Did you like your program topic? Why (not)?

I, along with the rest of my team, definitely enjoyed this project as I was able to try something new that I've never worked with before. While I had used Python before this class and more or less had at least some familiarity with the topics we discussed in this class, I had never used or even heard of Turtle graphics. My (limited) experience with Python graphics had always incorporated the Tkinter library, so it was interesting using something else.

Did your team stick with your initial bottom-up approach to program development, or did you switch to a different design approach at some point? Which approach did you prefer and why?

I think for the most part our team was able to stick with the initial bottom-up approach, however we did deviate away from it a little bit when refactoring our code to better handle invalid input and the turtle graphics. Our team chose to take advantage of the bottom-up approach and started writing a parent game class with defined methods to handle individual aspects of the game play. For example, the game class had methods such as switch_turn, place_piece and check_win. For longer projects, I enjoy the bottom up approach as compared to the top-down approach because I can fine tune individual aspects of the program in the beginning.

Did your final design closely match the original hierarchy your team developed? In what ways is it the same? In what ways is it different?

I think our final design did closely match the original hierarchy as we had a pretty good idea of the aspects a connect four game would entail. We did however deviate from the original hierarchy when we had to incorporate some error handling for example to prevent our game from crashing when the user inputs an invalid input.

Briefly describe the thing(s) you learned on your own, beyond what was covered in the lectures. How did you incorporate them into your program?

One of the things we heavily incorporated into our game that went beyond what was covered in lectures was the use of classes. We decided to create a parent game class to handle individual aspects of the gameplay. The game class was responsible for handling the actual gameplay mechanics such as filling in circles on the connect 4 board for each user's turn. Conversely, the main function was able to handle other aspects of the program such as printing the rules.

If you had more time to work on your program, what additional features would you want to add? Is there anything about your submitted program you would change?

One thing that I would definitely like to improve for the program if I had more time would be the graphics. For this program we were required to use turtle graphics, but there are separate Python libraries that make really nice graphics such as PyQT that I'd like to incorporate.

What was the most difficult part of this assignment? Please explain.

I think the most difficult part of this assignment wasn't really the coding aspect of the assignment but rather finishing this assignment with everything else going on. Because it's nearing the end of the semester, a lot of classes are beginning to wrap things up with exams, including this class. This assignment was definitely one of the longer assignments of the semester and it was

a little challenging trying to devote time to this assignment while worrying about the midterm for this class and the Wumpus project

Estimate the portion of the assignment completed by each member of your team

Surya: 40% - Surya spent a lot of time creating the logic for the actual game including most of the logic in the game class
Sumit: 30% - Sumit worked with Surya to create the GUI for the project with Turtle graphics
Srikar: 15% - Srikar handled some edge cases such as invalid input and helped clean up some of the code and graphics
Ronit - 15% - Ronit also handled some of the edge cases and helped refactor some of the code