CSCI 4342.001 Programming Assignment 05 (100 Points)

Suspense: December 02, 2021 NLT 12:30PM

OBJECTIVES:

- Demonstrate basic competency in Python programming.
- Implement Python functions.
- Write a basic interpreter.

ASSIGNMENT ASSISTANCE:

- This homework assignment is due at the beginning of the class period on the due date.
- This assignment is restricted to individual effort. You may not receive help from any other person.
- Any resource used (other than Dr. Nix or the course text) must be documented in the code (as comments) detailing the source and describing exactly what was learned and how that information was used. Submissions will be severely penalized if copied in part or in whole from any source.
- If you need help, visit your instructor during his posted office hours. If your schedule cannot accommodate any of these times, then email your instructor to schedule a different time.

PROBLEM DESCRIPTION:

1. Your task is to submit a program called `BreakOut.py` which reads in the Intcode program called `program.txt` (provided as a separate text file) and executes it.

2. When correctly interpreted by your solution, the Intcode program executes the game of breakout.

   a. If `IntCodeComputerV3.py` (Program 04) worked correctly, then your Intcode computer is complete except for the needed changed in the functions `getInput()` and `setOutput()`.

   b. In this assignment, you are to modify those two functions in order to provide the game with a proper graphical user interface using the functions available in `graphics.py` (also provided separately).

   c. You may choose to implement the game in one of two ways:
      i. Make the game interactively playable by the user.
      ii. Make the game so that it plays itself using a very simple AI.

3. Your function, `setOutput()`, should graphically display the game board which is updated using the results produced by the `04` opcode (the output instruction): every three output instructions specify the x position (distance from the left), y position (distance from the top), and tile id. The tile id is interpreted as follows:

    a. 0 is an empty tile. No game object appears in this tile.

    b. 1 is a wall tile. Walls are indestructible barriers.

    c. 2 is a block tile. Blocks can be broken by the ball.

    d. 3 is a horizontal paddle tile. The paddle is indestructible.

    e. 4 is a ball tile. The ball moves diagonally and bounces off objects.

4. For example, a sequence of output values like `1,2,3,6,5,4` would draw a horizontal paddle tile (1 tile from the left and 2 tiles from the top) and a ball tile (6 tiles from the left and 5 tiles from the top).

5. The software assumes that input is provided through a joystick. Your function, `getInput()`, should simulate the joystick by providing input via the following input instructions:

    a. If the joystick is in the neutral position, provide 0.

    b. If the joystick is tilted to the left, provide -1.

    c. If the joystick is tilted to the right, provide 1.

6. The provided input is generated depending on which implementation your solution provides:

    a. If the solution is a playable game, then the input is provided by either the keyboard or the mouse.
        i. If the keyboard provides the input, then provide instructions for which keys are used.
        ii. If the mouse provides the input, then use the functions within `graphics.py` for mouse interaction.

    b. If the solution is played by the AI, then the input is provided by the AI instead of the keyboard.

7. The game is also capable of outputting a single number that represents the player's current score.

    a. When three output instructions specify `X = -1`, `Y = 0`, the third output instruction is not a tile; the value instead specifies the new score to show in the segment display.

    b. For example, a sequence of output values like `-1,0,12345` would show `12345` as the player's current score.

8. The game should execute smoothly without any random lag, pauses to await user input, or extra key presses (e.g., the ‘`Enter`’ key).

9. You should not need to import any external library other than `graphics.py` for use in any solution (e.g., the use should not need to install `Pygame`). Use whatever functions within `graphics.py` that you find useful for your solution.

10. Your program should work in the Windows environment (PuTTY/SSH does not support graphics).

11. You should submit your work on D2L; no need to submit on the CS Server.

SUBMISSION:

- Review the Evaluation below to ensure you have met all the requirements.
- Submit electronic copy of `BreakOut.py` to D2L.

EVALUATION

a)	Project is late or not submitted at all.	-100
b)	Project does not run without throwing an exception.	-40
c)	Method signatures do not match the specification.	-20 per method
d)	Calculations/output are incorrect.	-15 per method
e)	Code is not well organized or properly indented.	-10
f)	Code is inadequately commented for readability.	-10
g)	Code does not contain student’s name, course section, and date of submission.	-10