

**COS10009**

**INTRODUCTION TO PROGRAMMING**

**Lecturer:** Jason HsienMing Yong

**Design Report for**

**Hangman**

**By:** Osama Iftikhar Warritich (102763363)

Contents

[**Custom Program** 1](#_Toc89651425)

[**Description** 1](#_Toc89651426)

[**Objectives** 1](#_Toc89651427)

[**Pros and Cons of the Program** 1](#_Toc89651428)

[**Pros** 1](#_Toc89651429)

[**Cons** 1](#_Toc89651430)

[**Flowchart** 2](#_Toc89651431)

[**Data Dictionary** 3](#_Toc89651432)

[**Main Functions** 4](#_Toc89651433)

[**Screenshots** 5](#_Toc89651434)

# **Custom Program**

* Hangman Game

# **Description**

This is a simple Hangman game using Ruby programming language Hangman program randomly selects a secret word from a list of secret words. The random module will provide this ability, so line 1 in the program imports it. Hangman is a popular word guessing game where the player attempts to build a missing word by guessing one letter at a time. After several incorrect guesses, the game ends, and the player loses. The game also ends if the player correctly identifies all the letters of the missing word. If a guessed letter appears in the word, all instances of it are revealed. If not, the guesser loses a chance. They win if the guesser figures out the secret word before they run out of chances.

# **Objectives**

* For entertaining Oneself
* No Internet connection is required
* Eradicates boredom

# **Pros and Cons of the Program**

## **Pros**

* The Program is very simple and easy to understand.
* Settings can be changed easily at anytime.

## **Cons**

* No GUI causing it to look less appealing.
* Only 1 player can play at a time.

# **Flowchart**

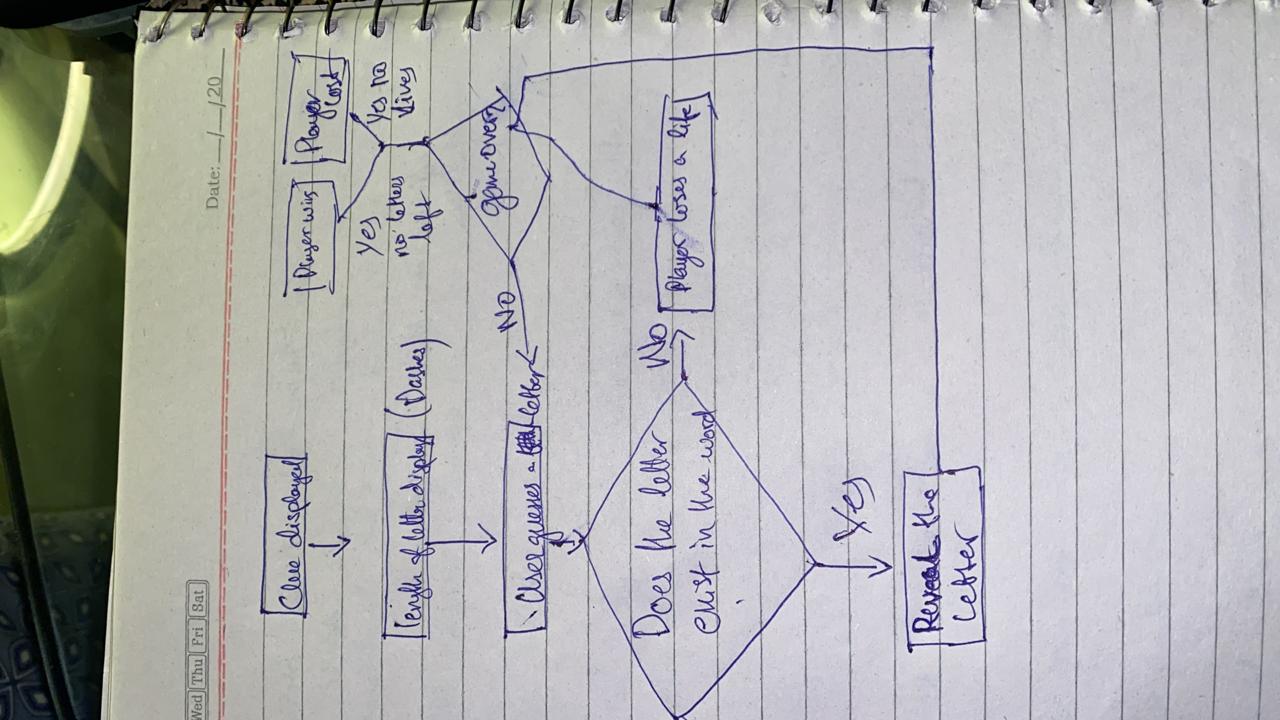


Figure 1: Flowchart of Hangman

# **Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Meaning** | **Example** |
| words | String | The words in the array that will be Guessed | wedding |
| Lives | Integer | Number of Chances to guess | 7 |
| Guess | String | The Letter Guessed | A |

Table 1: Data Dictionary

# **Main Functions**

|  |  |
| --- | --- |
| **Function** | **Meaning** |
| Words | This function stores the words that the Player will guess along with their respective clues. |
| initialize | This function assigns variables and sets the Letters size. |
| print\_teaser | This runs the update\_teaser and checks where the letter is placed and leaves the rest of the spaces blank. |
| update\_teaser | This function splits the word so each letter can be verified along with its position in the word, |
| make\_guess | This function makes sure that the if function keeps running until the lives are not over. |
| Main | This function runs the main function where the print\_teaser and make\_guess are called. |

Table 2: Main functions

# **Screenshots**

Text

Description automatically generated