

## Pflichtenheft

### 1. Softwaredefinition

Das System soll ein interaktives Tower-Defense-Spiel realisieren, in dem der Benutzer Verteidigungstürme auf einer Karte platzieren kann, um herannahende Gegnerwellen abzuwehren.

Das Spiel wird mit JavaFX entwickelt und bietet eine grafische Benutzeroberfläche (GUI), die eine Echtzeit-Visualisierung der Spielfeldaktivitäten ermöglicht.

### 2. Kurzbeschreibung

Die Software simuliert ein klassisches Tower-Defense-Spiel: Gegner bewegen sich auf einem festen Pfad, und der Spieler muss Türme strategisch platzieren, um sie zu stoppen.

Ziel ist es, eine bestimmte Anzahl von Wellen zu überleben, indem man verhindert, dass Gegner das Ziel erreichen.

### 3. Funktionsumfang

Grafische Darstellung eines Spielfeldes mit einem oder mehreren vorgegebenen Pfaden für Gegner.

Platzierung von Türmen auf definierten Bauflächen durch den Benutzer.

Schießen von Türmen auf Gegner innerhalb ihrer Reichweite.

Gegnerbewegung entlang eines vordefinierten Pfades.

Lebensanzeige für die Basis (wird reduziert, wenn Gegner durchkommen).

Geld- und Punktesystem: Belohnung für besiegte Gegner.

Mehrere Wellen mit steigendem Schwierigkeitsgrad.

Pause- und Startfunktion für das Spiel.

### 4. Mussziele

Fehlerfreie Gegnerbewegung entlang des Pfades.

Platzierung und Funktionsfähigkeit von Türmen (Schussreichweite, Schussfrequenz).

Lebensverwaltung und Spielende, wenn alle Leben verloren sind.

Übersichtliche grafische Oberfläche, die Spielfeld, Türme, Gegner und Statusinformationen (z.B. Leben, Gold) klar darstellt.

### 5. Optionale Ziele

Verschiedene Turmartentypen (z.B. Schnellfeuer, Flächenschaden, Verlangsamung).

Upgrade-System für Türme.

Unterschiedliche Gegnertypen (z.B. schnelle Gegner, gepanzerte Gegner).

Soundeffekte für Schüsse und Treffer.

Speicherung von Highscores.

## 6. Benötigte Ressourcen

JavaFX AnimationTimer zur threadbasierten Ausführung der Spielschleife (Game Loop).

Java Lambda-Ausdrücke zur Definition kompakter Event-Handler und Listener.

Java Stream API für funktionale Prüfungen und Filterungen von Turm- und Gegnerlisten.

## 7. Testdaten

Verschiedene Kartenlayouts zum Testen der Bewegungspfade.

Szenarien mit steigender Gegneranzahl und -geschwindigkeit.

Tests von Türmen mit unterschiedlichen Angriffsgeschwindigkeiten und -stärken.

Simulation von vielen gleichzeitigen Gegnern zur Belastungsprüfung.

## 8. Know-How

Kenntnisse in Java und JavaFX (Canvas, AnimationTimer, EventHandler).

Erfahrung mit objektorientierter Programmierung (für Gegner-, Turm- und Spiellogikklassen).

Grundlegendes Verständnis von Spielmechanik-Design (Balancing von Türmen und Wellen).

## 9. Grafische Oberfläche

Hauptspielfeld zeigt den Pfad der Gegner und baubare Flächen.

Seitenmenü mit:

Auswahl an verfügbaren Türmen.

Anzeige von aktuellem Geld, Leben und aktueller Welle.

Buttons für Pause, Start, Turm-Upgrades.

Animationen für Schüsse, Treffer und Gegnerbewegung