CprE 388 Final Project

Requirements for All Projects

- You should .zip your project files and source code and submit it on Blackboard Learn.
- Include in your .zip a README.txt file that gives a brief overview of your project.
- Demo your project to the TA during lab hours, office hours, or via appointment.
- All applications should be "polished"; an app icon is required. Your project should be free of major bugs.

Due Date

You should demo your project and turn it in before the end of your last lab session.
Section A - Tuesday, December 4th at 11:50am.
Section C - Thursday, December 6th at 11:50am.
Section B - Friday, December 7th at 9:50am.

Create Your Own Project

You are tasked to design your own final project. Two example projects are listed at the end of the document to give you an idea about the expected complexity of your app and the amount of details expected from the rubric and schedule.

Before starting the project, create a 150 point rubric listing major features of your app. These features should be described from the user's perspective and weighted based on the value they provide to the user. Also include a 4-week project schedule. Think of the month long project as four weekly sprints. Each week you work to add new features to the app. At the end of each week, you should have a working, deliverable, app with a few more features than in the previous week. Keep the amount of time that your project is "broken" to a minimum.

Discuss the project with the TA to see if the concept has enough complexity:

1. Name of your app
2. Project requirements / 150 point rubric
3. 4 week project schedule

Extra Credit: Have your app submitted and approved on the AppStore. This process can take up to two weeks, so you'll need to finish by the end of Thanksgiving break. Use your own developer account, or e-mail Chad for information on barrowing his.

CprE 388 Final Project

Example Project 1 – Scrabble Utility App

I will create a utility app that will help people cheat when playing Scrabble.
Requirements Rubric:

- (60 points) The anagram solver tab allows the user to enter 7 letters (including blank tiles), and will find words that contain a subset those letters from a dictionary.
- (30 points) A scrabble board tab allows the user to input their 7 letters, plus the contents of the board. Results are shown on a results tab, sorted by highest point value.
- (30 points) The options tab allows the user to choose from a few dictionaries (large and small).
- (25 points) The options tab allows the user to save and load a game.
- (10 points) The options tab allows the user to switch between the original Scrabble board and the Words with Friends board.
- (20 points) When selecting a result, the word is shown on the scrabble board.

Optional Features:

- Use a DAWG (directed acyclic word graph) for storing the dictionaries to increase speed.
- Solve a complex board using a 100,000+ dictionary (TWL) in less than 1 second.

Project Schedule:

Week 1 – Find dictionaries; start designing UI. Finish a basic anagram solver.

Week 2 – Finish the user interface for the board. Begin implementing the options tab.

Week 3 – Solve the Scrabble board (reusing the anagram solver code, solving it row by row, column by column)

Week 4 – Finish the scrabble solver. Test.

Example Project 2 – Create an iPhone Game

Create a game on the iPhone (PacMan, Space Invaders, Donkey Kong, Angry Birds)
Requirements Rebric:

- (70 points) A working game.
- (40 points) Allow the user to save their progress.
- (20 points) A feature that saves the high score and initials of the user.
- (20 points) Cool graphics!

Optional Features:

- Use a 3rd Party Graphics/Game library (Cocos 2D)
- High-production value is a plus.

Project Schedule:

Week 1 – Google and read tutorials. Create test apps to demo what is possible.

Week 2 – Start working on the game mechanics by combining the test apps.

Week 3 – Finish the game mechanics. Store the game state. Create high score system.

Week 4 – Test. Work on cool graphics.