

24-780 Engineering Computation

Project

Dues:	Team preference	Due 10/30 (Wed) 23:59 by E-Mail to TAs
	Conceptual design:	Due 11/04 (Mon) 23:59 @ Canvas
	Detailed design:	Due 11/12 (Tue) 23:59 @ Canvas
	Component code:	Due 11/19 (Tue) 23:59 @ Canvas
	Beta-version code:	Due 11/26 (Tue) 23:59 @ Canvas
	Final-version code:	Due 12/03 (Tue) 23:59 @ Canvas
	Final Presentations:	12/04 (Wed) Final Presentations

(Final presentation file must be uploaded to the Canvas by 12/3 23:59)

Team self-evaluation: Due 12/11 23:59 by E-Mail

Weight: $25\% = (2\% \text{ (Conceptual design)} + 2\% \text{ (Detailed design)} + 3\% \text{ (Component code)} + 3\% \text{ (Alpha-version code)} + 15\% \text{ (Final-version code \& presentation)}) \times \text{Weight from self evaluation}$

You will be designing and developing graphical and interactive software package using C++ for one of three applications:

- Education (e.g. digital flashcard)
- Entertainment (e.g. shooting game)
- Engineering (e.g. robot path planner)

Starting Fall 16, no tower-defense variant. We have seen too many tower-defenses in the past.

One team can have minimum four maximum six students. Note that everyone in your team must write some code, and the bigger the team, the more difficult to integrate individual codes.

An auditing student is NOT obligated to join a team. However, if a team accepts an auditing student, an auditing student can work on the project. One team can have up to one auditing student, and the work load of the auditing student must be substantially lower than a student registered for a regular grading option.

Each project member, except an auditing student, must be in charge of at least one functional component of the program. A functional component can be either one function or one class. If there is some uneven workload for coding, it is ok as long as each member contributes – a member who takes a light coding can take a heavier load in report writing or presentation.

The system should have two features: a graphical display (2D, 3D, or mixed) and user interactions. The amount of effort you will spend for the project should be three or four times of a regular problem set. You can take some ideas from an existing software program. However, there must be some original idea in your final product.

Your team has six weeks to make a conceptual design, make a detailed design, develop functional component codes, assemble component codes into a system, test the system and prepare a presentation with a live demo.

We will be using the Canvas for recording your team's software-design and development processes. Each week's progress, final product, and final presentation must be uploaded to the **File Exchange** area of the Canvas.

Team Member and Team Name Preference

You may express your preference regarding with whom you would like to team up. Each team must consist of minimum four to maximum six students. The group leader must send an E-Mail to Teaching Assistants with

- team name,
- the list of members, and
- team leader name.

by the deadline. Each team will have an area in the Canvas where you can post and share materials and codes for the project with team members. You can also start discussing what to build.

Conceptual design

Group Meeting 1

Meet as a group and brainstorm various ideas for your software product. Choose your teams' best product idea, and give it a temporary product name, and discuss the idea further by answering the following questions:

Product Identification

What are the basic functions of the product?

What are the special features of the product?

What are the performance targets?

Market Identification

What is the target user group?

How large is the user group?

What are the competing products?

System Description

What are the major functional components in the product?

(Assign each of the components to one of the team members.)

How should the functional components be assembled together?

It is also helpful to draw a few rough sketches of the product running images.

Immediately after the meeting, or during the meeting, post the product name, a one-paragraph description of the product, the Product Identification, the Market Identification and the System Description, on the Canvas under your team's directory. In your document, clearly indicate who will be in charge of each component.

Detailed design

Individual Work (between the first and second group meetings)

Each team member will refer to the conceptual design document and draft high-level design of your code with pseudo code, flow chart, or state-transition diagram. Post your high-level design document on the File Exchange area of your team's directory. Include your name in the high-level design document.

Group Meeting 2

Meet as a group to review and discuss the flowcharts and pseudo codes. After each member presents his/her flowcharts and pseudocode to the other team members, discuss and confirm that each member's component design fits the initial conceptual design. Modify flowcharts and pseudocode as necessary during the meeting.

In the same group meeting discuss how all the components should be assembled together to form a complete product. Discuss how all the components are interfaced to each other.

Also make sure to discuss and agree on:

- what computer platform the team will use for the final assembly of the functional components.
- who will be the assembly leader, who takes the lead in integrating all the components into a single assembly. (As this may be a challenging task, you should appoint a member who has stronger programming skills.)
- what data structures and classes should be defined and shared among the members.

Immediately after the meeting, or during the meeting, write a summary of your discussion, and post it on the Canvas under your team's directory.

Component code

Individual Work (Start immediately after the second group meeting.)

Each member writes the code for the component(s) that he/she is in charge of. Also write a code for debugging and test the component code. Post a 1-2 page document that describes the code that you developed and how you tested the functionality of the code. The assembly leader should keep in touch with other two members so that all the component codes will fit together in a consistent way.

Beta-version code

Group Meeting 3

After all the components are developed, meet as a group to assemble all the code into a single system. Work together as a group to debug and integrate all the code together.

Discuss how the two requirements, a graphical display and user interactions, can be achieved in the final product. Also decide how the functionality and the robustness of the system should be tested.

Immediately after the meeting, or during the meeting, write a summary of your discussion, and post it on Canvas under your team's directory.

Your alpha-version code needs to be complete in terms of functionality and has to run as a unified system even though it may have some bugs. Submit your alpha-version code on the Canvas and post a 2-3 page document to show that the system runs and achieves the intended functionality.

The document has to be posted to the Canvas under your team's directory.

Final-version code

Continue testing and debugging the system code, and submit your final-version code on the Canvas. The document has to be posted under your team's directory.

The final-version code submitted to the Canvas must include:

- Complete package that includes the executable file and necessary data files.
- A user guide.
- All the source files.
- Documentations that includes: (1) Compile instruction, (2) list of external toolkits and where these can be downloaded if any. (Everything that has been used in class doesn't have to be included in the list (2)),
- Final presentation package

Group Meeting 4

Meet as a group after your final-version code is completed. Discuss how the team will make a presentation and show a demo in the final presentation. Presentation package must also be uploaded to the Canvas under your team's directory.

Final Presentation

Each team will be asked to make a product presentation + Live Demo. Use two laptops, one for up-to-three-page product presentation (Imagine you are promoting your product in an exhibition), and the other for live demo. If you have more than one page, run it in an auto slideshow.

Since this is outside of the regular class hours, if you have something that you cannot miss, you can leave the presentation to your other team members. But, please make sure at least two team members can be in the presentation.

Also the final presentation package must be uploaded to the Canvas by the day before the presentation 11:59pm.

Set up your laptop so that you can run your program.

Team Self-Evaluation

After the completion of your project, I will ask everyone to grade your team members. I will set up a page so that everyone can anonymously (anonymous to other team members) submit the evaluation. The self-evaluation will be used for weighting the individual score for the project.