Assignment 4 – Game: Turtle in the Maze!

Title: Assignment 4 – Game: Turtle in the Maze!

This game and Assignments 4 and 5 were created by our TA Sitong Zhai! Thank you, Sitong!

Step 1 - Problem Statement:

Over the next two assignments, you are asked to complete a game in which the player (i.e., the user) moves a turtle through a maze, one cell at a time, with the objective of having the turtle finding its way out of the maze.

In **Assignment 4**, you will be setting up the maze.

In **Assignment 5**, you will be developing the "game engine", i.e., the part of the code that moves the turtle around the maze toward the exit gate. More details will be provided about this part of the game in our Assignment 5.

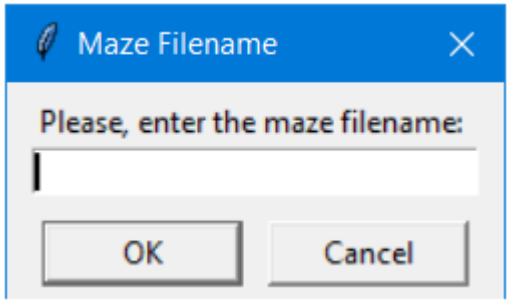
Requirements:

1. You must create your own Python 3 program using IDLE. You cannot use AI tools to create your program nor can you use someone else's program. For more details regarding Academic Honesty (or lack of) and what is permissible and what is not, please, read **A word about Academic Honesty** under **Lecture 1** on our course web site.

What to do:

1. Download the program `Assignment_4.py` and the four (4) text files posted under this assignment on our course website:
    a. `maze_1.txt`
    b. `maze_2.txt`
    c. `maze_3.txt`
    d. `maze_4.txt`

2. Execute `Assignment_4.py`, even though it is not complete yet! This program draws a white canvas on the computer monitor screen then it displays:



3. Enter the name of one of the text files you downloaded. In this example, I entered the filename `maze_1.txt`. Once I click OK, the program displays:

i.e., a yellow turtle. We are now ready to start Assignment 4.

4. Read the entire content of `Assignment_4.py` (comments and code) starting from the `***Main part of the program`, and reading each function (again, comments and code) as you encounter their call in the `***Main part of the program` or from other functions (for example, `drawMaze(…)` calls `drawSquare(…)`).

   This will take a while so plan for it. This must be done in order for you to understand what the code does and the data (variables) it utilizes.

5. Once you have a good understanding of the code and the data it uses, starts completing the first function `readDataFile(…)`, following the instructions given in `Assignment_4.py`.

   Test as you go!

   How?

   One way to test as you go is by printing the content of the variables your code is modifying and verify that the correct modification has occurred.

   Remember: Incremental development and testing!

6. The functions you need to complete are:
   - `readDataFile(…)`
   - `computeMazeWidthAndHeight(…)`
   - `findSymbolPosition(…)`
   - `drawMaze(…)`
   - `drawTurtle(…)`

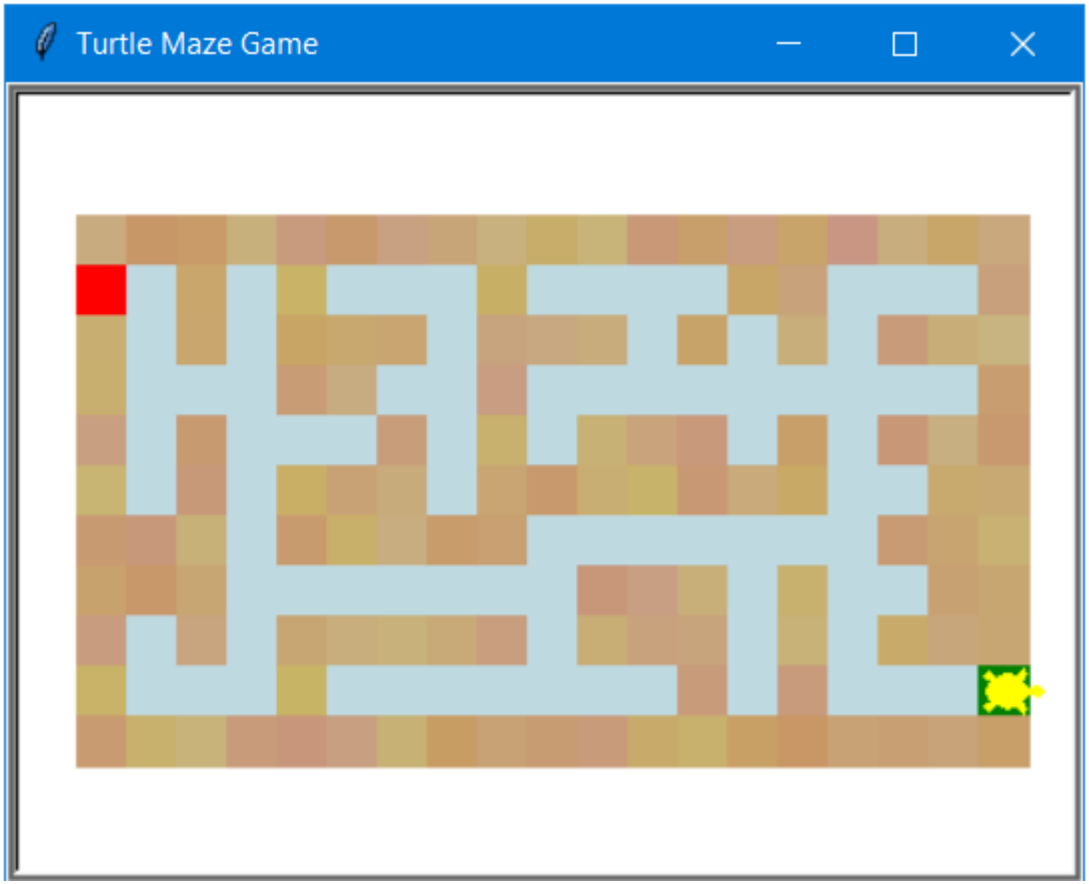
   Complete each of them following the instructions given in `Assignment_4.py`.

7. If the docstring of a function states: ***Do not modify the content of this function!***, then don't. Otherwise, your game may not execute as expected! ☹

Testing:

1. Test your final version using each of the data files (text files) you downloaded.

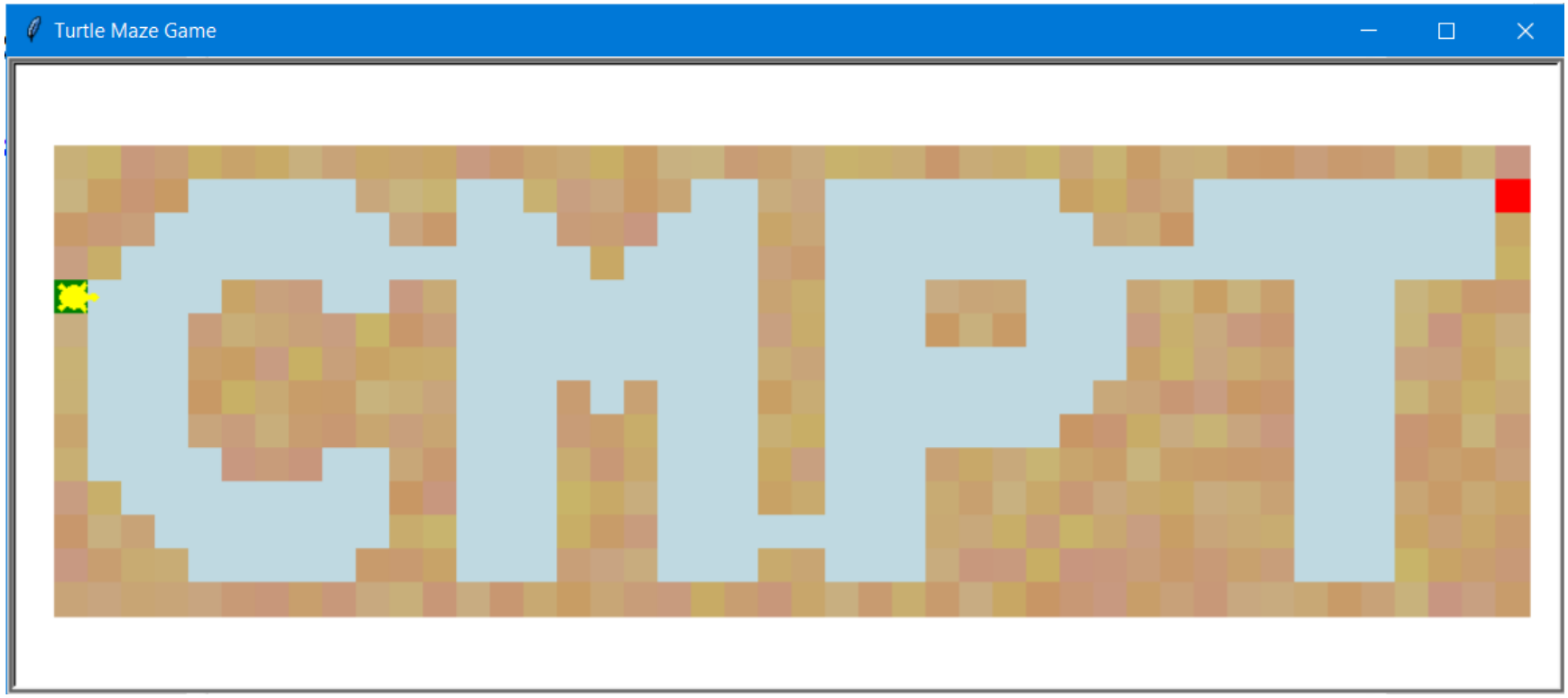
- If you enter the file **maze_1.txt** into the dialogue box, and your code has successfully been implemented, you will see the following maze on the computer monitor screen:



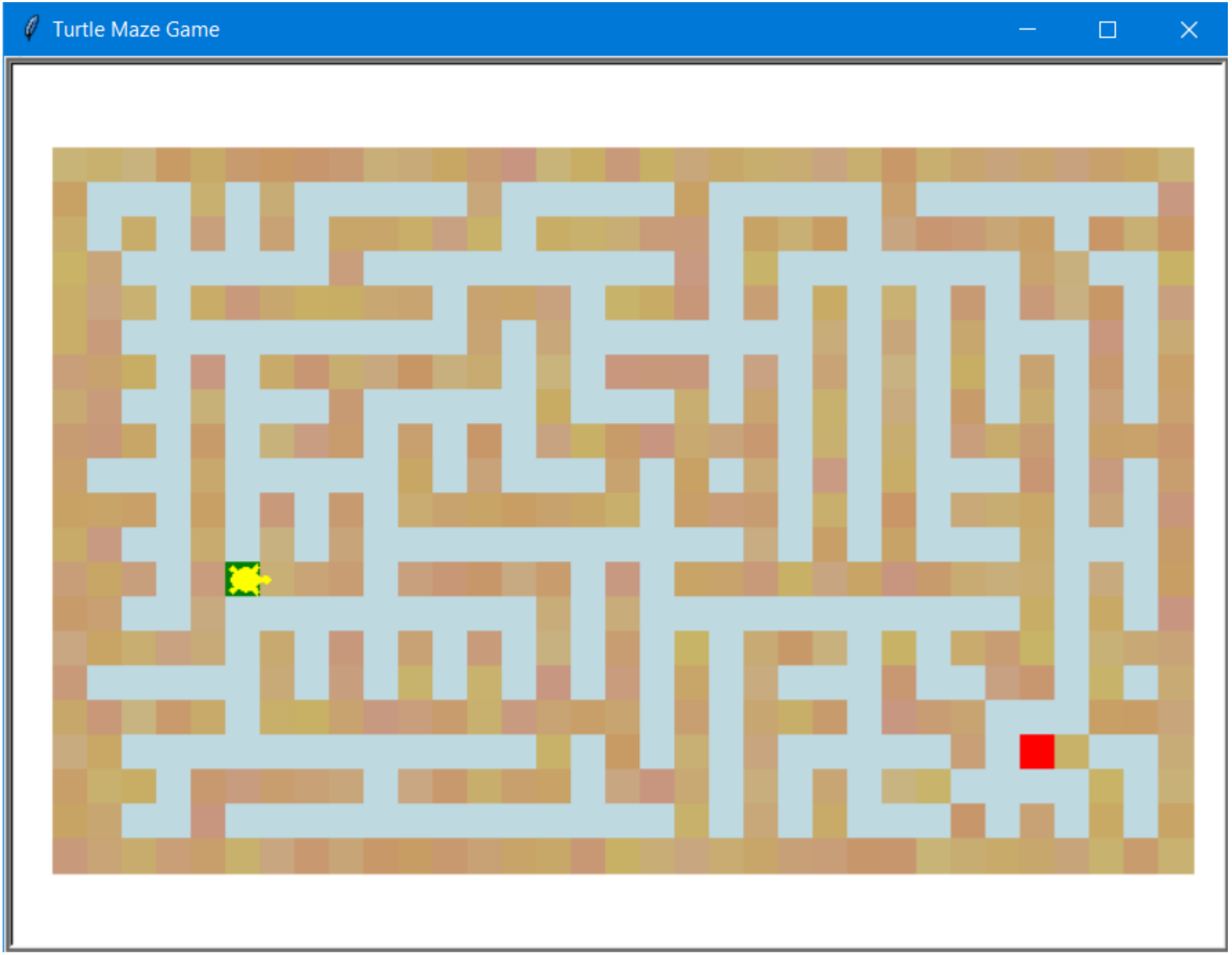
- If you enter the file **maze_2.txt** into the dialogue box, and your code has successfully been implemented, you will see the following maze on the computer monitor screen:

- If you enter the file **maze_3.txt** into the dialogue box, and your code has successfully been implemented, you will see the following maze on the computer monitor screen: (image below was resized to fit page)

.



- If you enter the file **maze_4.txt** into the dialogue box, and your code has successfully been implemented, you will see the following maze on the computer monitor screen


Turtle Maze Game

Submission:

- Submit your program `Assignment_4.py` on CourSys (https://coursys.sfu.ca/2024sp-cmpt-120-d3/). Click on the course activity called **Assignment 4**, then click on the option **Make Submission** on the left and finally, follow the instruction to browse for your program file.
- Note that you can submit your program as often as you wish. As long as your submissions are done before or on the due date and time, your assignment will be marked. CourSys will not stop you from submitting your program late, i.e., after the due date and time, but if your program is late, it will receive 0 marks.

How your Assignment 4 will be marked:

- When the TA marks Assignment 4, they will be looking at
    - whether your program solves the problem, i.e., your program displays the each of the above four mazes when you enter the associated data file name (10 marks),
    - whether your program satisfies **all the implementation details** given in the `Assignment_4.py` you downloaded and any **requirements** stated in this assignment (10 marks).
- The rubric for Assignment 4 is based on the above. Make sure your program satisfies the above before submitting your Assignment 4.

Enjoy!

Finally, there are no extension granted unless for medical reason once the **Official Medical Certificate** **has** been completed and submitted to the instructor.