Project 5 Design Document

**Program Requirements:**

This program will simply display a game of hangman. This game will have between 20 to 100 puzzles/words. These words will be found in a text file. The program will read from that file and randomly choose one of the words to start the game. The underscore character will represent each letter of the word. In that case, if the random word chosen is 5 characters long then 5 underscore characters will display on the screen letting the user know how many letters to guess. Once the user starts to guess, the program will let the user know what letter they have guessed along with the letters left to use to guess the word. If the user guess correctly, then the letter will show up in the correct spot. If not guessed correctly then it will remain an underscore character. Once the user has guessed all the letters to make up the word, the program will tell the user is a winner, but if the user pass their limit of guesses then the program will tell the user they have lost the game and display the correct word.

**Program inputs:**

* The guess variable used for users guesses
  + Char guess
  + Accepted in main function
* File containing the words for the game
  + Infile “puzzles.txt”
  + Accepted in readpuzzles function

**Program Outputs:**

* Main function
  + String alpha
    - Shows how many words available (alphabet)
  + Win/loss string
    - If user wins then “you win, Nice job!” statement will display
    - If user loses the “you lose, The word was :” statement will display
      * Along with the word for that round
* DisplayPuzzle Function
  + Underscore character
    - Used to take place of character while user is guessing
    - **Displayed if** user has **not** guessed the correct letter
  + Correct Guessed letter
    - Used to take position of correct placed letter in word
    - **Displayed if** user has **correctly** guessed the correct letter

**Test plan:**

First, make sure the infile has the correct number of puzzles and each puzzle will be one word. This program should be able to display an accurate game of hangman. There should be a correct number of underscores to represent the number of letters in the word. The user should not have unlimited guess and the program must have the letter display in the correct spot to have the word print out accurately if word is guessed. The display of the game should have spaces between the underscore or each letter. The winner should be displayed if user has won the game, other loser statement should be displayed if user has lost along with the correct word.

Sample runs:

**Case 1**: If user wins

B A S K E T B A L L

Available: CD FGHIJ MNOPQR UVWXYZ

Guess: A B S K E T L

You win. Nice job!

**Case 2**: if user never gets correct guess (loses)

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Available: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Guess: Z J N M P O W F

You lose. The word was: BASKETBALL

**Case 3**: if user loses with some guesses

\_ A S K \_ \_ B A \_ \_

Available: CDEFGHIJ LMNOPQR TUVWXYZ

Guess: A B S K

You lose. The word was: BASKETBALL

**Solution Overview:**

To begin, the proper libraries should be included for C++, these libraries are iostream, fstream, ctime, and ctsdlib.

Since, we have our correct libraries we can continue to create the functions. The usual is the main function; however, we are using four more functions for this program. The first one is called *readPuzzles.* The readPuzzles function returns nothing so it will be a void function and it has two arguments which is the string array of words and it is has the integer wordcount as a reference in the parameters. The next function is called *displayPuzzle* that also returns nothing with two arguments. Therefore, this function will also be a void. The arguments of this function is a string called word and the array of bool called guessed. The third function is called *found* with a bool since it will be returning a bool and it has three arguments the string word, array of bool guessed, and character guess, Finally, the last function is called *winner* that has bool return type so it will be bool function with two arguments which is the array of bool guessed, the integer type called letters that represents the number of letters in the puzzle.

Now that we know our functions name, type, and parameter, we can begin to code our program. The main function contains constants, so we will have constant integers representing the maximum of words, maximum of word size, the maximum of guesses from the user, and the number of letters in each word.

The main function begins by defining string alpha equal to the alphabet. It continues to define other variables such as the string of words and the integers of wordnum, letters, misses, numwords all set to 0. Along with the guess variable as a with the bool guessed as the max word size.

First the main function calls the readPuzzles function. This function opens the puzzle file and the fstream library is used to read from the file into an array of words variable. This function will also be using the wordcount variable to keep up with how many puzzles were read in the file. The main functions then continue by randomizing the words and setting the array to false for the underscores. Next, it calls the displayPuzzle function which will have a loop to go through each of the letters from the word while looking at the corresponding position of the guessed array. This function will then display the letter if it is guessed or display an underscore if not guessed. The main function will then show available letters and the users guesses. The found function is called in the if statement of main and is returning a true bool if letter is found and if not then it should be returning a false. This function will also change the guessed array to true at each position. The next function called in main is the winner function which is in a while statement of main. This function will determine a winner or not. This will be done by having the function have a loop that goes through each element. If all elements are true, then the function will return true and the user is a winner. If one false is found the user has lost and the function will return a false. Finally, the main function starts to end by letting the using cout statements to let the user know if they won or loss.

Lastly, the program will end with a return value as zero and close main function.

**Algorithm Flowchart:**

Diagram, schematic

Description automatically generated