CS 205 Final Project

Labyrinth or Blokus

There are four deliverables for this project:

	Description	**M/W/F class**	**Tu/Th class**
Deliverable I	Project Proposal	Friday, 3/10, 5pm	Friday, 3/10, 5pm
Deliverable II	Specifications, Task Breakdown and System Design	Monday, 3/27, 5pm	Monday, 3/27, 5pm
Deliverable III	Test Results and Project Status	Friday, 4/14, 5pm	Friday, 4/14. 5pm
Deliverable IV	Final Product	Friday, 4/28, 5pm	Thursday, 4/27, 6pm

Deliverable I: Project Proposal

The project proposal will consist of a short document listing the team members, who the team lead will be, and the language in which the team has decided to implement the application.

Deliverable II

Specifications, task breakdown, and system design

- Each group will together hand in a list of tasks for the system along with a difficulty estimate for each task, a relative importance of each task, dependency relationships among the tasks (if there are any), and a tentative owner for each task.
- A simple way to make this kind of list is by assigning easy/medium/hard and nice-to-have/need-to-have/essential to each entry, along with a team member's name next to the entry.

Deliverable II, continued

- Each group should also provide two pages of specifications for their system, and four pages of graphical models of the proposed system. It's up to each team to decide what form the specifications will take (e.g., a complete list of features and how those features are implemented) and which graphical models will be provided (e.g., class diagrams or one of the graphical models we discuss in class).
- The specifications should also include a description of all GUIs required for the project. As a guiding principle, the grade for this part of the project will derive from how easy it would be for another development team to create a system based on the specifications that you provide.
- Criteria for evaluation will consist of:
  - completeness and realism of task list
  - completeness of graphical models
  - clarity and quality of writing
- Specifications Document: 15% of grade

Deliverable III

Test Results and Project Status

- Each group will together hand in ten pages of evidence of testing of their system.
- This deliverable should be a mixture of screen shots (including erratic or correct behavior), code snippets (indicating where errors were located through testing), and verbal descriptions of the system's responses.
- Also, this document should contain a status update in the form of the task list from Deliverable II, with annotations showing the status of each task (on track, behind, completed, cancelled), along with any new tasks that have been created and what their status is. By this point you will have to have implemented a fair portion of your system.

Deliverable III, continued

- Criteria for evaluation will be:
  - quality of the documentation of the tests (do they really reveal how the code is performing?)
  - completeness and realism of task list
  - clarity and quality of writing
- Test-Results Document: 15% of grade

Deliverable IV: The Final Product

- The fourth deliverable is the completed product itself.
- For this deliverable, each team will submit instructions on how to play their implementation of the game along with the application itself (and instructions for compiling/linking/building the application).
- As a guiding principle, assume that any person who knows how to play Blokus or Labyrinth should be able to use the instructions to play the game using the application without having to communicate directly with the development team.
- Criteria for evaluation will be:
  - look and feel of the application
  - ease of use of the application
  - correctness of the application
- Final Product: 15% of grade

Appendix: the Source Code

- In addition, each team will submit the source code electronically to the TA.
- This should include in-code documentation explaining how the code is organized; what each class or function does; variable and class names that are intuitive; and clear identification in each part of the code of who the author of that part is.
- As a guiding principle, assume that "good" code is code that someone external to the team could read and fix or extend. Criteria for evaluation will be:
    - presence and descriptiveness of comments
    - uniformity of code
    - readability of code
- Source Code: 10% of grade

The Group Presentation

- Each team will give a 25-minute oral presentation in class, describing their system.
- The team leader will summarize the team's final product; each other team member will then describe a particular component (e.g., the GUI, the game-playing engine, etc.).
- Each team member should spend five minutes summarizing his/her part in the organization of the project: what he/she contributed to the original design; his/her role in system integration; and so on.
- When applicable, each team member should also spend five minutes demonstrating his/her part of the final product.

The Group Presentation, continued

- The team's presentation should also discuss what worked and what didn't work in terms of organization; and what the team would do differently if they could do the project over again.
- The team should plan to present their system on their own laptop in class.
- The final report should also include some summaries of gameplay results.
- Criteria for evaluation will be:
  - polish and salesmanship
  - comprehensiveness
  - objectiveness in self-evaluation and "lessons learned"
- Oral Presentation: 15% of grade

The Individual Report

- Each member will also hand in an individual report. This report will consist of three parts:
  1. The first part should be a half-page executive summary of what the program does and have an analysis of the development.
  2. In the second part, each team member will elaborate on the development processes: how the system was sketched out (including graphical depictions of the original and final system); how the task list was created; what meetings were held, and how often; how the system was broken down into components; who worked on which components; how testing was planned and executed; how the system was integrated; how much time and effort was spent on the activities of software development; etc.
  3. In the final part, the author will analyze the development process: describe which activities were accomplished easily, and which were difficult and why; how members communicated with each other and whether this communication was effective; what errors were uncovered during testing; whether development actually proceeded as planned; and so on.
- This document should be no more than ten pages, double-spaced, with 12-point font.

The Individual Report, continued

- Criteria for evaluation will be:
  - description of the development process, especially with regard to concepts covered in class
  - analysis of development process, especially with regard to the identification of what worked well and what didn't work well; and what you would do differently if you had to do this again (the "lessons learned")
  - quality of writing
- Written Report: 25% of grade

The Development Process

Design

- Teams should select a team leader (see description below). Teams should meet frequently to plan work. Time will be provided in class for meetings and discussion.

Version Control

- Each team should use a version-control system (e.g., github or gitlab).

Design reviews and status updates

- Periodically, the team will meet with the client (i.e., the instructor) to show proposed user interfaces and product behavior and to elicit and refine requirements (as needed). Dates of these reviews will be publicized in class and on the syllabus.

The Development Process, continued

Team leader responsibilities

- The team leader is responsible for guiding the initial design of the application, identifying the different components and subcomponents of the system. The team lead is also responsible for providing status updates to the instructor when necessary and for creating internal deadlines and deliverables.

Team member responsibilities

- Each team member is responsible for being pro-active: do not wait for your team leader to tell you what to do. Team members should help their team leader as much as possible by doing independent research and reporting their findings back to the team lead and to the group as a whole. They should also contribute to the design and planning stages as much as possible. Team members should keep a project journal describing their work throughout the project. This can be a soft-copy file with periodic entries; or an actual notebook; or separate sheets of paper. For extra credit, team members can submit this project journal at the completion of the project.

The Development Process, continued

Miscellaneous

- You will not be penalized if a member of your team does not finish his/her component or subsystem. However, the successful functioning of your component could be affected if another component on which you depend is not finished. In order to guard against such an impact, design your testing carefully so that you can demonstrate the functionality of your component without requiring other components to be complete.