**Hangman Game**

Name: Abdelrahman Abdallah Shalabi Mohamed Desoki

Neptun code: DOHDZF

Group: 3

1. Developer part

My game consisting of 6 files, named:

- main.cpp

- functionalities.cpp

-functionalities.h

-Intro.cpp

-Draw mistakes.cpp

-words. Text

Let me explain one by one and it is mission, well in **main.cpp** I run the test of my code, and the game by calling the defined base class and derived class, and it looks like that:  
Text

Description automatically generated

Then in **functionalities.h** I define all the classes I use and methods and attributes: here is the base class which run the game methods in it:  
Text

Description automatically generated

I define all the variables in it as protected to allow them to be inherited by derived class as the user may choose to play the game again, so I need to inherit all of them in that case.

Second thing in **functionalities.h** file is the derived class, which is small one, because most of its methods are already inherited from the base:

Text

Description automatically generated

Third part is: functionalities.cpp which almost all the game is handled in it, and also the main algorithm.

Let’s take it method by method:

Text

Description automatically generated

Here I am asking the user for his/her name to welcome him/her by their name, then   
A picture containing text

Description automatically generated

Int this method, I am printing the logo of the game ☺, then letting the user know the instructions before playing the game, then asking user about length of word he/she want to guess, after that asking the user about the level of difficulty, to set the game depending on the user choice, after that I assign this choice to the number of trials, to connect it to the mistakes after a while.  
Text

Description automatically generated

During the game I ask the user many times to enter a number, so I need to check always when this number is valid or not, so I decided to make this function to save time and not repeat the code every time  
Text

Description automatically generated

And in this function, I use exception handling, to throw an error message if the input is char->invalid input, if it’s invalid number then-> invalid number, then I print the limitation for the user to memorize it and check if the input is valid or not. After that I use this function to generate a random number, depending on the time on the laptop:

Text

Description automatically generated

Because I will need this number to connect it as an iterator to the file of words, and by knowing the size of the file (3001), then I put limits for the generated number to not exceed 3000.

Now I need to handle this file of words (words. Text)  
Text

Description automatically generated

So, in this function I open the file then loop through the content of the file then make a nested loop to check if the file has an uppercase letter, to let it be lowercase before saving it in my vector lastly, I close the file.

I need to check the right size of the word and test the game also

Text

Description automatically generated

So here I am starting to search in the file from the random number till the end, and to avoid collapse if the random number is = last element in the file I made a condition, to let it be 0 and start from the first, then I copy the right word in a char array called wordHandling, to use it afterwards with dash array. Here we came to the most important function in my game:  
Text

Description automatically generated

First I assign dashes to the dash array, print it, then I loop till the user win or lose, then I loop to check the user entered letter if it’s valid or not to not count mistake on another thing except letters, after that  
Text

Description automatically generated

I use to lower function to lowercase the letter which the user entered then I check if this letter entered before or not, after that I assign it to the array named check, then I loop to check if this letter is existing in the word itself or not, then I print the updated version of dashes with user correct trials, then I check the case of winning when the counter =length of the word, or the case of lose otherwise.   
Text

Description automatically generated

Let us turn into **intro.cpp file**, it is really small as it describes the 2 methods of the derived class.  
Text

Description automatically generated

Well in random function I narrow the scope of this number to decrease the probability of making it similar to the last one, then it is hard for the user to have similar word, then in the end function I ask the user if he/she want to play again or not, if yes then I will inherit all the methods from the base class in addition to the new random method, but if not then I will end the game.

Last file is draw mistakes which is related to functionalities.cpp but I separated it because it is so long, in this file I am defining the method of drawing the simulation to make game near to the reality  
Text

Description automatically generated

Ending with the man hanged as in real game ☺.

1. Testing part

If we run the project this is the output in the console:

Text

Description automatically generated

First the user needs to enter the name, then the instructions and logo will be printed, after that user will be required to enter the length of the word  
Text

Description automatically generated

Those inputs are all invalid, so it will keep asking till the input is valid, then if it is valid like this, the game will go on,   
Text

Description automatically generated

Okay after that user need to enter the level of difficulty then game must start,  
Text

Description automatically generated

As the photo describe, it will not count a mistake except it is a valid character, and if the guess is correct then dash will be replaced,

Text

Description automatically generated

If the guess is correct, then it will congratulate the user, print the score and ask if he/she want to play again:

Text

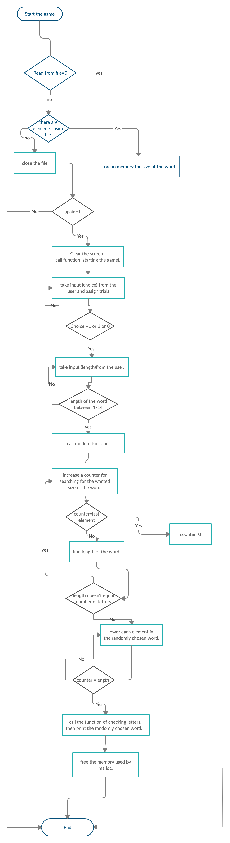
Description automatically generated

Let us suppose user choose to play again but this time want to lose,

Text

Description automatically generated

Then it will print for the user the score, and also the right word, and ask again the user about playing again.

3)Flowchart of the main idea  


It has the general idea of my project, not exactly but quite similar as abbreviation to my project.