

Dokumentation

Fachinformatiker Anwendungsentwicklung

Thema

Tower Defense

Prüfungsteilnehmer: Ricardo Liebig

Betrieb: cbm GmbH Bremen
Wegesende 3-4
28195 Bremen

Projektbetreuer: Sven Lilienthal

Durchführungszeitraum: 12.05.2025 bis 25.05.2025

Inhaltsverzeichnis

1	Einleitung	2
2	Projektbeschreibung.....	2
2.1	Projektumfeld	2
2.2	Organisatorische Vorgaben	2
2.3	Ziel des Projektes	2
2.4	Projektabgrenzung	3
2.5	Projektschnittstellen	3
3	Projektplanung	4
3.1	Ist-Analyse	4
3.1.1	Netzwerkplan Ist-Zustand	4
3.2	Soll-Planung	5
3.3	Projektablaufplan.....	5
3.4	Personalplanung	6
4	Realisierung	6
4.1	Technische Umsetzung	6
4.2	Spielmechanik	7
4.3	Manuelles Testing	7
5	Projektabschluss	8
5.1	Übergabe	8
5.2	Fazit und Ausblick	8
A	Anhang.....	8
A.1	UML-Klassendiagramm	8
A.2	Gantt Diagramm.....	9
A.3	Mockup.....	9
A.4	Nutzwertanalyse	10

1 Einleitung

Das Ziel dieses Projekts war die Entwicklung eines einfachen Tower-Defense-Spiels unter Verwendung der Programmiersprache Java und der Swing-Bibliothek für die grafische Benutzeroberfläche. Es sollte eine funktionale Basisversion eines Tower-Defense-Spiels erstellt werden, das es dem Spieler ermöglicht, Türme zu platzieren, Gegnerwellen abzuwehren und das Spielziel zu erreichen. Alle Projektphasen, von der Konzeption über die Entwicklung bis hin zu den Tests, wurden von einer Person durchgeführt. Die Wahl von Java und Swing als Technologien ermöglichte eine effiziente Entwicklung und die Nutzung von bereits vorhandenen Kenntnissen.

2 Projektbeschreibung

2.1 Projektumfeld

Das Projekt wurde auf einem Windows-PC in einer lokalen Entwicklungsumgebung umgesetzt. Für die Entwicklung kam Java 11 in Verbindung mit Swing zum Einsatz. Als IDE wurde IntelliJ IDEA verwendet, um eine effiziente und produktive Entwicklungsumgebung zu gewährleisten. Die grafische Benutzeroberfläche wurde vollständig mit Swing gestaltet, was eine einfache Handhabung und schnelle Anpassung der Benutzeroberfläche ermöglichte.

2.2 Organisatorische Vorgaben

Das Projekt wurde im Zeitraum vom 12.05.2025 bis 23.05.2025 durchgeführt und durfte 80 Arbeitsstunden nicht überschreiten.

2.3 Ziel des Projektes

Ziel dieses Projekts war die Entwicklung eines Tower-Defense-Spiels, bei dem die grundlegenden Mechaniken eines Tower-Defense-Genres umgesetzt werden. Dabei sollten neben der Platzierung von Türmen und dem Abwehren von Gegnerwellen auch die Ergebnisse in einer SQL-Datenbank gespeichert werden. Der Fokus lag auf der Integration der

Highscore-Funktionalität, bei der nach jedem Spiel (Sieg oder Niederlage) der Spielname und der Highscore gespeichert werden, um den Spielern eine Rückmeldung über ihre Leistungen zu geben.

2.4 Projektabgrenzung

Einzelne Handlungsschritte im Detail sind aus zeitlichen Gründen nicht Bestandteil dieses Projekts. Der Fokus liegt auf der Konzeption und Umsetzung der Kernfunktionen eines Tower-Defense-Spiels sowie der technischen Einbindung einer Highscore-Speicherung über eine SQL-Datenbank. Die folgende Liste gibt einen Überblick über den groben Projektablauf und zeigt, welche Punkte Bestandteil des Projekts waren und welche nicht:

- Entwurf und technische Richtlinien für das Spielsystem definieren
- Aufbau der Swing-Oberfläche zur Interaktion mit dem Spiel
- Implementierung der Spiellogik mit Bewegung, Angriff und Ressourcenverwaltung
- Anbindung einer SQL-Datenbank zur Highscore-Speicherung
- Implementierung der Benutzerinteraktion
- Durchführung eines Testdurchlaufs inklusive Datenbankeintrag
- Weitere Features wie Level-Auswahl, dynamische Gegnerpfade oder Online-Ranking sind nicht Bestandteil des Projekts

Damit bleibt das Projekt in einem klar umrissenen Rahmen. Es liefert eine funktionierende Basisversion, die bei Bedarf in zukünftigen Arbeitsschritten erweitert werden kann.

2.5 Projektschnittstellen

Eine zentrale Schnittstelle im Projekt ist die SQL-Datenbank, die mit der Java-Anwendung über JDBC kommuniziert. Nach jedem abgeschlossenen Spiel wird entweder der Spielname und der Highscore bei einem Sieg oder einer Niederlage in der Datenbank gespeichert. Die Datenbankinteraktionen

umfassen das Hinzufügen von Datensätzen sowie das Abrufen und Anzeigen des Highscores. Dies geschieht über vorbereitete SQL-Abfragen, die durch die Anwendung ausgelöst werden.

3 Projektplanung

3.1 Ist-Analyse

Die Ist-Analyse beschreibt die Ausgangssituation vor Beginn des Projekts. Es lagen keine bestehenden Projektstrukturen, Quellcodes oder vorbereiteten Module vor. Die Entwicklungsumgebung musste vollständig eingerichtet werden, einschließlich:

- Installation von Java 11 SDK
- Einbindung von Swing
- Installation und Konfiguration der IDE
- Einrichtung eines lokalen MySQL-Servers zur Speicherung von Spielergebnissen
- Konfiguration eines einfachen Datenbank-Schemas mit einer Tabelle zur Speicherung von Spielernamen und Punktestand

Die Netzerkanbindung diente ausschließlich für Recherche- und Updatezwecke. Das Projekt selbst lief vollständig lokal ohne externe Abhängigkeiten.

3.1.1 Netzwerkplan Ist-Zustand

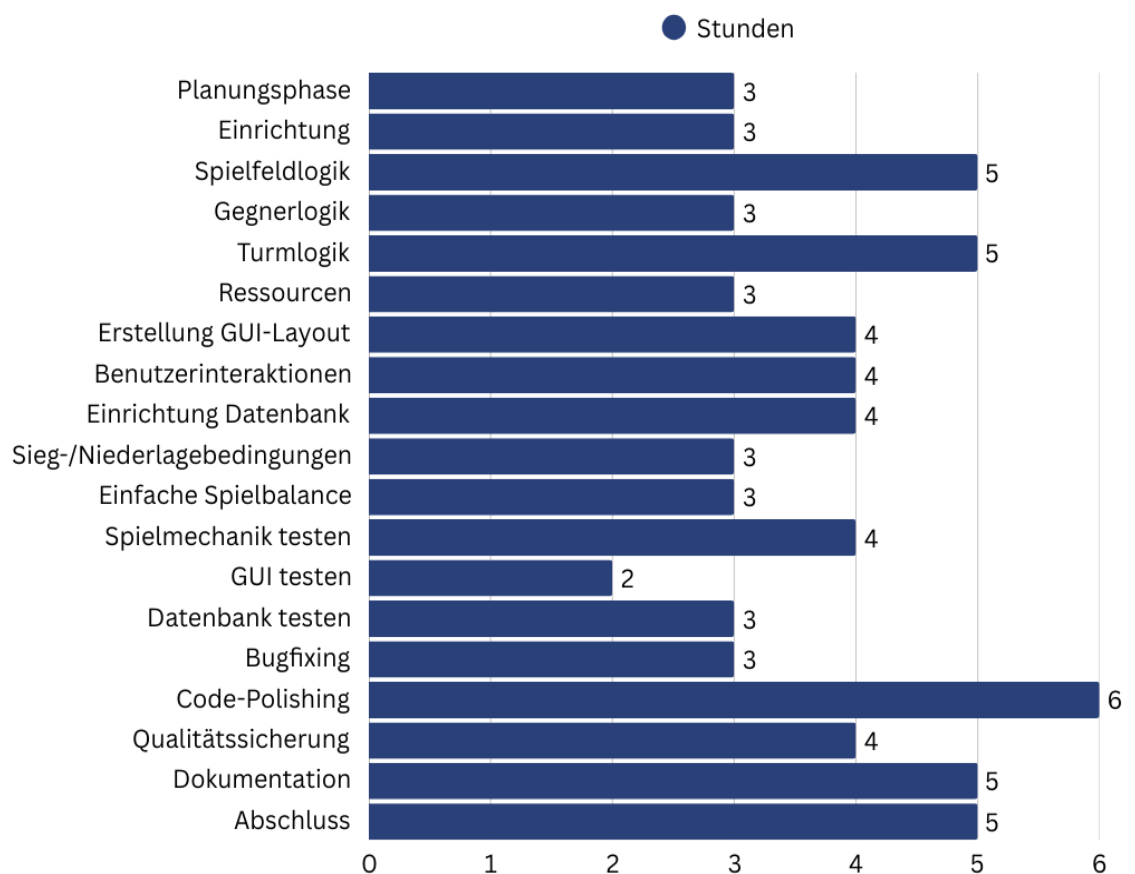
Da es sich um ein rein lokal durchgeführtes Einzelprojekt handelte, existierte keine strukturierte Netzwerkinfrastruktur. Die gesamte Entwicklungs- und Laufzeitumgebung lief auf einem einzigen Rechner. Eine Datenbankverbindung bestand lediglich lokal zur eigenen MySQL-Instanz. Es gab keine Verbindungen zu externen Servern oder Netzwerken. Ein Netzwerkplan im klassischen Sinne war daher nicht erforderlich.

Für zukünftige Erweiterungen, beispielsweise zur Bereitstellung eines Online-Highscores oder eines Multiplayer-Modus, müsste das Projekt um entsprechende Netzwerkschnittstellen ergänzt werden.

3.2 Soll-Planung

Im Rahmen der Soll-Planung wurde definiert, wie das Tower-Defense-Spiel technisch aufgebaut sein soll, welche Spielfunktionen umzusetzen sind und welche Struktur die Anwendung erhalten soll. Der Schwerpunkt lag auf der Realisierung eines funktionalen Spielflusses, bei dem Gegner in Wellen erscheinen und vom Spieler strategisch platzierte Türme abgewehrt werden. Das Spiel sollte modular aufgebaut sein, um spätere Erweiterungen zu erleichtern.

3.3 Projektablaufplan



3.4 Personalplanung

Das Projekt wurde vollständig allein durchgeführt. Es gab keine Aufgabenverteilung im klassischen Sinne. Alle Rollen wie Planung, Entwicklung, Testing und Dokumentation wurden von einer Person übernommen. Die Personalplanung war somit sehr übersichtlich:

Rolle	Zuständigkeit
Projektleitung	Zeitplanung, Projektstrukturierung, Zieldefinition
Softwareentwicklung	Implementierung der Anwendung mit Java und Swing
Datenbankverwaltung	Einrichtung der SQL-Datenbank und Anbindung via JDBC
Test-/ Qualitätssicherung	Funktionstests, Fehleranalyse, Korrekturmaßnahmen
Dokumentation	Erstellung dieser Projektdokumentation

4 Realisierung

4.1 Technische Umsetzung

Die Umsetzung des Spiels erfolgte in mehreren modularen Entwicklungsschritten. Der Fokus lag auf der Trennung von Spiellogik, grafischer Oberfläche und Datenbankanbindung, um eine wartbare und erweiterbare Codebasis zu schaffen. Die Anwendung wurde in Java entwickelt und verwendet Swing für die Benutzeroberfläche sowie JDBC für die Verbindung zur SQL-Datenbank.

Die Hauptmodule des Spiels umfassen:

- Spielsteuerung: Zentrale Steuerung der Spielmechanik
- Spielfeld: Repräsentation des Spielfelds mit logischer Rasterstruktur, Platzierungslogik für Türme und Pfadverfolgung für Gegner.
- Turm-Logik: Verwaltung der Türme inklusive Angriffsbereich, Nachladezeit.
- Gegner-Logik: Bewegung entlang eines vordefinierten Pfads, Lebenspunkte und Geschwindigkeit.

- **Benutzeroberfläche:** Gestaltung des Layouts mithilfe von Swing, Einbindung von Buttons zur Turmauswahl.
- **Datenbank-Modul:** Einfache Klasse zur Verwaltung von Verbindungen, Einfügen von Highscores und Abrufen von Daten zur Anzeige im Spiel.

4.2 Spielmechanik

Zu Beginn des Spiels startet der Spieler mit einer festen Menge an Ressourcen. Diese können genutzt werden, um Türme an vorgegebenen Positionen zu platzieren. Die Türme haben eine eigene Angriffsgeschwindigkeit und Reichweite. Die Gegner erscheinen in einer Welle und bewegen sich automatisch entlang eines festgelegten Pfads von einem Start- zu einem Zielpunkt. Wird ein Gegner nicht rechtzeitig eliminiert, erreicht er das Ziel und das Spiel wird beendet. Sobald die Lebenspunkte der Gegner auf null sinken, ist das Spiel verloren. Wird eine Welle erfolgreich abgewehrt, wird das Spiel beendet und der Spieler kann seinen Namen zum Highscore eintragen.

4.3 Manuelles Testing

Testphase	Erfolgreich?
Datenbankerstellung	Ja
Datenbankanbindung	Ja
Gegner, Türme mit Indikator, Projektile zeichnen	Ja
Lebensbalken in der Gegner Klasse	Ja
Gegner-Leben wird korrekt berechnet und nimmt Schaden	Ja
Spiel wird korrekt geupdatet	Ja
<u>move()</u> -Funktion von Gegner und Projektil	Ja
Spielereingaben werden korrekt verarbeitet	Ja
GUI-Benutzeroberfläche	Ja

5 Projektabschluss

5.1 Übergabe

Am 25.05.2025 fand die Übergabe an Herrn Lilienthal über GitHub statt. So ist es dem Dozenten möglich gewesen das Projekt ausgiebig zu prüfen.

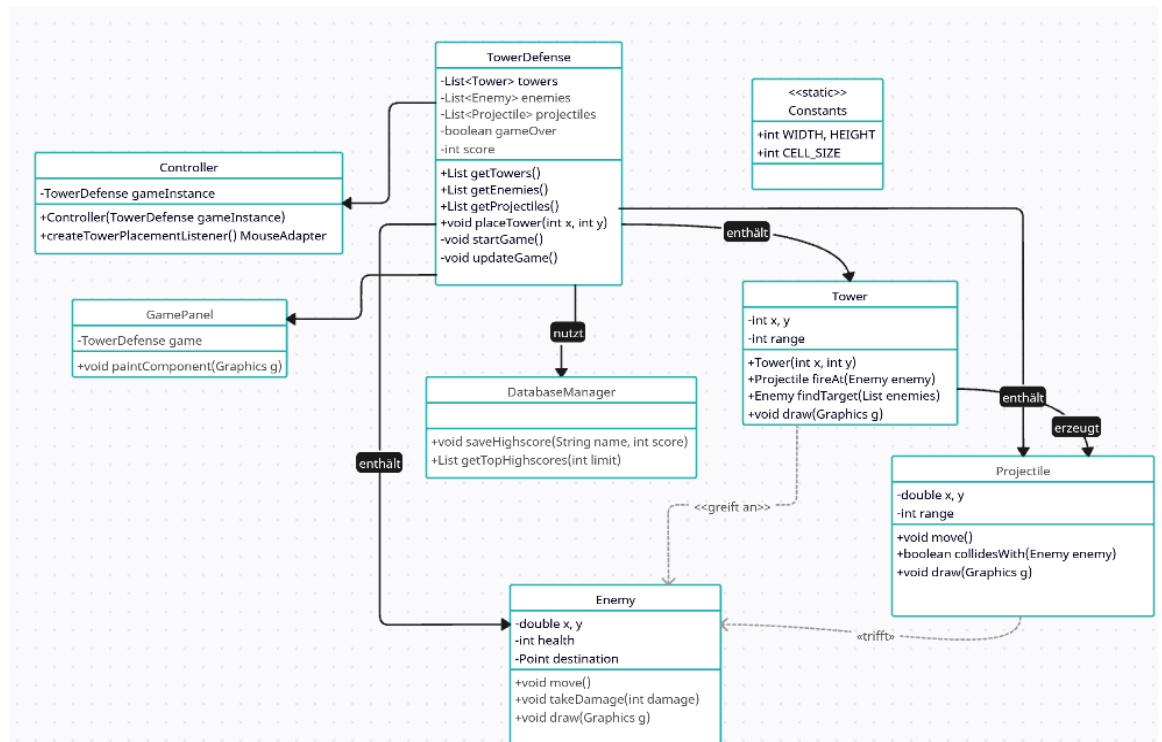
Auf diesem Wege konnte man die wichtigsten funktionalen und nicht-funktionalen Anforderungen überblicken und testen.

5.2 Fazit und Ausblick

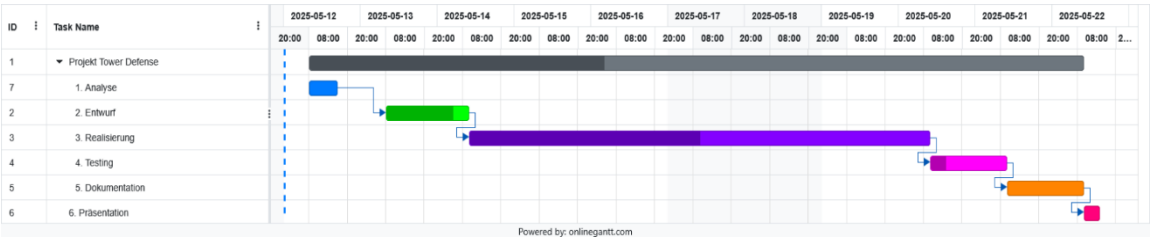
Das Projekt wurde erfolgreich abgeschlossen. Trotz Abweichungen bei den Ist-Zeiten war es möglich die angegebene Bearbeitungszeit von 80 Stunden einzuhalten. Dabei konnte ich viele Erfahrungen im Umgang mit Java und SQL-Datenbanken sammeln.

A Anhang

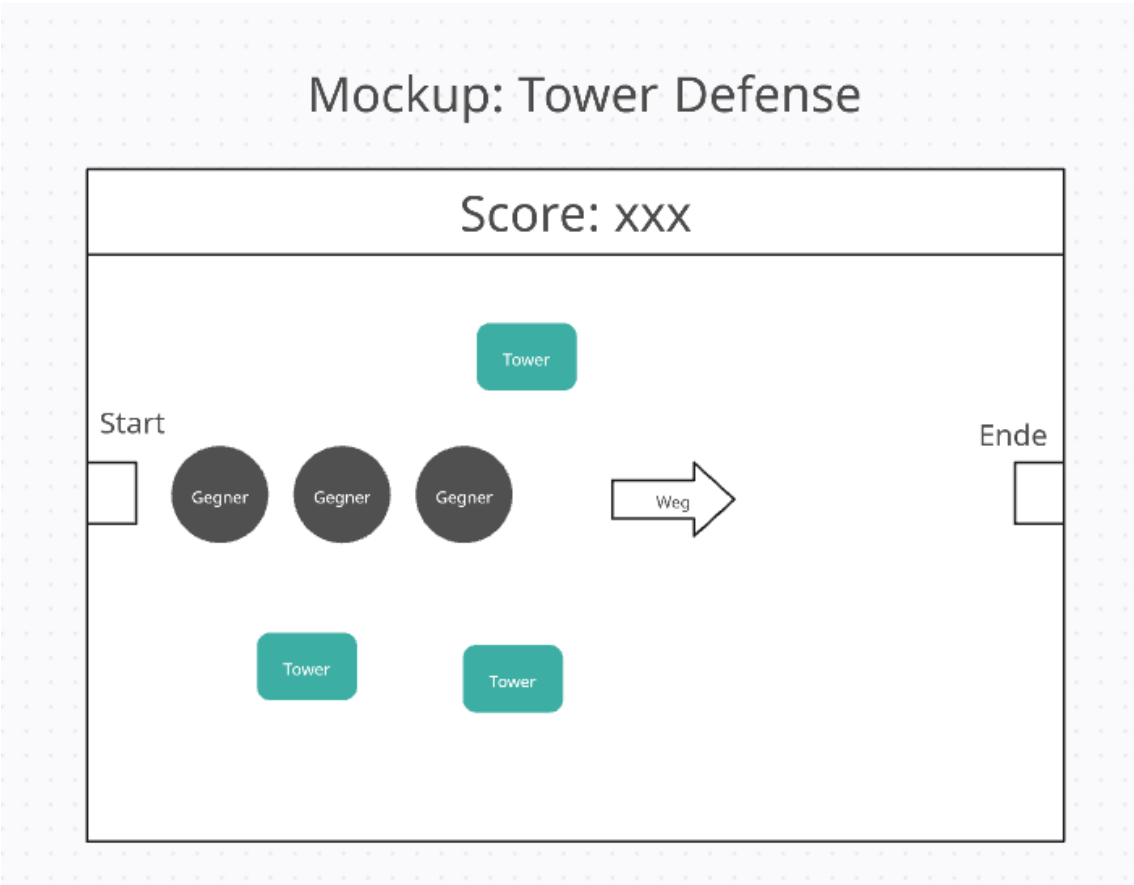
A.1 UML-Klassendiagramm



A.2 Gantt Diagramm



A.3 Mockup



A.4 Nutzwertanalyse

Kriterium	Gewichtung	Java	Python	JS
Einfachheit der Datenbankbindung	10%	9	7	6
Performance für 2D-Spiele	20%	6	4	5
Strukturierbarkeit / OOP	30%	8	6	5
Cross-Plattform-Fähigkeit	10%	8	7	9
Einarbeitungszeit	5%	5	9	8
GUI-/Grafikunterstützung	20%	6	5	6
IDE-Unterstützung	5%	6	6	6
Gesamtpunkte	100%	7,05	5,75	5,70