

# Pokémeme

Projektleitung: Mirio Eggmann  
Betreuer: Michael Beck  
Datum: 14.04.2015  
Firma: Die Schweizerische Post

## 1 Abstract (Kurzbeschreibung)

Als Java-Projekt werden wir ein Pokémon ähnliches Spiel Programmieren. Anstatt die richtigen Pokémon Figuren zu verwenden, werden wir Memes verwenden. Dies sind sozusagen Köpfe von verschiedenen Personen die abgeändert wurden. Ein bekanntes Meme ist das Trollface!

Unser Spiel besteht aus einem Multiplayermodus, indem man gegeneinander antreten kann. Vor dem Spiel kann man seine Pokémemes auswählen. Jede Figur hat vier Attacken. Ausserdem kann die Figur geheilt oder durch eine andere Figur getauscht werden, wenn sie wenig leben hat.



## Inhaltsverzeichnis

1	Abstract (Kurzbeschreibung).....	1
1.1	Einführung.....	5
1.2	Zielgruppe.....	5
1.3	Use-Cases.....	5
1.3.1	Use Case - Attacke ausführen.....	6
1.3.2	Use Case - Pokémeme Team zusammenstellen.....	7
1.3.3	Use Case - Pokémeme heilen .....	8
1.3.4	Use Case - Gegeneinander kämpfen.....	9
1.3.5	Use Case - Hilfe aufrufen .....	10
1.3.6	Use Case - Pokémeme auswechseln.....	11
1.3.7	Use Case - Zurück zum Hauptmenü.....	12
1.4	Funktionale Anforderungen.....	13
1.4.1	Pokémeme auswählen.....	13
1.4.2	Lebensanzeige.....	13
1.4.3	Attacke.....	13
1.4.4	Tauschen .....	13
1.4.5	Heilen .....	13
1.4.6	Hilfe.....	13
1.4.7	GUI.....	13
1.5	Nichtfunktionale Anforderungen .....	13
1.5.1	Kompatibilität .....	13
1.5.2	Benutzbarkeit .....	13
1.5.3	Performance.....	13
1.6	Lieferumfang.....	14
1.6.1	Applikation.....	14
1.6.2	Dokumentation.....	14
2	Projektorganisation .....	15
2.1	Organigramm.....	15
2.2	Zuständigkeiten der Mitarbeiter .....	15
2.2.1	Projektleiter.....	15
2.2.2	Externer Berater.....	15
2.2.3	Projektmitarbeiter .....	15
3	GUI Design.....	16
3.1	MockUp 1 - Menü.....	16
3.2	MockUp 2 - Team wählen.....	17
3.3	MockUp 3 - Kampfsystem .....	18
3.4	MockUp 4 - Hilfedesign .....	19
4	Datenhaltung.....	20
4.1	ERM.....	20



5	Arbeitspakete .....	20
5.1	Arbeitspaket A – Projektplanung.....	20
5.2	Arbeitspaket B – Projektantrag erstellen.....	21
5.3	Arbeitspaket C – Programm Backend.....	22
5.4	Arbeitspaket D – Programm Frontend.....	23
5.5	Arbeitspaket E – Programm Frontend und Backend zusammenführen.....	24
5.6	Arbeitspaket F – Testing .....	25
5.7	Arbeitspaket G – Arbeitsumgebung dokumentieren .....	26
5.8	Arbeitspaket H – Präsentation.....	27
5.9	Zeitplanung.....	28
5.9.1	GANTT .....	28
6	Testcases.....	29
6.1	Beschreibung der Testumgebung.....	29
6.2	Liste der Testcases .....	29
6.2.1	Test Case 1 – Pokémeme heilen.....	29
6.2.2	Test Case 2 – Beenden der Applikation .....	29
6.2.3	Test Case 3 – Zurück zum Hauptmenü .....	30
6.2.4	Test Case 4 – Pokémeme auswählen.....	30
6.2.5	Test Case 5 – Hilfe aufrufen.....	31
6.2.6	Test Case 6 – Gegeneinander kämpfen.....	31
6.2.7	Test Case 7 – Pokémeme Daten.....	32
6.2.8	Test Case 8 – Pokémeme wechseln .....	32
6.2.9	Test Case 9 – Namen auswählen .....	33
6.2.10	Test Case 10 – Start.....	33

## Abbildungsverzeichnis

Abbildung 1 Use-Cases .....	5
Abbildung 2: Organigramm der Projektorganisation .....	15
Abbildung 3 MockUp 1 - Menü.....	16
Abbildung 4 MockUp 2 – Team wählen.....	17
Abbildung 5 MockUp 3 - Kampfsystem.....	18
Abbildung 6 MockUp 4 - Hilfedesign .....	19
Abbildung 7 Gantt .....	28

## Tabellenverzeichnis

Tabelle 1 Arbeitspaket A- Projektplanung.....	20
Tabelle 2 Arbeitspaket B - Projektantrag erstellen.....	21
Tabelle 3 Arbeitspaket C - Programm Backend .....	22
Tabelle 4 Arbeitspaket D - Programm Frontend.....	23
Tabelle 5 Arbeitspaket E – Programm Frontend und Backend zusammenführen .....	24
Tabelle 6 Arbeitspaket F - Testing .....	25
Tabelle 7 Arbeitspaket G – Arbeitsumgebung dokumentieren.....	26
Tabelle 8 Arbeitspaket H - Präsentation.....	27
Tabelle 9 Test Case 1 - Pokémeme heilen.....	29
Tabelle 10 Test Case 2 - Beenden der Applikation.....	29
Tabelle 11 Test Case 3 - Zurück zum Hauptmenü .....	30
Tabelle 12 Test Case 4 - Pokémeme auswählen.....	30
Tabelle 13 Test Case 5 - Hilfe aufrufen .....	31
Tabelle 14 Test Case 6 - Gegeneinander kämpfen.....	31
Tabelle 15 Test Case 7 - Pokémeme Daten.....	32
Tabelle 16 Test Case 8 - Pokémeme wechseln.....	32
Tabelle 17 Test Case 9 - Namen auswählen.....	33
Tabelle 18 Test Case 10 - Start.....	33



## Anforderungsanalyse

### 1.1 Einführung

Wir werden als Team ein kleines Java-Spiel programmieren. Dabei handelt es sich um eine abgespeckte Version von Pokémon. Das Spiel beinhaltet einen Multiplayermodus. Dort spielt man gegen einen anderen Spieler auf einem Computer. Natürlich kann man vor dem Spiel sein Team selber zusammenstellen.

Zum Aufbau wir haben uns für 4 verschiedene Layouts entschieden. Das erste ist um sein Pokémeme Team aufzustellen, das nächste für einen Kampf, des Weiteren erstellen wir noch eines um Pokémemes auszuwählen und während dem Kampf zu tauschen, zum Schluss fügen wir noch ein kleines Hilfe Fenster ein, damit man weiss, wie das Programm bedient werden kann.

Dieses Spiel könnte falls noch genügend Zeit übrig ist, extrem erweitern werden. Man könnte den Multiplayermodus nicht mehr nur auf Computer spielen, sondern via Netzwerk über zwei Computer. Oder man könnte auch für die ganzen Eigenschaften der Pokémemes eine Datenbank einbinden.

### 1.2 Zielgruppe

Die Zielgruppe dieses Spieles sind begeisterte Fans von Pokémon, wie auch kleinere Kinder oder Jugendliche.

### 1.3 Use-Cases

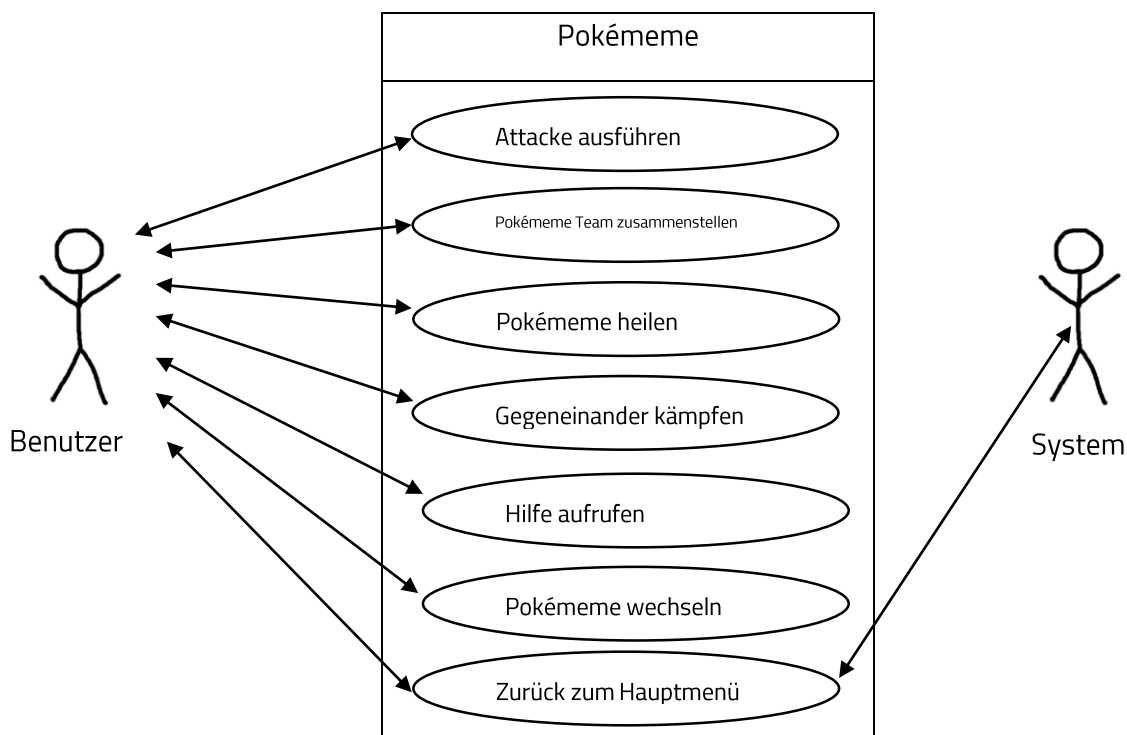


Abbildung 1 Use-Cases

### 1.3.1 Use Case - Attacke ausführen

**Aktor:** Benutzer

**Vorbedingungen:**

- Das Programm muss in Betrieb sein
- Ein Multiplayer Kampf muss ausgewählt werden.
- Das Pokémeme Team muss zusammengestellt sein  
UC: Pokémeme Team zusammenstellen
- Der Spieler befindet sich im Kampf mit dem Gegner
- Der Benutzer muss im Kampf Menü sein

**Hauptszenario**

1. Er drückt auf die gewünschte Attacke.
2. System gibt eine Meldung aus, welche Attacke ausgeführt wird und wer wen angreift.
3. Das Pokémeme führt die Attacke aus und trifft.
4. System zieht dem gegnerischen Pokémeme den Unterschied zwischen dem Angriff vom angreifenden und der Verteidigung vom Gegner ab (falls die Verteidigung höher ist werden trotzdem 5 Lebenspunkte abgezogen).

**Alternativer Ablauf**

4a) Das Pokémeme führt die Attacke aus, aber sie geht daneben.

4a1) Das System gibt eine Meldung aus, dass die Attacke daneben ging.

4b) Das Pokémeme führt die Attacke aus und erzielt einen Volltreffer und zieht somit mehr Leben ab.

4b1) Das System meldet, dass man einen Volltreffer erzielt hat.

4b2) System zieht dem gegnerischen Pokémeme den Unterschied zwischen dem Angriff vom angreifenden und der Verteidigung vom Gegner ab (falls die Verteidigung höher ist werden trotzdem 5 Lebenspunkte abgezogen) und multipliziert dies noch mit dem Volltreffer.

**Nachbedingung:**

- Dem gegnerischen Pokémeme wurde Leben abgezogen.

### 1.3.2 Use Case – Pokémeme Team zusammenstellen

**Aktor:** Benutzer

**Vorbedingungen:**

- Das Programm muss in Betrieb sein
- Ein Multiplayer Kampf muss ausgewählt werden.

**Hauptszenario**

1. Der Benutzer wählt die gewünschten Pokémemes und fügt diese zu seinem Team hinzu.
2. Das System fügt die gewählten Pokémemes einer Team Liste hinzu.
3. Der Benutzer bestätigt seine Wahl und drückt auf Start.
4. System nimmt die erstellte Pokémeme Liste und übergibt diese dem Spieler.
5. Das System leitet den Benutzer zum Kampf weiter.

**Alternativer Ablauf**

- 3a) Der Benutzer wählt ein falsches Pokémeme und fügt dies zu seinem Team hinzu.
- 3a1) Das System fügt das gewählte Pokémeme der Team Liste hinzu.
- 3a2) Der Benutzer löscht das falsche Pokémeme.
- 3a3) Das System löscht das falsche Element aus der Liste.
- 3a4) Der Benutzer bestätigt seine Wahl und drückt auf Start
- 3a5) Das System leitet den Benutzer zum Kampf weiter.

**Nachbedingung:**

- Das Team für den Multiplayer Kampf wurde zusammengestellt und der Benutzer ist bereit für den Kampf.

### 1.3.3 Use Case - Pokémeme heilen

**Aktor:** Benutzer

**Vorbedingungen:**

- Das Programm muss in Betrieb sein
- Ein Multiplayer Kampf muss ausgewählt werden.
- Das Pokémeme Team muss zusammengestellt sein  
UC: Pokémeme Team zusammenstellen
- Der Spieler befindet sich im Kampf mit dem Gegner

**Hauptszenario**

1. Der Benutzer drückt auf heilen
2. Das System nimmt dies entgegen und gibt dem Pokémeme, welches gerade im Spiel ist und von dem ist, welcher heilen eingesetzt hat 25 Lebenspunkte als zusätzliches Leben (geht nicht über die Maximale Anzahl Leben) zurück.

**Alternativer Ablauf**

-

**Nachbedingung:**

- Das Pokémeme vom Spieler welcher heilen eingesetzt hat, bekommt 25 Lebenspunkte zurück.



#### 1.3.4 Use Case - Gegeneinander kämpfen

**Aktor:** Benutzer

**Vorbedingungen:**

- Das Pokémeme Team muss zusammengestellt sein  
UC: Pokémeme auswählen
- Ein Multiplayer Kampf muss gestartet werden.
- Eine Attacke ist ausgewählt

**Hauptszenario**

1. Das Pokémeme greift das andere an.
2. Das System gibt eine Mitteilung

**Alternativer Ablauf**

1a) Alle Gegner sind bereits schon tot.

1a1) Das System gibt eine Meldung aus, dass der entsprechende Spieler gewonnen hat.

1a2) Der Kampf wird beendet.

1a3) Das Startmenü erscheint.

**Nachbedingung:**

- Das neue Pokémeme kämpft nun.

### 1.3.5 Use Case - Hilfe aufrufen

**Aktor:** Benutzer

**Vorbedingungen:**

- Das Programm ist gestartet.

**Hauptszenario**

5. Der Benutzer klickt auf den Hilfe Button
6. System zeigt die Anleitung an.

**Alternativer Ablauf**

2a) Er hat sich verklickt

2a1) Der Benutzer klickt auf den Home Button

**Nachbedingung:**

- Der Benutzer kann die Anleitung lesen.

### 1.3.6 Use Case - Pokémeme auswechseln

**Aktor:** Benutzer

**Vorbedingungen:**

- Das Pokémeme Team muss zusammengestellt sein  
UC: Pokémeme auswählen
- Ein Multiplayer Kampf muss gestartet werden.

**Hauptszenario**

1. Der Benutzer geht ins Auswechsel-Menü.
2. Er drückt auf das gewünschte Pokémeme.
3. System gibt eine Meldung aus, welches Pokémeme mit welchem ausgetauscht wurde.
4. Das Pokémeme wird gewechselt.
5. System zeigt die richtige Lebensanzeige an und speichert die neuen Werte.

**Alternativer Ablauf**

2a) Alle anderen Teammitglieder sind bereits schon tot.

2a1) Das System gibt eine Meldung aus, dass er keine Pokémemes zur Verfügung hat.

**Nachbedingung:**

- Das neue Pokémon kämpft nun.

### 1.3.7 Use Case - Zurück zum Hauptmenü

**Aktor:** Benutzer, System

**Vorbedingungen:**

- Ein anderes Layout als das Start-/Hauptmenü ist ausgewählt

**Hauptszenario**

1. Auf den Zurück Button klicken
2. System führt den Benutzer zurück zum Hauptmenü.

**Alternativer Ablauf**

-

**Nachbedingung:**

- Der Benutzer ist nun wieder im Hauptmenü.

## 1.4 Funktionale Anforderungen

### 1.4.1 Pokémeme auswählen

Beim Auswählen der Pokémemes kann man auf entsprechende Buttons klicken und somit diese zum Team hinzufügen. Auf den Buttons sind die Bilder der Pokémeme's.

### 1.4.2 Lebensanzeige

Während dem Kampf sieht man die beiden Pokémemes und eine Lebensanzeige. Damit kann man einschätzen ob man heilen oder angreifen soll.

### 1.4.3 Attacke

Die Attacke zieht am gegnerischen Pokémeme leben ab. Dies wird auch mit der Lebensanzeige dargestellt.

### 1.4.4 Tauschen

Wenn der Spieler sein Pokémeme austauscht, kommt das andere Pokémeme zum Hervorschein. Dabei wird auch die Lebensanzeige verändert, falls es mehr oder weniger leben hat.

### 1.4.5 Heilen

Wenn der Spieler sein Pokémeme heilt, füllt es die Lebensanzeige um einen gewissen Betrag auf.

### 1.4.6 Hilfe

Die Hilfe kann der Spieler im Menü aufrufen, indem er auf den Hilfe-Knopf klickt. Das Layout wird sich dabei verändern.

### 1.4.7 GUI

Das Fenster vom GUI soll sich vergrößern und verkleinern lassen.

## 1.5 Nichtfunktionale Anforderungen

### 1.5.1 Kompatibilität

Das Spiel sollte grundsätzlich Plattform unabhängig sein, jedoch benötigt man Java JRE. Getestet wurde das Programm nur auf einem Windows PC und ist auch nur dafür gedacht.

### 1.5.2 Benutzbarkeit

Das Programm beinhaltet eine kleine Hilfe, die über einen Hilfe Button erreichbar ist. Sie beinhaltet die wichtigsten Elemente und ihre Funktion.

### 1.5.3 Performance

Dieses Spiel nimmt nicht viel Performance in Anspruch. Da es nur eine 2D Grafik und fast keine Animationen hat.

## 1.6 Lieferumfang

Am Ende des Projekts müssen folgende Produkte abgegeben werden:

### 1.6.1 Applikation

Am Ende soll ein kleines Spiel entstehen, welches vollfunktionsfähig ist und keine Fehler enthält.

Die Ziele wurden erreicht und das Programm auch ausführlich getestet.

Das Spiel hat 4 Layouts: Hilfe, Kampfsystem, Pokémemes auswählen und das Startmenü.

Die verschiedenen Teams können gegeneinander kämpfen. Nach dem Kampf kommt man zurück in das Startmenü.

### 1.6.2 Dokumentation

Die Dokumentation (dieses Dokument) ist vollständig ergänzt worden. Alle geforderten Inhalte sind genügend genau beschrieben so, dass eine andere Gruppe das Projekt später weiterführen könnte.

Dazu gehören auch das Gantt Diagramm mit den geplanten Arbeitspaketen.

## 2 Projektorganisation

In diesem Kapitel werden die Zuständigkeiten im Projekt definiert.

### 2.1 Organigramm

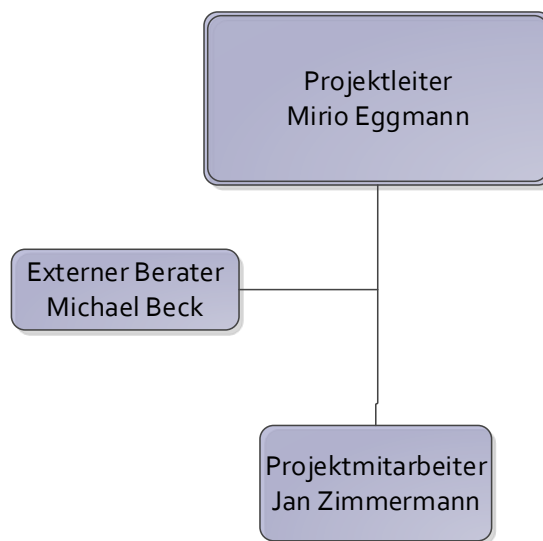


Abbildung 2: Organigramm der Projektorganisation

### 2.2 Zuständigkeiten der Mitarbeiter

#### 2.2.1 Projektleiter

Ein Projektleiter ist für die Planung und Steuerung eines Projekts zuständig. Weiter trifft der Projektleiter wichtige Entscheidungen und erteilt den Projektmitarbeitern die Aufgaben. Er trägt auch noch die Verantwortung, dass die geplanten Termine eingehalten werden.

#### 2.2.2 Externer Berater

Ein externer Berater hat die Aufgabe dem Projektleiter bei Entscheidungen und Veränderungen zu helfen. Um dem Kunden zu helfen entwickeln die Berater Lösungsvorschläge, welche sie dem Kunden dann vorstellen. Weiter führen externe Berater Auswertungen von Analysen aus.

#### 2.2.3 Projektmitarbeiter

Die Projektmitarbeiter arbeiten an Arbeitspaketen, welche sie vom Projektleiter bekommen haben. Um die Pakete zu erledigen, haben Sie nur eine begrenzte Zeit und sie werden durch den Projektleiter kontrolliert. Die Arbeiten werden neben der normalen Arbeit durchgeführt.

### 3 GUI Design

#### 3.1 MockUp 1 - Menü

Dieses Layout erscheint wenn man das Programm startet und stellt das Menü dar.

Nebst dem Titel und einem kleinen Bild enthält dieses Layout zwei Buttons. Der erste ist der Hilfe-Button. Mit dem gelangt man zum 4. MockUp Layout. Beim zweiten Knopf startet man das Pokémeme-Spiel. Mit diesem Knopf gelangt man zum zweiten MockUp Layout.

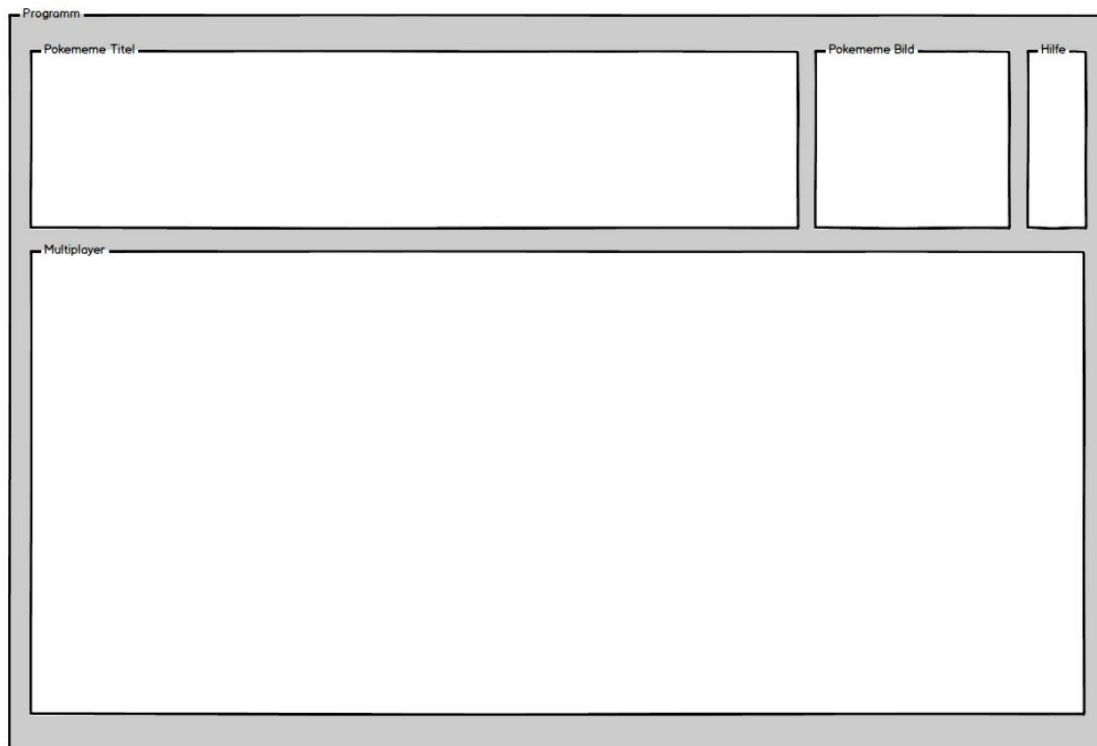


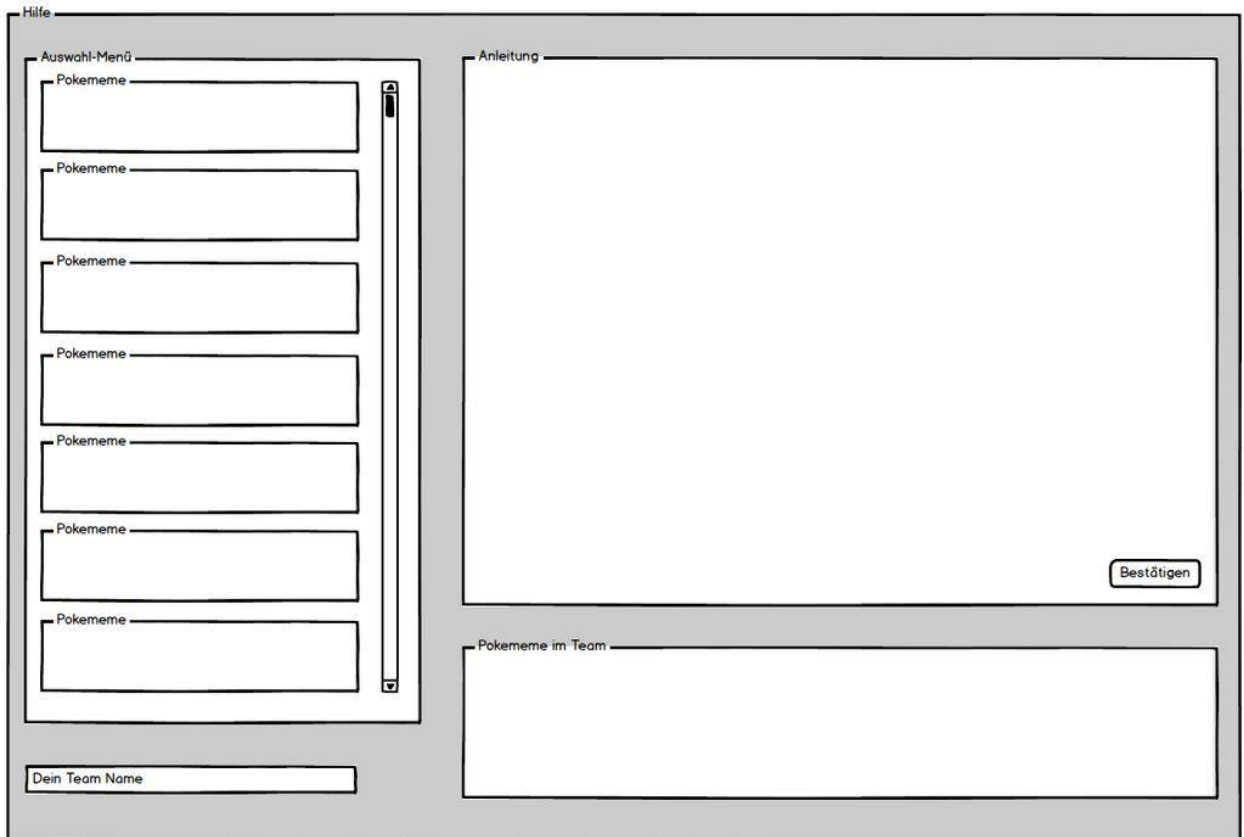
Abbildung 3 MockUp 1 - Menü



## 3.2 MockUp 2 - Team wählen

Dieses Layout kommt zum Einsatz, wenn man zu Beginn ein Pokémeme auswählt.

Hier kann man sein Teamname wählen und das Team mit den Pokémemes zusammenstellen. Wenn man bestätigen klickt, kommt man zum 3. MockUp-Layout.



The interface is titled 'Hilfe' in the top left corner. It is divided into several sections:

- Auswahl-Menü:** A vertical list of seven 'Pokememe' labels, each followed by a rectangular selection box. A vertical scrollbar is on the right side of this list.
- Anleitung:** A large, empty rectangular area on the right side of the interface.
- Bestätigen:** A button located in the bottom right corner of the 'Anleitung' section.
- Pokememe im Team:** A label above a large, empty rectangular box at the bottom right.
- Dein Team Name:** A label above a text input field at the bottom left.

Abbildung 4 MockUp 2 – Team wählen

### 3.3 MockUp 3 - Kampfsystem

Dieses Layout wird während den Kämpfen verwendet und zwar wenn man im Untermenü Pokémeme ist und seine Pokémemes anschauen oder tauschen will oder wenn man eine Attacke ausüben möchte. Wenn der Kampf fertig ist, kommt man zum Start-Hauptmenü. Dies ist das erste Layout.



Abbildung 5 MockUp 3 - Kampfsystem



### 3.4 MockUp 4 - Hilfedesign

Dieses Design ist für die Hilfe gedacht. Darin befindet sich nur die Anleitung zum Spiel wie auch die Informationen der Programmierer/Eigentümer des Spiels.



Abbildung 6 MockUp 4 - Hilfedesign

## 4 Datenhaltung

### 4.1 ERM

Wir haben uns entschieden zu Beginn keine Datenbank einzubinden, weil wir schon ohne diese viel zu tun und eher wenig Zeit haben. Daher werden wir dies nur noch ergänzen, falls wir noch genügend Zeit haben, nachdem wir alles Wichtige erledigt haben.

## 5 Arbeitspakete

### 5.1 Arbeitspaket A – Projektplanung

Damit wir unser Projekt optimal vorbereiten können müssen wir eine kleine Dokumentation schreiben. Diese soll ein Zeitmanagement beinhalten (Gantt), das geplante Layout vom GUI und sonstige Informationen zum Projekt.

Beschreibung	
Anfangsvoraussetzungen	Damit wir mit diesem Arbeitspaket beginnen können benötigen wir Java Kenntnisse, damit wir unser geplantes Projekt auch durchführen können. Weiter benötigen wir eine Idee, was für ein Projekt wir machen wollen. Damit wir auch schon eine Ahnung haben, wie wir das Projekt angehen wollen, müssen wir eine Dokumentation schreiben.
Ziele: (Ergebnisbeschreibung)	Das Ziel von diesem Arbeitspaket ist, dass wir ein passendes Projekt finden und dies mithilfe einer Dokumentation, einem MockUp und dem Gantt bis am Mittwoch 15.April vorbereiten.
Abhängigkeiten:	Dieses Arbeitspaket ist das erste von diesem Projekt und daher hat es noch keine Abhängigkeiten zu anderen Paketen.
Aufwand in 2h Schritten:	Wir rechnen mit 8 Stunden.
Benötigte Mittel:	<ul style="list-style-type: none"> <li>• Word</li> <li>• Excel</li> <li>• MockUp (myBalsamiq)</li> </ul>
Meilenstein [JA/NEIN]	JA

Tabelle 1 Arbeitspaket A- Projektplanung



## 5.2 Arbeitspaket B – Projektantrag erstellen

In diesem Arbeitspaket, besprechen wir mit dem Coach ob wir dieses Projekt machen dürfen. Er überprüft ob es von der Zeit her reicht und ob es komplex genug ist.

Beschreibung	
Anfangsvoraussetzungen	Damit ein Antrag für ein Projekt erstellt werden kann, müssen wir wissen, was für ein Projekt wir machen wollen und ob es realistisch ist, dies in 4.5 Tagen umzusetzen.
Ziele: (Ergebnisbeschreibung)	Das Ziel ist, dass ein gut beschriebener Projektantrag erarbeitet wird, welcher wir bis am Abend vom 14. April abgeben müssen.
Abhängigkeiten:	Das Paket hängt noch von keinem anderen ab und wird parallel zur Projektplanung durchgearbeitet, jedoch benötigt man schon eine Idee, was man machen will und wie man das in etwa durchführen will.
Aufwand in 2h Schritten:	Wir rechnen für dieses Paket 4 Stunden.
Benötigte Mittel:	<ul style="list-style-type: none"> <li>• Drucker</li> <li>• Coach (Unterschrift)</li> <li>• Word 2013</li> </ul>
Meilenstein [JA/NEIN]	JA

Tabelle 2 Arbeitspaket B - Projektantrag erstellen

### 5.3 Arbeitspaket C – Programm Backend

In diesem Arbeitspaket wird das Programm trocken ohne Benutzeroberfläche programmiert. Wir arbeiten nach Planung.

Beschreibung	
Anfangsvoraussetzungen	Die Planung für das Projekt muss abgeschlossen sein. Man benötigt das fehlerfreie UML-Diagramm.
Ziele: (Ergebnisbeschreibung)	Das Spiel funktioniert und die Ausgaben auf der Konsole stimmen. Die Benutzer könnten jetzt schon das Spiel spielen.
Abhängigkeiten:	<ul style="list-style-type: none"> <li>• Arbeitspaket A: Projektplanung</li> <li>• Arbeitspaket B: Projektantrag</li> </ul>
Aufwand in 2h Schritten:	Wir rechnen mit 20 Stunden.
Benötigte Mittel:	<ul style="list-style-type: none"> <li>• Eclipse</li> <li>• Java JDK</li> <li>• Java JRE</li> </ul>
Meilenstein [JA/NEIN]	JA

Tabelle 3 Arbeitspaket C - Programm Backend

## 5.4 Arbeitspaket D – Programm Frontend

Damit die verschiedenen Abläufe auf der Konsole ausgegeben werden, werden wir ein GUI miteinbeziehen. Das GUI besteht aus vier Layouts. Eines für das Kampfsystem, eins für das Startmenü, eins für die Hilfe/Anleitung und eins für die Auswahl des Pokémemes.

Beschreibung	
Anfangsvoraussetzungen	Vom Backend sollte schon ein bisschen vorhanden sein, da man das GUI auf das Frontend aufbaut, jedoch ist das Grundgerüst noch nicht vom Backend abhängig. Ausserdem braucht man auch die geplanten Layouts.
Ziele: (Ergebnisbeschreibung)	Die Benutzer können die Pokémemes einfach in ihr Team hinzufügen und auch die Memes ohne Probleme gegeneinander antreten lassen. Das Programm ist übersichtlich.
Abhängigkeiten:	<ul style="list-style-type: none"> <li>• Arbeitspaket A: Projektplanung</li> <li>• Arbeitspaket B: Projektantrag</li> </ul>
Aufwand in 2h Schritten:	Wir rechnen mit 20 Stunden.
Benötigte Mittel:	<ul style="list-style-type: none"> <li>• Eclipse</li> <li>• Java JDK</li> <li>• Java JRE</li> </ul>
Meilenstein [JA/NEIN]	JA

Tabelle 4 Arbeitspaket D - Programm Frontend

## 5.5 Arbeitspaket E – Programm Frontend und Backend zusammenführen

Damit die Kommunikation zwischen dem Frontend und dem Backend funktionieren, muss das gesamte zusammengeführt werden.

Beschreibung	
Anfangsvoraussetzungen	Das Backend und Frontend sollten beide schon beinahe fertiggestellt werden, es können aber auch schon vor dem Abschluss der beiden Pakete Sachen zusammengeführt werden.
Ziele: (Ergebnisbeschreibung)	Das Ziel der Zusammenführung ist, dass die Kommunikation zwischen dem Backend und dem Frontend ohne Probleme funktioniert. Wenn dies erfolgreich fertiggestellt wird, ist das Programm bereit für das Testing.
Abhängigkeiten:	<ul style="list-style-type: none"> <li>• Arbeitspaket A: Projektplanung</li> <li>• Arbeitspaket B: Projektantrag</li> </ul> <p>Weiter sollte das Backend und das Frontend schon ziemlich fortgeschritten sein, damit es möglich ist die beiden zusammenzuführen.</p>
Aufwand in 2h Schritten:	Wir rechnen mit 12 Stunden.
Benötigte Mittel:	<ul style="list-style-type: none"> <li>• Eclipse</li> <li>• Java JDK</li> <li>• Java JRE</li> </ul>
Meilenstein [JA/NEIN]	JA

Tabelle 5 Arbeitspaket E – Programm Frontend und Backend zusammenführen



## 5.6 Arbeitspaket F – Testing

Zwei komplexe Klassen werden getestet und vorhandene Fehler werden behoben. Für die Tests verwenden wir JUnit-Tests von Java.

Beschreibung	
Anfangsvoraussetzungen	Das fertige Spiel und Eclipse. Auch brauchen wir die das UML-Diagramm.
Ziele: (Ergebnisbeschreibung)	Die zwei komplexesten Klassen weisen keine Fehler mehr auf. Somit läuft das Programm fast Fehlerfrei!
Abhängigkeiten:	<ul style="list-style-type: none"> <li>• Arbeitspaket A: Projektplanung</li> <li>• Arbeitspaket B: Projektantrag</li> <li>• Arbeitspaket C: Programm Backend</li> <li>• Arbeitspaket D: Programm Frontend</li> <li>• Arbeitspaket E: Programm Frontend und Backend zusammenführen</li> </ul>
Aufwand in 2h Schritten:	Wir rechnen mit 4 Stunden.
Benötigte Mittel:	<ul style="list-style-type: none"> <li>• Eclipse</li> <li>• JUnit-Tests</li> <li>• Java JRE</li> <li>• Java JDK</li> </ul>
Meilenstein [JA/NEIN]	NEIN

Tabelle 6 Arbeitspaket F – Testing

## 5.7 Arbeitspaket G – Arbeitsumgebung dokumentieren

Da wir im Lieferumfang auch die Dokumentation beilegen, muss diese auch vom Team geschrieben werden. In diesem Arbeitsschritt, dokumentieren wir aber nur die 2. Dokumentation.

Beschreibung	
Anfangsvoraussetzungen	Es braucht das Programm Word 2013 & Excel 2013 von Microsoft und das fertige Spiel.
Ziele: (Ergebnisbeschreibung)	Das Projekt so dokumentieren, dass ihn jemand anders, der sich mit Java auskennt, dieses Projekt nachbauen/nachprogrammieren kann.
Abhängigkeiten:	<ul style="list-style-type: none"> <li>• Arbeitspaket A: Projektplanung</li> <li>• Arbeitspaket B: Projektantrag</li> <li>• Arbeitspaket C: Programm Backend</li> <li>• Arbeitspaket D: Programm Frontend</li> <li>• Arbeitspaket E: Programm Frontend und Backend zusammenführen</li> <li>• Arbeitspaket F: Testing</li> </ul>
Aufwand in 2h Schritten:	Wir rechnen mit 44 Stunden, wobei diese Präsentation immer parallel zum Programmieren läuft.
Benötigte Mittel:	<ul style="list-style-type: none"> <li>• Eclipse</li> <li>• Word 2013</li> <li>• Excel 2013</li> </ul>
Meilenstein [JA/NEIN]	JA

Tabelle 7 Arbeitspaket G – Arbeitsumgebung dokumentieren

## 5.8 Arbeitspaket H – Präsentation

Da wir unser Projekt noch präsentieren werden, müssen wir noch eine Präsentation erstellen, die das wichtigste von unserem Projekt festhält.

Beschreibung	
Anfangsvoraussetzungen	Es braucht das Programm PowerPoint 2013 von Microsoft und das fertige Spiel wie auch die Dokumentation.
Ziele: (Ergebnisbeschreibung)	Eine fertige Präsentation, das den Benutzern oder den Interessenten zeigt, was wir im Projekt gemacht haben und wie es funktioniert.
Abhängigkeiten:	<ul style="list-style-type: none"> <li>• Arbeitspaket A: Projektplanung</li> <li>• Arbeitspaket B: Projektantrag</li> <li>• Arbeitspaket C: Programm Backend</li> <li>• Arbeitspaket D: Programm Frontend</li> <li>• Arbeitspaket E: Programm Frontend und Backend zusammenführen</li> <li>• Arbeitspaket F: Testing</li> <li>• Arbeitspaket G: Arbeitsumgebung dokumentieren</li> </ul>
Aufwand in 2h Schritten:	Wir rechnen mit 6 Stunden.
Benötigte Mittel:	<ul style="list-style-type: none"> <li>• PowerPoint 2013 von Microsoft</li> <li>• Eclipse Programmierumgebung</li> <li>• Java JRE</li> <li>• Java JDK</li> </ul>
Meilenstein [JA/NEIN]	NEIN

Tabelle 8 Arbeitspaket H - Präsentation

## 5.9 Zeitplanung

### 5.9.1 GANTT

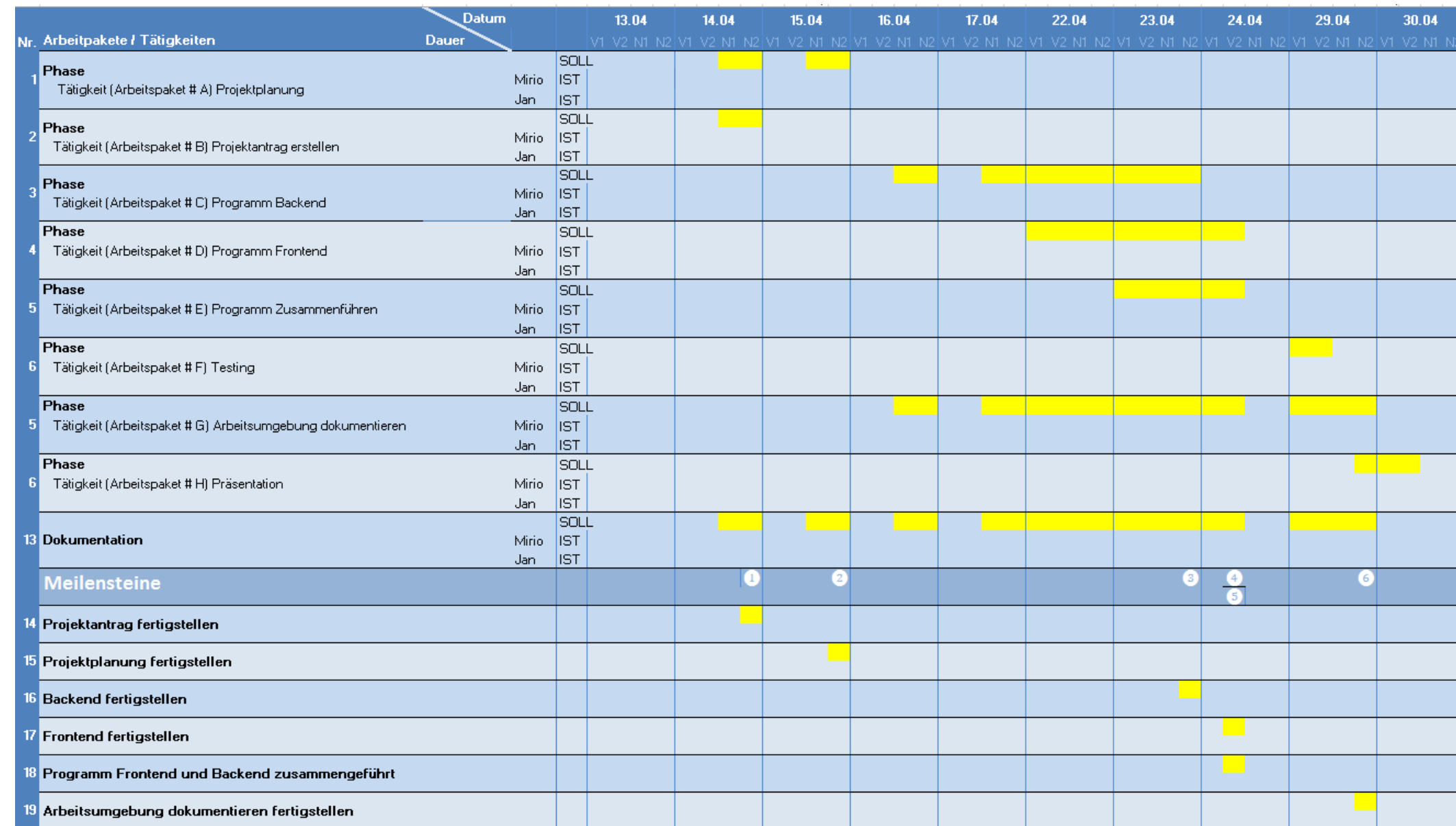


Abbildung 7 Gantt

Ein Feld entspricht 2h (wir rechnen mit einem 8h Tag)

## 6 Testcases

### 6.1 Beschreibung der Testumgebung

- Windows 7 Enterprise
- Java JRE Version 8 Update 40
- Java Development Kit Version 8 Update 40
- Eclipse 4.4.2 welches 2 GB Arbeitsspeicher zur Verfügung hat.
- 8 GB Arbeitsspeicher
- Intel Core i7 (Dual Core)

### 6.2 Liste der Testcases

#### 6.2.1 Test Case 1 – Pokémeme heilen

Test Case	ID: 1
<b>Kontext:</b>  In diesem Test-Case werden wir das Use-Case: Pokémeme heilen testen. Dort wird auf ein Button geklickt wenn man das Pokémeme heilen möchte. Das System fügt dann dem Pokémeme 25 Leben hinzu.	
<b>Erwartetes Resultat:</b>  Wenn ein Pokémeme nur noch 20 Lebenspunkte hat, und man auf diesen Heilen-Knopf klickt, hat es dann nach dieser Runde 45 Lebenspunkte.	

Tabelle 9 Test Case 1 – Pokémeme heilen

#### 6.2.2 Test Case 2 – Beenden der Applikation

Test Case	ID: 10
<b>Kontext:</b>  Unser letztes Test-Case handelt sich um das saubere Beenden der Applikation. Dazu haben wir in unserem Programm definiert, dass sich beim Beenden der App automatisch der Prozess selber killt.	
<b>Erwartetes Resultat:</b>  Wenn man den Taskmanager öffnet (Taskliste->Rechtsklick->Taskmanager->Prozesse) und das Programm beendet hat, sollte der Prozess dort nicht mehr auffindbar sein. (er heisst javaw.exe).	

Tabelle 10 Test Case 2 – Beenden der Applikation



### 6.2.3 Test Case 3 – Zurück zum Hauptmenü

Test Case	ID: 4
<b>Kontext:</b>  In diesem Test-Case werden wir das Use-Case: Zurück zum Hauptmenü testen. Wenn man sich verklickt hat (anstatt auf Multiplayer auf Hilfe geklickt), kann man über den Knopf: Zurück zum Hauptmenü gelangen.	
<b>Erwartetes Resultat:</b>  Klickt man auf den Hauptmenü-Knopf, gelangt man zum Hauptmenü und alles funktioniert.	

Tabelle 11 Test Case 3 - Zurück zum Hauptmenü

### 6.2.4 Test Case 4 – Pokémeme auswählen

Test Case	ID: 2
<b>Kontext:</b>  In diesem Test-Case werden wir das Use-Case: Pokémeme auswählen testen. Nachdem man auf den Multiplayer-Button klickt, im Start-/Hauptmenü, kommt man zum zweiten Mockup-Layout. In dem kann man sein Team zusammen stellen. Auf der linken Seite sieht man eine Auswahl von vielen Pokémemes. Die man mit bestätigen dem Team hinzufügen kann. Das hinzugefügte Pokémeme kann man unten im Fenster: Pokémeme im Team schliesslich sehen.	
<b>Erwartetes Resultat:</b>  Wenn man auf ein Pokémeme klickt und bestätigen drückt, soll dieses einer Liste hinzugefügt werden und unten im Fenster: Pokémeme im Team angezeigt werden.	

Tabelle 12 Test Case 4 - Pokémeme auswählen



#### 6.2.5 Test Case 5 – Hilfe aufrufen

Test Case	ID: 3
<b>Kontext:</b>	
In diesem Test-Case werden wir das Use-Case: Hilfe aufrufen testen. Wenn man nicht weiss, wie man dieses Spiel spielt, kann man im Hauptmenü auf den Hilfe-Button klicken. Dann gelangt man zu einem neuen Fenster mit dem vierten Mockup-Layout. Darin sind die Anleitung und die Informationen zu den Entwickler enthalten.	
<b>Erwartetes Resultat:</b>	
Das neue Fenster wird bei Knopfdruck auf Hilfe, im Hauptmenü, angezeigt und der Benutzer kann die jeweiligen Informationen lesen.	

Tabelle 13 Test Case 5 - Hilfe aufrufen

#### 6.2.6 Test Case 6 – Gegeneinander kämpfen

Test Case	ID: 5
<b>Kontext:</b>	
In diesem Test-Case werden wir das Use-Case: Gegeneinander kämpfen testen. Dort werden im dritten MockUp Layout zwei Pokémemes gegeneinander kämpfen. Wenn man einen Angriff auswählt, fügt dieser einem Pokémeme Schaden zu und die Lebenspunkte sinken. Wenn ein Pokémeme keine Leben mehr hat, stirbt es und kann somit nicht mehr gespielt werden.	
<b>Erwartetes Resultat:</b>	
Wenn man eine Attacke auswählt, fügt das angreifende Pokémeme dem anderen Schaden zu. Da es verschiedene Attacken gibt, kann man nicht klar definieren wie viel es abzieht.	

Tabelle 14 Test Case 6 - Gegeneinander kämpfen



#### 6.2.7 Test Case 7 – Pokémeme Daten

Test Case	ID: 6
<b>Kontext:</b>  Im 6. Test-Case kontrollieren wir, ob bei allen Pokememes die Stats ausgegeben werden (Name, HP, Verteidigung Attacken etc.) Dies findet im MockUp 2 „Team wählen“ statt. Wir werden beim Versuch auf alle Pokémemes klicken in der Suchleiste links und dann sollten auf der rechten Seite die Daten wie eben HP, Name etc erscheinen.	
<b>Erwartetes Resultat:</b>  Das erwartete Resultat ist, dass beim Drücken auf alle Pokémemes links in der Suchleiste, auf der rechten Seite die Daten ausgegeben werden und keine leeren Felder vorhanden sind.	

Tabelle 15 Test Case 7 - Pokémeme Daten

#### 6.2.8 Test Case 8 – Pokémeme wechseln

Test Case	ID: 7
<b>Kontext:</b>  In diesem Test-Case testen wir, ob das Switchen zwischen den Pokememes im Kampf funktioniert. Dazu navigieren wir im Layout 3 Kampfsystem und dort klicken wir unten rechts auf das Symbol, wo „Pokémeme“ darauf steht. Danach sollten in der unteren Hälfte auf der linken Seite die Pokémemes vom eigenen Team erscheinen und dort kann man dann darauf klicken und falls es noch lebt auswechseln.	
<b>Erwartetes Resultat:</b>  Wir erwarten, dass man auf Pokémeme drücken kann und danach auf ein eigenes Pokémeme, klicken kann und dies wird dann neu in den Kampf geschickt(muss noch leben).	

Tabelle 16 Test Case 8 - Pokémeme wechseln



#### 6.2.9 Test Case 9 – Namen auswählen

Test Case	ID: 8
<b>Kontext:</b>	
In diesem Test-Case schauen wir, ob wir beim erstellen vom Team einen Namen definieren können und dieser dann im Spiel übernommen wird. Dazu benötigen wir das MockUp 2 Team zusammenstellen und noch das Kampfsystem.	
<b>Erwartetes Resultat:</b>	
Beim erstellen vom Team wird der gewünschte Name eingegeben. Wir erwarten, dass während dem Kampf der richtige Name ausgegeben wird.	

Tabelle 17 Test Case 9 - Namen auswählen

#### 6.2.10 Test Case 10 – Start

Test Case	ID: 9
<b>Kontext:</b>	
Dieses Test-Case handelt sich um das starten vom Spiel. Das ganze passiert im MockUp 2 und der erste Spieler muss sein Team bereits zusammengestellt haben. Danach steht unten rechts von diesem Fenster start.	
<b>Erwartetes Resultat:</b>	
Wenn man auf Start drückt soll das Spiel gestartet werden und der erste Spieler soll seinen Zug machen können.	

Tabelle 18 Test Case 10 – Start