

Connor McStay

Juan Arias

SD1

5/4/17

## Final Write-up

### **Introduction & Abstract:**

The game of hangman is a simple game on paper but in programming and on the computer it's a completely different animal. The game of hangman is usually played by two players, one chooses a word and the other must guess the word by guessing one letter at a time either until the word is guessed or the guesser has exhausted the amount of tries they are allowed, usually six. My project does almost the same thing, my program reads from a list of words that have been predetermined and selects one at random. Next, the program determines how many characters are in the word and creates an appropriate amount of spots for the word to fit in and tells the user how many characters are in the random word. Additionally the program then gives the user X amount of tries plus six, this is to ensure no matter what word is given you only have 6 tries to guess the word. At any time a player may want to exit the game which is also possible and then the program ends, if the word is not guessed correctly after X amount of attempts the game ends and displays a message saying game over. However, if the user was to win then the game displays a congratulatory line of text for the user.

### **Motivation:**

The motivation for the creation of this program was mostly the deadline and that I needed one for the course, however, the reason I chose hangman is because this game has

been very relevant in my life. I personally have been playing hangman for years and was a large part of my childhood playing it with my mother during my free time. I knew the game was easy enough to draw on a scrap piece of paper but not i have a nw respect for the game and every time I play it from now on I will appreciate how somethings are just easier on paper then on a computer.

### **Background information:**

The goal of the project was for the student to be able to experience working on a project from concept to implementation. I have selected a topic and developing a program about it. This has allowed me to apply the material that has been covered in class as well as prepare for the future as a software developer.

### **Overview:**

My program starts off with importing all necessary utilities for the project such as a scanner and a random util. The first thing in the function is a string that contains multiple words that the program randomly selects for the game, Then the program welcomes the user to the game and informs them that the minus sign(-) will exit the game at any time for their convenience is they wish to stop playing the game. After this a boolean statement asking if the game is being played is set to true, the user starts off with the game already running. Next the program breaks the random word down and checks to see how many characters are in the word, then it adds six to the amount of characters and uses that number to determine the number of guesses the player has. The program then generates underlined blank spaces for the random word, If the word is not guessed and there are still tries the game enters into a while loop, In this loop it tells the user how long the word is, how many correct guesses they have by

printing out the correct guesses and placing them into their predetermined space, tell the user how many tries they have left, and then prompts the user to enter another guess. The game prevents a user from guessing multiple letters at a single time by only selected the first character in the sequence a user may enter, this keeps the game fair. Furthermore this loop checks if the user wants to exit the game and if the minus sign is entered it breaks the loop and sets the boolean at the beginning to false and the game ends. If any other input is given then the game continues until the word is guessed and it then congratulates the user or the user has run out of tries and has not guessed the word, if that's the case then the game ends and prints a game over screen with a small picture of a stick figure at the gallows.

#### UML diagram:

##### Hangman

---

Scanner input  
Random random

---

```
+static void main
String[] guess
boolean weArePlaying = true
if (weArePlaying)
char[] randomWordToGuess
tries= word + 6
int ammountOfGuess
char[] playerGuess
for (int i = 0 ; i < playerGuess.length; i++)
playerGuess[i] = '_';
boolean wordIsGuessed
int tries = 0
while (!wordIsGuessed && tries != ammountOfGuess)
char c = input.nextLine().charAt(0);
tries++;
if (c == '-')
weArePlaying = false;
wordIsGuessed = true
else
```

```

for (int i = 0 ; i < randomWordToGuess.length; i++)
if (randomWordToGuess[i] == c)
playerGuess[i] = c
if (isTheWordGuessed(playerGuess))
wordIsGuessed = true;
if (ammountOfGuess - tries == 0)
+static void printArray(char[] array)
for (int i = 0 ; i < array.length; i++)
+static boolean isTheWordGuessed(char[] array)
for (int i = 0 ; i < array.length; i++)
if (array[i] == '_') return false
return true;

```

### **Problems encountered:**

There has been many different bugs and problems while coding this program, the greatest problem I had was with the part of the program that only selects the first character and only a character to ensure no cheating. However, this caused many problems, For one i wanted the user to say yes before the game started but this could not be done easily without having to heavily modify my code or allowed creates an easy way to win the game. Ultimately I ended up changing this feature. Another challenge that occurred was with the way the game would tell the user if they won or not, each iteration would tell the user “game over” but would repeat the loop until the tries were exhausted, this was fixed by fixing the boolean values. Furthermore, one of the most troubling problems that was encountered was figuring out how to exit the game using the minus sign while there were still tries, this was fixed by writing a different function and changing boolean values around in an “If” statement. All of these bugs have been fixed for the most part of have been eliminated from the program in order to ensure the program runs flawlessly.

### **What could be added:**

Many different features can be added to the program with enough time, I did attempt to add a few more features but then the program would not compile and found it better to not fix what's not broken. An addition that can be made is how many words can be played. A way to expand this to possibly link the game to a page or randomly generated words or to any website with a lot of text and have to randomly pick a word to use. Another way this can be improved would be to fix the bug that it takes a try away even if the letter guessed is correct, however the way it is programmed that is more of an astatic thing then a functional thing. If the game was to expand then it could possibly be made multipler either linked with another computer or on the same computer. One user could enter a word they would like the other to guess and then the screen would clear and the game would run with that word set as the random word. If I really wanted to make the game more aesthetically pleasing I could have the hangman be drawn for each incorrect guess, however at that point a GUI in HTML and CSS would be easier to incorporate instead of a bunch of `System.out.println` to make it look like the game is interactive. All these features could be added but would take a lot more time to incorporate and at that point it would be easier to complete scrap and rewrite the entire program from scratch.

## **Conclusion:**

I have learned a lot of Java during the duration of this project, I have learned how to troubleshoot and how to utilize Eclipse to help solve problems that may occur. Also I have taken this project very seriously and even though it is not a lot of code I fully understand what everything does in it, Normally with a lot of code some lines do not make too much sense to me but with this game I understand everything that each line does and that makes me very interested and happy with this code. In my personally opinion I believe this project has been successful and has meet the requirements, some features I have learned in class have not been

implemented though because I could not find a use for them and I could achieve the same outcome without making a bunch of classes and extending the page and using other overly complicated methods. Keep it short and sweet. Overall I am proud with my work and am very happy I was able to create this game.