

Final Project Demonstration and Report Requirements

This handout describes the requirements for the final project demo and project report.

1 Demo (6%)

Your TA and I will see each group's final demo. 4% of the mark will be awarded based on functionality and 2% will be awarded individually based on the discussion of your implementation and answering questions. You will give an oral presentation describing your project. Each member of the group must perform part of the presentation. Be prepared to answer many questions about what you did, the issues you had, how it works, etc.

Please be *Ready to Run* at the appointed time. *Ready to run means that your code is up and running before the demo starts.* Time will be tight between some demos so we will not have time to wait for people to set up.

During the final demo, we will be:

- 1) Checking that there are no glitches in the video update
- 2) Ensuring that there are timely responses to user interface manipulations.
- 3) Checking the game logic functions and the speed and quality of the AI functions.
- 4) Checking the corner cases.
- 5) Expecting a demonstration of built-in error handling in your system. If the user enters an incorrect input sequence, your system should handle it without crashing. What types of errors have you foreseen? What is your method of handling them (remember error handling must be both predictable and reliable)?
- 6) Checking the quality of the overall game play and the multithreaded implementation.
- 7) Requiring a discussion/demonstration of any special features of your project *plus* individual responses to questions about your implementation.

You should be able to complete items 1-6 in 20 minutes (15 minutes is better) so think through the final demonstration presentation beforehand. Individuals may receive different grades for the demonstration depending on how well each of you knows the project.

2 Group Report (8%) [10 page maximum]

The report should be the complete documentation of your project and strictly technical information. It should be a self-contained document that includes all references or pointers to all the information and documentation that you needed, such as data sheets, etc. A test for completeness of the report is whether another person could take your report and figure out how to make your design work, modify it or maintain it. The TAs and I will also be looking at your own code for style and, especially, *comments!* (Remember Extreme Programming requires a defined coding standard).

Please follow this structure in your report:

Introduction: Provide an overview of the project. What is it you are trying to build and why? In other words, give a high-level description of the project goals. Give your team number and members and who was partnered with whom.

System Overview: Provide an overview of the system including (but not limited to):

- a Task/Stream diagram delineating who worked on each task
- a table/figure illustrating all of the threads used in your implementation (be sure to indicate which task(s) each of these threads implemented)
- a “User’s Manual” (i.e. how to operate the user interface)
- a list of bonus features you’ve chosen to include
- a table listing the number of unit tests created for each task (the code for the unit tests as well as your final implementation should be given in appendices)

Outcome: Results. How well it works (or not). Suggestions for further work or improvements. What type of Robustness/Reliability have you built into the code?

Description of Your Implementation: What do each of the threads in your system do? How do they work together (shared data structures, etc)? What unit tests are associated with what threads? What functions do these unit tests test?

Description of the Food Search/Maze Complete Algorithm: How do you choose what path to take to get to the food/complete the “maze”? Draw an algorithm flow chart (be sure it ties into your code). Be sure to include a description of how corner cases are handled (extra algorithm flow charts can be used if necessary).

Description of your “Artificial Intelligence” Algorithms: Algorithm description and flow chart (remember that this should help us understand your code, so be sure that it does =^)

Description of the extra (bonus) features in your implementation: A detailed description of the initial list from the system overview (e.g. what functions have you written to implement these features)? Why did you implement these features/why are they useful?

3 Individual Report (2%) [2 page maximum]

Here you have the opportunity to describe your contribution to the project and to give some additional feedback. This is where you can talk about the pain and anguish you went through in terms of what you tried, what worked, what did not, and how you eventually made things work.

It’s also a chance to address any challenges within the group dynamic/unfair division of work. The project is a significant component of your grade and I need to see significant evidence of the work you did on the project. Again, please follow the structure outlined below:

Introduction: Give your team number, your members and your partner(s). List all of the tasks you worked on personally.

What you did: Include discussions of items like:

- What function(s) did you act as navigator/driver for?
- How did you ensure that your tasks would integrate with your partnering group?
- What hurdles did you have to overcome?

- What did you do to ensure success, or at least improve the likelihood of success?
- What kind of source code control did you use?
- What did you learn?
- Were there any team dynamic issues?
- Anything else you spent your time on (related to the project :-)

Community Contribution: Please indicate your community contributions. I will observe who has been giving help on the bulletin board but indicate here what kinds of help you provided on the bulletin board and in the lab.

Course feedback: Any comments or suggestions you would like to make about the course for the future.

- Did the project timeline work, i.e., demos, deadlines? Could the project be started earlier?
- Did you like the “open” lab concept, i.e., do these modules by this time and do these tasks at the end? i.e., instead of trying to grade something each week. Please note, this is a new lab and new equipment, so your feedback will be helpful to us in determining if this is a good approach or if a different kind of project would be better.
- How did the lectures work with the labs? Independent of the labs, would more or less time on any particular topic(s) be useful? Any other constructive comments on content and organization would be appreciated.

NOTE: The two page maximum does not include the community contribution and course feedback. You can have an extra half page (2.5 pages in total) for the community contribution and course feedback sections.