

Projektdokumentation

GrowthV2

Datum: 11.05.2021

Erstellt von: Jellef Abbenseth

Version: 1.0

TABELLENVERZEICHNIS.....	C
ABBILDUNGSVERZEICHNIS	C
1. EINLEITUNG	1
1.1. PROJEKTBE SCHREIBUNG	1
1.2. PROJEKTZIEL.....	1
1.3. PROJEKTUMFELD	1
2. PROJEKTPLANUNG	2
2.1. PROJEKTPHASEN	2
2.2. RESSOURCEN	3
2.3. ZEITLICHER PROJEKTPLAN.....	3
3. ANALYSE	4
3.1. IST-ANALYSE.....	4
3.2. PROJEKTABGRENZUNG.....	4
4. KONZEPTION	4
4.1. ZIELPLATTFORM	4
4.2. OBERFLÄCHE UND INTERAKTION	5
4.3. EINBINDUNG ALTES PROJEKT.....	5
4.4. SPIELMÖGLICHKEITEN	5
4.4.1 <i>Erkunden</i>	5
4.4.2 <i>Ausruhe</i>	6
4.4.3 <i>Profil</i>	6
4.4.4 <i>Kampf</i>	6
4.4.5 <i>Speichern/Laden</i>	6
4.4.5 <i>Fenstergröße</i>	6
4.5. PFLICHTENHEFT	6
5. IMPLEMENTIERUNG.....	7
5.1. OBERFLÄCHE UND INTERAKTION	7
5.2. EINBINDUNG ALTES PROJEKT.....	7
5.3. SPIELMÖGLICHKEITEN	7
5.3.1 <i>Erkunden</i>	7
5.3.2 <i>Ausruhen</i>	7
5.3.3 <i>Profil</i>	8
5.3.4 <i>Kampf</i>	8
5.3.3 <i>Speichern/Laden</i>	9
5.3.3 <i>Fenstergröße</i>	9
5.4. ABWEICHUNGEN, ANPASSUNGEN UND ENTSCHEIDUNGEN	9
6. TESTUNG UND ABNAHME	10
6.1. QUALITÄTSSICHERUNG	10
6.2. INTERAKTIONSTEST	10
6.3. ABNAHME DES KUNDEN	10
6.4. NICHT BEHOBENE PROBLEME	10
7. DOKUMENTATION.....	10
8. FAZIT	11
8.1. SOLL-IST-VERGLEICH.....	11
8.2. AUSBlick	11
8.3. LESSONS LEARNED	12
LITERATURVERZEICHNIS.....	13
EIDESSTATTLICHE ERKLÄRUNG.....	13
ANHANG	I

A.1 LASTENHEFT	I
A.2 ZEITLICHER PROJEKTPLAN	II
A.3 STARTBILDSCHIRM GROWTH.....	II
A.4 GROWTH SPIEL MIT ABFRAGE	III
A.5 MOCKUP BENUTZEROBERFLÄCHE	III
A.6 FLIEßDIAGRAMM START-PANEL ABFRAGE ANZEIGE LADE-BUTTON	IV
A.7 FLIEßDIAGRAMM SAVE-PANEL SPEICHERN	V
A.8 PFLICHTENHEFT.....	VI
A.9 ABNAHMEPROTOKOLL.....	VI

Tabellenverzeichnis

Tabelle 1: Projektphasen.....	2
Tabelle 2: Hardware Ressourcen	3
Tabelle 3: Software Ressourcen.....	3
Tabelle 4: Personal Ressourcen	3
Tabelle 5: Projektphasen Soll-Ist-Vergleich	11

Abbildungsverzeichnis

Abbildung 1: Gantt Diagramm Zeitlicher Ablauf.....	II
Abbildung 2: Growth Startbildschirm.....	II
Abbildung 3: Growth Spiel mit Abfrage.....	III
Abbildung 4: Mockup Benutzeroberfläche.....	III
Abbildung 5: Fließdiagramm Start-Panel Abfrage Anzeige Lade-Button.....	IV
Abbildung 6: Save-Panel Speichern	V

Abkürzungsverzeichnis

Aktionen (Klick auf einen Button)

Spezielle Funktion, die etwas auslöst 5

Benutzeroberfläche

Ein Fenster eines Programms zur Steuerung für den Benutzer. 1

Erfahrungspunkte

Messwert, um festzustellen, ob Aktionen erfolgreich abgeschlossen wurden. 6

IntelliJ Community IDEA 2020.3

Eine Entwicklungsumgebung für verschiedene Programmiersprachen 4

Kommandozeile

Eine Eingabemaske von Windows für verschiedene Befehle. 1

Kontern

Einen Angriff abfangen und zurückwerfen. 5

Level

Messwert, um Vergleiche von Fähigkeiten anstellen zu können. 6

Pop-Up-Fenster

Kleines Fenster zur Abfrage oder Information 4

RPG (Roll-Playing-Game)* *RPG*

Touch-screen

Berührungsempfindlicher Bildschirm 5

Hinweis:

Alle mit * markierten Begriffe sind im Abkürzungsverzeichnis und Glossar aufgeführt.

1. Einleitung

Die Projektdokumentation wurde im Rahmen eines privaten Projekts erstellt. Im Folgenden soll aufgezeigt werden, wie das Projekt abgelaufen ist, welche Veränderungen sich bei der Durchführung nach der eigentlichen Planung ergeben haben, wie das Projekt beendet wurde und was der Verfasser im Rahmen des Projekts gelernt hat. Bei dem Projekt handelt es sich um eine Weiterentwicklung eines zuvor in Eigen Arbeit erstellten Programms.

1.1. Projektbeschreibung

Bei dem Projekt, Growth V2, handelt es sich um ein MVP (Minimal-Viable-Produkt)* als kleines textbasiertes RPG (Roll-Playing-Game)*. Es basiert auf dem zuvor erstellten Programm, Growth, welches bereits die grundlegenden Spielfunktionen umfasst. Hierbei konnte der Spieler durch Eingabe vorgegebener Zahlen das Spielgeschehen beeinflussen. Es soll eine graphische Benutzeroberfläche erstellt werden, die eine einfachere Handhabung ermöglicht.

1.2. Projektziel

Ziel des Projekts ist es die vorhandenen Spielfunktionen zu übernehmen und anzupassen. Zusätzlich sollte eine graphische Oberfläche erstellt werden, da das Spiel zuvor über Eingaben und Ausgaben in der Kommandozeile dargestellt wurde. Der Spieler soll in der Lage sein, durch vorgegebene Elemente das Spielgeschehen zu steuern, die Fenstergröße anzupassen, sowie einen Spielstand speichern und laden zu können. Letztlich soll das Programm als eigenständige Datei vorliegen.

Ein weiteres essenzielles Ziel ist es, bisher gelernte Elemente zu vertiefen und neue Elemente, wie die Benutzeroberfläche zu erlernen und anzuwenden. Es soll aufgezeigt werden, dass gewisse Grundfunktionen in jedem RPG zu finden sind.

1.3. Projektumfeld

Das Projekt wird in Eigenarbeit am heimischen Computer durchgeführt. Es ist das erste richtige Projekt, um den Umgang und die Anforderungen für ein Projekt zu erlernen und anwenden zu können. Weiterhin werden durch die Ausbildung als Wirtschaftsinformatiker in Zukunft weitere Projekte auf mich zukommen.

2. Projektplanung

In diesem Abschnitt werden die Projektphasen und Ressourcenplanung näher erläutert. Diese wurden anhand eines vorgegebenen Lastenhefts ([A.1 Lastenheft](#)) ermittelt und geplant.

2.1. Projektphasen

Tabelle 1: Projektphasen

Projektphase ▼	Geplan ▼
Projektplanung (Vorgehensweise und Pflichtenheft)	3
Informationsbeschaffung	
a) Tutorial: RPG mit GUI	5
b) Abspeichern und Aufrufen von daten	2,5
c) Standalone Version erstellen	2
Realisierung gemäß Sollkonzept	
a) Oberfläche und Instruktionen	
Erstellen des Fensters	6
Erstellen der Aktionen als Tests	3
b) Erstellen von Klassen und Methoden	
Mutter und Kindklassen	2
Methoden	10
c) Verbinden von Benutzeroberfläche, Aktionen und Methoden	
Grundfunktionen mit Rückgabewerten	10
Hinzufügen der Speicher und Ladefunktion	3
Tests und Compilieren als eigenständige Version	6
Dokumentation	
a) Vorlagen erstellen	3
b) Abbildung und Diagramme erstellen	8
c) Text erfassen	10
d) Korrektur	4
Abnahme und Übergabe protokoll mit abschl. Test	2
Gesamtstunden	79,5

2.2. Ressourcen

Folgende Ressourcen wurden für die Planung und Durchführung dieses Projektes benötigt:

Hardware:

Tabelle 2: Hardware Ressourcen

Personal Computer		Zubehör	
Prozessor	AMD Athlon™ X4 959 Quad	Bildschirmmodell	2x ASUS VS248
Festplatte	Kingston SH103S3120G	Bildschirmgröße	2x 22 Zoll
	ST2000DL003-9VT166		
RAM	16 GB DDR3		
Netzwerkkarte	Realtek Gigabit Ethernet		

Software:

Tabelle 3: Software Ressourcen

Softwareart	Bezeichnung
Betriebssystem	Microsoft windows 10 Pro
Entwicklungsumgebung	IntelliJ IDEA Community 2020.2

Personal:

Tabelle 4: Personal Ressourcen

Bezeichnung	Einsatzbereich
Entwickler	Umsetzung des Projekts

Zur Informationsbeschaffung und zur Problemlösung ist ein Internetzugang notwendig. Die dadurch benutzten Quellen sind im Literaturverzeichnis aufgeführt.

2.3. Zeitlicher Projektplan

Der zeitliche Projektplan (erstellt in Microsoft Excel) befindet sich im Anhang ([Abbildung 2: Gantt Diagramm Zeitlicher Ablauf](#)).

3. Analyse

In diesem Kapitel wird der Ist-Zustand analysiert und das Projekt genau abgegrenzt.

3.1. Ist-Analyse

Growth, welches die Basis für das Projekt ist besitzt nur als Startbildschirm zur Abfrage des Benutzernamens eine Benutzeroberfläche. Danach wird der Spieler über Pop-Up-Fenster gefragt, welche Aktionen er durchführen möchte. Das Ergebnis seiner Eingaben wird in der Konsole angezeigt (siehe [A.3 Startbildschirm Growth](#) und [A.4 Growth Spiel mit Abfrage](#)). Weitere Elemente des Programms sind das Auswählen von Aktionen. Diese sind Erkunden, Ausruhen, Profil, Angriff und Fliehen.

Kurze Erklärung zu den Aktionen:

Erkunden: Es gibt eine bestimmte Chance, dass ein Kontakt mit einem Monster eintritt.

Ausruhen: Sowohl während des Kampfes als auch außerhalb möglich. Regeneriert die Ausdauer und das Leben.

Profil: Gibt eine kurze Info über den Spieler wie Leben, Ausdauer, Schaden, Erfahrung usw.

Angriff: Du fügst dem Monster schaden zu, und wirst ebenfalls angegriffen.

Fliehen: Chance den Kampf zu verlassen.

Nach einer bestimmten Anzahl von gewonnenen Kämpfen steigt der Spieler im Level, der die Werte leicht erhöht. Nach weiteren Kämpfen wird das Spiel beendet. Der Spieler geht als Sieger hervor. Sollte im Kampf sein Leben auf null sinken wird das Spiel beendet. In diesem Fall hat der Spieler verloren.

Die Grundfunktionen werden in Growth V2 übernommen und angepasst beziehungsweise etwas angepasst. Zusätzlich wird das Komplette Spiel über die Benutzeroberfläche spielbar sein. Es wird nicht mehr länger eine Eingabe notwendig sein. Die Aktionen werden durch vorgegebene Knöpfe sog. Buttons ausgelöst.

3.2. Projektabgrenzung

Weiterhin nicht möglich sind weitergehende Funktionen, die in einem Textbasierten Rollenspiel zu erwarten sind. Dazu zählen unter anderem Gegner, die in der Lage sind auszuweichen, zu blocken, zu kontern oder sonstige fortschrittliche Fähigkeiten besitzen.

Monster werden auch noch nicht im Level aufsteigen. Weiterhin kann der Spieler keine Gegenstände ausrüsten, tränke verwenden oder andere Interaktionen abgesehen von den Grundfunktionen tun können. Das Spiel wird auch weiterhin ein MVP bleiben.

4. Konzeption

Im Folgenden werden die Projektziele entworfen.

4.1. Zielplattform

Das Projekt wird auf Basis von Java und Java-Swing programmiert. Es wird auf den Plattformen Windows 10 veröffentlicht. Das Spiel wird in deutscher Sprache entwickelt. Es ist nicht vorgesehen, dass das Spiel auf anderen Plattformen spielbar ist. Das Projekt wurde mit IntelliJ Community IDEA 2020.3 entworfen.

4.2. Oberfläche und Interaktion

Die Oberfläche soll einfach gestaltet und funktional sein, da es sich um ein MVP handelt. Zudem ist dies das erste Projekt, bei dem der Entwickler Grafische Oberflächen einsetzt. Deshalb wird die Oberfläche mit einfachen Elementen gestaltet werden. Die Anpassung an verschiedene Fenstergrößen wird im einfachen Rahmen erfolgen. Alles in allem soll die Oberfläche funktional sein.

Bevor der Umsetzungsphase wurden in Plänen verschiedene Mockups ([A.5 Mockup Benutzeroberfläche](#)) erstellt, um die spätere Erscheinung in groben Zügen festzulegen. Die Pläne wurden für alle Fenster, die in dem Spiel aufrufbar sind, entworfen.

Die Interaktion mit der Oberfläche wird nach der Eingabe des Spielernamens ausschließlich mit der Maus erfolgen. Dies erleichtert die Programmierung zusätzlich und würde es theoretisch erlauben mit einem Touch-screen zu arbeiten.

Es wurden auch zwei Fließdiagramme ([A.6 Fließdiagramm Start-Panel Abfrage Anzeige Lade-Button](#) und [A.7 Fließdiagramm Save-Panel Speichern](#)) skizziert. Sowohl die Pläne für die Oberfläche als auch für den Ablaufplan wurden vom Kunden genehmigt.

4.3. Einbindung altes Projekt

Da das alte Projekt noch fast ohne Benutzeroberfläche auskam, mussten die verschiedenen Klassen und Methoden vollständig umgeschrieben werden. Growth V1 wurde deshalb lediglich als Vorlage verwendet. Der Code selbst wurde jedoch fast ausschließlich von Grund auf neu geschrieben. Dies hatte mehrere Gründe:

1. Das Spiel wurde auf die Oberfläche aufgebaut und nicht wie ursprünglich auf mehreren Klassen, die miteinander verbunden wurden.
2. Da die Eingabe durch Aktionen (Klick auf einen Button) ersetzt werden, mussten die einzelnen Methoden von Grund auf umgeschrieben werden.
3. Es sind ein paar Funktionen wie das Kontern hinzugekommen, die einen Einfluss auf die Klassen haben.
4. Die Textausgabe wurde durch die Oberfläche vollkommen verändert, weshalb dies einige Änderungen zur Folge hatte.
5. Gewisse Kontrollstrukturen wurden überflüssig, diese wurden durch die Aktionen ersetzt.

4.4. Spielmöglichkeiten

Das Spiel bietet grundsätzlich drei Optionen, die Gegend erkunden austragen, sich ausruhen und das Profil einsehen. Die Gegend erkunden kann zu einem Kampf führen. Ein Spielstand kann zusätzlich gespeichert oder geladen werden. Weiterhin lässt sich die Oberfläche auf vier verschiedene Größen einstellen. Nach dem Spielstart sind drei Buttons sichtbar, Erkunden, Ausruhen und Profil. Zudem wird das Profil des Spielers dargestellt und angefragt, was er als nächstes tun möchte.

4.4.1 Erkunden

Durch das Drücken auf den entsprechenden Butten wird unter gewissen Umständen ein Kampf ausgelöst. Wenn kein Kampf ausgelöst wird, kann der Spieler wieder zwischen Erkunden, Ausruhen und Profil wählen.

Sollte ein Kampf ausgelöst worden sein, ändert sich die Anzahl der Buttons, diese Zeigen nun Angriff, Blocken, Ausruhen und fliehen an.

4.4.2 Ausruhe

Bei dieser Funktion werden die Lebenskraft und die Ausdauer wieder aufgefüllt, dem Maximum angepasst, sollte sich der Spieler nicht in einem Kampf befinden. Danach wird der Spieler wieder gefragt, was er nun tun möchte.

4.4.3 Profil

Der Spieler bekommt die aktuellen Informationen über den Charakter dargestellt. Sollte der Spieler genügend Erfahrungspunkte gesammelt haben, wird er im Level aufsteigen. Danach wird er gefragt, was er tun möchte.

4.4.4 Kampf

Diese Möglichkeit wird ausgelöst durch das „Erkunden“. Dem Spieler werden nun vier Möglichkeiten angegeben, um den Kampf zu einem Abschluss zu bringen:

- Angriff: Schadet dem Gegner, der Gegner wert sich und schadet dem Spieler
- Blocken: Blockt den Angriff, der Gegner kann dem Spieler einen Zug lang keinen Schaden zufügen. Löst die Möglichkeit des Konters aus.
- Konter: Der nächste Angriff des Gegners wird mit erhöhtem Schaden zurückgeworfen. Gegner wird eine Runde kampfunfähig, sollte er noch genug Restleben haben.
- Ausruhen: Der Spieler regeneriert sich ein bisschen, die Werte Leben und Ausdauer werden erhöht.
- Fliehen: Es besteht eine Chance den Kampf unverletzt zu verlassen. Sollte die Aktion fehlschlagen, fügt der Gegner dem Spieler Schaden zu.

Der Kampf wird beendet, entweder mit dem Tod des Gegners, was dem Spieler Erfahrungspunkte hinzugibt oder dem Tod des Spielers, dies führt zum Ende des Spiels.

4.4.5 Speichern/Laden

Der aktuelle Spielstand wird gespeichert, damit dieser zu einem späteren Zeitpunkt wieder geöffnet werden kann. Beim Laden wird ein gespeicherter Spielstand wieder geöffnet, sodass der Spieler mit den gleichen Fähigkeiten und Erfahrungen weiterspielen kann.

Das Speichern und Laden sind nur außerhalb des Kampfes möglich.

4.4.5 Fenstergröße

Diese Funktion ermöglicht das Verändern der Größe des Fensters. Es sind vier Werte vorgegeben. Eine Auswahl führt dazu, dass das Spiel mit neuer Größe geöffnet wird. Der Spieler landet dabei wieder auf dem Startbildschirm. Nichtgespeicherte Spielstände gehen dabei verloren.

4.5. Pflichtenheft

Das Pflichtenheft ist im Anhang ([A.8 Pflichtenheft](#)).

5. Implementierung

In diesem Kapitel möchte der Autor die Realisierung der verschiedenen Projektziele erläutern.

5.1. Oberfläche und Interaktion

Die Oberfläche wurde mit Swing erstellt und frei von Hand in Java geschrieben. Die einzelnen Elemente der Oberfläche wurden anhand von verschiedenen Funktionen realisiert, die miteinander kombiniert wurden. Bei Swing ist es üblich die einzelnen Elemente ineinander zu verschachteln, um am Ende die gewünschte Darstellung zu erhalten. Dabei wurde darauf geachtet möglichst einfache Layouts zu verwenden.

Der Grundaufbau ist ein Card-Layout, welches auf den Frame gelegt wurde. Die verschiedenen Karten stellen die sichtbare Oberfläche dar und wurden in eigenen Klassen entworfen und von einer Hauptklasse aus bei Bedarf aufgerufen. Dabei war es wichtig, darauf zu achten, dass die Weitergabe von wichtigen Variablen gegeben wurde. Zudem sind die einzelnen Karten miteinander verbunden, was dazu führt, dass sie sich untereinander ansprechen können.

Um die Funktionalität der Karten zu gewährleisten wurde im ersten Schritt über das eigentliche Card-Layout zwei Buttons eingefügt, mit denen die Karten einzeln hintereinander weg durchgeklickt werden konnten. Als im späteren Verlauf alle Karten untereinander aufrufbar waren, wurde diese Funktion wieder entfernt.

Die Oberfläche wird durch den Spielstart ausgelöst und gestartet. Daraufhin wird direkt die erste Karte angezeigt. Sämtliche Funktionen, die durch Klicken auf die verschiedenen Buttons ausgelöst werden, werden über Abfragen in gesonderten Klassen durchgeführt. Ergebnisse werden bei Bedarf zurückgegeben.

5.2. Einbindung altes Projekt

Bei der Einbindung des alten Projekts wurde nur ein Teil der Super- und Kindklassen für Spieler und Monster übernommen. Hauptsächlich ging es dabei um die einzelnen Variablen, welche sich nicht verändert haben. Die Methoden selbst wurden im späteren Verlauf dann durch Interaktionen mit der Oberfläche neu geschrieben. Somit war das alte Projekt eher eine Vorlage. Dennoch wurde darauf geachtet die Funktionen selbst beizubehalten, da dasselbe Ergebnis erzielt werden sollte. Von dem eigentlichen Spielverlauf konnte jedoch keine Methode oder Klasse übernommen werden, da die Änderungen so umfangreich waren, dass es einen größeren Aufwand bedeutet hätte, als diese einfach grundsätzlich neu zu schreiben.

5.3. Spielmöglichkeiten

Die Spielmöglichkeiten wurden durch die Interaktionen mit den Buttons ausgelöst. Wie die Funktionsweise der einzelnen Funktionen ist, wird nun genauer erläutert.

5.3.1 Erkunden

Die Funktion wurde so beschrieben, dass eine Zufallszahl generiert wird, daraufhin wird durch eine Abfrage überprüft, ob diese Zahl größer oder kleiner eines bestimmten Wertes ist. Dieser hat die entsprechende Funktion ausgelöst. Entweder gab es eine Rückgabe, dass „nichts“ gefunden wurde oder der Spieler ist in einen Kampf geraten. Dies hat damit auch die Karte des Card-Layouts gewechselt, ohne, dass der Spieler einen Einfluss darauf hatte.

5.3.2 Ausruhen

Das Ausruhen hat im ersten Schritt abgefragt, wo sich der Spieler derzeit befindet. War er in der „offenen Welt“, so hat er sich vollkommen erholt. Dabei wurde abgefragt, ob das Leben und die Ausdauer kleiner oder gleich des maximalen Lebens bzw. der maximalen Ausdauer waren, sollte

eines der beiden nicht zugetroffen haben, wurde dieser Wert einfach zurückgesetzt auf den maximalen Wert. Zudem wurde noch ein Abfragewert für den Zustand des Charakters auf „false“ gesetzt. Dies hat zur Folge, dass der Spieler ausgeruht ist.

Sollte sich der Spieler gerade in einem Kampf befinden, wurde einfach überprüft, wie hoch der aktuelle Wert war und mit einem festen Wert multipliziert, was eine Erhöhung um 20 % zur Folge hatte, solange der maximale Wert nicht überschritten wurde. Im Kampf hatte die Aktion sich auszuruhen allerdings auch zur Folge, dass der Gegner den Spieler ungehindert Schaden zufügen konnte. Die Idee, ausweichen zu können, wurde verworfen, da sie im ursprünglichen Spiel nicht enthalten und auch nicht gefordert war.

5.3.3 Profil

Diese Funktion hat abgefragt, ob ein gewisser Wert an Erfahrungspunkten überschritten wurde, sollte dies zutreffen, wurde das Level des Spielers erhöht, seine Werte um einen festen Betrag erhöht und geprüft, ob Level 3 erreicht wurde. Sollte Level 3 erreicht worden sein, führte dies zum Spielende mit dem Hinweis, dass der Spieler gewonnen habe.

Sollte Level 3 nicht erreicht worden sein, wurde die Erfahrung um einen festen Wert, der für den Levelaufstieg notwendig war, reduziert. Danach wurden die aktuellen Werte des Charakters ausgegeben.

5.3.4 Kampf

Der Kampf wurde in einer eigenen Karte des Card-Layouts dargestellt und ermöglichte dem Spieler die Auswahl zwischen vier Aktionen.

Angriff:

Der Spieler hat dem Monster einen gewissen Schaden hinzugefügt, hat dabei aber auch einen festen Wert an Ausdauer verloren. Dasselbe war auch andersherum der Fall, das Monster hat dem Spieler geschadet. Dabei wurde jedoch vorher überprüft, ob die Ausdauer einen bestimmten Wert nicht unterschritten hatte. Sollte dies der Fall gewesen sein, hätte der Spieler oder das Monster sich ausgeruht, statt dem Gegenüber zu schaden. Sollte von einem der Beiden das Leben unter oder gleich null sein, wurde der Kampf beendet. Der Tod des Monsters hat dem Spieler Erfahrungspunkte gutgeschrieben, der Tod des Spielers hat das Spiel beendet. Auch hier wurde wieder die Karte geändert, mit der Ausgabe „Du hast verloren.“.

Blocken:

Beim Blocken hat der Charakter eine Runde lang keinen Schaden erlitten. Daraufhin wurde die Funktion Konter aktiviert (ersetzte Angriff).

Konter:

Der Angriff des Gegners wird zurückgeworfen und der Schaden mit einem gewissen Wert multipliziert. Je stärker der Charakter war, desto wahrscheinlicher wurde es den Gegner mit einem Angriff zu besiegen. Wie auch beim Angriff führte der Tod von einem der Beiden zum Ende des Kampfes.

Ausruhen:

Hierbei wurden das Leben und die Ausdauer um einen gewissen Prozentsatz erhöht, dennoch konnte der Gegner dem Spieler noch Schaden hinzufügen. Der Tod des Spielers bedeutet auch hier das Ende des Kampfes.

Fliehen:

Es wurde ein Zufallswert generiert, sollte dieser unter oder über einem gewissen Wert liegen, war die Flucht entweder erfolgreich oder nicht. Eine erfolgreiche Flucht führte dazu, dass der Spieler zurück in die „offene Welt“ geht ohne Erfahrungspunkte zu sammeln. Sollte die Flucht nicht erfolgreich gewesen sein, hat der Gegner dem Spieler Schaden hinzugefügt.

5.3.3 Speichern/Laden

Beim Speichern wurde erst überprüft, ob der Ordner für den Speicherstand existiert. Dieser wurde anschließend erstellt, wenn dies nicht der Fall war. Dasselbe gilt auch für den Speicherstand. Das Speichern erfolgte über einen zusammengesetzten String. Diesem String wurden die Werte des Charakters gegeben und mit einem Trennzeichen markiert.

Beim Laden wurde erst überprüft ob ein Ordner und ein Spielstand existiert. Sollte ersteres und/oder letzteres nicht vorhanden sein, wurde einfach ausgegeben, dass kein Spielstand zur Verfügung steht. Sollten diese existieren wurde überprüft, welche Werte im String vorhanden sind. Dabei wurde der String mit zwei aufeinanderfolgenden Funktionen in ein Array zerlegt. Die einzelnen Strings wurden anschließend dem Charakter wieder zugeordnet und bei Bedarf umgewandelt. Somit wurde gewissermaßen ein neuer Spieler erstellt mit den alten werten, da der alte Spieler nicht mehr länger existiert. Diese Art der Datenspeicherung funktioniert allerdings nur in diesem Zusammenhang, da die Eingetragenen Zahlen ein festes Layout besitzen. Zudem ist diese Art der Speicherung nur bis zu einem gewissen Grad sinnvoll.

5.3.3 Fenstergröße

Die Einstellung der Fenstergröße erfolgte über fest vorgegebene Werte, das Verändern des Fensters durch Ziehen wurde programmseitig unterbunden. Dabei wurde das bisherige Spiel beendet und eine neue Funktion mit den neuen Werten geöffnet. Deshalb ist ein Weiterspielen auch nicht mehr möglich. Dies war, zum aktuellen Wissensstand die sinnvollste Lösung des Problems.

5.4. Abweichungen, Anpassungen und Entscheidungen

Es bedurfte einiger Anpassungen des vorherigen Teils, da dieser nicht so ohne weiteres übernommen werden konnte. Weiterhin wurde im Laufe der Zeit Ideen verworfen, die nicht Bestandteil des Pflichtenhefts waren.

Eine dieser Anpassungen war der Konter, welcher bei dem Kampf hinzugefügt wurde. Weiterhin wurden die Klassen fast vollständig überarbeitet, da das vorherige Projekt nicht auf Benutzeroberflächen ausgelegt war. Durch diese Anpassungen konnten zusätzlich noch gewisse Fehlerquellen unterbunden werden, da nicht länger falsche Werte eingegeben werden konnten, die abgefangen werden mussten. Solche Abfragen sind im Projekt nicht übernommen worden, da es nur eine Stelle gibt, bei der der Spieler selbst eine Eingabe tätigen konnte.

Im Laufe des Projekts sind mehrere Ideen aufgekommen, wie zum Beispiel die Ausgabe mehrerer Sprachen. Weiterhin gab es auch die Idee gewisse Werte, wie Stilmittel zentral zu erstellen und in den Klassen dann aufzurufen. Da sich eine Implementierung nach abschließen einiger wichtiger Meilensteine als zu großer Aufwand erwies, wurde diese Idee ebenfalls verworfen. Das dadurch entstandene Problem ist jedoch, dass die Stilmittel in jeder Klasse erneut definiert werden müssen, was eine nachträgliche Änderung erschwert.

Es gibt ein paar Fehler, die bewusst nicht korrigiert wurden, da sie die Funktionen des Spiels nicht beeinflussen. So ist zum Beispiel keine detaillierte Anzeige des Spielstandes möglich. Weiterhin wurde in Kauf genommen, dass das Ändern der Bildschirmgröße einen Neustart des Spiels zur Folge hat.

6. Testung und Abnahme

Die Testung wurde während der Projekterstellung regelmäßig durchgeführt. Nach Abschluss des Konzepts wurde eine vollständige Testung aller Funktionen durchgeführt.

6.1. Qualitätssicherung

Das Projekt selbst war recht komplex, zudem haben sich regelmäßige Veränderungen im Programmcode ergeben, was dazu führte, dass nach jeder Änderung die Funktionen getestet wurden, da die einzelnen Programmteile aufeinander aufbauen. Dadurch wurde Fehler bereits frühzeitig erkannt und konnten direkt behoben werden. Dadurch war es möglich jeden Meilenstein mit voller Funktionsfähigkeit abzuschließen. Die gesamte Entwicklungszeit konnte somit auch innerhalb der geplanten Zeiten gehalten werden.

Dies führte jedoch auch dazu, dass der Projektplan zeitlich sich vollkommen verändert hat. Zudem gab es dadurch verschiedene Meilensteine, die zwischen den ursprünglich geplanten lagen.

6.2. Interaktionstest

Während der verschiedenen Tests wurde versucht möglichst viele Szenarien nachzustellen. Dadurch konnten einige Fehler im logischen Aufbau des Programms gefunden und beseitigt werden.

6.3. Abnahme des Kunden

Der Kunde hat das Projekt übernommen, das Übergabeprotokoll ist beigefügt ([A.9 Abnahmeprotokoll](#)).

6.4. Nicht behobene Probleme

Es sind weiterhin Fehler vorhanden, welche nicht behoben wurde. Gründe hierfür sind:

- Das Spiel wird dadurch nicht wirklich beeinflusst
- Die Fehlerüberprüfung und -behebung würde den Zeitlichen Rahmen sprengen
- Ein Fehler ist nur unter ganz bestimmten Situationen zugegen, welches das Spiel nicht weiter beeinträchtigt.

7. Dokumentation

Die Gesamtdokumentation setzt sich aus 3 Teilen zusammen: dieser Projektdokumentation, einer Kundendokumentation und den Programmzeilen. Alle drei Dokumente werden als separate PDF-Datei erstellt.

Die Kundendokumentation bietet eine Erklärung über die Voraussetzungen, Spielweise und wichtigsten Spielerischen Funktionen.

Die Projektdokumentation schildert das Projektgeschehen, von der Planung, Umsetzung bis zur Übergabe, sowie einem Fazit.

Die Programmzeilen bietet eine Übersicht über die Veränderungen des Programmcodes sortiert nach den einzelnen Klassen. Dieser Programmcode wurde nach jedem großen Meilenstein mit einem neuen Datum aktualisiert, sodass eine Nachverfolgung der Änderungen in einem gewissen Rahmen möglich ist.

8. Fazit

In diesem Abschnitt möchte der Autor noch auf die gezogenen Schlüsse und Erfahrungen eingehen. Zudem gibt es einen abschließenden Vergleich zum Soll Konzept.

8.1. Soll-Ist-Vergleich

Tabelle 5: Projektphasen Soll-Ist-Vergleich

Projektphase	Geplante Zeit (in Stunden)	Ist (in Stunden)	Differenz (in Stunden)
Projektplanung (Vorgehensweise und Pflichtenheft)	3	3	0
Informationsbeschaffung			
a) Tutorial: RPG mit GUI	5	5	0
b) Abspeichern und Aufrufen von daten	2,5	1	2
c) Standalone Version erstellen	2	0,5	1,5
d) Detailplanung informieren	0	0,5	-0,5
Detailplanung	0	1,75	-1,75
Realisierung gemäß Sollkonzept			
a) Oberfläche und Instruktionen			
Erstellen des Fensters	6		
Erstellen der Weitergaben (Aktionen als Tests)	3	11,75	-2,75
b) Erstellen von Klassen und Methoden			
Mutter und Kindklassen	2	1	1
Methoden	10		
c) Verbinden von Benutzeroberfläche, Aktionen und Methoden			
Grundfunktionen mit Rückgabewerten	10	12	8
Hinzufügen der Speicher und Ladefunktion	3	3	0
Tests und Compilieren als eigenständige Version	6		
Fehlerbehebung	0	6	0
Dokumentation			
a) Vorlagen erstellen	3	1,5	1,5
b) Abbildung und Diagramme erstellen	8	8	0
c) Text erfassen	10	8	2
d) Korrektur	4	2	2
Abnahme und Übergabe protokoll mit abschl. Test	2	2	0
Gesamtstunden	79,5	67	12,5

Rückblickend auf dieses Projekt wurden alle Anforderungen zu 100% erfüllt. Das Spiel wurde nach Vorgaben und Vorlage des vorangehenden Projekts überarbeitet und mit einer grafischen Benutzeroberfläche erstellt. Es wurden in der Zeitplanung jedoch einige Abweichungen festgestellt, die schließlich zu einer wesentlich kürzeren Arbeitszeit führten. Dies ist damit zu erklären, dass einige Schritte ineinander verschmolzen sind. Gleichzeitig kamen noch Schritte hinzu, welche zwar etwas Zeit in Anspruch genommen haben aber die Fehlerquellen an sich reduziert haben.

8.2. Ausblick

Das Projekt hat seinen Zweck erfüllt, dennoch hat der Kunde in Aussicht gestellt, dieses Projekt in Zukunft zu erweitern und umzugestalten. Immerhin lässt sich durch dieses Projekt gut feststellen, was das zuständige Programmieren, der auch zugleich Kunde ist, gelernt hat und wie er dies umsetzt. Das erste Projekt ist ebenfalls in diesem Zusammenhang entstanden.

8.3. Lessons Learned

Der Autor hat im Zuge des Projekts viele Erfahrungen über grafische Benutzeroberflächen und deren Verknüpfung zu einem großen funktionsfähigen Projekt gelernt. Die grafische Benutzeroberfläche war dabei ein neues Themengebiet, welches erst gelernt werden musste, bevor es angewendet werden konnte.

Zudem hat der Autor einiges über die Funktionsweise von Klassen und deren Zusammenspiel gelernt. Durch die häufigen Änderungen des Codes lässt sich mitverfolgen, dass die Erfahrungen zu einer veränderten Denkweise geführt haben, um daraufhin bessere Lösungen zu entwerfen. Es wurde festgestellt, dass die Lernkurve durch Projekte wie diese deutlich größer ist als im normalen Unterricht.

Weiterhin hat der Autor bei diesem Projekt auch Themenbereiche gelernt, die mit der Programmierung selbst nur bedingt zusammenhängen. Das Erstellen einer richtigen Dokumentation, sowie genauere Planung waren notwendig, um ein solches recht großes Projekt über mehrere dutzenden Stunden zu verwirklichen.

Diese gesammelten Erfahrungen werden in den nächsten Projekten mit einfließen und weiter verbessert. Zudem hat dieses Projekt die Motivation des Autors gesteigert, da einfache kleine Aufgaben zwar interessant sind, dennoch auf Dauer recht langweilig werden.

Literaturverzeichnis

- RyiSnow's blog: <https://ryisnow.net/2017/04/30/how-to-make-a-text-adventure-game-with-gui-in-java/>
- Java GUI-Programmierung: <https://java-tutorial.org/>
- ...

Eidesstattliche Erklärung

Ich, Jellef Abbenseth, versichere hiermit, dass ich meine Projektdokumentation mit dem Thema „Growth V2 (Java)“ selbstständig verfasst habe und keine anderen als die angegebenen Quellen (**unvollständig, beim nächsten Projekt besser darauf achten!**) und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe.

Ebenso bestätige ich, dass ich bei der Erstellung der Dokumentation meiner Projektarbeit weder teilweise noch vollständige Passagen aus Projektarbeiten übernommen habe, die bei der prüfenden oder einer anderen Kammer eingereicht wurden.

Pfintzal, den 08.12.2020

Unterschrift

Jellef Abbenseth

Anhang

A.1 Lastenheft

Einführung:

Erstellung eines einfachen Textbasierten Rollenspiels mit Grafischer Oberfläche.

IST-Situation:

Es ist ein grundlegendes textbasiertes Rollenspiel vorhanden, das jedoch keine grafische Oberfläche besitzt.

SOLL-Zustand:

- Der Spieler soll seinen Namen wählen können
- Der Spieler soll seinen Spielstand speichern und laden können
- Das Spiel soll außerhalb der Entwicklungsumgebung spielbar sein
- Es handelt sich um ein MVP (Minimal Viable Produkt)

Schnittstellen:

- Unabhängig von einer Entwicklungsumgebung
- Java Plattform

Systemanforderungen:

Soll auf einem einfachen Computer abspielbar sein oder auf einem Tablett mit Touch-Funktion und Windows-Umgebung.

Qualitätsanforderungen:

Es soll spielbar sein, ohne Endlosschleifen.

Risiken und Akzeptanz:

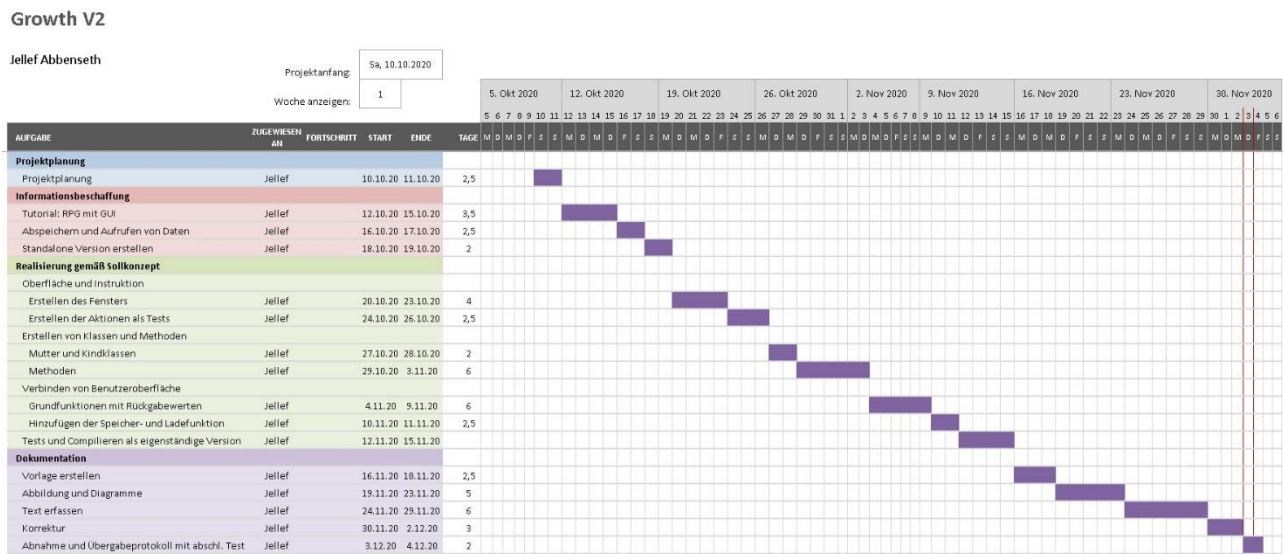
- Geplanter Aufwand ist zu gering
- Anpassung möglich, Ziel ist Fertigstellung

Abnahmekriterien:

Spielbares Spiel ohne Endlosschleife, unabhängig von der Entwicklungsumgebung und mit abschließender Dokumentation inklusive Tutorial.

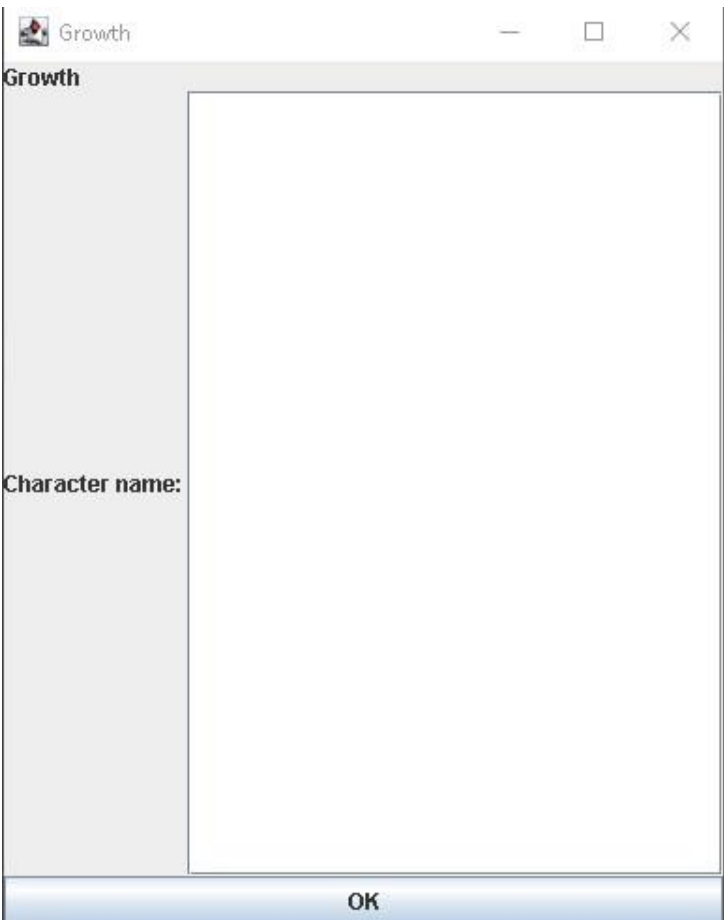
A.2 Zeitlicher Projektplan

Abbildung 1: Gantt Diagramm Zeitlicher Ablauf



A.3 Startbildschirm Growth

Abbildung 2: Growth Startbildschirm



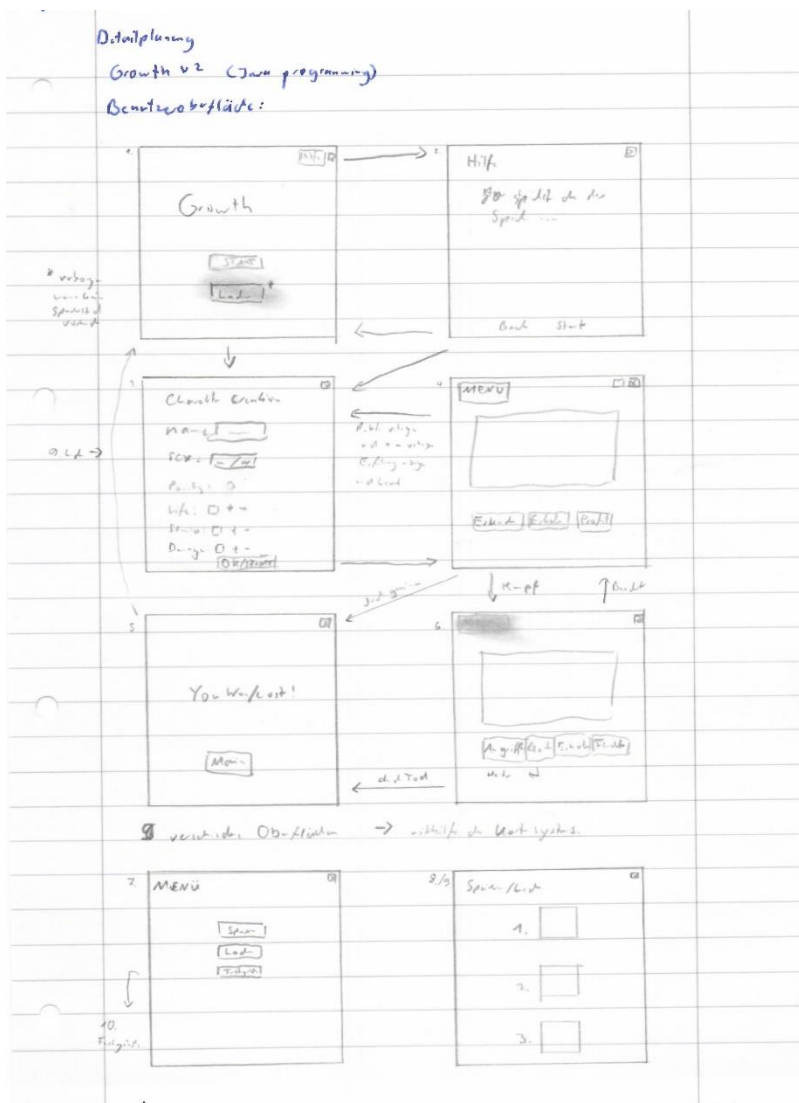
A.4 Growth Spiel mit Abfrage

Abbildung 3: Growth Spiel mit Abfrage



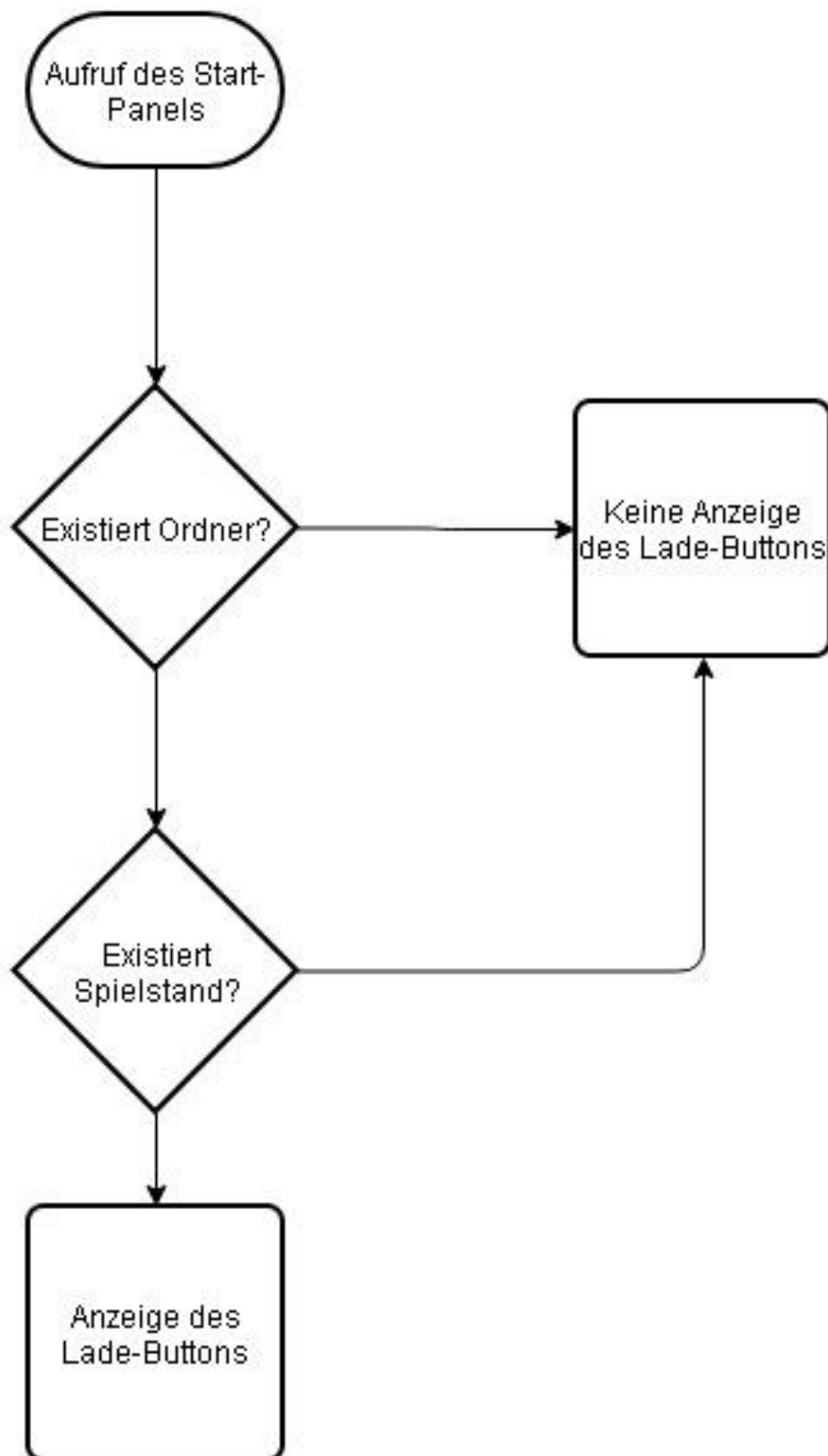
A.5 Mockup Benutzeroberfläche

Abbildung 4: Mockup Benutzeroberfläche



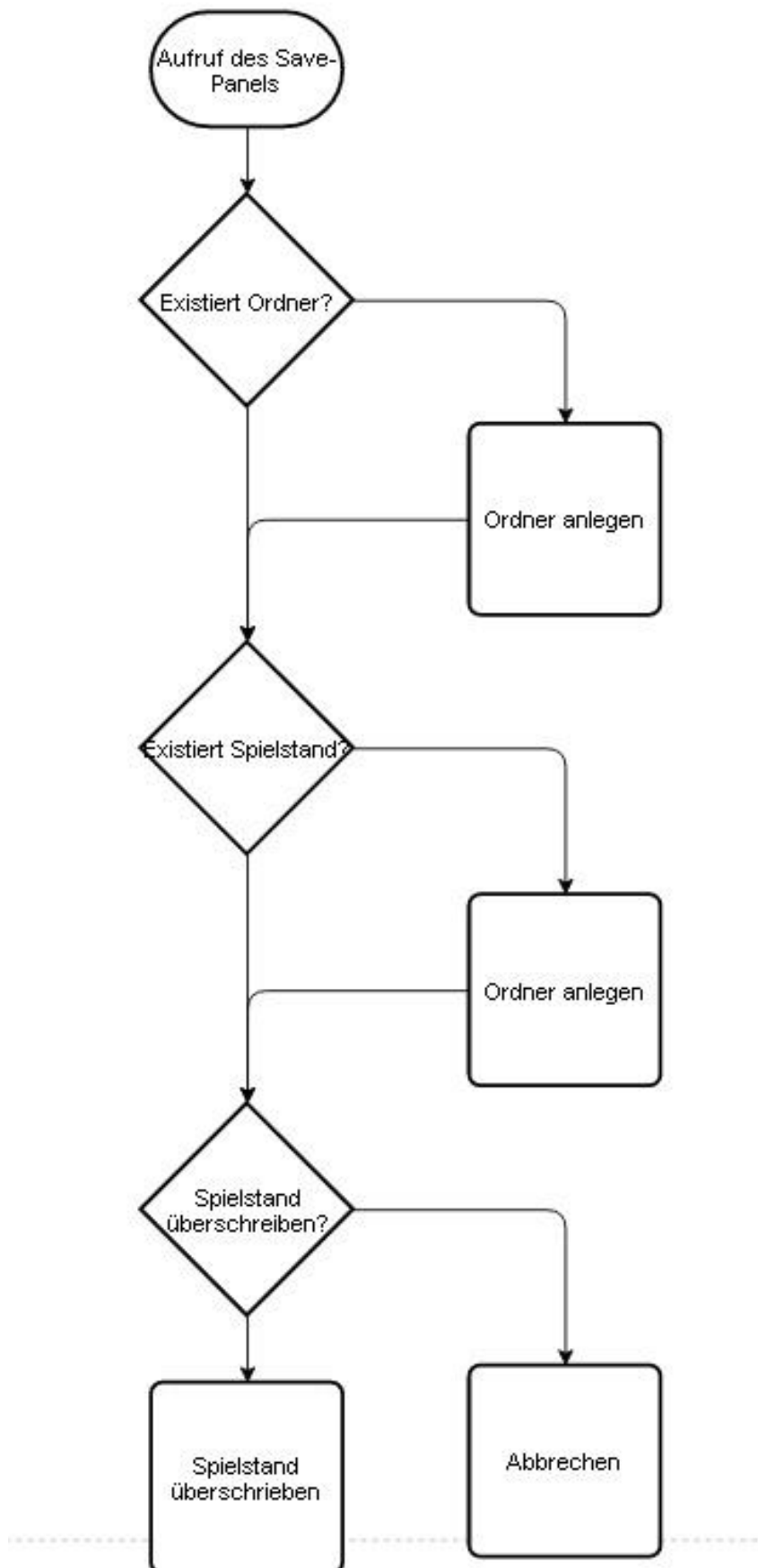
A.6 Fließdiagramm Start-Panel Abfrage Anzeige Lade-Button

Abbildung 5: Fließdiagramm Start-Panel Abfrage Anzeige Lade-Button



A.7 Fließdiagramm Save-Panel Speichern

Abbildung 6: Save-Panel Speichern



A.8 Pflichtenheft

Einführung:

Übernahme des Kundenprojekts aus Lastenheft. Ziel ist die Erstellung eines einfachen textbasierten Rollenspiels mit grafischer Oberfläche.

Ist-Zustand:

Das Spiel soll sich an einem bereits existierenden Rollenspiel orientieren, welches keine Oberfläche besitzt.

Durchführungsplanung:

Für das Projekt wurde eine detaillierte Projektplanung entworfen. Anhand der werden die einzelnen Schritte erläutert und der Aufwand zeitlich bestimmt. Projektdauer beträgt rund 79,5 h, was etwa 8 Wochen Zeitaufwand bedeutet. Da für dieses Projekt täglich nur Begrenzte Zeit zur Verfügung steht. Die Kosten belaufen sich auf rund 12.600,- €.

Aufgrund der Einschränkungen wird erwartet, dass die geplanten Stunden nicht ausreichen. Eine Steigerung der Arbeitsstunden auf 100 Stunden und dementsprechend höheren Ausgaben sind zu erwarten.

Angebot: Aufgrund der Risiken werden Kosten in Höhe von 15.000,- € angesetzt, die tatsächlichen Ausgaben werden in der Abschließenden Rechnung angegeben.

Nachtrag: Änderungen des Auftrags sind zu protokollieren und haben zusätzliche Kosten und Aufwendungen zur Folge.

A.9 Abnahmeprotokoll

Abnahmeprotokoll für das Projekt Growth V2

Beteiligte Personen:

Jellef Abbenseth

Ort: Pfinztal

Datum: 02.12.2020 17:00 Uhr

Das Spiel wurde vollumfänglich getestet, dabei hat es den im Lasten- und Pflichtenheft beschriebenen Anforderungen entsprochen.

Somit wird das Projekt vom Kunden als erfolgreich abgeschlossen erklärt.