CMPT 26 Project Group 8

Requirements Document

Liam Wilson
Tyler Kasper
Seth Traverse
Chang Liu

1. Domain:

The domain we have chosen is Entertainment. The application we will develop will be a drag-and-drop puzzle game that allows people to learn how to code in Java, C++, and Python. The concept will be similar to the way code.org (https://code.org/) or Tynker (https://play.google.com/store/apps/details?id=com.tynker.Tynker) teach coding in that you will be able to drag and drop modules that create code without the need to type. However, our game will be different in that:

- It will be specifically geared towards people wanting to learn Java, C++, and Python
- It won't be quite as simplistic or high level as code.org and Tynker; you will be working on real Java, C++, and Python code puzzles, with modules that create real Java code
- Whereas code.org and Tynker are geared more towards a young audience (child to young teen), this app will target a bit more of a mature audience (young adult and up) who are looking to seriously learn how to code in a fun way. As such, real code will be used and the puzzles presented will probably become more difficult than what is presented by code.org or Tynker.

2. The Problem:

The problem we are trying to solve is: How do you learn to code in a fun and engaging way? Poring over documentation and programming books can get boring very quickly. Videos on YouTube may show you what to do, but they might not actually engage you in the learning process. As well, you may not have the time to invest in reading lots of pages of code or watching an entire series of videos. This app will aim to teach Java, C++, and Python in logical increments within a short period of time, while engaging the user and challenging them in a way that will make what is being taught stick in their heads.

3. Purpose and Audience:

As mentioned previously, our audience will be young adults and up who want to learn how to code in Java, C++, and Python, but want to do it in a way that is:

- Fun
- Challenging
- Allows them to learn "on-the-go"
- Doesn't take up a lot of their time

Our app will be useful to:

- The Computer Science Community, because it will expand the Community by teaching more people how to code. It will also benefit the Community by providing a resource for anybody in the Community to learn or hone their skills in Java, C++, or Python.
- Businesses, because they may need a quick and efficient way to train new employees how to use Java, C++, or Python.

- Students, because they may need a quick and easy way to learn how to use Java, C++, or Python for a class they want to take or a co-op job they are planning to take.

4. Story Line

Upon opening the application once installed the user will be presented with a choice of three different languages (Java, C++, Python) and three different difficulties (Easy, Medium, and Hard) in two different selectors. This will affect the language and difficulty of the questions proposed so a user may escalate difficulty or change language at their own prerogative. The user will then select one of the difficulties for the chosen language and be presented with a grid of numbered buttons representing the questions of the chosen difficulty that the user may attempt. This grid will also act as a progress tracker as when a question is completed the grid button associated with that question will change appearance. Upon pressing one of these buttons in the grid the corresponding question will appear as a drag and drop puzzle for the user to attempt. A piece of incomplete code will appear in the top section of the screen with boxes as locations for draggable answers representing missing components. Below the question will be a bank of possible answers that the user may choose from to drag and drop into the empty spaces in the code. When the user has entered the correct pieces into the correct spaces a victory box will appear notifying the user of their completion of the puzzle. From this dialog the user may either attempt the next question or return to the language/difficulty selection. If a user is unable to complete a question they may advance to the next question at that difficulty level using a next button on the question screen.

The idea for this app comes from an amalgamation of many different apps, programs, and academic resources that we have used throughout our use. The question selection grid is like the question selection grid that many puzzle games use on iOS and Android. The drag and drop question and answer idea comes from a drag and drop programming tutorial called Scratch. The fill in the blank format for questions is also present on many quizzes and tests in the computer science faculty.

5. Required Functionality

- When the launcher icon is clicked, the application should open and display
  - The selector for language
  - The selector for difficulty
  - Three buttons for the difficulties: Easy, Medium, Hard
- When a user selects a language and difficulty the question selection activity should open and display:
  - A grid of numbered buttons representing the questions for that language at that difficulty level
  - A label instructing the user to select a question

  - Previously completed questions should have a visible difference in the appearance of the buttons
- When a user selects a question the question activity should open and display:
  - The incomplete code for the question
  - The draggable answers for the question
  - A “next” button
  - A “Main Menu” Button
- If a user successfully completes an activity a question completion popup should appear and display
  - A question completion message
  - A button to go to the next question
  - A button to return to the main menu
- For the question activity empty squares should appear where the holes in the code are and be able to receive and display a drag and drop answer
- Upon inserting the correct answers into the correct spots in the code the completion popup will display but otherwise will not open
- Questions must be able to be programmatically pulled from a resource file in some standard form (e.g. JSON or XML) for easy addition of new questions
- This app will keep track of questions that have been completed to allow users to see progress

6. Desired Functionality

We have a large list of possible functionality that we would like to choose from for delivery in the last sprint however there is likely more here than can be done in one sprint. As such a subset of these will likely be taken on in the last sprint.

- Hint Tokens
  - Every three (can change) questions a user correctly answers will earn the user a hint token
  - Hint tokens can be spent on a question to reveal one blank spaces correct answer in the code. The cost would be one hint per token
- User Statistics
  - The user would be able to navigate from the main menu to an activity that displays various statistics of their progress
- Reset Progress
  - The User would be able to completely reset all progress in the app from the main menu including any statistical data and records of completed questions
- Themes
  - The User would be able to choose from a list of pre-created themes to give the application a different look and style (e.g. Light Mode, Dark Mode)
- Victory Animation
  - Upon completion of a question the user would see a brief animation that acts as a visual queue that the question has been completed