Rohan Mallu

Dr. Wenjin Zhou

COMP. 4610: GUI Programming I

18 December 2024

HW5: Scrabble

For HW5, I implemented one line of Scrabble using HTML, CSS, JavaScript, and jQuery. I tried implementing AJAX to parse the JSON file, but I could not get it to work. Because of this issue, I just loaded my data into a string and parsed it using the JSON.parse(); function from Javascript. Through this assignment, I learned how to use jQuery UI more effectively and it helped me implement complex features of the game such as tracking letters and dragging tiles more easily. Here is the list of features I have implemented, divided into the categories of currently working, partially working, and fully working.

**Currently Working:**

* Seven random letters from the parsed JSON data are selected by the program. The program is able to generate tiles for them on the rack when the page loads. **The JSON object is stored as a string in Javascript and not as a separate .json file.**
* The tiles can be dragged from the rack and dropped onto the board. If they are dropped somewhere else, they will bounce back to the rack.
* The tiles can be dragged off the board and dropped back into the rack.
* The program can identify which letter tile was dropped onto the board by storing the letter in an array, updating the empty word string, and storing it point value in the initial score. When a letter tile is dragged off the board, the program pops the letter from the array, removes the letter from the empty word string, and removes the letter’s point value from the initial score.
* The initial score is tallied correctly for each tile added to the board.
* The board has four squares for double scores. When a tile is dragged onto one of these squares, the tile’s point value is doubled and added to the initial score.
* After the first tile has been placed, all subsequent letters are placed next to the first letter. If the player attempts to place a letter on a non-adjacent board square, the tile bounces back to the rack.
* If there are at least two tiles on the board, the “Submit Word” button is enabled.
* After clicking the “Submit Word” button, the tiles used are cleared from the board and the tile list. New tiles are generated to replace the used tiles in the tile list. The word string is set back to empty. The initial score is added to the total score before being set to zero for the next round. The number of tiles used for the word are subtracted from the tile count. The “Submit Word” button is set back to disabled and stays disabled if there are not at least two tiles on the board. **Tile regeneration does not take into account the amount of each tile available, only the total amount of all tiles (i.e. the user uses up all two F tiles, but the program generates additional F tiles after submitting a word or clicking the “New Tiles” button).**
* The user can keep playing the game until the player decides to quit or depletes all tiles at which point the “Submit Word” and “New Tiles” buttons are disabled.
* The user can click the “New Tiles” button to generate seven new tiles. When the user clicks the “New Tiles” button, the board is cleared if there are any tiles on it, the initial score is set back to zero, the rack is cleared, and new tiles are appended to the rack. **Tile regeneration does not take into account the amount of each tile available, only the total amount of all tiles (i.e. the user uses up all two F tiles, but the program generates additional F tiles after submitting a word or clicking the “New Tiles” button).**
* If the user wishes to restart the game, they can click the “Reset Game” button. This will reset the word string to empty, the initial score and total score to zero, and the total amount of tiles to 100, which will be decremented to 93, as the program clears the rack and generates seven new tiles for the rack.

**Fully Working:**

* The tiles can be dragged from the rack and dropped onto the board. If they are dropped somewhere else, they will bounce back to the rack.
* The tiles can be dragged off the board and dropped back into the rack.
* The program can identify which letter tile was dropped onto the board by storing the letter in an array, updating the empty word string, and storing it point value in the initial score. When a letter tile is dragged off the board, the program pops the letter from the array, removes the letter from the empty word string, and removes the letter’s point value from the initial score.
* The initial score is tallied correctly for each tile added to the board.
* The board has four squares for double scores. When a tile is dragged onto one of these squares, the tile’s point value is doubled and added to the initial score.
* After the first tile has been placed, all subsequent letters are placed next to the first letter. If the player attempts to place a letter on a non-adjacent board square, the tile bounces back to the rack.
* If there are at least two tiles on the board, the “Submit Word” button is enabled.
* The user can keep playing the game until the player decides to quit or depletes all tiles at which point the “Submit Word” and “New Tiles” buttons are disabled.
* If the user wishes to restart the game, they can click the “Reset Game” button. This will reset the word string to empty, the initial score and total score to zero, and the total amount of tiles to 100, which will be decremented to 93, as the program clears the rack and generates seven new tiles for the rack.

**Partially Working:**

* Seven random letters from the parsed JSON data are selected by the program. The program is able to generate tiles for them on the rack when the page loads. **The JSON object is stored as a string in Javascript and not as a separate .json file.**
* After clicking the “Submit Word” button, the tiles used are cleared from the board and the tile list. New tiles are generated to replace the used tiles in the tile list. The word string is set back to empty. The initial score is added to the total score before being set to zero for the next round. The number of tiles used for the word are subtracted from the tile count. The “Submit Word” button is set back to disabled and stays disabled if there are not at least two tiles on the board. **Tile regeneration does not take into account the amount of each tile available, only the total amount of all tiles (i.e. the user uses up all two F tiles, but the program generates additional F tiles after submitting a word or clicking the “New Tiles” button).**
* The user can click the “New Tiles” button to generate seven new tiles. When the user clicks the “New Tiles” button, the board is cleared if there are any tiles on it, the initial score is set back to zero, the rack is cleared, and new tiles are appended to the rack. **Tile regeneration does not take into account the amount of each tile available, only the total amount of all tiles (i.e. the user uses up all two F tiles, but the program generates additional F tiles after submitting a word or clicking the “New Tiles” button).**