

Proyecto Web GL

Santiago Gómez(6000331), Brayan Contreras(6000333), Edwin Acosta(600324)

Resumen: En el siguiente documento vamos a explicar cómo se realizó el proyecto en Web GL, el cual consiste en crear un algoritmo de relleno para cualquier polígono y adicionalmente se explicará cuáles fueron los scripts y herramientas principales para la elaboración del mismo.

Introducción

Para la finalización del proyecto se empleó un algoritmo de relleno el cual nos da la posibilidad de colorear las figuras básicas geométricas, que en este caso se pueden realizar por la unión de líneas para generar polígonos, rectángulos, triángulos, entre otras, además de círculos, los cuales son creados predefinida mente.

Para hacer el algoritmo de relleno nos basamos en el algoritmo Boundary Fill, para crear las líneas usamos el algoritmo DDA y para el círculo creamos un algoritmo propio.

Lo primero que vemos al abrir el proyecto en el navegador es una caja de opciones la cual nos brinda la posibilidad de elegir entre dos acciones, una es dibujar y la otra es rellenar, la acción dibujar nos permite crear líneas y círculos y por otro lado en la acción rellenar nos permite con un color establecido en la caja de color de la parte posterior hacer el respectivo coloreado de la figura creada en el programa.

Para llevar a cabo el proyecto se implementaron diferentes scripts los cuales se encuentran en la carpeta JS y consisten en:

- **Fill.js:** En este script se encuentra toda la lógica que realiza el algoritmo de relleno, para esto como ya habíamos mencionado, utilizamos el algoritmo llamado Boundary

fill el cual nos permite hacer el coloreado de una figura cerrada que tiene pintada su

frontera de un color determinado. El trata de colorear los cuatro píxeles que rodean al que es pasado como argumento. Esto genera una llamada a la función por cada píxel interior del polígono.

- **glUtils.js:** Se utiliza para la manipulación de matrices y posee utilidades para que el programa funcione.
- **index.js:** Posee la inicialización del contexto gl en el canvas y funcionalidades para la creación de la escena.
- **object.js:** Se encuentran los objetos línea y círculo que se implementan en el proyecto
- **signals.js:** Se utiliza para cargar los archivos fragment.glsl y vertex.glsl los cuales son cargados mediante un protocolo XML request.
- **uiUtils.js:** Se utiliza para poder manejar de manera adecuada las coordenadas las cuales fueron utilizadas mediante decimales.

También se utilizaron dos archivos con extensión. glsl (fragment y vertex) los cuales sirven para cargar parte de la configuración básica que necesita WebGL

Por último, en el index.html se llaman todos los scripts explicados anteriormente y se crean las acciones, desde este archivo se ejecuta todo el programa.

Consideraciones:

- Al abrir el archivo en el navegador como un proyecto local la consola nos reporta un error ya que este archivo HTML se abre localmente e intenta acceder a un archivo del sistema, pero el navegador no permite esto ya que sería una falta en la seguridad por lo que solamente

podremos abrir el proyecto en un servidor de aplicaciones para que de esta forma se pueda ejecutar de manera correcta. Usamos GitHub para abrir el programa y por medio del siguiente link podrán acceder a él para que el programa funcione correctamente.

- <https://u6000324.github.io/Computer-graphics---Projects-/webGl/proyecto/>
- Al usar el algoritmo Boundary Fill, se consumen muchos recursos del PC ya que es un algoritmo recursivo por lo tanto si se rellena una figura muy grande, tardará mucho en pintarse o incluso el navegador podrá cerrarse.
- A la hora de rellenar la figura creada, el color debe ser el mismo al de la frontera o trazo de la figura.

REFERENCIAS

- [1] <https://threejs.org/>
- [2] <https://www.hostinet.com/formacion/panel-alojamiento/que-es-un-index-html/>
- [3] <http://cannes.itam.mx/Alfredo/Espaniol/Cursos/Grafica/Relleno.pdf>
- [4] <https://www.geeksforgeeks.org/boundary-fill-algorithm/>
- [5] https://developer.mozilla.org/es/docs/JavaScript_code_modules/Using
- [6] https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object