

Team Game Project: Proposal Phase

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. While a more detailed specification of what the project will entail, and your team assignment, will follow, this first deliverable, undertaken individually, will form the basis of your project.

Why an Individual Assignment

The reason this component is assigned prior to your team assignment lies in the trouble with brainstorming. Specifically, the problem is that, in groups, frequently the first idea proposed becomes the one undertaken, or, at least, it colors the thinking on all ideas that follow. This is known in the creative cognition literature as “fixation”. [Smith 1994]

Subsequent to you turning in this assignment, you will be [most likely randomly] assigned to a project team. Your project team will then consider all of the team members’ proposals and select one to implement, merging any useful ideas and specifications from the other project proposals.

Selecting a Game: the Rules (and Considerations) of the Assignment

Select a game that **you** would like to implement for the class project, given the following requirements and notes about final project requirements:

- 1.) For simplicity’s sake, your game must be played on a grid; for complexity’s sake, the game must be played on a grid of 6x6 squares or greater (no Tic-Tac-Toe!).
- 2.) Again for simplicity and complexity, the game must already exist. This class is not specifically about game design, and some of you are not interested in game design; however, this project will provide an interesting “problem” for you to solve with code.
- 3.) You must select a game that you reasonably believe you could complete (with 2-3 teammates) by the end of the semester. I understand that “reasonably believe you could complete” may be a vague concept to you at this time. Undertaking this exercise will help you understand the complexity you are planning to undertake.
- 4.) The game should be turn-based and, when ultimately implemented, should be controlled by the mouse (as it will have a graphical user interface).
- 5.) As part of the assignment, you will have to include the ability to stop playing the game at any time and save the state of the game. Your game will also need to be able to read in a saved game state, allowing players to pick up where they left off.

Note that you will not be graded on the complexity of your game’s mechanics, but it must be a reasonably difficult software engineering problem for you to solve.



What to Turn In (the Deliverable)

Once you have selected a game, produce and turn in the following (PDF, text, RTF, or Word doc):

1.) **The game rules in your own words.** Be certain to cite the source from which you acquired the rules. (1 page minimum/maximum)

2.) A prototype declaration of classes, their description, member variables, member functions, that you expect to need to solve the “problem” of implementing your game. A declaration communicates an interface to the programmer. When you write this one, make sure I know what you mean! By description, I mean a (very) brief comment for each class and its members describing what they do in plain English. If a class’s member variables and functions carry very descriptive names, this is acceptable as well.

At a minimum, I expect a declaration of a game object that includes the ability to draw itself to the display. Where possible, you may build on any existing libraries (just be sure to note what header file is needed to link in the library). Remember that you will need to have a graphical user interface to the game and that you will have to somehow place boundaries on what moves are legal at any point in play. This documentation will serve as a first pass on the specification for your project. (1-3 pages, maximum)

A Final Note

By mentally walking through (2.), you should begin to develop a sense of how much work this project will be to implement, it should help you to make a decision about what game to implement.

A Few Suggestions (feel free to find others)

Connect Four, Chess, Checkers, Othello, Go, Sudoku, Crosswords. See also:

List of games played on an $N \times N$ grid: <http://boardgamegeek.com/wiki/page/Gameboards>

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 a 50-point individual homework assignment.**

1.)

Game meets requirements specified above	/ 10
Accurate description of how the game is played	/ 10
Citation of source of game design	/ 2

2.)

Declaration is human-readable	/ 12
Game class includes functions for I/O	/ 8
Game class includes functions for drawing to the screen	/ 8
total	/ 50

References

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

Smith, S. M. Getting into and out of mental ruts: A theory of fixation, incubation, and insight. In *The Nature of Insight*, R. J. Sternberg and J. Davidson, Eds. MIT Press, Cambridge, MA, USA, 1994, pp. 229–252.

