

FINAL PROJECT, Informatics 122, Software Design II

In this project, teams will collaborate on designing and implementing a single software system. The class will be divided into two large teams of about 15 students each. Each such team will be responsible for designing and building the *same* system. That is, we will design and build the *same* software system *twice*, each incarnation by a different group of students.

Each large team must figure out a way to partition the overall effort into smaller efforts. This can be done flexibly: the large team can decide upon how many smaller teams there will be, what the granularity of the smaller components will be, and how many students will form each of the smaller teams (this may be unevenly distributed, with smaller and larger subteams if so desired).

This document intentionally contains few specifications with regards to the assignment structure; it is up to the individual teams and subteams to decide upon the functionality of their components in the context of the needs of and facilities offered by the other components.

The objective is to design and build “Pac122”. “Pac122” is a simplified version of the classic PacMan game. In this game the player controls PacMan, who must collect all of the pellets on a level, while avoiding ghosts that try to eat him. Power pellets temporarily make the ghosts vulnerable to PacMan, causing them to flee.

Your version of the game must have 3 different levels. They can be designed however you like, but should consist of walls, pellets to be collected, and “power pills”. A flash implementation of the game can be found here for reference:

http://www.thepcmanwebsite.com/media/pacman_flash/

You do not need to exactly recreate each element of the game, but its spirit should be captured, in terms of clearing a board of pills, while avoiding ghosts.

Your version of the game can be played in single-player or multi-player mode. In single-player, the player is PacMan and the ghosts are controlled by the computer. Up to four other players can join, each using their own computer, possibly at a remote location. These players would control the movements of the ghosts in the game. Play is level by level, and the game keeps a perpetual list of high-scores.

The schedule will be as follows:

1. **Friday November 23:**
 - a. assignment out
2. **Tuesday November 27:**

- a. each student brings to class preliminary thoughts, written down and to be handed in to Kristina at the beginning of class, on how the overall system should be organized and partitioned;
 - b. each group, in class, agrees upon a preliminary breakdown of the overall system into subcomponents; rough functionality is assigned to each of the subcomponents; and subteams are assigned to each subcomponent or set of subcomponent(s);
 - c. by the end of the day, a summary of the agreement is e-mailed to Kristina
3. **Thursday November 29:**
- a. each subteam brings to class preliminary APIs, to be handed in to Kristina at the beginning of class, that it designed for its component(s);
 - b. in class, the subteams discuss the contents of the APIs and how suitable they are for each of the other teams to use; each subteam should ensure that the needs of their part of the program are met by the other subteams' API's, and vice versa
4. **Tuesday December 4:**
- a. each subteam brings to class final APIs, to be handed in to Kristina at the beginning of class, that it designed for its component(s);
 - b. from here on, *no* changes to the APIs are allowed, with the exception that Kristina may approve requests for such changes – change requests should be e-mailed to Kristina, André, Alex, and the other subteam(s) affected;
 - c. from here on, thus, the implementation of the system starts, faithfully following the design as was created;
 - d. each subteam should implement the component(s) for which they are responsible;
 - e. responsibility for integrating the code may be divided as you see fit.
5. **Thursday December 13:**
- a. each of the two overall teams brings to class a working version of their “Pac122”;
 - b. we play;
 - c. we discuss

You will be graded on:

- the overall quality of your team's implemented project;
- the quality of your subteam's submitted API and code;
- your subteam's helpfulness and responsiveness to questions and requests from other subteams; and
- your subteam's ability to “protect” the conceptual coherence of the component for which you are responsible

These last two points may seem at odds with one another, but this is part of the balance of designing a multi-part project. While we will not be able to provide many final answers, Kristina, André and Alex will be available for advice about approaching this very open-ended assignment.

With the project divided over about 15 students, each person should not need to write very much code. The challenge lies in effectively dividing the work so that each person's effort works together.

Finally, there will be team evaluation forms due on Tuesday December 4 and Thursday December 13. New evaluation forms are available online.

Good luck!

The two teams:

Team One:

- PURPURA, DAVID
- APPLEBAUM, WARREN
- CHISLUM, ALTON
- BOSCH, CHRISTOPHER
- LIU, ZHE
- SAM, VINH
- HUANG, ALLEN
- BEDFORD, AURORA
- KOLLA, SUBODH
- JONAS, NICHOLAS
- IGNACIO, JAN
- BAUTISTA, JEREMIAH
- CHIU, ARTHUR
- KWOK, MATHEW

Team Two:

- ESQUENAZI, NATHAN
- YEGANYAN, MICHAEL
- DEMPSEY, MITCHELL
- DAUZ, JONATHAN
- DYKZEUL, BRADLEY
- LEE, RICK
- STEWART, DAVID
- HIRANO, SEN
- LAVAVESHKUL, MICHAEL
- SHAFER, THOMAS
- SHI, LINDA
- BAMBAEEROW, CAMERON
- KNOBEL, JACOB
- LAM, CYNTHIA