CS 3050 Spring 2026
Final-Project Description

Preliminary comments

Some themes appeared frequently throughout the individual reports. Based on the what many people mentioned, here is some advice:

- do not procrastinate—get started early
- communicate
- communicate effectively
- make and maintain a task list in shared place (such as a Google doc); or use Trello
- do not sit around waiting for someone else to tell you what to do—play an active role
- do not do other people's work: trust your teammates to do the work they have signed up for
- get a sense of other team members' workloads outside of CS 3050
- get a sense of other team members' procrastination style
- know that for a team project, more structure/organization is almost always better that less structure/organization

Project choices

Your team will choose one of these projects:
(1) an interactive game
(2) a client/server system with a GUI on the front end and a database or datastore on the back end

If your team has an idea for a different kind of project, discuss your idea with me.

General comments:

- I expect each team to pick a project that will challenge them. I will take "degree of difficulty" into consideration during project evaluation.
- The final project has tight schedules. **Do not procrastinate**.
- Set reasonable expectations—do not be overly ambitious.

Project Choice #1: Interactive GUI-Based Game

The interactive game will be a software implementation of a game that already exists in non-computer form. In the past, Software Engineering students have implemented Sorry!, Blokus, Othello, chess, and others. If you pick a different game, I must first approve your choice. An arcade-style game (Pacman, Wii

Tanks) is also fine. Catan is also possible, but it will be challenging.

Requirements:

- the implementation must enable at least one human player to play against at least one AI player
- the game should be relatively sophisticated from the standpoint of the interactions required from a user
- the implementation should be GUI-intensive

Observations:

- the specifications will be very clear: implement the game you choose, according to the accepted rules of the game (or with modifications of the rules that you discuss with me)
- the design itself is up to you, but the interactions should be intuitive (i.e., the GUI and the play itself should be intuitive)
- the GUI will be relatively complex; if you don't have very much experience doing GUI development, there can be a steep learning curve
- the AI player doesn't have to be smart per se, but an AI player must understand the rules of the game

You can use your choice of language, although in the past, students who have chosen C++ have found C++ graphics libraries to be difficult to work with in a project of this scope. Student teams usually choose Python Arcade (or, less commonly, PyGame), and Python. Python Arcade is a newish, sophisticated framework for building games, and the documentation and examples are ample.

Project Choice #2: Client/Server System

For the client/server system, you can choose the database or datastore and also the language and frameworks for the front end and the back end. You must host your back end on a remote server (not as localhost). You can use UVM's silk server or a third-party cloud environment (e.g., AWS, Google Firebase, FlyIO, Heroku). FastAPI, a Python-based web framework, is a package that some student teams have used in the past, on a service such as Heroko. You'll have to pay (a little) for AWS or for Heroku.

Possible projects might be: a recipe system, an online reservation system for a restaurant, a web-ordering system, a buy/sell marketplace, or something else your team proposes.

Observations:

- the specifications are wide open here--your team will have to develop them, in part through discussions with me
- front-end/back-end web-development frameworks might be new to some or all of your team members

Version control

Teams are required to use github. In the past, I have sometimes used the record of a student's commits to help me evaluate that student's work.

Use branches—do not push to main. Make a plan for how your team will use branches. Feature branches enable easier collaborative development than a "one-branch-per-person" scheme.

Deliverables

Deliverable 0: Project Proposal

This describes the project your team will create and says what language(s) and framework(s) you will use.

* Due Saturday, Feb. 21st, 11:59 pm

Deliverable 1-A: Sprint One Goals

This document will have a set of tasks for Sprint One of your project. Describe each task as easy/medium/hard and nice-to-have/important/essential.

* Due Monday, Feb. 23rd, 11:59 pm

Deliverable 1-B: Sprint One Results

You should have implemented the Sprint One tasks. The result of Sprint One should be a working (albeit preliminary) version of your product: we should be able to clone your repo and run your code to see the results of Sprint One. Include also a short document describing (briefly) for each task from your list of sprint tasks either that you implemented the task, and how we can see the results; or that you did not implement the task, and describe why you did not do so. Describe also how each team member contributed to these sprint results. Provide a README file describing what (if any) additional packages are needed, and how a user should run your program.

* Due Sunday, Mar. 8th, 11:59 pm

Deliverable 1-C: Individual Reflection

Each person: submit a short document describing two things that are going well so far and two areas that need improvement.

* Due Sunday, Mar. 8th, 11:59 pm

Deliverable 2-A: Sprint Two Goals

This document will have a set of tasks for Sprint Two of your project. Describe each task as easy/medium/hard and nice-to-have/important/essential.

* Due Wednesday, Mar. 18th, 11:59 pm

Deliverable 2-B: Sprint Two Results

You should have implemented the Sprint Two tasks. We should be able to clone your repo and run your code to see the results of Sprint Two (and Sprint One). Include also a short document describing (briefly) for each task from your list of sprint tasks either that you implemented the task, and how we can see the results; or that you didn't implement the task, and describe why not. Describe also how each team member contributed to these sprint results.

* Due Sunday, Mar. 29th, 11:59 pm

Deliverable 3-A: Sprint Three Goals

This document will have a set of tasks for Sprint Three of your project. Describe each task as easy/medium/hard and nice-to-have/important/essential.

* Due Wednesday, Apr. 1st, 11:59 pm

Deliverable 3-B Sprint Three Results

Submit a zipfile containing your code, and include any additional resources needed to run your program. Include a link to your repo. We should be able to run your code to see the results of Sprint Three (and the two prior sprints). Include also a short document describing (briefly) for each task from your list of sprint tasks either that you implemented the task, and how we can see the results; or that you didn't implement the task, and describe why not. Describe also how each team member contributed to these sprint results. Also, **include a readme file** describing your program and (important) describing how to run your program.

* Due Sunday, Apr. 12th, 11:59 pm

Pylint analysis

Use pylint (https://pylint.pycqa.org/en/latest/index.html). Your code should have no fatal errors (F messages), no errors (E messages), no warnings (W messages), and no convention messages (C messages). I will run pylint on your code and use the report during my evaluation. You can suppress C0114 and C0116 checking.

Evaluation

Deliverable 1-A and 1-B: 20%
* quality of the source code
* quality and robustness of the program
* correctness of the program: have the sprint goals been met?

Deliverable 2-A and 2-B: 20%
* quality of the source code
* quality and robustness of the program
* correctness of the program: have the sprint goals been met?

Deliverable 3-A and 3-B: 20%
* quality of the source code
* quality and robustness of the program
* correctness of the program: have the sprint goals been met?

Above-and-beyond factor: 10%
* I will work with the TAs (I'm the TA this semester) to decide on this additional 10% for the project. If your team does a so-so implementation of an easy project, then you'll get fewer points here. If you tackle a tricky project and do a good job, you get more points. If you produce an excellent implementation of a straightforward project, you'll get more points.

Individual contribution/participation: 10%
* based on my understanding of each person's contribution/participation and on team member's evaluations

Individual Report

Answer these questions:

1. Identify two areas of improvement in your development process (not in your

code)—things that did not work well—and for each, describe what specific actions a team you are on could do next time to make these improvements.

2. Identify two areas in your development process that did work well, and say what your team did to make these aspects of your development work well.

3. Describe two things that you and your team did differently during the final project based on your experience during the warm-up project.

4. For the two things you listed on your warmup individual report ("list two things that you personally should do differently on the next project"), describe how you did them during the work of the final project; or why you didn't do them during the final project.

Elaborate in your replies to these questions—don't provide just a single-sentence answer to each question.

Also, provide an assessment in the form of a 5-4-3-2-1 rating of the effectiveness and contributions of yourself and of your team members. 5 is the best; 1 is the weakest. These are ratings—not rankings—and they will be kept confidential (I'm the only one who will see them). For each team member, describe briefly why you giving that rating.

15% of project grade

Due Sunday, Apr. 12th, 11:59 pm

Group Presentation

Each team will give a presentation to the class that describes their product and their development process. This will be similar to the presentation for the warm-up project. I'll provide more detail as we get closer to the end of the semester.

Final-project presentations: Monday, May 4th, 4:30 - 7:15 pm, Lafayette L300

5% of project grade

Summary of Due Dates

All deliverables are due by 11:59 pm on the date listed.

Saturday	Feb. 21st	Deliverable Zero (Project Proposal)
Monday	Feb. 23rd	Deliverable 1-A (Sprint One Goals)
Sunday	Mar. 8th	Deliverable 1-B (Sprint One Results)
Sunday	Mar. 8th	Deliverable 1-C (Individual Reflection)

Wednesday	Mar. 18th	Deliverable 2-A (Sprint Two Goals)
Sunday	Mar. 29th	Deliverable 2-B (Sprint Two Results)
Wednesday	Apr. 1st	Deliverable 3-A (Sprint Three Goals)
Sunday	Apr. 12th	Deliverable 3-B (Sprint Three Results)
Sunday	Apr. 12th	Individual Report
Monday	May 4th	Group Presentation, 4:30 - 7:15 pm