**Requirements**

* Operating systems (any)
* GCC Compiler
* VS Code
* Make

**Introduction**

* This is a text-based version of Hangman Game written in C.
* The program chooses a random word from the given word list in program using rand() function. It will prompt the user to guess a character.
* If right the word will be displayed with the guessed character revealed.
* If wrong the program will tell you how many guesses you have left and print the current state of the stick man which will progress with each wrong guess.
* The user is allowed 10 wrong guesses before the game ends.

**4W's and 1'H**

**Who:**

* Working all day can be monotonous therefore anybody can play this game for recreation.
* Can also prove beneficial for improving English Vocabulary.

**What:**

* This is a game which reads a random word from the wordlist, and asks the user the guess it correctly so that the User win the game otherwise the User looses after a specific number of trials.

**When:**

* It's a great game for recreation especially while you fee; your work routine is getting monotonous.

**Where:**

#### This game can be played anywhere independent of the platforms and can be act as a tool for improving vocabulary.

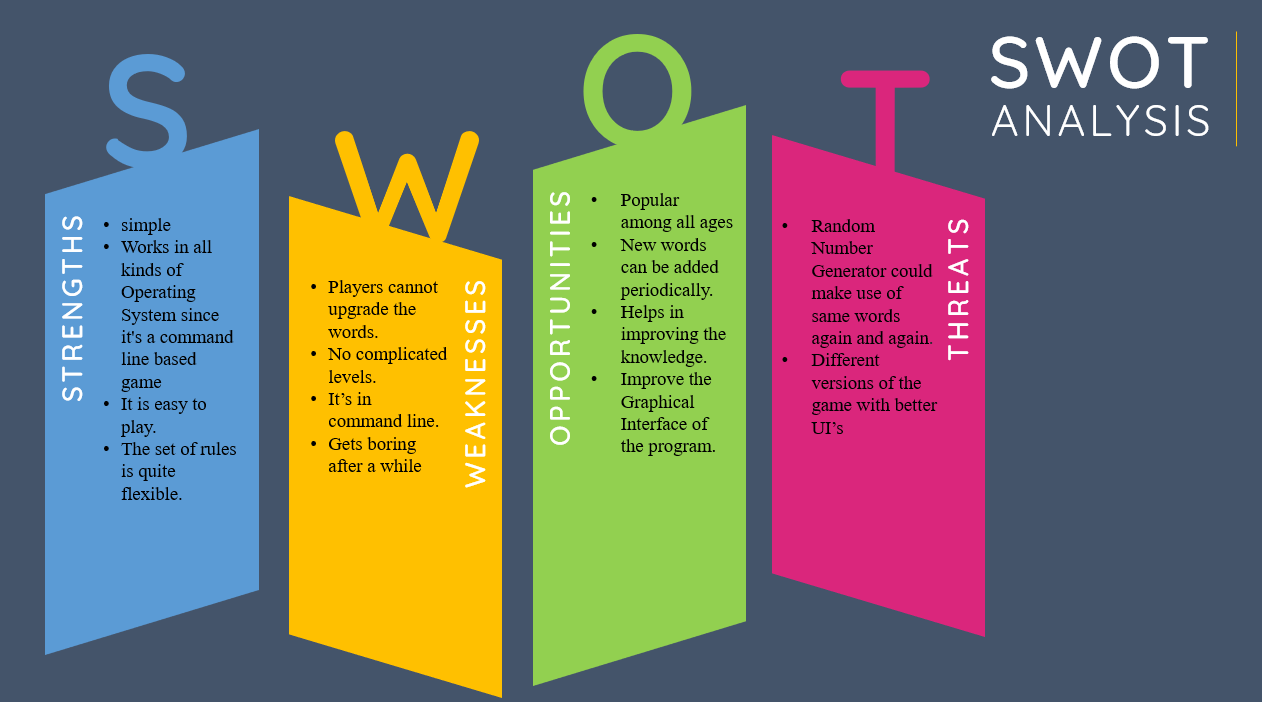
**How:**

#### Some words are stored in a dictionary. A random word is taken from dictionary.

#### The player gives the input as an alphabet to guess the word. The player gets 7 chances to guess the correct word.

#### For every incorrect letter the sketch of hangman begins. Once the formation is done the player will lose the game otherwise, he wins.

**SWOT Analysis**



**Detail requirements: - High Level Requirements:**

|  |  |  |
| --- | --- | --- |
| ID | Description | Status |
| HR01 | Able to access the word from file for guessing | IMPLEMENTED |
| HR02 | Proper flow while checking a letter and proceeding for next steps | IMPLEMENTED |
| HR03 | Game should end with accurate results | IMPLEMENTED |

**Low level Requirements:**

|  |  |  |
| --- | --- | --- |
| ID | Description | Status |
| LR01 | Able to access the word from file | IMPLEMENTED |
| LR02 | Able to store the incomplete word with blank spaces after every guess | IMPLEMENTED |
| LR03 | Able to check whether a letter from alphabet is present or not | IMPLEMENTED |
| LR04 | Able to specify user has won if the word is guessed within 7-10 incorrect guesses | IMPLEMENTED |