Vertraulich

IPA2018

autor(en) : Cassandra Corrodi

dokumentennummer : CCT-FFF-XXXX

version : 1.0

status : Draft

quelle : Atos

dokumentendatum : 24 Oktober 2018­

­

­

anzahl der seiten : 1

owner : Cassandra Corrodi

© Copyright 2018, Atos AG Alle Rechte vorbehalten. Reproduktion von Teilen oder dem Gesamten ist ohne vorherige schriftliche Genehmigung des Urhebers untersagt. Für Fragen oder Anmerkungen zu diesem Dokument wenden Sie sich bitte an Atos, 0795893817.

# Inhaltsverzeichnis

[1 Teil 1: Umfeld und Ablauf 4](#_Toc528163134)

[1.1 Einleitung 4](#_Toc528163135)

[1.2 Beteiligte Personen 5](#_Toc528163136)

[1.2.1 Durchführung der IPA 5](#_Toc528163137)

[1.2.2 Kandidat 5](#_Toc528163138)

[1.2.3 Fachvorgesetzter 5](#_Toc528163139)

[1.2.4 Ausbildungsverantwortlicher 5](#_Toc528163140)

[1.2.5 Experte 5](#_Toc528163141)

[1.2.6 Zweit Experte 5](#_Toc528163142)

[1.3 Aufgabenstellung 6](#_Toc528163143)

[1.4 Projektorganisation 6](#_Toc528163144)

[1.5 Vorkenntnisse 6](#_Toc528163145)

[1.6 Vorarbeiten 6](#_Toc528163146)

[1.7 Firmenstandards 6](#_Toc528163147)

[1.8 Zeitplan 7](#_Toc528163148)

[1.9 Arbeitsprotokoll 8](#_Toc528163149)

[2 Teil 2: Projekt-Dokumentation 18](#_Toc528163150)

[2.1 Informieren 18](#_Toc528163151)

[2.2 Planen 18](#_Toc528163152)

[2.3 Entscheiden 18](#_Toc528163153)

[2.4 Realisieren 18](#_Toc528163154)

[2.5 Kontrollieren 18](#_Toc528163155)

[2.6 Auswerten 18](#_Toc528163156)

[3 Teil 3: Anhang 19](#_Toc528163157)

[3.1 Quellenverzeichnis 19](#_Toc528163158)

[3.2 Abbildungsverzeichnis 19](#_Toc528163159)

[3.3 Tabellenverzeichnis 19](#_Toc528163160)

[3.4 Glossar 19](#_Toc528163161)

[3.5 Programmcode 19](#_Toc528163162)

Änderungshistorie

| Version | Datum | Beschreibung | Autor(en) |
| --- | --- | --- | --- |
| 0.1 | 21.11.2018 | Dokumentstruktur erstellt, Einleitung verfasst, mit Planung begonnen, Zeitplan vorbereitet | Cassandra Corrodi |
| 0.2 |  |  | Cassandra Corrodi |
| 0.3 |  |  | Cassandra Corrodi |

# Teil 1: Umfeld und Ablauf

## Einleitung

Im Rahmen der Probe IPA, welche bei Atos stattfindet wird das Spiel Yatzy realisiert. Dieses Projekt ist eine Aufgabenstellung von der vergangenen Swiss Skills Meisterschaft. Die Aufgabe ist es nun, diesen Auftrag mit den technologien zu realisieren, mit denen auch die eche IPA geschrieben wird. Zudem muss die Applikation sauber getestet werden und es muss eine gute Dokumentation vorgewiesen werden.

## Kandidatin

Im Sommer 2015 habe ich meine Lehre zur Informatikerin Fachrichtung Applikationsentwicklung bei Siemens Schweiz gestartet. Im ersten Lehrjahr habe ich dort die Basisausbildung absolviert. Alle Informatiklehrlinge vom gleichen Lehrgang waren während dieses Jahres zusammen, und mussten theoretische wie auch praktische Arbeiten lösen.

Im zweiten Lehrjahr wurden alle in die Abteilungen verteilt. Ab dann habe ich bei Atos gearbeitet und konnte in den Abteilungen praktische Erfahrungen im Bereich C# und Angular sammeln. Zudem durfte ich auch eine Woche im First-Level-Support arbeiten, bei dem ich auch nützliche Erfahrungen mit Kunden sammeln und verschiedene Probleme lösen konnte.

Im Moment arbeite ich im SWP Team, und implementiere mit ASP.NET MVC.

## Beteiligte Personen

### Durchführung der Probe IPA

Lehrbetrieb: Atos AG  
Adresse: Freilagerstrasse 28  
PLZ/Ort: 8047 Zürich

### Kandidatin

Name: Cassandra Corrodi  
Adresse: Alte Bergstrasse 7  
PLZ/Ort: 8707 Uetikon am See  
Telefon: 079 517 38 89

### Fachvorgesetzter

Name: Patrick Maurer  
Adresse: Freilagerstrasse 28  
PLZ/Ort: 8047 Zürich  
E-Mail: patrick.maurer@atos.net

### Ausbildungsverantwortlicher

Name: Jonas Knoll  
Adresse: Freilagerstrasse 40  
PLZ/Ort: 8047 Zürich  
Telefon: 058 558 38 58  
E-Mail: jonas.knoll@siemens.com

### Experte

Name: Torben Dziuk  
Adresse: Freilagerstrasse 28  
PLZ/Ort: 8047 Zürich  
E-Mail: torben.dziuk@atos.net

## Aufgabenstellung

Das Ziel dieses Projektes ist es, eine Probe IPA zu schreiben, damit der Kandidat für die echte IPA welche im Frühjahr stattfinden wird vorbereitet ist. Die Aufgabe wird sein, eine Aufgabe von den Swiss Skills in vier Tagen mit dazugehöriger Dokumentation zu realisieren.

Bei den Swiss Skills wird hauptsächlich die Oberfläche der Applikation bewertet. Das heisst wenn man die Applikation bedient, sollte diese einfach funktionieren. Dabei spielt es keine Rolle ob Clean Code Prinzipien oder eine saubere Implementierung generell eingehalten wurde. Zudem muss die Applikation nicht einmal getestet werden. Bei den Swiss Skills muss auch keine Dokumentation geschrieben werden.

Bei der Probe IPA wird es anders sein, denn es kommt nicht nur auf die Oberfläche an. Die Applikation muss zuerst sauber geplant werden, bevor man mit der Implementation beginnt. Während der Implementation ist es wichtig, den Code so sauber wie möglich zu schreiben, denn dieser kommt in den Anhang am Schluss der Doku.

Zudem muss die Applikation auf Herz und Nieren geprüft werden, damit die Qualität gewährleistet werden kann. Um das zu erreichen, werden die Testfälle bereits am Anfang des Projektes definiert.

Am Ende dieser Probe IPA, sollte die Kandidatin besser vorbereitet sein für den Lehrabschluss, und sich schon an die Routine von so einem Projekt gewöhnen.

### Spielregeln Yatzy

Da auf dem Projektbeschrieb von den Swiss Skills nicht so gut beschrieben ist, wie das Spiel funktioniert, wurde im Internet recherchiert, wie das Spiel genau funktioniert. Das Spiel ist auch unter dem Namen Kniffel bekannt.

**Spielregeln**Bei Yatzy geht es darum, mit den 5 Würfeln die Kombinationen zu erwürfeln, die in der Spieltabelle vorgegeben sind. Gespielt werden 13 Runden, da es 13 verschiedene Kombinationen gibt. Jeder Spieler darf pro Runde dreimal hintereinander würfeln.

Nach jedem Wurf muss er entscheiden, welche Würfel er "stehen" lässt. Ein Würfel wird durch Klick auf das rechts neben ihm befindliche Kästchen fixiert. Es erscheint ein Häkchen, außerdem wird der Würfel abgeblendet. Einmal stehengelassene Würfel können später durch einen erneuten Klick wieder "befreit" werden.

Ist bereits der erste Wurf perfekt, muss natürlich nicht erneut gewürfelt werden. Spätestens nach dem 3. Wurf muss jedoch die entstandene Würfelkombination in eines der 13 Felder eingetragen werden. Danach ist der nächste Spieler an der Reihe.

Besonders zu Beginn gibt es zumeist mehrere Möglichkeiten, eine Würfelkombination in die Tabelle einzutragen. Ist jedoch ein Feld bereits belegt, so kann der betreffende Eintrag nicht mehr korrigiert werden. Lässt sich eine gewürfelte Kombination in keines der ersten 12 Felder eintragen, so hat der Spieler einmalig durch das Feld "Chance" die Möglichkeit, einfach die Summe der 5 Würfel einzutragen. Ist auch dieses Feld schon belegt, muss ein freies Feld "gestrichen" werden. In diesem Fall werden 0 Punkte vergeben.

Sind alle Felder belegt, ist das Spiel zu Ende. Gewonnen hat der Spieler mit den meisten Punkten.

(Quelle: http://www.2meter3.de/yatzy/rules.html)

### Projektantrag

Der Projektantrag ist schon genau vorgegeben. Folgende Punkte sind also während der Implementation dieses Projektes zu beachten.

**Projekt**

Die Aufgabe ist es Yatzy für zwei Spieler zu implementieren. Zudem muss eine Datenbank erstellt werden, bei der die Highscores gespeichert werden können.

**Gameplay**

Die Spieler würfeln abwechselnd mit fünf Würfeln. Nach jedem Wurf wählt der Spieler, welche Würfel er behalten und welche er erneut würfeln möchte. Ein Spieler kann einen Teil oder alle Würfel in einem Zug bis zu zweimal wiederholen. Der Spieler muss in jeder Runde eine Punktzahl in eine Score-Box eingeben. Das Spiel endet, wenn alle Punkteboxen verwendet werden. Der Spieler mit der höchsten Gesamtpunktzahl gewinnt das Spiel.

**Scoring**

Das Scoring ist schon genau auf der Aufgabenstellung von den Swiss Skills definiert. Man sieht auf den folgenden beiden Screenshots das Scoring. Das Scoring in diesem Projekt kann vom Spieler ziemlich selbstbestimmt gewählt werden, denn die Spieler entscheiden selber wie das Scoring ausfallen wird.

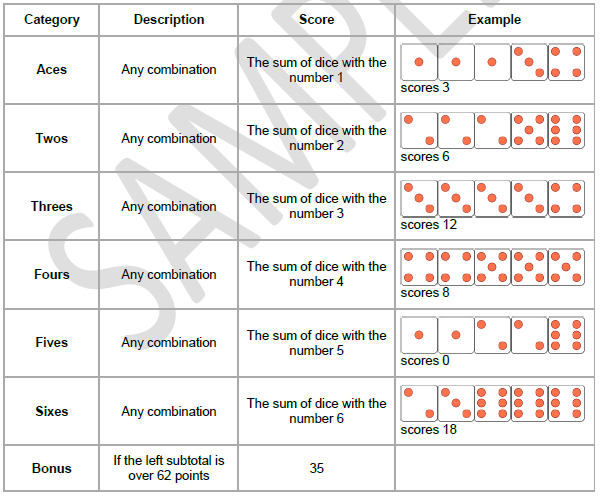


Abbildung 1: Scoring Zahlen einzeln

Auf der Abbildung sieht man das Scoring für die verschiedenen Würfelzahlen. Wie man sieht, werden diese, unabhängig davon, wie die Kombination ist, zusammengezählt. Jeder Spieler sollte wählen können, welche Würfel er behalten möchte, und welche er nochmals würfeln möchte.Die Spieler dürfen pro zug höchstens zwei Mal würfeln.

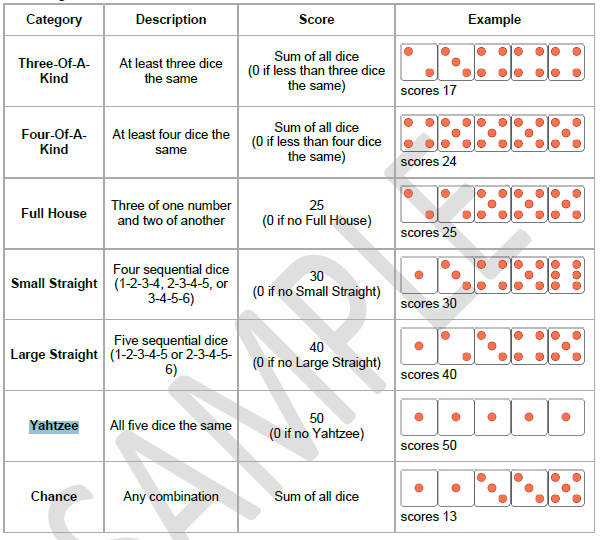
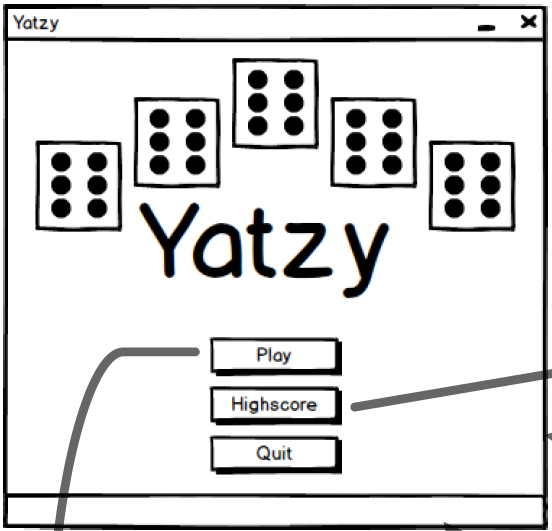


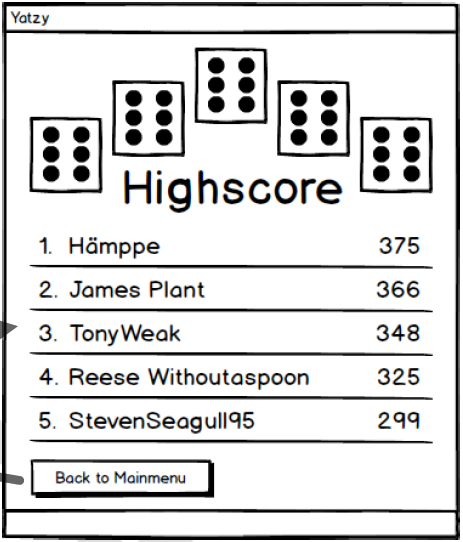
Abbildung 2: Scoring Zahlen Spezialfälle

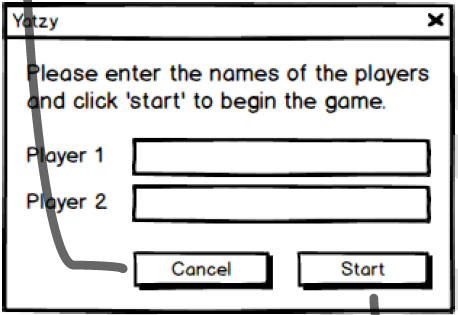
Auf dieser Abbildung sieht man, wie das Scoring ist, wenn man einen Spzialfall gewürfelt hat. Auf der rechten Spalte ist jeweils ein Beispiel ersichtlich, wie das Scoring je nach Fall ausfällt.

### Mockups

Das Frontend dieses Projektes ist bereits mehr oder weniger vorgegeben. Im Projektantrag der Swiss Skills befinden sich schon Mockups. Da ein Webprojekt erstellt wird, werden alle Fenster die gleiche grösse haben, und nicht so wie auf den Mockups alle anders sein.

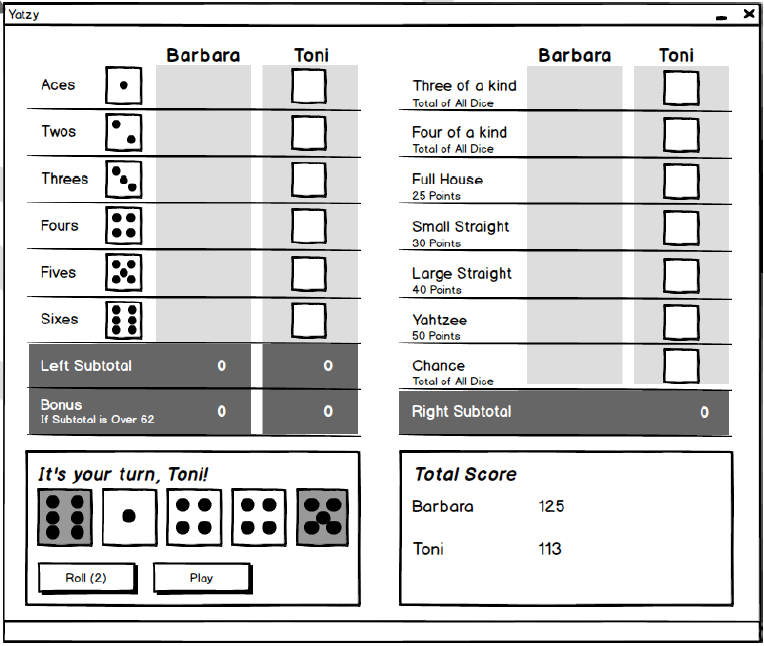
**Startpage**  
Auf der Startpage kann man ein Spiel starten, die Highscoreliste anschauen oder die Applikation beenden. Die Startpage sollte nur als Navigation dienen.

**Highscore List**auf der Highscore List sind die fünf besten Highscores aufgelistet. Von hieraus sollte es einen Button geben, mit dem man zurück auf das Hauptmenü, beziehungsweise Startpage geleitet wird. Diese Liste sollte automatisch aktualisiert werden, wenn jemand einen neuen Highscore erzielt, oder unter den neuen Top fünf gehört.

**Players**Wenn auf der Startpage auf «Play» geklickt wird, kommt man in eine View, bei der zwei Player erstellt werden müssen. Hier sollte eine Validierung der Inputs stattfinden, denn sie dürfen weder leer sein, noch mit Whitespaces gefüllt sein. Wenn eine fehlerhalte Eingabe gemacht wurde, darf kein neues Spiel gestartet werden und eine Error Message sollte aufploppen.  
Sonst können die Spieler einen beliebigen Namen wählen solange diese eine nicht zu lange Anzahl von Zeichen aufweist.

Wenn man auf den Start Button klickt, beginnt das Spiel. Bei Klick auf Cancel wird der Benutzer zurück auf das Startmenü geleitet.

**Game**

Hier sieht man das eigentliche Interface vom Spiel. Es hat zwei Spalten, die eine für das normale Scoring und die Andere für das spezielle Scoring.

Man sieht zwei Spalten. Für jeden Spieler eine. Der Spieler welcher gerade am Zug ist, sollte seine Spalte bearbeiten können. Unter der Spalte hat es einen Zähler welcher das Subtotal von der Spalte ausrechnet. Wenn ein Spieler einen Bonus gewonnen hat, wird dieser in einem Zähler unter dem andere Zähler zusammengerechnet.

Unter den beiden Spalten sieht man zwei Boxen. In der einen sieht man die Würfel und kann seine Würfe per Button kontrollieren. In der anderen sieht man den Total Score von beiden Spielern.

Wenn das Spiel beendet ist, wird man zurück auf die Startpage geleitet.

### Anforderungen des Arbeitgebers

### Technologiestack

IDE: Visual Studio 2017

Datenbank: SQL Server Management Studio

Frontend: Razor

Backend: ASP:NET MVC

Versionierung: GitHub undTortoiseGit

## Projektorganisation

### Projektmanagement Methode

Viele Projektmanagement Methoden sind eher darauf ausgerichtet, dass man diese im Team verwenden kann. Deswegen habe ich mich entschieden die Probe IPA mit IPERKA zu realisieren, da es sich sehr gut dafür eignet, wenn man alleine arbeitet.  
Alle Arbeitsschritte können gut in die verschiedenen Bereiche des IPERKAs geteilt werden, denn die verschiedenen Phasen sind so genau beschrieben, dass fast nichts schief gehen kann.

Ziel der Planung ist es, einen Lösungsweg für das Endprodukt zu erstellen. Hier müssen die Mittel und Arbeitsschritte genau geplant werden. Zudem muss hier definiert werden, wann was gemacht werden soll.

Zum Schluss wertet man aus, wie die Arbeit gegangen ist. Welche Schritte sind mir gelungen? Welche eher nicht? Wichtig dabei ist es, Erkenntnisse für das nächste Projekt zu sammeln.

Das ist der zeitaufwändigste Schritt von allen, denn die geplanten Arbeitsschritte müssen ausgeführt werden. Zudem muss man auch das gemachte dokumentieren.

Wenn die Arbeit erledigt hat, muss man diese kontrollieren. Man muss schauen ob alles erledigt wurde, und die Qualität des Programmes gewährleistet ist.

Nach der Planung muss definitiv entschieden werden, wie die Arbeitsschritte ausfallen werden. Dazu gehören auch Technologien und Wege, welche nicht schon im Voraus definiert wurden.

Man muss den Auftrag verstehen und sich ein Bild des zu ereichenden Ziels machen. Es müssen Abklärungen gemacht werden bis der Auftrag verstanden worden ist.

Abbildung 3: IPERKA

## Vorkenntnisse

Folgende Vorkenntnisse wurden in der Lehre gesammelt:

* ASP.NET MVC (6 Monate)
* SQL (Erfahrungen aus dem Basislehrjahr)
* Visual Studio mit ReSharper (3 Jahre)
* Git & TortoiseGit (2,5 Jahre)
* Angular Typescript (1 Jahr)
* Visual Studio Code (1 Jahr)
* C# (1 Jahr)
* jQuery (3 Monate)

## Vorarbeiten

Um während des Projektes nicht in Stress zu geraten, wurden ein paar Vorarbeiten geleistet. Diese sind auf der folgenden Liste aufgelistet.

* Die grobe Gliederung dieser Dokumentation wurde schon erstellt. Jede IPA Dokumentation ist anders, doch es gibt Bulletpoints welche alle Dokumentationen brauchen.
* Die IPERKA Abbildung wurde schon erstellt, damit während der Arbeit nicht so viele Grafiken erstellt werden müssen
* Text über Kandidatin wurde geschrieben.
* Liste der Beteiligten Personen wurde erstellt.

## Firmenstandards

Bei diesem Projekt stellt die Firma keine besonderen Standards. Tools und Technologie konnte selbst bestimmt werden.  
Atos hat ein Standard Template für Dokumentationen. Dieses wurde verwendet um die Dokumentation dieses Projektes zu schreiben.

## Zeitplan

### Meilensteine

Für das Projekt wurden Meilensteine festgelegt, welche die signifikanten Fortschritte eines Projektes darstellen. Da in diesem Projekt mit IPERKA gearbeitet wird, werden die Meilensteine nach jedem IPERKA Schritt gesetzt.

|  |  |  |
| --- | --- | --- |
| Schritt | Meilenstein | Datum/Zeit |
| I | Ich beginne die Arbeit und habe mich über verschiedene wesentliche Punkte informiert. Nach dem Informieren muss mir klar sein, was alles vorhanden ist, und was ich zu berücksichtigen habe während des ganzen Projektes. | 21.11.2018 um 16:30 Uhr |
| P | Ziel der Planung ist es, dass schon im Voraus überlegt wird, wie das Programm realisiert werden sollte. Die Planung ist ein Schritt, welcher wichtiger ist als die meisten denken. Denn hier ist das entscheidende, dass man während der Realisation nicht mehr so viel überlegen muss.  In diesem Schritt möchte ich Die Softwarearchitektur, die Views und die Datenbank geplant haben. | 22.11.2018 um 17:00 Uhr |
| E | Vor der Realisation sollte entschieden werden, wie die nicht vorgegebenen Schritte des Projektes vorgenommen werden sollten. Hier sollte dokumentiert sein, wie die Entscheidung gefallen ist. | 23.11.2018 um 12:00 Uhr |
| R | Bei der Realisation sollte das gesamte Projekt realisiert werden. Der Meilenstein ist zu Ende, wenn das Projekt ganz fertiggestellt ist. | 28.11.2018 um 12:00 |
| K | Nachdem das Projekt realisiert ist, muss es kontrolliert werden, weil die Qualität des Programmes gewährleistet werden muss. Ziel dieses Meilensteines ist es, Die geplanten Tests durchgeführt zu haben und die Ergebnissen sollten notiert werden. | 28.11.2018 um 17:00 |
| A | Ich schreibe die Reflexion und erstelle den Anhang. Somit wird die Dokumentation abgeschlossen, was auch der letzte Meilenstein dieses Projektes ist. | 28.11.2018 um 12:00 Uhr |

Tabelle 1: Meilensteine

## Arbeitsprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
|  | Tag 1 (1/2 Tag) 21.11.2018 | |  |
| Soll | | Ist | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
| Positives (Was lief gut?) | | | |
| Heute habe ich damit begonnen das Projekt zu realisieren. Dabei habe ich die Dokumentstruktur mit allen nötigen Kapiteln erstellt. Zudem habe ich mich über das Projekt informiert. Dabei habe ich die Aufgabenstellung und das Spiel Yatzy genau studiert. | | | |
| Negatives (Wo gab es Probleme?) | | | |
| Heute sind keine Probleme aufgetreten | | | |
| Zeitplanung | | | |
| I | | | |
| Beanspruchte Hilfestellung | | | |
|  | | | |
| Nächste Schritte | | | |
|  | | | |

Tabelle 2: Arbeitsjournal Tag 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | Tag 2 22.11.2018 | |  |
| Soll | | Ist | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
| Positives (Was lief gut?) | | | |
| Heute habe ich mich wieder hauptsächlich mit dem Dokumentieren befasst. Dabei konnte ich die ersten drei Schritte des IPERKAS fertigstellen. Das heisst ich habe mich informiert, geplant und entschieden wie das Projekt umgesetzt werden sollte.  Zudem konnte ich schon eine Visual Studio Solution vorbereiten, um | | | |
| Negatives (Wo gab es Probleme?) | | | |
|  | | | |
| Zeitplanung | | | |
|  | | | |
| Beanspruchte Hilfestellung | | | |
|  | | | |
| Nächste Schritte | | | |
|  | | | |

Tabelle 3: Arbeitsjournal Tag 2

|  |  |  |  |
| --- | --- | --- | --- |
|  | Tag 3 23.11.2018 | |  |
| Soll | | Ist | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
| Positives (Was lief gut?) | | | |
|  | | | |
| Negatives (Wo gab es Probleme?) | | | |
|  | | | |
| Zeitplanung | | | |
|  | | | |
| Beanspruchte Hilfestellung | | | |
|  | | | |
| Nächste Schritte | | | |
|  | | | |

Tabelle 4: Arbeitsjournal Tag 3

|  |  |  |  |
| --- | --- | --- | --- |
|  | Tag 4 27.11.2018 | |  |
| Soll | | Ist | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
| Positives (Was lief gut?) | | | |
|  | | | |
| Negatives (Wo gab es Probleme?) | | | |
|  | | | |
| Zeitplanung | | | |
|  | | | |
| Beanspruchte Hilfestellung | | | |
|  | | | |
| Nächste Schritte | | | |
|  | | | |

Tabelle 5: Arbeitsjournal Tag 4

|  |  |  |  |
| --- | --- | --- | --- |
|  | Tag 5 (1/2 Tag) 28.11.2018 | |  |
| Soll | | Ist | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
| Positives (Was lief gut?) | | | |
|  | | | |
| Negatives (Wo gab es Probleme?) | | | |
|  | | | |
| Zeitplanung | | | |
|  | | | |
| Beanspruchte Hilfestellung | | | |
|  | | | |
| Nächste Schritte | | | |
|  | | | |

Tabelle 6: Arbeitsjounal Tag 5

# Teil 2: Projekt-Dokumentation

## Informieren

## Planen

## Entscheiden

## Realisieren

## Kontrollieren

## Auswerten

### Reflexion

### Optimierungsbedarf

### Schlusswort

# Teil 3: Anhang

## Quellenverzeichnis

|  |  |
| --- | --- |
| Was? | Quelle |
|  |  |

Tabelle 7: Quellenverzeichnis

## Abbildungsverzeichnis

[Abbildung 1: Scoring Zahlen einzeln 6](#_Toc530643995)

[Abbildung 2: Scoring Zahlen Spezialfälle 7](#_Toc530643996)

[Abbildung 3: IPERKA 8](#_Toc530643997)

## Tabellenverzeichnis

[Tabelle 1: Meilensteine 9](#_Toc530643976)

[Tabelle 2: Arbeitsjournal Tag 1 11](#_Toc530643977)

[Tabelle 3: Arbeitsjournal Tag 2 12](#_Toc530643978)

[Tabelle 4: Arbeitsjournal Tag 3 13](#_Toc530643979)

[Tabelle 5: Arbeitsjournal Tag 4 14](#_Toc530643980)

[Tabelle 6: Arbeitsjounal Tag 5 15](#_Toc530643981)

## Glossar

|  |  |
| --- | --- |
| Begriff | Erklärung |
| Glossar | Im Glossar werden Begriffe eingefügt, welche erklärt oder übersetzt werden. |

Tabelle 8: Glossar

## Programmcode

Hier ist der Programmcode welcher in den fünf Tagen Projektzeit geschrieben wurde. Der Code wurde nach den verschiedenen Projekten getrennt.