

## **Verflixte 7 Bericht**

Gruppe: Asmen, Mohamad, Noah, Philipp, JP

Für unser Projekt "Die Verflixte 7" haben wir uns alle zusammengesetzt und zunächst darüber nachgedacht, wie wir am besten starten können.

Unser erster Schritt bestand darin, alle Spielregeln gründlich durchzugehen, um genau zu verstehen, worauf wir beim Schreiben des Programms achten müssen.

Für die erste Version war es vorgesehen, ein Objektdiagramm zu erstellen. Also setzten sich Philipp und JP an einen Tisch und begannen, das Spiel "Die verflixte 7" zu spielen, während Asmen Notizen machte, um später das Objektdiagramm zu erstellen.

Nachdem Asmen das Objektdiagramm fertig gezeichnet hatte, haben wir festgelegt, wer welche Aufgaben übernimmt. Wir entschieden uns dafür, dass Noah und Philipp an der Software arbeiten, JP sich um die GUI kümmert und Mohammad und Asmen das Design unseres Spiels gestalten.

Während Noah und Philipp damit begannen, das Spiel zu entwickeln, fingen die anderen an, die ersten Skizzen anzufertigen, um eine grobe Vorstellung davon zu bekommen, wie die GUI aussehen könnte. Wir haben uns dazu entschieden, die GUI schon am Anfang zu erstellen, damit wir später nicht so viel Zeitstress haben. Wir planten einen Startbildschirm, einen Bildschirm während des Spiels und zwei Endbildschirme. Anschließend begannen wir, Bilder auszusuchen, die zum Spiel passen würden.

Für Version 1 haben die Entwickler zuerst die Klasse Topf vervollständigt. Hinzugefügt wurden getter- und setter-Methoden für die Punkte im Topf. Die Klasse Würfel wird unverändert übernommen, da sie nichts neues brauchte.

Die Klasse Spieler wird im Konstruktor erweitert mit ergänzten Parametern und Referenzvariablen.

Die Methode wuerfeln() wurde implementiert, um den Punktestand nach dem Würfeln zu ermitteln. In Version 1 werden noch die Kommentare in fachgerechter Sprache geschrieben und anschließend als Version 1 abgespeichert.

Für Version 2 haben wir die Spielregeln überprüft und diese angepasst und zusätzliche Methoden und Variablen hinzugefügt, um die Regeln umzusetzen.

Die Methoden, die wir hinzugefügt haben, waren fehlende setter- und getter-Methoden, die für die Spielregeln gebraucht werden, wie z. B. `WurfAnzahlAngaben()` oder `topfLeeren()`. Diese Methoden sind notwendig, weil das Spiel sonst im UML auch nicht spielbar ist.

Version 1 und Version 2 werden separat gespeichert, um die bisherigen Fortschritte nachvollziehen zu können. Diese beiden Versionen, die schrittweise implementiert und erweitert werden, ermöglichen uns eine strukturierte Entwicklung und eine bessere Übersicht über das Spiel. Dazu wird es vom UML-Diagramm auch noch unterstützt. Diese Art von Entwicklung sichert uns ein möglichst fehlerfreies, funktionsfähiges Spiel.

### ***Abschluss der Version 2 und Beginn der Weiterentwicklung***

Nach der Fertigstellung von Version 2 haben wir uns unmittelbar an die Planung und Entwicklung der nachfolgenden Versionen gemacht. Unser Ziel war es, die Benutzeroberfläche (GUI) weiter zu verbessern und neue Funktionen hinzuzufügen.

### ***Verbesserungen in der Benutzeroberfläche und neue Funktionen***

JP konzentrierte sich auf die Optimierung der GUI und die Implementierung neuer Funktionen. Dazu gehörte die Integration zusätzlicher Buttons und Textfelder in Version 3. Noah und Philipp überprüften währenddessen die Funktionsfähigkeit des Codes mithilfe von UML-Diagrammen. Diese Diagramme halfen uns, eine klare Vorstellung davon zu bekommen, wie die verschiedenen Elemente platziert werden sollten.

Um die Platzierung der neuen Buttons und Textfelder zu planen, fertigten wir mehrere Skizzen an. Diese halfen uns, einen groben Überblick über unsere Vorstellungen zu bekommen. JP recherchierte zudem in verschiedenen Foren, um herauszufinden, wie die Form der Buttons angepasst werden kann, um eine ansprechende und funktionale Benutzeroberfläche zu gewährleisten.

### ***Anpassung der Endbildschirme***

Parallel dazu arbeiteten Mohammad und Asmen an der Gestaltung der Endbildschirme, die den Spielern anzeigen, wer gewonnen und wer verloren hat. Sie wählten passende Bilder aus und passten sie an, indem sie überflüssige Elemente entfernte und die Bilder zurechtschnitten. Die Auflösung der Bilder wurde verbessert, um sicherzustellen, dass sie unseren Anforderungen entsprechen und eine hohe visuelle Qualität bieten.

## ***Fehlerbehebung und neue Methoden***

Noah und Philipp waren zudem damit beschäftigt, Fehler aus der vorherigen Version zu beheben und den Code zu optimieren. Sie fügten neue Methoden hinzu, darunter die Methode `Auswerten()`. Diese Methode ermöglicht es uns, zu bestimmen, ob Spieler 1 keine Sieben geworfen hat, ob er eine höhere Punktzahl erreicht hat und ob er weniger Würfe benötigt hat als Spieler 2.

## ***Implementierung von Spielmechanismen***

Zusätzlich wurde eine if-else Verzweigung implementiert, die sicherstellt, dass der Gewinnertopf geleert wird und der Gewinn dem Konto des siegreichen Spielers gutgeschrieben wird. Dies ist ein wichtiger Schritt zur Verbesserung der Spielmechanik und zur Sicherstellung eines reibungslosen Spielerlebnisses.

## ***Abschluss der Version 3 und Übergang zu Version 4***

Nach der Beendigung von Version 3 haben wir deutliche Fortschritte erkennen können, sowohl in der Gestaltung der Benutzeroberfläche (GUI) als auch in der Optimierung des Codes. Diese Version brachte uns unseren ursprünglichen Skizzen und Vorstellungen ein großes Stück näher.

## ***Anpassungen und neue Funktionen in Version 4***

Mit dem Start von Version 4 hat JP die Anordnung der grafischen Elemente weiter verfeinert. Der Würfel-Button wurde mit einem individuellen Logo versehen, was nicht nur die Ästhetik, sondern auch die Benutzerfreundlichkeit verbessert. Ein weiteres wichtiges Update war die Gestaltung der Buttons im Startbildschirm, die nun transparent sind. Im Startbildschirm haben wir zudem die Auswahl zwischen "2 Spieler" und "Computer" implementiert. Diese Auswahlmöglichkeiten bieten den Nutzern unterschiedliche Spielerfahrungen. Die Transparenz der Buttons und Textfelder wurde auch im Spiel selbst übernommen, um ein einheitliches Erscheinungsbild zu gewährleisten.

## ***Implementierung der Spielmodi***

Die neuen Auswahlmöglichkeiten im Startbildschirm erforderten die Entwicklung eines entsprechenden Codes. Für den Modus "2 Spieler" wurde eine Funktionalität entwickelt, die es zwei Personen ermöglicht, abwechselnd gegeneinander zu spielen. Im Modus "Computer" spielt ein menschlicher Spieler gegen einen konfigurierten Computergegner. Hierbei haben wir zunächst den Computergegner erstellt und ihn in der Klasse des Schiedsrichters integriert. Dies ermöglichte eine kontrollierte und faire Interaktion zwischen dem menschlichen Spieler und dem Computer.

## ***Technische Herausforderungen und Lösungen***

Die Implementierung der transparenten Buttons und Textfelder stellte uns vor einige technische Herausforderungen. Wir mussten sicherstellen, dass die Transparenz

keine negativen Auswirkungen auf die Lesbarkeit und Benutzerinteraktion hat. Durch sorgfältige Anpassungen und Tests konnten wir diese Probleme erfolgreich lösen.

### ***Abschluss der finalen Version***

In dieser Version wurde die Software vollständig in die grafische Benutzeroberfläche (GUI) integriert und alle vorgesehenen Buttons und Textfelder wurden erfolgreich eingebunden.

Während der Entwicklung der finalen Version haben wir umfassende Tests durchgeführt, um letzte Probleme zu identifizieren und zu beheben. Dies umfasste sowohl kleinere grafische Fehler als auch funktionale Probleme.

### ***Welches Software-Entwicklungsmodell passt am besten zu unserer Vorgehensweise?***

Unsere Gruppe hat am besten nach einer abgeänderten Version des Scrum-Entwicklungsmodells gearbeitet. Im Scrum-Entwicklungsmodell wird in Sprints gearbeitet, das bedeutet, Entwicklungen werden schrittweise verfeinert. Wir haben uns einmal in der Woche in der Schule für eine Doppelstunde zusammengesetzt und zusammen an der Software gearbeitet. Dazu hatten wir die Gruppe in verschiedene Abteilungen eingeteilt, wie z.B. die Developer (Noah, Philipp), Design (Mohammad, Asmen) und den Gruppenleiter (JP), der auch an dem GUI gearbeitet hat.

### ***Wie ist die Arbeit verlaufen?***

Die Arbeit verlief insgesamt sehr gut. Wir konnten unsere Abgaben pünktlich erreichen und die Anforderungen des Projekts erfolgreich umsetzen. Jeder hat immer an dem weitergearbeitet, was erledigt sein musste.

### ***Was hat gut funktioniert?***

- Kommunikation und Zusammenarbeit
  - Wenn etwas nicht funktioniert hat, haben wir uns gegenseitig geholfen, um das Problem zu lösen.
- Flexibilität
  - Wir haben die Seite GitHub verwendet, mit der jeder schnellen Zugriff auf das Programm hatte, falls etwas benötigt wurde.

### ***Was hat nicht gut funktioniert?***

- Anforderungsmanagement
  - Es gab manchmal Unklarheiten bei den Anforderungen, die zu Stress führten.
- Ressourcenmanagement
  - Es war nicht immer optimal, wenn ein paar Teammitglieder krank waren, was nicht schlimm ist. Aber das führte zu Unklarheiten bei der nächsten Stunde, z.B. was wann gemacht wurde.

### ***Was würden wir beim nächsten Projekt genauso und was anders machen (bezogen auf Organisation und Vorgehen)?***

#### **Genauso machen:**

- Die Vorgehensweise beibehalten, sich in der Woche einmal für eine Doppelstunde zu treffen, war sehr effektiv.
- Regelmäßige Abgaben haben uns geholfen, das Programm Stück für Stück zu entwickeln und möglichst fehlerfrei zu halten.
- Unsere Kommunikation war sehr gut. Wir haben miteinander geredet, wenn Komplikationen auftraten. Dazu haben wir eine WhatsApp- und eine Teams-Gruppe genutzt, um von zuhause aus zu kommunizieren.

#### **Anders machen:**

- Mehr Planung. Damit ist gemeint, dass wir, bevor wir anfangen, uns mehr Gedanken über unser Programm machen und detaillierter wissen, was unser nächster Schritt ist.

- Mehr Verfügbarkeit von Teammitgliedern. Krankheiten kann man nicht vermeiden, aber es wäre besser, wenn alle immer anwesend wären.

## **Fazit**

Insgesamt war das Projekt ein großer Erfolg, und die Anwendung des Scrum-Entwicklungsmodells erwies sich als äußerst passend für unsere Gruppe. Die wöchentlichen Treffen für eine Doppelstunde ermöglichten bessere Fortschritte und halfen uns, den Überblick zu behalten und Fehler frühzeitig zu erkennen. Die Flexibilität innerhalb unserer Gruppe war sehr gut, da jeder jederzeit über WhatsApp oder Teams erreichbar war, was die Kommunikation erleichterte und schnelle Problemlösungen ermöglichte. Die Aufteilung der Gruppe in verschiedene Abteilungen: Entwicklung, Design und GUI-Management, war vorteilhaft. Jeder konnte seine Stärken benutzen. Die Rollenverteilung minimierte Stress und Konflikte, da jeder genau wusste, welche Aufgaben er zu erledigen hatte.

Unsere Kommunikationsstrategie mit regelmäßigen Updates und Meetings, persönlich und virtuell, hielt alle Teammitglieder auf dem gleichen Stand und förderte die Motivation. Rückblickend gibt es sicherlich Punkte, die wir beim nächsten Projekt verbessern können, wie eine detailliertere Planung im Voraus und eine bessere Sicherstellung der Verfügbarkeit aller Teammitglieder.

Dennoch überwiegen die positiven Erfahrungen und Erfolge. Die Anwendung des Scrum-Entwicklungsmodells und die Rollenverteilung waren entscheidende Faktoren für das Gelingen des Projekts. Dieses Projekt hat uns gezeigt, wie wichtig Teamarbeit, Kommunikation und Struktur in der Softwareentwicklung sind und hat uns wertvolle Erfahrungen für Zukunft gegeben.