Logotipo

Descripción generada automáticamente

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

ingeniería en Ciencias de la Computación

Graficación - Primavera 2023

Evaluación 2: Sistema Solar Alternativo y Texturas

Fernando José Gil Uribe

fernando.gilu@alumno.buap.mx

10 de marzo 2023

## Resumen

El resultado final de esta aplicación es un prototipo funcional que combina HTML, CSS y JavaScript para crear una experiencia interactiva y atractiva. La aplicación muestra una representación inventada de un sistema solar imaginario. Se han incluido secciones de código documentadas, lo que facilita la comprensión del desarrollo y permite la expansión de este en el futuro. Además, se describen los términos técnicos utilizados en el desarrollo para mejorar la comprensión del usuario. Las capturas de pantalla muestran el éxito de la aplicación en su objetivo principal de ofrecer una experiencia gráfica interactiva y atractiva.

## Introducción

En este reporte se presenta el desarrollo de una página web que utiliza Three.js para crear una simulación interactiva de un sistema solar imaginario. La simulación está diseñada para permitir a los usuarios explorar y los planetas y el sol de este sistema solar de manera visualmente atractiva. La página web fue desarrollada utilizando Three.js para la visualización 3D y HTML, CSS y JavaScript para la interfaz de usuario y la lógica de la aplicación. A lo largo del reporte se describen los detalles técnicos de la implementación y se discuten las lecciones aprendidas durante el proceso de desarrollo. Además, se presentan algunas posibles mejoras y extensiones futuras para la página web.

## Texturizado con ThreeJS

El texturizado en Three JS es relativamente fácil, primero se debe obtener la textura en archivo de imagen (en este caso se usan jpg) y guardarla en los archivos de la página, luego debemos crear un objeto TextureLoader que se encargará de cargar la textura a partir de la textura previamente guardada, posteriormente se usa la función *load()* perteneciente al objeto TextureLoader, de esta forma ya se habrá cargado la imagen al programa, luego se crea un objeto como normalmente se haría, creación de la geometría, luego declaración de un material añadiendo el mapeado del objeto (la textura) como el objeto de la textura cargada, en el caso de objetos planos se puede añadir la opción de “side : THREE.DoubleSided” para que la textura se vea por ambos lados del objeto.  
El código completo de texturizado puede ser revisado tanto en este proyecto y de una forma alterna en el siguiente repositorio: <https://github.com/josdirksen/learning-threejs/blob/master/chapter-10/09-canvas-texture.html>

## Explicación del código

Texto

Descripción generada automáticamente

Se crean algunas variables necesarias para el programa.

Texto

Descripción generada automáticamente

Función para generar una lista de coordenadas para x,y,z de una circunferencia

Texto

Descripción generada automáticamente

Función para generar una lista de coordenadas en x,z para una trayectoria en espiral sobre una trayectoria elíptica

Texto

Descripción generada automáticamente

Asignaciones y funciones para el manejo de la cámara, los controles de cámara y la escena.

Texto

Descripción generada automáticamente

Función de utilería para actualizar la posición y objetivo de la cámara

Texto

Descripción generada automáticamente

Funciones para el GUI que muestran a cada planeta.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Creación del GUI y asignación de funciones para cada planeta

Texto

Descripción generada automáticamente

Creación del sol y asignación a la escena, el sol se crea como ParametricGeometry para que se pueda crear como la tira de mobius.

Texto

Descripción generada automáticamente

Creación de mercurio como una esfera y con su propia textura.

Los demás planetas tienen creaciones similares a mercurio, variando únicamente la textura y la geometría.

Texto

Descripción generada automáticamente

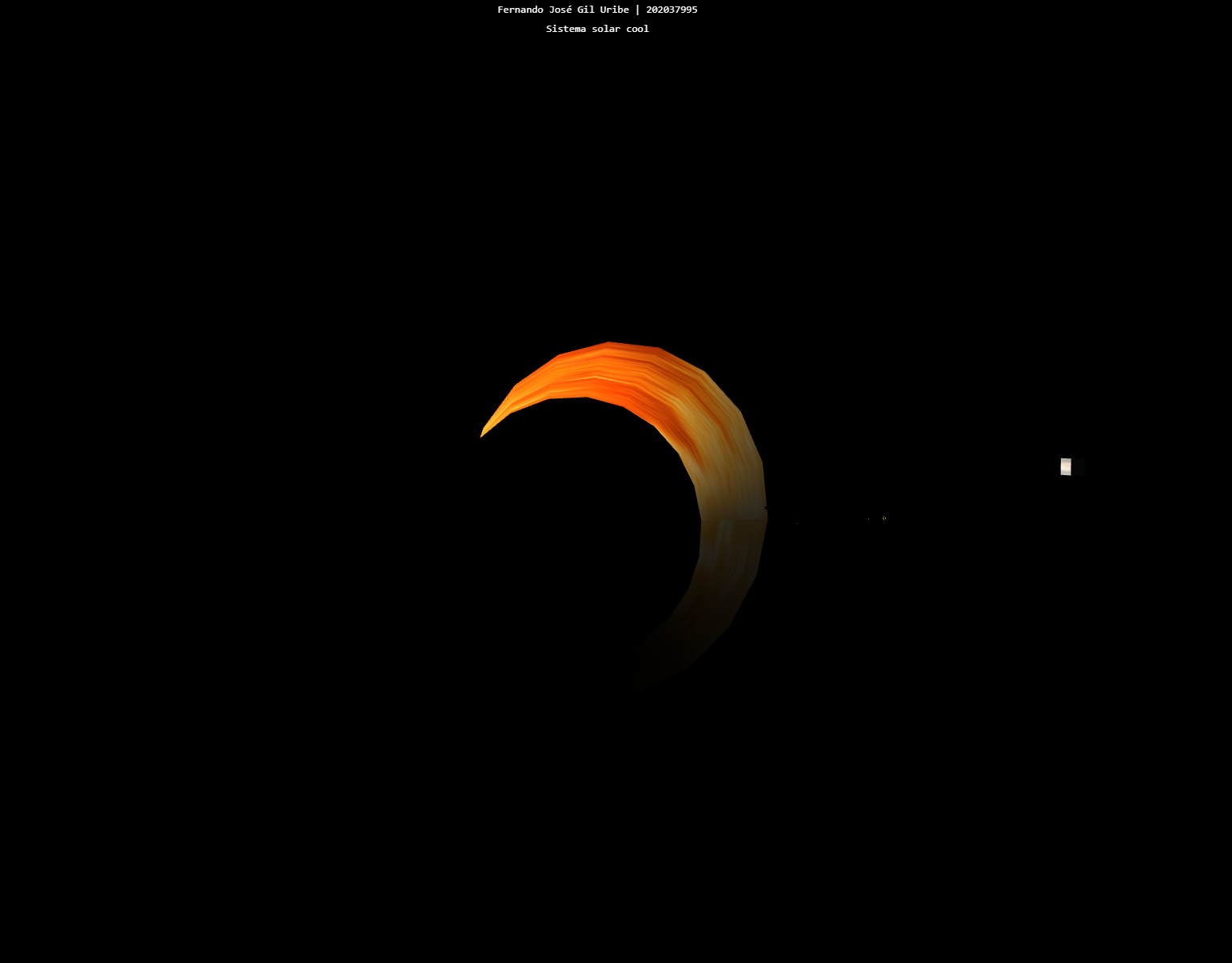
Creación de las luces, una luz central y otras *n\_lights* en ejes Z y Y.

Texto

Descripción generada automáticamente

Procesamiento de las animaciones de cada objeto dentro de la función render.

## Resultados



Aquí se puede ver el sol y varios planetas pequeños por la comparación de tamaño.



Aquí se ve la tierra en aumento porque se tiene la posibilidad de visualizar a un planeta en específico.

Dibujo de una persona

Descripción generada automáticamente con confianza media

Este es saturno mostrado como una botella de Klein, igualmente está rotando alrededor del sol.

## Conclusión

El trabajo que este programa implicó fue útil para entender cómo texturizar un objeto tridimensional para una página web, igualmente pude aprender cómo manejar los objetos y sus trayectorias para que varios objetos se movieran a la vez. También aprendí a utilizar las luces dentro del programa y hacer que se refleje la luz sobre los objetos de forma correcta incluso con múltiples fuentes de luz. En general el proyecto fue útil para el manejo de objetos como tal a diferencia de la tarea anterior que se centraba más al manejo de cosas no tan visuales.

## Referencias

<https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>

<https://threejs.org/docs/index.html?q=sphe#api/en/geometries/SphereGeometry>

<https://threejs.org/docs/index.html?q=text#manual/en/introduction/Creating-text>

<https://github.com/mrdoob/three.js/>