CS 242 Final Project Proposal Template

Crossword Generator
Karthik Bala (kbala2), Peiyuan Zhao

1. Abstract

1.1. Project Purpose

The purpose of this project is to create an html5 web game which helps players improve their crossword puzzle solving skills. The game will generate and display one set of crossing words at a time in a crossword themed grid. One of these words will be blank (save for the intersections with the other words) and players will fill it in, as in a bona fide crossword. Players may choose to change the difficulty of each puzzle through the GUI, as well as ask for hints. Players may create accounts to track their high scores over time.

The source of the crossword data: https://github.com/donohoe/nyt-crossword

1.2. Background/Motivation

I love doing crossword puzzles, and

Additionally, the GUI part would involve using web applications I've never used before, as well as extending an existing tool, both of which would appropriately challenge me as a programmer.

2. Technical Specifications

2.1. **Platform:** Website / Web app

2.2. **Programming Languages:** Python/HTML/CSS/Javascript/MySQL

2.3. **Stylistic Conventions:**

2.3.1. All the usual stylistic conventions for each programming language/set of tools/environments. For instance, snake_case for Python, capitalized table names in MySQL, camelCase for Javascript, etc.

2.4. **SDK:**

2.4.1. I don't think I'll use an SDK.

2.5. **IDE:**

2.5.1. Gedit.

2.6. **Tools/Interfaces:**

2.6.1. Google Chrome

2.6.2. Gedit

2.6.3. Flask

2.6.4. Required Javascript frameworks/API's

2.6.5. JQuery, Bootstrap for styling

2.6.6. MySql

2.6.7. Qunit for Javascript testing

2.7. **Target Audience:** Crossword puzzle enthusiasts

3. Functional Specifications

3.1. Features

The project's core functionality: given a list of crossword puzzle clues/answers, generate a set of crossing words in a grid. One of these will have blank spots.

The user should be able to:

-See the clues for each word

-Fill in the blank spots in the word

-Get letter by letter/word by word color-based feedback for answers

-Request and get hints for the current puzzle

-Create an account and login to track their high score

-Set the difficulty of the puzzle

4. Timeline:

4.1. Week 1

Goal: generate random, valid sets of intersecting words from the database of clue pairs. Remove all non intersecting letters from the desired "answer" word, and display the puzzle with its corresponding list of clues.

4.2. Week 2

Goal: Create a web based GUI using Flask/Javascript in which the puzzle is displayed. Allow for players to fill in the puzzle. Create a game loop which asynchronously loads the next puzzle.

4.3. Week 3

Goal: Add the following features: game clock, colored answer checking, backspace support, difficulty settings, dynamic collisions and length variation. Style the GUI – make it look nice.

4.4. Week 4

Goal: Add features such as hints, high scores, and player accounts (along with login/signup pages and a Player table in the database). Make logins persist across pages. Add security features such as defenses against SQL injection, XSS attacks, and encryption. Improve the styling.

5. Future Enhancements

Crosswords generally have themes around which the puzzle is based. The longer, multiple word answers involve plays on words relevant to the theme. It would be neat to use data mining or AI to allow for themed crosswords to be generated from the database. It would also be nice to have the difficulty dynamically increase over time.