|  |
| --- |
| Word (Class) |
| -string answer  -string visual  -vector<char> badGuess  -int misses  -int wiggle  -int spazz |
| +Word(string)  +void playWord |

**Explaining the Attributes/Functions of Class:**

string answer: the word that the user is attempting to guess.

string visual: the underscores of the letters trying to be guessed. The underscores will be changed to the correctly guess letter using a “for loop”.

Vector<char> badGuess: displaying the line of “incorrect guesses” made by the user.

int misses: the number of missed guesses the user has

int wiggle: the difficulty of getting out of the noose (ASCII Art display)

int spazz: in a while loop. The user hits the number 1 to not have the person die from the noose. In the while loop, the amount of times the person hits the 1 key is divided by 2, and if there is no remainder, the head of the hanging person is displayed on the right side. If there is a remainder, the head will display on the left side.

Word(string): Constructor for the class.

void playWord: Picks out word using file input/output from the designated set of words (5-8 letter words).

**Requirements:**

Create a “Hangman” game using classes and file input/output to randomize word selection, and display cleaner code in the C++ Visual Studio program.

We will need to create at least one class to organize our variables and functions since we will need certain functions to get specific words from a certain list, display the amount of letters as underscores, and take in the user input and loop through to identify correctly guessed letters.

The user will have many different ways to interact with the system. Not only will they have to guess the randomly generated word, but they will interact with a person being hung and try to save him by repeatedly hitting the 1 key.

Our goal is to keep the user engaged through the entire game from the moment they select their word length in the menu to the end of their gameplay.