

# Presentación de propuesta de proyecto en Computación grafica APLICACIÓN DE PRINCIPIOS FÍSICOS EN MINECRAFT (Noviembre de 2020)

Juan E Gómez, Daniel P Pedroza, Bayrón D Morales

**Resumen** – En 2010 fue liberada en GitHub ‘Threejs’, una biblioteca que utiliza el elemento canvas de HTML5 o WebGL, para mostrar gráficos animados por ordenador en 3D, es una librería de alto nivel, lo que facilita su uso y aplicación

Por otro lado, esta Physijs, una capa superior de la librería ammo.js, que permite la simulación física en un hilo separado, y así evita el impacto en el rendimiento de los cálculos en la aplicación y permite acelerar el tiempo de render.

Se pretende presentar una propuesta para el proyecto de quinto semestre de Computación Grafica en la Universidad Militar Nueva Granada, creando un videojuego de construcción que aplique principios de la física en el propio navegador web, utilizando las librerías Three.js, Physijs y conocimiento en JavaScript.

**Palabras Clave** - JavaScript, Librería., Physijs, Threejs.

**Abstract** - In 2010 was released in GitHub 'Threejs', a library that uses the canvas element of HTML5 or WebGL, to display 3D computer-animated graphics, is a high-level library, making it easy to use and apply

On the other hand, this Physijs, an upper layer of the ammo.js library, allows physical simulation in a separate thread, thus avoiding the impact on the performance of the calculations in the application and allowing to accelerate the rendering time.

We intend to present a proposal for the fifth semester project of Computer Graphics at the Universidad Militar Nueva Granada, creating a construction videogame that applies physics principles in the web browser itself, using the Three.js, Physijs and JavaScript knowledge libraries.

**Keywords** - JavaScript, Library, Physijs, Threejs.

## I. INTRODUCCIÓN

Este documento tiene como fin presentar una propuesta de proyecto en el curso de Computación Grafica, dicho proyecto pretende desarrollar un videojuego interactivo de construcción 3D en navegador web, utilizando la librería Three.js, basada en WebGL, Physijs basado en ammo.js, HTML y JavaScript. Para ello se va a aprovechar la posibilidad que brinda la primera librería para dotar al sistema de un control muy intuitivo y sencillo para el usuario, mediante desplazamiento del ratón y teclas de acceso directo. Adicionalmente aprovechar la interactividad que nos brinda el hecho de trabajar en un entorno 3D y dándole realismo a la escena mediante luces, sombras y principios físicos.

## II. OBJETIVOS

El objetivo de este Proyecto es el de realizar un videojuego de construcción 3D para navegador web, basado en la tecnología WebGL, su posterior evaluación y validación, comprobando que éste funciona tal y como se desea.

El objetivo principal podemos dividirlo en subobjetivos, facilitando el proceso de desarrollo y la organización del proyecto, además de permitir controlar mejor su temporalización. Estos subobjetivos son:

- Investigar acerca de las librerías que se usaran y realizar pruebas de escritorio.
- Integrar las librerías (Physijs & Threejs)
- Recrear una interfaz de usuario a través de HTML y CCS.
- Establecer los requisitos básicos y funcionalidad necesaria, así como los elementos con los que el usuario interactuará.

- Determinar los diferentes controles que el usuario utilizará en el proyecto y cómo estos funcionarán. Es decir, decidir las opciones básicas que el usuario tendrá y aplicarlas junto con los controles de la cámara en nuestro canvas.
- Conseguir crear cubos dinámicamente, partiendo de los ya creados y que su posicionamiento en el canvas sea intuitivo y sencillo.
- Añadir opciones básicas para el usuario, edición de cubos creados y adición de cubos personalizados.
- Crear la interfaz de usuario, con opciones adicionales, cambios de textura de los cubos, posicionamiento de la cámara y menú de ayuda.

### III. DESCRIPCIÓN DEL PROBLEMA

Tal y como se ha dicho anteriormente, en términos sencillos, lo que se va a desarrollar en esta propuesta de proyecto es un videojuego que debe cumplir con una serie de requisitos, como una interfaz de usuario, jugabilidad, rol, etc.... y para dicho fin se van a integrar 2 librerías para JavaScript además de estilos CSS y HTML.

Uno de los principales problemas que se encontraron al momento de realizar las pruebas, fue que al momento de adaptar los principios físicos como colisión, rebote y gravedad por medio de Physijs con las mallas 3d proporcionadas por Threejs es que la primera librería solo recibe modelos 3d importados y objetos de geometrías de Threejs de formas definidas, como BoxGeometry, EsphereGeometry entre otras, como se ve en las figuras 1 y 2, por lo que si se quisiera trabajar con nubes de puntos, o mayas de polígonos, se tiene que optar por otras alternativas a Physijs.

```
plane_geom = new THREE.BoxGeometry(100,2,100,10,10);
plane_mat = new THREE.MeshLambertMaterial({color: 0xeeeeee});
plane = new Physijs.BoxMesh(
    plane_geom,
    Physijs.createMaterial(plane_mat, .4, .8),
    0);
plane.rotation.z = -tiltAngle;
plane.__dirtyRotation = true;
```

Figura 1 Definición de un objeto de un cubo sujeto a leyes físicas

```
ball = new Physijs.SphereMesh(
    new THREE.SphereGeometry(Math.random() * (4 - 1) + 1, 16, 16),
    Physijs.createMaterial(new THREE.MeshLambertMaterial({color: 0xff0000, reflectivity: .8}),
    .4, .6),
    1);
```

Figura 2 Definición de un objeto de una esfera sujeta a leyes físicas

### IV. DESCRIPCIÓN DE LA PROPUESTA

Como se ha venido mencionando a lo largo del documento se propone implementar mediante el uso de JavaScript y

algunas librerías un videojuego tipo Minecraft, en donde se tenga la misma metodología de construcción de un mundo a partir de bloques, mismos que representan múltiples materiales, y que su comportamiento depende de este último.

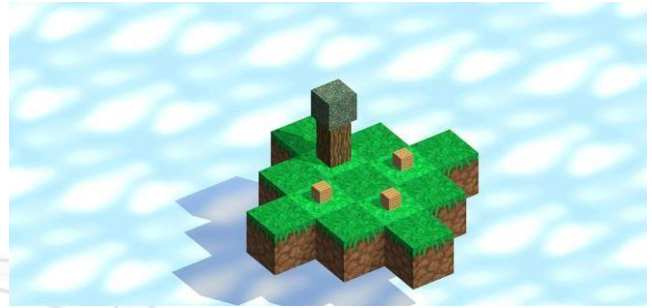


Figura 3 Propuesta mundo

Además, se plantea la posibilidad de que el jugador pueda crear estructuras, pero que estas obedezcan a leyes físicas, en donde si por ejemplo un cubo o una serie de cubos sobrepasan su capacidad de carga la estructura en cuestión colapse, o si se esta construyendo una estructura como un puente o un techo, si distribución de cargas no es la correcta, ocurrirá lo mismo.

En la siguiente secuencia de imágenes se muestra como sería el comportamiento del video juego.

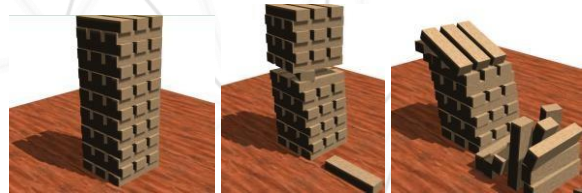


Figura 4 Ejemplo

Este es un juego de Jenga realizado con las mismas librerías propuestas en este documento. Puede verlo en <https://chandlerprall.github.io/Physijs/examples/jenga.html>

Cada tipo de bloque tendrá propiedades y comportamientos diferentes.

En esta propuesta se plantea tener 3 tipos de cubos, metal, madera y tierra.

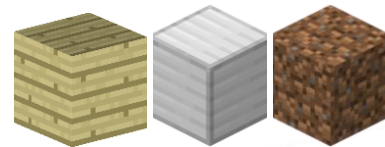


Figura 5 Bloques propuestos

### V. CONCLUSIONES

#### VI.

Al decir seguir este proyecto se esta aceptando un gran reto, ya que además del complejo algoritmo que se plantea, es necesario utilizar texturas, estructuras HTML y CSS.

A la fecha de la realización de este documento no se encontró ninguna forma de integrar THREEJS y PHYSIJS

Cuando se comenzó a plantear el proyecto se constato que los recursos necesarios como las librerías y sus diferentes aplicaciones eran necesarias, y es por esta razón que después de una búsqueda exhaustiva se encontraron dichas herramientas que nos ayudaron a la realización del proyecto, tales como:

- Pointer Lock Point
- Life Server Visual

Que ayudar a programar de manera sencilla tanto el movimiento como las cámaras y ciertos asuntos directos con la programación de las texturas, como las mismas físicas aplicadas al programa que permiten el choque de elementos y la creación de los mismos como la misma seleccione los objetos un tema que presento complicación tanto en la ejecución como en el planteamiento del proyecto.

Afortunadamente como se había dicho gracias a estas librerías y a conocimiento que se encontró en la red (el cual esta bien citada y referenciados al final de este documento con sus respectivos links), no se busco lucrarse de los dichos recursos solo usarlos para llegar a un fin.

Un Aspecto importante que se tuvo en cuenta fue las texturas, ya que como el objetivo del proyecto era que se parecía a minecraft, no obstante en el grupo se opto por tomar las físicas e interacción con el jugador lo que hacia que en las opciones se inclinara mas por utilidad y la calidad del programa en si que de su mismo aspecto, ya que una vez probada esta el aspectos son detalles triviales que se pueden pulir con un poco de trabajo.

```
<script
src="https://threejs.org/examples/js/controls/OrbitCon
trols.js"></script>
```

Dicho ejemplo es de una librería que se uso en el proyecto, y se puede ver en los enlaces del final. Y sobre todo dicho ejemplo es para el control de la cámara sin necesariamente usar matrices facilitando el trabajo y haciendo que los archivos usados se disminuyan.

No obstante, si no se tiene una conexión con el servidor o se llama de manera correcta este mismo no funcionara y mandara error puesto que la información necesaria para que funcione están en el mismo, pero no es un problema si se inicia desde el visual estudio

En dicho código se puede ver como se llaman los objetos que se van usar para la programación del movimiento y las cámaras en si.

```
let camera, cameraOrtho, scene, renderer, insetWidth,
insetHeight, mesh;
```

Para después con la misma librería poder darles un uso función posiciones y variables adecuadas para el trabajo.

```
camera.position.z = 1;
```

```
cameraOrtho = new THREE.OrthographicCamera( -
0.5, 0.5 , 0.5, -0.5, 0.01, 10 );
camera.add( cameraOrtho );
```

```
scene = new THREE.Scene();
scene.add( camera );
```

El Pointer Lock Control es una de las herramientas que mas trabajo facilito además de que es fácil de usar no requiere mucho trabajo y es una librería gratuita de GitHub, esta misma permite hacer juegos en 3d con movimiento de personajes en su debido entorno lo que realmente ahorro horas de trabajo haciendo el movimiento del mismo y asegurarse de que quedara bien ya que se delmito a hacer ajustes al código para que el movimiento quedara de manera natural y para nada incomodo al menos para la vista del jugador.

Un ejemplo claro es:



Imagen 6: Entorno simulado

Lo que se buscaba era esto pero era onbio que estaba delimitado por los movimientos del jugador o de la persona que estaban manipulando por lo que los controles que se implementaron cayeron de manera perfecta en su lugar.

#### Bibliografía:

- THEFIVEPLANETS.ORG. 2020. THE FIVE PLANETS – DISEÑO DE JUEGOS 3D PARA WEB. [ONLINE] AVAILABLE AT: <HTTP://THEFIVEPLANETS.ORG/> [ACCESSED 23 SEPTEMBER 2020].
- CHANDLERPRALL.GITHUB.IO. 2020. JENGA - PHYSIJS. [ONLINE] AVAILABLE AT: <HTTPS://CHANDLERPRALL.GITHUB.IO/PHYSIJS/EXAM PLES/JENGA.HTML> [ACCESSED 23 SEPTEMBER 2020].
- HTTPS://THREEJS.ORG/
- HTTPS://GITHUB.COM/CHANDLERPRALL/PHYSIJS/WIKI/

### BASIC-SETUP

- <https://jsfiddle.net/f2Lommf5/11653/>
- <https://github.com/mrdoob/three.js/blob/master/examples/jsm/controls/PointerLockControls.js>
- <https://threejs.org/docs/#examples/en/controls/PointerLockControls>

