

# Preparación para la creación de un Minecraft

## (Septiembre 23 de 2020)

Erick Santiago Robayo, Juan Pablo Jiménez Rodríguez, Gabriel Nicolás Chaves

### I. OBJETIVOS

#### General

- Realizar un juego basado en Minecraft ideal a como nos gustaría que fuera.

#### Específicos

- Dar uso de las herramientas y temas vistos en clase de computación gráfica.
- Utilizar shaders para una experiencia del juego mas realista.

Resumen - Para realizar este proyecto primero hay que tener en cuenta las bases para iniciarlo y sus componentes, se observa a Minecraft como un juego de cubos que permite crear y simular a su manera un tipo de mundo, principalmente suple una idea limitada solo por la mente del jugador, que necesita sus propias reglas para que su mundo funcione. Construido principalmente por cubos, el juego iniciaría con una estructura que solo se limita a primitivos puestos en un espacio que se establezcan como suelo, material, objeto, mineral, creación, etc.

Posterior a esto hay que proponer la diferencia entre cada cubo uno del otro agregándole cierta característica única y perceptible para que funcione el proyecto, siendo así, lo más básico y eficaz para hacer esto son las texturas, una diferencia en cada cubo que hará que un cubo de tierra sea diferente de un cubo de madera.

El siguiente tema a mencionar seria la creación de personaje y su manera de interactuar con el mundo ya planteado, Minecraft siempre ha creado un personaje muy geométrico y primitivo que simula el cuerpo de una persona común y corriente, y su manera de experimentar este juego son los ojos y la vida de este personaje, por ende, se tendría que proponer algo muy similar para seguir la idea de nuestra referencia principal.

### Índice de Términos – Cubos, Minecraft, Primitivas, Texturas.

### II. INTRODUCCION

**M**inecraft es un videojuego de construcción de mundo abierto, el cual se enfoca en la posibilidad que tiene el jugador para explorar, interactuar y modificar un mapa generado dinámicamente hecho en bloques de un metro cúbico. En adicional a los bloques, el entorno con plantas, criaturas e items. Algunas de las actividades en el juego incluye minar menas, pelear con criaturas hostiles y elaborar nuevos bloques y herramientas recolectando varios recursos que se pueden hallar en el juego. El modelo abierto del juego permite a los jugadores crear estructuras, hacer creaciones y obras de arte en diversos servidores multijugador o mapas de un solo jugador. Un juego basico que nos permite despertar la creatividad y que ademas es un juego muy útil para iniciar a los chic@s en el mundo de la programación ya que se ve algo de código Java, especialmente a la hora de modificar las clases de los personajes, crear variables y condicionantes. Los mods, son el camino para enseñarles código de forma sencilla. Además, para jugar en modo multijugador, podemos aprender a crear nuestros propios servidores.

Ademas Minecraft cuenta con la opcion de mejoras graficas que nos permitan una mejor experiencia de juego y a su vez personalizar los modos de juego tambien. Por esta razon daremos uso de shaders para este proyecto y darle un toque mas artistico y personal de un minecraft ideal.

### III. MARCO TEORICO

#### A. Programas

Para la realización de este proyecto, utilizaremos de una herramienta en especifica llamada Notepad++. También conocido como Notepad plus plus, es un editor gratuito de código fuente libre y un sustituto de Notepad que soporta varios lenguajes de programación. Funciona en entorno MS Windows, su uso está regulado por la Licencia Pública General de GNU. Basado en un potente componente de edición llamado Scintilla, Notepad++ está escrito en C++ y

usa directamente la API de Win32 y STL, lo que asegura una gran velocidad de ejecución y un menor tamaño del programa. Mediante la optimización de tantas rutinas como sea posible sin perder la interfaz amigable con el usuario.

Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores.

## B. Bibliotecas Y Plugins

### 1) THREE.JS

THREE.JS, es una librería bastante liviana y muy eficiente para generar y animar gráficos en 3D dentro del navegador, aprovechando las grandes novedades que nos ofrece HTML5 para la generación de contenidos multimedia. Aprovecha tanto las capacidades de HTML5 que es capaz de generar escenas 3D con WebGL, Canvas (2D) y SVG.

*Three.js* no es popular solo porque es simple a la hora de usarlo a través de JavaScript, sino porque tiene un equilibrio perfecto entre el diseño y la programación.

Permite, entre otras cosas, importar archivos 3D a partir de **Blender** o **Maya**, pudiendo generar terrenos u objetos totalmente complejos y de gran calidad.

La librería también incorpora potentísimos *shaders* que se pueden personalizar con **OpenGL Shading Language** (GLSL). Y sin duda incorpora todo lo necesario para crear escenas 3D, como son la posibilidad de manipular luces, cámaras, animar objetos, perspectivas, control de visualizaciones y mucho más.

#### a) ORBITCONTROLJS

Three.js tiene multitud de plugins que facilitan la vida del programador. Uno de estos plugins es OrbitControl. Con cargarlo, asignarlo a la cámara y ponerlo en la función que renderiza, ya podemos coger el control de nuestra cámara. Podremos usar la rueda para acercarnos o alejarnos, y arrastrar para girar la cámara.

## C. Shaders.

El shader es un concepto de desarrollo de videojuegos, que significa sombreador en castellano. El sombreador, o shader, es un programa o aplicación informática que se ejecuta en la tarjeta gráfica o GPU, y que permite realizar las sombras de los gráficos en un videojuego.

Este programa se encarga de realizar los diferentes cálculos de cómo quedarían las sombras si le diera la luz. Teniendo en cuenta las características de cada cuerpo o cada personaje,

calcula cuál es la sombra que se debería generar en la vida real en función de cómo incide la luz en un cuerpo.

Además, es necesario saber que también se utilizan para realizar transformaciones de vértices y coloreados de píxeles con el objetivo de crear algunos efectos especiales como el fuego, la niebla o la iluminación.

Hay distintos lenguajes de programación de alto nivel para ello. De alto nivel significa que tú escribes algo sencillo y el lenguaje, compilador en realidad, ya se encarga de pelearse con el hardware para que haga lo que tú quieres. En este caso las tarjetas gráficas poseen montones de núcleos (pipelines) que ejecutan las tareas que les indicas. Cada uno de esos núcleos ejecutará un shader, pudiendo ser de distinto tipo, como la iluminación, el sombreado, el reflejo en el agua, la niebla que clarea un píxel más cuanto más lejos esté de la cámara, etc.

### 1) Shaders por Hardware

#### a) Vertex shader

Un sombreador de vértices es una función de procesamiento de gráficos que se utiliza para agregar efectos especiales a los objetos en un entorno 3D al realizar operaciones matemáticas en los datos de los vértices de los objetos. Cada vértice puede definirse mediante muchas variables diferentes. Por ejemplo, un vértice siempre se define por su ubicación en un entorno 3D utilizando las coordenadas x, y, y z-. Los vértices también se pueden definir por colores, coordenadas. Los vértices también se pueden definir por colores, texturas y características de iluminación. Los Vertex Shaders no cambian realmente el tipo de datos; simplemente cambian los valores de los datos, de modo que un vértice emerge con un color diferente, texturas diferentes o una posición diferente en el espacio.

#### b) Geometry shader

Entre el vértice y el sombreador de fragmentos hay una etapa de sombreado opcional llamada sombreador de geometría. Un sombreador de geometría toma como entrada un conjunto de vértices que forman una sola primitiva, por ejemplo, un punto o un triángulo. El sombreador de geometría puede transformar estos vértices como mejor le parezca antes de enviarlos a la siguiente etapa del sombreador. Lo que hace que el sombreador de geometría sea interesante es que puede convertir la primitiva original (conjunto de vértices) en primitivas completamente diferentes, posiblemente generando más vértices de los que se dieron inicialmente.

#### c) Fragment Shader

Los sombreadores de fragmentos son técnicamente una etapa de sombreado opcional. Si no se usa ningún sombreador de fragmentos, los valores de color del fragmento de salida tienen valores indefinidos. Sin embargo, los valores de profundidad y estencil para el fragmento de salida tienen los mismos valores que las entradas.

Esto es útil para hacer renderizado donde la única salida útil es la profundidad del fragmento y desea utilizar la profundidad calculada por el sistema, en lugar de alguna otra profundidad. Esta representación de solo profundidad se utiliza para operaciones de mapeo de sombras, así como para optimizaciones de profundidad previa al paso.

#### D. *Shaders en Minecraft*

Minecraft es un juego que lleva bastante tiempo en el mercado, un juego simple y sin impresión visual al espectador y a medida que va pasando el tiempo, su base original se ha quedado atrás. Hoy en día las gráficas de los videojuegos han evolucionado a tal punto que los espectadores quieren ver más realidad. Es por esto, que la gran comunidad de este famoso juego ha querido realizar creativas maneras para mejorar el espacio visual del juego. A continuación, se dedicará un espacio para algunos de los shader más conocidos en la comunidad de Minecraft y unas explicaciones de cómo están generadas.

##### 1) *SEUS (LOS INCREÍBLES SHADERS DE SONIC ETHER)*

SEUS es el paquete de sombreadores de Minecraft de piedra de toque para cualquiera que quiera sentir que ha entrado en un juego completamente diferente. Llamado SEUS para abreviar, este paquete es una modificación de otro paquete de sombreadores muy querido y, como tal, hará que tu mundo de Minecraft se vea tan bien como puedas esperar. Luz natural suave, lluvia que agrega un brillo brillante a cada superficie que toca, nubes generadas por procedimientos y mucho más te esperan. También siempre se actualiza, con una versión reciente que equipa a los jugadores con un efecto de trazado de rayos falso de Minecraft, que es simplemente increíble en acción.

SEUS (Sonic Ether's Unbelievable Shaders) es un paquete de sombreadores para Minecraft que se puede usar con OptiFine o GLSL Shaders Mod (heredado).



SEUS Renewed es una reinención de las versiones heredadas de SEUS que le brinda imágenes de calidad con un rendimiento razonable utilizando métodos de renderización tradicionales basados en rasterización. Hay algunas versiones de desarrollo inéditas de SEUS Renewed, pero la mayor parte de mi tiempo de desarrollo últimamente se dedica al desarrollo de SEUS PTGI.

SEUS PTGI es una versión experimental de SEUS que incluye una implementación de software totalmente personalizada de trazado de rayos que no requiere una tarjeta gráfica RTX y funcionará en cualquier tarjeta gráfica NVIDIA (aunque las tarjetas de gama baja pueden tener problemas de rendimiento. La compatibilidad con AMD aún se está trabajando en). La "PTGI" en el nombre significa "Iluminación global con trazado de ruta", que es la característica principal de este proyecto. También incluye reflejos con trazado de rayos.



##### 2) *SHADERS DE CHOCOPIC13*

Gráficos claros y nítidos con algunos hermosos efectos de agua e iluminación brillante: los sombreadores de Minecraft de Chocopic13 son ciertamente atractivos. Sin embargo, una de las principales razones para revisar este sombreador es que viene en una variedad de versiones según el equipo que tenga, desde extremadamente exigente hasta el nivel de tostadora. El extremo inferior no es tan atractivo, pero casi sin impacto en el rendimiento es difícil de superar.



Chocapic13's Shaders mejora la calidad gráfica de tu Minecraft como cualquier otro Shader, pero Chocapic13's Shaders lo hace tratando de parecer más a la realidad y dar una experiencia muy realista que no dan otros Shader. Las apuestas de sol con este Shader son magníficas, los movimientos de los árboles y caída de las hojas son muy naturales.

#### IV. CONCLUSIÓN

Si queremos realizar un Proyecto interesante, es importante tener muy claro los diferentes conceptos de Shaders, los cuales serán un plus clave que nos permita experimentar fuera de las clases de computación grafica un mundo de creaciones importantes en el mundo de los videojuegos.

Además, es claro que será un trabajo de dedicar un buen tiempo para entregar un trabajo a la expectativa de la clase.

#### RECONOCIMIENTO

Para la documentación de este informe, utilizamos referencias únicamente de internet, que nos permitieron el estudio de los conceptos necesarios para la formulación e interpretación de los mismos.

#### REFERENCES-BIBLIOGRAFIA

- [1] Genbeta, página WEB, Introducción a Three.js, la librería 3D número uno para HTML5 (<https://www.genbeta.com/desarrollo/introduccion-a-three-js-la-libreria-3d-numero-uno-para-html5>).
- [2] Trescomatres, página WEB, THREE.JS – EJEMPLO MOVER CÁMARA (ORBITCONTROL), (<https://blog.trescomatres.com/2018/08/threejs-ejemplo-mover-camara-orbit-control/>).
- [3] NVIDIA, página WEB, Sombreadores de vértices, (<https://www.nvidia.com/en-us/drivers/feature-vertexshader/>).
- [4] Learn OpenGL, página WEB, Geometry Shader, (<https://learnopengl.com/Advanced-OpenGL/Geometry-Shader>).
- [5] OpenGL, página WEB, Shader de fragmentos, ([https://www.khronos.org/opengl/wiki/Fragment\\_Shader#:~:text=A%20Fragment%20Shader%20is%20the,a%20%22fragment%22%20is%20generated.](https://www.khronos.org/opengl/wiki/Fragment_Shader#:~:text=A%20Fragment%20Shader%20is%20the,a%20%22fragment%22%20is%20generated.))
- [6] PCGamesn, página WEB, (<https://www.pcgamesn.com/minecraft/minecraft-shaders-best-graphics-mods>)