



: GRIFFITH COLLEGE DUBLIN

Student name:	Evandro Gomez Quintino		
Student number:			
Faculty:	Computing Science		
Course:	BSc / HCC / Other	Stage/year:	1
Subject:	Computer Programming		
Study Mode:	Full time	Part-time	X
Lecturer Name:	Eoin Carroll		
Assignment Title:			
No. of pages:			
Disk included?	Yes / No		
Uploaded to Moodle?	Yes / No		
Additional Information:	(ie. number of pieces submitted, size of assignment, A2, A3 etc)		
Date due:	11/05/2018		
Date submitted:	11/05/2018		

Plagiarism disclaimer:

I understand that plagiarism is a serious offence and have read and understood the college policy on plagiarism. I also understand that I may receive a mark of zero if I have not identified and properly attributed sources which have been used, referred to, or have in any way influenced the preparation of this assignment, or if I have knowingly allowed others to plagiarise my work in this way.

I hereby certify that this assignment is my own work, based on my personal study and/or research, and that I have acknowledged all material and sources used in its preparation. I also certify that the assignment has not previously been submitted for assessment and that I have not copied in part or whole or otherwise plagiarised the work of anyone else, including other students.

Signed:

Date: 11/05/2018

Please note: Students MUST retain a hard / soft copy of ALL assignments

Design Document - Hangman

by Evandro Gomez Quintino



Contents

Introduction.....	4
Step by Step	4
The idea.....	4
Design	6

Introduction

The proposal of this document is to present in a short and clear way how I managed to code my game "Hangman".

The game itself could be coded in different ways but for academic reasons I tried to use only the statements given in class: Looping, variables, if, methods and files reading/creating. The only statements that I have used different from this rule was Global Variables and ArrayLists.

I hope that you enjoy the reading and thank you for playing my game.

Step by Step

The idea

As you can see in the image below, everything started with myself questioning how I would manage to select one of the words in the text files (divided into difficulties easy, medium and hard.txt); how I would storage this variable; convert the variable's length into dashes; compare the player's guess with the variable and finally printing the results. The image below isn't clear, but this was the first draft that I decided to draw to start my logic to solve the problem.

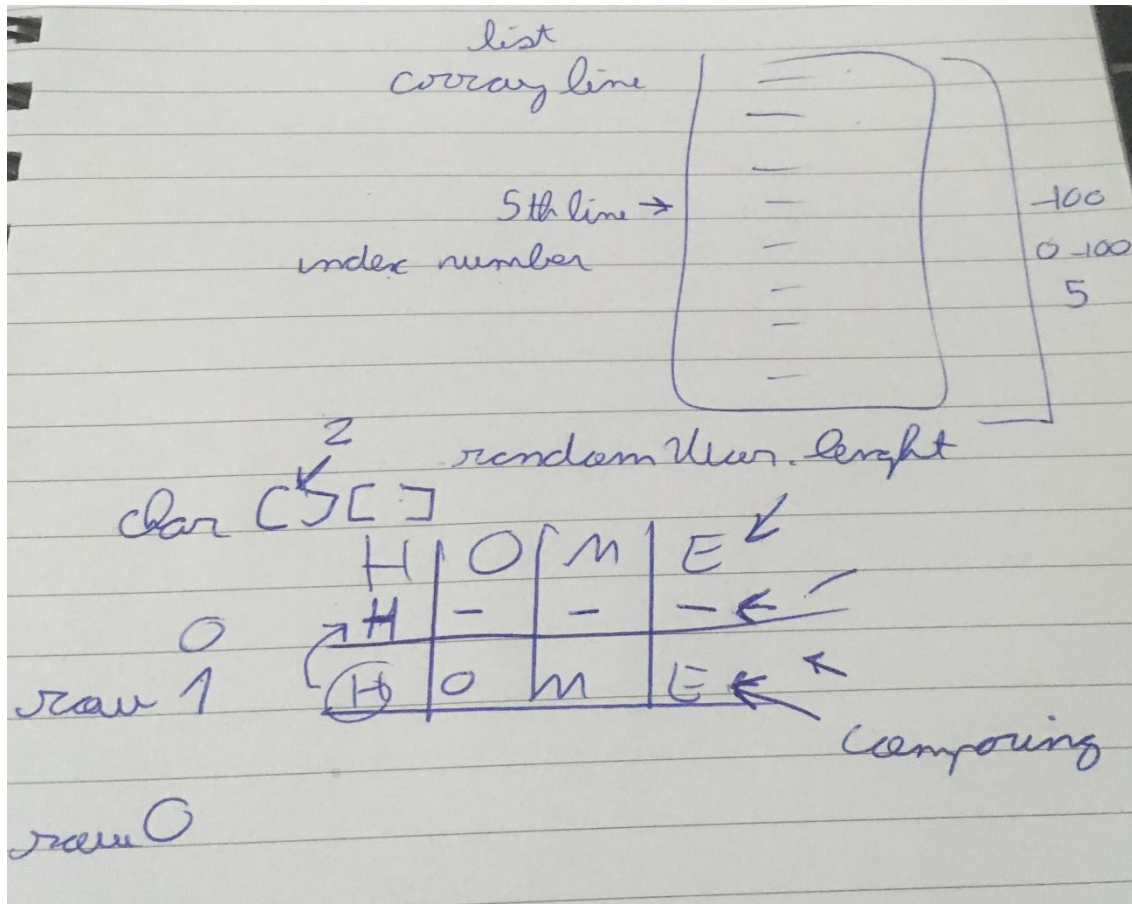


Figure 1 "Everything started simply with: pen and a piece of paper".

Basically, my solution for this problem was using something that I hadn't used so far: Two Dimensional Arrays. Using the image below I can explain it better:

0	1	2	3	The index position here is fundamental for my idea and the reason because I believe that a Two Dimensional Array suits perfectly my necessities
H	O	M	E	The second row I drew the word that my program was storing
-	-	-	-	The third row was converting the word into dashes
H	O	M	E	The last row was representing the players input (or guesses)

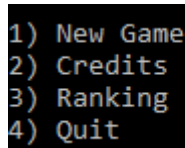
2D Array = char [][] dashWord – Line181

After this logic created, was easy to put in practice my idea. So, we could assume:

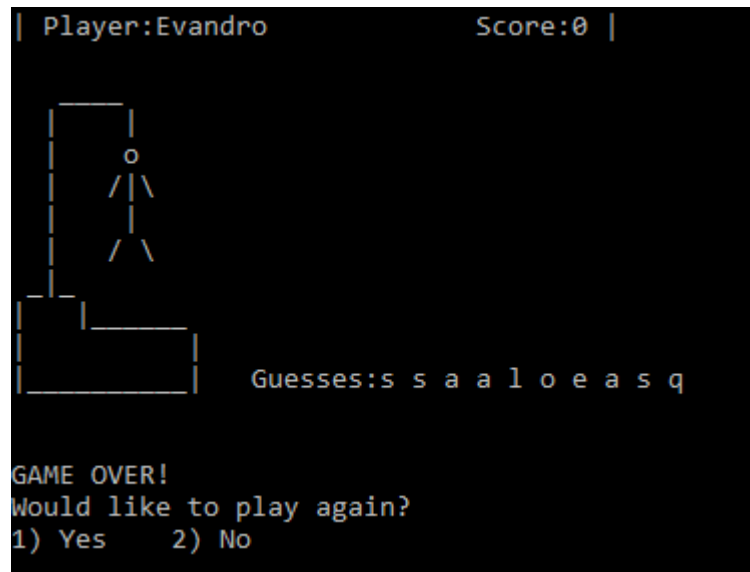
- The program will start loading the document respecting the difficulty selected by the user: easy, medium or hard.
 - The words, sorted in lines, would be added to my Array List.
 - The reason for using Array List its because I could easily generate a random number and use it as an index to select one of the items in my Array List.
- Generating a random number, the program will pick the index position in my Array List.
- This word will be storage in my 2D Array in the row [1].
- The 2D Array will convert the word into dashes and storage it in the row [0]
 - The last two steps were solved using a looping.
 - That's why I believe 2D Array was the key for my problem. Because having my counter at 0 in my looping for example, I could check rows [0] and row [1] at the same time and apply the necessary actions. Converting row [0] and checking player's guess in row [1].
- The game will collect the user's guess and compare with rows [0] and [1].
- Using another looping:
 - The looping will use the player's guess to check row [1].
 - Correct Answer: the program will change the respective dash position by the letter.
 - Incorrect Answer: player loses life.
 - After checking each index position, the looping will print row [0] and any correct answer will be showed to the player.

Then the rest of the program was basically complement the game, with options and more features. But the main idea is exactly as described above.

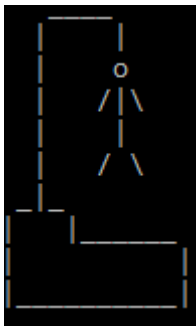
Design



For this game I tried to keep the design simple and clear. As my last game, I kept the menu based in numbers, so the user just need to press basically a few keys to advance in the game.



During all the match, the player can check his nickname on the top as long with his actual score. And all the letters guessed by the player will be registered.



The hangman was created from an ASCII image that I found on the internet. It is setup to show up every time that the user type incorrectly a letter.