CSCI 201L - Final Project
15.0% of course grade

Introduction

Throughout this course you have learned topics such as user interface design, GUI components, multi-threaded programming, and networking. Up until this point in the semester, you have worked alone on homework assignments that have had strict guidelines. This project is going to expose you to working in a group on a large project of your own creation from conception to deployment. You will need to utilize all of the topics you have learned throughout the course in this project.

Description

You will be working in groups of 4-6 people from any CSCI 201 section to create an application of your choosing. You must abide by the specifications below, though anything not specified can be implemented however you want. You will create a multi-player game that executes across computers using networking. The game can be any type of game that you want within the following:

- The game must be a multi-player video game
- There has to be a way for one player or one team to win
    - You can have multiple players on a team, but there must be at least two teams composed of human players
- Your game must incorporate graphics to make it more aesthetic
- Each player must have his own character, whatever that character may be
- Players must be able to chat with an individual player, all players on their own team, a selected group of players, or all players
    - This needs to happen asynchronously so it does not interfere with the rest of the game
- The display for each user must be different. In other words, a board game like Monopoly where everyone sees the entire board is not acceptable.
- The game must be written in Java, though if you would like to adapt it to an Android device, that would be acceptable.
- Statistics must be stored in a database located somewhere.

If you struggle to come up with an idea or want to know if an idea is an appropriate endeavor, feel free to discuss with your professor or one of the CPs.

Group Work

Part of the purpose of this project is to work together in a team. There will most likely be some issues within the team, and I hope that you all will work out the issues. Each group will be assigned a CP to work with throughout the project. If you are unable to resolve issues within the group, please let your CP know. If the problem persists, let your professor know so that a discussion can be mediated among the team members. In extreme cases, groups can be dissolved or people removed from groups. At that point, the sub-groups will be responsible for the entire application. During the group presentation each member will showcase his/her contribution to the project. If you let your group do all the work, you will have nothing to show, and this will affect your individual grade.

Specific Requirements

You will need to abide by good object-oriented guidelines with your program. Part of your grade will be based on the efficiency of the design and implementation of your code as portrayed in your documentation as well as your code.

With multiple programmers on a team, you will need to set up a version control system for managing your program. There are multiple version control systems available to use, including CVS, SVN, git, etc.

You will need to submit documents for each major phase in the software engineering lifecycle – high-level requirements, technical specifications, detailed design document, testing documentation, implementation (code), and deployment document. Your final program will be demonstrated during a specified time, and you will have to **prove** that your program is working correctly. We will not run specific test cases, though we may ask you to do something during the demo time.

Tips

Since there are a lot of parts to the project, you may be tempted to assign specific parts to specific people. This is fine, but it is important that everyone in the group reviews each part before submission. You should probably have an internal deadline of at least one day earlier than the actual submission deadline so the rest of the group is able to review each document before submitting. Of course, with the implementation, this should be done much earlier so the entire group is able to test and work out bugs.

Much of the work might be able to be accomplished separately, especially with the technologies we have at our disposal for remote work. I am still going to require at least two hours a week to meet as a group and work out any issues and discuss upcoming tasks.

There is a chance that you may come up with something that is publishable. If you think that you have come up with something that is novel, please talk with me and we can discuss trying to write a paper to submit to a conference or journal. This is not a requirement of the project, but it is definitely something that will help your resume for the future.

Submission Instructions

In Eclipse, your project should be named "CSCI201_FinalProject_username". For example, if your username is “jmiller”, you would name your project "CSCI201_FinalProject_jmiller". This will help us for grading in case we need to run your program after submission.

You should always have a comment at the top of your code with all of your team members’ names in it. Do this at least once in the .java file where the main method exists.

Submit the implementation of your final project the same way you submit your assignments. Submission instructions are posted on the Assignments page of the course web site. Only one person from the project needs to submit each document, but make sure all team members’ names are on every document so we can assign points accordingly.

Grading Criteria

% of Final Grade	**Due Date**	**Criteria**
1.0%	2014-10-12	Submit team members, weekly meeting time, and project proposal
1.0%	2014-10-19	High-Level Requirements document
1.0%	2014-10-26	Technical Specifications document
2.0%	2014-11-09	Detailed design document
1.0%	2014-11-16	Testing document and testing code
1.0%	2014-11-23	Deployment document
1.0%	2014-11-30	Code completion (though testing and bug fixes can still be implemented)
5.0%	2014-12-01 – 2014-12-04	Demonstration/Presentation
2.0%	2014-12-07	Complete documentation

Description of Grading Criteria

All documents should be submitted as PDFs.

Submit team members, weekly meeting time, and project proposal

You will need to select your group of 4-6. You also need to find a two hour block where the entire team is able to meet each week. Determine a location to meet. You must also come up with the idea for your project. Submit the idea along with a brief overview of the project which highlights how you will use a GUI, multi-threaded code, and networking. This proposal should be less than one page in length and should only provide a conceptualization of the project. The specifics of the implementation and other parts of the project will come in future documents.

High-Level Requirements document

All teams will need to write up all of the details about their project. You must write out these specifications as if you were a client requesting for a program to be written that performed the tasks provided. The document must be an overall description of the behavior of your program.

Technical Specifications document

Based on the feedback you receive on your requirements document, you will then need to write up technical specifications. You will submit both your requirements document and your detailed technical specifications document since you may have had to modify your requirements based on provided feedback. Make sure you clearly delineate anything that you added, deleted, or changed in your specifications. This can be accomplished using Track Changes in Microsoft Word (or a similar feature in other word processors).

Detailed Design document

Based on the feedback you receive on your technical specifications document, you will write up your detailed design document. This should show any difficult algorithms, the exact GUI (though slight modifications will still be allowed), the database schema (ER diagram), the hardware/software requirements, class diagrams, inheritance hierarchies, etc. This will likely be a lengthy document.

In addition, submit your requirements and specifications documents again with any changes clearly delineated using a feature such as Track Changes.

Testing document and testing code

You will need to prove to us that your program is working the way it is supposed to. That means that you will need to include sufficient test cases to prove that your program is working. This

includes all types of testing, including white box testing, black box testing, unit testing, regression testing, etc. Not only should you create a document in this step, but you should also create the test code. You will be able to add other test cases and code throughout the implementation phase, but this will hopefully provide you with a good set of test cases to ensure proper operation of your application.

In addition to the testing document, submit your requirements, specifications, and detailed design document with any changes clearly delineated using a feature such as Track Changes.

Code completion (though testing and bug fixes can still be implemented)
You will submit your code wrapped up in a zip file that can be imported into Eclipse and executed. By this date, the new development of your program should be completed, and only bug fixes and testing should be performed. You shouldn't be adding new features into your program after this date.

In addition to the code, submit your requirements, specifications, detailed design document, and testing document/code with any changes clearly delineated using a feature such as Track Changes. You probably will have added more test cases as development proceeds, so mark those.

When you submit your implementation, include a README.txt file in your project in the doc/ folder of your project that explains what works and what doesn't, along with any other nuances with your project about which the graders should be aware.

Deployment document

We won't actually be deploying your program in a live environment, but if we were to, this document should be what is followed. It needs to include step by step of what to do to take the code from Eclipse and deploy it. This will most likely consist of some other configurations also (i.e. outside libraries, server settings, Java download/install, etc.).

In addition to the deployment document, submit your requirements, specifications, detailed design document, testing document/code, and updated code with any changes clearly delineated using a feature such as Track Changes. You probably will have added more test cases and fixed some bugs as development proceeds, so mark those.

Demonstration/Presentation

You will demo your program and prove to us the correctness of it. We may ask you to show something specific, but you will be driving the demonstration. Also during this time, each group member will have a chance to showcase what he/she contributed to the project. No formal slides or presentation is required, but you will have to walk us through the program and show us all of the features. I assume this will take between 5-10 minutes.

Nothing is needed to be submitted for this part.

Complete documentation

You will put all of your documents together into one document, include a cover page, table of contents, and page numbers, and submit the entire set of documentation. Again, if anything has changed from the last submission, mark the changes clearly with a feature such as Track Changes. This document should be nice enough that it could be printed and presented to the client of this application as the final document.