



Figure 1: HSMW-Logo

**Hochschule Mittweida** Fakultät Angewandte Computer- & Biowissenschaften Professur Medieninformatik

**Belegarbeit im Modul: Game Programming**

**Erstprüfer:** Prof. Dr. rer. nat. Marc Ritter **Co-Prüfer:** Manuel Heinzig, M.Sc., Alexander Kühn, M.Sc.

**Studierender:** Niki Hendel

**Sommersemester 2025**

# Inhaltsverzeichnis

- 3. Einleitung
- 4. Game Design Konzept
- 5. Architektur des Spiels
- 6. Technische Besonderheiten
- 7. Modifikationen und Eigenleistungen
- 8. Spielsteuerung
- 9. Fazit
- 10. Literaturverzeichnis
- 11. Selbstständigkeitserklärung

# Einleitung

In dieser Belegarbeit geht es um die Entwicklung eines Shoot 'em up Space Games. Mein Ziel ist es, ein dynamisches und unterhaltsames Spiel zu kreieren, das Spaß macht und gleichzeitig ein paar interessante Mechaniken bietet.

Für die Umsetzung setze ich auf die Simple DirectMedia Layer (SDL)-Bibliothek, die mir hilft, Grafik, Audio und Eingaben effizient zu handhaben.

Neben den Grundmechaniken gibt es verschiedene Level mit unterschiedlichen Herausforderungen, eine bunte Mischung an Gegner-Typen und ein Scoring-System, das für Wettbewerbsspaß sorgt.

In dieser Arbeit dokumentiere ich den gesamten Entwicklungsprozess – von der ersten Idee bis zum fertigen Spiel. Ich gehe auf die verschiedenen Phasen der Entwicklung ein, beschreibe die verwendeten Technologien und Methoden und reflektiere über die Herausforderungen und Lösungen, die während des Projekts aufgetaucht sind. Am Ende gibt es noch einen Ausblick auf mögliche Erweiterungen und Verbesserungen für die Zukunft.

# Game Design Konzept

## Genre

Shoot 'em up, Arcade

## Spielwelt

“game name” spielt in einer pixeligen Weltraumumgebung, inspiriert von Arcade-Spielen der 80er. Die Grafik nutzt eine begrenzte Farbpalette und einfache geometrische Formen.

## Spielziel

Sammle Punkte durch das Besiegen von Gegnern und das Abschließen von Levels. Navigiere dein Raumschiff durch Gegnerwellen und überlebe ihre Angriffe.

## Spielmechaniken

- **Steuerung:** Einfache Tastensteuerung für ein klassisches Arcade-Feeling.
- **Gegner:** Unterschiedliche Gegner mit einfachen, aber herausfordernden Bewegungsmustern.
- **Leveldesign:** Lineare Level mit zunehmendem Schwierigkeitsgrad.
- **Scoring-System:** Highscore-System zur Motivation.

## Grafik und Sound

Pixelige Grafik und 8-Bit-Soundeffekte mit einfachen, sich wiederholenden Hintergrundmelodien.

# Architektur des Spiels

Das Spiel "Ahhhh Space-Game oder so" ist als klassisches 2D-Shoot 'em up konzipiert und nutzt eine einfache, aber effektive Architektur, um ein flüssiges und reaktionsschnelles Spielerlebnis zu bieten. Die Architektur des Spiels lässt sich in mehrere Hauptkomponenten unterteilen: Eingabe, Spiel-Logik, Rendering und Audio.

## Eingabe

Die Eingabe-Komponente ist für die Verarbeitung von Benutzereingaben zuständig. Diese Komponente nutzt die SDL-Bibliothek, um Tastatur- und Joystick-Eingaben zu erfassen. SDL bietet eine einfache und effiziente Möglichkeit, Benutzereingaben zu verarbeiten und sicherzustellen, dass das Spiel auf verschiedene Eingabegeräte reagieren kann. Die Eingabe-Komponente übergibt die verarbeiteten Eingaben an die Spiel-Logik, die entsprechend auf die Benutzeraktionen reagiert.

## Spiel-Logik

Die Spiel-Logik ist das Herzstück des Spiels und enthält die Hauptmechaniken und Regeln. Diese Komponente ist für die Aktualisierung des Spielzustands, die Kollisionserkennung und die Steuerung der Spielobjekte verantwortlich. Die Spiel-Logik wird in einer Hauptschleife ausgeführt, die kontinuierlich den Zustand des Spiels aktualisiert und auf Benutzereingaben reagiert.

Die Spiel-Logik umfasst auch die Verwaltung von Spielobjekten wie dem Raumschiff des Spielers, Gegnern und Projektilen. Jedes Spielobjekt hat seine eigenen Eigenschaften und Verhaltensweisen, die durch die Spiel-Logik gesteuert werden.

## Rendering

Die Rendering-Komponente ist für die Darstellung des Spiels auf dem Bildschirm verantwortlich. Auch hier kommt die SDL-Bibliothek zum Einsatz, die eine effiziente Handhabung von 2D-Grafiken ermöglicht. Die Rendering-Komponente nimmt die Spielobjekte und ihren aktuellen Zustand von der Spiel-Logik entgegen und rendert sie auf dem Bildschirm.

SDL bietet Funktionen zum Laden und Darstellen von Sprites, zum Zeichnen von geometrischen Formen und zum Verwalten von Animationen. Die Rendering-Komponente nutzt diese Funktionen, um die Spielwelt und ihre Objekte darzustellen.

## **Audio**

Die Audio-Komponente ist für die Wiedergabe von Soundeffekten und Hintergrundmusik zuständig. SDL bietet auch hier einfache und effiziente Funktionen zum Laden und Abspielen von Audio-Dateien. Die Audio-Komponente spielt Soundeffekte für verschiedene Spielereignisse ab, wie z.B. das Abfeuern von Projektilen, Kollisionen.

## **Zusammenarbeit der Technologien**

Die verschiedenen Komponenten des Spiels arbeiten eng zusammen, um ein kohärentes und unterhaltsames Spielerlebnis zu bieten. Die Eingabe-Komponente erfasst die Benutzereingaben und übergibt sie an die Spiel-Logik, die den Spielzustand aktualisiert. Die Rendering-Komponente nimmt den aktualisierten Spielzustand entgegen und stellt ihn auf dem Bildschirm dar. Gleichzeitig spielt die Audio-Komponente passende Soundeffekte ab, um das Spielerlebnis zu bereichern.

Die Verwendung von SDL als zentrale Bibliothek ermöglicht eine effiziente und einfache Handhabung der verschiedenen Komponenten. SDL bietet eine einheitliche Schnittstelle für die Eingabe, das Rendering und die Audio-Wiedergabe, was die Integration und Zusammenarbeit der verschiedenen Technologien erleichtert.

Durch diese Architektur und die Verwendung von SDL kann "Ahhhh Space-Game oder so" ein flüssiges und reaktionsschnelles Spielerlebnis bieten, das an die Arcade-Klassiker der 80er Jahre erinnert.

## Technische Besonderheiten

## Technische Besonderheiten

### Kollisionserkennung

Die Kollisionserkennung ist ein zentraler Aspekt des Spiels, der sicherstellt, dass Interaktionen zwischen Spielobjekten korrekt erkannt und behandelt werden. Für die Kollisionserkennung wird ein einfaches, aber effektives System verwendet, das auf der Achsenausgerichteten Begrenzungsbox (Axis-Aligned Bounding Box, AABB) basiert.

- **AABB Kollisionserkennung:** Jedes Spielobjekt, einschließlich des Raumschiffs des Spielers, der Gegner hat eine Begrenzungsbox, die seine physische Präsenz im Spiel darstellt. Die Kollisionserkennung überprüft, ob sich diese Begrenzungsboxen überschneiden. Wenn eine Überschneidung erkannt wird, wird eine Kollision ausgelöst, und das Spiel reagiert entsprechend.
- **Optimierung:** Um die Leistung zu optimieren, wird die Kollisionserkennung nur für Objekte durchgeführt, die sich in der Nähe zueinander befinden. Dies reduziert die Anzahl der notwendigen Überprüfungen und verbessert die Effizienz des Spiels.

### SDL für Steuerung und Interaktion

Die Simple DirectMedia Layer (SDL)-Bibliothek wird im Spiel für verschiedene Aspekte der Steuerung und Interaktion verwendet. SDL bietet eine plattformübergreifende Schnittstelle, die die Entwicklung von Spielen vereinfacht und eine effiziente Handhabung von Eingaben, Grafiken und Audio ermöglicht.

- **Eingabehandhabung:** SDL wird verwendet, um Benutzereingaben von der Tastatur und dem Joystick zu erfassen. Die Bibliothek bietet Funktionen zum Abfragen des Zustands von Tasten und Joystick-Achsen, was eine präzise und reaktionsschnelle Steuerung des Raumschiffs ermöglicht. Die Eingabe-Komponente des Spiels nutzt diese Funktionen, um die Bewegungen und Aktionen des Spielers zu erfassen und an die Spiel-Logik weiterzugeben.
- **Rendering:** SDL wird auch für das Rendering der Spielgrafiken verwendet. Die Bibliothek bietet Funktionen zum Laden und Darstellen von Sprites, zum Zeichnen von geometrischen Formen und zum Verwalten von Animationen. Die Rendering-Komponente des Spiels nutzt diese Funktionen, um die Spielwelt und ihre Objekte auf dem Bildschirm darzustellen.
- **Audio-Wiedergabe:** SDL bietet einfache und effiziente Funktionen zum Laden und Abspielen von Audio-Dateien. Die Audio-Komponente des Spiels nutzt

diese Funktionen, um Soundeffekte und Hintergrundmusik abzuspielen und so das Spielerlebnis zu bereichern.

Durch die Verwendung von SDL für die Steuerung, das Rendering und die Audio-Wiedergabe kann das Spiel ein flüssiges und reaktionsschnelles Spielerlebnis bieten, das an die Arcade-Klassiker der 80er Jahre erinnert.



# Modifikationen und Eigenleistungen

## Modifikationen an bestehenden Vorlagen oder Tutorials

- **SDL Initialisierung und Hauptschleife:** Basierend auf einem grundlegenden SDL-Tutorial wurde die Initialisierung der SDL-Bibliothek und die Hauptschleife des Spiels implementiert. Diese Vorlage wurde erweitert, um zusätzliche Funktionen wie die Handhabung von Benutzereingaben, das Rendering von Spielobjekten und die Kollisionserkennung zu integrieren.
- **Kollisionserkennung:** Ein einfaches Kollisionserkennungssystem wurde aus einem bestehenden Beispiel übernommen und an die Bedürfnisse des Spiels angepasst. Das System wurde erweitert, um verschiedene Arten von Kollisionen zu handhaben und die Spiel-Logik entsprechend zu aktualisieren.

## Beschreibung der Eigenleistungen

Neben den Modifikationen an bestehenden Vorlagen und Tutorials wurden zahlreiche Eigenleistungen erbracht, um das Spiel zu entwickeln und zu verbessern.

- **Spiel-Logik und -Mechaniken:** Die gesamte Spiel-Logik, einschließlich der Steuerung des Raumschiffs und der Handhabung wurde von Grund auf entwickelt. Dies umfasste die Erstellung von Funktionen zur Aktualisierung des Spielzustands, zur Kollisionserkennung und zur Steuerung der Spielobjekte.
- **Benutzeranpassungen:** Funktionen zur Anpassung der Steuerung und des Spielverhaltens wurden entwickelt, um den Spielern die Möglichkeit zu geben, das Spiel an ihre persönlichen Vorlieben anzupassen. Dies umfasste die Implementierung von Optionen zur Konfiguration der Tastatur- und Joystick-Steuerung.

# Spielsteuerung

## Tastatursteuerung

Die Steuerung des Spiels ist einfach und intuitiv gestaltet, um ein klassisches Arcade-Erlebnis zu bieten. Die folgenden Tasten werden verwendet, um das Raumschiff zu steuern und mit dem Spiel zu interagieren:

Taste	Funktion
Pfeil hoch / Arrow up	Bewegen Sie das Raumschiff nach oben
Pfeil links / Arrow left	Bewegen Sie das Raumschiff nach links
Pfeil runter / Arrow down	Bewegen Sie das Raumschiff nach unten
Pfeil rechts / Arrow right	Bewegen Sie das Raumschiff nach rechts
Leertaste	Schießen
R	Restart
Esc	Spiel pausieren

## Menüsteuerung

Im Menü können Sie die folgenden Tasten verwenden, um durch die Optionen zu navigieren und Einstellungen zu ändern:

Taste	Funktion
W	Navigieren Sie nach oben
A	Navigieren Sie nach links
S	Navigieren Sie nach unten
D	Navigieren Sie nach rechts
Enter	Auswahl bestätigen
Delete	Zurückgehen

## Fazit

In dieser Arbeit wurde die Entwicklung eines Shoot 'em up Arcade-Spiels beschrieben, das sich an den Klassikern der 80er Jahre orientiert. Das Ziel bestand darin, ein einfaches, aber unterhaltsames Spielerlebnis zu schaffen einer lebendigen Spielwelt und einigen experimentellen physikalischen Elementen bereichert wird.

Während des Entwicklungsprozesses wurden verschiedene Technologien und Techniken eingesetzt, darunter die Verwendung der SDL-Bibliothek für die Handhabung von Eingaben, Grafiken und Audio. Die Architektur des Spiels wurde so gestaltet, dass sie eine effiziente und reaktionsschnelle Spielmechanik ermöglicht. Die Kollisionserkennung und die Steuerung des Raumschiffs wurden erfolgreich implementiert, um ein flüssiges und fesselndes Spielerlebnis zu bieten.

Insgesamt konnte durch diese Arbeit ein funktionsfähiges und unterhaltsames Arcade-Spiel entwickelt werden, das an die Klassiker der 80er Jahre erinnert und gleichzeitig moderne Spielmechaniken integriert.

## Literaturverzeichnis

- SDL Dokumentation: Offizielle Dokumentation der SDL-Bibliothek, die für die Entwicklung des Spiels verwendet wurde.
- A Beginner's Guide to Game Development with SDL: Tutorials zur Einführung in die Spielentwicklung mit SDL.
- Game Programming Patterns: Ein Buch über Programmiermuster in der Spieleentwicklung, das als Inspiration für die Architektur des Spiels diente.

# Anhang

Im Anhang finden Sie zusätzliche Materialien, die im Rahmen dieser Arbeit erstellt oder verwendet wurden.

## Bilder

Hier können Screenshots des Spiels oder Diagramme der Spielarchitektur eingefügt werden.

Spiel-Screenshot

Figure 2: Spiel-Screenshot

Architektur-Diagramm

Figure 3: Architektur-Diagramm

## **Selbstständigkeitserklärung**

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst habe. Alle verwendeten Quellen und Hilfsmittel sind vollständig angegeben.

[Ort], [Datum]

[Dein Name]