**ALX THE MAZE PROJECT**

**Header Files:**

1. **maze.h**

* **Purpose:**
  + Header file containing macros, function declarations, and external variables for the maze project.
* **Contents:**
  + Macros defining screen dimensions, texture details, map dimensions, and others.
  + Includes necessary libraries (**SDL**, **stdio.h**, **stdbool.h**, etc.).
  + Function declarations for **parseMap**, **loadTextures**, **initSDL**, **updateRenderer**, **closeSDL**, **input**, **quit**, **raycaster**, **renderWalls**, **renderBGFlat**, and **renderBGTex**.
  + Declaration of external variables, including the window, renderer, texture, buffer, tiles, position (**pos**), direction (**dir**), camera plane (**plane**), and frame time (**time**).

2. **global.h**

* **Purpose:**
  + Header file defining global variables used across multiple files in the project.
* **Contents:**
  + Extern declarations for global variables: **window**, **renderer**, **texture**, **buffer**, **tiles**, **pos**, **dir**, **plane**, and **time**.

3. **structs.h**

* **Purpose:**
  + Header file defining a structure for 2D point coordinates.
* **Contents:**
  + Definition of the **point\_t** structure with **x** and **y** coordinates of type double.

**Source Files:**

1. **parseMap.c**

* **Purpose:**
  + Source file containing the implementation of the **parseMap** function to parse a 2D array from a file to create a maze.
* **Contents:**
  + Includes necessary libraries.
  + Implementation of **parseMap** function that opens a file, reads its content, and creates a 2D maze array.

2. **loadTextures.c**

* **Purpose:**
  + Source file containing the implementation of the **loadTextures** function to load textures from files.
* **Contents:**
  + Includes necessary libraries.
  + Implementation of **loadTextures** function that loads textures into a 3D array (**tiles**) based on the map name.

3. **render.c**

* **Purpose:**
  + Source file containing functions to render walls, floor, and ceiling.
* **Contents:**
  + Implementation of **renderWalls** function to draw wall slices to the buffer.
  + Implementation of **renderBGTex** function to draw floor and ceiling with textures.
  + Implementation of **renderBGFlat** function to draw ceiling and floor for flat maze.

4. **SDL.c**

* **Purpose:**
  + File containing functions for initializing, updating, and closing SDL.
* **Contents:**
  + Implementation of **initSDL** function to initialize SDL, create a window, renderer, and texture.
  + Implementation of **updateRenderer** function to update the renderer with the buffer content.
  + Implementation of **closeSDL** function to destroy texture, renderer, and window.

5. **input.c**

* **Purpose:**
  + File containing functions to handle user input.
* **Contents:**
  + Implementation of **input** function to check user input for movement.
  + Implementation of **quit** function to check if the user quits.

6. **raycaster.c**

* **Purpose:**
  + File containing the main raycasting logic to render walls, floor, and ceiling.
* **Contents:**
  + Implementation of **raycaster** function to cast rays, calculate intersections, and render walls.
  + Uses **renderWalls** to draw wall slices.
  + Uses **updateRenderer** to display the rendered content.

**Main Function - main.c**

* **Purpose:**
  + The main function that executes the game loop, handles user input, and orchestrates the rendering process.
* **Contents:**
  + Initialization of global variables.
  + Parsing command line arguments for map selection and texture usage.
  + Initializing SDL, parsing the maze, and loading textures.
  + Running the game loop, calling **raycaster** for rendering and **input** for user interaction.
  + Exiting the program when the user quits.

**Compilation:**

* This Makefile is responsible for compiling the Simple 3D Maze project. It specifies the compiler, source files, compilation flags, and linker flags.

The **Variables:**

* **CC**: Compiler used (gcc).
* **SRC**: Source files location (all files in the **src** directory).
* **NAME**: Executable name (maze).
* **RM**: Command to remove files.
* **CFLAGS**: Compiler flags for optimization, debugging, and warnings.
* **LFLAGS**: Linker flags including SDL2 and SDL2\_image libraries.
* **SDLFLAGS**: Additional flags for SDL configuration.

**Targets:**

* **all**: Default target that compiles the executable.
  + Compiles source files using the specified compiler, flags, and libraries.
* **clean**: Target to remove temporary files and the executable.
  + Uses the **RM** command to delete files that match the pattern **\*~** (temporary files) and the executable (**$(NAME)**).

**Execution:**

* **Execution Command:**

To compile the project we use **make -f Makefile**

#### **Dependencies:**

* SDL2 Library: For window creation, rendering, and input handling.
* SDL2\_image Library: For loading image textures.

**Notes:**

* The project follows a modular structure, separating functionality into different files.
* The raycasting algorithm is responsible for creating the 3D effect of walls.
* User input is checked for movement and quitting the program.
* Textures are loaded based on the specified map, enhancing the visual experience.

This documentation provides an overview of the project structure, file purposes, and compilation/execution instructions.