

# Hangman

Adam Wu

2023 December 20

## 1 Purpose

The aim of this project is to practice with string arrays in C++. The best way to do that is to re-create the hangman game in the terminal!

## 2 How to Use the Program

First clone the repository to get all the source files. The next step is to go to folder where this project is on your local machine and go to the *src* folder. To run hangman you will need to compile the program first, then you can run it. In the next section, you will find out how to compile, run, and removal of the program.

### 2.1 Compiling the program

You will just need to type `[make]` and the program will start compiling. If you have never programmed before, a demonstration of this program compiling will be shown below.

```
$ make
c++ -c -o hangman.o hangman.cpp
c++ -c -o hangman_helpers.o hangman_helpers.cpp
g++ -std=c++17 -Wall -g -o hangman hangman.o hangman_helpers.o
$
```

after successfully compiling, try and type `ls`. When you do, you will notice a new file with a different color have emerged. That is the program executable. With that, we can move on to learning how to run the program.

### 2.2 Running the program

Now to run the program, you will need to do the following:

```
$ ./hangman <secret word or phrase>
```

For this program, the allowed characters are lower cased characters and punctuation like a space, an apostrophe, and a dash. Other punctuation and capital letters will be unhallowed. From there on, the game should be very intuitive.

### 2.3 Removal of the program

To remove the executable, you will need to type *make clean*. Soon after, you can type *ls* to check that your folder does not contain the colored executable anymore.

---

## 3 Program Design

This program only consists of 3 files. `hangman.cpp`, `hangman_helpers.cpp`, and `hangman_helpers.h`. The `hangman` helper file help with clearing the screen, the ascii art of the hangman, and validating the string array. The main `hangman` file then utilizes the helper function for the main game.

### 3.1 Data Structures

### Algorithms

### 3.2 Function Descriptions

## 4 Results

### 4.1 Numeric results

### 4.2 Error Handling

## References