**Christopher Rockett**

**CS 162 Final Project**

My final project is a maze game called “Motorcycle Maze Craze.” Within it, the turtle GUI pops up for a user to play the maze. My code meets all 12 of the requirements for the project.

**Code Organization:**

I split my code into different classes and functions while maintaining the PEP 8 code style guidelines to keep my code clean. The code is very readable, allowing others to navigate and find spots easily in the code.

**Conditions:**

Within my code I use a lot of if statements and loops to create things like the menu, and movement of the turtle within the GUI.

A screen shot of a computer program

Description automatically generated

**Collection Types:**

Using lists, dictionaries, and tuples I set my program to keep track of data needed for it to run. The data to store all the different things in the maps and having turtle draw it out is what most of these collection types are used for.

**Object-Oriented Design Principles:**

My code makes use of multiple object-oriented design principles. I used the idea of inheritance to create a parent class of Elements for the subclasses Player and Map to use so I didn’t have to rewrite code a lot. I also used encapsulation in most of classes so that most of the data that operates with each other are in one unit. Lastly, I used polymorphism by setting a variable that is used in multiple functions throughout the code.

**Testing:**

My code tests the movement functions of the player and the custom exception I created. However, the only error when running my pytest is in collecting the information. It’s having a problem with the for loop menu that requires an input. I could not figure out a way to get past this.

A screen shot of a computer

Description automatically generated

**User I/O:**

The code makes use of user input by asking them what difficulty the want and having them press the arrow keys in which it responds by moving the turtle within the code.

**File I/O:**

The code takes .txt files for the maps to be read and created in the GUI to save space in the code. It also outputs all the coordinates the player has been to on a separate .txt file for if the user wishes to see their move list.

A screenshot of a computer screen

Description automatically generated

**Input Validation and Verification:**

I used if statements and try and except statements to prevent any unwanted input getting into and breaking the code while simultaneously letting the user know.

**Recursion:**

Within the solve\_maze() function I use recursion in an algorithm to make sure the maze is solvable before the user plays it. It uses a for loop to keep checking coordinates and possible places to go in the maze until it either gets stuck or finds the ending place.

**Error Handling:**

I used try and except blocks for the menu of the game. Therefore, for any popular errors are inputted it lets the user know for them to try again. I also created my own exception class seen at the top of the code. This gives an error when the user puts an input that’s longer than 2 characters.

**GUI Components:**

For the Maze Game I used both tkinter and turtle GUI to make the game visually. For most of it I used turtle to have a tracer and to create the maps, looking at the snake game we previously made and copying some of the code. I used tkinter for the pop window, letting the user know they have beat the maze.

**Documentation:**

I used PEP8 style, pydocstyle, and pycodestyle to help format my code. All my functions and classes have the proper docstrings to explain each of them while using comments to label any chunks of code.

**Visual Gameplay of Code**

**Start and Completion of Medium Level**

**A screenshot of a computer game

Description automatically generatedA screenshot of a computer game

Description automatically generated**

**Start and Completion of Easy Level**

**A screenshot of a computer game

Description automatically generatedA screenshot of a video game

Description automatically generated**

**Start and Completion of Hard Level:**

**A screenshot of a computer game

Description automatically generatedA screenshot of a computer game

Description automatically generated**

**Code Letting User Know There’s a Solution**

**A black screen with white text

Description automatically generated**