DOOR MAZE BUILDER AND GAME

Written by Byron Burks using the Python programming language

INTRODUCTION

The development of this game was begun in November of 2011, and has continued to be improved and expanded since then. It is basically a set of two Graphical User Interfaces (GUIs), the first of which allows the user to create a maze, while the second allows someone to navigate through any maze created with the first GUI. The maze itself is simply a building with several rooms that are linked with doors, some of which are locked. Whoever plays the game must start from the first room, find any levers needed to unlock/open closed doors, and make their way to the final room to win the game. The newest version of the game allows you to build a maze with up to 98 rooms, by creating a building with 7 rows, 7 columns, and 2 floors of square shaped rooms. You can specify how many rows, columns and floors you want when you first create a new maze.

INSTALLATION

SOURCE CODE (FROM GITHUB.COM)

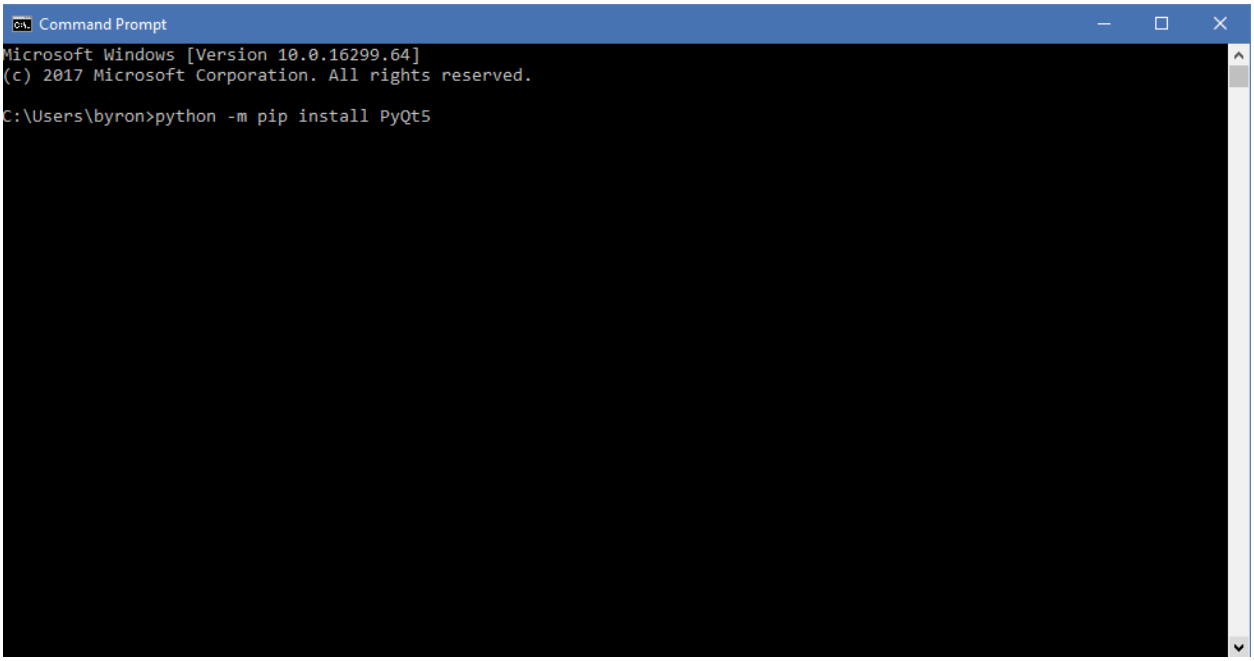
PYTHON VERSION AND DEPENDENCIES

Version 4.0 of the Door Maze Game was built using Python 3.5 and PyQt5. Other Python versions 3.4 or greater should work, if you also have PyQt5 and its dependencies installed.

INSTALLING DEPENDENCIES ON WINDOWS

To obtain Python, go to www.python.org, and locate the downloads page. The recommended version is Python 3.5 32-bit or 64-bit (64-bit versions of Python may utilize more than 4 GB of RAM on a 64-bit machine; otherwise there are few differences between them). After you select and download the desired Python version, install it by double-clicking on the file. Next, launch a Windows command prompt (search for "cmd.exe" on the taskbar or start menu). When you see something like the screenshot below, type the following command and press enter:

```
python -m pip install PyQt5
```



This command should automatically find, download and install PyQt5 and its dependencies.

Installing Dependencies on Linux

For Linux users, open a Bash terminal and execute

```
>>python3 -m venv <DIR>
```

```
>>source <DIR> /bin/active
```

Where <DIR> is the directory in which to create a virtual environment with Python.

Next, install PyQt5 by executing

```
>>pip install PyQt5
```

This should complete the installation of all the dependencies for the Door Maze Game.

Running the Door Maze Game

To obtain the source code from GitHub, go to the repository called python-games by byronburks92 on www.github.com and then click the green button (Clone or Download). Select "Download ZIP" to download the entire project from GitHub. Finally, extract the downloaded file, and locate the folder "door maze game v4_0". Copy that folder to a convenient place on your hard drive. Before attempting to run the game, copy the contents of the folder "door maze images" and paste it until the parent folder, so that they are alongside the *.py and *.ui files. The scripts you would need to run are called "build_maze4.py" for the Maze Builder GUI and "door_maze_game4.py" for the Door Maze Game GUI. Follow the steps below to launch the scripts based on which operating system you are running.

For Windows, you may either write a simple Batch script, or open a command prompt and run the code from there. I recommend writing a Batch script by opening Notepad (not Wordpad or other editors with more complex formatting) and typing these lines:

```
@echo off
python name_of_script.py
pause
```

Save the file as "name_of_script_run.bat" in the same folder as the python script you wish to run and be sure that the extension "TXT" is not selected below the filename. Finally, close Notepad, and double-click the BAT file to run the script.

For Linux, you will need to open a terminal and navigate to the directory that contains the scripts using

```
>>cd <DIR>
```

where <DIR> is the full path name of the script. To launch the script, first verify that the correct virtual environment is active by using

```
>>source <VDIR> /bin/active
```

Where <VDIR> is the "venv" directory you should have set up when installing Python. Finally, use

```
>>python name_of_script.py
```

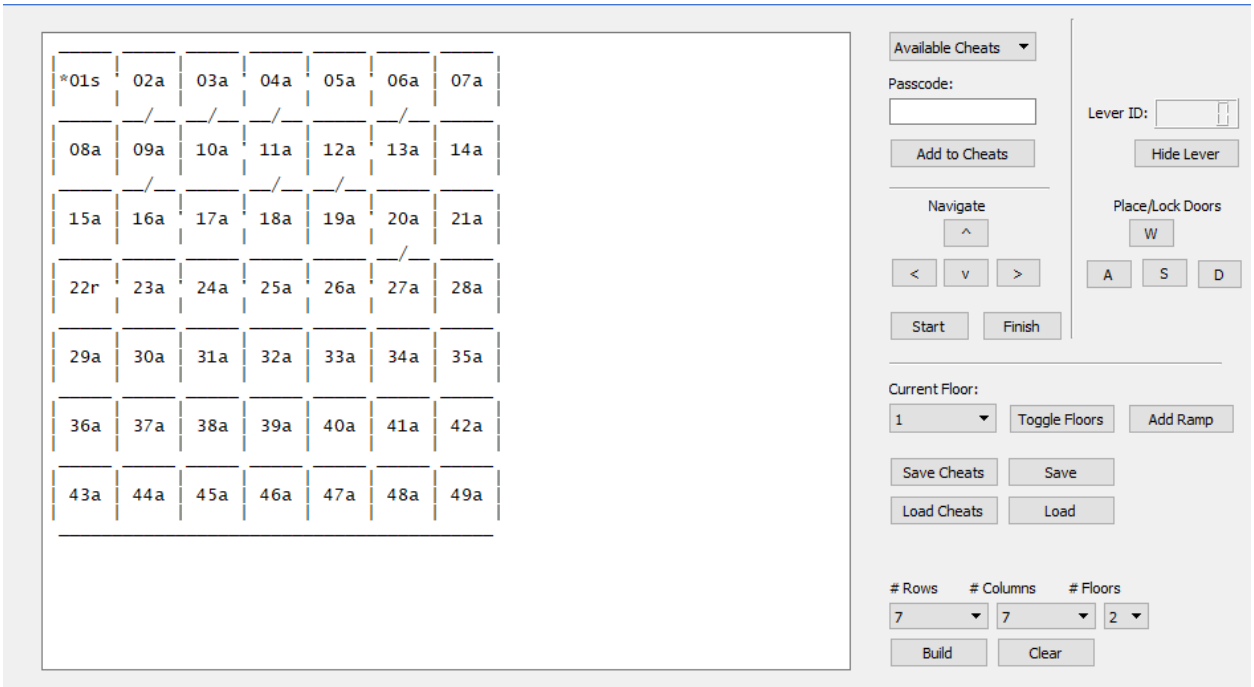
to run the script.

How to Play

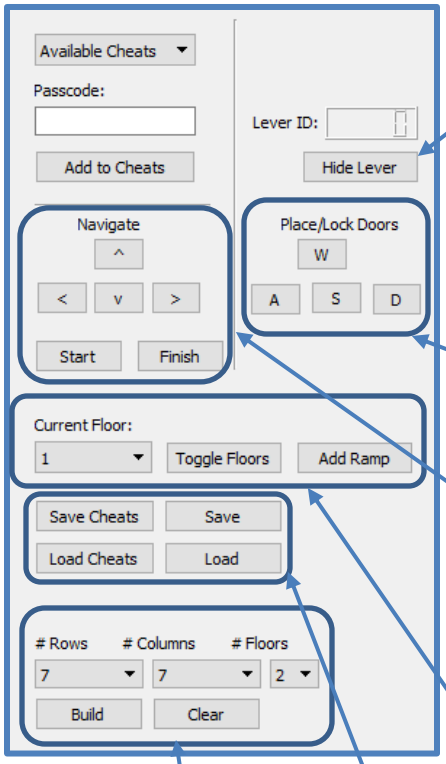
Building a Maze

Layout

The maze building program may seem very complicated at first, but once you learn how to use it, you will find that it provides a quick and simple way to create a maze. Check out the layout in the following screenshot.



This is the entire GUI with the display section filling up most of the window. Notice how the maze is displayed as a grid of rooms. Each of the openings in the walls represents a door. The numbers are only for reference within the builder; they do not appear when you are playing the game. The letter next to each number specifies the contents of that room. The letter “a” means the room is empty, “k” means a key/lever is hidden in the room, “r” tells you that the room contains a ramp, “s” means the room is the starting point, while “f” marks the finish. If an asterisk appears next to the room code, then that room is currently selected for editing purposes. Below is a screenshot of only the side bar which is on the right of the GUI.



Hide Lever: places a lever within the active room to be used to close a door. The ID just allows you to differentiate between the levers. Shortcut: press the

Place/Lock Doors: each button corresponds to the location where the door will be placed ("w" = top, "d" = right, "a" = left, and "s" = bottom). If there is already a door in the specified location, then the door is closed with the active lever if there is one (when "Lever ID" is not equal to "0"). Shortcuts are simply the keys that match the letters on the buttons.

Navigate, Start, Finish: use the navigation buttons to move from one room to the next. The next room in the direction specified becomes the active (current) room, which will capitalize the letter within that room in the display. Shortcuts: the arrow keys with the corresponding directions. To set the active room as

Current Floor, Toggle Floors, Add Ramp: use the Current Floor drop-down menu (combo box) to change the active floor (if the maze has more than 1). The change will be reflected in the display, allowing you to navigate and build on the other floor. You can also use the Toggle Floors button to change floors, or press the "f" key. Use the Add Ramp button to place a ramp in the active room. The ramp can be used in

Save, Load: To load a previously saved maze, press the "Load" button, and to save the maze you created/modified press the "Save" button.

Building a new maze: Use the three combo boxes to set the desired number of rows, columns, and floors. Once that is done, click "Build" to create a new maze with those dimensions. The "Clear" button will erase the current maze so that you can build a new one. Remember to save

WALKTHROUGH

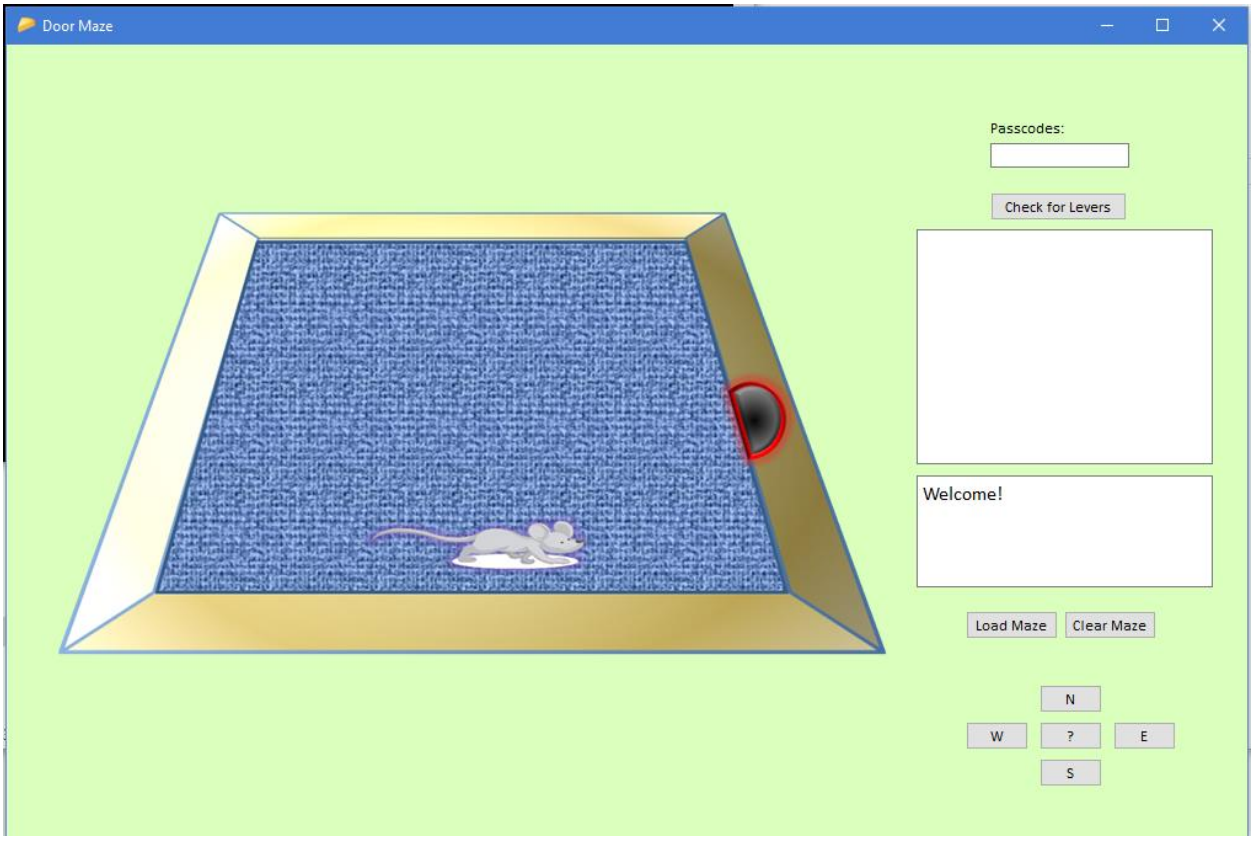
The following steps serve as a guide to building your own maze. Refer to the layout descriptions above to find all the relevant controls. IMPORTANT: there is no "undo" button; if you are afraid you will lose all your work, you may save it periodically so that you have a backup.

1. Choose the dimensions; up to 7 rows, 7 columns, and 2 floors are allowed in this version of the game (98 rooms total). Tips: choose no less than 3 rows or columns for either dimension of the maze, or else it will be difficult to design. Add a second floor if you want an extra challenge.
2. Clear any maze that may already be open, and click "Build" to build the new one.
3. Use the navigation controls to activate the room you want to be the starting point and click "Set as Start".
4. Navigate to the room you want to be the finish and click "Set as Finish".
5. Place doors and ramps to connect the rooms to create a maze. Tips: depending on how challenging you want the maze to be, try to create a route through the maze that requires the player to visit a lot of rooms before reaching the finish, complete with dead-ends and multiple ramps.
6. Hide keys/levers in strategic places, using them to close certain doors (optional). Tips: make sure all the levers are accessible to whoever plays it. Examples of doors you might want to close are the door to the finishing room, a door to a room that contains the next key that you need (be careful with that), and more.
7. Look over the maze, making sure that it is possible to complete it in the game. Make any additional changes if needed.
8. Save the maze with a name that does not have spaces or special characters (except for the underscore, "_") and that does not start with a number. Examples: "maze101.mz" and "door_maze01.mz" are allowed, but NOT "101maze.mz" or "maze#14.mz". Note that the extension ".mz" is added automatically to whatever filename you choose. This file type may only be opened within this GUI and the Door Maze Game GUI (it is an unintelligible binary pickle file).
9. You have now built your maze. Keep reading to find out how to play it using the second GUI.

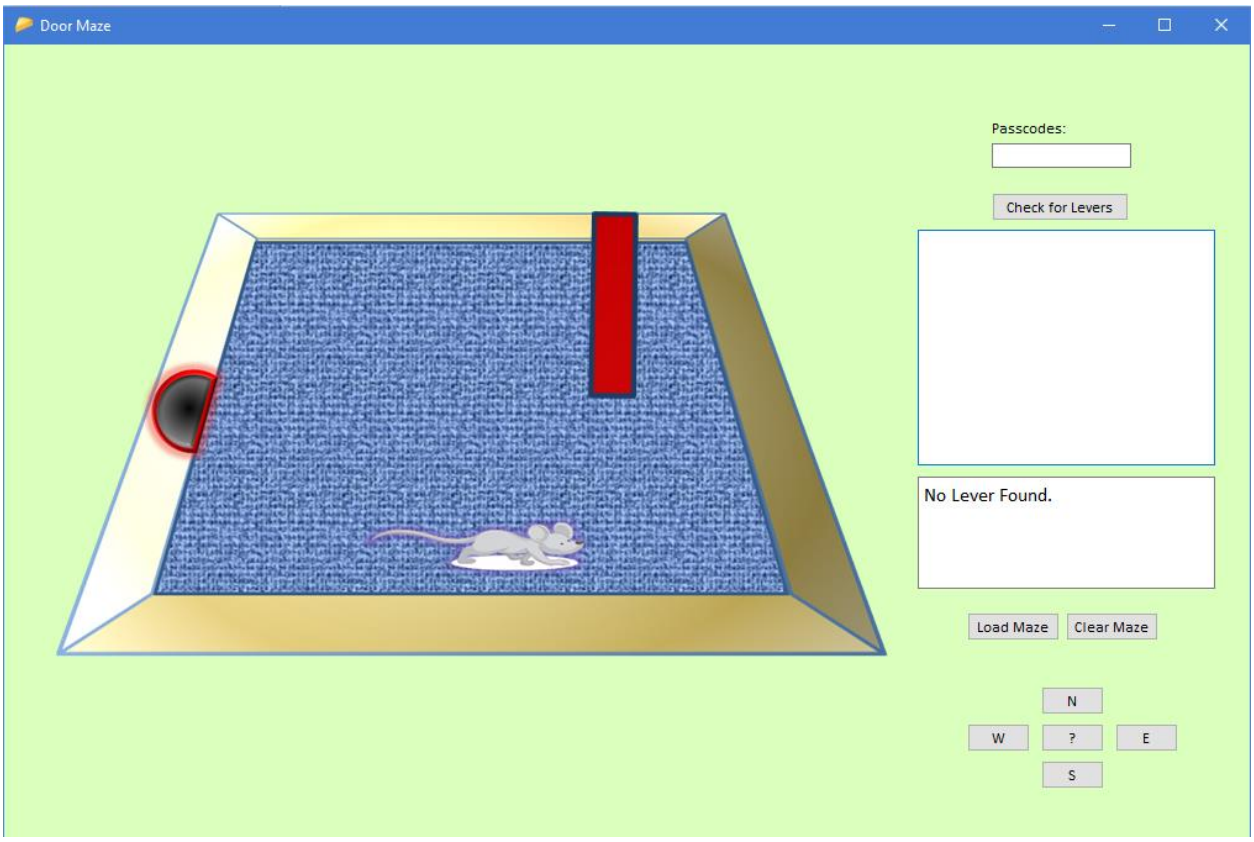
Playing the Game

Layout

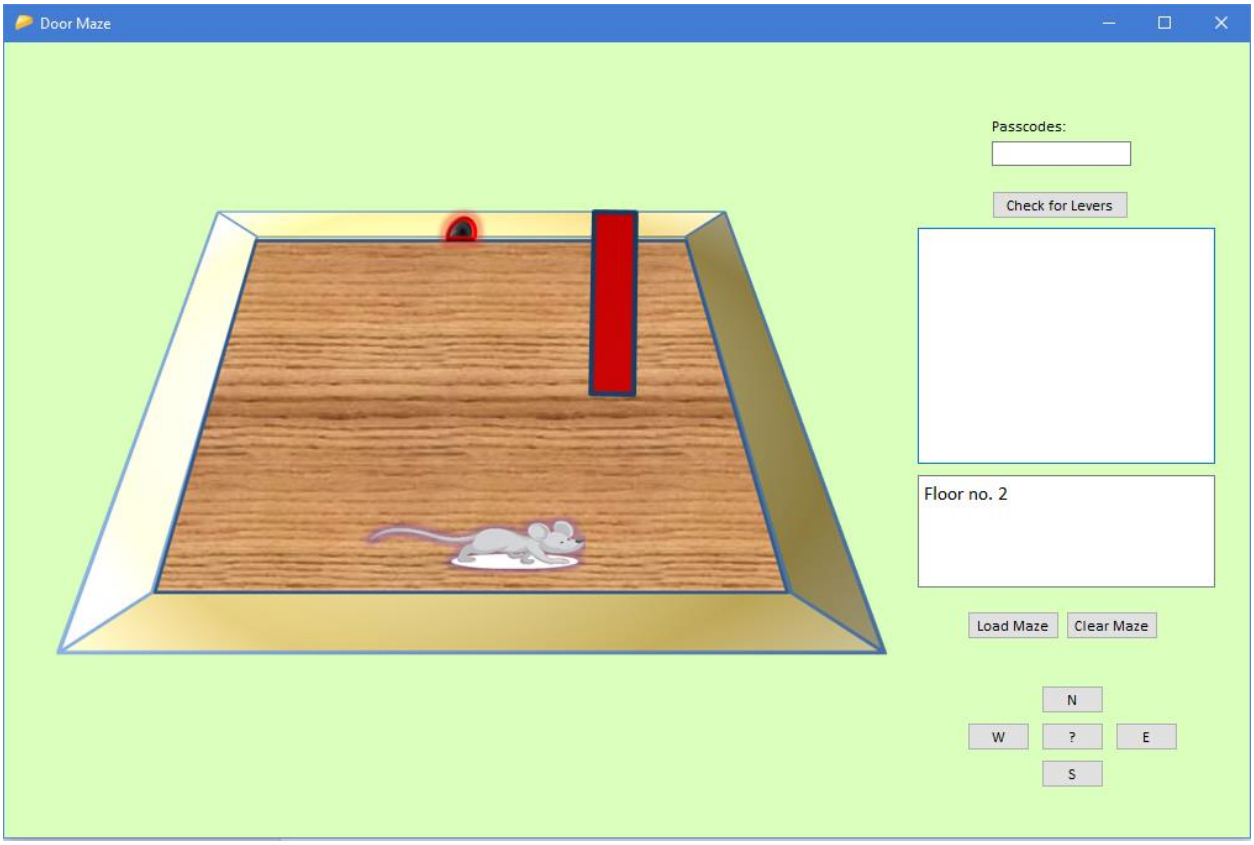
Refer to the screenshots below for the layout of the door maze game.

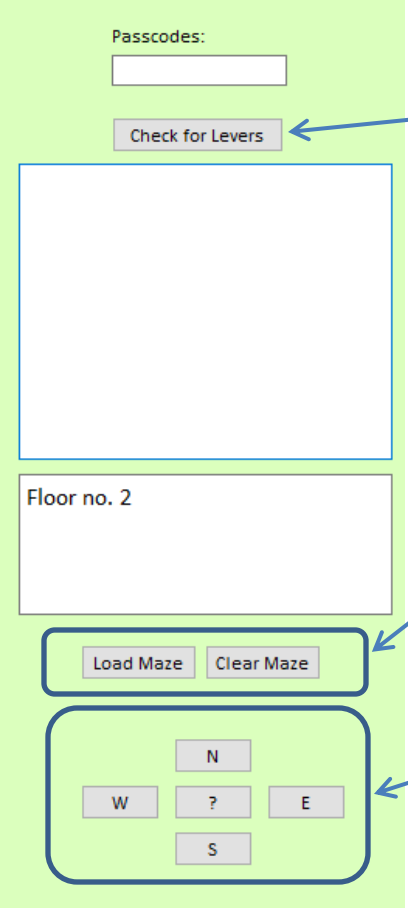


Just like the builder, the game has a display area that will allow the user to view each room as they enter it, although this GUI now displays actual images rather than ASCII art. There is also a couple message boxes that will keep the user informed as they try to solve the maze.



This room contains a ramp to the second floor. As seen in the next screenshot, the second story has a wood floor rather than carpet to differentiate.


Door Maze
Passcodes:
Check for Levers
Floor no. 2
Load Maze
Clear Maze
N
W
?
E
S



Search for Keys: this button will search for any levers that may be hidden in the current room. This button may be used an unlimited number of times for different rooms. Shortcut: the "k" key.

Load Maze, Clear Maze: Use these buttons to load a previously built maze, or clear the current maze from the GUI.

Navigation controls: the "N", "W", "E", and "S" buttons will allow you to enter the door on the side indicated by the compass-like directions. The "?" button uses the ramp if there is one in the room. Shortcuts: arrow keys for going through doors, and the "L" key for using a ramp.

WALKTHROUGH

The following step serve as a guide to playing the door maze game. Refer to the layout descriptions above to find all the relevant controls.

1. Load the maze of your choice by clicking the "Load Maze" button.
2. Once the first room is displayed, use the arrow keys to use doors where available (doors are marked by arrows that appear in the middle of each wall). Each time you use a door, you are immediately taken into the next room. If there is an elevator in a room (see screenshot above), press the "L" key if you want to use it. The following points have no specific order.
    a. Watch for the following messages in the message box:
        i. "Welcome!" appears when you first load the maze, and disappears whenever the next message comes up.
        ii. "Blocked!" appears when you try to go through a door that is not there.
        iii. "Door locked!" appears when you attempt to go through a locked door without a key.
        iv. "No Levers Found." appears when you search for levers when there is none.

v. “Lever Found!” appears when you find a lever in a room.
vi. “Door Unlocked!” appears when you are successful at unlocking a door.
vii. “Floor No. 1” and “Floor No. 2” appear when you change floors using a ramp.
viii. “Thanks For Playing!” appears when you reach the final room.

b. Search for levers by pressing the “K” key. Any time a lever is found, the corresponding door unlocks/opens immediately.
c. Watch for closed doors (They are the same shape, but with a wood-textured cover over the hole).
d. Try to keep track of where you are as you navigate the maze. If you are stuck, try mapping out the maze on a piece of paper as you navigate, taking note of each turn you make.

3. When you reach the final room, you will have won the game, and the “Thanks For Playing!” message will be displayed.

Comments and Updates

Let me know if you have any questions or feedback, and I will be happy to do what I can.

This last section of the documentation will be used for frequently asked questions, additional comments, bug reports, and information about any future updates.

Enjoy the game!