Project 2: Group Game

The aim of the group project is build a second, somewhat more complex arcade style game from the ground up, in a team setting. Groups of 2-4 members work best, with 3 often being an optimal number.

For this project, you will have roughly 5-6 weeks to go from concept to final product. My expectation is that this should be enough time to perform a **solid** implementation of a **multiplayer game** akin to a Networked Gauntlet, or Super Mario Brothers (obviously, there's no need to have 40 hours of game play here). If you want to do a game that is relatively light on graphics, (say poker, or backgammon) that's probably ok too, but I will want to know in your design document where your effort is being placed (perhaps AI). You can view the lowbar complexity requirement as containing all the elements of PacMan (with some additional complexity in 'state based behavior' due to power ups), plus networking, and a large scrolling world.

Teams of 4 will be held to a higher standard for the overall game (this can be obtained either via increased complexity or improved polish).

Deliverables:

- Project Pitch (brief in-class presentation)
- Design Document (written document)
- Individual Status Reports (Due each Sunday after the design document is due)
- Group Status Report (brief in-class presentation)
- Working Game (on display at showcase)
- Git Repository & Documentation (turned in at showcase)

Each deliverable is described in detail in the following sections:

Part 1: Project Pitch *(15 pts, Due Nov 2)*

As with Project 1, here the pitch is a short, in-class presentation for you to share a game concept with the class and get some immediate feedback before you finalize your proposal. The pitch should be a short (5min) presentation that contains the essence of your proposal:

- A description of the game idea and overall mechanics (what will the player do?)
- A description of how your game meets the complexity requirement.
- Thoughts on how you will task out the development between group members

You will be graded on how well you communicate the ideas behind your game, however, I will also give you feedback immediately as to whether I think the game is reasonable in terms of complexity – hopefully, this will help you refine your ideas before the design document is due.

Part 2: Design Document *(65 pts, Due Nov 4) [Soft Deadline]*

[65 points] Design Document:

As with the previous project, the first formal step will begin with a high level design document. In this first week, you should map out the core ideas behind your project. Again, this document should serve to help me understand your project and what to expect when it is complete. I don't want details like a class hierarchy or UML.

Your document should come in around four or so pages and should contain three main sections:

Game Overview: Here, you should describe at a high level what the game is. Plan to spend around a page/page and a half on this part. You should be sure to cover at least the following points:

1. A description of the game itself?
2. What's the genre; what's the camera position?
3. How will the game be played? What is the player's objective?
4. What are the visual entities in the game?
5. What types of interactions are possible?
6. What makes this idea interesting, or why do you think this will be fun?

This section should let anyone know what to expect in your final project. To help clarify your intention, it makes sense to draw parallels to existing games (e.g., we're going to build a multi-player/multi-level defender with bosses at each level's end). **Note that the description here *should all be with respect to your low-bar* expectations**. Don't set your sights too high here, there will be time for that later. This section should describe a feature set that you're ready to commit to and that meets the expectations outlined in the first section of this document.

Development Strategy: This section can be brief, but I want to know how you intend to schedule your work.

Begin by describing your starting point: do you have existing code you'll be reusing or are you starting from scratch? Break the development process into 2-3 main subgoals/milestones and describe the role(s) each team player will play to achieve each of these milestones.

Be sure that one of your milestones corresponds to the "Lab" session on November 20. I will want to see this milestone functioning and you should describe the feature set that you expect to be complete for this deliverable. As before, the goal here is not to write the timeline in stone, but to help you map the project out, and begin working in a well paced manner. Feel free to make this section a bulleted list if that seems most appropriate to you.

High Bar: The fourth section of your document should show how your low-bar version could be transformed into your *high-bar* version. Keep this section realistic, and use the knowledge you gained from Project 1 to scope your ideas.

Low Bar Checklist: Finally, *on a separate page*, create a checklist of low-bar items. In essence, you want to highlight the features you've discussed above and boil it down into ~6-12 features that I can look for in your final product. Each item should have a short name followed by a brief description of the item (1-2 sentences typically). As with Project 1, I will use this list to evaluate your project and gauge the overall complexity to ensure it will earn an A/A- if well polished.

> **Draft Complexity Rubric:** The following point system is meant as a guideline and is subject to adjustment/revision. I've put this in, in response to the comments from the class this year, so let me know if this is useful or makes things harder/less clear. I've highlighted what I see Multiplayer Networked Gauntlet to contain with respect to complexity. You can get credit for going "above gauntlet" complexity; however, once you exceed Gauntlet complexity, you will get reduced points for further additions.

- Scrolling World (0, 20 pts). [M, S, Z]
- Platforms and gravity (0, 30pts**)** [S]
- Realtime Game: (0, **15** pts) [M,S,Z]
- Multiplayer: (0, **10** pts) [M,S]
- Multiprocess (50 pts): [M]
  - You can earn these points by separating your game into two processes; this will have very similar implementation difficulty to networking in the lab environment. The requirements here are to create a client/server separation to the game, although it's ok if the client and server are run on the same machine. A requirement is for critical game events to be displayed by the server process (via stdout), and for a command-line client to be able to interact with the server/other client (a full featured command-line client is not necessary, but some indication that real—time-message passing is happening is the goal here). You can't get these points *and* the Networking points. Just one or the other (this is essentially networking).
- Networking (25, **?** pts): a simple networking component (quasi-turn based, messages that don't need to be passed with precise timing);

    - If you want to implement "better" networking, please discuss it with me first. Possibilities include: networking over LAN (real time networking well functioning over a LAN); networking functioning over the internet

- Power ups (10, 20, 25+ pts) – a few simple power ups that are implemented with ~3-5 lines of code and a separate static graphic fit into the lowest category; scale this up by quantity or complexity (e.g., powerups that require new animation sequence); 25+ points for extremely diverse and rich set of powerups [M: 10, S: 20, Z: 10]

- Art (0, **10**, 20) – incorporating procured genre relevant artwork; complex artwork requiring many assets or animation sequences [M: 10, S: 20, Z: 10]

- Isometric world (**0**, 25 pts) – this implies genre relevant artwork; but not necessarily complex artwork with many assets or animation sequences.

- Rich behavior (**0**, 10, 20+) – enemy/character behavior that is situation/character specific (simple: requires multiple parameterizations, more complex: requires multiple algorithms)

- JoyCon controls (10) – enable controls with a Nintendo switch joycon controller. My setup includes Mac Laptop to which I can also add switch joycons. I expect if you do this, there will be some back and forth to ensure that your setup works on my hardware. Thus, 10 points.

- Other? – is there a basic category that's missing here? Feel free to propose it along with a complexity "score".

Super Mario Brothers: ~ 115pts , Networked Multiplayer Guantlet: ~115pts

Grading: The paper will be graded as follows:

15 points will be based on the overall quality of your delivery;
50 points will be based on your coverage of the remaining items above
up to 10pts of extra credit will be given for extremely creative projects

A note on creativity: designing a good game is a creative process, and one that has some inherent risk. That is, not all creative and interesting ideas will actually yield fun and interesting games. If you want to clone an existing game, that's fine, and it's a good way to help ensure that your game will be reasonably fun. However, in doing so, you've essentially outsourced the creative task to someone else.

Turning it in: The paper is nominally due at midnight. **Instead of emailing me the document, you must check it into a git repository and give me read access to that repository** (you can add me to your project on the encs gitlab server). When I clone your repository, I want to see the design document and that there is at least one checkin for each group member – it can be trivial (creating the issues.mkd file, for example), but I should be able to verify that everyone in the group has commit access.

Part 3: Individual Status Reports *(7 pts per report)*

After the design document is due you must begin to send me individual status reports each Sunday. Thus, the status reports are due:

November 8
November 15
November 22
November 29

The status reports should be a brief synopsis of the week's activity to keep me informed about how your group is working. In particular I want you to address the items (often one sentence per item will be sufficient)

1. What did *you* do since the last status report?
2. How is your team working together? (if you're having team dynamics issues, let me know!)
3. How confident are you about meeting your low bar objectives?

The status reports are critical because they can help me intervene if a group is not functioning well. It is in your best interest to be honest and forthcoming in these reports as we may be able to avoid costly missteps if issues can be identified early. **Set an alarm on your cell phone!**

Part 4: Group Status Report *(15 pts, Due November 16)*

The status report is a short, in-class presentation to help share your progress with the rest of the class. As with the project pitch, the status report should be short (~4min) but it should also convey enough detail that it's clear you're making progress on your game. In particular you should:

- Prepare a screen shot or demo of your game in it's current state
- Describe what remains to be done to meet your low bar goal
- Describe one technical challenge you faced and how you solved it

You will be graded on how well you deliver your report and communicate the issues surrounding your technical challenge.

Part 5: Game Showcase *(out of 310pts, Due December 9)*

The final game showcase will be held on the last day of class, or during our final exam slot as decided by the class. As with Project 1, *attendance at the showcase is mandatory*. Guests are typically invited to the showcase, so it's critical that your game is finished and in good form!

Deliverables for the showcase:

- a git repository demonstrating the CS447 workflow with your final version as the working copy
- a readme indicating: (1) controls for the game; (2) cheat codes; (3) a list of the originally proposed low-bar goals and their current status (complete/partially complete/incomplete); (4) a list of other goals that were completed but not proposed in the original low-bar.
- a statement indicating the licensing terms for your game (e.g., a creative commons license)

Roughly speaking the grading will follow these guidelines:

- 60 points will be for basic functionality. If you've created some sort of functioning game, you will get these points.
- 100 points will be based on playability and the overall level of polish. If the game is fun to play, balanced and has appropriate controls, it will score well in playability. If the game is aesthetically pleasing (art and sound), has appropriate transitions and splash screens, and meaningful starting and end states, it is probably polished. A game that is both playable and polished will likely impress a non-computer science student.
- Technical complexity points will be awarded roughly corresponding to the draft rubric above; Mutliplayer networked gauntlet will achieve ~120 points in complexity.
- 30 points will be based on how well your git repository follows the CS447 workflow and whether you've included the readme and license terms as described above
- I'll give upto 10% extra credit for games that go beyond expectations in terms of quality or complexity.
- Gauntlet, scoring full credit for git/docs and playability will score in at ~305/310 points.