

Homework 5 — Game Night!

Due Tuesday, November 19th @ noon; Friday, November 22nd @ 11:59pm;

Tuesday, December 10th @ 11:59pm

(5, 45, 150 points)

Late submissions will not be accepted for the due date on December 10th

Credit:

- `checkpoint0.[pdf|txt]` (checkpoint 0)
- `checkpoint1.[pdf|txt]`, `prototype.pdf` (including user testing results!),
`finalproject.zip` (including all code so far) (checkpoint 1)
- `finalproject.zip` (final deadline)

Instructions:

You may work with a partner for this assignment. You must store your code in a private repository if you are working with a partner. Everyone must complete interview grading after your final submission has been turned in.

We will be holding project presentations during the final week of class/finals week. Attendance for all project presentations is **required**.

Your job is to design and implement a board game that has a graphical user interface. This game can be a game of your own design, it can be a near-replica of a game that already exists, or it can be loosely based on a game that already exists.

Gameplay requirements (minimum):

1. Give a brief description of how your game is played and what (if any) game it is based off of.
2. Your game must have some sort of "playing field"—a board, where cards are displayed, etc, showing the current state of the game. Briefly describe what yours will look like.
3. Your game must have a consistent theme. What will it be?
4. You must have at least **3** different types of resource/game piece/card that are related to each other. What are they?
 - a. These should be represented by one struct/enum class/class that you will define
5. Players must be able to acquire the items from #4. How?
6. You must have at least **2** different buildings/structures/upgrades/power-ups that are related to one another. What are they?
 - a. These should be represented either by one base class and one derived class or by different instances of the same class, as appropriate for your implementation.
7. You must have at least **1** other mechanic (events that occur, special cards, a non-player unit, trading, etc). What is it?
8. Your game must be multiplayer. How many players will you allow?



9. The computer must be able to control any number of the players (including all players).
What will the basic computer strategy be?
10. Your game must have well-defined begin and end states. What are they?
11. (Do you have any other features that you are hoping to incorporate?)

Simulation requirements (minimum):

1. Your GUI must include a mechanism to automatically simulate any number of games between 1 and 10 (or more) being played between computer players.
2. After each game has been simulated, your GUI should display a graph showing how many games which player has won so far.
 - a. If accumulation of resources from game to game is more important than who won an individual game for your project, your graph should display the accumulation of resources over time.

Technical requirements:

1. You must build your game in c++, using Qt.
2. Your interface must be legible.
 - a. This *must* be a graphical UI.
 - b. You must produce a low-fidelity prototype and conduct user testing. The game should be easily playable.
3. You should plan on separating your objects into separate files as makes semantic sense.
4. You must use and implement **1** of the following design patterns: flyweight, iterator, factory, and prototype.
5. You must have an appropriate inheritance relationship between at least **2** objects. They should have at least one method that is virtual and is appropriately used.
 - a. inheriting from Qt objects does not fulfill this requirement

Checkpoint 0 (Tuesday, November 17th @ noon):

checkpoint0.pdf

1. Your name & your partner's name (if you have a partner).
2. If you have a partner, invite Felix to collaborate on your private repository.
3. Address the gameplay requirements. For each requirement, answer the question at the end of the item.
4. What is your design proposal for the underlying object models for your project? (include design pattern and inheritance relationship proposals here)
5. What do you planned to have completed for Checkpoint 1? This should be a detailed list. Make sure to address what a user should be able to see/do by this checkpoint.

Checkpoint 1 (Friday, November 22nd at 6pm):

prototype.pdf (can be turned in on paper before the end of day on the 22nd directly to Felix)

1. Low-fidelity prototypes & user feedback from at least 2 people not in your group.



- a. Feedback can be from anyone: friends, family, classmates
- b. Be specific when you write down the feedback—what elements/interactions should be changed/different, was there something about the design that the person interacting with your prototype appreciated?
- c. Use the materials from lecture 18 as guidelines for creating your prototype and conducting your user testing

checkpoint1.pdf

2. What you planned on doing for this homework deadline. (Copy + pasted from Checkpoint 0)
3. What you actually accomplished for this deadline.
 - a. Note any differences and explain why they occurred. Prefer honesty over excuses.
4. What you have left to complete before the final deadline.
5. Screenshots of where your program is currently at. They don't have to be exhaustive but they should adequately depict the current state of your project running.

finalproject.zip

- Your commented code up until this point.

Final Deadline (Tuesday, December 10th at 11:59pm):

finalproject.zip

- Your commented code, fully implemented.

Interview Grading and Project Presentations

- Interview grading will occur during the final week of classes and finals week.
- Attendance at project presentations for all students from all sections is mandatory.
- All projects will be presented on **TBD** during which we will award prizes to the projects for different categories. The categories will be:
 - Best project to show to a friend
 - Best project to show during an interview
 - Most fun
 - Most beautiful
- Prizes will consist of eternal fame and glory.

