Hangman

**Basic Description (first understanding)**

The program will start by randomly choosing a word from a list of 40 words.

The user will then choose how many words they want to play.

It will display the length of the word on screen.

The user will input singular letters to try and guess what it is.

If they get a letter right, the word will be displayed with said letter included.

If the same letter appears twice in the word, it only needs to be guessed once.

If they guess a letter not in the word,

The incorrect letter will also be displayed.

The word and correct letters so far will be displayed too.

They will have 10 guesses. This goes down after each guess. An incorrect guess results in +0 and correct results +1.

If they guess every letter in the word they win and progress to the next word.

If they run out of guesses, the game will end.

**Variables**

*Turns -* The number of guesses you have for each word – Integer

*Guesses -* An empty variable - String

*GameLength -* How many words the user choses to play (Max of 40 - Min of 1) - Integer

*GamePlayed* - A counter for how many games have been played - Integer

*Word -* The current word trying to be guessed - string

*Letter -* The letter guessed by the user - string

**Constants**:

*WORDLIST* - The 40 word list of words - list

**Modules:**

*WORDGUESS* - The module that allows for guessing of letters and such – function

**Imports:**

*Random* - Allows for a random word to be chosen from the wordlist using *random.choice()*

*Time* - used to add a delay between displays to make the game run smoother

**Pseudo code**

Set *WORDLIST* to the list of all 40 words

Set *guesses* to 10

Set *gameLength* to 0

Ask user for number of games they want to play. Set this to *gameLength*

Set *word* to a random word from *WORDLIST*

Display *word* length in underscores (with spaces in between)

Ask user to input a *letter*

Create wordGuess module

If *letter* is present in *word*

Display the word with the guessed letter in its correct space/s

Display incorrect guessed letters

Display remaining *guesses*

Else *letter* not in *word*

Display the word with the current correct guessed letters

Display incorrect guessed letters

Take away one *guesses*

Display remaining *guesses*

If *guesses* is equal to or less than 0

Break from loop

Elif word is completed

Break from loop

Else *guesses* is greater than 0

Repeat loop

If *guesses* is equal to or less than 0

Display full hangman ASCII

Game over

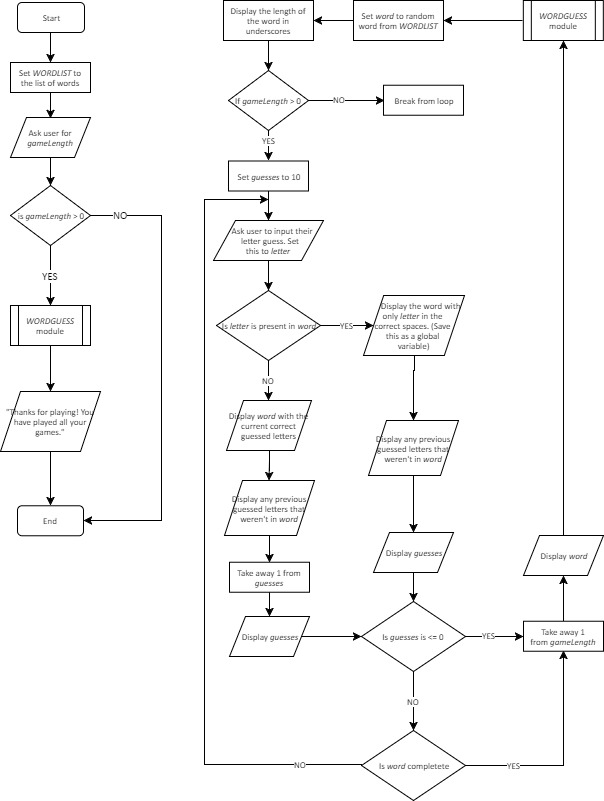
Display full word

Else

Display full word

Display congratulations

repeat

**Flowchart**