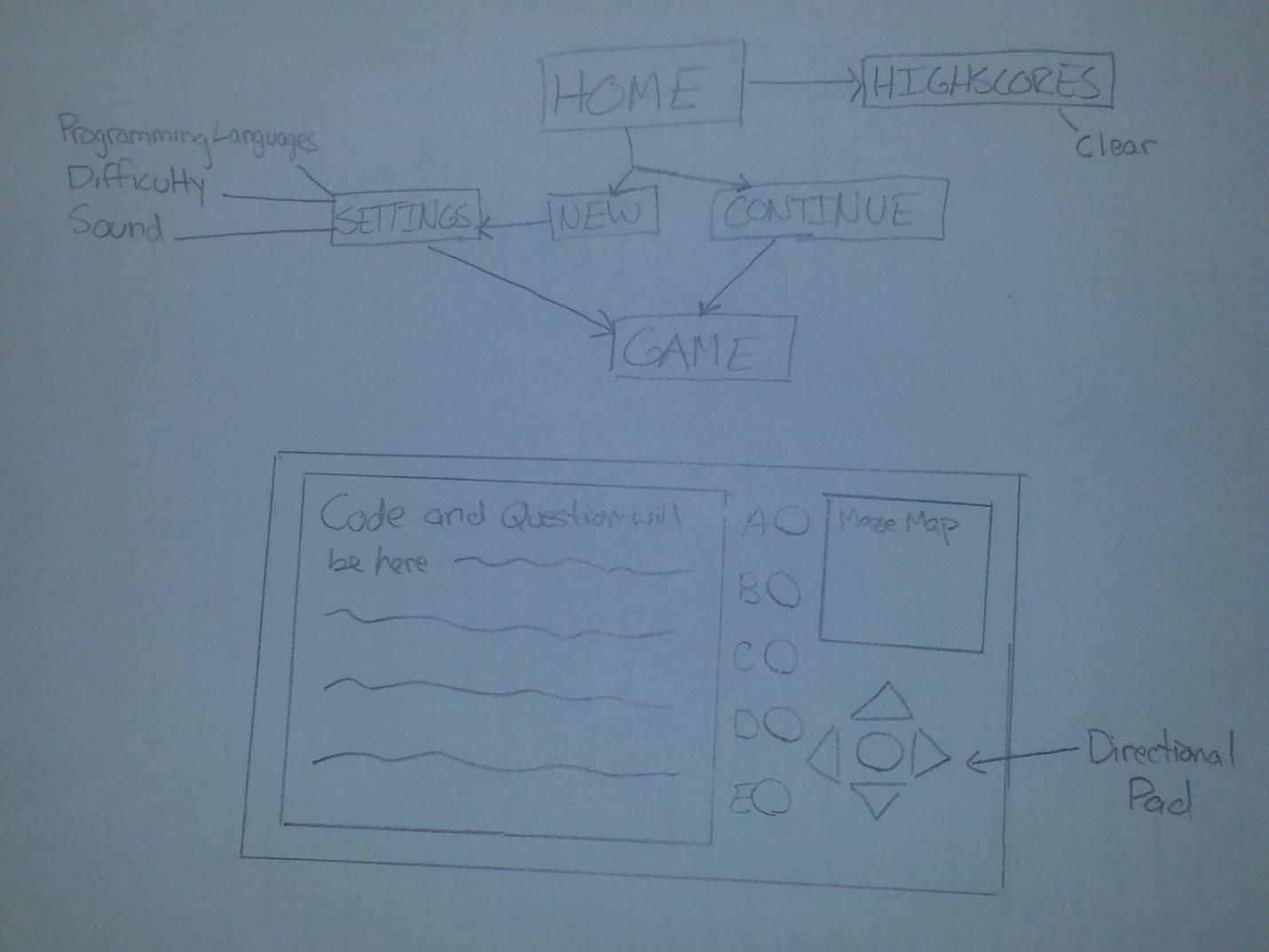
Product Backlog

1. Problem Statement: There is a need for an educational, interesting, mobile, puzzle game that can help teach basic programming skills.
2. Background: In today’s society, it is almost a necessity to understand programming languages; however, there is not any kind of game that teaches programming and entertains the user at the same time. The domain of our project would be the intersection of entertaining games, and educational tools. The targeted users of our product will be programmers of any age range who want to learn more about programming. There are no applications that are directly similar to our idea; however, popular gaming applications as well as popular educational tools could compete with our application to become well known. The limitations of most of the competing applications are that they are all either functioning as only a game, or as only an educational tool, where as we are striving to create an educational tool and a game in one.
3. (System Model Diagram on last page)

System Model Explanation: After starting the application, the user will be brought to a home screen. Once here, the user will have three options. The user may start a new game, continue a game he/she has started, or check high scores. If the user starts a new game, he/she will be taken to a settings screen with the options to adjust programming language of questions given, difficulty of the game, and sound of the game. After this the user will proceed to the game. If the user continues a game, he/she will simply be taken to the game where he/she left off. If the user chooses high scores, he/she will be taken to a high score screen with an option to clear the list if desired. Once in the game, the user must navigate through a maze using the provided directional pad. Throughout the maze, the user will be presented with multiple choice questions which he/she must answer using the provided A, B, C, and D buttons.

1. Requirements:

* Must Be Done:
  + Functional:
    - Multiple Choice Questions
      * Create a database containing multiple choice programming questions
    - Answer Verification
      * Verify that all answers to all questions are correct and understandable
    - Settings
      * Implement difficulty and sound options within the game
    - High score
      * Implement a local high score table within the game along with something to clear the scores if desired
    - New/continue
      * Allow the user to start a new game or continue where he/she left off
    - Maze/Map Creation and Functionality
      * Create a database with maze variations, along with maps containing the layout of each map.
      * Add functionality to the mazes and maps
    - Basic Game Rules/Setup
  + Non-Functional:
    - Database Performance
      * Ensure that the database sends questions, maps, and mazes efficiently
    - UI Performance
      * UI should be seamless
    - Secure Questions
      * Make sure only we can alter/add questions
    - Modifiability of Questions/Algorithms
      * Make sure questions can be easily modified
    - Modifiability of Maps/Mazes
      * Make sure maps and mazes can be easily modified
    - Question Algorithm Reliability
      * Make sure the algorithm for fetching questions is reliable
    - UI Testability
      * Make sure UI is testable
* Will Be Done If Time Allows
  + Functional:
    - Drag ‘n Drop Questions
      * Add a database of questions requiring the user to drag sections of code to the correct spot in the program
    - Graphics Definition
      * Add improved graphics to polish the look of the game
    - Seamless Screen Transitions
      * Make sure all screen transitions are smooth
    - Plot/Story/Motivation
      * Add a plot to give the user more of a reason to play
    - Actual Save Feature
      * Add an actual save feature rather than just saving the state of the game when leaving
  + Non-Functional:
    - Question Encryption
      * Encrypt the questions for added security
    - Record Statistics
      * Track statistics of players to allow for a score that can be shared/compared with others
    - User Modified Questions
      * Implement user suggestions or user added questions

System Model Diagram:

GitHub repository: <https://github.com/cheskett/Section-16>