Pflichtenheft

Project: *GlowingBroccoli*

Version 3.3

Änderungsgeschichte

Datum	Version	Autor	Beschreibung
2020-10-05	1.0	Robertz J.	Dokument erstellt
2020-10-19	2.0	Robertz J.	Einleitung und allgemeine Beschreibung bearbeiten
2020-10-19	2.1	Wegmann S.	Funktionale Anforderungen bearbeiten
2020-10-19	2.2	Roy J.	Dokument bearbeiten
2020-10-19	2.3	Pross N.	Dokument bearbeiten
2020-10-22	3.0	Pross N.	Dokument in LaTeX gesetzt
2020-10-23	3.1	Pross N.	Use Cases bearbeiten
2020-10-24	3.2	Roy J.	Dokument bearbeiten
2020-10-24	3.3	Roy J., Pross N.	Use Cases bearbeiten

Unterschriften

Pross Naoki _____

Robertz Joanne _____

Roy Jérôme _____

Wegmann Steven _____

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Zweck

Im vorliegenden Dokument sind die Anforderungen definiert, welche im Projekt “Glowing-Broccoli” umgesetzt werden müssen. Es beschreibt den Auftrag zwischen Auftraggeber und Auftragnehmer. Der Ausdruck Pflichtenheft ist hier im Sinne der IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE Std 830-1998 verwendet. Die dort definierte Requirements Specification beinhaltet sowohl die Benutzeranforderungen (Lastenheft gemäss DIN 69901-5) als auch Realisierungsvorgaben an die Entwicklungsgruppe (Pflichtenheft gemäss DIN 69901-5).

1.2 Produküberblick

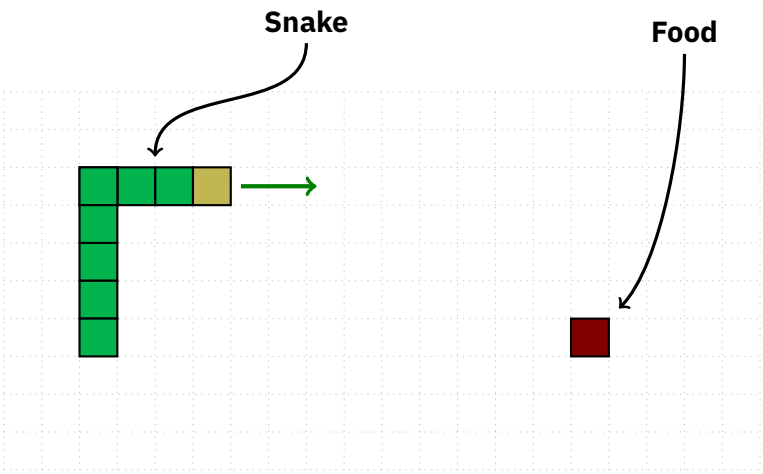

Abbildung 1: Beispiel dafür, wie die grafische Oberfläche für das Use-Case Play aussehen sollte.

Unser Team hat sich dafür entschieden das klassische “Snake” Spiel zu programmieren. Im Spiel sind Sie eine Schlange, deren Ziel es ist, sich auf einer Karte zu bewegen und die zufällig auftauchende Nahrung zu essen. Wenn eine Nahrungseinheit gegessen wird, dehnt sich die Schlange aus und der Spielpunkt wird erhöht. Ziel des Spiels ist es, die höchstmögliche Punktzahl zu erreichen. Der Benutzer verliert sein Spiel, wenn der Kopf der Schlange einen festen Gegenstand (die Wände der Spielkarte) berührt oder wenn der Kopf einen Teil seines Körpers frisst. In dieser Version der Schlange wird es verschiedene Ebenen geben. Jede Stufe hat eine Siegpunktzahl zu erreichen, um zur nächsten überzugehen. Wenn der Spieler die zuletzt vorgeschlagene Stufe erreicht, hat er das Spiel gewonnen.

1.3 Definitionen, Akronyme und Abkürkungen

Widget Diese Umgebung wird innerhalb des Fensters aufgebaut und dient um einen dynamischen Verlauf, wie z.B. ein Spiel, aufzubauen.

2 Allgemeine Beschreibung

2.1 Systemübersicht

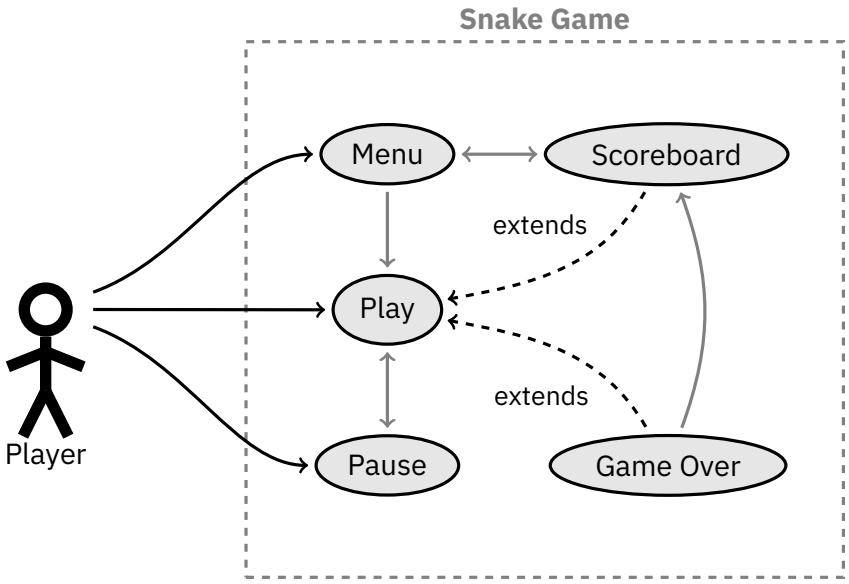

Abbildung 2: Use-case Diagramm

Das Spiel Snake verfügt über eine grafische Schnittstelle, mit der Sie interagieren können. Der Benutzer entscheidet, wann das Spiel durch Drücken der Starttaste gestartet wird. Zu Beginn des Spiels bewegt sich die Schlange in eine bestimmte Richtung und es erscheint eine zufällige Portion Nahrung. An diesem Punkt wartet das Spiel auf Eingaben des Benutzers, um die Richtung zu ändern. Der Benutzer kann die Richtung der Schlange mit den Pfeilen oder mit den `WASD`-Tasten wählen.

2.2 Produktfunktionen

- Das Spiel wird mittels einem Startknopf eröffnet.
- Die Schlange muss mittels den vier Pfeiltasten (oder `WASD`) bewegt werden können.

- Das herumliegende Essen muss sich der Schlange anschliessen als Punkt, damit sie sich nach jedem Bissen auch vergrössert.
- “Game Over” muss erscheinen, wenn die Schlange aus einer bestimmten Abbruchsbedingung stirbt.
- Ein Scoreboard zeigt dem Benutzer die Ingame-Punkte der vorherigen Spiele an.

2.3 Benutzereingenschaften

Dieses Projekt soll Benutzergruppen ansprechen die gerne ihre Zeit in einen lustigen Klassiker investieren möchten. Eine Altersbeschränkung gibt es hier nicht. Auch Vorkenntnisse im Bereich der Informatik werden zum spielen nicht gebraucht.

2.4 Einschränkungen

Keite.

2.5 Annahmen und Abhängigkeiten

Die Software soll auf allen der drei Hauptplattformen, nämlich Windows 10, MacOS und Linux laufen. Dies kann durch C++ als Programmiersprache und das plattformübergreifende Qt Framework erreicht werden. Von der User Hardware wird erwartet, dass sie die Fähigkeit hat einfache Grafiken darzustellen und über eine moderne grafische Umgebung verfügt.

2.6 Priorisierung der Anforderungen

Die Funktionen, welche in §2.2 ersichtlich sind, beschreiben die Muss-Anforderungen welche benötigt werden, um den minimalen Zweck der Software zu erfüllen. Als Soll-Anforderungen wird die Implementation von optisch schöneren Grafiken beschrieben. Die Wunsch-Anforderung beinhaltet das Einfügen von weiteren Funktionen im Spiel, wie zum Beispiel verschiedene Früchte oder die Implementation eines “Scoreboards”.

3 Externe Schnittstellen

Es gibt keine speziellen externen Schnittstellen für diese Software. Sie benötigen lediglich eine Tastatur und einen Bildschirm.

4 Funktionale Anforderungen

4.1 Actors

Actors	Beschreibung
Player	Die Eingaben des Users beeinflussen den Verlauf des Spiels.

Tabelle 1: Liste der Actors

4.2 Kurzbeschreibung der Use Cases

Use Case	Beschreibung
Menu	Der Spieler kann einen Namen eingeben und wählen, ob er spielen oder die *Scoreboard* sehen möchte.
Play	Das Spiel *Snake* spielen.
Pause	Beim betätigen der Systemtaste P oder ESC wird das Spiel pausiert. Beim erneuten drücken, geht das Spiel weiter.
Game Over	Der Spieler hat das Spiel verloren, sein Punktestand wird auf dem Bildschirm angezeigt.
Scoreboard	Es wird eine Liste der früheren Spiele desselben oder eines anderen Spielers angezeigt.

Tabelle 2: Kurzbeschreibung der Use Cases

4.3 Use Case Menu

Vorbedingungen Es gibt eine grafische Umgebung und das Programm muss laufen.

Nachbedingungen Keine.

Nicht-Funktionale Anforderungen Keine.

Haputszenario Der Benutzer wird angefordert eine Entscheidung zu treffen, das Spiel starten, ein Username wählen, das *Scoreboard* sehen oder das Programm schliessen.

Unterszenarien Keine.

Fehlerszenarien Es gibt keine grafische Umgebung.

Regeln Keine.

Anmerkungen Keine.

Beispiele Keine.

4.4 Use Case Play

Vorbedingungen Die Tastatur muss vorhanden sein.

Nachbedingungen Keine.

Nicht-Funktionale Anforderungen Keine.

Haputszenario Das *Snake* Spiel lauft.

Unterszenarien Keine.

Fehlerszenarien Es ist keine Username oder Tastatur vorhanden.

Regeln Die Schlange wird mit der Tastatur gesteuert. Die Schlange darf sich nicht aus dem sichtbaren Bereich entfernen. Es ist der Schlange auch nicht erlaubt, ihren eigenen Körper zu durchqueren. Wenn beides geschieht, kommt es zu einem Game Over.

Anmerkungen Keine.

Beispiele Keine.

4.5 Use Case Pause

Vorbedingungen Es muss ein laufendes Spiel geben.

Nachbedingungen Das Spiel ist pausiert.

Nicht-Funktionale Anforderungen Bei Betätigung des Knopfes soll der User mit möglichst wenig Verzögerung eine Pause der Spielpartie erzwingen können, die er aber später weiterspielen kann.

Haputszenario Ein Menu gibt die möglichkeit zu fortfahren oder zum Hauptmenu zu zurückkehren.

Unterszenarien Keine.

Fehlerszenarien Es gibt kein laufende Spiel.

Regeln Wenn das Spiel lauft, wird es unterbrochen. Wenn das Spiel bereits angehalten ist, sollte es wieder aufgenommen werden. Es sollte auch möglich sein, das Spiel zu beenden und zum Hauptmenü zurückzukehren.

Anmerkungen Keine.

Beispiele Keine.

4.6 Use Case Game Over

Vorbedingungen Es muss ein laufendes Spiel geben.

Nachbedingungen Das Spiel endet.

Nicht-Funktionale Anforderungen Keine.

Haputszenario Der Spieler wird informiert, dass er verloren hat. Sein *Score* (Punktzahl) wird auf den Bildschirm gezeigt.

Unterszenarien Keine.

Fehlerszenarien Keine.

Regeln Der Spieler kann ein Neustart anfordern, das *Scoreboard* sehen oder zum Hauptmenü zurückzukehren.

Anmerkungen Keine.

Beispiele Die Schlange frisst sich selbst.

4.7 Use Case Scoreboard

Vorbedingungen Programm muss laufen.

Nachbedingungen Keine.

Nicht-Funktionale Anforderungen Keine.

Haputszenario Eine Liste der vorherige Spiele wird auf den Bildschirm gezeigt.

Unterszenarien Keine.

Fehlerszenarien Keine.

Regeln Keine.

Anmerkungen Keine.

Beispiele Keine.

5 Sonstige Anforderungen

Die grafische Qualität des Spiels wird einfach und begrenzt sein, da wir die Grundfunktionen von Qt zum Zeichnen der verschiedenen Komponenten verwenden. Diese Software wurde für die aktuelle Version der Betriebssysteme Linux, Windows 10 und Mac OS Catalina entwickelt. Es wird keine Updates für neuere Betriebssysteme geben.

Der Benutzer muss sich nicht registrieren, um das Spiel spielen zu können. Es gibt jedoch die Idee, vergangene Spiele lokal zu speichern, so dass Sie die in verschiedenen Spielen erzielten Ergebnisse vergleichen können.

Referenzen

[1] IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE Std 830-1998