**Program Description**

Our group consisted of Seth Ingham and Andy Nguyencuu. We worked on it together using Atom so that we could work on it simultaneously and collaborate. However, it did not work as well as we wanted, and we ended up working on it on the same computer. We worked on most of the classes together and divided it up on a method basis rather than by class. Rather than divide up the work, we sat together and whenever one of us would have an idea on how to do something we would write it. We have the Art class which was the abstract base class that handled the processing of the ASCII art. It held basic information such as height and file path as well as getters, so that the derived classes have that in store. We also had the Hang class. This was the class that handled all the file inputs as well as displaying the gallows. It is derived from the art class, and it has its own constructor, draw function and destructor. The main function runs in a continuous while loop - displaying the main menu each time to allow the player to either start a new game, load a saved game, or quit on each iteration. On each game start, it creates new objects for the art and the word, initialized by either the default values reset on every play through, or by the values loaded if the player chooses. For every clear and printing on the screen, it calls the draw functions of the dynamically created objects, using conditionals to print out stuff based on context. The word class was where we handled displaying the word as it was guessed, as well as the marker showing where the current guessed letter is. It is also where the word is randomly selected and tested to meet the difficulty selected. The word is chosen using a random number generator and is then pulled from the array at that position and is checked for length. If it is too short or too long another random number is generated. It is also where the guess is processed to see if it is in the word and determines whether the user was right or wrong and acts accordingly. It also constructs the underscores for the display so the user can see the length of the word. It also houses the vector that checks for already guessed letters.

Program instructions:

Run Hangman.exe (or compile it)

Follow the instructions on the screen!

Profit.