

Pflichtenheft

Projektbezeichnung	Projekttag 2020 – Space Invaders
Projektgruppe	Patrick Matern, Monique Vanessa Ostermann, Erik Elges, Joy Webster
Bearbeitungszeitraum	20.04.2020 bis 21.04.2020

1. Einleitung

1.1. Ziel und Zweck des Dokuments

Dieses Pflichtenheft beschreibt die Umsetzung des Projektes im Rahmen der Projekttag 2020 an der Carl-Severing Berufsschule. Im Zuge des Projektes soll das Spiel Space Invaders programmiert werden.

1.2. Ausgangssituation/Ist-Zustand

Für die Ausbildung zum Fachinformatiker für Anwendungsentwicklung finden im zweiten Halbjahr des ersten Ausbildungsjahres die Projekttag statt, in denen ein eigenständiges Projekt geplant und umgesetzt werden soll. Das Thema des Projektes soll selbst gewählt und die erforderlichen Schritte zur Umsetzung geplant werden. Der Durchführungszeitraum betrug planmäßig den 20.04.2020 bis 21.04.2020. Aufgrund des zu diesem Zeitpunkt herrschenden Kontaktverbotes im Zuge der Corona-Krise konnte der Zeitraum jedoch von den Projektgruppen selbst gewählt werden. Ziel der Projekttag ist die Aneignung von Fähigkeiten und die Vorbereitung auf das Abschlussprojekt der IHK-Prüfung.

2. Ziel/Formulierung Auftrag

Von der zugehörigen Projektgruppe wurde als Thema der Projekttag das Spiel Space Invaders gewählt. Ziel ist es eine eigene Variante des bekannten Shoot-em-up Computerspiels Space Invaders von Taito zu entwickeln. Das Spiel soll in Java (vorzugsweise objektorientiert) programmiert werden und zumindest die grundlegende Funktionalität des Vorbildes besitzen. Darüber ein eigener Menü-Bildschirm, die Modifizierung über Einstellungen und eine Erweiterung des Spielprinzips im Sinne von Power-Ups geplant.

3. Technische Anforderungen

Für die Grundfunktion des Spiels soll auf einer begrenzten Fläche die Spielfigur des Nutzers im unteren Bereich positioniert werden. Auf die Figur des Nutzers kommen vom oberen Rand eine bestimmte Anzahl Gegnerfiguren zu. Um diese Bewegung zu realisieren muss abwechselnd ein Schritt nach links oder rechts und dann nach unten realisiert werden. Sowohl die gegnerischen Figuren, als auch die Figur des Nutzers muss über die Fähigkeit verfügen zu schießen. Wird eine generische Figur von einem Schuss der Spielfigur getroffen, muss diese vom Spielfeld entfernt werden. Wird die Spielfigur von gegnerischen Geschossen

getroffen, soll sie nur eine bestimmte Anzahl von Schüssen zulassen, danach muss sie ebenfalls vom Spielfeld entfernt werden. Tritt dieser Fall ein, hat der Nutzer das Spiel verloren.

Eine weitere Niederlage-Bedingung ist, wenn die Gegner den unten Spielfeldrand erreicht haben.

Entfernt der Nutzer mit seiner Spielfigur alle Gegner vom Spielfeld endet das Spiel ebenfalls, allerdings hat der Spieler in diesem Fall gewonnen.

Die Spielfigur des Nutzers soll sich mit der linken und rechten Pfeiltaste der Tastatur am unteren Spielfeldrand nach links und rechts bewegen können. Mit der Leertaste soll die Spielfigur einen Schuss abgeben können.

Alle genannten Anforderungen sollen funktionsfähig sein und graphisch über ein Fenster ausgegeben werden.

3.1. Erweiterte Anforderungen

Neben der genannten Grundanforderung des Projekts wurden noch weitere Anforderungen formuliert. Diese erhöhen zwar die Qualität des Spiels, gefährden aber, werden sie nicht erfüllt, nicht die grundlegende Funktionalität. Dennoch wird ihre Einhaltung angestrebt.

3.1.1. Spielmenü

Wenn die Anwendung gestartet wird soll vor dem eigentlichen Spielbildschirm ein Hauptmenü angezeigt werden. Über dieses können verschiedene Möglichkeiten ausgewählt werden. Diese Möglichkeiten unterteilen sich in den Start des eigentlichen Spieles und die Weiterführung zu einem Einstellungsmenü, in dem Optionen wie die Schwierigkeit des Spiels geändert werden kann. Vorzugsweise gibt es einen weiteren Punkt, der eine Bestenliste von bisher gespielten Spielen anzeigt.

3.1.2. Unterschiedliche Themen

Über das in 3 erwähnte sollte die Möglichkeit bestehen für das Spiel verschiedene Themes auszuwählen. Nach diesen werden dann die Spielfiguren und der Hintergrund des Spielfeldes geändert. Dazu müssen für die Figuren und den Hintergrund unterschiedliche Bilder oder Grafiken in die Anwendung eingebunden werden.

3.1.3. Power-Ups

Das Grundspiel soll über Power-Ups erweitert werden. Die Power-Ups können von der Spielfigur des Nutzers eingesammelt werden und verstärken die Fähigkeiten dieser. Mögliche Umsetzungen für Power-Ups können sein: Spielfigur kann schneller schießen oder sich bewegen, Gegner brauchen länger bis sie wieder schussbereit sind, Gegner bewegen sich langsamer, Spielfigur kann von gegnerischen Schüssen nicht getroffen werden.

3.2. Entwurfsdokumentation

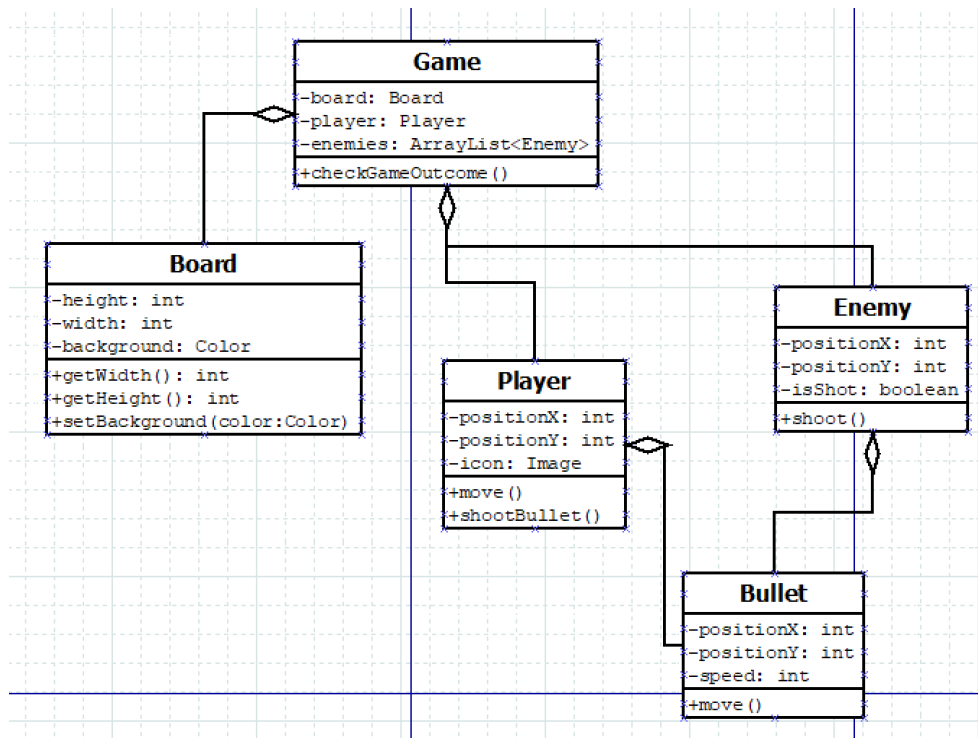


Abbildung 1: erstes UML für Projekt

Abbildung 1 zeigt die erste Idee zur objektorientierten Umsetzung der Anwendung. Das gezeigte Klassendiagramm dient nur zur ersten Übersicht einer möglichen Realisierung. Die finale Version der Anwendung kann von der hier stehenden abweichen. Möglicherweise sind noch weitere Klassen, Attribute oder Methoden notwendig.

3.3. Optionale Features

Zum momentanen Zeitpunkt sind keine optionalen Features vorgesehen.

3.4. Problemanalyse

Bei der Umsetzung des Projektes sind mehrere Probleme denkbar. Zum einen gestaltet sich der zeitliche Rahmen und die damit zusammenhängende Einschätzung schwierig. Aufgrund der noch mangelnden Erfahrung des Projektteams herrschten bereits Schwierigkeiten den Aufwand für die Realisierung korrekt einzuschätzen. Dadurch könnten zum einen zeitliche Probleme eintreten, wenn für bestimmte beispielsweise weniger Zeit eingeplant wurde als tatsächlich nötig. Zum anderen kann es auch Probleme in der Realisierung geben.

Zwar wurde im Vorfeld über die Programmierung von kleinen Computerspielen recherchiert und sich auch einige objektorientierten Beispiele angesehen, trotzdem blieb die Einschätzung des Aufwands weiterhin schwierig.

Die Realisierung könnte sich aus dem Grund schwierig gestalten, weil das Projektteam noch keine voll fundamentierten Erfahrungen in der Programmierung besitzt. Weiterhin arbeitet

jedes Mitglied im Berufsalltag mit einer anderen Programmiersprache, weswegen es zusätzlichen Einarbeitungsaufwand geben könnte.

Um diesem Punkt etwas entgegen zu wirken wurde sich für die Programmiersprache Java entschieden, da alle Mitglieder des Teams zumindest über Grundkenntnisse verfügen und diese Sprache auf in der Berufsschule verwendet wird. Zusätzlich arbeitet ein Mitglied auch im betrieblichen Umfeld mit dieser Sprache. Mit Java sollte daher die Summe der Defizite am niedrigsten gehalten werden.

Denkbar wären auch Schwierigkeiten bei der grafischen Umsetzung des Spieles. Bisher hat kaum ein Mitglied des Projektteams grafische Anwendung programmiert. Möglich wären deshalb beispielsweise Probleme, dass die eigentliche Bewegung der Spielelemente nicht korrekt angezeigt wird oder ähnliches. Hier ist eventuell auch mit zusätzlicher Einarbeitungszeit zu rechnen.

4. Rahmenbedingungen

4.1. Zeitplan

In der nachfolgenden Tabelle sind die wichtigsten Termine mit Datum festgehalten.

Termin	
16.03.2020	Abgabe Projektantrag & Zeitplanung
20.04.2020	Erster Projekttag
24.01.2020	Zweiter Projekttag
03.05.2020	Abgabe des fertigen Projektes

Zusätzlich zu den Terminen wurde für die beiden Projektstage eine detaillierte Zeitplanung mit Arbeitspaketen aufgestellt.

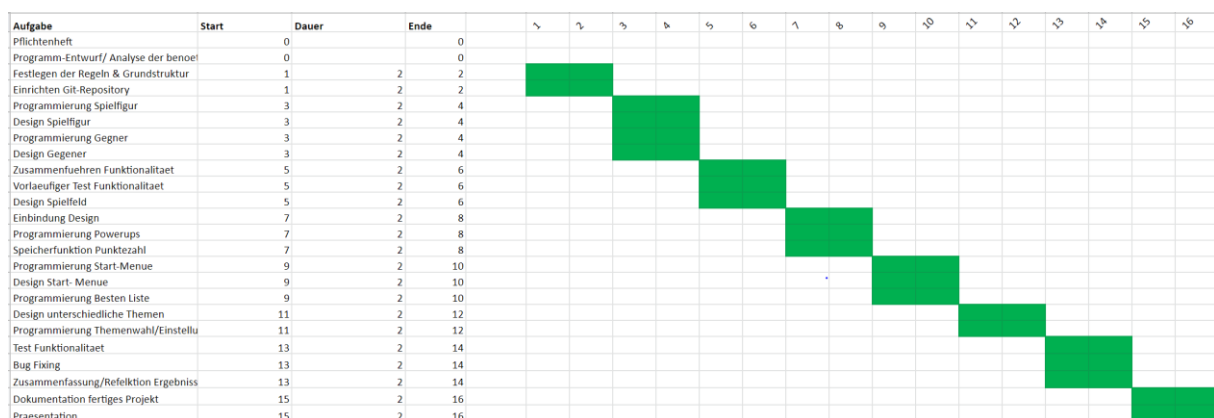


Abbildung 2: Gantt-Planung für Projekt

Die Zeiteinheiten stehen für die Schulstunden in denen das Projekt in seiner ursprünglichen Planung durchgeführt werden sollte (1 = erste Schulstunde der beiden Projektstage, 2 = zweite Schulstunde der beiden Projektstage, usw.)

4.2. Erforderliches Material

Für das Projekt wird kein zusätzliches Material benötigt. Als Programmiersprache wurde Java ausgewählt. Für die Realisierung der Anwendung wird deshalb ein Rechner mit entsprechender IDE und einem JDK benötigt. Die Anwendung soll mit Maus und Tastatur gesteuert werden, weiterführende Peripherie-Komponenten sind deshalb nicht notwendig.

5. Glossar

Im Rahmen des Pflichtenheftes müssen keine fachspezifischen Begriffe erläutert werden.