United International University

QUEST FOR EXCELLENCE

ASSIGNMENT Details

Assignment Title: HANGMAN GAME

Course Name: Advanced Object Oriented Programming Laboratory

Course Code: CSE 2118/CSI 124

Submitted by:

Name: Yeasir Arafat

ID: 011 201 035

Section: “B”

Department: Computer Science & Engineering.

Date of submission: 8-31-2021

Submitted to:

FAHIM SHAHRIAR

Lecturer

United International University

Table of contents

Introduction: This report contains the documentation of the HANGMAN Game that I've made using JAVA FX and Scene Builder.
Whole Project Here : HANGMAN GAME

Game Logic:

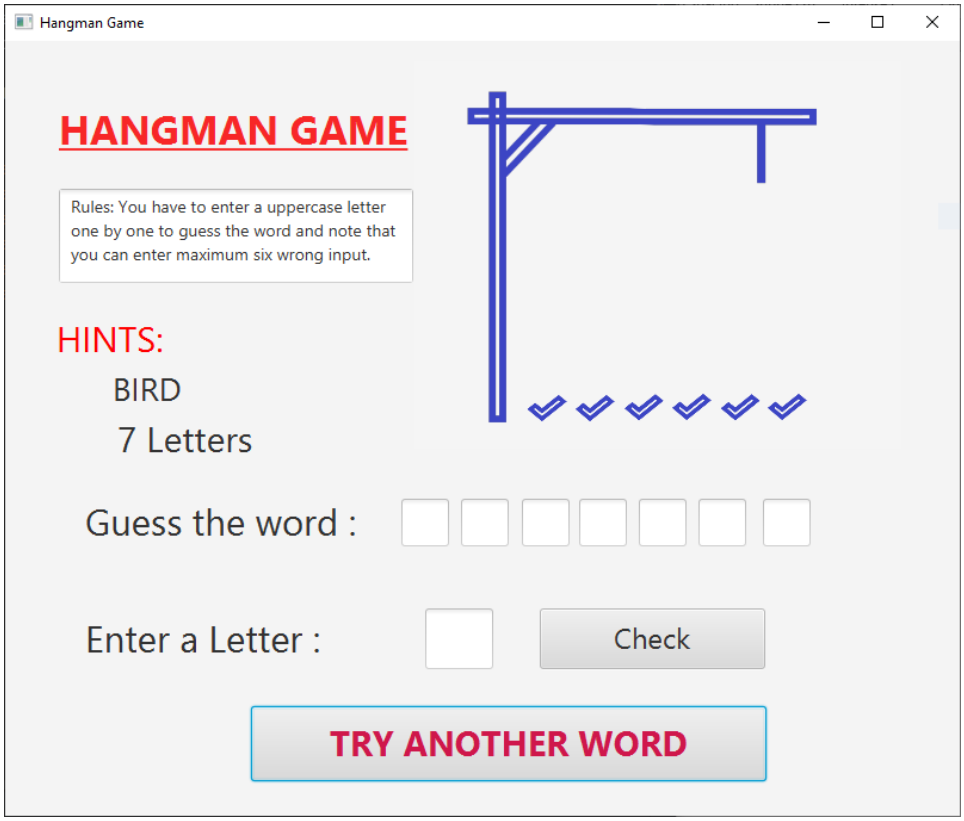
- First of all the user will be given two hints and he needs to guess the word.
- The word will be randomly picked from a String array.
- The user has to input an uppercase letter.
- If the letter matches with any character in the word, then the letter will be added to the textfield at its respective position.
- And if the letter dosent match with any character in the String, then the picture will change and also chances/life will be decremented.
- Initially the life/chance will be set to 6.
- That means the user can try attempting wrong letter maximum 6 times.
- Every picture will add body parts one by one.
- When all the body parts will be added to the picture, the game will be over.
- And if the user guesses the word before the life becomes 0, then the user will win.

SCENE-1 : This is the first scene where I've added a picture, a button and label.

Start Playing button switches to a new scene once it is clicked. The necessary code for switching scene is given below.

```
public void changeScene(ActionEvent event) throws IOException {
    Parent parent = FXMLLoader.load(getClass().getResource("gameScene.fxml"));
    Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();
    window.setTitle("Hangman Game");
    window.setScene(new Scene(parent, 800, 650));
    window.show();
}
```

SCENE-2: In scene2, I've added some Labels, non editable TextFields , a editable TextField where user will give the input, added some buttons and added ImageView for showing the picture.



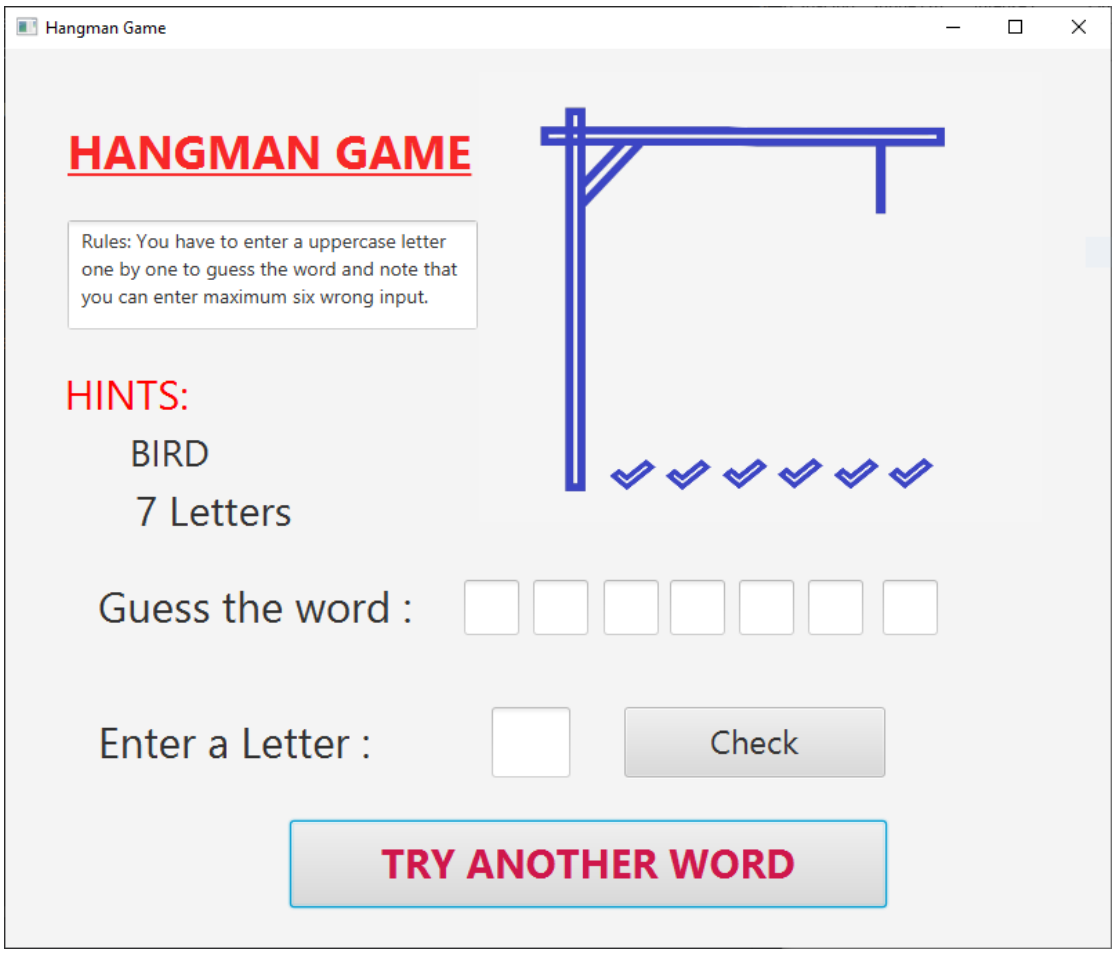
I've added some words and hints to a string array. Whenever the scene is opened, it will randomly select a string from the string array. The String is divided into two parts, one is the actual word and another is for giving hints to the user. Whenver user clicks **Start Playing** or **Try Another Word** Button, then the **setHint()** method will call with the help of **initialize()** method. And then setHint() method will set the hints, also it will make visibility false for the unnecessary TextFields required to fill the word. For example if the word has 4 letter, then it is gonna make only 4 TextFields visible.

```
public class Controller2 {
    @FXML
    ImageView img;
    Image image2 = new Image(getClass().getResourceAsStream("images/2.png"));
    Image image3 = new Image(getClass().getResourceAsStream("images/3.png"));
    Image image4 = new Image(getClass().getResourceAsStream("images/4.png"));
    Image image5 = new Image(getClass().getResourceAsStream("images/5.png"));
    Image image6 = new Image(getClass().getResourceAsStream("images/6.png"));
    Image image7 = new Image(getClass().getResourceAsStream("images/7.png"));
    @FXML
```

```
TextField tf1;
@FXML
TextField tf2;
@FXML
TextField tf3;
@FXML
TextField tf4;
@FXML
TextField tf5;
@FXML
TextField tf6;
@FXML
TextField tf7;
@FXML
TextField tf8;
@FXML
TextField input;
@FXML
Label hint;
@FXML
Label letter_count;
@FXML
Label hint_label;

String[] data = {
        "MEXICO COUNTRY","HEDWIG BIRD","KUAKATA BEACH","CANADA COUNTRY",
        "DOCTOR PROFESSION","FOOTBALL GAME","TEACHER MENTOR",
        "LEOPARD ANIMAL","BICYCLE TRANSPORT","SALMON FISH","SPARROW BIRD",
        "PARROTS BIRD","EAGLE BIRD","TRAIN TRANSPORT","SHIP TRANSPORT",
        "ENGINEER PROFESSION","BANKER PROFESSION","CRICKET GAME"
};

int random = new Random().nextInt(data.length);
String word_hint = data [random];
String[] split = word_hint.split(" ", 2);
String word = split[0];
String hint_str = split[1];
int letter_size = word.length();
```

```
public void initialize(){
    setHint();
}
public void setHint(){
    hint.setText(hint_str);
    letter_count.setText(letter_size+" Letters");

    if(letter_size==7){
        tf8.setVisible(false);
    }
    if(letter_size==6){
        tf7.setVisible(false);
        tf8.setVisible(false);
    }
    if(letter_size==5){
        tf6.setVisible(false);
        tf7.setVisible(false);
        tf8.setVisible(false);
    }
    if(letter_size==4){
        tf5.setVisible(false);
        tf6.setVisible(false);
        tf7.setVisible(false);
        tf8.setVisible(false);
    }
}
```

What will happen when user click **Check Button?**

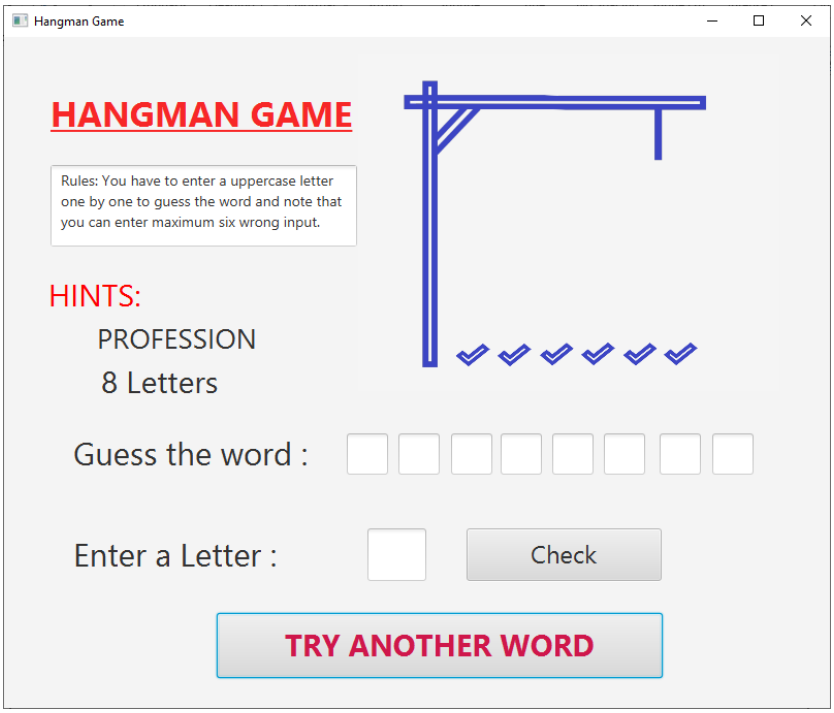


When the user Enters a letter and clicks the **Check** Button, it is going to call **CheckInput()** method. In CheckInput method, the letter that is given by the user will be extracted from the TextField and it will be checked with the word that is randomly selected previously. If the letter is available in the word, then it will find the indexes of the letter and then it will call **setLetter()** method with the **index and the letter** as parameter and the **setLetter()** method will set the letter to the TextField at its index. And if the word doesn't contain the letter then it will call **setImage()** method and it will change the picture of the ImageView and also decrement the number of life available.

```
public void CheckInput(){
    String str = input.getText();
    if (word.contains(str)) {
        int index = 0;
        for(int i=0; i<word.length(); i++) {
            char c = word.charAt(i);
            if (String.valueOf(c).equals(str)) {
                setLetter(index, Character.toString(c));
            }
            index++;
        }
    }
    else {
        setImage();
    }
}

public void setLetter(int index,String str){
    if(index==0)
        tf1.setText(str);
    else if(index==1)
        tf2.setText(str);
    else if(index==2)
        tf3.setText(str);
    else if(index==3)
        tf4.setText(str);
    else if(index==4)
        tf5.setText(str);
    else if(index==5)
        tf6.setText(str);
    else if(index==6)
        tf7.setText(str);
    else if(index==7)
        tf8.setText(str);
}
```

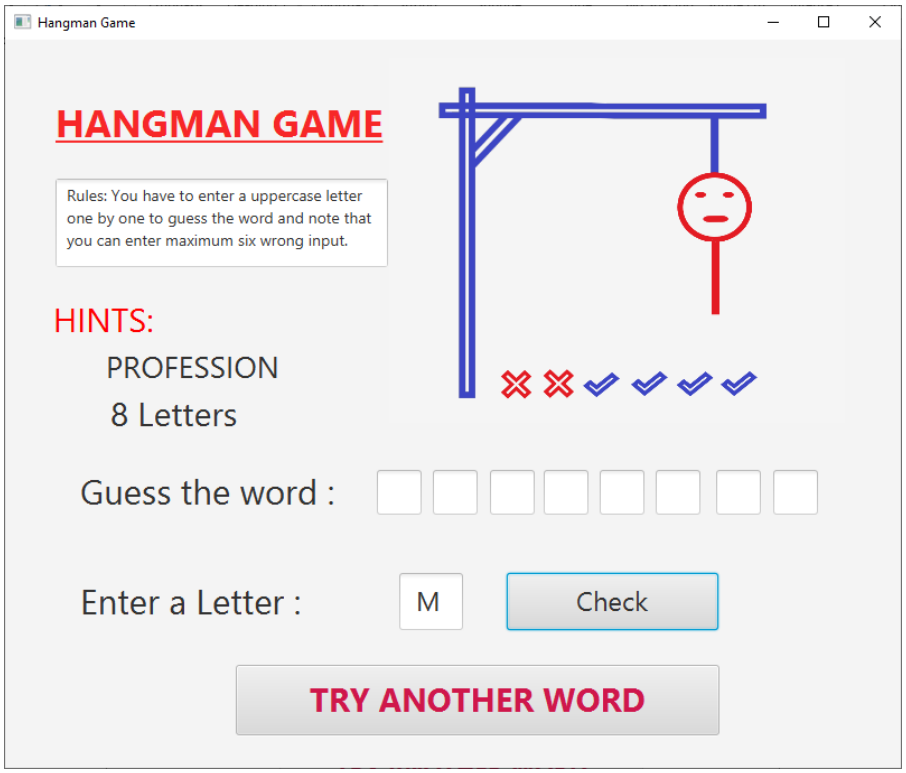
```
int life=6;
public void setImage(){
    if(life==6)
        img.setImage(image2);
    else if(life==5)
        img.setImage(image3);
    else if(life==4)
        img.setImage(image4);
    else if(life==3)
        img.setImage(image5);
    else if(life==2)
        img.setImage(image6);
    else if(life==1)
        img.setImage(image7);
    life--;
}
```

GAME SCREENSHOTS & WOKRING PROCESS:

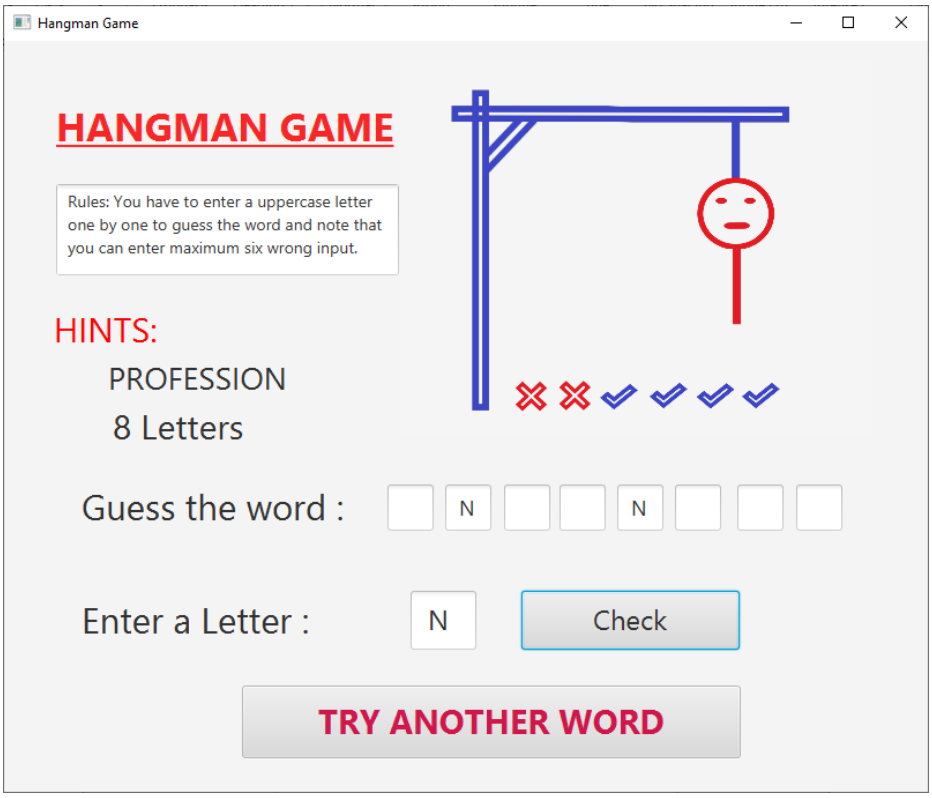


Suppose user enters **A** and click **Check** button, in that case, the letter **A** is not available in the word. So, it will change the picture and decrement the life.

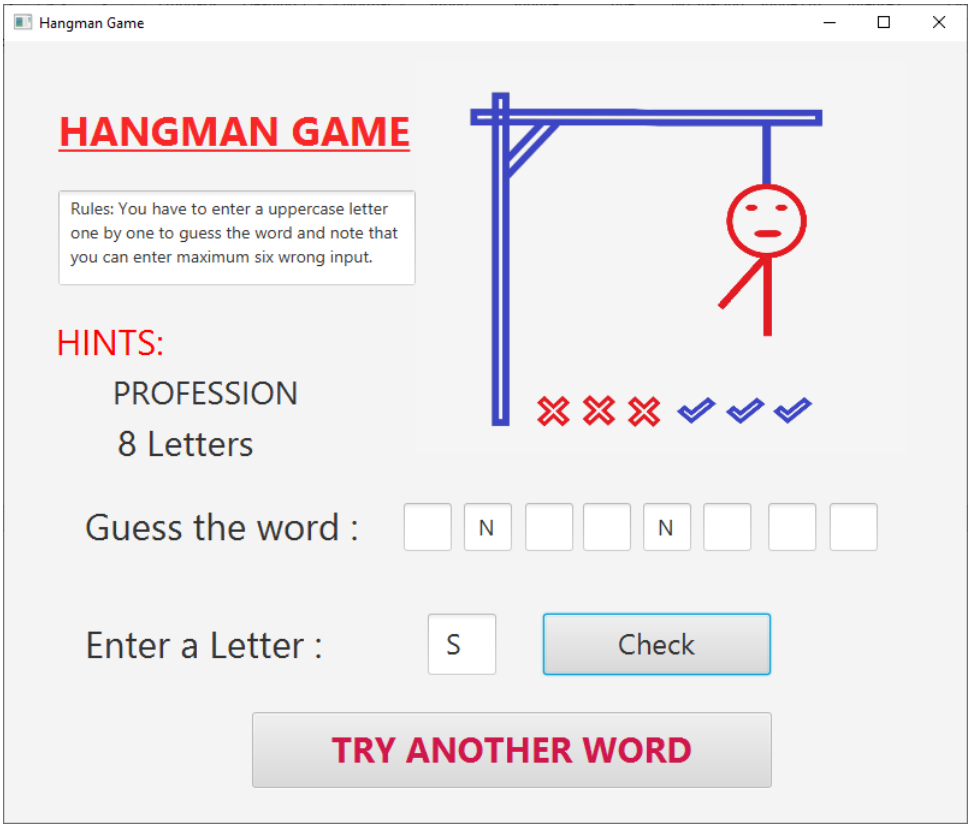
So again user enters **M** and clicked **Check** Button. In that case **M** is not available in the word, so it will change the picture and decrement the life.



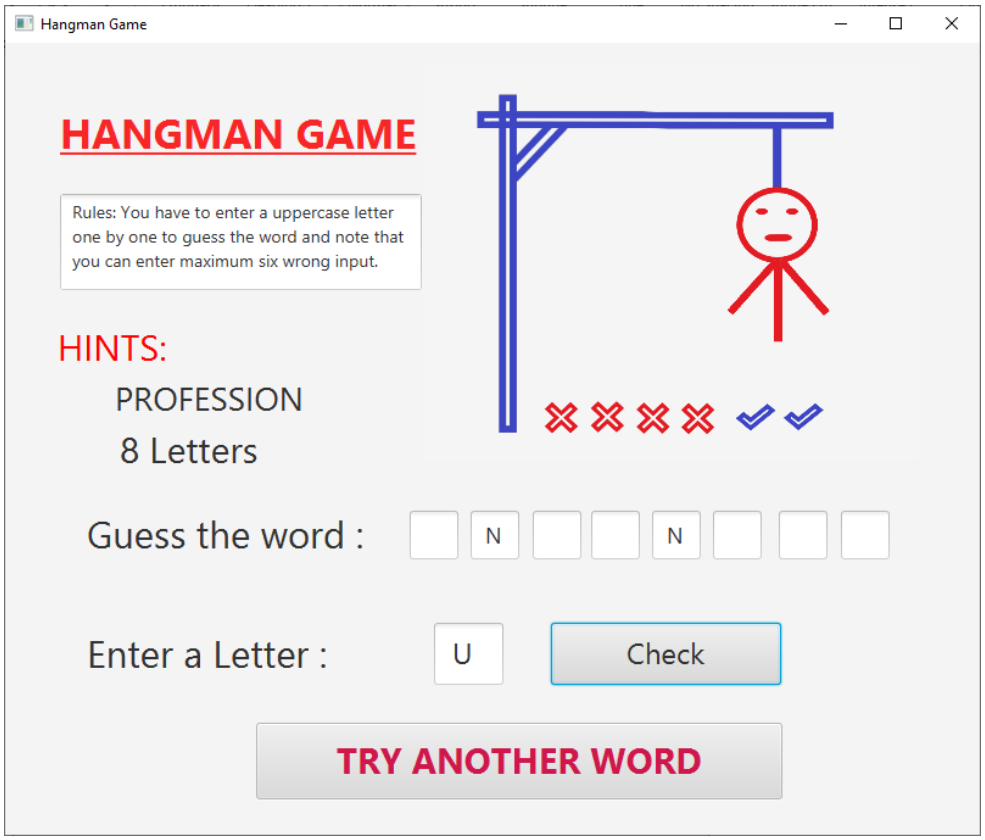
Enters **N**. In that case, the letter is available in the word two times , So the letter will setted to its correct position.



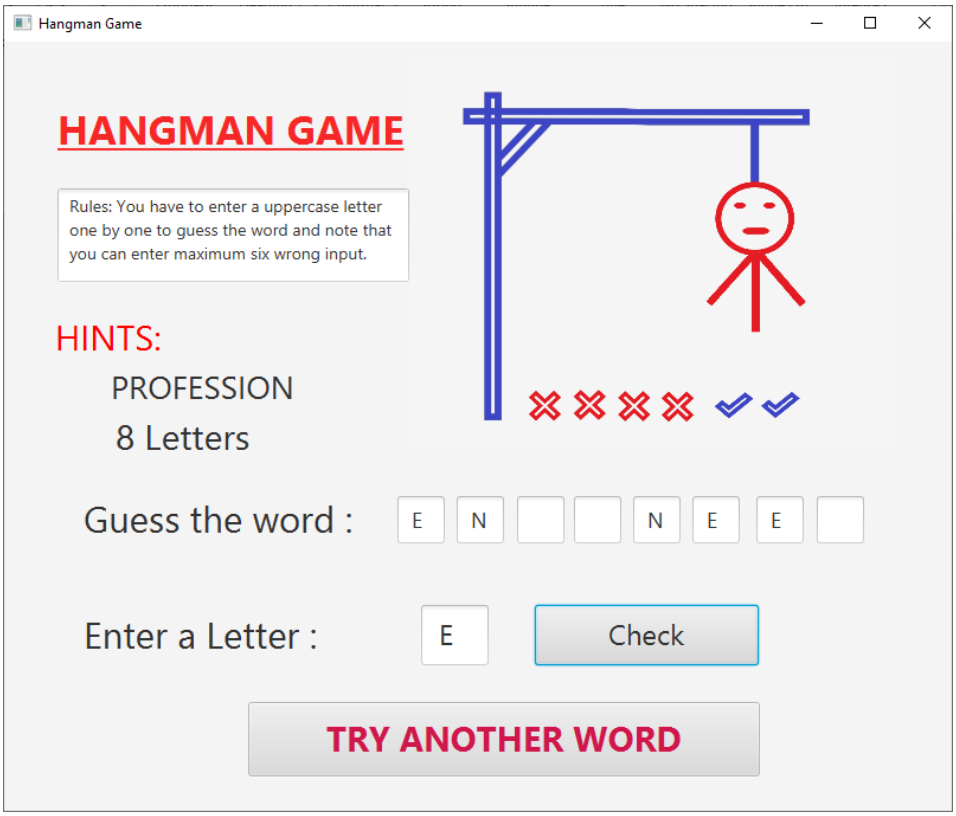
Again **S** is not in the word, so the picture changes and decrement life.



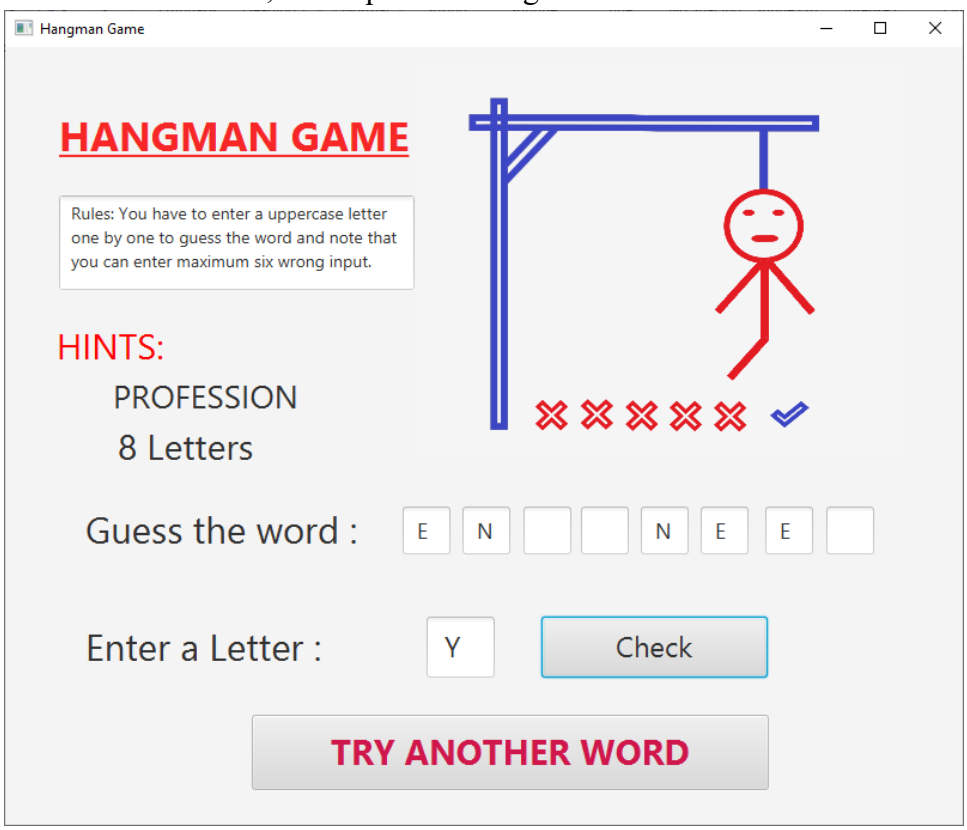
U is not in the word, so the picture changes and decrement life.



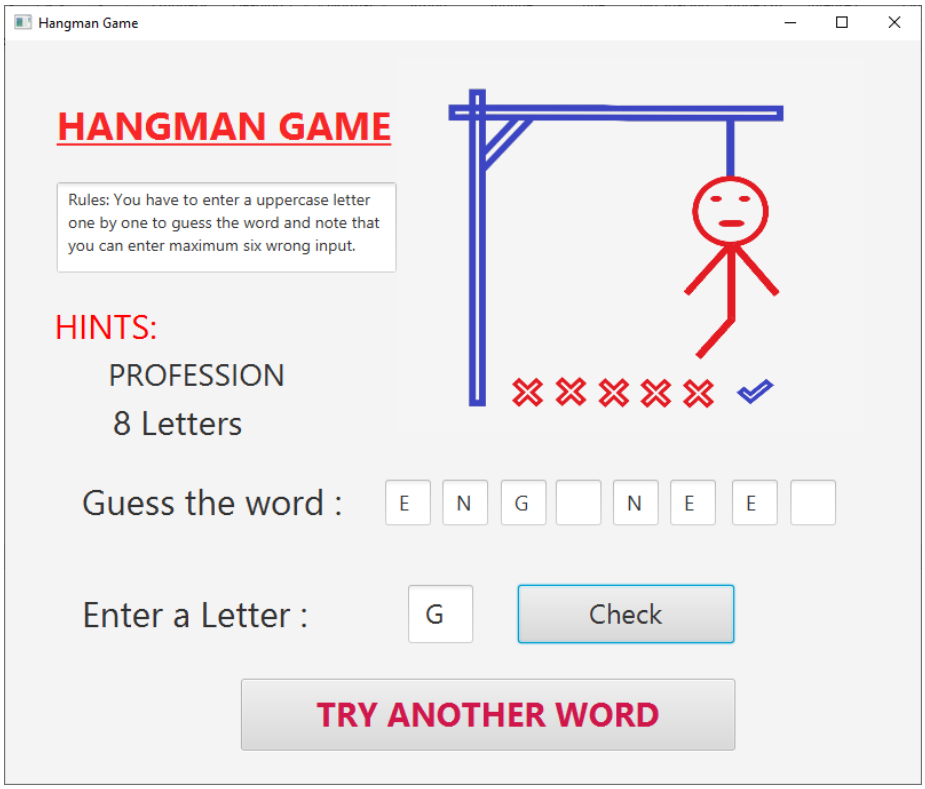
E is available in the word there times , so **E** will be added to its correct positions.



Y is not in the word, so the picture changes and decrement life.

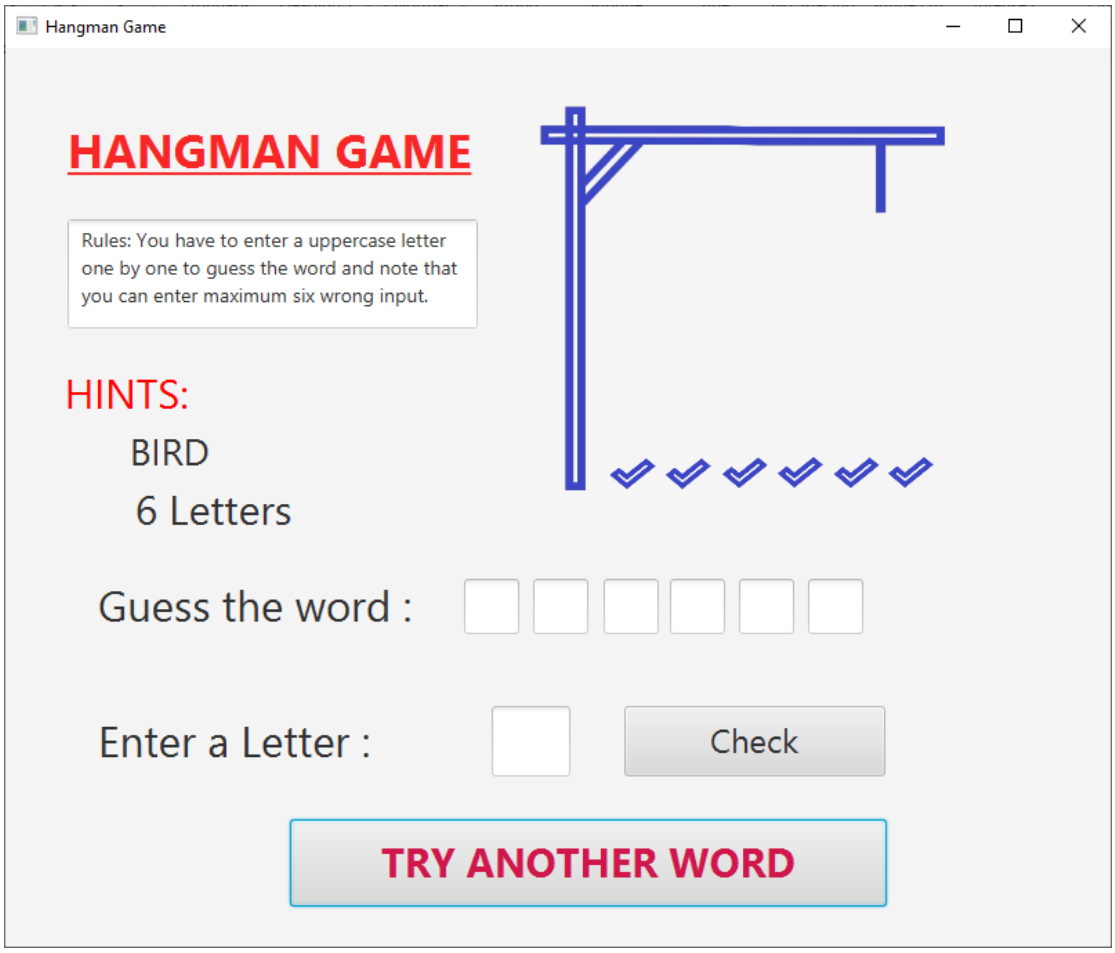


G is available in the word, so **G** will be added to its correct position.



C is not in the word, so the picture changes and decrement life. And also the game will be over!

Now if the user clicks on the **TRY ANOTHER WORD** button, then it will reopen the same Scene and it will pick a random word from the string array.



This was the documentation of the game that I made. The full project of the game is available at this link. Thank you.