**CS 5110/6110 Program 1 (20 points)**

# Project

This assignment is intended to help you become familiar with python. This can be done in pairs (as described in the syllabus). If done in pairs, only one copy of the assignment is submitted.

Consult the material in the course module named Preliminaries. Download python, pycharm, and install pygame. Some in the class are computer science coaches, so feel free to go to the coaching center for help installing.

Using wormy.py as a starting point, make the changes listed below. Since the program is written for one apple and one worm, you will need to plan your data structures carefully. Test your results after each step.

1. Change the name of the game (which displays at the beginning) to be something new.
2. Change the colors that display at the beginning. For example:

Icon

Description automatically generated with low confidence

1. Use a grid with more cells.
2. Have multiple apples on the grid. New apples will be generated as they are eaten.
3. Allow the user(s) to control two worms using the arrow keys (for one worm) and four other keys (for the other worm). The keypad arrow keys will be used to move both worms in the same direction at the same time. The keypad arrow key names are K\_KP2, K\_KP4, K\_KP6, and K\_KP8.
4. Show the scores for each worm separately. You can use whatever scoring system you like.
5. The worm dies when it hits a wall, the other worm, or itself. The worm is left behind as a stone. If a stone is hit by a worm, the worm dies.
6. Implement some variation of this idea: Allow the worms to shoot each other. If the worm is hit, it cuts him in pieces (based on where the ray hits him). The tail is left behind as a stone. If a stone is hit by either worm, the worm dies.
7. For 6110 students: Add a third worm which is not controlled by the user but moves (somewhat) intelligently.
8. Bonus (2 points): Implement an additional feature of your choice.

Create a short video demo of your project. Submit the video with your project. In the video, explain the original feature you implemented. I used obs studio <https://www.wikihow.com/Create-a-Screencast>. The instructions were perfect!

Create a readMe.txt file which identifies any bonus features and telling the user what keys to use for motion.

Submit the entire project as one zip file.

# Hints:

1. Importing pygame
2. Open File > Settings > Project from the PyCharm menu.
3. Select your current project.
4. Click the Python Interpreter tab within your project tab.
5. Click the small + symbol to add a new library to the project.
6. Now type in the library to be installed, for example pygame, and click Install Package .
7. For testing, slow the movement down by changing the frames per second (FPS) to 5 or less.
8. If you aren’t careful in your design, this will be a mess. Create separate parameterized modules to do various things.
9. Use lots of lists. They are easy to pull apart by the inclusion of for loops.
10. Don’t hesitate to print to the console while using pygame. That was helpful when I was trying to understand the code.
11. Parameter passing is a little tough to get used to (as it is neither by reference or by value). I often returned the values I needed (so there was no concern with how parameters were passed).
12. For example, I made building the worms a function which returned the list of worms and the directions they were to go initially.**def** getWormCoords(ct):  
     directions = [RIGHT,LEFT]  
     multiplier = [-1,1]  
     wormCoords = []  
     **for** i **in** range(ct):  
     #build worms  
     **return** wormCoords,directions
13. You can step through two parallel arrays in sync by first zipping the lists together. See the following example in which there are several worms that are colored differently.

**def** drawWorms(wormCoords,colors):  
 **for** aworm,color **in** zip(wormCoords,colors):  
 **for** coord **in** aworm:  
 x = coord[**'x'**] \* CELLSIZE  
 y = coord[**'y'**] \* CELLSIZE  
 wormSegmentRect = pygame.Rect(x, y, CELLSIZE, CELLSIZE)  
 pygame.draw.rect(DISPLAYSURF, color[0], wormSegmentRect)  
 wormInnerSegmentRect = pygame.Rect(x + 4, y + 4, CELLSIZE - 8, CELLSIZE - 8)  
 pygame.draw.rect(DISPLAYSURF, color[1], wormInnerSegmentRect)

For example, if there are two worms, wormCoords may look like:

[[{'x': 20, 'y': 26}, {'x': 20, 'y': 25}, {'x': 20, 'y': 24}], [{'x': 27, 'y': 33}, {'x': 27, 'y': 32}, {'x': 27, 'y': 31}]]

Colors represent a pair of colors for each worm and may look like: [[(0, 255, 0), (0, 155, 0)], [(30, 144, 244), (176, 224, 230)]]