Project 2 (Part 1): Due November 2

[80 points] Design Document:

As with the previous project, the first 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 close to five 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 at least two pages on this part. You should be sure to cover at least the following points:

1. How will the game be played?
2. What types of interactions are possible?
3. What are the visual entities in the game?
4. What will the players' goals be?
5. 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 at the end in "*Where to set your sights"*.

Development Strategy: In this section, you should describe what you see to be potential sticking points in your low-bar goals. *Begin by describing your starting point: do you have existing code you'll be reusing or are you starting from scratch?* Then examine what you see to be the hardest parts of the project or that parts with the most unknowns. Describe how you will approach these aspects of development. Finally, break the development process into subgoals/milestones and describe the role(s) each team player will play to achieve each of these milestones. I don't want to see a day by day description of your plan, think of this as what you believe you'd plan to say in a weekly status report. Be sure to include an alpha milestone and a description of what you expect to be complete at that point. 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. This part will probably be 1-2 pages, but this limit should be considered flexible, the goal here should be completeness, not page count.

Technical Showpieces: Each game should have two technical showpieces. If you're in a four-person group, or a group of three 547 students, you should have three showpieces. In general, a technical showpiece is either: an aspect of your game that makes it more complex to implement (and more technically interesting) than say Pac-Man, Space Invaders or Frogger; or an aspect of the design process that makes your project more ambitious. High quality AI, interesting datastructures/algorithms for collision detection and point location, or a multiplayer game that uses networking all count as technical showpieces. Similarly, adding a group member who is off campus, or incorporating play testers into your project in a formal way would all count as well. You should spend at least a page on describing the technical showpieces you are planning

for your low-bar implementation, basically the goal here is to convince me that they are non-trivial.

High Bar: The final section of your document should show how your low-bar version could be transformed into your *high-bar* version. This is the place where you can dream a bit (but try to keep it realistic). Think of the high-bar as where you expect the project to go if you get really excited about it and everything else goes according to plan. Spend as much space as is required to paint the picture of how this game would be different from the low-bar proposal, and to describe how the new features would integrate into your development plan time line.

As before, the goal here is three-fold: to help you plan out your game before you begin implementation; to ensure the scope of the project is reasonable; and to provide you with an opportunity to pitch an idea using a written document.

Where to set your sights: After design, you'll have about five weeks to finish your project, so keep this in mind when determining how complex it should be. My expectation is that this should be enough time to implement a reasonably complex arcade style game. Think about something along the lines of Gauntlet or Super Mario Brothers or Legend of Zelda (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 be sure to describe in your design document where that effort is being placed (perhaps networking and AI).

Grading: The paper will be graded as follows:

60% of the points will be based on how well you address the questions above;
30% of the points will be based on the overall quality of your delivery;
10% of the points will be based on how creative or interesting the project is

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. As a result, I will take a points off of your design in the creativity department for clones. For all other aspects of the project, I'll treat clones along the same lines as original game ideas – so there will be no further penalty for making a clone. The only potential exception to this will be extra credit where I may give some additional points to games that are especially creative or especially well implemented, etc.

Due Date: The paper is nominally due at midnight, but I'll try to get these back to you as soon as possible so you can begin coding. So, if you finish early (especially a day or two) feel free to turn it in—I may have a chance to read it sooner than I would otherwise.