Hangman Game – PseudoCode

**HTML Elements:**

* Game Header
* Instructions on how to play
* Game Info (wins, losses, # of guesses, letters already guessed”

(Always on Screen)

* Previous Guesses:
* Guesses Used Out of 10:
* Wins:
* Losses:

**Starting Variables/Components**

* Database of words (wordBank array?)
* Var guessesRemaining = 10
* previousGuesses = []
* wins #
* losses#
* Var chosenWord = Randomly chosen words in the wordBank array
* Var wordLength = length of randomWord
* An element that shows an underline for each character in the word chosen from the array wordBank

**Page is loaded, listen for a key to be pressed.**

* Random word is chosen from the wordBank
* For length of randomWord = display an underline and display any random characters (i.e. &)
* Listen for user to press a key

**When a key is pressed**

* Save key press as userGuess (uppercase)
* **Test**: If key is not = to A - Z letters
  + **If true,** Compare the guess to the previousGuesses
    - If match, do nothing (or display message already guessed that)
    - If not a match: call Compare function to compare userGuess to index in the word
  + **If false**
    - do nothing (or display “invalid key message”;

Compare Function:

* Compare userGuess to letter in each index of randomWord
  + If a match: replace the underline in the corresponding position for that letter (for each instance of the letter)
    - Check if there are any letters remaining to guess:
      * If letters remaining, go back to listening for user
      * If no letters remaining, run Game Won function
  + If not a match: decrease guesses by 1 for as long as guesses<10
    - If guesses = 10, run Game Lost

**Game Won Function:**

* Increase value of wins by 1
* Display “User Wins”
* Display method to play again that when clicked picks a new word, but does not reset wins/losses

**Game Lost Function:**

* Increase value of losses by 1
* Display: You Lost,
* Display the correct answer (Either in a message or make it appear in the blanks
* Give a way to restart the game (i.e. button clicked that calls the startGame Function
* previousGuesses=[]

**Random word (Florida Gators) – 14 characters w/ the space, 13 without**

**Underline function**

**- Split the words apart at the spaces (randomWord1, randomWord2,etc;)**

**-For each character in randomWord1, replace the value with “\_” (newrandomWord1)**

**- On the screen show: newRandomWord1 + “ “ newRandomWord2**

**}**

**userChoice = key pressed (string)**

**wordToGuess = the random word that was chosen (string)**

**Need to figure out:**

1. **How to tell the system that a particular character in the string matches the username for each of instance of that character**
2. **For each instance of the match, an action needs to happen in the HTML (replace an underline)**

**wordT**