Emma, Abhay, Darren, Ashley
CSC260
Professor Cass
15 February 2024

Grid Layout:
We tested the grid layout to make sure it was what we wanted by displaying it in the GUI. From there we were able to make stylistic changes like changing the size and spacing of the buttons, and the font and size of the letters on the buttons. Initially displaying the grid took some trial and error as we also had to figure out javaFX components like creating a layout and setting the scene and stage.

Button(s) Selection:
We tested if a selected button was being processed by printing the selected letters to the terminal. When we expanded our implementation to use a MouseDragEvent to select a combination of letters, printing to the terminal remained helpful to see which buttons were being added to create a string of characters.

Scoreboard layout
We tested the scoreboard layout similarly to how we tested the grid layout. To see how the scoreboard would appear in our view we displayed the initialized version which had a score and bonus words count set at 0 and an empty progress bar. This let us set the correct size and placement in the main stage of the display. It was easy to test the update scoreboard functions because the scoreboard was already being displayed.

Incrementing score
To ensure the score was incrementing we printed the current score to the terminal at various points in our code. This allowed us to ensure that the right functions were being called at the right time to successfully increment the score. This was helpful when our score output was incorrect. We caught a mistake where we had the score increment once in viewmodel main and again when the scoreboard was updated again.

Regular words/ Dictionary
We checked the integration of our dictionary by printing to the terminal at different stages of our integration. First, we tested the ability to determine if a given word was in our dictionary file and our wordDict class was behaving as desired. From there we connected the model viewmodel and model to check if a selected word was in the dictionary.

Bonus words
After successfully integrating the selected words, dictionary, and score we were able to focus on processing bonus words. We use a DFS to attempt to find most of the words in the word grid while the remaining words are considered to be bonus words.

Loading new game
We tested the ability for users to load their own games by displaying prompts on the GUI and using existing functions in WordGridView. From there we were able to tweak what the GUI should look like and see whether or not the new tiles were being uploaded to a new board.

Saving/ loading saved game

We used many different methods to test saving and loading the same game. We created jUnit tests to test our save and load functions. This lets us check if saved characters were being saved and if a game state exists to be loaded. We do this by checking to see if the grid, dictionary, and the loaded game are null or not. To integrate this function into the rest of our system we added more print statements throughout the game to ensure the sequence of functions being called was consistent with our design to generate the desired output.

Moving forward

We are in the process of adding more jUnit tests to figure out how to successfully use bonus words and load specific levels.