



# SPIELEDOKUMENTATION

Leonhard Wegers

11BG3T



# PFLICHTENHEFT

## PRODUKTEINSATZ

Minimalistische Grafik aus Rechtecken und Kreisen hält die Grafikkartenauslastung so gering wie möglich, der Code beinhaltet keine aufwändigen Rechnungen und vermeidet unnötige Wiederholungen so gut wie möglich.

## PRODUKTFUNKTIONEN

Das Spiel verfügt über eine GUI. Die GUI ist mit JFrame Komponenten realisiert und besteht aus einem Hauptmenü im Vollbildmodus und mehreren Dialogfenstern sowie dem Spielfenster, welches Geräteunabhängig gleichgroß ist. Im Hauptmenü ist ein JButton, der ein Infofenster mit der Spielerklärung öffnet. Ein anderer JButton startet das Spiel. Eine Update Klasse sorgt in einem eigenen Thread dafür, dass das Spiel genau 60mal pro Sekunde geupdatet wird. Die Framerate ist unabhängig von der Updaterate und orientiert sich stattdessen am darstellenden Monitor. Eine regelmäßige Updaterate sichert eine gleiche Spielgeschwindigkeit, egal wie schnell das System ist, solange 60 Updates pro Sekunde möglich sind. Während dem Spiel kann der Benutzer durch Tastatureingaben und Mausklicks mit dem Spiel in Echtzeit interagieren.

## PRODUKTDATEN

In einer txt-Datei wird der erreichte Highscore gespeichert und im Hauptmenü angezeigt.

## PRODUKTLLEISTUNG

Das Hauptmenü besteht nur aus drei Buttons und einem Label, daher ist die Ladezeit quasi nicht vorhanden. Die Buttons erlauben sofortige Benutzereingabe.

Dadurch, dass das Spiel in Echtzeit abläuft und keine großen Datenmengen geladen werden müssen, dauert keine Interaktion länger als 3s.

Da Benutzerinteraktion nur auf Buttons und vordefinierten Tastatur- und Mauseingaben besteht, sind fehlerhafte Eingaben nicht möglich, bzw. das Programm macht nichts, solange erwartete Eingaben getätigt werden.

## ERGÄNZUNGEN

Das modulare System in dem Waffen geschrieben sind, erlaubt die Ergänzung von neuen Waffentypen durch eine neue Subklasse, lediglich das Schussverhalten müsste angepasst werden.

Alle Spiel-Objekte sind generell in Arraylisten gespeichert und neue können durch eine Funktion erstellt werden. Die Anzahl und Anordnung von Objekten im Level sind demnach schnell zu verändern.

Die Kollisionsklassen, erlauben Kollisionsüberprüfung für neue Rechtecke und Kreise egal für welchen Objekttypen.

# SPIELERKLÄRUNG

Nach Spielstart wird das Hauptmenü geöffnet. Dort wird der Highscore aus einer txt-Datei ausgelesen und angezeigt. Es gibt drei Knöpfe. Der Info Knopf zeigt die Spielanleitung an, der Exit Knopf beendet das Programm und der Start Knopf startet ein neues Spiel. Beim Spielstart wird eine neue Instanz der GameController Klasse erstellt, sowie in einem separaten Thread ein Update-Thread erstellt. Der Update-Thread taktet das Spiel 60mal pro Sekunde. Der GameController erstellt eine neue GUI und spawn die

Spielobjekte wie Spieler, Wände und Gegner. Über den Update Thread wird eine Update Funktion beim GameController ausgelöst, die Berechnungen für Collisionen und Bewegung ausführt. Der Spieler interagiert über die InputController Klasse mit dem Spiel. Die InputController Klasse gibt Tastatureingaben sowie Mouseclicks and das Spielerobjekt weiter. Alle Spielobjekte sind in ArrayLists in der Gamecontrollerklasse gespeichert. Der Highscore wird nach Spielende in eine txt Datei gespeichert.

# ARBEITSBERICHTE

---

27.04.

-Erster Versuch Grafik darzustellen, Spieler-Objekt gibt gröÙe, position und Farbe an GUI Objekt. Das Spiel wird aktiv gerendert, der Code dafür kommt von der Seite [https://www.gamedev.net/tutorials/\\_/technical/general-programming/java-games-active-rendering-r2418/](https://www.gamedev.net/tutorials/_/technical/general-programming/java-games-active-rendering-r2418/)

<https://stackoverflow.com/questions/2442599/how-to-set-jframe-to-appear-centered-regardless-of-monitor-resolution> hat bei der zentrierung des Fensters geholfen

-Tasteneingabe in Bewegung umgesetzt, Spieler kann mit wasd vertikal und horizontal sowie diagonal bewegt werden. Syntax für Keylistener ebenfalls von [https://www.gamedev.net/tutorials/\\_/technical/general-programming/java-games-active-rendering-r2418/](https://www.gamedev.net/tutorials/_/technical/general-programming/java-games-active-rendering-r2418/)

---

28.04.

Input von Keylistener zu KeyBinds geändert

<https://stackoverflow.com/questions/22741215/how-to-use-key-bindings-instead-of-key-listeners>

Diagonale Bewegungsgeschwindigkeit auf geradlinige Bewegungsgeschwindigkeit reduziert, Position als Point2D gespeichert

---

29.4. UND 30.4

-Kollisionssystem erstellt. Kollision von Rechtecken kann ermittelt werden. Wand Klasse erstellt und Kollision mit Wand ermöglicht. Der Bildschirmrand kann ebenfalls nicht mehr verlassen werden.

-Vollbildmodus <https://stackoverflow.com/questions/1155838/how-can-i-do-full-screen-in-java-on-osx> kann mit Escape beendet werden

---

03.05.

-Mithilfe von Zeichnung die benötigte If-Abfrage überlegt um die Kollision von Kreisen mit Rechtecken zu überprüfen.

-kreis Rechteck Kollision in Code gebracht und Projektil Klasse erstellt

---

04.05.

-Kollision von Spieler und Projektil implementiert sowie Funktion um Projektile malen zu können gemacht. MaxLeben und Leben als Eigenschaften des Spielers hinzugefügt

-Methode um das Fenster zu schließen ohne Fehlermeldung von <https://stackoverflow.com/questions/1234912/how-to-programmatically-close-a-jframe>

---

05.05.

-Um Projektil zu Punkt zu bewegen erst fehlerhafte Formel benutzt, dann statt Steigung eine Berechnung über winkel von <https://stackoverflow.com/questions/39818833/moving-an-object-from-one-point-to-another> genommen.

-Bei Mausklick wird ein Projektil in Richtung Mausklick geschossen, wenn Projektile den Bildschirm verlassen werden sie gelöscht. Kontakt mit Wand-Objekten löscht das Projektil ebenfalls. Feuern nur auf linke Maustaste beschränkt.

---

10.05.

- Update und Start Klasse erstellt um gleichmäßige Updates und Multi-Threading zu ermöglichen. Gleichmäßige Updates von <https://stackoverflow.com/questions/63515194/how-to-run-a-code-60-times-per-second-in-java> Thread starten mit Hilfe von <https://dbs.cs.uni-duesseldorf.de/lehre/docs/java/javabuch/html/k100142.html#:~:text=Die%20Klasse%20Thread%20ist%20Best%20andteil,Beenden%20von%20Threads%20zur%20Verf%C3%BCgung.>

---

17.05.

- Lösung für unterschiedliche Bildschirmauflösungen gefunden, Windows Einstellungen ignoriert mithilfe von <https://stackoverflow.com/questions/47613006/how-to-disable-scaling-the-ui-on-windows-for-java-9-applications>

---

19.05.

-Waffenklasse erstellt, Sub Klassen für verschiedene Waffentypen erstellt. Aktuell ausgerüstete Waffe wird auf dem Bildschirm angezeigt

---

24.05. UND 25.05.

-Waffenpickup erstellt um Waffen im Spiel aufzusammeln. Die Shotgun schießt jetzt mehrere Kugeln und einer Kegelform, Projektile haben eine maximale Distanz die sie sich bewegen. Die Waffenpickups verteilen bei Berührung vorher zufällig erstellte Waffen und werden anschließend gelöscht.

-Magazingröße und übrige Kugeln werden gezählt und angezeigt. Wenn das Magazin leer ist wird ein Timer gestartet nach dem das Magazin wieder voll ist und man wieder schießen kann. Syntaxhilfe von <https://docs.oracle.com/javase/7/docs/api/java/util/Timer.html> und <https://docs.oracle.com/javase/7/docs/api/java/util/TimerTask.html>

-Nachladen wird durch Text angezeigt

---

Ab 25.5. vergebens an Online-Multiplayer gearbeitet. Aufgrund der knappen Zeit musste das Spiel kurzfristig auf Singleplayer umgestellt werden.

---

13.06.

-Hauptmenü erstellt. Hilfe von:

<https://docs.oracle.com/javase/tutorial/uiswing/layout/none.html>

<https://alvinalexander.com/java/java-set-jframe-size/>

<https://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>

<https://stackoverflow.com/questions/22452930/terminating-a-java-program>

-Pause hinzugefügt

-Levelaufbau geändert

-Highscore in txt Datei gespeichert. altes Projekt zum BufferedReader und BufferedWriter als Hilfestellung.

---

14.06.

Den Cursor im Spiel durch eine eigene Textur ersetzt

<https://stackoverflow.com/questions/40592495/how-to-create-custom-cursor-images-in-java>

<https://docs.oracle.com/javase/tutorial/2d/images/loadimage.html>

---

20.06.

Texturen für die Waffen hinzugefügt. Die Waffen werden zum Mauscursor ausgerichtet und gleichhoch gescaled.

<http://mrbool.com/how-to-display-flip-and-provide-a-reflection-to-the-image-with-java/25983>

