**Project 1 Writeup**

**Maze Runner Simulation**

**Coded by Eric Diaz**

**Class: 48102**

**Professor: Dr. Lehr**

**About Maze Runner Simulation**

The Maze Runner is a game where the player is to travel through narrow corridors and use teleporters to win the game. The controls to move the player are the same as countless computer games out in the market using the WASD keys. The different characters on the map represent the player [‘O’], walls [‘x’], and the paths the player can navigate through [‘.’]. The game starts off by reading from one of “map.dat” files and places the player according .

The player can choose to save the game at any time by pressing the letter ‘P.’ The save function works by writing the player position, the level, and the map layout to the file “player.dat”. When the player relaunches the game, they can choose to continue where they left off by pressing ‘P’ to read from “player.dat” and setting the variables prow and pcol to the location last stored in “player.dat”.

I had a few minor setbacks when writing the code. First was my error in using global variables. I chose to use global variables because I started the project not knowing about how to reference variables in functions. Multiple functions utilized and manipulated the global variables to achieve the desired result. I then attempted to fix the global variable problem by referencing variables in the parameters which led me to have functions with many, many parameters. I then rewrote the code in the function Main and used minor functions for aesthetic purposes. Interestingly enough, the amount of lines in my code decreased by about 100 lines when I stripped my code of functions.

**Major variables and uses**

**Integers**

Level - The map that is loaded is dependent on the level

**Booleans**

Game - The “Game Loop” depends on this variable. It is true while the player is playing the game and is set to false when they want to quit.

Round - The “Round Loop” depends on this variable. It’s true when the player is navigating through the “map.dat” and false when they want to quit

Load - Called once to load the last saved game if player chooses to and then set false for the remainder of the game.

**Characters**

Pmov - Called every time the round loops. The input is the direction the player wants to move.

**Fstream**

Dim - Reads the “map.dat” files to get and set the dimensions of the playing field.

gameSave - Writes to “player.dat” to save the position and level the player last saved on.

**Strings**

dimS - Dimension String. Used to output the map.

crsStr - Crash String. String used to check if player moved on a ‘T’ or ‘A’.