SIENA*college*
Computer Science

Computer Science 225
Advanced Programming
Siena College
Spring 2021

Final Project

Matching groups to ideas through 2:40 PM, Friday, April 23, 2021
Groups must be formed and repositories created by: Monday, April 26, 2021
Proposals due: 2:40 PM, Friday, April 30, 2021
Demos: Monday, May 17, and Friday, May 21
Final submission: 2:40 PM, Friday, May 21, 2021

For the remainder of the semester, much of your time for the course will be spent completing a final project, worth 10% of your course grade. This project will be graded out of 300 points.

You will choose one or more significant programs of interest to you and your teammates to design and implement using Java. You have a great deal of freedom in choosing what to program for this project. However, taken together, your programs must be an effective demonstration of your Java programming skills. More details can be found below. As part of the purpose is to develop your abilities to work collaboratively, you are expected to work in groups. Groups of size 2 may be approved, but most should involve 3-5 team members. Groups must be formed and all repositories created (regardless of group size) by Monday, April 26, 2021.

Basic Requirements

You may find it difficult to estimate the programming effort that will be required for programs you are considering. Please discuss ideas with with me before going too far.

Your project should showcase many of the programming skills you have worked on this semester. In particular, all projects must include all of the following, in at least one of the programs you choose to implement.

- A meaningful object-oriented design, making effective use of interfaces, abstract classes, inheritance, etc.
- Event-driven programming including Java graphics and Java Swing GUI components.
- Threads for animation or computational speedup.
- Appropriate use of data structures.
- At least one Java or general programming feature or construct that we have not specifically studied in this or prerequisite classes.

Similar assignments in the past have tended to gravitate toward games, but this is by no means a requirement. People have implemented (often simplified) version games like Tetris, Minesweeper, Space Invaders, Pac Man, Angry Birds, Frogger, Breakout, and various 2D scrolling games. If you are going the games route, do not limit yourself to the above list by any means. Have a look at your favorite list of old Atari games or check for games in your favorite app store for lots of ideas.

Applications outside of the realm of games could include simulations, larger-scale computations that take advantage of threads, data processing, image processing, among others.

Basically, find something that you're interested in doing, and have some fun with it.

Matching Groups to Project Ideas

Through 2:40 PM, Friday, April 23, 2021, we will work toward matching people into groups by what they are interested in doing, and to make sure we end up with a nice variety of projects, by contributing to a shared document. Please start thinking about some possible applications you would like to implement right away and put those ideas in the document. This document can also help those looking to form groups to find others with similar interests.

Repository Creation

Rather than creating repositories through the GitHub Classroom mechanism we usually use, each group will create a repository for their project. It should be housed in one group member's GitHub account. Other group members should be granted write access and your instructor should be granted read access (if the repository is private, you may also just make it public). Send the repository's URL to your instructor as soon as it's ready to go, no later than Monday, April 26, 2021.

The Proposal

By 2:40 PM, Friday, April 30, 2021, submit a proposal (by committing and pushing a file `proposal.pdf` or by creating a GitHub Markdown file `proposal.md` in your repository), at most one page in length, that briefly describes the program(s) you wish to write, which of the requirements each program is expected to satisfy, and how you plan to go about it. Describe the major milestones for your project, a rough schedule for achieving these milestones, and which milestones you believe are most important for your project to be considered a success. Your proposal should make it clear that you have an interesting and worthwhile set of programs to write, and that it is feasible to complete them in the time available.

You should also propose a breakdown of the 250 points among your programs. If you are proposing more than one program (which is typical of most projects), suggest the number of points that should be earned for successful comletion of each program. Further break these down into points for completing major pieces of functionality. You should **not** include points for things like documentation, style, efficiency, and good use of version control. These are expected of all projects and would be subject to separate penalties if unsatisfactory.

Your proposed timeline should give intended dates for the completion of specific design tasks and functionality implementation.

Working Collaboratively

As we continue to work in a mix of in-person and remote modes, a significant challenge will be to find ways to collaborate effectively. Of course, we will make good use of GitHub for this, but groups are encouraged to establish communication mechanisms that work for their own situations. This could include web conferencing, using GitHub Issues, a Slack discussion, shared documents, lots of email, etc.

Most likely, you'll end up with a mix of some synchronous team programming, in-person where possible or with screen sharing, and breaking out tasks for subgroups to work on, then integrate them back together.

Your collaboration will be measured as it has in earlier problem sets by the Git commits from each team member's account.

The Project Code

You should submit your fully-documented source code by committing and pushing to your repository by 2:40 PM, Friday, May 21, 2021.

Post-Project Report

Also by 2:40 PM, Friday, May 21, 2021, submit a project report (by committing and pushing a file `report.pdf` or by creating a GitHub Markdown file `report.md` in your repository), at most one page in length. This report should include instructions on how to run each program. You should also state how each basic requirement (the list from the "Basic Requirements" section of this document) was met, how course style guidelines were met, and how project-specific requirements from your proposal were met. Also summarize the group's use of git and GitHub for collaboration and version control.

The Demos

You will be expected to show off your projects to your classmates, and, if you wish, to the wider Siena community.

- During class on Monday, May 17, and Friday, May 21, each group will make a brief demonstration (10 minutes or so) of your project to your classmates, using screen sharing in our Zoom meeting. This is the chance to ask each other questions also. We will pick time slots later.

- Each group will also create a video presentation and post it somewhere that your classmates can see it. These can be pretty informal, and can be created with a captured Zoom session or any technology you're comfortable with. Links to these videos are due at 2:40 PM, Friday, May 21, 2021, so they can be distributed to everyone. Please include a link in the `README.md` of your repository.

Academic Honesty Guidelines

Collaboration within a group is unrestricted. Since each group is working on different programs, you are free to discuss your projects with each other. If you wish to use or refer to any software libraries or outside source code beyond the standard Java API, check first, and cite their usage appropriately. All sources must be cited properly. If in doubt about anything related to Academic Honesty, ask now and avoid problems later!

Final Thoughts

You have several weeks, so the expectation is for several weeks of work. You will not be able to do a good job if you put it off. The expectation is not for you to produce a Ph.D. thesis, but this project should be much more than your average problem set assignment.

Grading

This assignment will be graded out of 300 points.

Feature	Value	Score
Proposal	25	
Program Functionality	250	
Video Demo	10	
In-class Demo	5	
Post-project Report	10	
Git commit/messages penalties (up to -25)	0	
Workload fairness penalties (individual)	0	
Style/documentation penalties (up to -50)	0	
Total	300	