Einzelprojekt 24/25 - Starfighter Alliance

Software-Projekt

Universität Ulm

Bachelor of Computer Science

Alhasan Ramadan
Matrikelnummer: 1170485

Pflichtenheft

Unter Aufsicht von

Aaron Molt

Institut für Softwaretechnik und Programmiersprachen, Ulm

Prof. Dr. Matthias Tichy, Sabrina Böhm, Valentin Kolb

Wintersemester 2024/25

Contents

1 Spielidee und Hintergrund

Dieses Pflichtenheft fasst die Anforderungen an das Endlos-Flug-Spiel Starfighter Alliance zusammen, welches in diesem Wintersemester (WiSe 24/25) als Einzelprojekt entwickelt werden soll.
"Die Rebellen-Allianz muss durch einen Meteoritengürtel fliegen, um vor der großen Macht des Imperiums zu fliehen. Dabei stehen sich nun ein Starfighter und eine Horde an Meteoriten gegenüber. Könnt ihr mehr Meteoriten ausweichen als ein Yedi oder endet ihr frühzeitig wie ein Stormtrooper?"[1]
In diesem Dokument werden die genaue Funktionsweise sowie die Implementierungsrichtlinien beschrieben.

2 Einleitung

Der Zweck dieses Dokuments ist es, eine detaillierte Beschreibung der Anforderungen, sowie der Benutzerschnittstelle für die Anwendung Starfighter Alliance bereitzustellen. Es wird abgegrenzt, welche Anforderungen erfüllt werden müssen, damit die entwickelte Anwendung vom Kunden akzeptiert wird. Im Folgenden wird auf die Systemfunktionalität und die Systemeinschränkungen, sowie Annahmen über das Produkt eingegangen. Zuletzt werden detaillierte Anforderungsspezifikation bereitgestellt.

3 Allgemeine Beschreibung

Starfighter Alliance ist ein Single-Player-Game aus dem Genre der Side-Scrolling Endless Runners.

"Ein Endless Runner ist ein Actionspiel, linear im Design ohne Ende, ohne Pausen oder Pausen, ohne Stufen oder wechselnde Levels. Es hat ein durchgehendes Level. Das Spiel beginnt langsam und leicht. Schauplätze sind unter anderem Wege, Straßen und Eisenbahnschienen. Mit zunehmender Geschwindigkeit des Spiels scheint der Spielercharakter immer schneller und schneller zu laufen, als wäre er auf einem riesigen Laufband. Hindernisse nähern sich immer schneller und schneller, was immer schnellere Reaktionen erfordert (...). Der Spieler erliegt der Intensität und das Spiel ist vorbei. Ironischerweise werden sie „endlos" genannt, denn für Spieler sind sie alles andere als das. Für die meisten kommt das Ende viel zu früh.[2]

Es nimmt genau ein Spieler / eine Spielerin an einer Spielpartie teil. Ziel des Spiels ist es, mit dem Raumschiff im Weltall möglichst viele Wellen an Meteroiten zu überstehen.

[1] Aus der Einleitung im Lastenheft
[2] Übersetzt aus jerrymomoda.com

4 Spezifische Anforderungen

Dieser Abschnitt enthält eine vollständige und detaillierte Auflistung sämtlicher spezifischer Anforderungen, die an das System gestellt werden. In diesem Zusammenhang wird eine gründliche Beschreibung des Systems sowie seiner verschiedenen Funktionen und Merkmale präsentiert. Alle wesentlichen Anforderungen werden klar und präzise formuliert, um sicherzustellen, dass das System den festgelegten Standards entspricht und alle gewünschten Leistungs- und Funktionsanforderungen erfüllt. Zudem werden hier die Interaktionen zwischen den verschiedenen Systemkomponenten und deren jeweilige Aufgaben beschrieben, um die Systemarchitektur vollständig zu verstehen.

4.1 Funktionale Anforderungen

Im Folgenden werden alle Anforderungen, die die grundlegenden Aktionen des Softwaresystems spezifiziert.

4.1.1 Rahmenbedingungen

ID	FA 1
TITEL:	Single-Player-Spiel
BESCHREIBUNG:	Dem Spieler steht nur eine Einzelspieler-Option zur Verfügung
BEGRÜNDUNG:	Es legt fest, dass keine Mehrspieler-Option vorhanden sein wird, und schränkt somit die Funktionen der Interaktion ein.
ABHÄNGIGKEITEN:	NFA1, NFA3
RISIKEN:	-

ID	FA 2
TITEL:	Konfiguration
BESCHREIBUNG:	In der Start-Ansicht soll eine JSON-Datei als Konfigurationsdatei ausgelesen werden. In dieser Datei werden Eigenschaften wie Anzahl der Leben und Raumschiffwerte festgelegt.
BEGRÜNDUNG:	Die Konfigurationsdatei ermöglicht es, zentrale Spieleeinstellungen wie die Anzahl der Leben und Raumschiffwerte flexibel anzupassen, ohne den Code ändern zu müssen. Dies erleichtert zukünftige Anpassungen und eine benutzerfreundliche Steuerung der Spielparameter.
ABHÄNGIGKEITEN:	JSON-Konfigurationsdatei
RISIKEN:	Fehlerhafte oder unvollständige Datei kann zu falschen Spieleinstellungen führen.

ID	FA 3
TITEL:	Farben-Server
BESCHREIBUNG:	Mit einem Client-Server-Modell wird vor Spielstart ein Server benutzt, um die Farbe des Raumschiffes festzulegen. Auf Anfrage an den Server wählt der Server zufällig eine von neun Farben aus und schickt diese dem Client zurück.
BEGRÜNDUNG:	Die Verwendung eines Client-Server-Modells zur Festlegung der Raumschiff-Farbe ermöglicht eine dynamische und zufällige Farbauswahl, die für jedes Spiel einzigartig ist. Dies trägt zur Individualisierung und Vielfalt des Spielerlebnisses bei.
ABHÄNGIGKEITEN:	NFA2
RISIKEN:	Verbindungsprobleme könnten dazu führen, dass keine Farbe zugewiesen wird.

ID	FA 4
TITEL:	Server-Antwort
BESCHREIBUNG:	Im Falle einer erfolgreichen Anfrage des Clients ist der Antwort Code des Servers 200 und das data-Feld in der JSON-Datei enthält einen String mit der gewünschten Farbe. Falls die Anfrage fehlgeschlagen ist, ist der Code 400 für illegale Anfrage und 500 im Falle eines internen Fehlers im Server.
BEGRÜNDUNG:	Diese Antwortstruktur stellt sicher, dass der Client korrekt über den Status der Anfrage informiert wird.
ABHÄNGIGKEITEN:	NFA2
RISIKEN:	Fehlerhafte Statuscodes oder fehlende Antwort könnten den Spielfluss stören.

ID	FA 5
TITEL:	Soundeffekte
BESCHREIBUNG:	Das Spiel wird für das Zerstören von Hindernissen und Schüsse bestimmte Soundeffekte abspielen.
BEGRÜNDUNG:	Die Soundeffekte sorgen für akustische Rückmeldungen bei wichtigen Ereignissen, wie dem Zerstören von Hindernissen und Schüssen. Sie verstärken das Spielerlebnis.
ABHÄNGIGKEITEN:	NFA3
RISIKEN:	-

4.1.2 Weltraum

ID	FA 6
TITEL:	Weltraum als Spielumgebung
BESCHREIBUNG:	Der Weltraum stellt den Hintergrund des Spiels dar. Das Raumschiff kann diesen nicht verlassen.
BEGRÜNDUNG:	Diese Anforderung beschreibt eine Einschränkung und Regelung der Spielumgebung, die das Verhalten des Raumschiffs definiert.
ABHÄNGIGKEITEN:	-
RISIKEN:	-

4.1.3 Raumschiff

ID	FA 7
TITEL:	Steuerung des Raumschiffes
BESCHREIBUNG:	Das Raumschiff kann sich in alle Richtungen (links, rechts, hoch und runter) bewegen, jedoch nicht drehen. Die Steuerung erfolgt mittels der Pfeiltasten (oder W,A,S,D).
BEGRÜNDUNG:	Die Bewegung und Steuerung des Raumschiffes gilt als essentielle Spielmechanik und bestimmt direkt den Spielablauf.
ABHÄNGIGKEITEN:	-
RISIKEN:	Koordinierung in der Spielwelt

ID	FA 8
TITEL:	Tempo des Raumschiffes
BESCHREIBUNG:	Das Raumschiff bewegt sich immer mit konstantem Tempo und stoppt, sobald keine Richtungstaste mehr betätigt wird. Die Bewegungsgeschwindigkeit des jeweiligen Raumschiff wird hierbei mit einbezogen werden.
BEGRÜNDUNG:	Diese Anforderung präzisiert das Verhalten der Steuerung. Somit gilt sie ebenfalls als grundlegende Mechanik zur Steuerung der Spieldynamik.
ABHÄNGIGKEITEN:	FA15
RISIKEN:	Feinabstimmung für Raumschiffe mit verschiedenen Werten

ID	FA 9
TITEL:	Leben des Raumschiffes
BESCHREIBUNG:	Das Raumschiff hat eine gewisse Anzahl an Leben. Die Leben werden bei Aufprall mit einem Hindernis einzeln nacheinander abgezogen.
BEGRÜNDUNG:	Die Reduzierung der Leben ist eine direkte Spielverhaltensfunktion, die eine Interaktion mit einem Hindernis beschreibt und das Überleben und die Herausforderung im Spiel bestimmt.
ABHÄNGIGKEITEN:	FA28
RISIKEN:	Komplexität bei der Implementierung der Kollisionslogik, Performance-Probleme bei der Berechnung von Kollisionen

ID	FA 10
TITEL:	Größe des Raumschiffes
BESCHREIBUNG:	Das Raumschiff ist ein quadratisches Objekt, das im Weltraum bewegt werden muss. Die Größe des Raumschiffs muss dem Weltraum angepasst sein und sollte ca. 1/10 der Breite des Weltraumes betragen.
BEGRÜNDUNG:	Diese Anforderung beschreibt die Eigenschaften des Raumschiffs, einschließlich der Größe und Form. Sie hat direkten Einfluss auf die Darstellung des Raumschiffs im Spiel und die Hitbox hat essentiellen Wert für die Grundspielregeln.
ABHÄNGIGKEITEN:	FA31, FA33
RISIKEN:	-

ID	FA 11
TITEL:	Auswahl des Raumschiffes
BESCHREIBUNG:	Dem Spieler steht vor Spielbeginn eine Auswahl aus einer Menge an verschiedener Raumschiffe zur Verfügung. Es muss mindestens eines der folgenden Raumschiffe eingebaut werden: Millenium Falke, Y-Sternenjäger, X-Sternenjäger.
BEGRÜNDUNG:	Diese Aussage beschreibt eine konkrete Funktion des Spiels, nämlich die Auswahl eines Raumschiffs. Es handelt sich um eine Interaktionsmöglichkeit des Spielers.
ABHÄNGIGKEITEN:	FA15, FA36
RISIKEN:	-

ID	FA 12
TITEL:	Werte des Raumschiffes
BESCHREIBUNG:	Die auswählbare Raumschiffe haben jeweils eine unterschiedliche Bewegungsgeschwindigkeit und eine unterschiedliche Schusskraft. Diese Werte sollen beim Auswählen der Raumschiffe dem Spieler / der Spielerin angezeigt werden.
BEGRÜNDUNG:	Die Anzeige dieser Werte beim Auswählen des Raumschiffs stellt sicher, dass der Spieler eine informierte Wahl treffen kann. Das betrifft die Spielmechanik und die Interaktion mit dem Spieler.
ABHÄNGIGKEITEN:	FA23
RISIKEN:	-

4.1.4 Schüsse

ID	FA 13
TITEL:	Abfeuern eines Schusses
BESCHREIBUNG:	Der Spieler kann mit durch Drücken der Leertaste einen Schuss abfeuern.
BEGRÜNDUNG:	Dies ist eine direkte Steuerungsfunktion des Spiels.
ABHÄNGIGKEITEN:	FA27
RISIKEN:	-

ID	FA 14
TITEL:	Zerstörungskraft der Schüsse
BESCHREIBUNG:	Die abgefeuerten Schüsse des Spielers können Hindernisse durch Treffer zerstören.
BEGRÜNDUNG:	Hier wird die spezifische Funktion der Schüsse beschrieben.
ABHÄNGIGKEITEN:	Der Trefferschaden hängt von der Schusskraft des jeweiligen Raumschiffes ab(FA15).
RISIKEN:	-

ID	FA 15
TITEL:	Aufprallverhalten der Schüsse
BESCHREIBUNG:	Sobald ein Schuss ein Hindernis trifft, verliert das Hindernis eine bestimmte Anzahl an Leben, abhängig von der Schusskraft des ausgewählten Raumschiffes. Nach einem Aufprall löst sich der Schuss auf.
BEGRÜNDUNG:	Die Anforderung beschreibt die Interaktionsweise der Schüsse mit Hindernissen.
ABHÄNGIGKEITEN:	FA15, FA23, FA29
RISIKEN:	-

ID	FA 16
TITEL:	Bewegung der Schüsse
BESCHREIBUNG:	Ein abgefeuerter Schuss fliegt vom oberen Rand des Raumschiffs mittig aus der Abschussstelle senkrecht nach oben bis er ein Hindernis trifft oder aus dem Bild verschwindet.
BEGRÜNDUNG:	Diese Anforderung beschreibt die Bewegungsfunktion der Schüsse.
ABHÄNGIGKEITEN:	-
RISIKEN:	-

ID	FA 17
TITEL:	Schussrate
BESCHREIBUNG:	Der Spieler muss, sobald er einen Schuss abgegeben hat, einen Timer abwarten, um den nächsten Schuss abfeuern zu können.
BEGRÜNDUNG:	Die Schussrate ist eine einschränkende Eigenschaft, die sich direkt auf die Spieldynamik mit Hindernissen auswirkt.
ABHÄNGIGKEITEN:	FA23
RISIKEN:	-

4.1.5 Hindernisse

ID	FA 18
TITEL:	Bewegung der Hindernisse
BESCHREIBUNG:	Die Hindernisse haben eine bestimmte Geschwindigkeit, die über Zeit leicht ansteigt.
BEGRÜNDUNG:	Diese Anforderung präzisiert das Verhalten der Hindernisse. Somit gilt sie ebenfalls als grundlegende Mechanik zur Steuerung der Spieldynamik.
ABHÄNGIGKEITEN:	FA31
RISIKEN:	Fehlerhafte Geschwindigkeitsskalierung könnte zu unspielbaren Hindernisbewegungen führen.

ID	FA 19
TITEL:	Leben der Hindernisse
BESCHREIBUNG:	Ein Hindernis hat eine Anzahl an Leben. Durch Schüsse können Hindernisse Leben verlieren und zerstört werden, sobald sie keine Leben mehr besitzen. Die Anzahl der Leben hängt dabei von der Größe des Hindernisses ab (größere Hindernisse haben mehr Leben).
BEGRÜNDUNG:	Die Funktionsweise der Leben der Hindernisse ist eine direkte Spielverhaltensfunktion, die eine Interaktion mit dem Spieler beschreibt und das Überleben und die Herausforderung im Spiel bestimmt. Sie gilt als eine weitere Kernmechanik.
ABHÄNGIGKEITEN:	FA25, FA30
RISIKEN:	-

ID	FA 20
TITEL:	Größe der Hindernisse
BESCHREIBUNG:	Jedes Hindernis hat eine bestimmte Größe. Die Höhe und Breite der Hindernisse sind variable, jedoch ist die Höhe der Hindernisse pro Welle nicht unterschiedlich.
BEGRÜNDUNG:	Die konstante Höhe der Hindernisse pro Welle sorgt für Vorhersehbarkeit und Balancing, damit die Spieler sich besser auf die Herausforderungen einstellen können.
ABHÄNGIGKEITEN:	FA31
RISIKEN:	-

4.1.6 Wellen

ID	FA 21
TITEL:	Wellenverhalten
BESCHREIBUNG:	Eine Welle bezeichnet das Erscheinen von Hindernissen auf einer Höhe, die sich gemeinsam mit der gleichen Geschwindigkeit senkrecht von oben nach unten bewegen.
BEGRÜNDUNG:	Dies beschreibt die Art und Weise, wie eine Welle generiert wird und beschreibt das Wellenverhalten.
ABHÄNGIGKEITEN:	Synchronisiert die Hindernisbewegungen und hat direkten Einfluss auf die Spielumgebung und die Platzierungslogik der Wellen.
RISIKEN:	Synchronisierungsfehler könnten Wellen unterschiedlich schnell erscheinen lassen.

ID	FA 22
TITEL:	Generation der Hindernisse in den Wellen
BESCHREIBUNG:	Pro Welle wird eine beliebige Anzahl an Hindernissen generiert werden, solange mindestens die 1,5-fache breite des Raumschiffs als Weg für das Raumschiff irgendwo in der Welle frei bleibt.
BEGRÜNDUNG:	Diese Anforderung gilt als wichtige Einschränkung der Wellen und damit als Balancing-Methode.
ABHÄNGIGKEITEN:	FA10, FA33
RISIKEN:	-

ID	FA 23
TITEL:	Wellenabstand
BESCHREIBUNG:	Zwischen zwei Hinderniswellen muss mindestens 3 mal die Höhe des Raumschiffs als Platz frei gelassen werden. Hindernisse, die am unteren Rand den Weltraum verlassen, werden automatisch zerstört und verschwinden.
BEGRÜNDUNG:	Diese Anforderung beeinflusst die Schwierigkeit des Spieles und verändert die Spieldynamik.
ABHÄNGIGKEITEN:	Platzierung von Hindernissen
RISIKEN:	Fehlerhafte Abstandseinstellung könnte zu Überlappungen oder unerwünschten Pausen zwischen Wellen führen.

4.1.7 Spielablauf

ID	FA 24
TITEL:	Start-Ansicht
BESCHREIBUNG:	Der Spieler soll nach dem Starten des Spiels einen Start-Screen als Erste Ansicht sehen. Hier soll der Spieler durch Buttons zwischen dem Starten des Spiels, Laden einer Konfigurationsdatei und Beenden des Spiels navigieren können.
BEGRÜNDUNG:	Diese Anforderung beschreibt die Erste Ansicht und dessen Navigationsfunktion.
ABHÄNGIGKEITEN:	(FA28?)
RISIKEN:	-

ID	FA 25
TITEL:	Auswahl-Ansicht
BESCHREIBUNG:	Es soll eine Raumschiff-Auswahl-Ansicht existieren, die es dem Spieler ermöglicht, ein Raumschiff für das Spiel auszuwählen. Eine Liste von verfügbaren Raumschiffen wird angezeigt, jedes mit einem Vorschaubild, der Geschwindigkeitseigenschaft und Schusseigenschaft. Durch Anklicken des gewünschten Raumschiffs wird in die Spiel-Ansicht gewechselt.
BEGRÜNDUNG:	Diese Anforderung beschreibt die zweite Ansicht und dessen Auswahlsfunktion und die Funktion, Raumschiffe und dessen Werte abzubilden.
ABHÄNGIGKEITEN:	FA24
RISIKEN:	-

ID	FA 26
TITEL:	Spiel-Ansicht
BESCHREIBUNG:	Es soll eine Spiel-Ansicht als Hauptspielbereich existieren. Dieser Screen gilt als Hauptspielbereich, wird nach der Auswahl-Ansicht angezeigt und enthält den Weltraum mit dem ausgewählten Raumschiff und Hindernis-Wellen. Während des Spiels werden die ganze Zeit Informationen über die Anzahl bestandener Wellen, gesammelte Punkte und die derzeitigen Leben des Raumschiffes angezeigt werden. Das Raumschiff wird am unteren Bildschirmrand mittig positioniert. Bei Drücken eines Startknopfes beginnt ein sichtbarer Countdown, der von 3 auf 0 herunterzählt. Nach Ablauf des Countdowns ertönt ein Startsignal, das entweder als Ton oder durch den Text „Start“ angezeigt wird. Nach dem Startsignal kann das Raumschiff gesteuert werden und die Hinderniswellen beginnen. Ein Zähler, der die Anzahl der Wellen anzeigt, sowie eine Punkteanzeige werden am Bildschirmrand sichtbar sein. Wenn eine Welle von Hindernissen den unteren Bildschirmrand erreicht, erhöhen sich die Punkte. Wenn eine Welle komplett zerstört wurde gibt es mehr Punkte.
BEGRÜNDUNG:	Diese Anforderung beschreibt den Hauptspielbereich, in dem alle wichtigen Elemente des Spiels – das Raumschiff, Hindernisse und Statusinformationen – dargestellt werden. Die Anzeige von Statusinformationen wie Punkte, Leben und Wellenzahl gehört zu den Kernmechaniken des Spiels, da sie dem Spieler Rückmeldung über seine Fortschritte und seinen Zustand im Spiel gibt.
ABHÄNGIGKEITEN:	FA25
RISIKEN:	-

ID	FA 27
TITEL:	Spiel-Pausierung
BESCHREIBUNG:	Es soll dem Spieler möglich sein, über einen Button das Spiel zu pausieren. Dies wird über einen weiteren Screen/Popup erreicht. Zusätzlich wird hier eine Fortsetzungsmöglichkeit über einen Button eingesetzt.
BEGRÜNDUNG:	Diese Anforderung ermöglicht es dem Spieler, das Spiel temporär zu unterbrechen und später fortzusetzen. Die Funktion der Pausierung stellt eine Interaktionsmöglichkeit dar und verbessert die Benutzerfreundlichkeit, da der Spieler das Spielgeschehen jederzeit anhalten kann, um z. B. eine Pause einzulegen.
ABHÄNGIGKEITEN:	FA26
RISIKEN:	-

ID	FA 28
TITEL:	End-Ansicht
BESCHREIBUNG:	Wenn das Spiel zu Ende ist, wechselt die Spiele-Ansicht zur End-Ansicht und der Spielende-Screen zeigt das Raumschiff, die Anzahl der Wellen und die Punkteanzahl an. Zusätzlich zeigt er die besten 10 Ergebnisse des eigenen Highscores an. Das Ergebnis des jeweiligen Spiellaufs wird hier hervorgehoben. Es existiert ein Button, mit dem der Spieler nach Ende zurück zur Anfangs-Ansicht geleitet wird.
BEGRÜNDUNG:	Diese Anforderung beschreibt die Ansicht nach dem Ende des Spiels, in der die Spielergebnisse und der Highscore angezeigt werden. Die Anzeige der besten 10 Ergebnisse sowie der aktuellen Leistung des Spielers ist eine wichtige Funktion zur Motivation und bietet dem Spieler eine Rückmeldung über seine Erfolge. Diese Darstellung gehört zur Funktionalität des Spiels, da sie den Abschluss des Spiels und die Bewertung des Spielergebnisses umfasst.
ABHÄNGIGKEITEN:	FA26, FA27
RISIKEN:	-

ID	FA 29
TITEL:	Highscore
BESCHREIBUNG:	Die 10 besten Ergebnisse des Highscores eines Spielers werden in einer extra Datei gespeichert. Dazu sollen mindestens die Anzahl an Wellen und die Punkte des aktuellen Spieldurchlaufs angezeigt werden.
BEGRÜNDUNG:	Dies beschreibt die Funktion einer Abspeicherung des Highscores. Die Speicherung der 10 besten Spielergebnisse ermöglicht eine kontinuierliche Verfolgung der besten Runs und fördert den Wettbewerb, indem die Daten dauerhaft erhalten bleiben.
ABHÄNGIGKEITEN:	-
RISIKEN:	Datenverlust bei fehlerhafter Speicherung.

4.1.8 Optional

ID	FA 30
TITEL:	Automatische Zielerfassung für Schüsse
BESCHREIBUNG:	Das Spiel implementiert ein System zur automatischen Zielerfassung für Schüsse, sodass der Spieler auch bei motorischen Einschränkungen spielen kann.
BEGRÜNDUNG:	Diese Funktion ist wichtig für die Zugänglichkeit des Spiels und trägt dazu bei, dass das Spiel für ein breiteres Publikum spielbar bleibt.
ABHÄNGIGKEITEN:	-
RISIKEN:	Erhöhter Entwicklungsaufwand, um das Zielerfassungs- und Schusssystem genau zu justieren.

ID	FA 31
TITEL:	Farbeinstellungen für rot-grün Schwäche
BESCHREIBUNG:	Das Spiel bietet eine Option, bei der die Farben für Spieler mit rot-grün Schwäche angepasst werden können.
BEGRÜNDUNG:	Diese Funktion ist wichtig, um das Spiel für alle zugänglich zu machen, insbesondere für Menschen mit Farbsehschwächen.
ABHÄNGIGKEITEN:	-
RISIKEN:	Falsche Kontrastwahl

ID	FA 32
TITEL:	Sprachauswahl
BESCHREIBUNG:	Das Spiel bietet eine mindestens englische Sprachoption für alle Texte und Sprachausgaben.
BEGRÜNDUNG:	Die Möglichkeit, die Sprache zu wählen, stellt sicher, dass das Spiel für internationale Spieler zugänglich ist. Sie erleichtert es, das Spiel zu verstehen und zu genießen, und ist entscheidend für eine größere Reichweite des Spiels.
ABHÄNGIGKEITEN:	-
RISIKEN:	-

ID	FA 33
TITEL:	Schriftgrößenanpassung
BESCHREIBUNG:	Das Spiel ermöglicht es, die Schriftgröße der Textanzeigen anzupassen.
BEGRÜNDUNG:	Die Anpassung der Schriftgröße sorgt dafür, dass Spieler mit eingeschränkter Sehkraft die Textinformationen problemlos lesen können.
ABHÄNGIGKEITEN:	-
RISIKEN:	Layout-Anpassung für verschiedene Bildschirmgrößen.

ID	FA 34
TITEL:	Code-Fenster
BESCHREIBUNG:	In der Start-Ansicht gibt es ein Textfeld, wo man bestimmte Codes eingeben kann, um extra Inhalte freizuschalten.
BEGRÜNDUNG:	Diese Elemente tragen zur Dynamik des Spiels bei und fördern das Gefühl der Neugier.
ABHÄNGIGKEITEN:	-
RISIKEN:	Balancing

ID	FA 35
TITEL:	Dimension-Wechsel-Button
BESCHREIBUNG:	In der Start-Ansicht sollte es einen Button geben, der bei Druck die Dimension des Spiels verändert.
BEGRÜNDUNG:	Funktionsverhalten eines Knopfes.
ABHÄNGIGKEITEN:	NFA8
RISIKEN:	-

ID	FA 36
TITEL:	Untertitel-Textfeld
BESCHREIBUNG:	Für die Untertitel wird ein Textfeld mit dem jeweilig gesprochenem Satz angezeigt.
BEGRÜNDUNG:	Funktionsweise eines Textes.
ABHÄNGIGKEITEN:	NFA9
RISIKEN:	-

4.2 Nicht-funktionale Anforderungen

Dieser Abschnitt spezifiziert die nicht-funktionalen Anforderungen an das Softwaresystem. Nicht-funktionale Anforderungen beziehen sich auf die Umstände, unter denen die geforderte Funktionalität erbracht werden muss. Sie beschreiben die Art und Weise, wie etwas zu tun ist.[3]

4.2.1 Rahmenbedingungen

ID	NFA 1
TITEL:	Robustheit
BESCHREIBUNG:	Die Anwendung darf nicht absturzen. Bei 100 Spielen darf maximal 1 Spiel aufgrund eines ¨ Fehlers abgebrochen werden.
BEGRÜNDUNG:	Dies soll dem Spieler / der Spielerin ein gutes Spielerlebnis offenbaren.
ABHÄNGIGKEITEN:	-A
RISIKEN:	-

ID	NFA 2
TITEL:	Lokales Spiel
BESCHREIBUNG:	Die Anforderung setzt fest, dass das Spiel auf einem einzelnen Gerät laufen wird
BEGRÜNDUNG:	Diese Anforderung beeinflusst die Betriebsumgebung und Systemanforderungen, wirft also den technischen Aspekt des Projektes auf
ABHÄNGIGKEITEN:	-
RISIKEN:	-

ID	NFA 3
TITEL:	Server-Anbindung
BESCHREIBUNG:	Der Server ist unter der URL softwaregrund.pro/jekt/ws/color erreichbar und unterstützt Verbindungen über ws:// auf Port 80 und wss:// auf Port 443. Er arbeitet nach dem Websocket-Protokoll. Eine WebUI zum Testen des Servers ist unter https://softwaregrund.pro/jekt/ verfügbar.
BEGRÜNDUNG:	Die Angabe der Serververbindung stellt sicher, dass der Client korrekt mit dem Server kommunizieren kann und ermöglicht einen einfachen Zugang zum Testen.
ABHÄNGIGKEITEN:	-
RISIKEN:	-

[3] Definition aus Department of Informatics Uzh (Seite 3 f.)

4.2.2 Sprachen und Technologien

ID	NFA 4
TITEL:	Anwendung- und Implementierungssprache
BESCHREIBUNG:	Die Anwendungssprache kann Deutsch oder Englisch sein. Die Implementierungssprache und Dokumentationssprache müssen auf Englisch sein.
BEGRÜNDUNG:	Dies bezieht sich auf die Sprachenstandards und Dokumentation, nicht auf die Funktionsweise des Spiels.
ABHÄNGIGKEITEN:	-
RISIKEN:	-

ID	NFA 5
TITEL:	Entwicklungsumgebung
BESCHREIBUNG:	Das Spiel wird in Godot entwickelt, und für die Grafiken werden Blender und Figma verwendet.
BEGRÜNDUNG:	-
ABHÄNGIGKEITEN:	Mögliche Kompatibilitätsprobleme zwischen den Tools.
RISIKEN:	

ID	NFA 6
TITEL:	Plattformkompatibilität
BESCHREIBUNG:	Die Anwendung muss mindestens auf entweder Windows 10 bzw. 11, auf Linux oder MacOS laufen.
BEGRÜNDUNG:	Beschreibt unter einem technischen Aspekt die unterstützten Plattformen, ohne Auswirkungen auf das Spielverhalten zu haben.
ABHÄNGIGKEITEN:	Kompatibilität mit NFA3, mögliche plattformspezifische Bibliotheken oder Systemanforderungen.
RISIKEN:	Unterschiede bei der plattformübergreifenden Anpassung. Aufwand für neue Tests und Optimierungen.

ID	NFA 7
TITEL:	Benutzerfreundlichkeit
BESCHREIBUNG:	Das Spiel muss eine intuitive Benutzeroberfläche bieten, die es Spielern ermöglicht, alle Funktionen ohne eine ausführliche Anleitung zu verstehen.
BEGRÜNDUNG:	Eine intuitive Benutzeroberfläche erleichtert den Spielern das Verständnis der Funktionen, sodass sie schnell und ohne Anleitung spielen können. Das verbessert die Benutzererfahrung.
ABHÄNGIGKEITEN:	-
RISIKEN:	-

4.2.3 Hindernis-Typ

ID	NFA 8
TITEL:	Typen von Hindernisse
BESCHREIBUNG:	Die Hindernisse sollen als Meteoriten oder auch als andere Weltraumelemente dargestellt werden.
BEGRÜNDUNG:	Hier wird eine Funktion der Ästhetik beschrieben, die Eigenschaften der Hindernisse ändern sich nicht. Somit hat dies keinen Einfluss auf das atomare Spielerlebnis.
ABHÄNGIGKEITEN:	-
RISIKEN:	-

4.2.4 Optional

ID	NFA 9
TITEL:	3D-Modus
BESCHREIBUNG:	Das Spiel kann alternativ in einem 3D-Modus oder in verschiedenen Perspektiven wie First-Person oder Third-Person gespielt werden.
BEGRÜNDUNG:	Der 3D-Modus und die Perspektiven sind eine Erweiterung des Spielerlebnisses, die nicht die grundlegende Funktionalität beeinflussen.
ABHÄNGIGKEITEN:	Zusätzliche Ressourcen für die 3D-Entwicklung und -Optimierung.
RISIKEN:	Höherer Entwicklungsaufwand und mögliche Performance-Probleme.

ID	NFA 10
TITEL:	Untertitel
BESCHREIBUNG:	Das Spiel enthält Untertitel für alle gesprochene Texte. Verbessert die Zugänglichkeit für hörgeschädigte Spieler.
BEGRÜNDUNG:	Untertitel sind eine Zugänglichkeitsfunktion, die das Spielerlebnis verbessert, ohne die Hauptfunktionalität des Spiels zu ändern.
ABHÄNGIGKEITEN:	-
RISIKEN:	Entwicklungsaufwand für die Untertitel.

ID	NFA 11
TITEL:	Übergänge zwischen Screens im ”PowerPoint-Style”
BESCHREIBUNG:	Die Übergänge zwischen den verschiedenen Bildschirmen des Spiels sollen animiert und dynamisch wie in PowerPoint dargestellt werden.
BEGRÜNDUNG:	Diese Art von Animation betrifft lediglich die Benutzeroberfläche und ist daher eine nicht funktionale Anforderung, die das visuelle Erlebnis verbessert, ohne die Spielmechanik zu beeinflussen.
ABHÄNGIGKEITEN:	-
RISIKEN:	Zu viele Animationen können die Spielperformance beeinträchtigen.

ID	NFA 12
TITEL:	Charakterstimmen bei Schiffswahl
BESCHREIBUNG:	Bestimmte Raumschiffe sollen beim Auswählen von Charakteren Sprachzeilen haben.
BEGRÜNDUNG:	Diese Funktion betrifft die Benutzeroberfläche und das Audio, was das Spielerlebnis erweitert, jedoch keine grundlegende Spielmechanik beeinflusst.
ABHÄNGIGKEITEN:	FA5
RISIKEN:	Höherer Aufwand für die Erstellung und Integration der Sprachaufnahmen.

ID	NFA 13
TITEL:	Umgebungs-Szenerie auswählbar
BESCHREIBUNG:	Es können Umgebungen wie die Verfolgung von Jango Fett durch Obi-Wan gewählt werden.
BEGRÜNDUNG:	Diese Funktion verändert die Umgebung und die visuelle Gestaltung des Spiels, aber nicht die Kernspielmechanik.
ABHÄNGIGKEITEN:	-
RISIKEN:	Erhöhter Entwicklungsaufwand für neue Umgebungen.

ID	NFA 14
TITEL:	Kamerafolgung beim Start des Raumschiffs
BESCHREIBUNG:	Zu Beginn des Spiels gibt es eine Kameraführung, die das Raumschiff kurz verfolgt.
BEGRÜNDUNG:	Diese Funktion betrifft die visuelle Darstellung und nicht die zugrundeliegende Spielmechanik.
ABHÄNGIGKEITEN:	-
RISIKEN:	Zusätzlicher Entwicklungsaufwand für die Kameraführung.

ID	NFA 15
TITEL:	Barrel Rolls
BESCHREIBUNG:	Das Raumschiff kann Barrel Rolls (Kreisflüge) durchführen.
BEGRÜNDUNG:	Diese Funktion ist eine zusätzliche Gameplay-Möglichkeit, die die Kontrolle und das visuelle Erlebnis erweitert, jedoch keine grundlegenden Spielfunktionen betrifft.
ABHÄNGIGKEITEN:	NFA8
RISIKEN:	Schwierigkeit bei der Implementierung von Flugeffekten.

ID	NFA 16
TITEL:	Schäden beim Verlieren eines Lebens
BESCHREIBUNG:	Das Raumschiff wird beschädigt und zeigt sichtbare Veränderungen wie brennende Stellen.
BEGRÜNDUNG:	Diese Funktion betrifft nur die visuelle Darstellung der Spielfigur und nicht die zugrunde liegende Spielmechanik.
ABHÄNGIGKEITEN:	-
RISIKEN:	Möglicherweise zusätzliche Performance-Anforderungen durch Schadensdarstellung.

ID	NFA 17
TITEL:	Geheimfunktionen in der Read.me
BESCHREIBUNG:	Bestimmte Code-Funktionen, wie ”Order 66” oder ”Starkiller”, sind nur in der Read.me des Spiels dokumentiert.
BEGRÜNDUNG:	Diese Funktionen sind optionale Spielmechaniken, die nicht die grundlegende Spielsteuerung betreffen, sondern lediglich zusätzliche Inhalte darstellen
ABHÄNGIGKEITEN:	-
RISIKEN:	Niemand findet die

5 Systemstruktur

Die Systemstruktur beschreibt die wesentlichen Komponenten und deren Interaktionen innerhalb des Spiels. In diesem Kapitel wird die Architektur des Spiels im Detail dargestellt, um einen Überblick über die verschiedenen Systembestandteile und ihre Zusammenarbeit zu geben. Es wird gezeigt, wie die einzelnen Module – von der Benutzeroberfläche bis hin zu den Spiellogiken – in ein funktionierendes System integriert werden. Dabei wird auf die Struktur und Organisation der Hauptkomponenten, sowie auf die Kommunikationsflüsse zwischen ihnen eingegangen.

5.1 Zustandsdiagramme

In diesem Abschnitt werden die Zustandsdiagramme für Starfighter Alliance vorgestellt. Sie zeigen die verschiedenen Phasen des Spiels und die Übergänge zwischen den Zuständen. Eine detaillierte Beschreibung der Diagramme folgt.

5.1.1 Screens

Das Diagramm beginnt mit dem Starten der Anwendung, wodurch der erste Zustand, der Start-Screen, aktiviert wird. Dieser Zustand ist in die Unterzustände Start und Settings unterteilt. Im Start-Zustand erfolgt die Initialisierung der Tasten (Entry), und die Hauptfunktion wartet auf die Eingabe des Spielers (Do). Der Spieler hat drei Optionen: den Exit-Button drücken, um die Anwendung zu verlassen, den Settings-Button, um in den Settings-Zustand zu wechseln. Im Settings-Zustand wird beim Eintritt die Initialisierung der Einstellungen vorgenommen (Entry), die Do-Funktion zeigt alle Optionen an, und beim Verlassen (Exit) werden die Eingaben des Spielers gespeichert. Der Spieler kann zurück zum Start-Zustand oder zum nächsten Zustand über den Start-Button wechseln. Im Selection-Zustand werden alle Raumschiffe angezeigt (Entry: Show all spaceships), und die Do-Funktion wartet auf die Eingabe des Spielers. Drückt der Spieler den Back-Button, geht es zurück zum Start-Screen. Wählt der Spieler ein Raumschiff, wird die Konfiguration gespeichert (Exit: Save player config) und der Übergang zum Game-Screen erfolgt. Im Game-Screen-Zustand startet der Countdown-Zustand (Entry: Start Timer), wartet auf den Timer (Do: Wait for timer) und spielt das Startgeräusch ab (Exit: Play start sound). Während des Countdowns gibt es zwei Optionen: Drückt der Spieler den Pause-Button, wechselt der Zustand in den Pause-Zustand (Entry: Initialize buttons, Do: Wait for player input), und nach Drücken des Back-Buttons geht es zurück zum Countdown, wenn dieser noch nicht abgelaufen ist. Ist der Countdown vorbei, geht es in den Run-Zustand. Sobald der Countdown abläuft, geht es automatisch in den Run-Zustand. Im Run-Zustand sind vier Subzustände aktiv, die in den nächsten Kapiteln detailliert beschrieben werden. Die Do-Funktion zeigt die Spielstatistiken an (Show stats). Die Abbruchbedingungen für alle Subzustände sind der Pause-Button oder die Zerstörung des Raumschiffs, was zum End-Screen führt. Im End-Screen-Zustand wird der Punktestand gespeichert (Exit), und der Highscore wird angezeigt (Do). Der Spieler kann entweder den Restart-Button drücken, um zurück zum Selection-Screen zu wechseln, oder die Anwendung über den Exit-Button verlassen.

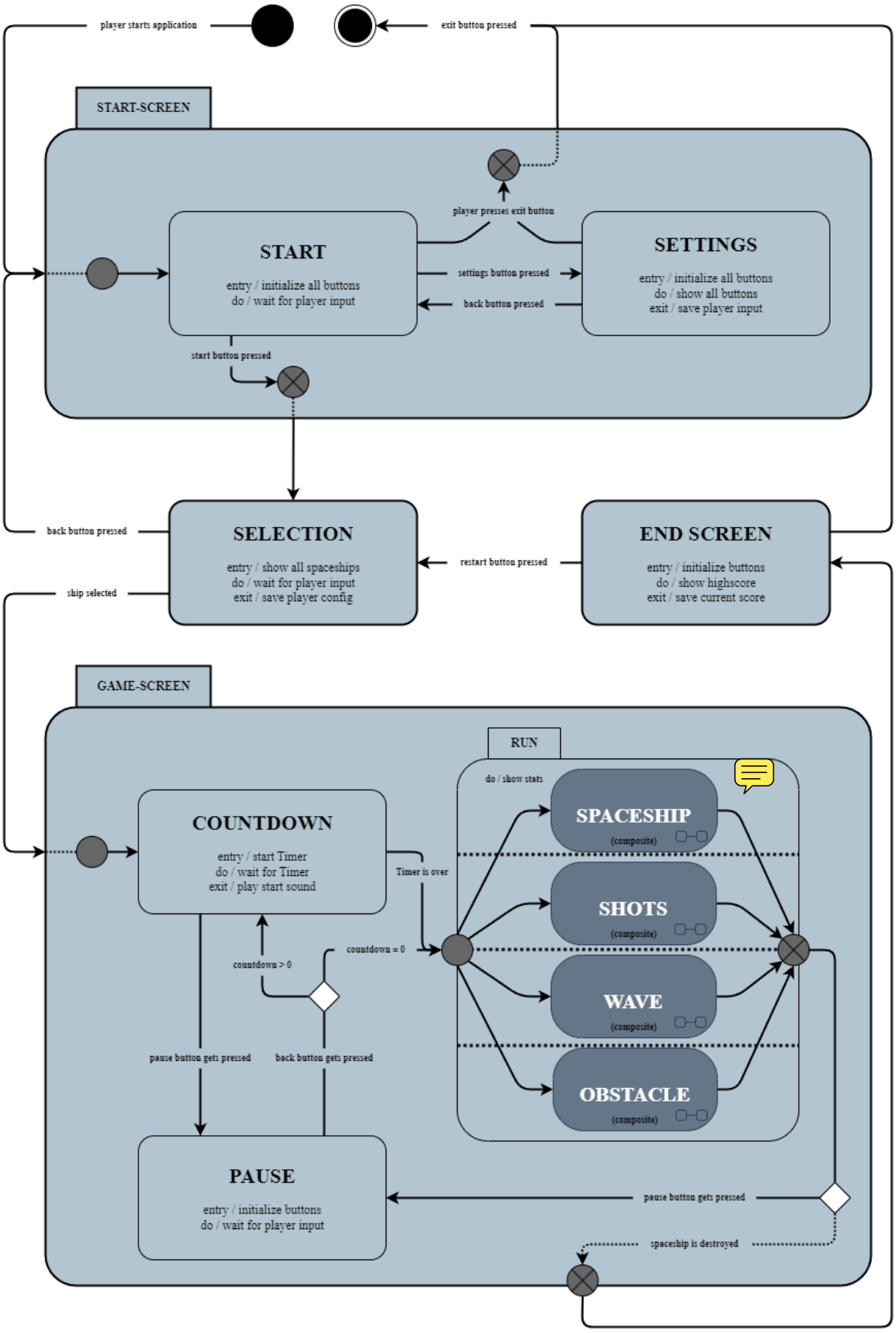

Figure 1: screens-state-diagram.png

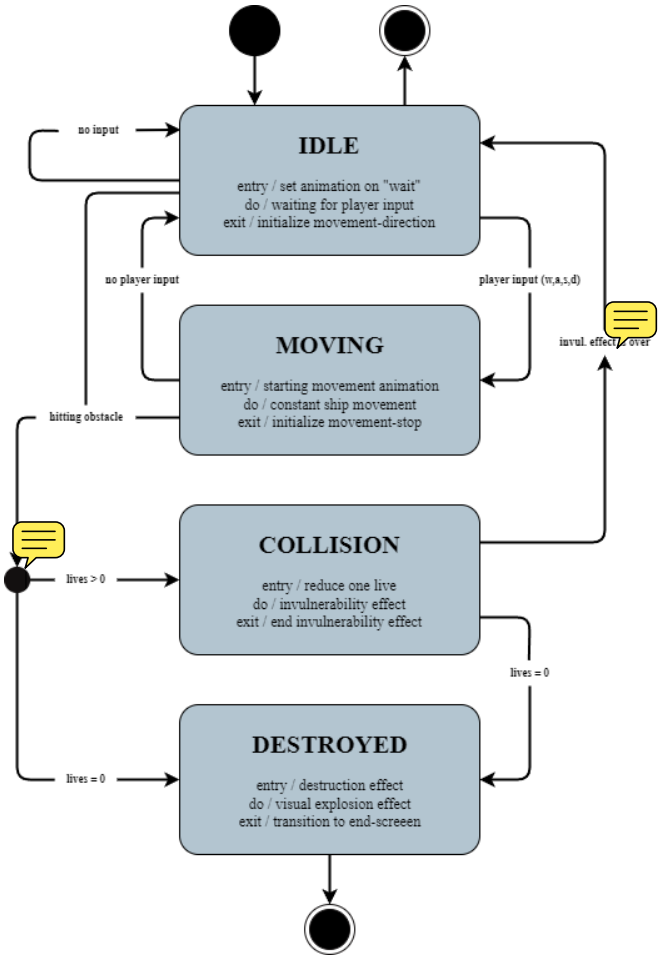

Figure 2: spaceship-state-diagram.png

5.1.2 Spaceship

Das Diagramm beschreibt die Zustände des Raumschiffs im Spiel. Zu Beginn befindet sich das Raumschiff im Zustand Idle, welcher als Ruhemodus fungiert. Beim Eintritt (Entry) wird die Animation auf „Warten“ gesetzt, die Do-Funktion wartet auf eine Eingabe des Spielers, und beim Verlassen (Exit) wird die Bewegungsrichtung initialisiert. Solange keine Eingabe erfolgt, bleibt das Raumschiff im Idle-Zustand. Erfolgt jedoch ein Spielerinput (z. B. durch die Tasten W, A, S oder D), wechselt es in den Moving-Zustand. Im Moving-Zustand wird beim Eintritt die Bewegungsanimation gestartet, die Do-Funktion sorgt für die kontinuierliche Bewegung des Raumschiffs, und beim Verlassen wird die Bewegung gestoppt. Hört der Spielerinput auf, wechselt das Raumschiff zurück in den Idle-Zustand. Trifft das Raumschiff in einem der Zustände (Idle oder Moving) auf ein Hindernis, wird der Übergang abhängig von der verbleibenden Lebensanzahl gewählt. Sind die Leben auf 0 gesunken, wechselt das Raumschiff in den Destroyed-Zustand. In diesem Zustand wird beim Eintritt ein Zerstörungseffekt ausgelöst, die Do-Funktion zeigt eine visuelle Explosion, und beim Verlassen erfolgt der Übergang zum End-Screen. Wenn noch Leben übrig sind, wechselt das Raumschiff in den Collision-Zustand. Beim Eintritt wird ein Leben abgezogen, die Do-Funktion aktiviert einen Unverwundbarkeitseffekt, und beim Verlassen endet dieser Effekt. Zur Sicherheit wird in diesem Zustand erneut geprüft, ob die Lebensanzahl tatsächlich nicht auf 0 gefallen ist. Nach Ablauf des Unverwundbarkeitseffekts kehrt das Raumschiff in den Idle-Zustand zurück.

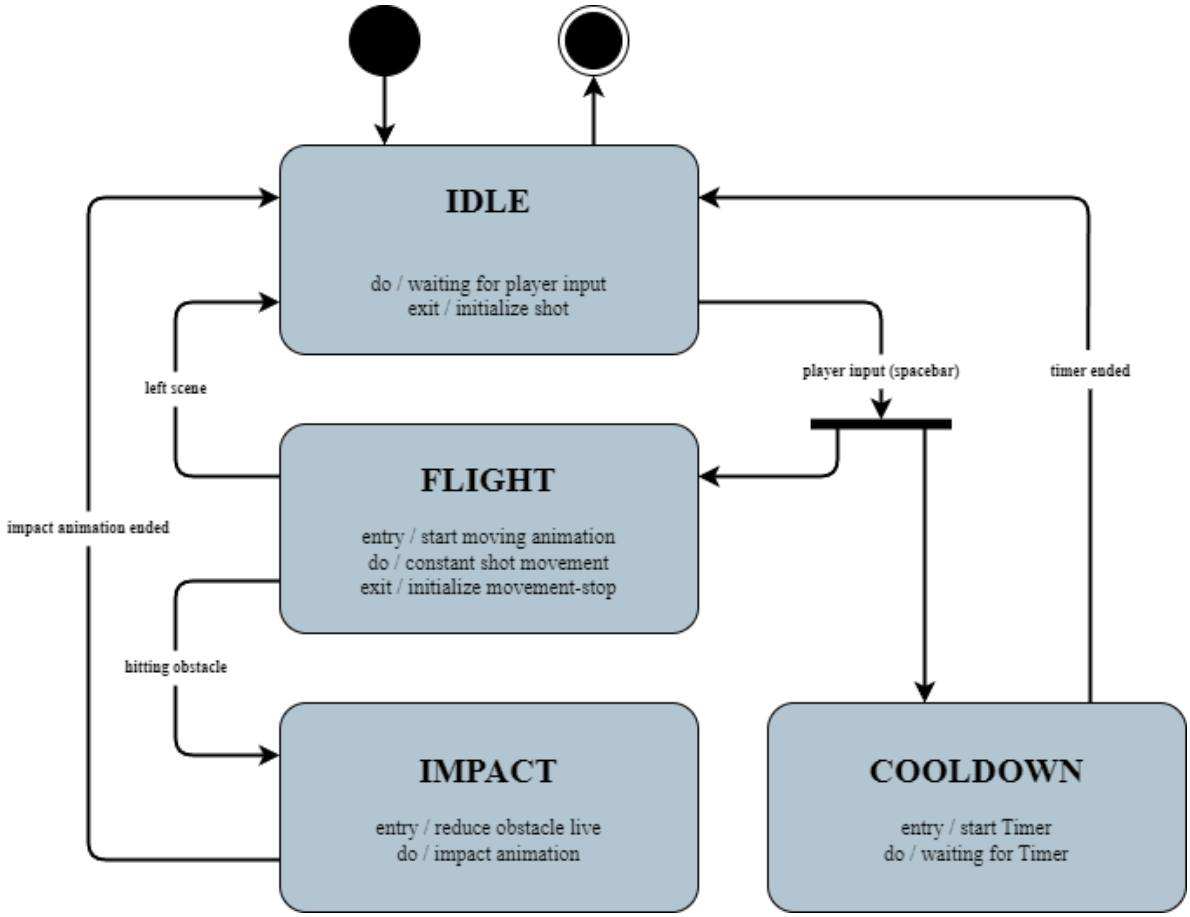

Figure 3: shots-state-diagram.png

5.1.3 Shot

Im Idle-Zustand wartet das System auf die Eingabe des Spielers (do: waiting for player input). Bei einem Tastendruck auf die Leertaste aktiviert sich die Schussmechanik, wodurch die Zustände Cooldown und Flight gleichzeitig eintreten. Im Cooldown-Zustand startet ein Timer (entry: start Timer), und während der Laufzeit des Timers kann kein weiterer Schuss abgegeben werden (do: waiting for timer). Sobald der Timer abgelaufen ist, wechselt das System zurück in den Idle-Zustand. Im Flight-Zustand wird die Bewegung des Schusses gestartet (entry: start moving animation) und währenddessen kontinuierlich fortgesetzt (do: constant shot movement). Verlässt der Schuss den Bildschirm, wird ebenfalls zum Idle-Zustand gewechselt. Trifft der Schuss jedoch ein Hindernis, erfolgt ein Wechsel in den Impact-Zustand mit einer Reduktion der Lebenspunkte des Hindernisses (entry: reduce obstacle life) und der Darstellung der Einschlag-Animation (do: impact animation). Nach Abschluss der Animation kehrt das System in den Idle-Zustand zurück.

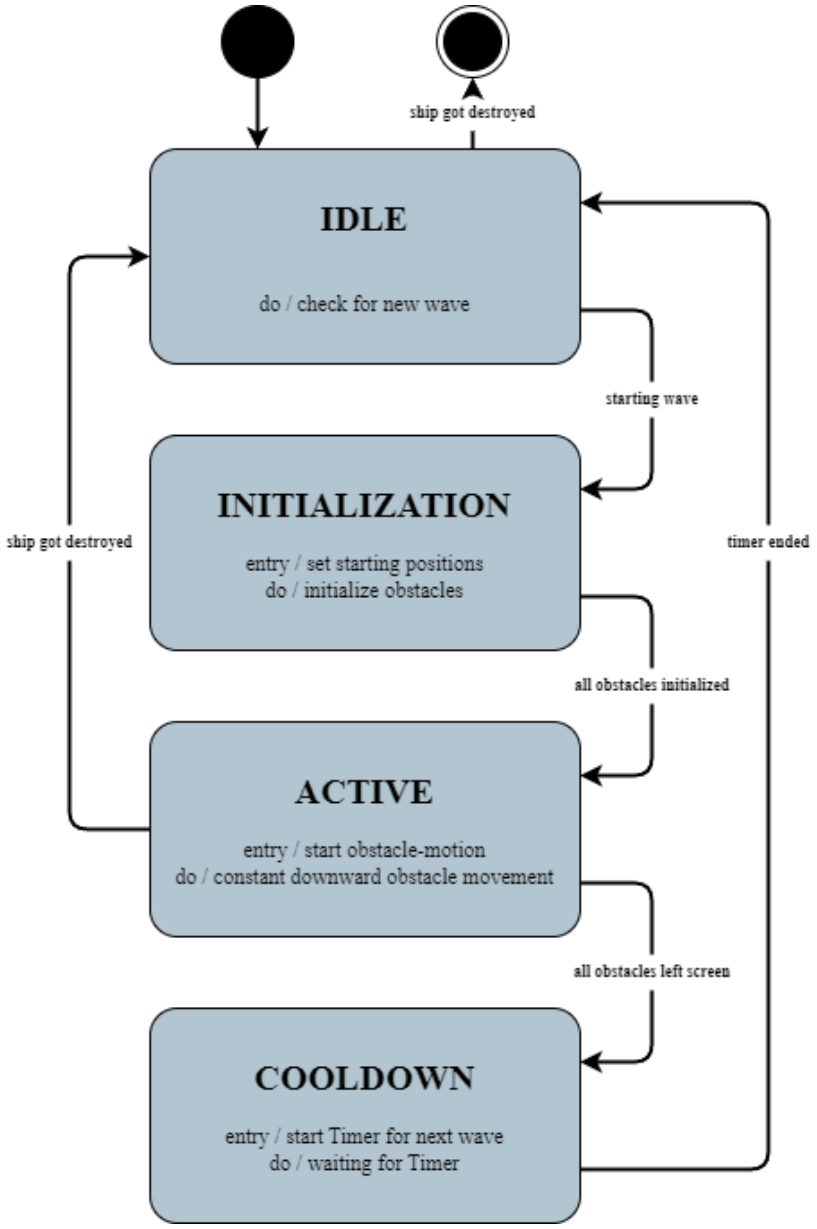

Figure 4: wave-state-diagram.png

5.1.4 Wave

Wir beginnen im Idle-Zustand, der fortlaufend auf den Start einer neuen Welle prüft (do: check for new wave). Mit Beginn einer Welle wechselt das System in den Initialization-Zustand (entry: set starting positions der Hindernisse, abhängig vom Abstand) und startet die Erzeugung aller Hindernisse (do: initialize obstacles). Sobald alle Hindernisse bereit sind, wechselt das System in den Active-Zustand, wo die Hindernisbewegung eingeleitet wird (entry: start obstacle motion) und ein kontinuierlicher Abwärtsbewegungsprozess läuft (do: constant downward obstacle movement), wodurch die Welle aktiv dargestellt wird. Falls das Raumschiff zerstört wird, kehrt das System in den Idle-Zustand zurück oder verlässt das Diagramm. Sobald alle Hindernisse den Bildschirm verlassen haben, wechselt das System in den Cooldown-Zustand, in dem ein Timer für die nächste Welle gestartet wird (entry: start timer for next wave, do: waiting for timer). Bei Ablauf des Timers wechselt das System zurück in den Idle-Zustand.

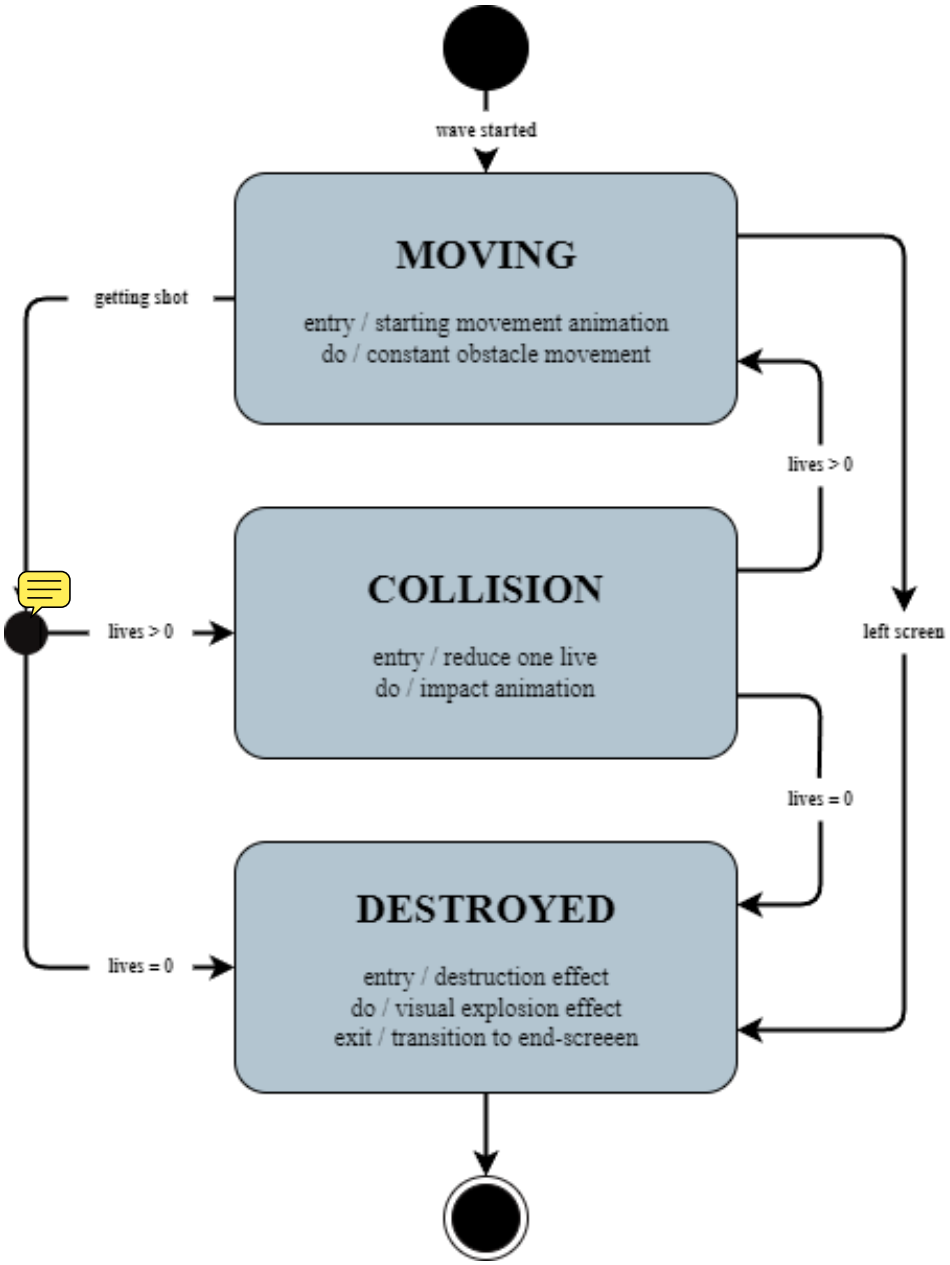

Figure 5: obstacle-state-diagram.png

5.1.5 Obstacle

Wenn eine Welle startet, beginnt das Hindernis im Moving-Zustand, in dem die Bewegungsanimation startet (Entry: starting movement animation) und die konstante Bewegung erfolgt (Do: constant obstacle movement). Wird das Hindernis getroffen, gilt: Hat es keine Leben mehr, wechselt es in den Destroyed-Zustand mit Zerstörungseffekt (Entry: destruction effect), visueller Explosion (Do: visual explosion effect) und Übergang zum End-Screen beim Verlassen. Hat das Hindernis noch Leben, wechselt es in den Collision-Modus, in dem ein Leben abgezogen (Entry: reduce one life) und eine Aufprall-Animation gezeigt wird (Do: impact animation). Danach wird erneut geprüft: Ist das Leben auf 0, wechselt es in den Destroyed-Zustand, andernfalls kehrt es in den Moving-Zustand zurück. Verlässt das Hindernis im Moving-Zustand den Bildschirm, erfolgt ebenfalls der Übergang in den Destroyed-Zustand.

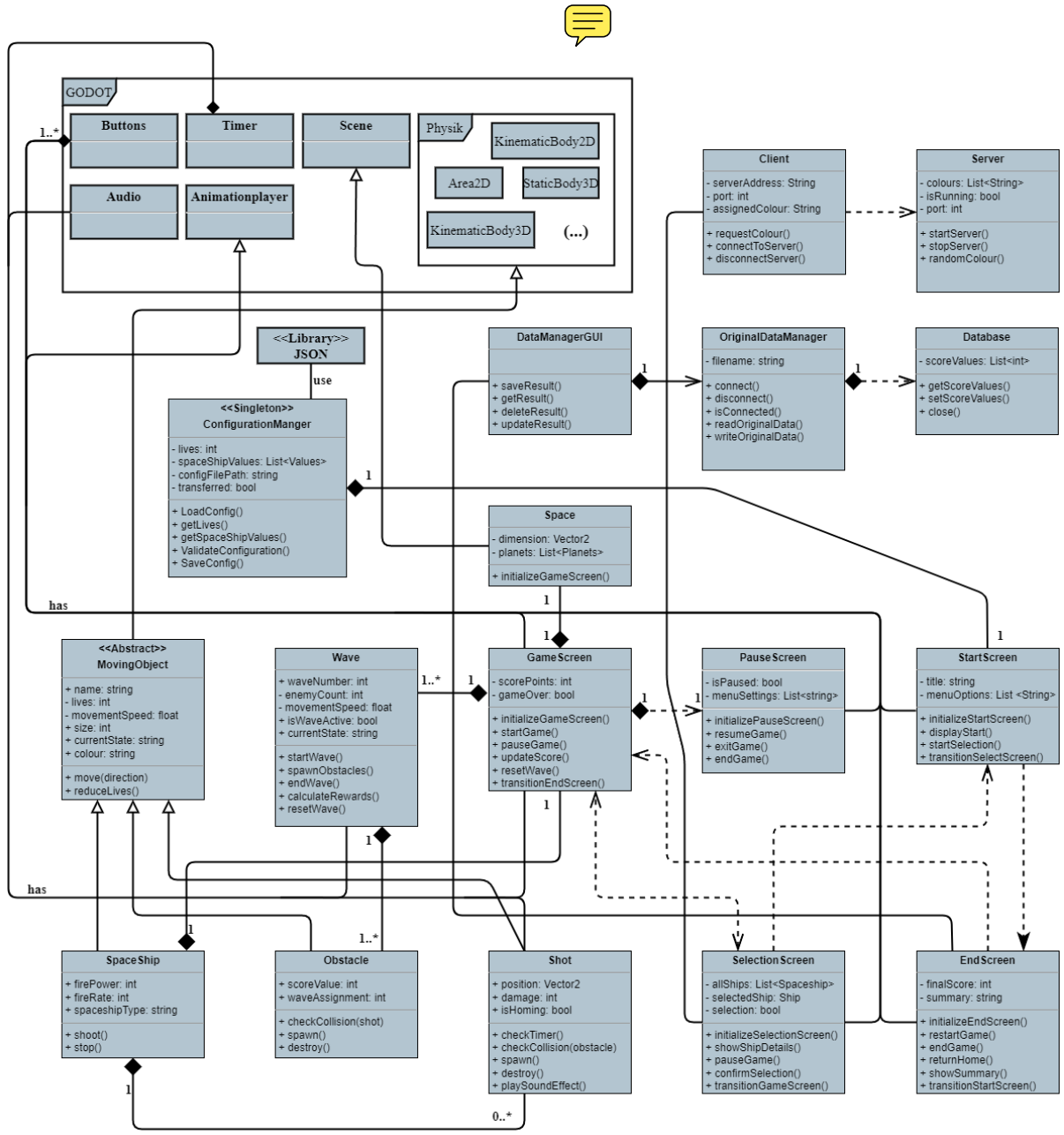

Figure 6: uml-class-diagram.png

5.2 Klassendiagramm

In diesem Abschnitt wird das Klassendiagramm[4] für Starfighter Alliance vorgestellt. Es zeigt die verschiedenen Klassen des Systems, deren Attribute und Methoden sowie die Beziehungen zwischen ihnen. Eine detaillierte Beschreibung des Diagramms nach Ansichten der Software-Architektur folgt.

[4]Referenz aus sparxsystems.com

5.2.1 Logical View (Logische Sicht)

Die logische Sicht beschreibt die statische Struktur des Systems, einschließlich der wichtigsten Klassen und deren Beziehungen:

- **Hauptbildschirme des Spiels**: Die Klassen StartScreen, GameScreen, PauseScreen, EndScreen und SelectionScreen verwalten die verschiedenen Zustände des Spiels.
- **Datenmanagement-Komponenten**: Klassen wie DataManagerGUI, OriginalDataManager und Database kümmern sich um die Speicherung und Verwaltung von Highscores und Spieldaten.
- **Netzwerkkomponenten**: Die Klassen Client und Server ermöglichen Netzwerkfunktionen für den Multiplayer oder den Austausch von Daten.
- **Spielobjekte und Mechaniken**: Klassen wie SpaceShip, Wave, Obstacle und Shot modellieren die Spielfunktionalitäten. Die Basisklasse MovingObject fasst Eigenschaften und Methoden für bewegliche Objekte zusammen.

5.2.2 Implementation View (Implementierungssicht)

Die Implementierungssicht beschreibt die Organisation des Codes und die Verwendung von Bibliotheken und APIs:

- **Datenmanagement und JSON-Verarbeitung**:
  - Die Klasse ConfigurationManager liest eine JSON-Konfigurationsdatei ein, um zentrale Spieleinstellungen wie Raumschiffwerte und Leben bereitzustellen.
  - DataManagerGUI und OriginalDataManager organisieren den Zugriff auf die Database[5] zur Speicherung von Highscores.
- **Godot-API-Integration**[6]:
  - Komponenten wie Button, Timer, AnimationPlayer und KinematicBody2D aus der Godot-Engine werden für Benutzerinteraktionen, Animationen, Bewegung und Physik genutzt.

[5] Aufbau der Datenklasse aus researchgate.net
[6] GODOT Implementierung übernommen aus docs.godotengine.org

5.2.3 Process View (Prozesssicht)

Die Prozesssicht beschreibt die dynamischen Abläufe und Interaktionen der Klassen während der Laufzeit:

- **Spielsteuerung**: Die Klasse GameScreen verwaltet den Hauptspielablauf. Methoden wie startWave() und updateScore() steuern die Spiellogik.
- **Interaktionen zwischen Spielobjekten**: Objekte wie SpaceShip, Shot und Obstacle interagieren dynamisch, zum Beispiel durch Kollisionserkennung.
- **Datenoperationen**: DataManagerGUI ruft Daten aus der Database ab oder speichert neue Einträge.
- **Netzwerkoperationen**: Client und Server synchronisieren Daten über das Netzwerk.

5.2.4 Deployment View (Verteilungssicht)

Die Deployment-Sicht beschreibt die physikalische Verteilung der Software-Komponenten:

- **Lokale Architektur**: Das Spiel läuft auf einem Client-Gerät mit der Godot-Engine, die sowohl für Rendering als auch für die Spiellogik verantwortlich ist.
- **Netzwerkarchitektur**: Der Server wird auf einer separaten Maschine oder in der Cloud betrieben, um Daten von mehreren Clients zu synchronisieren.
- **Datenhaltung**: Die Database (zum Beispiel SQLite oder ein anderes lokales Speichersystem) speichert die Highscores auf dem Client oder Server.

5.2.5 Benutzeroberfläche und Godot-Integration

Das System nutzt verschiedene Komponenten der Godot-API für die Benutzeroberfläche und Spielfunktionen:

- **Interaktive Elemente**: Button und Timer werden in den Bildschirmen (StartScreen, PauseScreen etc.) verwendet, um Benutzerinteraktionen zu steuern.
- **Audio und Animationen**: Die Komponenten Audio und AnimationPlayer werden genutzt, um Soundeffekte und Animationen abzuspielen.
- **Physik und Kollisionserkennung**: Komponenten wie KinematicBody2D und StaticBody2D ermöglichen Bewegung und Kollisionserkennung für Objekte wie SpaceShip, Shot und Obstacle.