SIENAcollege
Computer Science

Computer Science 225
Advanced Programming
Siena College
Spring 2022

Final Project

Matching groups to ideas through 11:59 PM, Friday, April 1, 2022
Groups must be formed and repositories created by: Monday, April 4, 2022
Proposals due: 2:40 PM, Monday, April 11, 2022
Demos: 10:00 AM, Friday, May 6, 2022(Academic Showcase)
Final submission: 11:59 PM, Friday, May 6, 2022

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 4, 2022.

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 at least one example of each of the following, though not every program you choose to implement would include examples of each.

- 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. 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 and some of your classmates are interested in doing, and have some fun with it.

Matching Groups to Project Ideas

Through 11:59 PM, Friday, April 1, 2022, 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 the shared document linked from Canvas. 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.

An effective group will include those with different strengths to bring to the project. Some of you like to come up with ideas and can design at a high level. Others might be best at coming up with a class hierarchy and identifying objects and their functionality. You might really enjoy cranking out code for specific tasks. Group members will likely contribute in different ways, but everyone must be a substantial contributor as documented in the final report (see below).

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 4, 2022.

The Proposal

By 2:40 PM, Monday, April 11, 2022, 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 program or set of programs to write, and that it is feasible to complete the proposed work 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.

For example, if you were planning to implement two programs: a Dig Dug game and a simulator for the card game "War", a possible breakdown might look like this:

- Dig Dug game – 150 points
  - Basic game layout and control panel GUI – 25 points
  - Dig Dug motion/digging – 25 points
  - Enemy motion – 25 points
  - Dig Dug can blow up enemies/drop rocks on them – 25 points
  - Enemies kill Dig Dug if they touch him – 20 points
  - Configuration and game play for one level – 20 points
  - Scoring/life counter/game over – 10 points
- War Card Game simulation – 100 points
  - Representation of the card deck – 10 points
  - Generate random deals for the simulation – 5 points
  - Run a simulation of one game, including multiple "wars" and wars when one player does not have a full set of cards to add – 30 points
  - Simulation of redecking options (always on top, always on the bottom, choose by what's advantageous) – 20 points
  - Gathering and presentation of simulation stats – 25 points
  - Analysis of whether a player can meaningfully impact their chances of winning with a redecking strategy – 10 points

Your proposed timeline should give intended dates for the completion of specific substantial design tasks and functionality implementation. You should have design and/or implmentation goals specified at least weekly. You would likely refer to the functionality listed above and propose deadlines for their design, implementation, and testing. You could also include deadlines for learning about technologies that will support the goals, such as if you need to learn how to use network socket communication or interface with a database from Java.

Working Collaboratively

As everyone is very busy and schedules can be difficult to coordinate, a significant challenge will be to find ways to collaborate effectively. Working together in person is likely most effective, but might not always work. Of course, we will make good use of GitHub for collaboration, 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 in part by the Git commits from each team member's account. When some code is committed that was a direct collaboration, other group members who were part of that collaboration should be mentioned in the commit message.

The Project Code

You should submit your fully-documented source code by committing and pushing to your repository by 11:59 PM, Friday, May 6, 2022.

Post-Project Report

Also by 11:59 PM, Friday, May 6, 2022, submit a project report (by committing and pushing a file `report.pdf` or by creating a GitHub Markdown file `report.md` in your repository). This should be no more than a page or two in length. This report should include

1. Instructions on how to run each program.
2. A statement on 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.
3. A brief summary of the group's use of git and GitHub for collaboration and version control.
4. A paragraph written by each group member outlining their specific contributions toward the design, implementation, and testing of the project.

The Demos

During the Academic Showcase on the last day of the semester, we will have a session scheduled where you will show off your projects to your classmates, and to the wider Siena community. Details of this will be announced when we know the schedule for that day.

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	
Academic Showcase	10	
Post-project Report	15	
Git commit/messages penalties (up to -25)	0	
Workload fairness penalties (individual)	0	
Style/documentation penalties (up to -50)	0	
Total	300	