**Hangman Game**

Table of Contents

[1. DESCRIPTION 3](#_Toc51913290)

[2. OVERVIEW 3](#_Toc51913291)

[3. REQUIREMENTS 4](#_Toc51913292)

[3.1 Hangman Functional Requirements: 4](#_Toc51913293)

[3.2 Hangman Non-Functional Requirements 6](#_Toc51913294)

[4. TEST PLAN AND TEST CASES 7](#_Toc51913295)

[5. EXPECTED OUTPUT 8](#_Toc51913296)

[6. CONCLUSION 9](#_Toc51913297)

[7. FUTURE SCOPE 9](#_Toc51913298)

[8. REFERENCE 10](#_Toc51913299)

# 

# 1. DESCRIPTION

Hangman is a popular word game in which one player (the "chooser") chooses a secret word and another player (the "guesser") attempts to guess the word one letter at a time. If a guessed letter appears in the word, all instances of it are revealed. If not, the guesser loses a chance. If the guesser figures out the secret word before he or she runs out of chances, he or she wins. If not, the player who chose the word wins. Traditionally, chances are tracked using a stick figure drawing of a person being hanged from a gallows. The figure is drawn one body part at a time, and the guesser loses when the entire figure has been drawn. This game is also the basis for the TV game show Wheel of Fortune.

# 2. OVERVIEW

Game play

In our implementation of Hangman, the computer will take on the role of the "chooser" and the human player will be the "guesser." The computer will secretly choose a word from a list (see below) and show the player how many letters are in the word by displaying a sequence of blanks (underscores). Then, the computer will begin asking for guesses. If the player guesses a letter that is in the secret word, all blanks representing an instance of that letter should be replaced by the letter. If the guessed letter is not in the word at all, the player should lose a chance and a new part of the Hangman figure should appear. If the player guesses a letter he or she has already guessed, he or she should not lose a chance, even if that letter is not in the word. If the player guesses all letters in the word, he or she wins. If the Hangman figure is completed, the player loses. In either case, the secret word should be revealed after the game is over.

Word Status

As the game is played, the player should be shown the current guessed status of the secret word. Letters that have been correctly guessed should be shown in the correct locations. Unguessed letters will appear as blanks. At the beginning of the game, no letters will have been guessed, and the only information shown to the player will be a sequence of blanks, with one blank for each letter in the secret word. As the player guesses letters correctly, blanks representing guessed letters should be replaced by those letters.

So, for example, if the secret word is "screwdriver" and the player has guessed 'e,' 's', 'r', and 'd,' the current word status would be "s *r e*d r e r".

Chances/The Hangman

The player will have six "chances" to guess the word. Guessing a correct letter does not cost a chance. Each missed chance will cause a new piece of the Hangman to appear. The six pieces of the Hangman are: head, body, left arm, right arm, left leg, right leg. You may use a stick figure for your Hangman, but if you would like to be more creative with the appearance, feel free to do so. No matter what your hangman looks like, though, it should include these six pieces and no more.

Game End

The game can end in one of two ways:

If the player has guessed the complete secret word, he or she wins.

Otherwise, if the player has run out of chances and the complete Hangman has been drawn, the player loses.

In either case, when the game ends the host should stop asking for guesses. The host should inform the player whether he or she won or lost, and the assistant should reveal the entire secret word.

# 3. REQUIREMENTS

## Hangman Functional Requirements:

1. Starting the Application

Allow the user to enter on the command line the IP address of a remote computer that is running a Hangman Word Server.  If the user doesn't provide an IP address, use as a default the address "local host."

At the start of each game the application will request a word from the Hangman Word Server specified by IP address.

The word obtained from the Word Server will be designated as the "hidden" word the player tries to guess.

If no connection can be made to a Word Server on the given IP address display an error message on the console "Apparently no hangman word server is running at <IP address>."

If a word is obtained from the Word Server, game play begins by offering the player their first turn.

B. Playing a game

At each turn the application will display a visual indicator of how many letters are in the hidden word and if any of the letters have been correctly guessed they are shown in the proper position in which they appear in the word.

The application will display a "guess count" which shows how many incorrect guesses the player has made.  An incorrect guess is guessing a letter which is not in the hidden word.

The application will allow the player to enter a letter.

If the letter entered is not between A and Z display a message "Invalid move" and allow the player to guess again (without penalty).

When the player enters a valid letter the application will check to see if the game is over and if not will continue to the next turn.

C. Ending a game

The player wins by correctly guessing all the letters in the hidden word.

The player loses if he/she makes seven incorrect guesses.

If the player wins the application will display a congratulatory message.

If the player loses the application will display a consolation message and will reveal the hidden word.

When the game is over (either win or loss) the application will offer the player an opportunity to begin another game.

If the player indicates they want to play again, a new game is started.

If the player indicates they do not want to play again, the application is terminated.

Note: there is no way for a player to request termination of play during the middle of a game.

## 3.2 Hangman Non-Functional Requirements

General Guidelines: Performance and reliability are not very important.  Priority  should be given to adaptability, maintainability, and usability.

Modifiability

1. If it is desired to change the number of turns in a game, the developer will be able to make the required changes in < 1 person-hours.

Adaptability

1. The user must be able to specify an alternate user interface at execution time on the command line.
2. Any alternate user interface must be able to "plug in" at run time without recompiling the application.

Reliability

1. Since the program is purely for recreation and involves no user data, reliability is of low importance.

Security

1. The program will not access any user data files or programs.
2. The program will not alter or replace any system files.

Usability

1. A new user should be able to play a complete game of hangman in less than ten minutes.
2. A new user should commit less than one error in use of the game (e.g. selecting the wrong letter) every ten minutes.
3. A user who is familiar with the rules of Hangman be able to correctly operate the program without any written documentation.

Testability

1. (optional) If the user provides an IP address of zero then skip getting a word from the Word Server and use a default word "calpoly" as the hidden word.

Performance

1. Desired Response Times (not critical) : At game start: less than five seconds

Maintainability

1. It is desirable that the game administrator be able to modify the words data file using a simple text editor.
2. Adding a new word to the word data file should take less than ten minutes.

# 4. TEST PLAN AND TEST CASES

1. To check command line arguments are handle correctly.

2. To check whether word are loading properly based on the guess of user.

3. Test to check whether the user is prompted to guess next letter after each guess.

4. To check the guess that the user made are handled correctly.

5. To check all correctly guessed and incorrectly guessed are letters are displayed or not.

6. To check proper built of hangman for every incorrect guess.

7. To check whether there is congratulatory or consoling message displayed at the end of game based on win or loss.

# 

# 5. EXPECTED OUTPUT

The following example game illustrates a player trying to guess the word *hangman* using a strategy based solely on letter frequency. As the player continues, a part of the stick figure on the noose is added. Once a full body is drawn, the game is over, and the player lost.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | [Hangman-0.png](https://en.wikipedia.org/wiki/File:Hangman-0.png)   |  |  | | --- | --- | | Word: | hangman | | Guess: | E | | Misses: |  | |
| 1 | [Hangman-1.png](https://en.wikipedia.org/wiki/File:Hangman-1.png)   |  |  | | --- | --- | | Word: | \_ \_ \_ \_ \_ \_ \_ | | Guess: | T | | Misses: | e | |
| 2 | [Hangman-2.png](https://en.wikipedia.org/wiki/File:Hangman-2.png)   |  |  | | --- | --- | | Word: \_ \_ | \_ \_ \_ \_ \_ \_ | | Guess: | A | | Misses: | e, t | |
| 3 | [Hangman-2.png](https://en.wikipedia.org/wiki/File:Hangman-2.png)   |  |  | | --- | --- | | Word: | \_ A \_ \_ \_ A \_ | | Guess: | O | | Misses: | e, t | |
| 4 | [Hangman-3.png](https://en.wikipedia.org/wiki/File:Hangman-3.png)   |  |  | | --- | --- | | Word: | \_ A \_ \_ \_ A \_ | | Guess: | I | | Misses: | e, o, t | |
| 5 | [Hangman-4.png](https://en.wikipedia.org/wiki/File:Hangman-4.png)   |  |  | | --- | --- | | Word: | \_ A \_ \_ \_ A \_ | | Guess: | S | | Misses: | e, i, o, t | |
| 6 | [Hangman-5.png](https://en.wikipedia.org/wiki/File:Hangman-5.png)   |  |  | | --- | --- | | Word: | \_ A \_ \_ \_ A \_ | | Guess: | N | | Misses: | e, i, o, s, t | |
| 7 | [Hangman-5.png](https://en.wikipedia.org/wiki/File:Hangman-5.png)   |  |  | | --- | --- | | Word: | \_ A N \_ \_ A N | | Guess: | R | | Misses: | e ,i, o, s, t | |
| 8 | [Hangman-6.png](https://en.wikipedia.org/wiki/File:Hangman-6.png)   |  |  | | --- | --- | | Word: | \_ A N \_ \_ A N | | Guess: |  | | Misses: | e, i, o, r, s, t | |
| The guessing player has lost this game as the diagram had been completed before all the letters were guessed. | | |

# 6. CONCLUSION

The purpose of the program is to have some fun at work when you’ve finished all of your projects and you do not have internet access or games! There are no calculations in the code, but it did require a lot of loops

# 7. FUTURE SCOPE

* Level extension.
* Introducing a new game features.

# 8. REFERENCE

The reference I took for hang man to make the project more understandable for human language are given below

* <https://en.wikipedia.org/wiki/Hangman_(game)>
* [www.youtube.com](http://www.youtube.com)
* <https://teals-introcs.gitbooks.io/introduction-to-computer-science-principles/content/project_4.html>