Dokumentation
Modul226A

Von
Endrit Lena

Inhaltsverzeichnis

Henkerspiel

Anforderungen

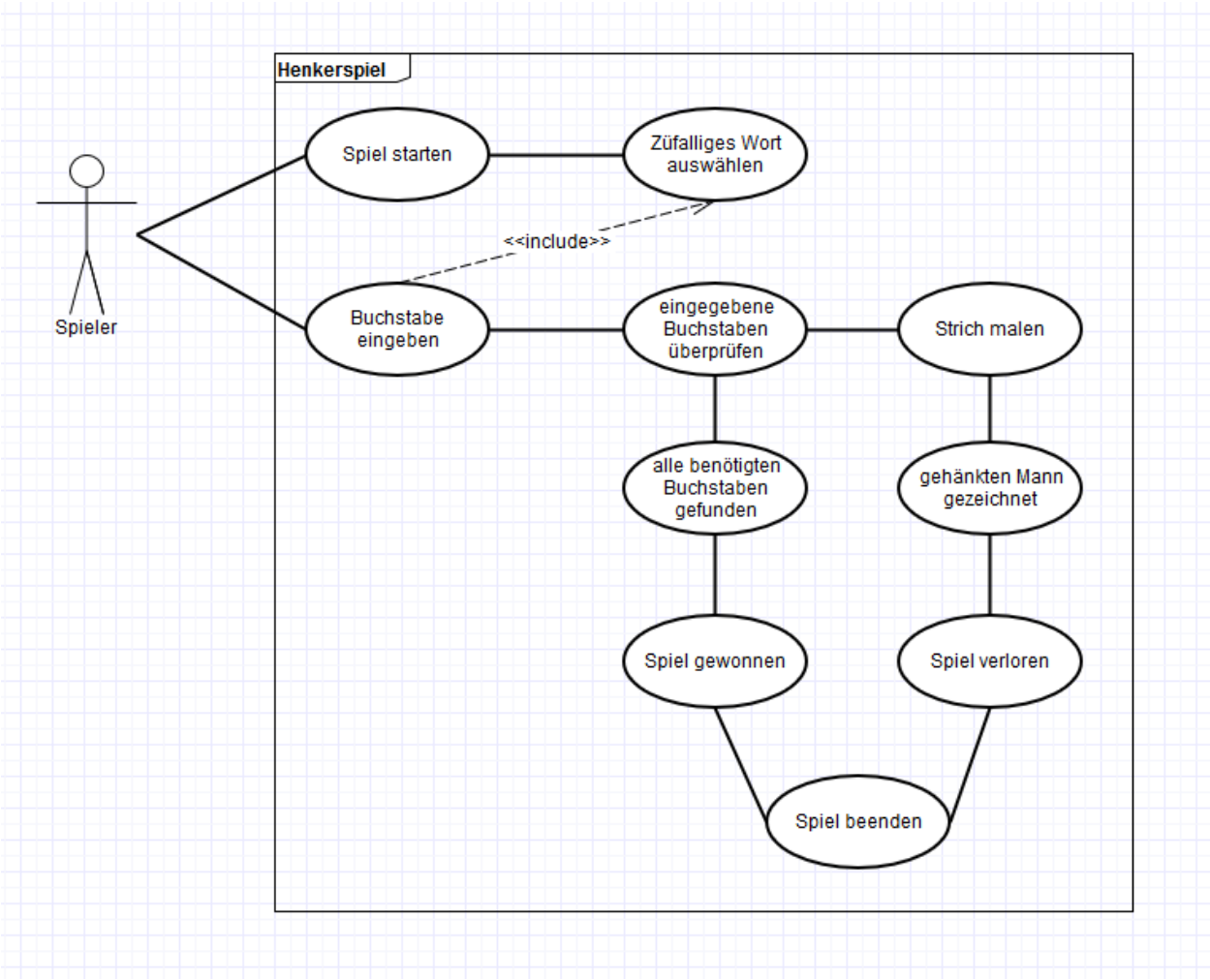
Funktionale Anforderungen

- Das Spiel sollte starten können
- Ein zufälliges Wort sollte ausgewählt werden
- Das ausgewählte Wort nicht sofort angezeigt werden
- Das ausgewählte Wort sollte als Striche angezeigt werden
- Der Spieler sollte Buchstaben eingeben können
- Bei richtigen eingaben sollten die Buchstaben angezeigt werden
- Bei falschen eingaben sollten striche gemalt werden
- Wenn das Wort gefunden wurde sollte der Spieler gewinnen
- Wenn der Strichmännchen gezeichnet wurde sollte der Spieler verlieren

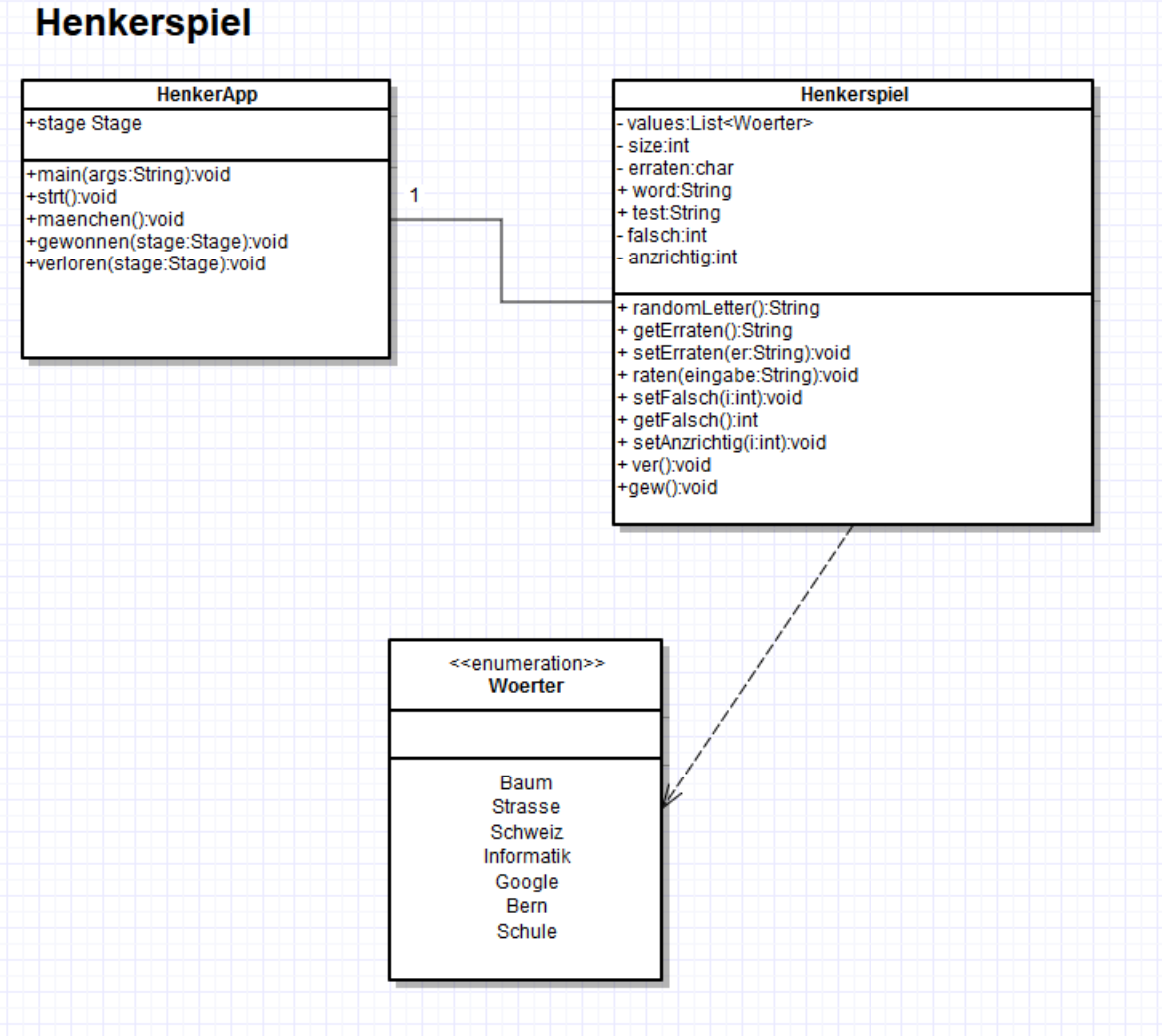
Nicht Funktionale Anforderungen

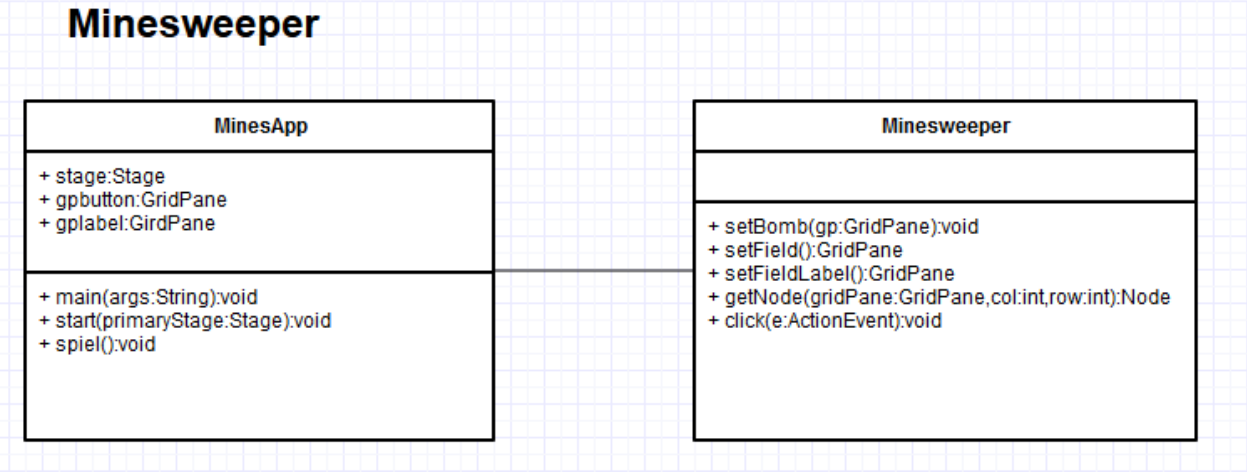
Anforderung	**Beschreibung**	**Wichtigkeit**		
		Hoch	Mittel	Klein
Produktanforderungen				
Speicherplatz	< 100MB		x	
JDK	min. version 8	x		
Unternehmensanforderungen				
Peripheriegeräte	Tastatur, Maus		x	
Externe Anforderungen				
Ethische Anforderungen	Keine Fluch Wörter oder so etwas in der art.	x		

Use-Case Diagramm



Klassendiagramme





Code erläutern

```
package ch.csbe.henkerspiel;

import java.util.Arrays;
import java.util.List;
import java.util.Random;

/**
 *
 * @author endrit.lena
 *
 */

public class Henkerspiel {

	private static List<Woerter> values = Arrays.asList(Woerter.values());
//verbindet sich mit den Enum Woerter
	private static int size = values.size();	//erstellt einen integer namens
size welche die grösse des enums speichert
	private static Random random = new Random();  //erstellt ein Random welches
benutzt wird um irgendetwas zurfällig zu wählen
	private static char[] erraten;   //erstellt einen character namens erraten
	public static String word;//erstellt einen String namens word
	public static String test;//erstellt einen String namens test
	private static int falsch = 0;   //erstellt einen integer namens falsch mit
dem wert 0
	private static int anzrichtig = 0;	//erstellt einen integer namens
anzrichtig mit dem wert 0

	//randomLetter wählt ein zufälliges Wort vom Enum Woerter
	public static String randomLetter(){
		String a = values.get(random.nextInt(size)).toString();
	    return a;

	  }

	//getErraten nimmt das zufällige wort
	public static String getErraten(){
		String tostring = "";
		for(int i = 0;i<erraten.length;i++){
			tostring += erraten[i];
		}
		return tostring;
	}

	//setErraten setzt das zufällige Wort
	public static void setErraten(String er){
		erraten = new char[er.length()];
		for(int i =  0; i<er.length(); i++){
			if(er.charAt(i) == ' '){
				erraten[i] = ' ';
			}
			else{
				erraten[i] = '_';
```

```
                }
            }
        }

        //raten schaut ob der eingegebene Buchstabe mit einem buchstaben des Wortes
identisch ist.
        public static void raten(String eingabe) throws Exception{
            boolean richtig = false;
            for(int j = 0; j < erraten.length; j++){
                if(word.charAt(j) == eingabe.charAt(0) || word.charAt(j) ==
eingabe.charAt(0)-32){ //wenn es identisch ist wird der buchstabe ausgegeben
                    erraten[j] = word.charAt(j);
                    richtig = true;
                    anzrichtig++;
                    if(word.length() == anzrichtig){
                        gew();
                    }
                }
            }
            if(!richtig){//wenn es nicht identisch ist wird eine Linie des
Strichmänchen gezeichnet
                falsch++;
                HenkerApp.maenchen();
                if(falsch == 10){
                    ver();
                }
            }
        }

        //setFalsch wird ausgeführt wenn ein Buchstabe falsch ist
        public static void setFalsch(int i){
            falsch = i;
        }

        //getFalsch gibt die anzahl der falsch eingegeben Buchstaben dem integer
falsch zurück
        public static int getFalsch(){
            return falsch;
        }

        //setAnzrichtig wird ausgeführt wenn ein Buchstabe richtig ist
        public static void setAnzrichtig(int i){
            anzrichtig = i;
        }

        //ver verlinkt zu verloren() im HenkerApp wenn man verloren hat
        public static void ver() throws Exception{
            HenkerApp.verloren(HenkerApp.stage);
        }

        //gew verlinkt zu gewonnen() im HenkerApp wenn man gewonnen hat
        public static void gew() throws Exception{
            HenkerApp.gewonnen(HenkerApp.stage);
```

```
package ch.csbe.henkerspiel;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.Pane;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Line;
import javafx.scene.text.Font;
import javafx.stage.Stage;

/**
 *
 * @author endrit.lena
 *
 */

public class HenkerApp extends Application{

	//Line & Circle dienen zur zeichnung des Mänchen
	static Line li1 = new Line(100, 100, 0, 100);
	static Line li2 = new Line(0, 150, 0, 20);
	static Line li3 = new Line(70, 100, 0, 100);
	static Line li4 = new Line(0, 50, 0, 20);
	static Circle c = new Circle(100, 100, 10);
	static Line li6 = new Line(0, 50, 0, 20);
	static Line li7 = new Line(20, 80, 0, 100);
	static Line li8 = new Line(-20, 80, 0, 100);
	static Line li9 = new Line(20, -80, 0, -100);
	static Line li10 = new Line(-20, -80, 0, -100);
	public static Stage stage;

	//main startet die Anwendung
	public static void main(String[] args) {
		launch(args);
	}

	//Hier steht was geschehen soll wenn die Anwendung gestartet wird
	@Override
	public void start(Stage primaryStage) throws Exception {
		stage = primaryStage;
		strt();		//führt zur strt Methode
	}

	//strt ist das GUI, welches man sieht wenn die Anwendung startet
	public static void strt(){
		Henkerspiel.setFalsch(0); //setzt den Integer falsch auf 0
		Henkerspiel.setAnzrichtig(0);	//setzt den Integer anzrichtig auf 0
```

```
            Henkerspiel.word = Henkerspiel.randomLetter();      //Speichert das
zufall gewählte Wort vom randomLetter im String word
            Henkerspiel.setErraten(Henkerspiel.word);     //setzt das ausgewählte
wort vom String word zum Character erraten
            Group gr = new Group();   //erstellt eine neue Gruppe
            Scene scene = new Scene(gr, 600, 450); //erstellt eine neue Scene
            Pane pane = new Pane();   //erstellt eine neue Pane
            Pane pa = new Pane();     //erstellt eine andere neue Pane
            Label l = new Label("Henkerspiel");    //erstellt einen Label welches
"Henkerspiel" heisst und wird als Titel im spiel benutzt
            Button b = new Button("Enter");  //erstellt einen neuen Button mit dem
name "Enter"
            //erstellt ein neues Label welches das Gesuchte Wort ausgibt
            Label wort = new Label("Gesuchtes Wort: " + Henkerspiel.getErraten() + "
["  + Henkerspiel.word.length() + " Buchstaben]");
                  System.out.println(Henkerspiel.word);  //gibt das gesuchte Wort
in der Konsole aus
            Label lb = new Label("Buchstabe eingeben:");  //ein neues Label welches
über den TextField mit der beschriftung "Buchstabe eingeben:" steht
            TextField txt = new TextField(); //erstellt ein neues TextFiel in
welchem man Buchstaben eingeben kann

            li1.setLayoutX(380); //setzt die X-Position auf die angegebene zahl
            li1.setLayoutY(200);      //setzt die Y-Position auf die angegeben zahl
            li1.setVisible(false);    //macht den Label unsichtbar

            li2.setLayoutX(420);
            li2.setLayoutY(150);
            li2.setVisible(false);

            li3.setLayoutX(420);
            li3.setLayoutY(70);
            li3.setVisible(false);

            li4.setLayoutX(490);
            li4.setLayoutY(150);
            li4.setVisible(false);

            c.setLayoutX(390);
            c.setLayoutY(110);
            c.setVisible(false);

            li6.setLayoutX(490);
            li6.setLayoutY(200);
            li6.setVisible(false);

            li7.setLayoutX(490);
            li7.setLayoutY(130);
            li7.setVisible(false);

            li8.setLayoutX(490);
            li8.setLayoutY(130);
            li8.setVisible(false);

            li9.setLayoutX(490);
```

```
            li9.setLayoutY(350);
            li9.setVisible(false);

            li10.setLayoutX(490);
            li10.setLayoutY(350);
            li10.setVisible(false);

            b.setLayoutX(275);
            b.setLayoutY(135);
            b.setFont(new Font("Arial", 15));      //setzt die Schriftart auf Arial
und die Schriftgrösse auf 15
            b.setOnAction(new EventHandler<ActionEvent>() {     //dies macht, das
der Button etwas macht wenn man drauf klickt
                @Override public void handle(ActionEvent e) {
                        try {
                               Henkerspiel.raten(txt.getText());
                        } catch (Exception e1) {
                               e1.printStackTrace();
                        }
                        wort.setText("Gesuchtes Wort: " + Henkerspiel.getErraten()
+ "  ["  + Henkerspiel.word.length() + " Buchstaben]");
                        txt.clear();
                }
            });

            lb.setLayoutX(10);
            lb.setLayoutY(115);
            lb.setFont(new Font("Arial", 15));

            txt.setLayoutX(10);
            txt.setLayoutY(135);
            txt.setPadding(new Insets(5, 120, 5, 5));

            wort.setLayoutX(10);
            wort.setLayoutY(65);
            wort.setFont(new Font("Arial", 15));

            l.setLayoutX(10);
            l.setLayoutY(10);
            l.setFont(new Font("Arial", 30));

            pa.setPadding(new Insets(0, 200, 350, 0)); //setzt die Grösse des Pane
            pa.setStyle("-fx-background-color: white;"    //setzt Hintergrundfarbe,
Rahmen und Rahmenfarbe der Pane
                               + "-fx-border-style: solid;"
                               + "-fx-border-color: gray;");
            pa.setLayoutX(350);
            pa.setLayoutY(50);

            pane.setPadding(new Insets(0, 600, 450, 0));
            pane.setStyle("-fx-background-color: #e6e6e6;");

            //macht das die erstellten Sachen auf der scene ausgegeben werden
(sozusage macht sie sichtbar)
```

```
            gr.getChildren().addAll(pane, pa, l, wort, txt, lb, b, li1, li2, li3,
li4, c, li6, li7, li8, li9, li10);

            stage.setScene(scene);    //setzt die scene
            stage.setTitle("Henkerspiel");   //gibt der scene einen Titel
            stage.show();//zeigt die scene

      }

      //maenchen ist die methode die den Höngenen mänchen zeichnet
      public static void maenchen(){
            switch(Henkerspiel.getFalsch()){
            case 1: li1.setVisible(true);break;    //macht die Linien & Circle
sichtbar
            case 2: li2.setVisible(true);break;
            case 3: li3.setVisible(true);break;
            case 4: li4.setVisible(true);break;
            case 5: c.setVisible(true);break;
            case 6: li6.setVisible(true);break;
            case 7: li7.setVisible(true);break;
            case 8: li8.setVisible(true);break;
            case 9: li9.setVisible(true);break;
            case 10: li10.setVisible(true);break;
            }

      }

      //gewonnen öffnet sich wenn man das gesuchte Wort gefunden hat
      public static void gewonnen(Stage stage){
            Group gr = new Group();
            Scene scene = new Scene(gr, 600, 450);
            Label l = new Label("Gewonnen");
            Button b = new Button("Neustart");

            b.setLayoutX(125);
            b.setLayoutY(135);
            b.setFont(new Font("Arial", 30));
            b.setOnAction(new EventHandler<ActionEvent>() {
                @Override public void handle(ActionEvent e) {
                        strt();
                }
            });

            l.setLayoutX(10);
            l.setLayoutY(10);
            l.setFont(new Font("Arial", 30));

            gr.getChildren().addAll(l,b);

            stage.setScene(scene);
            stage.setTitle("Gewonnen");
            stage.show();
      }
```

```
	//verloren öffnet sich wenn man das gesuchte wort nicht gefunden hat und das
Strichmänchen gezeichnet wurde
	public static void verloren(Stage stage){
		Group gr = new Group();
		Scene scene = new Scene(gr, 600, 450);
		Label l = new Label("Verloren");
		Pane pa = new Pane();

		l.setLayoutX(10);
		l.setLayoutY(10);
		l.setFont(new Font("Arial", 30));
		Button b = new Button("Neustart");

		b.setLayoutX(125);
		b.setLayoutY(135);
		b.setFont(new Font("Arial", 30));
		b.setOnAction(new EventHandler<ActionEvent>() {
		    @Override public void handle(ActionEvent e) {
				strt();
		    }
		});

		l.setLayoutX(10);
		l.setLayoutY(10);
		l.setFont(new Font("Arial", 30));

		gr.getChildren().addAll(l,b, pa,li1, li2, li3, li4, c, li6, li7, li8,
li9, li10);

		stage.setScene(scene);
		stage.setTitle("Verloren");
		stage.show();

		li1.setLayoutX(380);
		li1.setLayoutY(200);

		li2.setLayoutX(420);
		li2.setLayoutY(150);

		li3.setLayoutX(420);
		li3.setLayoutY(70);

		li4.setLayoutX(490);
		li4.setLayoutY(150);

		c.setLayoutX(390);
		c.setLayoutY(110);

		li6.setLayoutX(490);
		li6.setLayoutY(200);

		li7.setLayoutX(490);
		li7.setLayoutY(130);

		li8.setLayoutX(490);
```

```
            li8.setLayoutY(130);

            li9.setLayoutX(490);
            li9.setLayoutY(350);

            li10.setLayoutX(490);
            li10.setLayoutY(350);

            pa.setPadding(new Insets(0, 200, 350, 0));
            pa.setStyle("-fx-background-color: white;"
                                + "-fx-border-style: solid;"
                                + "-fx-border-color: gray;");
            pa.setLayoutX(350);
            pa.setLayoutY(50);
        }
}

}
}
```

Testfälle

Testfall: 1	**Titel: Spiel starten**
Datum: 25.11.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Hangman
Vorbedingung:	Spiel wurde programmiert
Beschreibung:	Spiel starten
Nachbedingun:	Das Spiel startet

Testfall: 2	**Titel: zufälliges Wort**
Datum: 25.11.2016	Art: Funktionale Anforderungen
Autor: Endrit Lena	Bezug: Hangman
Vorbedingung:	Wörter existieren damit diese ausgewählt werden können
Beschreibung:	Wort wird zufällig ausgewählt
Nachbedingun:	Es wird ein zufälliges Wort ausgewählt

Testfall: 3	**Titel: Wort unsichtbar**
Datum: 25.11.2016	Art: Funktionale Anforderungen
Autor: Endrit Lena	Bezug: Hangman
Vorbedingung:	Wort wird ausgewählt
Beschreibung:	Wort sollte nicht sichtbar sein
Nachbedingun:	Wort ist nicht sichtbar

Testfall: 4	**Titel: Wort als Striche**
Datum: 25.11.2016	Art: Funktionale Anforderungen
Autor: Endrit Lena	Bezug: Hangman
Vorbedingung:	Wort wird ausgewählt und nicht angezeigt
Beschreibung:	Buchstaben werden durch Striche ersetzt
Nachbedingun:	Wort wird als Striche angezeigt

Testfall: 5	**Titel: Buchstaben eingeben**
Datum: 25.11.2016	Art: Funktionale Anforderungen
Autor: Endrit Lena	Bezug: Hangman
Vorbedingung:	Ein TextField ist vorhanden
Beschreibung:	Buchstaben eingeben
Nachbedingun:	Buchstaben können eingegeben werden

Testfall: 6	**Titel: richtige Buchstaben anzeigen**
Datum: 25.11.2016	Art: Funktionale Anforderungen
Autor: Endrit Lena	Bezug: Hangman
Vorbedingung:	Wort muss ausgewählt sein und Buchstaben sollten eigegeben werden können
Beschreibung:	Einen richtigen Buchstaben eingeben und schauen ob diese ausgegeben werden
Nachbedingun:	Richtige Buchstaben werden ausgegeben

Testfall: 7	**Titel: Striche zeichnen durch falsche Eingabe**
Datum: 25.11.2016	Art: Funktionale Anforderungen
Autor: Endrit Lena	Bezug: Hangman
Vorbedingung:	Wort muss ausgewählt sein und Buchstaben sollten eigegeben werden können
Beschreibung:	Einen falschen Buchstaben eingeben und schauen ob ein strich gemalt wird
Nachbedingun:	Striche werden gezeichnet

Testfall: 8	**Titel: Wort gefunden**
Datum: 25.11.2016	Art: Funktionale Anforderungen
Autor: Endrit Lena	Bezug: Hangman
Vorbedingung:	Wort muss ausgewählt sein und Buchstaben sollten eigegeben werden können
Beschreibung:	Buchstaben eingeben und Wort herausfinden
Nachbedingun:	Spieler gewinnt

Testfall: 9	**Titel: Strichmännchen gezeichnet**
Datum: 25.11.2016	Art: Funktionale Anforderungen
Autor: Endrit Lena	Bezug: Hangman
Vorbedingung:	Wort muss ausgewählt sein und Buchstaben sollten eigegeben werden können
Beschreibung:	Falsche Buchstaben eingeben und Strichmännchen zeichnen lassen
Nachbedingun:	Spieler verliert

Fazit

Mir hatte es eigentlich Spass gemacht diese Spiele zu programmieren, dennoch konnte ich Minesweeper nicht vollenden. Dafür aber Hangman. Bei Minesweeper hatte ich zuerst das Problem, dass ich nicht wusste wie ich beginnen sollte. Nachdem ich aber einen Weg gefunden habe konnte ich noch viel erstellen. Das GUI per Code zu erstellen hat mir viel Freude bereitet. Bei meine Minesweeper kann schnell verlieren und sehr schwer gewinnen, da ich es nicht geschaft habe, dass in der Nähe der Mienen eine zahl angezeigt wird. Bei Hangman hatte ich ebenfalls viel Freude das GUI zu erstellen und ebenfalls den Spielablauf zu programmieren. Ausserdem hatte ich bei Hangman mehr Freude, weil dieses funktioniert im Gegensatz zu Minesweeper. Mit dem GitHub hatte ich keine Probleme da ich beim Starten vom GitHub ein Video fand welches mir zeigte wie ich die Daten hochladen konnte.