**Your Name: Krystian Confeiteiro**

**Your Section: 10**

**Date: 10/08/20**

1. **What is the title of your project?**

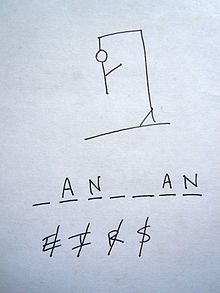
Digitalized Hangman

1. **Describe your project in DETAIL.** With a few paragraphs, describe your project with necessary details. For example, explain the rules of the game; or explain properties and functionalities of the database. Include pictures and drawings that help explain your idea.

The objective of Hangman is to guess the correct letters and/or words without completing the “Hangman” where if completed, the figure would be dead, and the game would be over. Normally, you start by drawing the structure the Hangman is being hanged with, then the Hangman would have two arms, legs, head, and a body, but to give the user more guesses, you can add other objects such as a hat, a watch, shoes, etc. Based on the number of letters in the word, the user will try to guess the correct letters or word if there are already enough letters guessed correctly. If the guess correctly, then they win the game.

Since this game would normally be played on paper or a whiteboard, I will have to put the “hangman” figure into an array along with the incorrect letters or words. For the correct letters guessed, I would have them replace the underscore that is a placeholder for the letter that the user has to guess.

To program this, I would have to know the number of guesses the user has, which would normally be 7 (structure, head, 2 legs, 2 arms, body). Then I would create a FOR loop with a running total which would be 7 as I mentioned. Inside the FOR loop, I would have and IF statement that would see if the user input is the same as the randomly generated word. If it is the same, the dash would be replaced and if not, a part would be added to the figure and an incorrect guess to the box.



As can be seen in this picture, the drawing has the post, body, one arm, and a head, 4 parts for the 4 incorrect guesses below the unknown word. In the unknown word, 2 letters were guessed correctly hence the A’s and N’s. The guesser only has 3 more guesses before the drawing is complete.

1. **How will your user interact with your program?**

Note: You must select only one of these three options. You are not to mix them in one project

**a.** Command Window **b.** Dialog Boxes **c.** Graphical-User Interfaces (GUIs) \*

1. **If implementing a game or other existing program that has existing standards, are there any attributes that you will be changing or not be including.** *(For example, if you implement Battleship and only intend to implement 4 ships instead of 5, list those changes here and explain why.)*

As I mentioned in question 2, the standard for Hangman is 7 guesses, I will be keeping that standard for my program as I feel it would be easier than adding extra guesses by included extra objects to put on the Hangman figure such as hat, watch, boots, etc.

1. **If implementing a game, will your game be: single player, multiple human players, or vs. a computer Artificial Intelligence (AI) player?** *If you will have AI player(s), explain to what extend they are able to think and follow strategies. A computer player with random actions is not an AI. Also be cautious of choosing multiple-human-players mode as it may greatly reduce the complexity of the program and result in disqualification.*

Hangman is a single-player game but can be played with a group. For this project, I am going to have only one person guess so that way there is only a single Hangman figure and only a single set of guesses.

1. **Go over the programming techniques required in your project – reflect on how you will include these in your code.** *(No submission required for this question)*

|  |
| --- |
| **INPUTS:** |
| User inputs/interface (e.g., input, dialog boxes, GUI) |
| File input (e.g., xlsread, dlmread, getcsv) |
| Random numbers (e.g., rand, randi) |
| **DATA TYPES:** |
| Numeric data and strings |
| Arrays (i.e., vectors, matrices, cell arrays) |
| **DATA PROCESSING:** |
| Rounding (i.e., round, ceil, floor) |
| Counting/calculating (e.g., running total, sum, mean, multiplication) |
| Organizing/analyzing (e.g., sorting, searching) |
| Array manipulation (i.e., referencing, slicing, augmenting, and diminution) |
| **LOGIC:** |
| Conditions (i.e., relational operators and Boolean operators) |
| Conditionals (i.e., if statement, switch/case) |
| Loops (i.e., while loop, for loop) |
| Nesting (loops and conditionals) |
| Error checking (all inputs appropriately error checked) |
| Programmer-defined functions (input parameters and return values) |
| String functions (e.g., sprintf, strcat, strfind, strcmp) |
| Other built-in function not discussed in class |
| **OUTPUTS/DISPLAY:** |
| Display relevant outputs to user/interface (e.g., fprintf, dialog boxes, GUI) |
| File output (e.g., xlsread, dlmread, putcsv) |
| Plotting (with appropriate data, formatting, and labeling) |

1. **Based on your reflections for question 6, answer the following questions.**
2. Arrays or cell arrays. What data do you intend to store in a vector, matrix, or cell array and how will you be manipulating it?

Since Hangman is a guess-based game, I will store all the incorrect guesses, whether they are words or letters, into a single vector that has two 7 row and 1 column with each row being an incorrect guess. For the Hangman figure, I am thinking of having an array where it would print different shapes depending on the guess number with the incorrect guesses vector next to the graphic.

1. Programmer-defined functions. Reasonable use of programmer-defined functions will make the project more organized and easier to debug. Think which parts of the project could be implemented in programmer-defined functions.

I do not have great knowledge of programmer-defined functions. I cannot think of a good part to use a programmer-defined function.

1. File Input (using something like dlmread() or xlsread()). How would you think you will use data file reading/writing?

I could use a file input to read a list of words that the program would randomly choose from and that would be the word that the user would have to guess.