



INTRODUCCIÓN

A través de este proyecto se integran herramientas gráficas tales como GLUT y OpenGL para implementar una aplicación gráfica 3D en donde se pueda simular un laberinto, este se dibuja por medio de la interpretación de una matriz, se utilizan shaders e iluminación para poder darle una buena perspectiva de el punto de vista del jugador, el cual se presenta a través de una vista de tipo FPS.

OBJETIVO

El proyecto tiene como objetivo crear un escenario o laberinto, donde por medio del teclado el jugador puede explorar éste y llegar a su salida. Existen tres vistas: una vista GUI que modifica el laberinto, una aérea para mostrar la posición actual del jugador y otra en primera persona.

Arquitectura

Game_driver

En sí éste es el punto central del programa, cuenta con diferentes variables para el ángulo de la cámara y sus vistas, pero nombraremos lo más importante, ésta se compone de una ventana principal con tres sub-ventanas, la primera para el menú interactivo donde se deciden los temas del laberinto y su nivel de dificultad, la segunda para una vista aérea de éste y la tercera para una vista en primera persona.

Maze

En esta clase se contiene la matriz del laberinto. Además éste dependiendo del tema, que predefinido o escogido por el usuario, aplica la tonalidad de luz y colores correspondientes. También revisa si el próximo movimiento de player es válido.

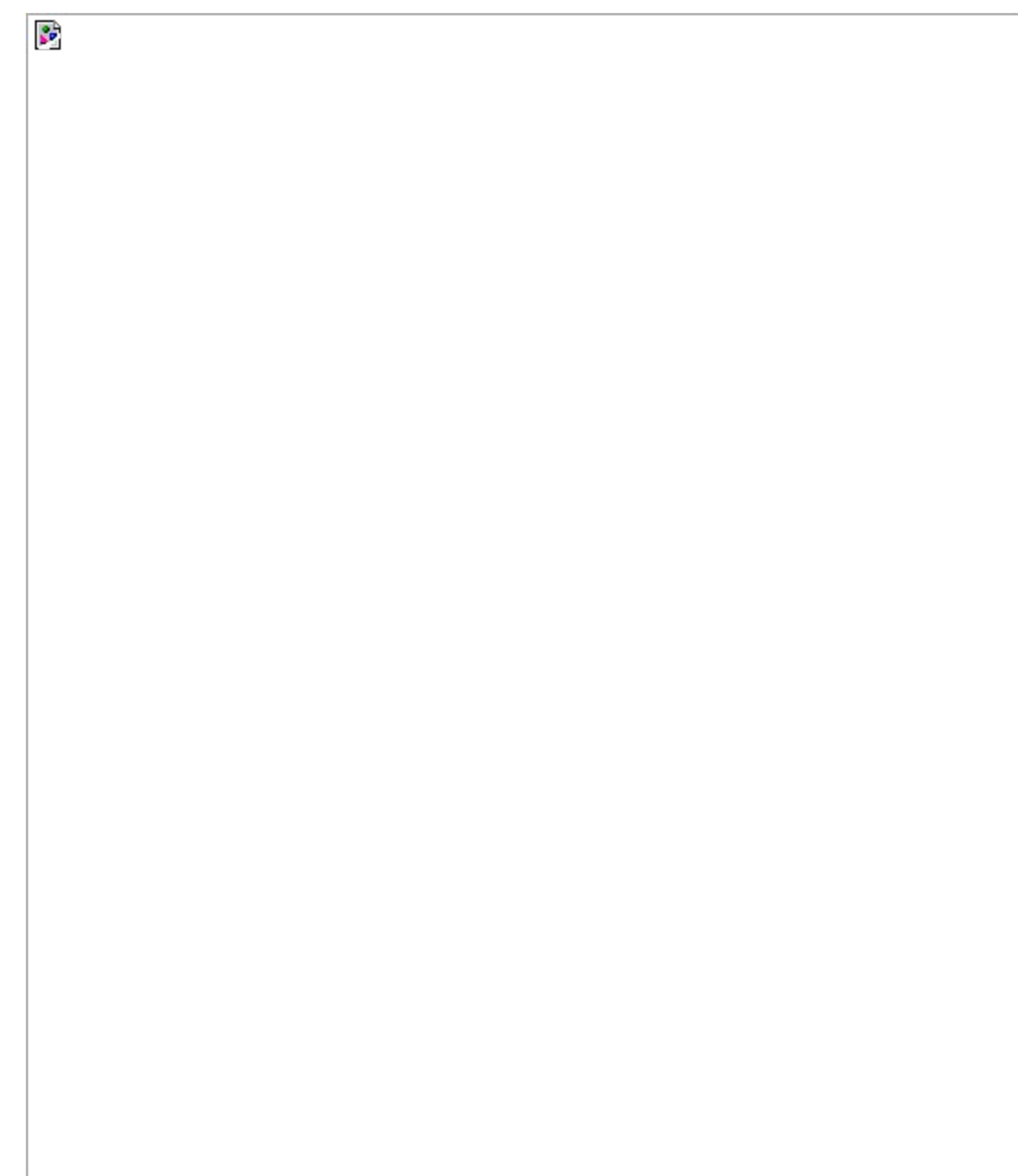
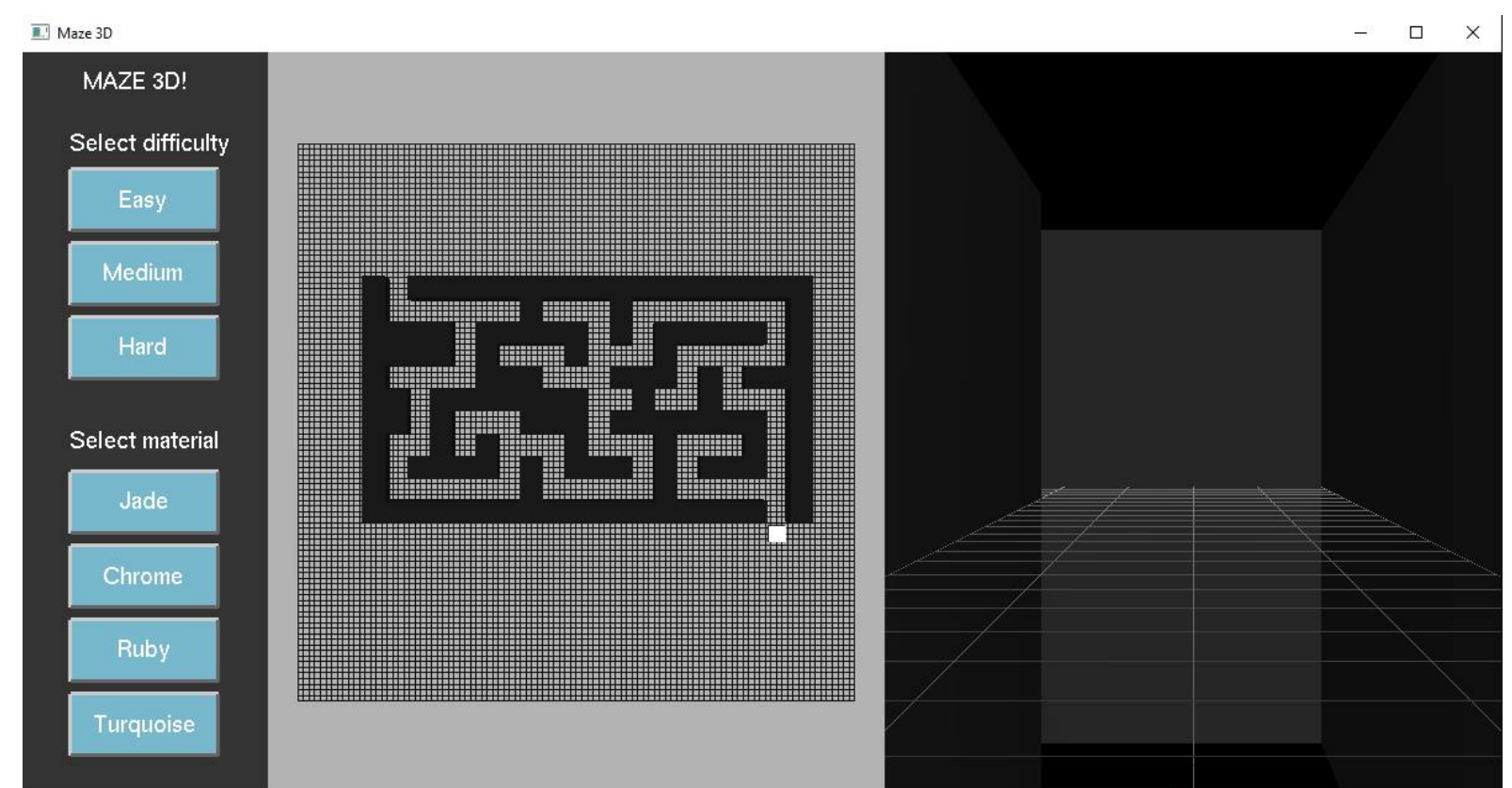
Player

La clase representa al jugador, sirve para guardar en sus variables la dirección en la que se dirige y la posición en la que se encuentra, cuenta con métodos para girar a izquierda y/o derecha y uno para moverse al frente.

RESULTADOS

La matriz de laberinto actual se tiene como propiedad de la clase Maze y se reconfigura copiando algún laberinto almacenado cada vez que se da un click en los botones de dificultad. El material del laberinto se reconfigura antes de pintar según la selección del botón que se haya hecho. El jugador sólo es capaz de moverse por caminos del laberinto, lo cual impide que atravesase paredes o se mueva libremente después de terminar el laberinto.

• IMÁGENES



CONCLUSIONES

Nos agradó bastante el poder integrar diferentes conceptos y mismas prácticas del curso. De hecho nos complicamos un poco al pensar cómo integrar las partes de opengl para la comunicación entre clases, lo cual resolvimos posteriormente. Consideramos que el proyecto quedó simple pero eficiente.

BIBLIOGRAFÍA

1. OpenGL.(n.d.) RobTheBlok. 2016. Recuperado de https://nccastaff.bournemouth.ac.uk/jmacey/RobTheBlok/www/opengl_programming.html
2. The Industry's Foundation for High Performance Graphics. (n.d.). Retrieved April 27, 2016, from <https://www.opengl.org/documentation/>
3. Palomino, L. (n.d.). Presentaciones y Programas del M. Luis Palomino. Retrieved April 22, 2016, from <http://miscursos.itesm.mx/webapps/portal/frameset.jsp?url=/webapps/blackboard/execute/launcher?type=Courseid=3406801url=>



Página de internet:

<https://github.com/alvarolaguna/MAZE-3D>