PROYECTO FINAL COMPUTACIÓN GRÁFICA

Juan Pablo Bermúdez Gómez 6000198
Laura Alejandra Hurtado Ladino 6000279
Ruben Felipe Contreras Guzman 6000312
Universidad Militar Nueva Granada
Junio 3 de 2020
Tercer Corte

***Resumen*—Este documento presenta el proyecto final de la materia de computación gráfica del tema investigado (JSON, Grafos y su aplicación) para el diseño de un escenario y así poder tener un mejor desarrollo de la interfaz del proyecto.**

I. Introducción

Se presenta la entrega final del escenario, diseño e implementación del espacio trabajó en el proyecto, recordemos que se trata de una especie de barrio de una ciudad para generar una ruta o más bien un recorrido de un punto a otro con la manipulación manual del usuario y de quien lo maneje es el responsable de hacer una retroalimentación digital de tal recorrido [8]. Esto se realiza mediante conceptos y aplicaciones nuevas que hemos investigado, estos conceptos son el Algoritmo de Dijkstra y el Algoritmo de Prim que a continuación se va a explicar de que se tratan y qué hacen en el marco teórico del presente documento.

II. Marco Teórico

En esta sección se pretende mostrar bocetos y diseño e implementación del proyecto teniendo en cuenta toda la información anterior, este proyecto se tratará de calculo de rutas, explicando como ir de un punto a otro punto de un lugar especifico como por ejemplo: en un centro comercial una pantalla muestra como llegar de tu ubicaciónn actual a una tienda de este centro comercial o la ruta para llegar de la casa a alguna droguería o supermercado cercano. Para ello necesitamos saber y entender qué son los algoritmos de Dijkstra y Prim y para qué sirven.

- Algoritmo de Dijkstra es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de vértices en un grafo con pesos en cada arista. Posee múltiples aplicaciones donde se aplican los grafos, es necesario conocer el camino de menor costo entre dos vértices dados: Distribución de productos a una red de establecimientos comerciales, distribución de correos postales y Sea G = (V, A) un grafo dirigido ponderado.

Figura 1.

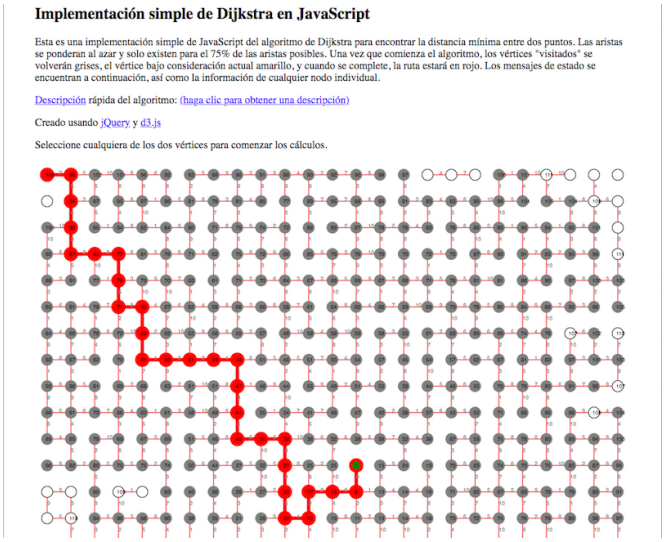

Figura 2.

- El problema del camino más corto de un vértice a otro consiste en determinar el camino de menor costo, desde un vértice u a otro vértice v. El costo de un camino es la suma de los costos (pesos) de los arcos que lo conforman.
- Hablando un poco de sus características éste es un algoritmo greddy, es decir, una estrategia de búsqueda por la cual se sigue una heurística consistente en elegir

la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima.
Trabaja por etapas, y toma en cada etapa la mejor solución sin considerar consecuencias futuras.

- Algoritmo de Prim
  es un algoritmo perteneciente a la teoría de los grafos para encontrar un árbol recubridor mínimo en un grafo conexo, no dirigido y cuyas aristas están etiquetadas.
  En otras palabras, el algoritmo encuentra un subconjunto de aristas que forman un árbol con todos los vértices, donde el peso total de todas las aristas en el árbol es el mínimo posible. Si el grafo no es conexo, entonces el algoritmo encontrará el árbol recubridor mínimo para uno de los componentes conexos que forman dicho grafo no conexo.
- El algoritmo incrementa continuamente el tamaño de un árbol, comenzando por un vértice inicial al que se le van agregando sucesivamente vértices cuya distancia a los anteriores es mínima. Esto significa que en cada paso, las aristas a considerar son aquellas que inciden en vértices que ya pertenecen al árbol.
  El árbol recubridor mínimo está completamente construido cuando no quedan más vértices por agregar.
  El escenario posee los siguientes componentes:
- Modelado de edificios con sus respectivas ventanas.
- Modelado de la carretera.
- Diferentes tiendas y puntos de encuentro.
- Cámara en primera persona.
- Diferentes materiales y colores del montaje.

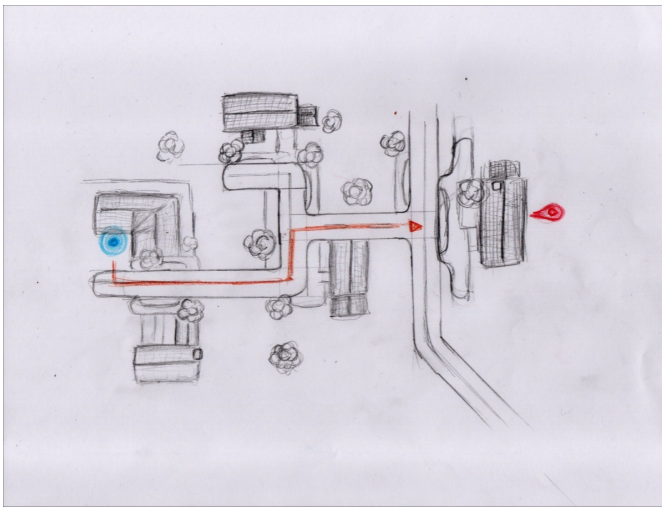

Figura 3.

III. Uso de grafos y nodos

Lo que se realizó en la escena fué diseñar dos caminos, la ruta 1 representa el color rojo que como se muestra en la siguiente figura describe el camino más largo. Por otro lado la ruta 2 representando el color azul muestra el camino más corto que se pretende guiar al usuario para así poder visualizar la unión de los nodos. Los nodos están hechos con esferas uniéndolas con líneas representándolas como un egde.

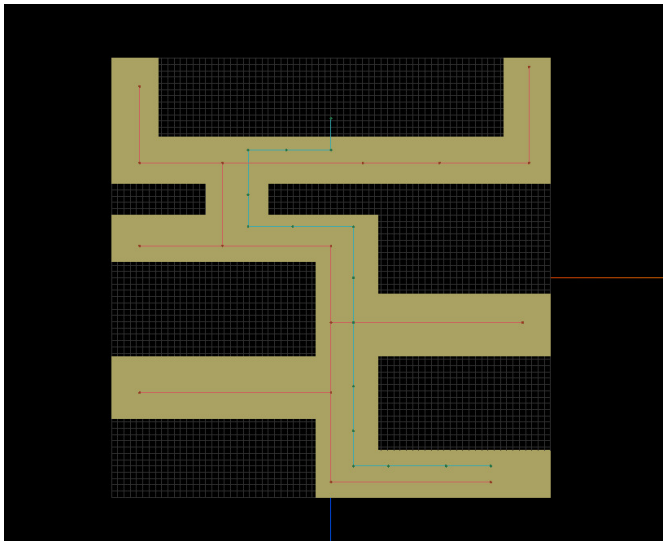

Figura 4.

IV. Diseño del Proyecto

Una de las cosas importantes que se va a tomar en cuenta, es la creación de la escena en la cual se realizara el calculo de la ruta, para eso se desarrollara el lugar en 3D en el cual la persona podrá observar los pasos a seguir para llegar a su destino, un ejemplo cercano es en la siguiente imagen (fig 4) basada en un video sobre JSON en three.js en el cual se observa una escena selva tica, donde la persona va recorriendo cada parte de esta.

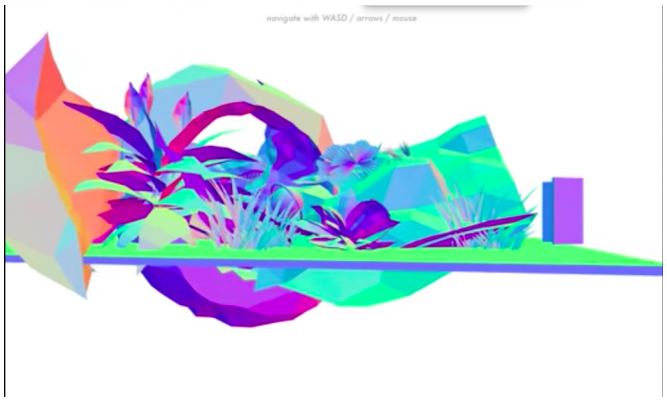

Figura 5.

Este v ıdeo es una referencia importante para empezar a disen ar el espacio que la persona va a observar, pero en este caso se tratara de un escenario urbano donde explicara la ruta para ir de un punto a otro. Se creo una escena parecida a una zona urbana,donde se crearon las casas, la carretera y donde la persona sabra como llegar de un punto a otro.

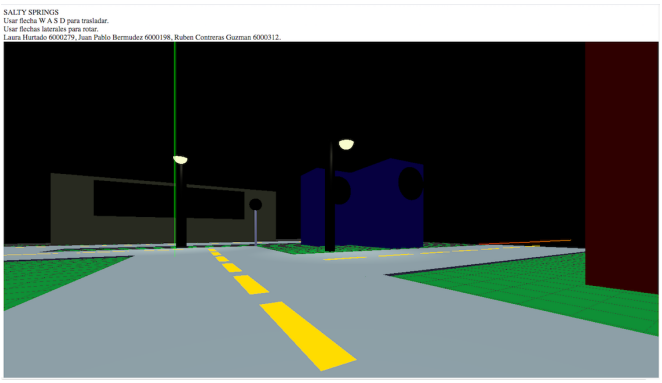

Figura 6.

V. Conclusiones

- Para la creaciónn de modelos THREE.js tiene contemplado la utilización n de modelos de software de creación de contenido y para ello ha desarrollado un plugin para exportarlos desde Blender, Maya o Max a formato JSON que pueden ser importados directamente al entorno THREE.js. Estos plugin pueden ser obtenidos gratuitamente en GitHub.
- Debido a la no estandarizacio n del concepto y la va- riabilidad entre diferentes programas y lenguajes de programacio n 3D, es posible concluir que la direccio n de los ejes x, y, z en three.js es causa comu n de confusio n. En este caso (tomando como referencia un monitor de computadora de escritorio en posicio n vertical) el eje X crece positivamente hacia la derecha del usuario mirando la pantalla, el eje Y crece positivamente hacia arriba y el eje Z crece positivamente hacia afuera de la pantalla (hacia el usuario).

VI. Referencias

[1] https://www.youtube.com/watch?v=4njnviuvt1Q
[2] https://threejs.org/docs/api/en/lights/AmbientLight
[3] Threejs.org. (2020). three.js docs. [online] Available at: https://threejs.org/docs/api/en/loaders/ObjectLoader [Accessed 4 Mar. 2020].
[4] https://thefiveplanets.org/blog/three-js-pimeros-pasos/
[5] https://threejs.org/docs/examples/en/loaders/OBJLoader
[6] Tutorial d3.js - Jortilles. (2020). Retrieved 28 April 2020, from http://blog.jortilles.com/tutorial-d3-js/
[7] https://www.ecured.cu/Algoritmo$_{d}e_{D}ijkstra$
[8] http://bl.ocks.org/sdjacobs/3900867adc06c7680d48
[9] https://www.nealbohling.com/code/dijkstra/