**API\_HangmanGame**

The purpose of this API\_HangmanGame is to help the programmer create their own version of this small game without having to start from scratch. The only changes that the user will need to make to the API itself is just if they desire to change the words and hints that will be given to the player. For further information read this documentation.

The following table is here to show the methods that exists within this API, a brief description, the arguments each method takes and their respective return types.

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Return Type | Args | Desc |
| createScene | **Scene** | **BorderPane bp, int w, int h** | **Creates scene of the given width (w) and height(h) defining the size of the GUI. If both variables are <100 it will create a default size GUI (500x500)** |
| currentW | **String[]** | **none** | **Initialize a String[] where the words for the game will be store. Here is where you can define which word will be guessed** |
| wordBoxes | **GridPane** | **String cw** | **Creates a GridPane where TextField[] tfs size is defined by each letter of the String entered (each letter of the word will be entered in a different TextField.)** |
| moveUp | **Void** | **TextField[] tfs** | **This method changes the current TextField in use once this is filled to the next one. This method is meant to work along with “wordBoxes”** |
| Createbutton | **Button** | **String name, String style** | **Generates a button every time it is called. Both arguments define characteristics that make each button different.** |
| getHint | **String** | **String crWord** | **Define a switch statements for each word and returns hints for each of them (you have to type the hints you want to give the users.)** |
| guessWord | **Int** | **String cWord, int Points** | **Checks if the current word(cWord) matches with the user’s input by checking each letter entered in the TextFields. Whether the letter entered is in the right order, the background of the TextField changes. While keeping track of the score.** |
| showWinner | **Void** | **Int s, Image img** | **Creates and display a second stage to show an image every time the user clicks on guess word. This method was created to work along with guessWord to create the scene and add an image whether the user wins or loses.** |
| ranNum | **Int** | **none** | **Method's purpose is to generate a random number that will be used in other methods.** |
| cWord | **String** | **none** | **This method assigns a random number to the current word for later use along with other methods** |

**One-Line Description of all files included in .zip**

**API\_HangmanGame.java is the API (currentW & getHint ARE THE ONLY METHODS YOU CAN CHANGE to personalized)**

**HangmanGame.java is the example of the program working with the API as it was planned originally.**

**HangmanGame\_Template*.java* The file that inherits from the API. Follow the comments labeled “CODE GOES HERE:” to create your own version of the game!**

**MyImages.java is a file created to storage the winner/loser images that we will use in the game.**

**Example Results**

**Graphical user interface

Description automatically generated with medium confidence**

**A picture containing graphical user interface

Description automatically generated**