

Bachelorarbeit

Berührungslose Interaktion:

Konzeption und Entwicklung eines Videospiels mit kamerabasierter
Gestensteuerung

Thomas Ehm

Erstprüfer: Prof. Dr. rer. nat. Guido Falkemeier

Zweitprüfer: Prof. Dr.-Ing. Alexander Kutter

Technische Hochschule Ostwestfalen-Lippe

Fachbereich 2 Medienproduktion



Hinweis zur Sprache

Aus Gründen der besseren Lesbarkeit wird in dieser Arbeit stellenweise das generische Maskulinum verwendet. Alle Personenbezeichnungen gelten, sofern nicht anders gekennzeichnet, gleichermaßen für alle Geschlechter.

Inhaltsverzeichnis

01. Einleitung	4
02. Kamerabasierte Eingabemethoden	6
03. Spielidee	9
04. Spielanforderungen	17
05. Technologiewahl	21
06. Umsetzung des Prototyps	28
07. Spielanleitung	45
08. Reflexion & Fazit	65
09. Anhang	67



01

EINLEITUNG

01 Einleitung

Berührungslose Interaktion hat in den vergangenen Jahren durch technologische Innovationen wie der Apple Vision Pro und anderen VR-Brillen, Einzug in heimische Wohnräume und Consumer-Produkte erhalten. Dabei gelten Hand- und Fingergesten als natürliche Eingabeform für virtuelle Interfaces, die sich mit Wisch- und Streichgesten im Raum bedienen lassen. Sie erweitern die wahrnehmbare Realität der Nutzerinnen und Nutzer. Hohe Eintrittshürden schließen jedoch viele potenzielle Kunden aus, denn VR-Brillen und Konsolen sind kostspielig und erfordern meist zusätzliche Hardware mit geeigneter Sensorik. Obwohl es Lösungen gibt, die genauso interessant sein können, mit weniger Ressourcen auskommen und keine teure Extrahardware benötigen, sind diese weniger bekannt. Diese Arbeit geht genau diesen Umstand an, indem ein gestengesteuertes Spiel entwickelt wird, welches ausschließlich die Kamera des Laptops/PCs für Eingaben und zur Steuerung nutzt. Diese Geräte sind bereits mit den benötigten Ressourcen ausgestattet, weit verbreitet und werden täglich von vielen genutzt. Mit „Not So Friendly Neighbours“ wird eine neu interpretierte Web-Variante des Spiels Space Invaders entwickelt, die Hand- und Fingergesten als Eingabe- und Steuerungsmethoden nutzt. Kamerabasiertes Hand-Tracking (MediaPipe Solutions) und eine webbasierte Game Engine (Babylon.js) bilden die technische Basis des Projekts. Der Schwerpunkt dieser Arbeit liegt in der Steuerungslogik und Usability für eine Zielgruppe die wenig bis keine Gaming-Erfahrungen hat. Die Herausforderung liegt einerseits darin, bisherige knopfbasierte Eingabemethoden in berührungslose Gesten zu übersetzen und mit derselben Präzision und Lesbarkeit zu gestalten, wie sie physische Interaktionen bieten. Andererseits soll ein möglichst reduziertes, intuitives und doch verständliches Interface gestaltet werden.

Das Ziel dieses praktischen Projekts ist die Erstellung eines webbasierten Spiels, das mit Handgesten und Körperbewegung gespielt werden kann und lediglich eine handelsübliche Webcam benötigt.

Um einen Spieleprototypen mit berührungsloser Steuerung und einem verständlichen Interface zu entwickeln, werden zunächst existierende Produktlösungen betrachtet. Aus diesen Beobachtungen und der eigenen Spieleidee werden Anforderungen an das Spieleprinzip abgeleitet und in konkrete Kriterien überführt. Anschließend werden geeignete Technologien für die Umsetzung des Spiels anhand dieser Kriterien verglichen. Nach der Auswahl der Technologien wird die praktische Umsetzung der Spielemechanik und der berührungslosen Steuerung dokumentiert. Anhand des Prototyps wird eine Spieleanleitung formuliert die einen Überblick über die relevanten Spielelemente gibt und den Einstieg in das Spiel erleichtern soll. Abschließend wird der Umsetzungsprozess des Spiels und der eingesetzten Programme reflektiert.

02

KAMERABASIERTE EINGABEMETHODEN

02 Kamerabasierte Eingabemethoden

Die ersten Schritte des kamerabasierten Trackings im Bereich der Spielekonsolen wurden in den frühen 2000er gemacht. Mit der EyeToy Kamera und dem gleichnamigen Spiel EyeToy für die Playstation 2, führte Sony eine Art Trackingsystem auf Spielekonsolen ein. In dem Spiel erfasst die Kamera Bewegungen des Körpers, die im Spiel als Aktionen ausgeführt werden. 2010 führte Microsoft mit der Xbox und der Kinect eine Kamera ein, die zusätzlich noch Tiefensensoren für ein genauereres Tracking enthielt.

Während EyeToy und Kinect kamerabasierte Eingabemethoden auf Konsolen und im Spielekontext einführten, hat sich die Idee inzwischen zu einem allgemeinen Interaktionsprinzip weiterentwickelt. Im nächsten Kapitel wird die Gestensteuerung definiert und anhand aktueller Beispiele aus VR/AR sowie dem Alltag eingeordnet.

02.1 Tracking und Gestensteuerung

02.2 Gestaltungsprinzipien für kamerabasierte Gesteninteraktionen

02 Kamerabasierte Eingabemethoden

02.1 Tracking und Gesteuerung

Gesteuerung ist eine Form der Interaktion mit Geräten über Hand- und Körperbewegungen. Anders als bei Touchdisplays, die auf direkte Berührung reagieren, werden bei der Gesteuerung Position und Bewegungen der Hände oder des Körpers als Eingaben erfasst (vgl. Host Europe, 2025). Vor allem moderne VR-Headsets möchten eine immersive Mixed-Reality Erfahrungen durch das Handtracking schaffen, in denen Nutzer per Handgesten direkt mit interaktiven Inhalten interagieren können (vgl. Jeon, Kim & Kim, 2023). Durch das Handtracking Update für die Oculus Quest ist es nun möglich, VR-Videospiele per Gesteuerung zu spielen ohne einen physi-

schen Controller zu verwenden. Meta bietet dafür ein Set aus vier Gesten (Point and Pinch, Pinch and Drag, Palm Pinch und Touch) an, über die in den Spielen interagiert werden kann (vgl. Meta, o.D.). Gesteuerung hat auch in anderen Bereichen des alltäglichen Lebens Einzug gefunden, etwa in Spülmaschinen, LED-Lampen und digitalen Werbetafeln. Im Automobilbereich arbeiten Hersteller an Systemen zur der Lautstärkeregulierung und zur Steuerung von Navigationsfunktionen per Handgesten (vgl. Maerzke, 2023; wizAI solutions GmbH, o.D.; Lewis, 2023).

02.2 Gestaltungsprinzipien für kamerabasierte Gesteuerung

Laut Host Europe ist die Gesteuerung im Vergleich zu bekannten Wischgesten auf Touchdisplays für viele Nutzer noch ungewohnt und empfiehlt bei der Entwicklung eines eigenen gestenbasierten Steuerungskonzepts folgende Punkte zu beachten (vgl. Host Europe, 2025):

- Einfache und natürliche Gesten
- Feedback bereitstellen
- Anpassungen ermöglichen
- Konsistenz in der Gestaltung
- Umfeld berücksichtigen



03

SPIELIDEE

03 Spielidee

Dieses Kapitel beschreibt die konzeptionelle Basis des Spieleprototyps. Zunächst wird das Spielprinzip von Space Invaders zusammengefasst und wesentliche Elemente für das eigene Projekt abgeleitet. Anschließend werden gestalterische Entscheidungen wie das Spielesetting, Figurendesign und Farbkonzept erläutert und die Grundidee der Steuerung skizziert.

03.1 Inspiration: Space Invaders

Space Invaders ist ein Shoot-Em-Up Spiel, in dem eine außerirdische Bedrohung abgewehrt werden muss. Der Spieler bewegt eine Kanone horizontal am unteren Bildschirmrand, um eine Alien-Formation mit Einzelschüssen zu zerstören. Mit jedem eliminierten Gegner bewegt sie die gesamte Welle schneller auf den Spieler zu und ihre Angriffe werden schneller. Zusätzlich stehen dem Spieler temporäre Schutzmauern zur Verfügung, hinter denen er sich kurzzeitig vor den gegnerischen Angriffen verstecken kann. Ziel des Spiels ist alle Gegner zu vernichten bevor sie den unteren Rand des Bildschirms erreichen. Mit jedem Treffer sammelt der Spieler Punkte für den Highscore. (vgl. Wikipedia 2025a; Game Archive - No Commentary Gameplay 2015). In einer späteren Version des Spiels von Herausgeber Activision erhielt der Titel eine kurze Hintergrundgeschichte, die in Form kurzer Videosequenzen erzählt wird. Relevant für diese Arbeit ist insbesondere die Abschlusssequenz nach dem Durchspielen: Im Abspann ist ein gelbes Alien zu sehen welches sich schnell über den Bildschirm bewegt (vgl. Space Invaders Wiki o.D.).

03.2 Idee und Zielgruppe

„Not So Friendly Neighbours“ entstand aus der Idee heraus, ein webbasiertes Spiel zu entwickeln, das an das Arcade-Spiel „Space Invaders“ und deren Ableger angelehnt ist: Gegner erscheinen in Wellen, die sich auf den Spieler zubewegen und müssen rechtzeitig eliminiert werden. Das Spielprinzip wird dabei bewusst simpel gehalten, um den Zugang für Personen zu ermöglichen, die nur wenig Gaming-Erfahrung haben.

Damit der Spiele-Prototyp kein weiterer Klon der Invaders Spiele wird, werden die beschriebenen Spielemente (2.1) neu interpretiert und an das eigene Spiel angepasst.

03 Spielidee > 03.3 Gestaltungselemente des Spiels > Setting

Setting

Als 3D-Space Shooter befindet sich die Spielwelt in einer Weltraumumgebung. Da der Hintergrund kein spielbares Element in der Welt sein soll, muss er möglichst ablenkungsfrei sein und sich klar von den relevanten Inhalten für das Spiel unterscheiden. Für dieses Setting bietet sich die Darstellung einer eher dunkleren Hintergrundumgebung mit vielen kleinen Himmelskörpern an. Für den Prototypen wurde deshalb eine statische Bildtextur als Hintergrundumgebung gewählt, die die dunkle Umgebung des Weltraums und eine hohe Anzahl von Sternen widerspiegelt.



Abbildung 3.1: Blue Nebulae Spheremaps (Spheremap/Hintergrundtextur). Hochgeladen von TonyS, 15.09.25. Quelle: SpaceSphereMaps.com, Zugriff am 26.11.2025, <https://www.spacespheremaps.com/blue-nebulae-spheremaps/>.

03 Spielidee > 03.3 Gestaltungselemente des Spiels > Protagonist

Protagonist

Angelehnt an das gelbe Alien, das in der Endsequenz einer Space Invaders Version zu sehen ist (Abbildung 2.2) und dort als letzte verbleibende Bedrohung erscheint, wird diese Figur in „Not So Friendly Neighbours“ zum Protagonisten. Entsprechend dieser Referenz wird im Spielprinzip auf eine Regeneration von Lebenspunkten verzichtet, da das gelbe Alien in der Vorlage den Angriff nicht übersteht. Die bewusst niedliche gestaltete Figur unterstützt zusätzlich die Umkehrung vom Antagonisten zum Helden (Abbildung 2.3).



Abbildung 3.2: Screenshot aus dem YouTube Video SPACE INVADERS 1999 (Activision) – Ending Scene (Upload: 14:07.2021), Timcode 00:16.
Quelle: YouTube, Zugriff am 20.12.2025, https://www.youtube.com/watch?v=wv_ysajxodg



Abbildung 3.3: Eigene Darstellung, erstellt mit OpenAI (Bildgenerierung)

03 Spielidee > 03.3 Gestaltungselemente des Spiels > Farbe, Stil und Schrift

Farbe, Stil und Schrift

Das Farbkonzept für das User Interface schließt an die gelbe Farbe des Aliens an und ist die primäre Farbe dieses Prototyps. Sie wird für die Darstellung der Zielhilfen, Labels, Bezeichner und für die nutzbaren Fähigkeiten in einer Spielrunde verwendet. In den Menüs wird sie für die Titel und das Highlighten der UI-Buttons verwendet. Alle Elemente mit diesen Farben haben zusätzlich einen Neon-Effekt, durch den sie sich noch stärker vom dunklen Spielhintergrund abheben.

Violett ist die zweite Leitfarbe und wird hauptsächlich für Informationen und Werte (Wellennummer und Score-Punkte) genutzt, die sich im Verlauf einer Spielrunde kontinuierlich ändern. Aktive Elemente wie das Schild und die Projektilen des Spielers werden ebenfalls in violett dargestellt. Die Farbe Rot besitzt ebenfalls einen Neon-Effekt und wird gezielt für negative Effekte, wie den gegnerischen Angriffen, aber auch für die Beenden/Schließen-Buttons in den Menüs, verwendet.

Fließtexte in den Menüs werden in weiß dargestellt, um eine optimale Lesbarkeit auf dem dunklen Hintergrund zu gewährleisten. Die gewählte Schriftart ist **Oxanium** (Google Fonts) und wird im Spiel für Überschriften in **BOLD**, und für Unterüberschriften sowie die Labels in **MEDIUM** verwendet.

WAVE: 3

TARGETS: 9/13

SCORE: 350

Abbildung 3.4: Informationsübersicht einer aktiven Runde (Wellen, Gegneranzahl, Punkte). Screenshot des Prototyps „Not So Friendly Neighbours“, Stand 13.12.2025



Abbildung 3.5: Informationsübersicht einer aktiven Runde (Wellen, Gegneranzahl, Punkte). Screenshot des Prototyps „Not So Friendly Neighbours“, Stand 13.12.2025

03 Spielidee > 03.3 Gestaltungselemente des Spiels > Steuerung und Story

Steuerung

In diesem Spiel soll auf klassische Eingabemethoden wie einem Controller, einer Maus oder Tastatur verzichtet werden. Der Spieler nutzt stattdessen seinen Körper für die Bewegung des Charakters und seine Hände für die direkte Interaktion mit den Spieleinhalten. Bei der Entwicklung der Gestensteuerung werden die in Kapitel 02.2 dargestellten Prinzipien beachtet.

Entwicklung der Story

Aufgrund der Neuinterpretation von Gut und Böse, werden die Gegner in diesem Spiel nicht als Raumschifflotte der Erde dargestellt, sondern als Weltraummüll mit unterschiedlichen Bedrohungsstufen. „Die Idee, Müll im Weltraum oder direkt in der Sonne zu entsorgen, ist ein wiederkehrendes Internetphänomen, das insbesondere in den sozialen Medien diskutiert wird (vgl. U.S. Department of Energy 2023).“ In der Welt des Protagonisten wird dieses Gedankenspiel zur hypothetischen Realität: Menschengemachter Abfall wird ins All ausgelagert und driftet als Müllwolke durch den Orbit, die die Heimat des Protagonisten bedroht und zum Konflikt des Spiels wird. Das Design der Gegner richtet sich dabei nach teils fiktiven, aber auch ganz realen Umständen und Objekten:

• Realität

Durch Fast Fashion, Verpackungsmaterialien die schwer bis gar nicht recycelt werden können und kurzlebige Elektronik, entstehen riesige Müllberge auf unserem Planeten. Darüber hinaus schätzt die Europäische Weltraumorganisation ESA (Stand 21. Oktober 2025) das ca. 1,2 Millionen Trümmerteile der Größe >1cm-10cm, sowie ca. 140 Millionen Objekte der Größenklasse >1mm-1cm als Weltraumschrott unsere Erde umkreisen. Darunter befinden sich unter anderem Raketenstufen und andere Trümmerteile (vgl. ESA 2025).

• Fiktion

Die Episode 8 „A Big Piece of Garbage“ der ersten Staffel von Futurama behandelt eine nahende Bedrohung des Planeten Erde. Bei der Bedrohung handelt es sich um eine gigantischen Müllansammlung, die in den 2000er Jahren mit einer Rakete ins Weltall geschossen wurde, um mit der wachsenden Müllentwicklung des Planeten fertig zu werden. Tausend Jahre später kehrt dieser Müllberg nun zurück und stellt eine ernsthafte Bedrohung dar (vgl. Popov Neven 2010).

03 Spielidee > 03.3 Gestaltungselemente des Spiels > Gegnerdesign



Gegnerdesign

Das Gegnerdesign ist angelehnt an die zwei zuvor beschriebenen Kategorien: Irdischer Müll und Weltraumschrott. Viele dieser Gegenstände besitzen wir selbst oder sind uns durch unser Umfeld oder Trends bekannt. Innerhalb der Kategorie Weltraumschrott wird anhand der Bedrohungsstufe weiter unterschieden.

Der allgemeine Müll mit der niedrigsten Bedrohung, kommt im Spiel als am häufigsten vor und wird über Alltagsgegenstände (Abbildung 3.6) repräsentiert. Gegner mit höherer Bedrohungsstufe werden durch Objekte wie defekte Satelliten, Asteroiden, alten Raketen oder Atommüllbälle, dargestellt (Abbildung 3.7)

03 Spielidee > 03.3 Gestaltungselemente des Spiels > Gegnerdesign



Shampoo Flasche



Asteroid



Satellit



Rakete



Container



Atommüll



04

ANFORDERUNGEN

04 Spielanforderungen

Auf Basis des zuvor dargestellten Konzepts und der Analyse bestehender Produkte und Lösungsansätze aus Kapitel 3, werden im folgenden Abschnitt Anforderungen an das eigene webbasierte und gestengesteuerte Videospiel abgeleitet. Diese gliedern sich in Anforderungen an das Spielprinzip, die User Experience und die technischen Anforderungen.

04.1 Spielprinzip

04.2 Interaktion und User Experience

04.3 Technische Anforderungen

04 Spielanforderungen > 04.1 Spielprinzip

Spielrunde

Spielerinnen und Spieler müssen in aufeinanderfolgenden Wellen angreifende Gegner eliminieren. Die Runde endet, wenn der Spieler kampfunfähig ist. Das Ziel ist es so lange wie möglich zu überleben und Punkte für einen Platz auf der Highscore Liste zu sammeln.

Spielzeit

Eine Runde lässt sich in wenigen bis zehn Minuten spielen. Die Dauer ist abhängig vom Geschick des Spielers. Zwischen den Wellen gibt es eine Übergangszeit von wenigen Sekunden, welche zum einen die Wellen voneinander abgrenzen und zum anderen dem Spieler eine kurze Ruhepause geben, um sich auf die kommende Welle einzustellen.

Gegnergestaltung

Es existieren mehrere Gegnertypen mit klar unterscheidbaren Bewegungen, Angriffsarten und Bedrohungsstufen geben. Diese Gegner unterscheiden sich im Aussehen, Anzahl der Lebenspunkte, Geschwindigkeit und Angriffsmuster. Der Spieler lernt Gegner zu erkennen und zu priorisieren.

Schwierigkeitssteigerung

Mit fortschreitenden Wellen steigt die Schwierigkeit weiter an und wird durch wachsende Gegneranzahl, erhöhte Geschwindigkeit, unterschiedlichen Gegnertypen und weiteren Mechaniken beeinflusst.

Benutzeroberfläche

Während der Spielrunde soll der Spieler einen Überblick über aktuelle Werte erhalten. Sie sollen anzeigen, wo er sich gerade befindet, welche Aufgabe in dem Moment ansteht und wie viele Punkte er bereits hat. Dies sollte so gestaltet sein, dass der Spieler sie schnell und ablenkungsfrei lesen kann.

Game Over und Highscore

Endet eine Runde, soll eine Auswertung des Spiels angezeigt werden, die die erreichten Punkte und den Fortschritt sichtbar macht. Es wird eine Highscore Liste bereitgestellt, auf die sich Spieler mit ihren Ergebnissen der Runde eintragen können.

Interaktion und Steuerung

Der spielbare Charakter und die Aktionen sollen vollständig berührungslos per Handgesten steuerbar sein. Die Handgesten werden so gestaltet, dass sie natürlich und intuitiv wirken und klar voneinander unterscheidbar sind.

Menüsteuerung

Die Navigation in den Spielemenüs soll ebenfalls berührungslos erfolgen und sich von den Gesten in der Spielerunde unterscheiden.

04 Spielanforderungen > 04.2 Interaktion und User Experience

Spieleinformation und Lesbarkeit

Wichtige Informationen werden so gestaltet, dass Spielerinnen diese schnell erfassen können. Schriften, Farben und Kontraste ermöglichen eine gute Lesbarkeit, vor allem in den Spielrunden.

Visuelles Feedback

Der Nutzer erhält für jede Interaktion in der Spielrunde und in den Menüs visuelle Rückmeldungen. Dieses Feedback ist so gestaltet, dass es das Verständnis des Spielers unterstützt und die Spielerfahrung intensiviert.

Gesten und Lernbarkeit

Die Umsetzung der Gestensteuerung soll für den Spieler leicht verständlich und umsetzbar sein. Die Gesten werden so definiert, dass sie natürlicheren und einfachen Bewegungen entsprechen, ohne großen körperlichen Einsatz zu fordern.

04 Spielanforderungen > 04.3 Technische Anforderungen

Performance

Für das Projekt wird eine Bildwiederholfrequenz von mindestens 30 Bildern pro Sekunde angestrebt, um eine flüssige Darstellung zu gewährleisten. Für eine direkte Reaktion soll die Latenz zwischen Eingabe und Ausführung minimal gehalten werden.

Plattform

Das Spiel soll mit modernen Browsern auf Desktopgeräten kompatibel sein. Für das Spiel wird lediglich eine einfache Webcam und ein internetfähiger Computer benötigt.

Fehlerhandhabung

Während der Umsetzung auftretende Fehler werden so gelöst, dass der Spieler keine Konsequenzen für seine Runde trägt. Insbesondere im Hinblick auf das laufende Spiel sollen Fehler rechtzeitig abgefangen werden, die das Spieelerlebnis beeinflussen oder zum Abbruch bringen können.

Erweiterbarkeit

Das Projekt ist so umgesetzt, dass das Spiel leicht um weitere Mechaniken, Gegner und Gesten ergänzt werden kann und Anpassungen an der Nutzeroberfläche und dem Spielprinzip leicht durchgeführt werden können.

05

TECHNOLOGIEAUSWAHL

05 Technologieauswahl

In diesem Kapitel wird aus den zuvor ermittelten Anforderungen an das Spiel eine Liste mit Kriterien für die Spiele Engine und das Trackingsystem erstellt.

05.1 Auswahl der Game Engine

05.2 Auswahl des Trackingsystems

05 Technologieauswahl > 05.1 Auswahl der Game Engine

Für die Umsetzung werden einige Kriterien an Spiele-Entwicklungsumgebungen (GEK) gestellt und anschließend zum Vergleich herangezogen. Die Auswahl beschränkt sich hierbei auf Technologien, die für webbasierte 3D-Spiele konzipiert sind.

Game Engine Kriterien (GEK)

- GEK-1:** Die Engine soll für den Einsatz in browsern konzipiert sein.
- GEK-2:** Sie unterstützt Darstellungen in 3D über die eine Spielemechanik realisiert werden kann.
- GEK-3:** Der Editor unterstützt die Darstellung von HUD Elementen, Overlays und Benutzer-oberflächen in oder über der Szene.
- GEK-4:** Die Engine besitzt eine gute Einbindungsmöglichkeit um Webelemente getrennt von der Engine umsetzen zu können.
- GEK-5:** Es muss eine Bildwechselfrequenz von mindestens 30 FPS (Frames Per Second) gewährleisten.
- GEK-6:** Das Programm ist OpenSource, kostenfrei und hat einen vollständigen Leistungsumfang.
- GEK-7:** Die Game Engine verfügt über visuelle Hilfsmittel, wie einen Editor und weitere unterstützende Werkzeuge bei der Entwicklung.

05 Technologieauswahl > 05.1 Auswahl der Game Engine

Tabelle

KRITERIUM	Babylon.js + Editor	Three.js	PlayCanvas	Unity WebGL	Godot Web
GEK-1	✓✓	✓✓	✓✓	●	●
GEK-2	✓✓	● Fokus auf Rendering	✓✓	✓✓	✓
GEK-3	✓✓ (Babylon GUI)	●	✓	✓	✓
GEK-4	✓✓	✓✓	✓	✓	●
GEK-5	✓✓	✓✓	✓✓	●	●
GEK-6	✓✓ (OpenSource)	✓✓ (OpenSource)	●	●	✓✓ (OpenSource)
GEK-7	✓✓ (Babylon Editor, + weitere Babylon Tools)	●	✓ (Web Editor)	✓✓ (Unity Editor)	✓

(Bewertungsskala: ✓✓ = erfüllt, ✓ = möglich, ● = eingeschränkt, ● = nichtzutreffend)

Vergleich

Aus der Tabelle ergibt sich, dass sich grundsätzlich mehrere Engines dafür geeignet sind ein 3D-Spiel im Browser zu realisieren. Im Vergleich zu Babylon.js, welches explizit für webbasierte Spiele konzipiert ist, bieten Unity WebGL und Godot lediglich einen Export für den Webkontext an. Daraus resultiert ein eventueller Mehraufwand bei der Optimierung für webbasierte Inhalte. Three.js setzt den Fokus auf das Rendering von Szenen und Objekten im Web, hat aber im Vergleich zu Babylon.js, keine „Out-of-the-Box“ Lösungen für Spielemechaniken. PlayCanvas erfüllt ebenfalls viele Kriterien, hat aber im Vergleich zum Babylon Editor einen erhöhten Mehraufwand bei der geplanten Umsetzung des Spiels. Zwar besteht auch mit PlayCanvas die Möglichkeit einer komfortablen Umsetzung in einem Editor, jedoch bedarf es eines kostenpflichtigen Abonnements um diesen zu nutzen.

Auswahl

Auf Grundlage des Vergleichs wird Babylon.js als Game Engine für die Umsetzung des Spieleprototyps gewählt. Ausschlaggebend ist, dass Babylon.js und der Babylon Editor explizit für die Entwicklung webbasierter 3D-Spielen konzipiert ist und die spielerelevanten Funktionen kostenlos und „Out-of-the-Box“ mitbringt.

05 Technologieauswahl > 05.2 Auswahl des Trackingsystems

Für die berührungslose Steuerung wird ein Trackingsystem benötigt, das Handgesten und Körperbewegung über eine einfache Webcam zuverlässig erkennt und diese verarbeiteten Daten in der gewählten Spiele Engine nutzbar macht. Dafür werden folgende fünf Trackingsystem Kriterien (TSK) aufgestellt:

Trackingsystem Kriterien (GEK)

- TSK-1:** Es wird lediglich eine handelsübliche Webcam als Eingabemethode benötigt. Zusätzliche Hardware darf nicht erforderlich sein.
- TSK-2:** Die Erkennung der Eingaben (Handgesten und Körperbewegung) und die Rückmeldung im Spiel erfolgt in Echtzeit.
- TSK-3:** Bietet die Möglichkeit die Eingabemethoden eigenständig, simpel und frei zu definieren.
- TSK-4:** Die Bilddaten werden ausschließlich lokal im Browser der Nutzer verarbeitet und ausgewertet.
- TSK-5:** Das System ist für webbasierte Anwendungen geeignet und leicht in die Game Engine zu integrieren.

05 Technologieauswahl > 05.2 Auswahl des Trackingsystems

Tabelle

KRITERIUM	MediaPipe Solutions	TensorFlow	Leap Motion	Kinect
TSK-1	✓✓	✓✓	●	●
TSK-2	✓✓	✓✓	✓✓	✓✓
TSK-3	✓✓	✓✓	●	✓
TSK-4	✓✓	✓✓	●	●
TSK-5	✓✓	✓	✓✓	●

(Bewertungsskala: ✓✓ = erfüllt, ✓ = möglich, ● = eingeschränkt, ● = nichtzutreffend)

Vergleich

Neben den kamerabasierten Trackingsystemen existieren weitere hardwarebasierte Alternativen wie Leap Motion oder Kinect. Über spezielle Sensoren bieten diese Systeme zwar eine genauere Erfassung der Eingaben, sind aber in erster Linie nicht für den Webkontext ausgelegt und benötigen neben der Hardware auch spezifische Entwicklungsbausteine oder sind teilweise Plattformgebunden. Diese Punkte stehen jedoch im Widerspruch zu den Kriterien, dass das Spiel möglichst kostenfrei entwickelt und ohne zusätzliche Hardware im Browser gespielt werden kann. Daher beschränken sich die Trackingsysteme, die die Kriterien erfüllen auf Googles MediaPipe Solutions und TensorFlow.js.

05 Technologieauswahl > 05.2 Auswahl des Trackingsystems

MediaPipe

MediaPipe ist ein Framework von Google welches Lösungsansätze für die Integration von künstlicher Intelligenz oder Maschine Learning in Projekten bereitstellt. Laut Google können diese Ansätze plattformübergreifend in bereits bestehende Anwendungen eingebunden und an eigene Anforderungen weiter angepasst werden. (vgl. Google AI Edge, MediaPipe Solutions guide, 2025).

Die zwei relevanten Lösungen im Kontext dieses Projekts sind die „Hand landmark detection“ für die Hände und die „Pose landmark detection“ für den Körper. Beide Lösungen ermitteln über maschinelles Lernen Knotenpunkte der Hände (mit 21 Landmarken pro Hand) und des Körpers (mit 33 Landmarken) und geben diese Daten in Bild- oder Weltkoordinaten aus. Diese Daten und Knotenpunkte können mittels TypeScript/JavaScript für eigene Funktionen in der Anwendung verwendet werden. (vgl. Google AI Edge, Hand landmarks detection guide, 2025; Google AI Edge, Pose landmarks detection guide, 2025).

MediaPipe bietet mit „Gesture recognition“ ein weiteres Modul zur Erkennung konkreter Handgesten an. Dieser Ansatz beschränkt sich jedoch auf nur wenige vordefinierte Handgesten. Eine Erweiterung der Gesten ist nur über entsprechendes Training der ML-Modelle realisierbar (vgl. Google AI Edge, Gesture recognition task guide, 2025)

TensorFlow

TensorFlow ist eine „End-to-End Plattform, die ein einfaches Bauen und Bereitstellen von Machine-Learning (ML) Modellen ermöglicht (vgl. TensorFlow, WhyTensorFlow o.D.)“. Mit TensorFlow.js stellen sie eine JavaScript-Bibliothek zur Verfügung, die die Entwicklung von ML-Modellen über den Browser ermöglicht. Neben dem Training eigener Modelle bieten sie auch eine Auswahl vortrainierter Tracking-Modelle für webbasierte Projekte an (vgl. TensorFlow, TensorFlow.js Models, o.D.). Laut der READ.me des GitHub-Repositories, wird von TensorFlow zur Erkennung der 21 Hand Landmarks zurzeit nur das Handtracking Model von MediaPipe selbst bereitgestellt (vgl. TensorFlow, tfjs-models-hand-pose-detection, o.D.).

Auswahl

Der direkte Unterschied von TensorFlow zu MediaPipe Solutions besteht also darin, dass der Nutzer frei wählen kann, ob er bestehende Modelle nutzen oder ein eigene erstellen will. Da das Erstellen und Trainieren von Modellen im Rahmen dieses Projekts einen höheren Aufwand darstellen, wird sich bewusst für die simplere Variante von MediaPipe entschieden.



06 UMSETZUNG DES PROTOTYPUS

06 Umsetzung des Prototyps

In diesem Kapitel wird die Vorgehensweise bei der Entwicklung des Spielprototyps beschrieben. Das Ziel war es, ein browserbasiertes 3D Spiel zu erstellen, dass mit einer einfachen Webcam berührungslos und ohne zusätzliche Hardware, Maus oder Tastatur, spielbar ist. Der Spieler nutzt Hände und Körper, um mit den Spieleinhalten zu interagieren.

06.1 Projekteinrichtung/Projektstruktur

06.2 Szenenaufbau und visueller Look

06.3 Steuerung

06.3.1 Gesten

06.4 Game Loop

06.4.1 EndlessWaveManager.ts

06.5 Schwierigkeitssteigerung

06 Umsetzung des Prototyps > 06.1 Projekteinrichtung/Projektstruktur

Für die grundlegende Einrichtung des Projektes wurde der Babylon.js Editor verwendet, der beim Anlegen einer neuen Szene eine geeignete Projektstruktur und die benötigten Grundmechanismen für ein lauffähiges Spiel im Browser bereitstellt. Der Editor bietet die Möglichkeit die Szene über ein lokalen node.js Server direkt im Browser zu testen und Anpassungen an der Szene direkt mitzuverfolgen. Über einen integrierten Filebrowser besteht zudem die Möglichkeit Code-Skripte für die Spielemechanismen und Funktionen direkt an Szenenobjekte zu hängen. Im Inspector-Tab des Editors wurden diese Parameter dann über Eingabefelder weiter angepasst, ohne die Werte im Code ändern zu müssen.

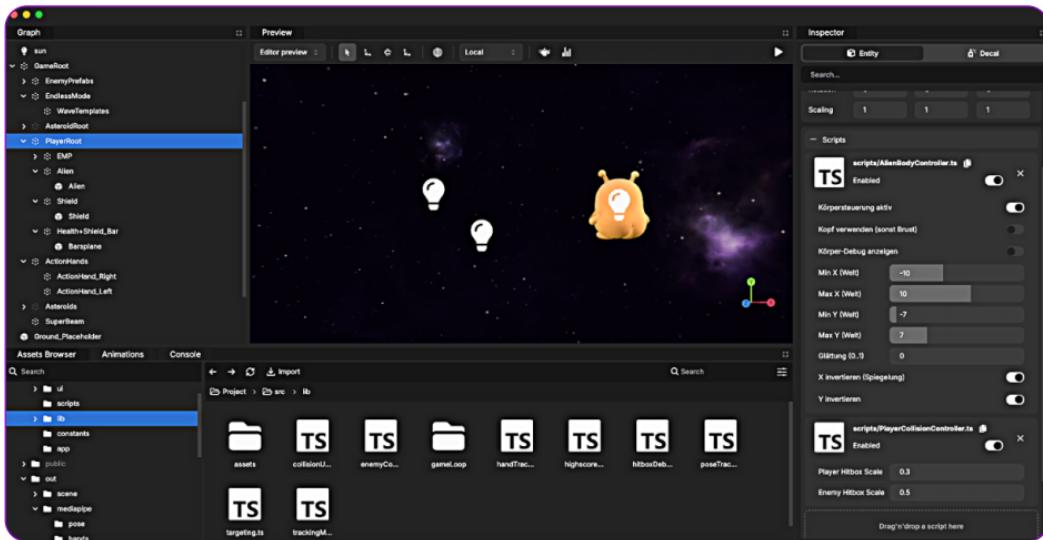


Abbildung 6.1 Babylon Editor Oberfläche mit dem geöffneten Projekt. .

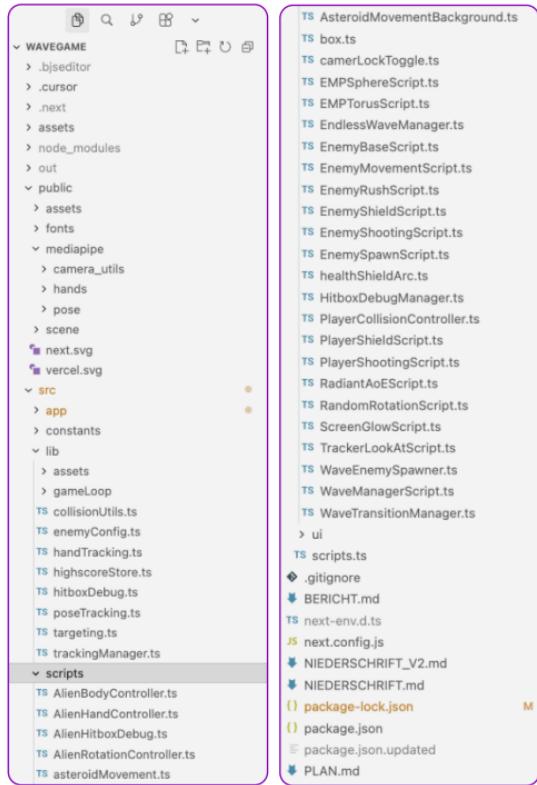


Abbildung 6.2 Dateistuktur des Projekts in Cursor.

06 Umsetzung des Prototyps > 06.1 Projekteinrichtung/Projektstruktur

Auf diese Weise wurden relevante Kernmechanismen wie das Gegnerverhalten oder das Wellensystem an entsprechende Objekte in der Szene gekoppelt. Diese und weitere Module wie die Trackingverarbeitung, die Berechnung des Scores und Menüs, werden zwar über ein zentrales Skript in das Spiel eingebunden, funktionieren aber unabhängig vom Geschehen in der Spielszene. Die programmatische Implementierung der Skripte und des User Interfaces erfolgte in der Entwicklungsumgebung Cursor und dem integrierten Coding-Agenten.

Der Editor wurde genutzt, um die Spielszene und die visuellen Objekte zu erstellen und entsprechend in der Welt zu platzieren. Mit Cursor und dem integrierten Coding-Agenten wurden anschließend Skripte für Spielelogik erstellt, an die Szenenobjekte gehangen, und im Browser getestet. Dadurch konnten die Funktionen direkt geprüft und bei Unstimmigkeiten oder Fehlern direkt darauf reagiert werden. Während der gesamten Umsetzung wurde darauf geachtet, dass mit jeder Anpassung das Spiel weiterhin nutzbar blieb. Somit entstanden im gesamten Prozess durch stetiges Erweitern und Anpassen der funktionalen Mechanismen und visuellen Elementen der Szene, alle zentralen Bausteine (Spielszene, Gameplay, Tracking und Steuerung, Menüs und Userinterfaces) des Spiels. In den folgenden Unterkapiteln wird die Umsetzung der einzelnen Bausteine näher beschrieben und erläutert.

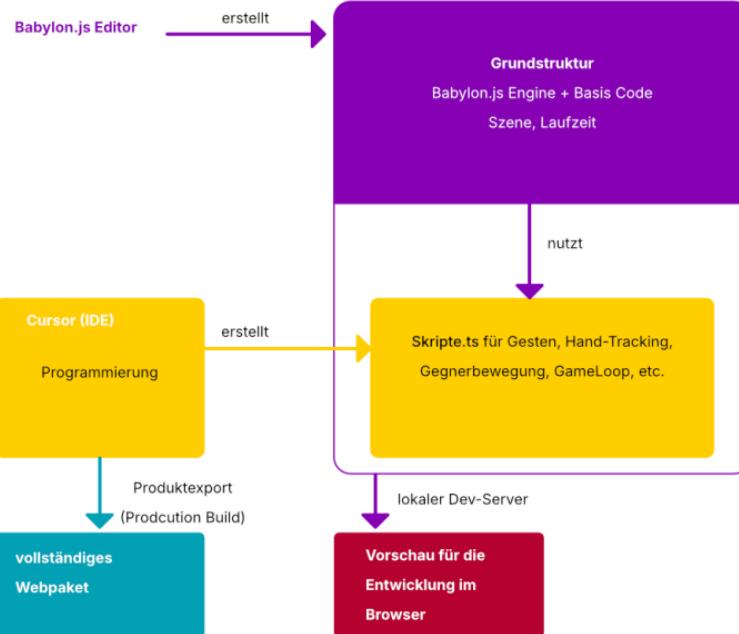


Abbildung 6.3 Projektstruktur und Wechselwirkung der Werkzeuge

06 Umsetzung des Prototyps > 06.2 Szeneaufbau und visueller Look

Parallel zu der Projektstruktur erstellt der Babylon Editor eine Beispielszene mit der relevanten Struktur, die weiter angepasst und durch die benötigten 3D-Modelle ergänzt wurde.

Beleuchtung und Lesbarkeit

Für die Belichtung kommen mehrere Lichtquellen in der Szene zum Einsatz. Diese verleihen den Objekten zum einen durch die Schattenbildung eine höhere Plastizität und Tiefe. Zum anderen sorgt das Licht für eine höhere Sichtbarkeit, in dem es die 3D-Modelle stärker vom dunklen Hintergrund hervorhebt. Diese Wirkung wird zudem durch den eingesetzte Bloom-Effekt verstärkt, der vorangestellt als Stilmittel eingesetzt wurde, da er den Objekten ein sanftes Glühen verleiht.

Assets und Organisation

Die Gegner und der spielbare Charakter wurden als 3D-Modelle im .glb-Format in den Assets-Ordner des Projekts eingefügt und anschließend in der Szene platziert. Über den Scene-Graph wurden die Objekte dann den logischen Bereichen des Spiels in die jeweiligen Node, z.B. Enemy-Prefabs für die Gegnermodelle und Playeroot für den Spielcharakter, zugeordnet. Durch diese Organisation konnten im späteren Verlauf der Entwicklung leichter weitere Modelle hinzugefügt oder wiedergefunden werden. Ferne vereinfachte diese das Bearbeiten oder Austauschen von Objekten. Anpassungen der Größenverhältnisse und Texturen wurden über den Inspector Tab an dem jeweiligen Modell durchgeführt.

Kamera

Angelehnt an klassische Arcadespiele wurde für diesen Prototypen eine feste Position und Ausrichtung der Kamera in Third-Person Perspektive festgelegt. Bildausschnitt, Blickrichtung und Tiefe wurden so gewählt, dass das Geschehen auf dem Spielfeld überblickt werden kann. Der Spielhintergrund im Weltraumstil ist eine Mesh-Box, die den gesamten dreidimensionalem Raum umschließt. Als Material dient hierfür eine Bildtextur, die mit dem Mesh verknüpft ist. Da das Spiel in einem Weltraumszenario angesiedelt ist, wurden keine sichtbaren Spielraumbegrenzungen in die Szene eingefügt. Der Spielraum wird durch die Kameraposition (negativer Z-Raum) und einem fest definierten Spawn-Bereich (positiver Z-Raum) der Gegner in der Szene definiert. Der steuerbare Charakter befindet sich im Weltursprung bei Z = 0 und kann nur entlang der horizontalen und vertikalen Achse bewegt werden um den Gegnern auszuweichen.

06 Umsetzung des Prototyps > 06.3 Steuerung

Die Steuerung des Spiels basiert auf den Landmark-Daten die durch das Hand- und Körpertracking von MediaPipe aus dem Videostream ermittelt werden. Anhand dieser Daten wurden zunächst klare Zustände und Gesten definiert, die in eindeutige Aktionen im Spiel übersetzt wurden. Beide Tracking Pipelines ermitteln pro Frame die Landmarks der Hände (21LM) und des Körpers (33 LM), und liefern Positions-werte in Form von Koordinaten zurück. Aus diesen Koordinaten konnten über Berechnungen weitere Informationen zur Distanz, Verweildauer und Positionsänderungen ermittelt werden. Anhand dieser Parameter wurden die Gesten mit festgelegten Werten und Bedingungen definiert.

Landmarks: Hand Tracking



Abbildung 6.4: Visualisierung der Hand Landmarks (MediaPipe Hand Landmarker).

Quelle: Google AI Edge (MediaPipe Solutions),

https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker (abgerufen am 16.12.2025)

Landmarks: Pose Tracking

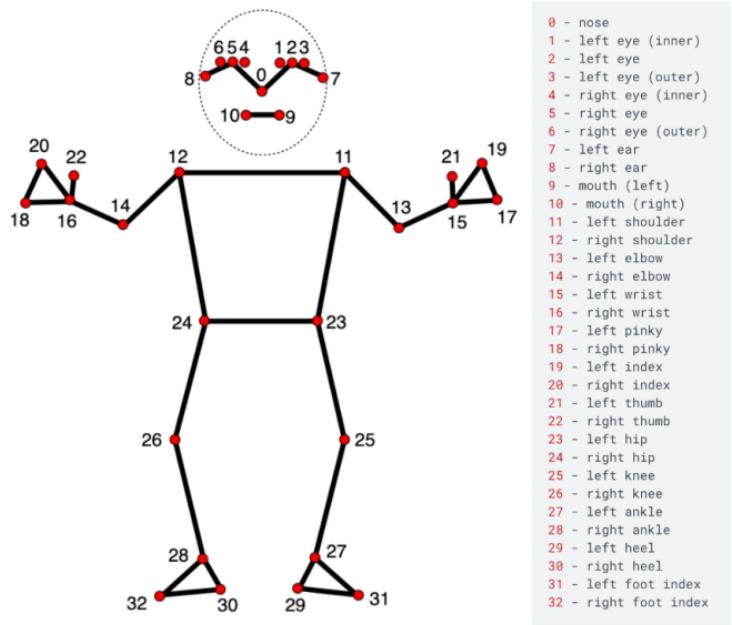


Abbildung 6.5: Visualisierung der Körper Landmarks (MediaPipe Pose Landmarker model).

Quelle: Google AI Edge (MediaPipe Solutions),

https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker (abgerufen am 16.12.2025)

06 Umsetzung des Prototyps > 06.3 Steuerung > 06.3.1 Gesten

Im folgenden Kapitel wird die Entwicklung der Gesten zur Steuerung der Webanwendung beschrieben. Dabei wird zwischen In-Game Gesten und den Gesten für die Menüsteuerung unterschieden, da diese nur in den jeweiligen Zuständen (Spielrunde oder Menü) des Spiels eingesetzt werden können.

In-Game Gesten

Die In-Game Gesten wurden speziell für den Einsatz in einer laufenden Runde erstellt und beziehen sich auf die Interaktionsmöglichkeiten mit den Objekten innerhalb der Spielwelt.



Abbildung 6.6: Spielfeld mit dem Charakter, der Gegnerwelle, HUD im Vordergrund.

Gesten der Menüsteuerung

Die Menüsteuerungsgesten sind speziell für die Navigation in den Menüs und Overlays konzipiert worden. Sie sind den Lösungen anderer Entwicklungen berührungsloser Interaktionsformen, wie der Apple Vision Pro oder der Oculus Quest, nachempfunden.

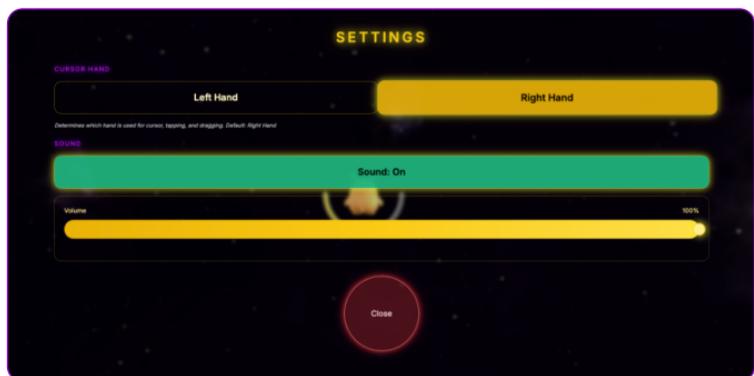


Abbildung 6.7: Ansicht des Einstellungsmenüs.

06 Umsetzung des Prototyps > 06.3 Steuerung > 06.3.1 Gesten

ZIELEN UND SCHIEßen

Die „Zielen“-Geste wurde über eine simple Nutzung der Positionskoordinaten der Zeigefingerspitzen (Landmark 8) beider Hände realisiert. Im Spiel symbolisieren Fadenkreuze die Position des getrackten Landmarks der Zeigefingerspitzen.

Für die Schussgeste werden die Daumenspitze (Landmark 4) und der große Knöchel des Mittelfingers (Landmark 10) auf ihre Distanz zueinander geprüft. Fällt die Distanz auf einen bestimmten Wert, wird dies als Auslöser erkannt und im Spiel in eine konkrete Aktion (Schuss) übersetzt.

Die Landmarks wurden dabei bewusst gewählt, weil sie zusammen mit der Zielgeste an die bekannte Fingerpistole erinnern. Auf Dauer ist die resultierende Geste angenehmer durchzuführen, als den Daumen auf die Landmark 5 oder 6 anzulegen. Zusätzlich wurde eine weitere Bedingung an die Schussgeste definiert, damit die Geste bewusst durchgeführt werden muss: Hält der Spieler aus Bequemlichkeit die Schussgeste (8 auf 10) dauerhaft, wird die Schussfrequenz auf ein Projektil pro Sekunde gedrosselt. Aktives Durchführen der Geste wird mit einer höheren Schussfrequenz belohnt.

```
124
125  // Schussgeste-Distanz (einstellbar)
126  let SHOOT_DISTANCE = 0.04; // Distanz zwischen Daumenspitze und Mittelfinger-Knöchel (Standard: 0.04)
127
```

Abbildung 6.8: Codeauszug. Distanz der Landmarks für den Auslöser der Schießgeste. Stand 13.12.2025

Durchführung

Fadenkreuz

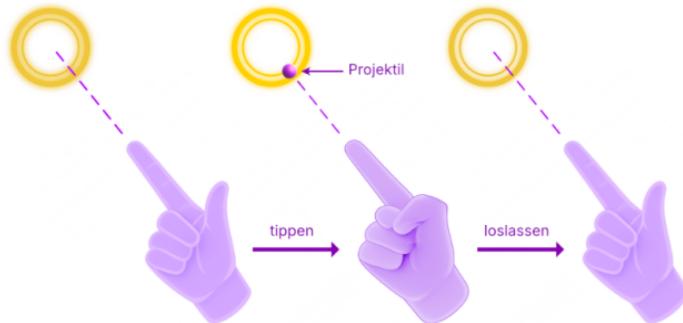


Abbildung 6.9: Illustration der Handgestenabfolge zum Zielen und Schießen. Hände generiert mit ChatGPT

06 Umsetzung des Prototyps > 06.3 Steuerung > 06.3.1 Gesten

SCHILDGESTE

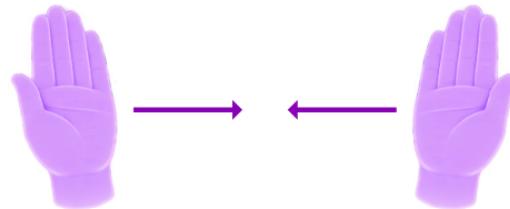
Für die Schildgeste müssen die Handseiten des kleinen Fingers zusammengelegt werden. Dabei wird die Distanz zwischen den Landmarks 17-20 beider Hände paarweise verglichen (17-17 & 18-18 & 19-19 & 20-20). Um ein unbeabsichtigtes Auslösen des Schildes auszuschließen, wurde eine Bedingung gesetzt um diese Geste als legitime Eingabe zu erkennen. Daher muss die Schildgeste für zwei Sekunden gehalten werden, bevor die Eingabe akzeptiert und der Schild im Spiel aktiviert wird.

```
101
102 const SHIELD_GESTURE_DISTANCE = 0.08; // Maximaler Abstand zwischen Landmarks (erhöht von 0.05 für größere Toleranz)
103 const SHIELD_GESTURE_HOLD_TIME = 2000; // ms: Geste muss 2 Sekunden gehalten werden
104 const SHIELD_COOLDOWN = 10000; // ms: 10 Sekunden Cooldown nach Aktivierung
105 const SHIELD_GRACE_PERIOD = 300; // 300ms Toleranz für kurze Ausreißer
106
```

Abbildung 6.10: Codeauszug. Definition der Mindestdistanz, Haltedauer, Wartezeit und Toleranzwerte für die Schild-Geste. Stand 13.12.2025

Durchführung

1.



2.



2 Sek. halten

Abbildung 6.11: Illustrierte Handgeste zum aktivieren des Schildes. Hände generiert mit ChatGPT

06 Umsetzung des Prototyps > 06.3 Steuerung > 06.3.1 Gesten

PAUSE GESTE

Für den Aufruf des Pausenmenüs wurde eine Handposen-Kombination gewählt, bei der aus beide Händen der Buchstabe T geformt werden muss. Die relevanten Landmarks für einer Erkennung dieser Geste sind Landmarks 9 und 12. Um ein unbeabsichtigtes Auslösen dieser Geste auszuschließen, wurden zwei Bedingungen eingeführt:

1. Die Geste wird nur dann vom System interpretiert, wenn sich die Landmark 9 der einen Hand in der Nähe der Landmark 12 auf der anderen Hand befindet. Dies gilt für beide Hände gleichermaßen. Dadurch wird verhindert, dass das Pausenmenü mit nur einer Hand aufgerufen werden kann.
2. Die Geste muss bewusst für mindestens 1,5 Sekunden gehalten werden, damit sie nicht unbeabsichtigt ausgelöst wird, falls die Hände sich während des Spielens kurzzeitig überlappen.

Durchführung



Abbildung 6.12: Pausengeste zum Pausieren des Spiels. Erstellt mit ChatGPT (Bildgenerierung)

06 Umsetzung des Prototyps > 06.3 Steuerung > 06.3.1 Gesten

CHARAKTER NAVIGATION

Wie in Abbildung 6.5 zu sehen ist, existiert im Posetracking kein fester Knotenpunkt für die Mitte der Brust. Damit der Charakter zentriert gesetzt und gesteuert werden kann, wird die Positionscoordinate (eigene Landmark) über eine Berechnung des Mittelpunkts zwischen den Landmarks 11 und 12 bestimmt. Die ermittelten Koordinaten werden anschließend an das AlienBodyController-Skript gesendet und dort in die entsprechenden Weltkoordinaten des Spielfeldes übersetzt.

```
57 const nose = landmarks[0];
58 const leftShoulder = landmarks[11];
59 const rightShoulder = landmarks[12];
60
61 const head = nose
62 ? {
63   x: nose.x,
64   y: nose.y,
65   z: nose.z,
66   visibility: nose.visibility,
67 }
68 : undefined;
69
70 const chest =
71 leftShoulder && rightShoulder
72 ? {
73   x: (leftShoulder.x + rightShoulder.x) / 2,
74   y: (leftShoulder.y + rightShoulder.y) / 2,
75   z: (leftShoulder.z + rightShoulder.z) / 2,
76   visibility: Math.min(leftShoulder.visibility ?? 1, rightShoulder.visibility ?? 1),
77 }
78 : undefined;
```

Durchführung

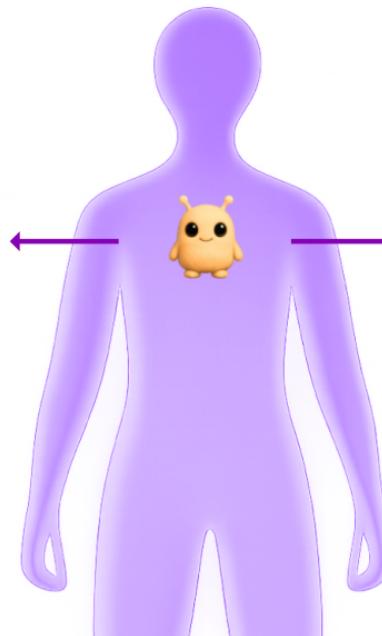


Abbildung 6.13: Codeauszug. Berechnung der Position zwischen den zwei Schulterlandmarks 11 und 12.

Stand 13.12.2025

Abbildung 6.14: Position des Charakters am Körper. Spieler muss seinen Körper bewegen um die Figur im Spiel zu bewegen. Generiert mit ChatGPT

06 Umsetzung des Prototyps > 06.3 Steuerung > 06.3.1 Gesten

HANDCURSOR

Für die visuelle Repräsentation eines Cursors wurde ein Kreiselement an die Daumenspitze (Landmark 4) gehängt, mit dem die Nutzer über die Oberfläche navigieren können. Ähnlich wie bei einer Steuerung mit der Maus, kann der Nutzer hier über Buttons in den Menüs hovern und mit einer Tap-Geste seine Auswahl bestätigen.

TAP

Um in den Menüs die Buttonauswahl zu bestätigen, müssen Zeigefingerspitze (Landmark 8) und Daumen (Landmark 4) kurzzeitig zusammengeführt werden. Dafür wurde ein Distanzwert zwischen den Landmarks definiert, welcher bei einer Unterschreitung die damit verknüpfte „Klick“-Aktion durchführt. Um sicher zu stellen, dass Spieler sich nicht „tippen“, wurde diese Aktion an den Handcursor gekoppelt, mit der Bedingung, dass ein „Klick“ nur dann ausgeführt wird, wenn der Handcursor auf einem Button liegt.

HOLD

Werden Daumen und Zeigefinger über einen längeren Zeitraum gehalten, wird dies als Hold-Geste interpretiert. Dafür müssen beide Landmarks den Distanzwert der Tap-Geste unterschreiten und dauerhaft im Schaltbereich gehalten werden. Über diese Aktion können Schieberegler (falls vorhanden) in den Menüs betätigt werden.

Durchführung

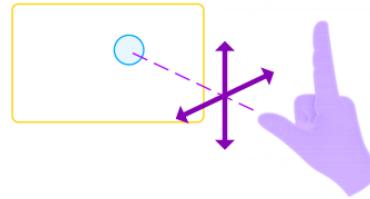


Abbildung 6.15: Dargestelltes Handcursorelement an der Daumenspitze (Landmark 4), Hand generiert mit ChatGPT. Stand 20.12.2025

Durchführung

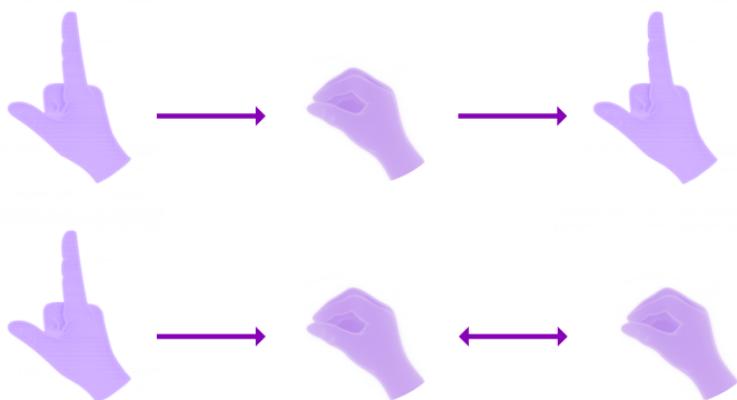


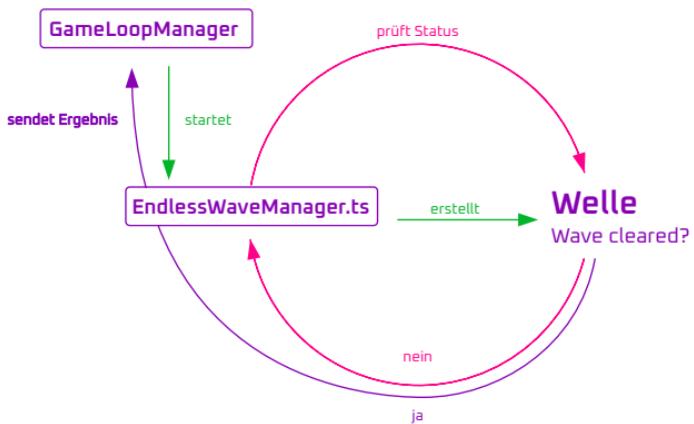
Abbildung 6.16: Darstellung des Bewegungsablaufs für das Durchführen der Tap und Hold-Geste. Hände generiert mit ChatGPT, Stand 20.12.2025

06 Umsetzung des Prototyps > 06.4 Game Loop

Eine Spielerunde besteht aus aufeinanderfolgenden Gegnerwellen, die der Spieler möglichst lange überleben muss. In jeder Welle sammelt der Spieler für das Vernichten von Gegnern Punkte für den Highscore. Das Spiel endet, wenn der Lebensbalken auf null fällt. Diese Spielschleife wird von dem GameLoopManager (GLM) verwaltet und ist Ausgangspunkt für den Spielstart und übermittelt bei jeder abgeschlossenen Welle den Status (Wave Cleared) an weitere Systeme des Spiels.

EndlessWaveManager.ts

Der EndlessWaveManager (EWM) ist zuständig für das eigentliche Wellenprinzip des Spiels. Über den EWM werden die wichtigsten Mechanismen, wie die Wellenbildung, Bewegungsrichtung, Gegnerinstanzen und die Phasen zwischen den Wellen des Spiels definiert, verwaltet und geprüft.



Gegnerinstanzen und Wellenbildung

Der EWM erstellt die benötigten Instanzen der Gegnermodelle anhand vordefinierter Richtlinien, wie die Gegneranzahl oder den Gegnertypen-Mix, und bündelt diese in einer Welle. Das Skript prüft kontinuierlich wie viele Gegner noch auf dem Spielfeld sind und registriert alle Gegner auf dem Spielfeld. Sobald diese eliminiert wurden oder sie die globale Killzone ($Z=-20$) erreichen, übermittelt der EWM das Ergebnis (Wave Cleared) an den GLM und wartet auf dessen Freigabe für die nächsten Welle.

Abbildung 6.17: Erstellungs und Prüfungsschleife des EndlessWaveManager.ts. Stand 20.12.2025

06 Umsetzung des Prototyps > 06.4 Game Loop

Spawnbereich und Spawn Position

Über den EWM wird ein dreidimensionaler Raum für die Spawnpositionen der Gegnerinstanzen mithilfe des Editors definiert. Die exakte Position jeder Gegnerinstanz wird zufällig gesetzt, damit die Wellen keinen vorgefertigten Eindruck erwecken.



Abbildung 6.18: Parametereinstellungen des EndlessWaveManagers als angehängtes Skript am Loop-Node im Editor. Eigener Screenshot.

Stand 13.12.2025

Bewegungsrichtung und Ziel

Ebenso wird im EWM die Bewegungsrichtung und Rotation und der Zielbereich für die Gegnerobjekte definiert. In mehreren Zwischentests im Editor wurden die optimalen Werte für die Flugbahn und die Basisgeschwindigkeit ermittelt. Grundsätzlich bewegen sich alle Gegner in Richtung des negativen Z-Bereichs (zur Position der Kamera) in der Szene. Wie bei der Gegnerposition wird auch die Basis-Fluggeschwindigkeit und die Rotation der Modelle zufällig gesetzt.

06 Umsetzung des Prototyps > 06.5 Schwierigkeitssteigerung

Für eine anspruchsvollere Wellenbildung, steigender Herausforderung und um neue Reize beim Spielen zu setzen, wurde eine mehrschichtige Schwierigkeitssteigerung entwickelt. Die Komplexität dieses System besteht darin, dass im Verlauf einer Spielrunde sich nicht nur die Anzahl der Gegner erhöht, sondern auch weitere Gegnertypen mit unterschiedlichen Angriffsarten und Schildvariante eingeführt werden. Die Gegnertypen besitzen zudem unterschiedliche Statuswerte (Lebenspunkte, Angriffsschaden, Geschwindigkeits-multiplikatoren und Schilde) die die Bedrohungsstufe ausmachen.

Gegnertypen und weitere Statuswerte

Typ (ID)	Rolle/Kernidee	Hits	Koll.-Schaden	Punkte	Schildfähig	Kann entkommen
Grunt (1)	Standard-Gegner, „Grundrauschen“	1	10	10	x	✓
Charger (2)	Rusher, Ansturm auf den Spieler	2	20	15	✓	✓
Hunter (3)	Präziser Rusher, verfolgt aktiv	3	30	25	✓	x
Gunner (4)	Schütze, Projektilangriffe	3	15	15	✓	✓
Tracker (5)	Zielt/ Schüsse mit „homing“	3	15	20	✓	✓
Radiant (6)	Flächenschaden	4	20	30	x	x
Container (7)	Spawnt Gegner beim Tod	4	25	25	x	✓

Tabelle 6.1: Tabellarische Übersicht der Gegnertypen mit ihrer Rolle, Lebenspunkten (Hits), Kollisionsschaden, Punkte für das Eliminieren, Schildfähigkeit und ob der Gegner entkommen kann. Stand 13.12.2025

06 Umsetzung des Prototyps > 06.5 Schwierigkeitssteigerung

Gegnermix und Verteilung

Im Verlauf einer Spielrunde werden ab vorab festgelegten Wellen neue Gegnertypen eingeführt. Diese Wellen werden im Code als Einführungswellen bezeichnet und dienen dazu, den Spieler in Abständen vor neue Herausforderungen zu stellen. Die jeweilige Einführungswelle eines Gegners richtet sich an die Bedrohungsstufe des Typs.

• Zuweisung im Code

```
private getEnemyWeightCurves(): EnemyWeightCurve[] {  
    return [  
        { typeID: 1, waveUnlock: 1, weightWave1: this.gruntWeightWave1,  
        weightWave50: this.gruntWeightWave50 }, // Grunt: Welle 1  
        { typeID: 2, waveUnlock: 4, weightWave1: this.chargerWeightWave1,  
        weightWave50: this.chargerWeightWave50 }, // Charger: Welle 4  
        { typeID: 4, waveUnlock: 8, weightWave1: this.gunnerWeightWave1,  
        weightWave50: this.gunnerWeightWave50 }, // Gunner: Welle 8  
        { typeID: 5, waveUnlock: 15, weightWave1: this.trackerWeightWave1,  
        weightWave50: this.trackerWeightWave50 }, // Tracker: Welle 15  
        { typeID: 3, waveUnlock: 22, weightWave1: this.hunterWeightWave1,  
        weightWave50: this.hunterWeightWave50 }, // Hunter: Welle 22  
        { typeID: 6, waveUnlock: 28, weightWave1: this.radiantWeightWave1,  
        weightWave50: this.radiantWeightWave50 }, // Radiant: Welle 28  
        { typeID: 7, waveUnlock: 38, weightWave1: this.containerWeightWave1,  
        weightWave50: this.containerWeightWave50 }, // Container: Welle 38  
    ];  
}
```

Codeauszug 6.1: Programmatische Implementierung der Einführungswellen. Stand 13.12.2025

• Einführungswellen

Gegnertyp	Einführungswelle
Charger	4
Gunner	8
Tracker	15
Hunter	22
Radiant	28
Container	38

Tabelle 6.2: Tabellarische Übersicht der Gegnertypen mit ihrer Rolle, Lebenspunkten (Hits), Kollisionsschaden, Punkte für das Eliminieren, Schildfähigkeit und ob der Gegner entkommen kann. Stand 13.12.2025

06 Umsetzung des Prototyps > 06.4 Schwierigkeitssteigerung

Progressionstabelle – Gewichte je Typ in Prozent (Zielbild bis Welle 50):

Typ/ Welle	Grunt	Charger	Gunner	Tracker	Hunter	Radiant	Container
1	100	0	0	0	0	0	0
4	99	"1"	0	0	0	0	0
8	94	5	"1"	0	0	0	0
15	61	23	15	"1"	0	0	0
22	63	18	11	7	"1"	0	0
28	64	14	8	10	3	"1"	0
38	62	9	5	12	6	5	"1"
50	57	6	3	13	8	6	7

Hinweis: Wert = "1", bedeutet dass in der Einführungswelle der jeweilige Typ genau 1x vorkommt.

Tabelle 6.3: Tabellarische Übersicht der Einführungswellen, der Gegnertypen und dem Gegnermix.

Stand 13.12.2025

Ein weiterer Faktor, der die Schwierigkeitssteigerung vorantreibt, ist die prozentuale Verteilung der einzelnen Gegnertypen innerhalb einer Welle. Genauso wie bei der Einführung neuer Gegnertypen, wurde bei der Verteilung darauf geachtet, dass nicht sofort eine hohe Anzahl desselben Typs erscheint. Statt einer kontinuierlichen Steigerung der Schwierigkeit durch neue und stärkere Gegner wurde bei der Verteilung darauf geachtet, dass das Mischverhältnis über die Zeit stärker variiert. Das bedeutet, dass ein neuer Gegnertyp ab seiner Einführungswelle nur für eine gewisse Zeit öfter vorkommen wird.

Die Entscheidung fiel dabei auf Gegner ähnlichen Typs (Charger & Hunter und Gunner & Tracker), aber unterschiedlicher Bedrohungsstufe. Charger und Hunter haben beispielsweise die Eigenschaft ihre Fluggeschwindigkeit zu erhöhen, mit dem Ziel, eine direkte Kollision mit dem Spielecharakter zu erzwingen. Charger ändern ihre Flugrichtung einmalig, wodurch diesem Gegner ausgewichen werden kann. Hunter hingegen aktualisieren kontinuierlich die Flugrichtung auf die Position des Charakters, wodurch diesem Typen nicht ausgewichen werden kann.

07

SPIELANLEITUNG

In einer weit entfernten Galaxie unseres Universums lebt das niedliche Alien ein glückliches Leben auf seinem Asteroiden. Eines Tages bemerkt es jedoch seltsame Gegenstände auf seinem Asteroiden, die es zuvor noch nie gesehen hat. Beim Untersuchen fällt ihm auf, dass die unbekannten Objekte der Gesundheit seiner Heimat schaden. Es beschließt die Ursache herauszufinden. Dabei entdeckt es in seinem Orbit eine riesige Ansammlung menschengemachten Mülls, der durch die Tiefe des Universums direkt auf seinen Asteroiden zusteuert. Entschlossen seinen Lebensraum zu verteidigen, stellt es sich der heraneilenden Bedrohung entgegen.

Als Beschützer deines Asteroiden eliminierst du mit gezielten Schüssen die Gefahr und verteidigst deinen Orbit vor der maßlosen Verschmutzung. Geht dir ein Gegner durch die Lappen, musst du schnell reagieren um ein Zusammenstoßen mit den Objekten zu vermeiden. Dein Ziel ist es, Punkte zu sammeln und möglichst lange zu überleben, um dir deinen Platz in der Highscore-Liste zu sichern.

07 Spielanleitung

Technische Voraussetzung:

Bevor du dich auf die galaktische Reise begeben kannst, müssen einige technische Voraussetzungen erfüllt sein. Diese beziehen sich hauptsächlich auf die Hard- und Software deines Computers.

Hardware

Für dieses Spiel benötigst du einen internetfähigen Computer und eine einfache Webcam. Spielst du auf einem Laptop reicht die integrierte Kamera deines Geräts vollkommen aus.

Software

Stelle sicher, dass dein Betriebssystem aktuell ist und verwende einen modernen Browser (Chrome/Safari/Firefox) im Vollbildmodus. Führe ggf. ein Update deines Browsers durch.

Browser Empfehlung

Verwende Chrome mit aktiverter Grafik-Beschleunigung. Diese Option findest du in den Einstellungen des Browsers unter dem Menüpunkt „System“.

07 Spielanleitung

Vorbereitung:

Für ein optimales Spieleerlebnis empfehlen wir dir einige Vorkehrungen zu treffen, bevor du loslegst.

Dein Körper ist der Controller

In diesem Spiel ist dein Körper Werkzeug und Waffe zugleich. Du bist der Controller und bestimmst welche Eingaben gemacht werden. Damit du ihn effektiv und zuverlässig nutzen kannst, müssen dein Körper inkl. deiner Hände korrekt erfasst werden können. Positioniere dich stehend, ca. 70 Centimeter vor die Kamera und achte darauf, dass Hände und Handgelenke frei und nicht durch Ärmel bedeckt sind. Trage Kleidung, die einen hohen Kontrast zu der Farbe deiner Haut bietet.

Für eine angenehmere Körperhaltung halte deine Ellenbogen nah an deinem Oberkörper und strecke deine Hände in die Kamerarichtung. Positioniere die Kamera in der Höhe zwischen Brust und Bauchnabel. Passe ggf. den Neigungswinkel nach oben etwas an, so dass deine Handinnenflächen erfasst werden können.

Vermeide so gut es geht, deine Arme und Hände während des Spielens zu überkreuzen. Dies mindert einerseits eine schnelle Ermüdung der Arme und beugt zudem Fehlinterpretationen durch das Tracking-System vor.

Dein Raum und deine Umgebung

Eine gut beleuchtete Umgebung ist essenziell für eine zuverlässige Steuerung. Stelle deshalb sicher, dass die Lichtquelle deinen Raum und deine Hände gut ausleuchtet. Wichtig ist, dass starke Schattenbildung an deinen Händen minimiert wird. Nutze hierfür ggf. eine weitere Lichtquelle, die dich und deine Hände aus der Kamerarichtung beleuchtet.

Um Fehlinterpretationen des Trackings zu minimieren, achte auf einen neutralen Hintergrund. Stelle sicher, dass sich keine anderen Personen oder Bilder auf denen Menschen abgebildet sind, neben oder hinter dir befinden.

07 Spielanleitung

Hinweis

Eine Tracking-Reinitialisierung kann sowohl beim Laden des Spiels als auch während des Spielens vorkommen. Das ist normal und läuft im Hintergrund. Als Folge wird das Spiel kurzzeitig pausiert, bis das Tracking wieder neu initialisiert ist und Hände und Körper erkennt. Falls du das Gefühl hast, dass das Tracking ständig abbricht oder nicht richtig initialisiert, prüfe die oben genannten Punkte (Vorkehrungen und deine Hardware).



07 Spielanleitung

Spielaufruf

Den Spiele-Prototyp „Not So Friendly Neighbours“ kannst du direkt im Browser aufrufen in dem du die URL (<https://nsfn.prototype.schonspaet.de>) in die Suche eintippst, diesem Link hier folgst: [Not so Friendly Neighbours](#) oder auf den "Start Run" Button (unten) klickst.



Not So Friendly Neighbours

Ladebildschirm

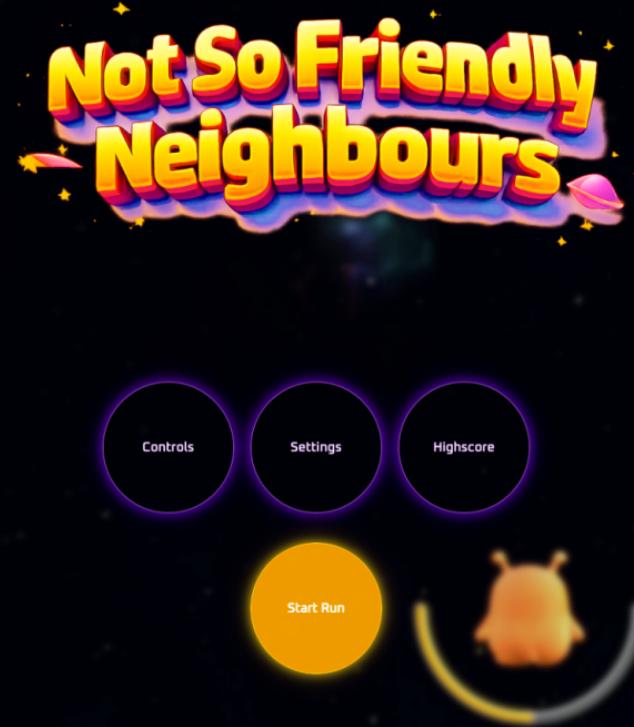
Der Ladebildschirm zeigt dir den Titel des Spieleprototyps und den Fortschritt des Spieleaufbaus. Sobald alles geladen ist, wirst du automatisch zum Startbildschirm weitergeleitet.

100%

07 Spielanleitung

Startmenü

Im Startmenü erhältst du eine Übersicht der Menüpunkte „Controls“, „Settings“ „Highscore“ und „Run“. Ab hier kannst du bereits auf deine Maus oder das Trackpad und die Tastatur deines Gerätes verzichten und steuerst deine Eingaben per Handgesten. Standardmäßig ist die rechte Hand als aktive Steuerungshand für die Menüpunkte ausgewählt. Falls du eine Menü-Steuerung mit der linken Hand bevorzugst, kannst du diese in „Settings“ auf die Hand deiner Wahl legen. Egal welche Hand du bevorzugst, mit beiden ist die Bedienung in allen Menüs dieselbe: Mit dem Handcursor zielst du auf die Buttons und führst eine Kneifbewegung mit deinem Zeigefinger und Daumen aus, um die Auswahl des Handcursors zu bestätigen.



07 Spielanleitung

Spielrunde

Mit einem tap auf Run startest du deine Runde und stellst dich der annähernden Bedrohung entgegen. Die Gegner erscheinen wellenweise auf dem Spielfeld und bewegen sich auf dich zu. Deine Aufgabe ist mit gezielten Schüssen die Gefahr in deinem Orbit zu eliminieren.

Navigation

Das Alien reagiert auf deine Körperbewegung und folgt deiner Position im Raum. In dem du dich bewegst kannst du gefährlichen Objekten und gegnerischen Angriffen ausweichen und verschaffst dir einen Überblick über die Gegner auf dem Spielfeld.



07 Spielanleitung

Basisangriff

Zielen

Nutze zum Anvisieren der Gegner die Zeigefinger beider Hände. An diesen befinden sich Markierungen, die dir dabei helfen die Zielrichtung zu finden. Ob du richtig anvisierst, erkennst du daran, dass sich die Farbe des Fadenkreuzes ändert. Jetzt bist du bereit einen Treffer zu erzielen.



Fadenkreuz

Schießen

Du eröffnest das Feuer, in dem du mit deinen Daumen auf die großen Knöchel deines mittleren Fingers tipst. Je schneller du tipst, desto mehr Schüsse kannst du abfeuern.

Spezialfähigkeit: EMP

Zusätzlich zum Basisangriff steht dir mit dem Elektromagnetischen-Impuls (EMP) eine weitere Fähigkeit zur Verfügung, die dir etwas mehr Kontrolle über das Spielfeld verschafft. Der EMP dehnt sich über die gesamte Spielfläche aus und vernichtet einfache Gegnertypen (Grunts) und entfernt in späteren Wellen den Schutzschild der Gegner. Um einen EMP zu aktivieren, halte deine Hände in den EMP-Flächen. Der Impuls startet automatisch, wenn der Angriff vollständig geladen ist. Nach jedem Spezialangriff benötigt diese Fähigkeit eine Regenerationszeit, um sie wieder einsetzen zu können. Diese Zeit wird dir in den EMP-Flächen als Countdown angezeigt.

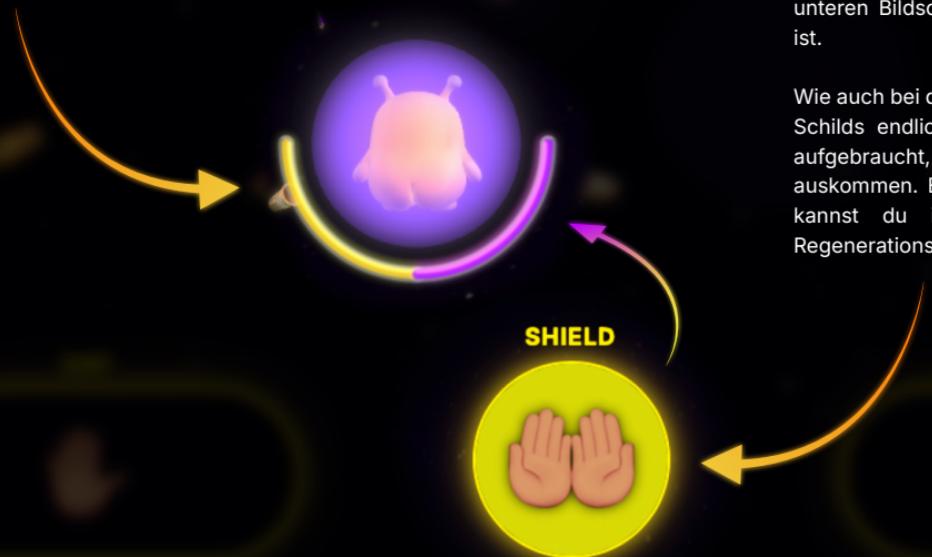


EMP

07 Spielanleitung

Statuswerte und Spielhilfen

Das Alien verfügt, wie bei jedem Lebewesen, nur ein Leben mit einer Anzahl an Lebenspunkten, welche sich bei Schaden nicht wieder regeneriert. Vermeide direkte Kollisionen mit Gegnern und feindlichen Projektilen, um lange auf dem Spielfeld zu bleiben. Den Status der Lebensanzeige kannst du über den gelben Balken einsehen, welcher mit dem Alien mitschwimmt.



Plasma Schild

In brenzligen Situationen, wenn du den Überblick verlierst oder du einem Treffer nicht mehr entkommen kannst, hast du die Möglichkeit einen Schutzschild zu nutzen. Dieser dient als Puffer und fängt für dich Kollisions- und Projektilschäden ab. Du aktivierst den Schild, in dem du deine Hände an den Handaußenseiten zusammenlegst und für zwei Sekunden hältst. Orientiere dich dabei an dem Schild-Icon mittig am unteren Bildschirmrand, wie die Handgeste durchzuführen ist.

Wie auch bei den Lebenspunkten, ist die Nutzungsdauer des Schildes endlich und schützt dich nur temporär. Ist diese aufgebraucht, musst du für einige Zeit ohne einen Schild auskommen. Erst wenn dieser sich wieder regeneriert hat, kannst du ihn erneut einsetzen. Die verbleibende Regenerationszeit erkennst du an dem Schild-Icon.

07 Spielanleitung

WAVE: 5

TARGETS: 3/13

SCORE: 345

Spielinformationen

Am oberen Bildschirmrand befinden sich Labels die dir wichtige Informationen zu deiner Spielrunde liefern:

- **WAVE** zeigt dir die Wellennummer an, in der du dich gerade befindest.
- **TARGETS** gibt an, wie viele Gegner sich aktuell auf dem Spielfeld befinden.
- **SCORE** ist deine bisher erreichte Punktzahl für das Eliminieren von Gegnern.



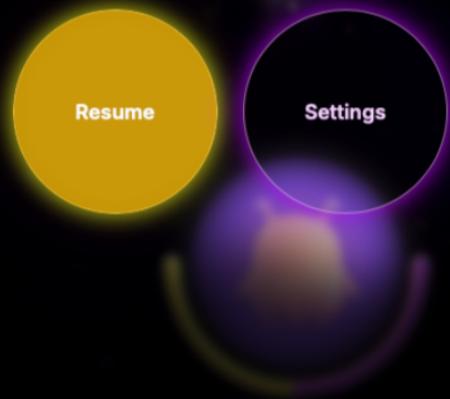
07 Spielanleitung

PAUSE

Pausenmenü

Während eines Runs kannst du das laufende Spiel kurzzeitig pausieren und das Pausenmenü aufrufen. Führe dafür die Time-Out Geste aus. Diese kennst du aus verschiedenen Sportarten. Forme mit den Händen ein T und halte diese Pose für einige Sekunden.

Im Pausenmenü hast du die Möglichkeit die aktuelle Runde zu beenden oder weiterzuspielen. Gegebenenfalls kannst du hier noch Einstellungen in den „Settings“ tätigen, falls du diese benötigst.



Hinweis

Im Pausenmenü wird die aktuelle Welle nur stark verlangsamt und läuft im Hintergrund weiter. Gegnerische Projektilen können dich so immer noch treffen.



07 Spielanleitung

Wellen

Jede Welle stellt einen eigenen Abschnitt deines Runs dar und ist mit einer kurzen Einblendung zwischen den Wellen (Wave Cleared) gekennzeichnet. Mit jeder Welle steigt zudem die Schwierigkeit gegenüber der Vorherigen weiter an.

Im Verlauf des Spiels steigt die allgemeine Geschwindigkeit stetig an. Diese beschränkt sich nicht nur auf die Fluggeschwindigkeit, sondern beeinflusst auch die Angriffs geschwindigkeit späterer Gegner.

Mit jeder weiteren Welle steigt die allgemeine Anzahl an Gegnern auf dem Spielfeld. Ausnahmen sind Wellen in denen neue Gegnertypen eingeführt werden. Dadurch hast du die Chance dich erstmal auf die neue Bedrohung art einzustellen.

In festgelegten Abständen werden neue Gegnertypen mit besonderen Eigenschaften und Angriffsarten in den Wellen eingeführt. Diese Gegnertypen sind ab ihrer Einführung fester Bestandteil jeder weiteren Welle.

Gegner

In diesem Spiel wirst du auf unterschiedliche Gegnertypen treffen, abhängig davon wie weit du es schaffst. Jeder Gegner hat unterschiedliche Werte, Eigenschaften und Angriffstypen.

Grunts

Grunts sind simple Gegnertypen und werden hauptsächlich durch einfache Alltagsobjekte dargestellt, die entlang des Orbit schweben. Sie besitzen keine eigene Angriffsart und sind mit einem einzigen Schuss eliminiert. Bei einer Kollision mit der Spielfigur fügen sie jedoch geringfügigen Schaden an, was vermieden werden sollte. Dieser Gegnertyp kommt im Wellenmix aus Gegnern am häufigsten vor.



07 Spielanleitung

Charger & Hunter

Dieser Gegnertyp hat es auf direkten Kollisionsschaden mit dem Spieler abgesehen. Sie unterscheiden sich in ihrer Treffsicherheit und dem Schaden, den sie verursachen.

Charger

Charger sind Asteroiden und sind natürliche Objekte in Weltraum. Aufgrund der Anziehungskraft des Weltraummülls ändern sie jedoch spontan ihre Flugrichtung und beschleunigen mit einer erhöhten Geschwindigkeit in deine Richtung. Schaffst du es nicht sie zu eliminieren, versuche ihnen auszuweichen.



Charger

Hunter sind alte Raketenstufen früherer Weltraumerkundungen der Menschen mit dem Ziel, außerirdisches Leben zu erkennen und dieses aufzusuchen. In ihnen befindet sich restlicher Treibstoff, der sich durch den kosmischen Unterdruck erneut entzündet und zu einer massiven Beschleunigung führt. Aufgrund ihres Defektes ist die Raketenstufe nicht mehr in der Lage selbstständig abzubremsen und wird mit dir kollidieren. Weißt du wie du das verhindern kannst?



07 Spielanleitung

Gunner & Tracker

Diese Gegnertypen besitzen aktive Angriffsarten, mit denen sie dich angreifen und aus dem Orbit holen möchten.

Gunner

Gunner sind alte und brüchige Behältnisse die nicht vollständig entleert worden sind bevor sie entsorgt wurden. Nun treiben sie durch den Weltraum und verstreuen ihren Inhalt im Orbit der direkt auf dich zu fliegt. Vermeide es davon getroffen zu werden.



Tracker

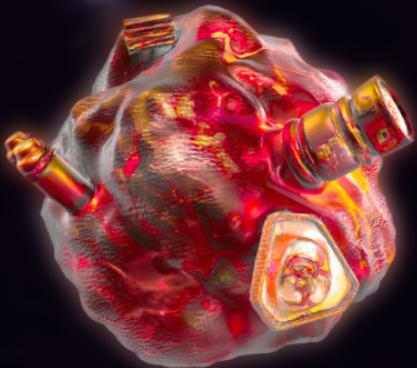
Tracker sind alte Weltraumsatteliten, die durch die Kollision mit anderen Himmelskörpern ihren Weg in den Orbit gefunden haben. Wiederholte Einschläge haben die innere Technik der Funkanlage durcheinander gebracht. Statt Funksignale senden sie nun glühendes Material in deine Richtung, welchem du nicht ausweichen kannst. Finde heraus wie du den Schaden minimieren kannst.



07 Spielanleitung

Radiant

Radianten sind Giftmüll-Ansammlungen, die sich aus giftigen Materialien zusammengesetzt haben. Durch diese Mischung entsteht eine gefährliche chemische Reaktion im inneren, wodurch sie zu einer tickenden Zeitbombe werden. Versuche den Radianten schnellstmöglich zu neutralisieren, bevor sie explodieren und radioaktive Strahlung verstreuhen.



Container

Obwohl dieser **Großraumbehälter** keine direkte Gefahr aufgrund seiner niedrigen Geschwindigkeit und fehlender Angriffsmethoden darstellt, solltest du diesen Koloss nicht unterschätzen. Er beinhaltet eine riesige Menge unsortierter und nicht recyclebarer Objekte, die du bereits kennst. Zerstörst du den Container, werden zusätzliche Gegner im Orbit verstreut.



07 Spielanleitung

Weitere Menüs und Übersichten

Wave Cleared

Sobald alle Gegner einer Welle vom Spielfeld verschwunden sind, gilt diese Welle als „cleared“. Dieses wird die über das „Wave Cleared“-Overlay bestätigt und gleichzeitig ein Countdown zur Vorbereitung der nächsten Welle untergezählt.



Abbildung 7.1: Zwischenansicht der Wellen mit Countdown zur nächsten Welle. Screenshot des Prototyps (Stand 13.12.2025).

Settings

Im Menüpunkt Settings kannst du einige Einstellungen zu deiner Steuerhand durchführen. Hier kannst du wählen mit welcher Hand du dich durch das Menü navigieren möchtest. Zusätzlich kannst du wählen, ob du während des Spielens Hintergrundmusik und Soundeffekten hören möchtest und die für dich passende Lautstärke wählen.

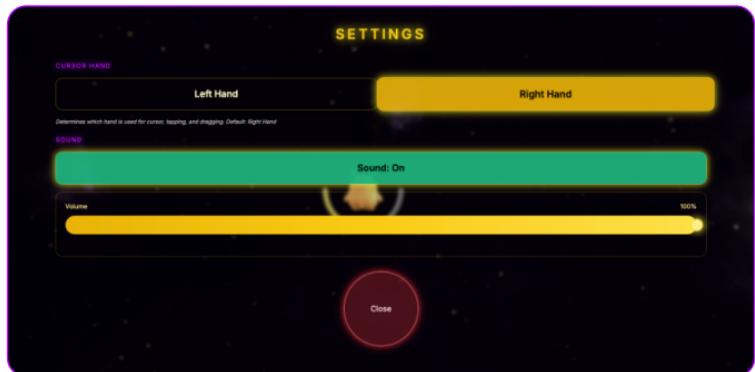


Abbildung 7.2: Oberfläche des Einstellungsmenüs mit Buttons zum Wechseln der Steuerhand und Soundeinstellungen. Screenshot des Prototyps (Stand 13.12.2025).

07 Spielanleitung

Run Terminated

Endet das Spiel landest du auf dem RUN TERMINATED Screen und erhältst einen Missionsreport. Dieser liefert dir eine Statistik zu deinem Lauf, damit du auf den ersten Blick erkennst, wie deine Performance war. In die Statistik fließen deine Score-Punkte und die Anzahl der Waves ein. Zusätzlich bekommst du die Gesamtanzahl der Gegner und eine Auflistung der einzelnen Gegnertypen, die du in diesem Lauf eliminiert hast. Von hier aus kannst du dir die aktuellen Einträge des Leaderboards ansehen oder über RETRY zurück zur Startseite gelangen und einen neuen RUN starten.

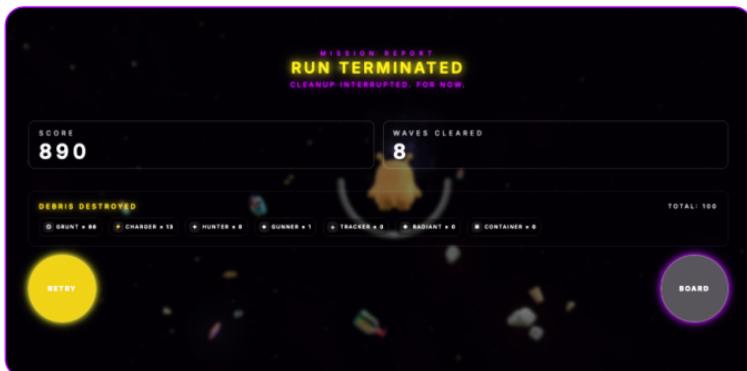


Abbildung 7.3: Mission Report Overlay. Hier werden überlebte Wellen, so wie die Anzahl eliminiert Gegner angezeigt und Punkte angezeigt. Screenshot des Prototyps (Stand 13.12.2025).

Hast du genügend Punkte erhalten, erscheint noch vor dem RUN TERMINATED - Screen eine Maske, auf der du über einen Platz in der Highscore Liste informiert wirst. Hier hast du die Möglichkeit dich auf der Liste mit deinem Spielernamen zu verewigen. Hierfür musst du sogar deine Tastatur nutzen, um deinen Namen eintragen zu können.

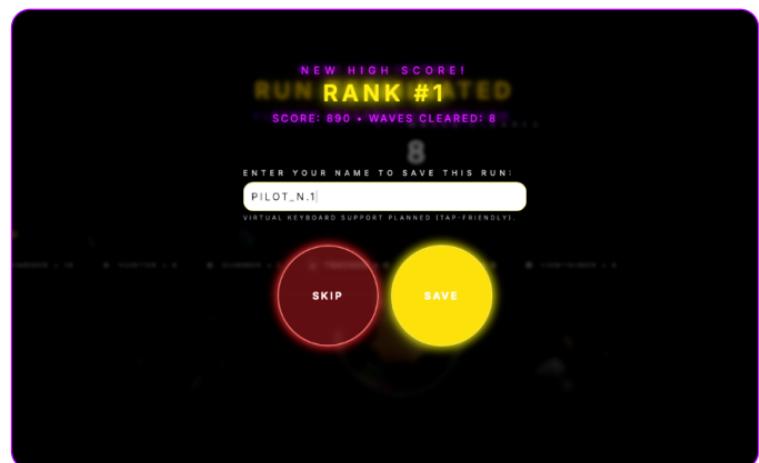


Abbildung 7.4: Pop-Up für das Erreichen eines Highscores mit der Übersicht der vorherigen Informationen (Score, Wellenanzahl) und dem Rang. Screenshot des Prototyps (Stand 13.12.2025).

07 Spielanleitung

Leaderboard

Vor und nach jedem RUN kannst du dir das Leaderboard mit dem Highscore ansehen. Navigiere dich einfach aus dem Startmenü per Tap auf die Ansicht der Liste. Hier werden die Top 50 Spieler mit Namen, erreichen Punkten und überstandenen Wellen, aufgeführt.

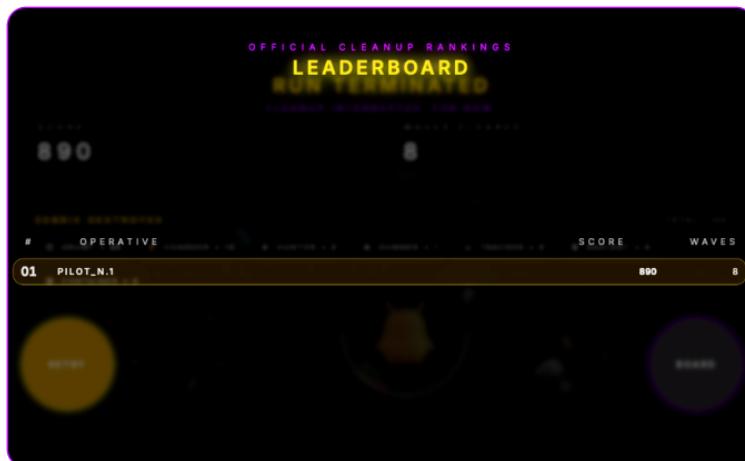


Abbildung 7.5: Leaderboardansicht des Spiels. Hier werden die Top 50 Spieler mit ihrem erreichten Score und der überstandenen Wellenanzahl aufgeführt. Screenshot des Prototyps (Stand 13.12.2025).

Controls

Für eine Übersicht der Gesten navigiere dich in das Controls-Menü. Dort findest du eine Auflistung der verwendbaren Gesten und eine direkte Anleitung wie du sie in diesem Spiel einsetzen kannst.



Abbildung 7.6: Informationsoberfläche „Controls“ mit der Übersicht aller in diesem Spiel verfügbaren Steuerungsgesten. Screenshot des Prototyps (Stand 20.12.2025).

08 Reflexion und Fazit

Im Rahmen dieser Arbeit wurde ein webbasiertes Spielprototyp entwickelt, welcher per Handgesten und Körperbewegungen gespielt werden kann. Der Fokus lag auf der Machbarkeit und der praktischen Umsetzbarkeit eines Spiels mit berührungsloser Steuerung für Desktopbrowser, weshalb eine Smartphone Kompatibilität nicht vollständig implementiert wurde. Ebenso wurden noch keine Soundeffekte und Hintergrundmusik in diesen Prototypen integriert, auch wenn die Schaltflächen im Einstellungsmenü dies andeuten. Für eine zukünftige Version des Spiels könnte demnach eine Sound-Landschaft entwickelt und an gewisse Aktionen oder Zustände geknüpft werden, die das Spieelerlebnis intensivieren. Momentan werden die Einträge der Highscore Liste nur in dem eigenen Browser gespeichert. Für eine dauerhafte Speicherung der Spielerdaten müsste die aktuelle Implementierung der Highscore Liste auf eine globale Lösung angepasst werden, bei der es nur eine zentrale Liste für alle Spielrunden gibt.

Im Rückblick war die gewählte Entwicklungsumgebung für dieses Spiel ein entscheidender Faktor dafür, dass der Prototyp schnell entwickelt werden konnte. Für dieses Projekt waren Cursor AI und der Babylon.js besonders hilfreiche Mittel für die gestalterische Arbeit und die programmatiche Entwicklung eines webbasierten Videospiels. Der Editor erleichterte das Erstellen der Szene und die Organisation und Integration von 3D-Objekten. Die Implementierung der Grundmechaniken und Abläufe konnten mit der Unterstützung durch den Coding Agenten von Cursor AI gut umgesetzt werden. Bei auftretenden Fehlern und großen Änderungen im Code, konnte dadurch eine Lösung gefunden werden. Jedoch musste regelmäßig evaluiert werden, ob der verfolgte Ansatz oder die Reihenfolge von Abfragen und die Kollisions-Checks tatsächlich das erwartete Ergebnis liefern. Die generierten TypeScript-Skripte für die Spielemechanik, das

Gegnerverhalten und die Steuerung, ließen sich unkompliziert aus dem internen Filebrowser an die Szenenobjekte hängen und direkt im Browser testen. Dadurch konnte die noch fehlende Funktion zum Animieren von Objekten und der Szene im Editor durch gecodete Skripts teilweise kompensiert werden. Was nicht umgesetzt werden konnte sind die Charakteranimationen für verschiedene Zustände wie Leerlauf, Bewegung, Sterben, Schießen und Explodieren der 3D Objekte in der Szene. Diese würden dem Spiel eine höhere Qualität verleihen, müssten aber im Vorfeld in externen Programmen wie Blender erstellt werden.

Die Gesteuerung ließ sich grundsätzlich schnell definieren, einbinden und im Projekt nutzen. Eine größere Herausforderung lag in der Erkennung unbeabsichtigter Eingaben bei ungünstiger Handhaltung oder der Überlappung der Hände während des Spielens. Das liegt zum einen daran, dass eine herkömmliche Webcam keine echten Tiefeninformationen liefert und diese von MediaPipe nur geschätzt werden. Zum anderen spielen äußere Faktoren wie Lichtverhältnisse, Schattenbildung und Kameraposition eine wichtige Rolle und beeinflussen die Zuverlässigkeit des Trackings stark. In einem kleinen Spieltest mit drei Personen, wurde bestätigt, dass die Nutzung von diesen Rahmenbedingungen abhängig ist und auch die Hardware des Rechners eine entscheidende Rolle für eine reibungslose Spielerfahrung spielt. Positiv hervorgehoben wurde, dass die Auswirkung des Trackings bereits im halbtransparenten Hintergrund des Startmenüs sichtbar ist. So lässt sich der spielbare Charakter noch vor dem Start einer Spielrunde über den Bildschirm bewegen und lässt erahnen, dass der Körper und die Hände für die Steuerung genutzt werden könnten. Zudem wurde der Vorschlag geäußert die Menüsteuerungsoptionen bereits auf der Startseite anzuzeigen, um beim Eintritt ins Spiel die erste Aktion durchführen zu können. Für eine

08 Reflexion und Fazit

ausgereiftere Version wären zusätzliche Maßnahmen für die Robustheit der Steuerung sinnvoll. Weitere iterative Tests könnten zudem dabei helfen Bedingungen für die Trigger, wie Toleranzbereiche von Landmarks, Kameraabstände und Haltezeiten der Gesten noch sicherer zu gestalten. Zusätzlich müsste geprüft werden, inwiefern Personen mit motorischen Einschränkungen, Schmerzen oder fehlender Finger und Hände der Zugang zum Spiel gewährleistet werden kann. Eine mögliche Lösung wäre es, die Steuerung personalisierbar zu gestaltet, und das Setzen alternativer Landmarks zu ermöglichen.

Fazit

Ziel dieser Arbeit war die Entwicklung einer webbasierten Spieleprototyps, der mit Handgesten sowie der Körperbewegung spielbar ist. Im Fokus stand die Umsetzung eines verständlichen und steuerbaren User Interfaces für berührungslose Eingaben. Visuelle Zustände, hoher Kontrast und klare UI-Elemente ermöglichen eine gute Lesbarkeit, ohne vom Spielgeschehen abzulenken. Direktes Feedback durch Indikatoren und sichtbare Zustandswechsel unterstützen sowohl die Navigation in den Menüs als auch die Interaktion während der Spielrunde. Das Gesten-Mapping ist so ausgelegt, dass alle relevanten Spielinhalte bedient werden können, wodurch auf die Nutzung einer Maus und Tastatur verzichtet werden kann. Die entwickelten Gesten sind dabei an natürliche und bekannte Handhaltungen angelehnt, wodurch eine schnelle und leichte Lernbarkeit gewährleistet werden kann. Gleichzeitig zeigt der Prototyp aber auch einige Grenzen diese Eingabemethode: Die Zuverlässigkeit des Trackings ist stark abhängig von äußeren Bedingungen wie Licht, Schatten und der Leistung der genutzten

Hardware. Neben der Gestaltung der Gesten müssen Konditionen für die Trigger definiert werden, um Fehlinterpretationen des Trackings abzufangen. Insgesamt zeigt der Prototyp, dass der Einsatz von gestenbasierten Eingabemethoden in Browserspielen, unter der Berücksichtigung einiger Voraussetzungen, geeignet ist.



09 Anhang

09.1 Quellenverzeichnis

European Space Agency (ESA) (2025): *Space Environment Statistic (DISCOSweb)*. Letzte Aktualisierung: 21.10.2025. Verfügbar unter: <https://sdup.esoc.esa.int/dicosweb/statistics/> (abgerufen am 19.12.2025).

Game Archive – No Commentary Gameplay (2015): *Space Invaders 1978 – Arcade Gameplay* [Video]. YouTube. Verfügbar unter: <https://www.youtube.com/watch?v=MU4psw3ccU> (abgerufen am 03.12.2025).

Google AI Edge (2025): *MediaPipe Solutions guide* (MediaPipe Solutions Guide). Verfügbar unter: <https://ai.google.dev/edge/mediapipe/solutions/guide> (abgerufen am 16.12.2025)

Google AI Edge (2025): *Hand landmarks detection guide* (MediaPipe Solution – Hand Landmarker). Zuletzt aktualisiert am 13.01.2025 (UTC). Verfügbar unter: https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker (abgerufen am 10.11.2025)

Google AI Edge (2025): *Pose landmarks detection guide* (MediaPipe Solution – Pose Landmarker). Zuletzt aktualisiert am 13.01.2025 (UTC). Verfügbar unter: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker (abgerufen am 10.11.2025)

Google AI Edge (2025): *Gesture recognition task guide* (MediaPipe Solution – Gesture Recognizer). Zuletzt aktualisiert am 13.01.2025 (UTC). Verfügbar unter: https://ai.google.dev/edge/mediapipe/solutions/vision/gesture_recognizer (abgerufen am 10.11.2025)

Google AI Edge (2025): *Hand Landmarker* (MediaPipe Solutions – Hand Landmarker) [Abbildung]. Verfügbar unter: https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker (abgerufen am 16.12.2025)

Google AI Edge (2025): *Pose Landmarker* (MediaPipe Solutions – Pose Landmarker) [Abbildung]. Verfügbar unter: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker (abgerufen am 16.12.2025)

Google Fonts (2025): *Oxanium* [Schriftart]. Verfügbar unter: <https://fonts.google.com/specimen/Oxanium> (abgerufen am 20.12.2025).

Host Europe (2025): *Wie funktioniert die Gesteuerteuerung? Begriff, Funktionsweise und intuitive Gestaltung im Überblick*. Host Europe Blog. Verfügbar unter: <https://www.hosteurope.de/blog/wie-funktioniert-die-gesteuerteuerung-begriff-funktionsweise-und-intuitive-gestaltung-im-ueberblick/> (abgerufen am 20.12.2025)

Jeon, S., Kim, S.-K., & Kim, S. (2023): *Research on Virtual Reality Games Using Hands as User Interface for Oculus Quest*. Korean Society for Computer Game, 34(4), 17-22. ResearchGate. Verfügbar unter:
[https://www.researchgate.net/publication/377209420 Research on Virtual Reality Games Using Hands as User Interface for Oculus Quest](https://www.researchgate.net/publication/377209420)
(abgerufen am 20.12.2025)

Lewis, B. (14.12.2023): *Besseres Fahrerlebnis mit Auto UX-Technologie*. Mouser German Blog (Bench Talk). Verfügbar unter:
<https://www.mouser.de/blog/besseres-fahrerlebnis-mit-auto-ux-technologie>
(abgerufen am 20.12.2025)

Maerzke, S. (06.10.2023): *Geräte-Neuheiten auf der IFA 2023: Superlative fürs Smart Home*. Küchen&Design Magazin. Verfügbar unter:
<https://www.kuechen-design-magazin.de/geraete-neuheiten-auf-der-ifa-2023-superlative-fuers-smart-home/>
(abgerufen am 20.12.2025)

Meta. (o.D): *Learn about Hand and Body Tracking on Meta Quest*. Verfügbar unter: <https://www.meta.com/de-de/help/quest/290147772643252/>
(abgerufen am 20.12.2025)

Popov Neven (2010): *Futurama – A Big Piece Of Garbage* [Video]. YouTube. Verfügbar unter: <https://www.youtube.com/watch?v=Le4Sync85zo>
(abgerufen am 3.12.2025).

Space Invaders Wiki (o. D.): *Space Invaders (Activision)*. In: Space Invaders Wiki. Fandom. Verfügbar unter:
[https://spaceinvaders.fandom.com/wiki/Space_Invaders_\(Activision\)](https://spaceinvaders.fandom.com/wiki/Space_Invaders_(Activision))
(abgerufen am 03.12.2025).

TensorFlow (o.D.): *WhyTensorFlow*. Verfügbar unter: <https://www.tensorflow.org/about>
abgerufen am 20.12.2025)

TensorFlow (o.D.): *For JavaScript*. Verfügbar unter: <https://www.tensorflow.org/js> (abgerufen am 20.12.2025)

TensorFlow (o.D.): *For JavaScript – Models*. Verfügbar unter: <https://www.tensorflow.org/js/models>
(abgerufen am 20.12.2025)

TensorFlow (o.D.): *tfjs-models-hand-pose-detection* (GitHub-Repository), Commit: [703c4df](#). Verfügbar unter: <https://github.com/tensorflow/tfjs-models/tree/master/hand-pose-detection>
(abgerufen am 20.12.2025)

U.S. Department of Energy, Office of Nuclear Energy (2023): *3 Reasons Why We Don't Launch Nuclear Waste into Space*. Energy.gov, 05.04.2023. Verfügbar unter: <https://www.energy.gov/ne/articles/3-reasons-why-we-don-t-launch-nuclear-waste-space>
(abgerufen am 20.12.2025)

Wikipedia (2025a): *Space Invaders*. In: Wikipedia – Die freie Enzyklopädie.

Zuletzt bearbeitet am 18.08.2025. Verfügbar unter:

https://de.wikipedia.org/wiki/Space_Invaders

(abgerufen am 03.12.2025).

wizAI solutions GmbH (o.D.): *Digital Signage, sensorgesteuert*.

Verfügbar unter: <https://wizperzone.com/digital-signage-hardware/digital-signage-sensorgesteuert/>

09.2 Software/Hardware und Werkzeuge

Kategorie	Toll/Werkzeug	Version	Zweck
Game Engine	Babylon.js	8.34.0	Umsetzung des 3D-Spiels (Rendering, Szene, Gameplay)
Editor	Babylon Editor	5.2.3	Szenenaufbau, Platzierung von Objekte
Tracking	Google MediaPipe Solutions: · Hand detection · Pose detection · Camera Utils	0.4.1675469240 0.5.1675469404 0.3.167546686	Kamerabasiertes Hand- und Körpertracking zur Gesten-/Bewegungssteuerung
Programmiersprache	TypeScript/JavaScript	5.8.3	Implementierung der Spiel- und Steuerlogik im Web (Vorgabe)
Entwicklungsumgebung	Cursor AI + Coding Agent	2.1.39 (Universal)	Programmierung, Codegenerierung, Debugging und Organisation des Quellcodes
Charakter/Gegnerdesign (Bild)	OpenAI (Bildgenerierung)	ChatGPT 5.1 ChatGPT 5.2	Erstellung des Gegner-/Charakterdesigns
Titelbild	OpenAI (Bildgenerierung)	ChatGPT 5.2	Erstellung des Titelbilds „Not So Friendly Neighbours“
3D-Assets	Hyper3D Rodin	Gen-2 (1120) V1.9	Erstellung der 3D Assets aus den Bildern
Bildbearbeitung	Affinity by Canva	3.0.2.3912	Bildbearbeitung, Erstellung der Diagramme
Hardware	MacBook Pro 16-inch Chip: AppleM1 Max Memory: 32 GB	macOS Tahoe 26.1	Arbeitsmittel
Browser	· Safari · Chrome · Firefox	· Version 26.1 (21622.2.11.11.9) · Version 143.0.7499.147 · 146.0.1 (aarch64)	Entwicklungstests

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit mit dem Titel:

Berührungslose Interaktion: Konzeption und Entwicklung eines Videospiels mit kamerabasierter Gestensteuerung

Selbstständig, ohne unerlaubte fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

Detmold, 20.12.2025

Ort, Datum

Unterschrift