adicional a ello un esquema gráfico para plasmar este en el código respectivo.

El Trabajo previo que tenemos dentro del grupo de trabajo es la realización y creación de diversas escenas con el uso de Geometrías, Materiales y Cámaras de igual forma aplicando herencia , eventos del teclado y priorizando la implementación de transformaciones matrices y operaciones dentro del entorno de Three Js.

IV. DESARROLLO DE LA PRÁCTICA

Para el desarrollo de la práctica, el grupo de trabajo se pone de acuerdo para la realización de las estructuras e interacciones que tendrá el usuario, después de tener esto claro es necesario buscar e investigar qué tipo de funciones son las que se trabajarán y usarán.

Saber dónde se ubicarán los shaders y a qué objetos de la escena a crear deben tener necesariamente dichos shaders. Dependiendo del tipo de escena es necesario analizar qué a qué elementos y cómo se les aplicarán alguno de diferentes tipos de sombreadores.

Seguidamente queda la implementación del código,



correcciones de posiciones en la escena y al finalizar la escena implementación de shaders para obtener una mejor visualización y análisis de lo que se obtiene en la escena.

Lo restante es socialización y proyección de la escena, comentarios, correcciones y puntos a resaltar por parte del docente.

V. CONCLUSIONES

Los shaders son un adicional a la creación de una escena, que intenta darle un poco más de realismo, e intensifica los colores de esta, resaltando los puntos claves que pretende dar el creador, con respecto a la iluminación que este usa.

Gracias a que hay dos tipos de shaders, como los de vértices, que manejan el procesamiento de vértices individuales y esto nos permite trabajar en la forma de la geometría. Al igual también están los Shaders de fragmentos, los cuales toman la geometría rasterizada del objeto desde los vértices y le aplican color. Entre estos dos sombreadores podemos influir en cómo se dibujan los objetos tanto en su geometría física como en su color y textura.

REFERENCIAS

- [1] <https://threejs.org/docs/#api/en/materials/ShaderMaterial>
- [2] <https://threejsfundamentals.org/threejs/lessons/threejs-shadertoy.html>
- [3]