**Hangman Research and Analysis**

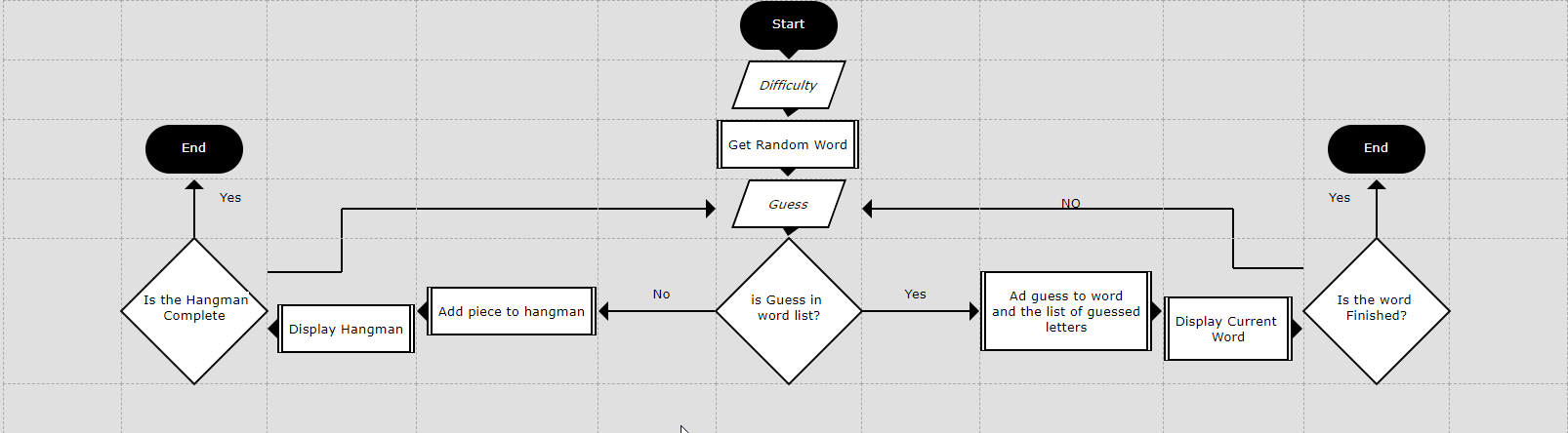
**Interview summary**

The completed program should be able to play basis hangman with one user and one computer, guessing longer type words or more multiple words in a sort of phrase would be more preferred, all of these must be common words and be in the English language, there would be no numbers only stuff that’s included in the English alphabet this also means no punctuation. The normal mode that is defaulted on the game must have 12 lives and there must also be other options to change the difficulty increasing or decreasing the amount of lives, and increasing or decreasing the difficulty of certain words or phrases, perhaps a different word list interly. The use should be able to guess the entire word at any time throughout the game and if its wrong it continues building the “hangman” and decreases the lives, like guessing a normal character would. There should be a displayed list of previously guessed characters and words maybe. The user should guess what letter by using a set of clickable buttons, and those buttons should no longer be clickable when clicked initially. The design of the program must be easy to understand and navigate and the program should support working on computers.

**Hangman Build Summary**

A phrase or word is chosen from a predefined wordlist entered by the user. The word is then hidden on the screen, this is done so the user cannot see what letters are contained in the word they can just see how long it is. In total they have 12 guesses, every time they do not select a letter that is contained inside the hidden word, they lose a life and the hangman builds until its complete where they loose and the game ends. If they guess all the letters correctly without loosing all of there lives they win the game.

**Flow Chart**

****

**Input, Process, Storage, Output Chart:**

|  |  |
| --- | --- |
| **Input**  Play Game  See Leaderboard  Letter Guess  Word Guess | **Process**  Is Guess right or wrong?  Is Hangman or Word complete? |
| **Storage**  High Score  Previous Games | **Output**  You lose/You Win  Gallows  Word  Guessed Letter |

**Data In Lists**

**Word List**

|  |  |
| --- | --- |
| Number of Words | 854 |
| Longest Word | Distribution |
| Shortest Word | So |
| Average Word Length | 5 |
| Range of Characters Used | English lexicon |

**Movie List**

|  |  |
| --- | --- |
| Number of Words | 246 |
| Longest Word | Dr Strangeglove Or How I Learned To Stop Worrying And Love The Bomb |
| Shortest Word | M |
| Average Word Length | 10 |
| Range of Characters Used | English Alphabet + Numbers |

**Summary and Discussion**

I personally think there needs to become change in how long the maximum and minimum length of word can be , especially in the movie list. The length of the different names of each movie varies hugely with the longest movie being 55 characters long while the shortest movie being only one character long, I also believe that regular words should be limited to around 4-8 characters, n my research and analysis I have discovered that some games could go on for a extended amount of time if played in a certain way, there is no way to avoid it as its just the nature of the hangman style but we should make the experience as easy to do as possible by giving it a nice design style and layout. To fulfil this we could use lots of different colours and certain colours which attract users to certain actions or buttons such as vowels, it would also be nice to implement a sort of menu where the user could view the current word list/ movie list/ and set the difficulty for the hangman game as well as view the lead board.

**Full Requirements and order**

1. Get a word to guess

1.1 List of words must be stored in file

* + - 1. File must be accessed at the load of the game without asking the user
      2. If the file is unavailable, an error message is displayed to the user followed by an open file dialog box
  1. A random word is selected at the start of the game from the list
  2. Words to guess should not be repeated in a particular “session” of games
  3. For testing purposes, there should be a button that will toggle the display of the chosen word

1. Input a letter to guess.
   1. Each available letter is shown on an individual button
      1. Once the letter has been guessed it will be greyed out, and no longer functions as a button
   2. When guessed the screen will display a message saying whether it is in the word or not.
      1. If not in the word, it will be added to a list of “Guessed Letter”
      2. If it is in the word, it will be added to its place in the chosen word
   3. If the guess is wrong, a new piece will be added to the gallows
2. Check to see if the Word is completed and End Game.
   1. There will be two tests needed to accurately detect when or if the game has finished
   2. The first one will be detecting when the hangman character has been completed
      1. If the lives of the hangman = 12
      2. The game will output a game over message.
   3. The other will be detecting when the word is complete
      1. If all the letters in the word have been guessed, or if the entered word has been guessed correctly.
      2. The game will display a winning message and the game will finish
   4. A record of the finished game will be sent to the Leader board file.
3. Add piece to the hangman.
   1. If the last guessed letter is not in the word or the guessed word is wrong, add a piece to the hangman
      1. Test the hangman to see how much there is already
      2. Add the next item in the list to the image just +1 , and display it
   2. If the Hangman’s all 12 pieces are complete, then game is over and display a lost message
4. Check for Previously guessed letters
   1. When the user guesses a new letter, add it to a list of guessed letters.
   2. If the letter is already in the list of guessed letters, don’t allow it to be selected again. Perhaps gray the letters out when selected once.
5. Outputting the win or lose screen
   1. Test for when the game is over
   2. Test for whether the game has been won by the user or lost
      1. If won, output the win screen and save stats to file
      2. If lost, output the lose screen and save stats to file
6. Displaying high scores
   1. If the user selects High Scores or Leaderboard, display previous game statistics for that user that have been saved
      1. Open the stats file and read each line
      2. Display and format each line in turn
      3. Allow the user to scroll through all the previous games