CS224 – Project 0: Writing a Jack Application

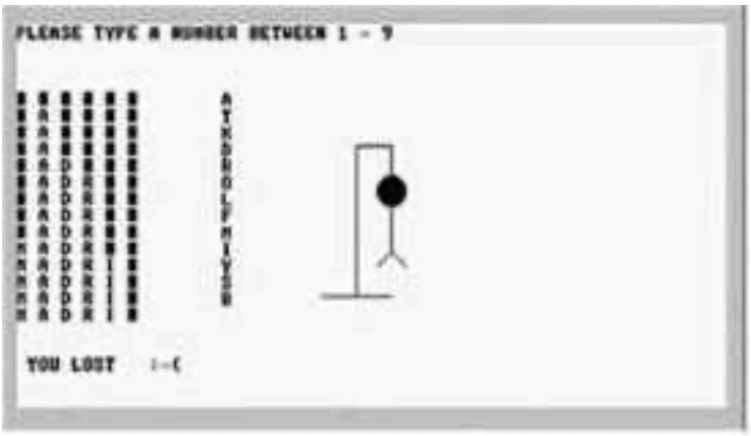
Background

Jack is an object oriented programming language like Java, although its syntax is a bit more "clunky". It is a weakly typed language, meaning that types conformity in assignment statements is not strictly enforced. Unlike Java, you must use keywords like `do` and `let` for method calls and assignment statements. Moreover, curly brackets must be used even in single statement blocks. Also beware that the language does not enforce operator precedence in expressions. (You will appreciate all these changes when you write the compiler for Jack.)

Jack makes use of a standard library package which can be viewed as an interface between the language and a simple OS.

Objective

You will get acquainted with the Jack language for the purpose of writing a Jack compiler in future projects. You will write a Hangman game in Jack. A screenshot of the program could look like:



I made a couple of choices in my solution to keep things simple. First, I hard coded a single word for the game. Secondly, I printed out underscores for unguessed letters.

Criteria for Success

When compiled with the supplied JackCompiler and loaded into the supplied VMEmulator your code may be executed to play a game of Hangman. The game should print the word with the appropriate underscores as well as the entered letter at each guess. Additionally, the game draws the hangman graphics for missed guesses. The game ends when either the word is completed or the entire hangman stick figure is drawn.

Resources

The relevant reading for this project is Chapter 9 in the nand2tetris text. I have also supplied a zip file contain a code skeleton for Hangman, the Main program file, and all the relevant standard library files.

Tools

The JackCompiler and VMEmulator are available on our phoenix server

To compile your Hangman program, open up a Terminal in the directory containing your Hangman directory and type `JackCompiler Hangman`. Once your program compiles without errors, launch the VMEmulator. In that emulator use the load button to load the **entire** Hangman directory. Select no animation and hit the run button. Play your game of Hangman.

Submission and Assessment

Submit in Canvas your entire Hangman directory as a single ZIP archive. Your project will be graded using the rubric provided in the Canvas submission.