Quiz Game Project

# Defining

## Potential Users

This program will be presented to a computer programming class, potentially my family, and possibly some friends. All of these people are somewhat nerdy so this is the type of questions that I can use in the game. This can also affect the presentation of the game. By using a “hacker’s type” presentation it can fit with the theme of asking nerdy questions.

## Design Constraints

It will be advantageous to be able to add questions without recompiling source code and republishing questions. Also possible would be pulling questions from an online server such as GitHub which is a possibility later on in development.

# Ideating

To handle questions easily, a class should be used to represent a question. Questions will be labeled A through D. A limitation here is that only capital letters will be used to represent answers. Online access will use either a csv or multiple line text file.

# Prototyping

Initial attempts to produce this game involved text appearing with an animation inspired by “The Matrix”. This was never completed due to inability to redraw previously rendered parts of the screen. Class based questions were successfully implemented however remote file access was unsuccessful in initial attempts.

# Testing

Hopefully I can get my existing prototype checked by the other students in Programming Level 2 and potentially some level 1 students. This will check if the interface is effective in presenting what is requested for information.

## Improvements

Iteration 2 needs to have some delays between each question. This will make it more readable to the user of what is happening. Then, external questions using online sources is unsuccessful. I will now try to design offline question reading if possible.

## Issues

On the mac computers the ifstream is not detecting the file. This is likely due to a permissions or path related error. Due to time constraints, online quizzes will for now be abandoned. Instead, the user will be able to choose a topic of quizzes from a collection of possible options.

## Slight Miscommunication

Unfortunately, I was unaware in the initial intent of this project was to ask about the design process. This means previous design ideas must be abandoned such as asking about “nerdy” topics. It also means offering multiple categories can be split into the six phases of the design process.

## Multiple categories

After switching to design process questions, I added a category for each step in the design process. The player can take as many or as few categories as they choose before ending the quiz and completing the game.

# Making

With the engine completed, I now had to program in all of the questions. I wanted to have at least 5 questions in each category. With 6 categories this meant 30 questions in total. This was programmed in using the existing engine. Additionally, graphical effects were introduced such as dashes above and below text, and display of which category each question was a part of.

# Sharing

After sharing the project with the other Programming Level 2 students, the key feedback was the limitation of the programs input. The program only allowed the user to input capital letters. While I would have liked to fix this issue, it is rooted in the main game engine using a string to represent the correct answer (“A”). To fix the issue, a conversion would need to be made from uppercase to lowercase letters which is not easily documented. I will begin work to make the improvement.

# Upper Case Improvements

After receiving the feedback, a mapping operation was created to convert between a lower-case letter to an upper-case letter. It was difficult because C++ does not natively support string#toupper and there was no easily usable loop to use char#toupper and return the value as a string. To solve this a simple map was created checking for a lower case letter a, b, c, d, e, f, or g, and converting it to the uppercase letter A, B, C, D, E, F, or G.