Team Game Project: Design Phase

You have now produced individual program specifications for the final project, been placed into your final groups, and begun discussion. We have given you feedback on the initial proposal. Now is the time to decide what your group will do, and how you will do it. For this assignment, you will finalize your game selection, justify the choice, and develop the header file(s) to implement the program.

Think about the game you have selected, and consider how the game rules map to code. Take into account the feedback provided to you by your TA and your instructor. Consider what is reasonable for 3–4 people to develop in 3 weeks. What graphics and GUI classes will you need to draw the board and pieces? What classes will you need to develop to control for valid moves? How will all of the parts fit together to make a playable game? What will be the format of your input / output files? How will you divide up the work?

Once you have done that, think about your program in a top-down sort of way. Break it down into smaller and smaller pieces of code on paper. Once you have done this, convert those pieces into a set of classes with data members and functions.

As a refresher, the following is the project assignment specification:

“Games are systems of rules, in which players make choices [Salen and Zimmerman 2004]. Game rules, based on game state, restrict the choices a player may make. For example, if we are playing Tic-Tac-Toe and I have selected the middle space as my first move, you are now restricted to the other eight spaces. In this assignment, you will **select a game and implement its rules, with a graphical user interface and the ability to save game state**.

“You will craft a **game played on a grid**. This team project will test your ability to build simple graphical user interfaces, handle erroneous input from the user (prevent them from making illegal moves), and your ability to do file I/O and handle errors.

What to Turn In (the Deliverable)

Produce and turn in the following (PDF, text, RTF, or Word doc):

1.) The title of the game your group has selected, and a revised description of the game rules in your own words. Include some description of the program components that will be needed for part 2. As before, be certain to cite the source from which you acquired the rules. (1 page minimum/maximum)

2.) A breakdown of your group members’ assignments. Who will develop which component (from 1)? **All group members must be involved with the program design, implementation, demo, and final report.** (1 page minimum/maximum)

3.) Describe the following:

   a. How does the program start up? What will you present the user? What will be the controls? Don’t forget to account for loading an existing game state.

   b. Once the game has started (either a new game or an existing, loaded game), what options are available to the player? How are those options created in code? (For example, if the game is checkers, the user can move one piece diagonally or jump pieces only in a certain direction. You might implement this by having buttons on all of the available pieces and available spaces so the user clicks an existing piece, then clicks a new location, then the program checks for validity and executes the move if it is valid.)

4.) One or more header files that declare the classes that you will use to develop your program. These should be complete and syntactically correct; you should be able to compile them. Build them from your descriptions above, then fill in the definitions to build your program. Make sure to comment your code as needed to make it readable!

Grading

The intent of this assignment is to get you thinking about how to select a problem, and design a medium-sized program to “solve” it, using object-oriented principles. You will be graded accordingly. **This assignment will count as 25% of your team project grade (effectively, 5% of your final average for the class).**

1.)	
Accurate description of how the game is played	/ 4
Citation of source of game design	/ 1
2.)	
Balanced assignments for team members	/ 5
3.)	
Clear description of how the program will work	/ 7
4.)	
Declaration connects to description in 3.	/ 3
Declaration is human-readable	/ 2
Declaration follows object-oriented principles of inheritance	/ 3
total	**/ 25**

References

Salen, K., and Zimmerman, E. *Rules of Play: Game Design Fundamentals*. MIT Press, Cambridge, MA, USA, 2004.