Anthony Beaulieu

Professor François Neville

CS 2270

12/6/2023

Hangman Project

This project I started with the barebones HTML template and moved into CSS to section out the webpage to make it more understandable on my end where each output was being sent to.

We built this program off the back of example functions we went over in class these were: btnClick\_onclick(), wrdBlendr(), getVars(), setVars(), printIMG(), and simple HTML forms to catch user input to compare and contrast within our functions.

The main logic of our game comes in the btnGuess\_onclick function, this function is taking each user guess and checking if it belongs to “rite” or “rong” strings. Whatever letter that is passed in the HTML form is then added to the browsers URL to update the webpage, it updates with “rong=&rite=” also has a check for if the rong.length is greater than the picture count in our library we reset the rite and rong strings and also revert the picture back to zero.

Another main function for game logic is the function isValid() this function checks if the user input is within the ascii value of alphabetic lower case characters from a-z, in the function we also lowercase the users guess and assign it to its ascii value to make sure that we aren’t out of our range that we are checking in the isValid function. We also check that the character entered is valid, and if it is not valid we return a javascript alert with a message displaying that what the user entered is wrong.

WordBlendr function is what creates our secret word that the user must guess in order to win at the hangman game. It creates two arrays out of two words, the size of those arrays is based on the length of the words given, those arrays are also filled with asterisks as placeholders for the users guess to be compared to the real word and if that is a match the word is added to the rite string and if wrong its added to the rong string.

printIMG prints out an image with the given index “num”. This function is called in the btnGuess function to reset the game and it is also called in the HTML body to update the picture with the rong.length.

getVar and setVar are our address bar functions that allow us to set the address bar to new variables based on the games inputs, this allows the game to cycle through the correct pictures for each wrong guess, updates the blended word, and ultimately displays if the user has won or lost at the game. We pass addr, pos, key as the parameters in setVar and getVar can display our keys with similar parameters given.

In our HTML body we initialize our strings, our seekrit word, and use getVar to update our web address with our strings that are updated in our btnGuess\_onclick function. We also use the rite and rong variables in the document.write blocks that print to the webpage that the user was right or wrong.

For CSS it was fairly simple I only really changed the position of the hangman box, I did google and find a new unit to use I used 100vh on the height of the box and this adjusts with the viewport that the user has, also we set this display to flex and this changes the position of the box based on the width of the viewport to keep it centered. We used an old windows background for the main background, laid on top an opaque box to specify where the game is sitting, created a nice green button to sort of match the green in the background. Applied some CSS to our textboxes, hangman box, hangman images, and used the global caller to reset at the top.