Just My Type
CEN 4020
9/11/2016

Progress Report - Milestone 4

User stories implemented:

- As a user, I want a single player game mode that is fun. (BasketBall Game).
- As a user, I want refined typing test that can calculate my words per minute, so that I can measure my speed and progress.
- As a user, I want a functional settings menu, so that I can change the look of the app.
- As a user, I want a scoreboard that is connected to typing test, so that I can see my old scores and other users scores.
- As a user, I want an accurate Words per Minute calculation that progressively updates as I enter a new word.
- As a user, I want a typing test that has the ability to have the duration of the test adjusted.
- As a user, I want a typing test that I can easily reset my progress as I see fit.
- As a user, I want a typing test that has a variety of words to accurately test my typing ability and speed.
- As a user, I want to be able to easily dismiss the keyboard by clicking anywhere in the interface of the Typing Test.

Paired programming partners:

- Lauren added functionality to the settings menu, improved design of typing test.
- Andrew refined the Typing Test and did other smaller things around the application to make TypingTest.swift more modular and fit well into the TypingTest scene.
- Andre helped with refinement of the TypingTest class and the TypingTestViewController creation.
- Jorge and Asa created a new animation function in UIViewExtension.swift that allows new words to slide in after the user correctly types the current word.
- Richard implemented Basketball game mode prototype along with sprite work and work towards the creation of the scoreboard. Basketballviewcontroller basketballscene gameplay scene

Design Changes:

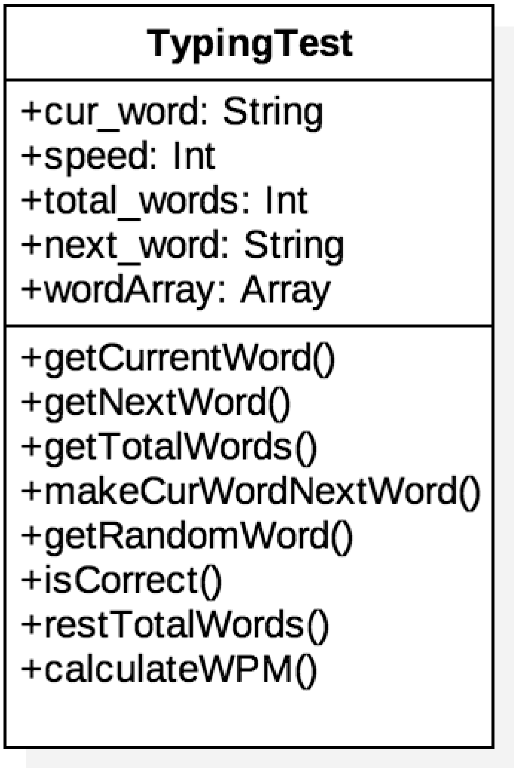
We have not made any design changes to our project so far.

What will be implemented in the next iteration:

- Other Game Modes

    - Basketball
    - 100 Yard Dash
    - Bowling
- Multiplayer Functionality
    - Multiplayer Functionality through the GameKit framework provided by Apple
- Leaderboards integrated with GameCenter such that players can compete with friends related by Apple ID’s.
- Ability for user to do a Typing Test that has sentences verses random words in a random order to replicate a more realistic typing situation.
- Implement User Interface testing such that the application can be more accurately and thoroughly tested through the rest of development.
- Add some background music and noises for when you get a word correct or incorrect for ambiance in the game.
- Test implementing a Word Per Minute calculation that takes into account errors and size of the words
- Refined Design
    - Move Typing Test to single player section
    - Put logo on the screen above the initial menu
    - Make Typing Test have a more pleasing User Interface

Class Diagram:



- Attributes
    - cur_word
        - String Type
        - Holds the current word in the TypingTest Class
    - speed
        - Int type
        - Holds the current speed in WPM calculated based on time passed and total_words
    - total_words
        - Int Type
        - Holds the current amount of correct total words
        - Is incremented by the isCorrect() function
        - Is reset to value 0 by resetTotalWords()
    - next_word
        - String Type

    - Holds the next word such that you can print this out to the user so they can be ready for the coming up word while typing the cur_word
  - wordArray
    - Array of Strings
    - Currently loads from WordList.txt using the data.components function
    - WordList.txt is a list of words that is separated by ‘\n’
- Operations
  - getCurrentWord()
    - Returns a String of the cur_word
  - getNextWord()
    - Returns a String of the next_word
  - getTotalWords()
    - Returns an Int of the total_words
  - makeNextWordCurWord()
    - Void that alters cur_word to equal next_word so that the next_word can be randomized
  - getRandomWord()
    - Returns a random word and also stores this random word into next_word
  - isCorrect()
    - Returns a bool based on whether the given parameter string is equal to the cur_word
    - If true
      - Make call of makeNextWordCurWord() to replace the current word with the next
      - Call getNextWord() to replace the old next word
      - Adds 1 to total_words
    - If false
      - Make current word do animation signifying it is false