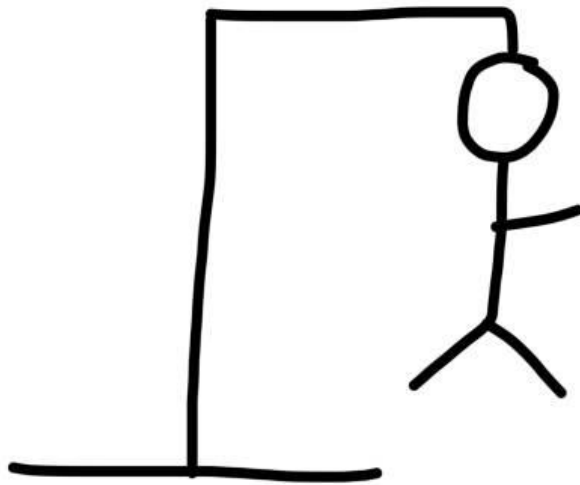


Dokumentation zum Projekt

GALGENMÄNNCHEN



H A N _ M A N

Projektarbeit von

Celine Gugel Nr. 2707682

Lena Steinbrink Nr. 5513041

Valentin Müller Nr. 5461128

Jannik Oßwald Nr. 1527904

und Thi Tuong Vy Nguyen Nr. 8610085

Inhalt

1.	Ideenfindung	3
2.	Spielerklärung.....	3
3.	Benötigte Funktionen	3
3.1	Auswahl Spielmodus.....	3
3.2	Buchstabeneingabe	3
3.3	Ausgabe der Konsole	4
3.4	Spielablauf	4
4.	Spielaufbau.....	5
4.1	Draw Hangman	5
4.2	Hangman Java.....	5
4.3	Spiel.java.....	7
5.	Zusammenarbeit	14

1. Ideenfindung

Um eine Idee davon zu bekommen, welches Spiel wir umsetzen wollen, haben wir uns zunächst selbst informiert, um uns anschließend für ein Spiel zu entscheiden. Zuerst kam uns hierbei das Spiel Flappy Bird in den Sinn. Da hierzu jedoch eine Grafik benötigt werden würde und wir vor allem das in den Vorlesungen angeeignete Wissen nutzen wollten, haben wir uns schlussendlich für das altbekannte Spiel Galgenmännchen entschieden.

2. Spielerklärung

Bei dem Spiel Galgenmännchen muss ein Begriff erraten werden. Es sind jedoch nur Striche in der Anzahl der zu erratenen Buchstaben des Wortes gegeben. Der Spieler muss nun einen Buchstaben eingeben. Ist der Buchstabe im Wort vorhanden, erscheint er an der passenden Stelle. Bei falsch geratenen Buchstaben bildet sich nach und nach eine Abbildung eines Galgens. Wurden zu viele Buchstaben falsch genannt, sodass der Galgen vervollständigt ist, so hat der Spieler das Spiel verloren. Sollte der Spieler jedoch vorher das Wort erraten, so kann er sich als Sieger verkünden.

3. Benötigte Funktionen

Damit wir mit der Programmierung beginnen können, muss zunächst festgelegt werden, wie wir das Spiel aufbauen wollen.

3.1 Auswahl Spielmodus

Wenn der Spieler das Spiel startet, soll er zunächst die Auswahl haben, ob er gegen den Computer spielen möchte oder ein Spiel mit eigener Worteingabe starten möchte. Bei einem Spiel gegen den Computer soll aus einem Array mit vorgegebenen Wörtern ein Wort zufällig ausgegeben werden und das Spiel startet. Wird ein Spiel mit eigener Worteingabe gewählt, so wird über die Scanner-Funktion ein Wort als String abgefragt. Damit bei dem Beginn des Spiels die Spielauswahl nicht mehr in der Konsole zu sehen ist, fügen wir 60 Leerzeichen ein. Es gibt auch noch einen dritten Modus, mit welchem man das Spiel verlassen kann.

3.2 Buchstabeneingabe

Wir legen eine neue Klasse Spiel. Somit kann man die Funktion zur Buchstabeneingabe programmieren mit den verschiedenen Bedingungen. Die erste Bedingung ist ein Char, damit nur der erste Buchstabe genommen wird. Die zwei anderen Bedingungen sind, dass das Wort zum einen mindestens zwei Zeichen hat und zum anderen, dass es keine Leerzeichen oder Sonderzeichen enthält. Zudem braucht man eine Funktion, die den Buchstaben mit dem gesuchten Wort vergleicht

und uns später diesen schon benutzten Buchstaben auch ausgibt. Außerdem benötigen wir ein Array mit den Buchstaben an der entsprechenden Stelle vom Wort für die Ausgabe in der Konsole.

3.3 Ausgabe der Konsole

Wenn die Eingabe des Spielers richtig ist, soll „Hurra! Deine Eingabe war richtig!“ auf der Konsole erscheinen. „Falsche Eingabe“ erscheint, wenn Eingabe falsch ist. Um den Spieler zu informieren, dass das Spiel soeben gestartet wurde, soll "Das Spiel Hangman wurde gestartet! " ausgegeben werden. Um den Nutzer die Auswahlmöglichkeiten zu zeigen wird "Spielmodus 1: Spiele gegen den Computer" und "Spielmodus 2: Gib ein eigenes Wort ein (Multiplayer)" angezeigt. Falls der Spieler das Spiel frühzeitig beenden möchte, sollte es eine Option geben, mit welcher er das Spiel beenden kann. Beispielsweise „Wenn du das Spiel beenden möchtest, kannst du exit eingeben.“ Hat der Spieler das Spiel nicht beendet und einen der beiden Spielmodi ausgewählt, erscheint „Es wurde Spielmodus 1/2 gewählt“, um dem Spieler mitzuteilen, in welchem Spielmodus er sich befindet. Als nächstes soll „Ein Buchstabe bitte:“ erscheinen, als Aufforderung für den Spieler ein Wort einzugeben. „Das Spiel wird beendet“, wenn der Spieler das Spiel beendet.

3.4 Spielablauf

Damit das Spiel am Ende problemlos läuft, haben wir uns Gedanken darüber gemacht, welche Funktionen, Variablen und Parameter sinnvoll wären.

Funktionen	Variablen/Parameter
„validateInput“ überprüft, ob die Eingabe im definierten Wertebereich liegt	char input String validLetters
„prüfeEingabe“, überprüft die Eingabe des zu erratenen Worts mindestens 2 Buchstaben hat und alle gültigen Buchstaben beinhaltet	boolean wert
„schonVersucht“ überprüft, ob der Buchstabe zuvor schonmal eingegeben wurde	char test
„contains“ überprüft, ob das gesuchte Wort den eingegebenen Buchstabe enthält	char in String pruefWert
„verglChar“, vergleicht den eingegebenen Buchstaben mit den schon versuchten Buchstaben und dem gesuchten Wort	char eingabe String gesWort
„getstatus“ zeigt den Zwischenstand des Spiels, gibt am Ende aus, ob der Spieler das Spiel gewonnen oder verloren hat	int fehlerAnzahl boolean gewonnen boolean verloren

„zeichneSpiel“, um die Zeichnung dann auch ausgeben zu können

4. Spielaufbau

4.1 Draw Hangman

Wir haben uns Gedanken darüber gemacht, wie wir den Galgen und die kleine Person darstellen möchten. Dazu haben wir zunächst einige Optionen ausprobiert und uns am Ende für die Folgende entschieden. Um diese nun darstellen zu können, haben wir zunächst eine Klasse namens DrawHangman erzeugt.

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    Scanner scanner = new Scanner(System.in);  
    int a=0;  
    while (a<13) {  
        System.out.println("get new  
number:");  
        a = scanner.nextInt();  
        switch (a) {
```

Wir haben eine switch-Anweisung angelegt und 11 cases erstellt. Bei jedem case kommt immer ein weiterer Strich für den Galgen und das Strichmännchen hinzu. Durch System.out.println werden die Striche abgebildet. Nach jedem break wird die Schleife komplett verlassen und nach der Schleife fortgesetzt. Ist der letzte case durchlaufen, so wird das Spiel durch scanner.close() beendet.

```
import java.util.Scanner;  
  
public class DrawHangman {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Scanner scanner = new Scanner(System.in);  
        int a=0;  
        while (a<13) {  
            System.out.println("get new number:");  
            a = scanner.nextInt();  
            switch (a) {  
                case 1: System.out.println("  _");  
                        System.out.println("/  \\");  
                        break;  
                case 2: System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("/  \\");  
                        break;  
                case 3: System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("/  \\");  
                        break;  
                case 4: System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("/  \\");  
                        break;  
                case 5: System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("/  \\");  
                        break;
```

```
                case 6: System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("/  \\");  
                        break;  
                case 7: System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("/  \\");  
                        break;  
                case 8: System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("/  \\");  
                        break;  
                case 9: System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("/  \\");  
                        break;  
                case 10: System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("/  \\");  
                        break;  
                case 11: System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("  |");  
                        System.out.println("/  \\");  
                        break;  
            }  
        }  
        scanner.close();  
    }  
}
```

4.2 Hangman Java

Wir haben eine Klasse Namens Hangman erstellt und einen Scanner angelegt. Zuerst soll es eine Abfrage des Spielmodi geben. Durch System.out.println angezeigt, dass das Spiel gestartet wurde. In der nächsten Zeile wird der „Spielmodus1: Gegen den Computer“ und eine Zeile darunter „Spielmodus 2: Gib ein eigenes Wort (Multiplayer)“ ausgegeben.

Damit es keine Fehlermeldung gibt, wird _____

Um eine Auswahl zwischen Spielmodus 1 und 2 zu gewährleisten, haben wir eine switch-Anweisung, also eine Mehr-Wege-Verzweigung angelegt. Der Ausdruck „spielmodus“ lässt sich somit in case 1 und 2 untergliedern. Case 1 steht für den Spielmodus 1. Es wird angezeigt „Es wurde Spielmodus 1 gewählt“. Wir haben eine Auswahl an Wörtern hinterlegt und eine zufällige Zahl für die Wortauswahl. Wichtig hierbei ist es, die Wörterlänge als Maximum anzugeben, da

Damit der Computer nun ein zufälliges Wort von den vorher angelegten Wörtern aussucht, haben wir eine random Methode angelegt.

Durch System.out.println(woerter[randomNumber]) wird nun eines der hinterlegten Wörter angezeigt. Die Variable input haben wir auf das zufällig gewählte Wort gesetzt, weil

In Case zwei ist der Spieler selbst für die Erstellung eines zu ratenden Wortes verantwortlich. Wieder wird ausgegeben, in welchem Spielmodus man sich befindet, jedoch wird der Spieler aufgefordert ein Wort einzugeben. Die Variable wird auf das eingegebene Wort gesetzt.

default -> System.out.println("Falsche Eingabe!")

Spiel gg = new Spiel(input);

char inChar;

input = "Start";

Damit der Spieler immer weiter raten kann, haben wir eine for-Schleife angelegt.

Zudem haben wir die abweisende while-Schleife angelegt, welche, so lange der Spieler nicht „exit“ schreibt, in Kraft tritt. Während der Spieler sich in dieser Schleife befindet, wird „Ein Buchstabe bitte“ beim Spieler angezeigt. Sollte der Spieler „exit“ schreiben, so wird „Du hast das Spiel beendet“ angezeigt und das Spiel wird durch „scan.close();“ geschlossen.

```
1 import java.util.Scanner;
2
3 public class Hangman {
4     public static void main(String[] args) {
5
6         Random random = new Random();
7
8         int randomNumber = random.nextInt(max + min) + min;
9         System.out.println(randomNumber);
10        System.out.println(woerter[randomNumber]);
11
12        // Die Variable input wird auf das zufällig gewählte Wort gesetzt
13        input = woerter[randomNumber];
14
15        case 2 -> {
16            // Spielmodus 2
17            System.out.println("Es wurde Spielmodus 2 gewählt.\n\n");
18            System.out.println("Gib ein zu erratendes Wort ein.");
19            // Die Variable input wird auf das eingegebene Wort gesetzt
20            input = scan.nextLine();
21
22            default -> System.out.println("Falsche Eingabe!");
23        }
24
25        Spiel gg = new Spiel(input);
26
27        char inChar;
28        input = "Start";
29
30        for(int i=0; i<60;i++) {
31            System.out.println();
32        }
33
34        while(!gg.getStatus().equals("exit")) {
35            System.out.print("Ein Buchstabe bitte: ");
36            input = scan.nextLine();
37            for(int i=0; i<30;i++) {
38                System.out.println();
39            }
40            inChar = input.charAt(0);
41            gg.verglChar(inChar);
42            gg.zeichneSpiel();
43        }
44    }
45}
```

```

49      System.out.println(woerter[randomNumber]);
50
51      // Die Variable input wird auf das zufällig gewählte Wort gesetzt
52      input = woerter[randomNumber];
53
54      case 2 -> {
55          ///////////////////////////////////////////////////////////////////
56          /////////////////////////////////////////////////////////////////// Spielmodus 2 ///////////////////////////////////////////////////////////////////
57          ///////////////////////////////////////////////////////////////////
58      }
59
60      System.out.println("Es wurde Spielmodus 2 gewählt.\n\n");
61
62      System.out.println("Gib ein zu erratendes Wort ein:");
63      // Die Variable input wird auf das eingegebene Wort gesetzt
64      input = scan.nextLine();
65
66      default -> System.out.println("Falsche Eingabe!");
67
68
69      Spiel gg = new Spiel(input);
70
71      char inChar;
72      input = "Start";
73
74      for(int i=0; i<60;i++){
75          System.out.println(i);
76
77          while((gg.getStatus()&&!(input.equals("exit")))) {
78              System.out.print("Tein Suchtabe bitte: ");
79              input = scan.nextLine();
80              for(int i=0; i<30;i++){
81                  System.out.println(i);
82
83                  inChar = input.charAt(0);
84                  gg.verglChar(inChar);
85                  gg.zeichneSpiel();
86
87              }
88              System.out.println("Du hast das Programm beendet");
89              scan.close();
90
91          }
92
93      }
94
95  }
96
97  }
98
99  }
100
101  }
102
103  }
104
105  }
106
107  }
108
109  }
110
111  }
112
113  }
114
115  }
116
117  }
118
119  }
120
121  }
122
123  }
124
125  }
126
127  }
128
129  }
130
131  }
132
133  }
134
135  }
136
137  }
138
139  }
140
141  }
142
143  }
144
145  }
146
147  }
148
149  }
150
151  }
152
153  }
154
155  }
156
157  }
158
159  }
160
161  }
162
163  }
164
165  }
166
167  }
168
169  }
170
171  }
172
173  }
174
175  }
176
177  }
178
179  }
180
181  }
182
183  }
184
185  }
186
187  }
188
189  }
190
191  }
192
193  }
194
195  }
196
197  }
198
199  }
200
201  }
202
203  }
204
205  }
206
207  }
208
209  }
210
211  }
212
213  }
214
215  }
216
217  }
218
219  }
220
221  }
222
223  }
224
225  }
226
227  }
228
229  }
230
231  }
232
233  }
234
235  }
236
237  }
238
239  }
240
241  }
242
243  }
244
245  }
246
247  }
248
249  }
250
251  }
252
253  }
254
255  }
256
257  }
258
259  }
260
261  }
262
263  }
264
265  }
266
267  }
268
269  }
270
271  }
272
273  }
274
275  }
276
277  }
278
279  }
280
281  }
282
283  }
284
285  }
286
287  }
288
289  }
290
291  }
292
293  }
294
295  }
296
297  }
298
299  }
300
301  }
302
303  }
304
305  }
306
307  }
308
309  }
310
311  }
312
313  }
314
315  }
316
317  }
318
319  }
320
321  }
322
323  }
324
325  }
326
327  }
328
329  }
330
331  }
332
333  }
334
335  }
336
337  }
338
339  }
340
341  }
342
343  }
344
345  }
346
347  }
348
349  }
350
351  }
352
353  }
354
355  }
356
357  }
358
359  }
360
361  }
362
363  }
364
365  }
366
367  }
368
369  }
370
371  }
372
373  }
374
375  }
376
377  }
378
379  }
380
381  }
382
383  }
384
385  }
386
387  }
388
389  }
390
391  }
392
393  }
394
395  }
396
397  }
398
399  }
400
401  }
402
403  }
404
405  }
406
407  }
408
409  }
410
411  }
412
413  }
414
415  }
416
417  }
418
419  }
420
421  }
422
423  }
424
425  }
426
427  }
428
429  }
430
431  }
432
433  }
434
435  }
436
437  }
438
439  }
440
441  }
442
443  }
444
445  }
446
447  }
448
449  }
450
451  }
452
453  }
454
455  }
456
457  }
458
459  }
460
461  }
462
463  }
464
465  }
466
467  }
468
469  }
470
471  }
472
473  }
474
475  }
476
477  }
478
479  }
480
481  }
482
483  }
484
485  }
486
487  }
488
489  }
490
491  }
492
493  }
494
495  }
496
497  }
498
499  }
500
501  }
502
503  }
504
505  }
506
507  }
508
509  }
510
511  }
512
513  }
514
515  }
516
517  }
518
519  }
520
521  }
522
523  }
524
525  }
526
527  }
528
529  }
530
531  }
532
533  }
534
535  }
536
537  }
538
539  }
540
541  }
542
543  }
544
545  }
546
547  }
548
549  }
550
551  }
552
553  }
554
555  }
556
557  }
558
559  }
560
561  }
562
563  }
564
565  }
566
567  }
568
569  }
570
571  }
572
573  }
574
575  }
576
577  }
578
579  }
580
581  }
582
583  }
584
585  }
586
587  }
588
589  }
590
591  }
592
593  }
594
595  }
596
597  }
598
599  }
600
601  }
602
603  }
604
605  }
606
607  }
608
609  }
610
611  }
612
613  }
614
615  }
616
617  }
618
619  }
620
621  }
622
623  }
624
625  }
626
627  }
628
629  }
630
631  }
632
633  }
634
635  }
636
637  }
638
639  }
640
641  }
642
643  }
644
645  }
646
647  }
648
649  }
650
651  }
652
653  }
654
655  }
656
657  }
658
659  }
660
661  }
662
663  }
664
665  }
666
667  }
668
669  }
670
671  }
672
673  }
674
675  }
676
677  }
678
679  }
680
681  }
682
683  }
684
685  }
686
687  }
688
689  }
690
691  }
692
693  }
694
695  }
696
697  }
698
699  }
700
701  }
702
703  }
704
705  }
706
707  }
708
709  }
710
711  }
712
713  }
714
715  }
716
717  }
718
719  }
720
721  }
722
723  }
724
725  }
726
727  }
728
729  }
730
731  }
732
733  }
734
735  }
736
737  }
738
739  }
740
741  }
742
743  }
744
745  }
746
747  }
748
749  }
750
751  }
752
753  }
754
755  }
756
757  }
758
759  }
760
761  }
762
763  }
764
765  }
766
767  }
768
769  }
770
771  }
772
773  }
774
775  }
776
777  }
778
779  }
780
781  }
782
783  }
784
785  }
786
787  }
788
789  }
790
791  }
792
793  }
794
795  }
796
797  }
798
799  }
800
801  }
802
803  }
804
805  }
806
807  }
808
809  }
810
811  }
812
813  }
814
815  }
816
817  }
818
819  }
820
821  }
822
823  }
824
825  }
826
827  }
828
829  }
830
831  }

```

4.3 Spiel.java

Wir haben eine Klasse Spiel erstellt. In dieser haben wir ein String gesWort angelegt, welcher _____.

Durch `int fehlerAnzahl = 0` _____. Damit _____
haben wir einen `char [] wortArr` angelegt.

```
private ArrayList <Character> schonVersucht = new ArrayList<Character>();
```

```
private boolean gewonnen = false;
```

```
private boolean verloren = false;
```

Mit `private static final String validLetters = "abcdefghijklmnopqrstuvwxyzäöü"` haben wir festgelegt, welche Buchstaben eingegeben werden können.

```
Spiel(String gesWort){  
    this.gesWort = gesWort.toLowerCase();  
    this.wortArr = new char[gesWort.length()];  
    for(int i = 0; i<wortArr.length; i++) {  
        wortArr[i] = '_';  
    }  
}
```

```
}
```

Damit der eingegebene Buchstabe mit dem gesuchten Wort und dem schon versuchten Buchstaben verglichen werden kann, haben wir `public void verglChar(char eingabe)` verwendet.

```
    eingabe = Character.toLowerCase(eingabe);
    if(validateInput(eingabe)) {
        if (schonVersucht(eingabe)) {
            System.out.println("Diesen Buchstaben hast du
schon versucht");

        }else if(contains(eingabe, gesWort)) {
            for(int i = 0; i<gesWort.length(); i++) {
                if(eingabe == gesWort.charAt(i)){
                    wortArr[i]=eingabe;
                }
            }
        }else{
            fehlerAnzahl++;
            schonVersucht.add(eingabe);
        }
    }else {
        System.out.println("Eingabe ist nicht gueltig, bitte erneut
versuchen");
    }
}
```

Mit `private boolean schonVersucht(char test)` wird überprüft, ob der Buschstabe zuvor schonmal eingegeben wurde.

```
for(int i=0; i < schonVersucht.size(); i++){
    if(test == schonVersucht.get(i)){
        return true;
    }
}
//schonVersucht.add(test);
```



```

        return false;
    }
}

```

Damit überprüft wird, ob das gesuchte Wort den Buchstaben enthält, wird `private static boolean contains(char in, String pruefWert)` benötigt.

```

        for(int i =0; i<pruefWert.length(); i++) {
            if(in == pruefWert.charAt(i)) {
                return true;
            }
        }
        return false;
    }
}

```

`private static boolean validateInput(char input)` { überprüft, ob die Eingabe im definierten Wertebereich liegt.

```

        input = Character.toLowerCase(input);
        if(contains(input,validLetters)) {
            return true;
        }else{
            return false;
        }
    }
}

```

Um den aktuellen Spielstand zu zeichnen, haben wir `public void zeichneSpiel()` in unseren Code integriert.

```

        System.out.println("\n");
        switch (fehlerAnzahl) {
            case 0: break;
            case 1: System.out.println(" ____");
                    System.out.println("/  \");
                    break;
            case 2: System.out.println(" |");

```

```

        System.out.println("  |");
        System.out.println("  |");
        System.out.println(" __|__");
        System.out.println("/  \");
        break;

case 3: System.out.println("  |'");
        System.out.println("  |");
        System.out.println("  |");
        System.out.println(" __|__");
        System.out.println("/  \");
        break;

case 4: System.out.println("  _____");
        System.out.println("  |'");
        System.out.println("  |");
        System.out.println("  |");
        System.out.println(" __|__");
        System.out.println("/  \");
        break;

case 5: System.out.println("  _____");
        System.out.println("  |'  `");
        System.out.println("  |");
        System.out.println("  |");
        System.out.println(" __|__");
        System.out.println("/  \");
        break;

case 6: System.out.println("  _____");
        System.out.println("  |'  `");
        System.out.println("  |");
        System.out.println("  |");
        System.out.println(" __|__");
        System.out.println("/  \");
        break;

```

```

case 7: System.out.println(" ____");
        System.out.println(" |'  `");
        System.out.println(" |  O");
        System.out.println(" |");
        System.out.println(" __|__");
        System.out.println("/  \");
        break;

case 8: System.out.println(" ____");
        System.out.println(" |'  `");
        System.out.println(" | /O");
        System.out.println(" |");
        System.out.println(" __|__");
        System.out.println("/  \");
        break;

case 9: System.out.println(" ____");
        System.out.println(" |'  `");
        System.out.println(" | /O\\");
        System.out.println(" |");
        System.out.println(" __|__");
        System.out.println("/  \");
        break;

case 10: System.out.println(" ____");
        System.out.println(" |'  `");
        System.out.println(" | /O\\");
        System.out.println(" | /");
        System.out.println(" __|__");
        System.out.println("/  \");
        break;

case 11: System.out.println(" ____");
        System.out.println(" |'  `");
        System.out.println(" | /O\\");
        System.out.println(" | /\\");

```

```

        System.out.println(" __|__");
        System.out.println("/  \");
        break;
    }
    System.out.println();
    for(int j =0; j<wortArr.length; j++) {
        System.out.print(wortArr[j] + " ");
    }System.out.println();
    System.out.println();
    System.out.print("Schonmal geratene Buschstaben: ");
    System.out.println(schonVersucht);
    System.out.println("\n");
}

public boolean getStatus(){
    if(fehlerAnzahl >= 11) { // wurden mehr als 11 Fehler gemacht gilt
        das Spiel verloren

        verloren = true;
    }
    gewonnen = true;
    for (int i = 0;i < gesWort.length();i++) {
        /*Wenn alle Buchstaben erraten wurden ist das spiel
        gewonnen.

        * Dies wird über eine for-Schleife geprüft die alle erratenen
        Buchstaben mit dem gesuchten wort abgleicht

        * wenn ein Buchstabe fehlt wird die gewonnen Variable
        wieder auf falsch gesetzt*/

        if(wortArr[i] != gesWort.charAt(i)) {
            gewonnen = false;
            break;
        }
    }

    /*sofern das Spiel noch nicht gewonnen oder verloren wurde wird
    der Status true ausgegeben

```

```

        * das spiel läuft also noch weiter*/
        if (gewonnen == false && verloren == false) {
            return true;

        }else if(gewonnen == true) { //wenn das Spiel gewonnen wurde wird
eine Siegernachricht ausgegeben und der Spielstatus auf falsch gesetzt

            System.out.println("Hurra du hast das Spiel gewonnen!!!!");
            return false;
        }else{
            /*wenn das Spiel verloren wurde wird eine Trostnachricht
ausgegeben und der Spielstatus auf falsch gesetzt

            *trifft keine der vorigen abfragen zu gilt das spiel als
verloren dies ist notwendig da java bedingte returns nicht anerkennt

            *und immer ein default das in jedem fall ausgelöst werden
kann braucht*/

            System.out.println("Leider hast du dieses Spiel verloren.
Vielleicht klappt es ja beim nächsten Mal.");

            System.out.println("Das zu erratende Wort war übrigens " +
gesWort + ".");

            return false;
        }
    }

    public static boolean prüfeEingabe(String Wort){ //Überprüft ein Wort ob es
mindestens 2 Buchstaben hat und alle buchstaben akzeptiert sind. Diese Überprüfung muss extern
durchgeführt werden.

        boolean wert = true;
        for(int i=0; i<Wort.length(); i++) {
            if(!validateInput(Wort.charAt(i))){
                wert = false;
                break;
            }
        }

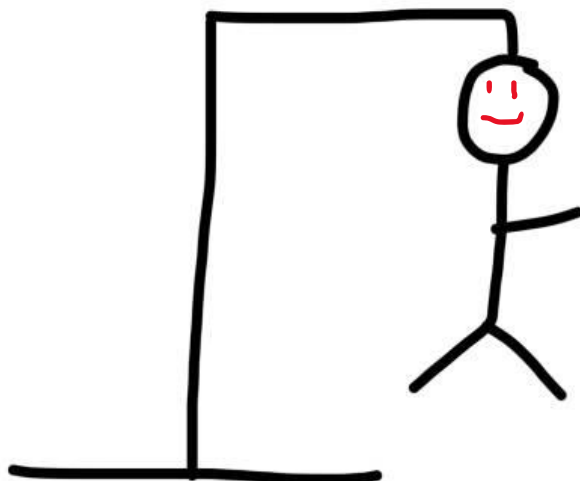
        if(Wort.length() < 2) {
            wert = false;
        }
    }

```

```
        return wert;  
    }  
}
```

5. Zusammenarbeit

Für den Start unseres Projektes haben wir uns zusammen Online getroffen. Hier haben wir Java/Eclipse mit GitHub verknüpft und jeder von uns hat einen eigenen Branch angelegt. Des Weiteren hat jeder erstmal über das Wochenende für sich Ideen gesammelt. Unser nächster Gruppentermin hat in Präsenz stattgefunden und wir haben uns alle zusammen und gemeinsam ausgetauscht. Hier haben wir auch festgelegt, wer an welcher Gruppenaufgabe mehr beteiligt sein möchte und wie unser weiteres Vorgehen erfolgt. Insgesamt hat unsere Gruppenarbeit gut funktioniert und wir kamen gut zurecht.



HANGMAN