

Ostbayerische Technische Hochschule Amberg-Weiden
Fakultät Elektrotechnik, Medien und Informatik

Studiengang Künstliche Intelligenz

Studienarbeit AR/VR

von

André Kestler

Der Igel Simulator

Bearbeitungszeitraum: von 01. Dezember 2021
 bis 02. Februar 2022

1. Prüfer: Prof. Dr.-Ing. Gerald Pirkel

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufgabenstellung	1
1.2	Projektmanagement	1
1.2.1	GitHub	1
1.2.2	Trello	1
1.2.3	Discord	2
1.2.4	Unity	2
2	Spielvorstellung	3
2.1	Spielmechanik	3
2.2	Immersion	4
3	Entwicklung	5
3.1	Igel Informationen	5
3.2	Soundeffekte	6
3.3	Station 7: Rasenmäher	6
3.3.1	Bezier-Kurve	6
3.3.2	Implementierung	7
3.3.3	Problematik	8
3.4	Station 8: Straße	9
3.4.1	Auto Design	9
3.4.2	Implementierung	10
3.5	Station 4: Zaun	10
3.6	Spielende	11
4	Zusammenfassung und Ausblick	12

4.1	Zusammenfassung	12
4.2	Ausblick	12
	Literaturverzeichnis	14
	Abbildungsverzeichnis	15

Abkürzungsverzeichnis

AR	Augmented Reality
VR	Virtual Reality
MR	Mixed Reality

Kapitel 1

Einleitung

1.1 Aufgabenstellung

Im Rahmen des Studiengangs Ausgewählte Themen AR/VR soll ein Projekt im Themenbereich Augmented Reality (AR), Virtual Reality (VR) oder Mixed Reality (MR) bearbeitet werden. Die Bearbeitung des eigens ausgesuchten Projekts soll in einer Gruppe erfolgen. Die Gruppe für dieses Projekt besteht aus folgenden Personen: Johannes Horst, Saniye Ogul, Stefanie Hofmann, Magdalena Bienefeld und mir (André Kestler). Als Projekt wurde ein Igel Simulator als VR-Spiel entworfen und programmiert. Mit der VR-Brille Oculus Quest kann das Spiel an einem Computer mithilfe des Link-Kabels gespielt werden.

1.2 Projektmanagement

1.2.1 GitHub

Zur Versionsverwaltung des Projekts wurde GitHub verwendet. Dort hat Johannes Horst ein Repository für das Projekt angelegt und jeder Teilnehmer hat seine Arbeiten dort einchecken können. Außerdem ist durch dieses Tool jeder auf dem aktuellsten Softwarestand und Fehler können einfach überblickt werden. [3]

1.2.2 Trello

Bei Trello handelt es sich um einen Onlinedienst, der ein Projekt bei der Verwaltung und Verteilung von Aufgaben unterstützt. Durch ihn lassen sich Arbeitsboards anlegen und

in diesem Listen für bestimmte Themengebiete. In den Listen können Karten für die zu bearbeitenden Aufgaben erstellt werden und Benutzer zugeordnet werden. Der Vorteil ist, dass man immer einen Überblick über die Aufgaben hat und zum Ende des Projekts eine Übersicht über die bearbeiteten Themen hat. [8]

1.2.3 Discord

Zur Absprache und Klärung von Problemen wurde auf das Kommunikationstool Discord zurückgegriffen. Darin wurden wöchentlich Meetings angesetzt, um den Zwischenstand und die Aufgabenverteilung festzulegen.

1.2.4 Unity

Für das zu bearbeitende Thema hat sich meine Gruppe für, die in der Vorlesung kennengelernte Spielentwicklungsumgebung, Unity entschieden. Die Anwendung wurde mit der Unity Version 2020.3.20f1 entwickelt. [9]

Kapitel 2

Spielvorstellung

2.1 Spielmechanik

Die Idee meiner Gruppe, war es ein Spiel zu entwerfen, welches zusätzlich zu den unterhaltenden Faktor auch noch einen lehrenden Einfluss besitzt. In einer virtuellen Welt spielt der Spieler einen Igel in First-Person. Dazu kann sich der Spieler mit den Controllern auf der Karte, welche in Abbildung 2.1 dargestellt ist, fortbewegen. Im Laufe des Spiels lernt der Anwender wissenswertes über Igel und muss sein Wissen bei den verschiedenen Stationen auf die Probe stellen. Das erfolgreiche Bearbeiten der Stationen ist nötig, da man als Errungenschaft jeweils ein Blatt erhält. Wenn alle Blätter gesammelt wurden, wird das Spiel im Haus der Igelfamilie beendet.



Abbildung 2.1: Übersicht über die Spielkarte 1: Tutorial, 2: Essen, 3: Trinken, 4: Zaun, 5: Teich, 6: Feind, 7: Rasenmäher, 8: Straße. Quelle: Eigene Aufnahme

Zur Steuerung wurde eine Kombination aus Teleportation und Fortbewegung mit den Joysticks der Controller gewählt. Es wurde keine reine physische Fortbewegung implementiert, da die Karte des Spiels sonst auf die jeweilige begehbare reale Umgebung angepasst werden muss. Damit ist das Spiel auch unabhängig der räumlichen Möglichkeiten spielbar.

2.2 Immersion

Ein Spiel wird als immersiv bezeichnet, wenn der Spieler vollständig in die fiktive Welt eintaucht und dabei die Wahrnehmungen der realen Umgebung sinken oder gänzlich verschwinden. Dies wurde in dem Spiel mithilfe der Größe der modellierten Umgebung versucht zu erreichen. Da ein Igel relativ klein ist, wurde die Welt mit seinen Gegenständen dementsprechend größer modelliert. Dieser Sachverhalt ist in Abbildung 2.2 zu sehen. Für ein schöneres Spielgefühl sind zudem noch Audiobestandteile eingefügt, welche für ein besseres Ambiente sorgen. So sind an der Straße die Autos oder bei der Station des Rasenmähers dieser zu hören.



Abbildung 2.2: Darstellung der Größenunterschiede. Quelle: Eigene Aufnahme

Es sei aber zu erwähnen, dass durch die Wahl der Fortbewegungsmethode und der Low-Poly Grafik der Effekt der Immersion gemindert wird.

Kapitel 3

Entwicklung

3.1 Igel Informationen

Da das Spiel einen Einblick in die Lebensweise eines Igels geben soll, mussten im ersten Schritt die dazu nötigen Informationen eingeholt werden. Dazu wurde zum einen im Internet danach gesucht und Magdalena Bienefeld hatte sich mit Jana Zwanziger aus Führt getroffen, die eine Pflegeeinrichtung für Igel führt. Dadurch konnten Informationen aus erster Hand gesammelt werden. [5]

Um Igel zu unterstützen, soll man ihnen als Nahrungsmöglichkeit keine Milch zur Verfügung stellen, sondern Wasser. Igel können Milch nicht verarbeiten, da sie Laktoseintolerant sind. Am liebsten Fressen sie Käferlarven, Raupen oder Regenwürmer. Zur Unterstützung der Nahrungsfindung kann der Mensch ihnen Katzenfutter oder Eier zur Verfügung stellen. Auf keinen Fall dürfen sie Äpfel, Nüsse oder andere Speisereste erhalten.

Neben den natürlichen Fressfeinden wie dem Dachs, dem Uhu oder dem Fuchs sind in den letzten Jahren durch den technologischen Fortschritt die Straßen mit den Autos und vor allem die Rasenmäherroboter hinzugekommen, da diese meistens auch dann in den Gärten fahren, wenn Igel auf ihren Streifzügen sind. Durch den Verlust von naturnahen Gärten verlieren Igel immer weiter ihren Lebensraum.

Igel halten auch einen Winterschlaf, da sie in den Monaten zu wenig Nahrung finden würden. Abhängig von der Witterung suchen sie sich gegen November einen Unterschlupf. Während des Schlafs fahren Igel ihre Körpertemperatur und den Stoffwechsel auf ein Minimum herunter. Vor ihren Schlaf fressen sie sich genug Fettreserven an. [4, 7]

Diese Elemente wurden in Form von den Stationen im VR-Spiel umgesetzt.

3.2 Soundeffekte

Nachdem das erste Konzept der Welt festgehalten wurde und die Rahmenbedingungen festgestanden waren, ging es auf die Suche nach Soundeffekten. Bei der Suche nach möglichen Audiodateien war es insbesondere wichtig auf die Lizenzen zu achten. So wurde hauptsächlich Musik mit der CC0 Lizenz verwendet. Diese Lizenz ermöglicht es, ohne den Urheber zu kontaktieren oder zu erwähnen seine Arbeit zu verwenden. Zudem ist es gestattet, dass die Audiodateien auch bearbeitet werden. [1, 2, 6]

Ein Teil der Musik wurde mit der open-source Software Audacity geschnitten. Somit konnten sie auf die passende Länge gekürzt werden oder aus einem längeren Soundtrack der interessierende Teil herausgeschnitten werden.

Die im Spiel zu hörenden Texte wurden durch Saniye Ogul aufgenommen und eingespielt. Zudem hat sie die gefundenen und geschnittenen Soundeffekte bei den einzelnen Stationen implementiert.

3.3 Station 7: Rasenmäher

3.3.1 Bezier-Kurve

Eine Kurve im \mathbb{R}^3 ist durch eine Abbildung

$$c : [a, b] \mapsto \mathbb{R}^3, t \mapsto \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} \quad (3.1)$$

gegeben. Der Parameter t kann dabei als Variable über der Zeit interpretiert werden und die Kurve wird somit im Intervall $t \in [a, b]$ durchlaufen.

Zu jedem Zeitpunkt soll $b(t)$ eine gewichtete Summe der Kontrollpunkte der Kurve sein:

$$b(t) = B_0(t)b_0 + B_1(t)b_1 + B_2(t)b_2 + B_3(t)b_3. \quad (3.2)$$

Es soll folgendes gelten:

- $B_0(t) + B_1(t) + B_2(t) + B_3(t) = 1 \quad \forall t \in [a, b]$
- $B_i \geq 0 \quad \forall t \in [a, b], i = 0, 1, 2, 3.$
- $t = 0$ ist $B_0^3(0) = 1$ und $B_1^3(0) = B_2^3(0) = B_3^3(0) = 0$
 $\Rightarrow b(0) = b_0$
- $t = 1$ ist $B_3^3(1) = 1$ und $B_0^3(1) = B_1^3(1) = B_2^3(1) = 0$
 $\Rightarrow b(1) = b_3$

Für eine kubische Bezier-Kurve $b(t)$ werden die Bernstein-Polynome als Gewichtsfunktion verwendet. Diese haben folgende Gestalt und erfüllen die oben genannten Eigenschaften: [10]

$$B_k^3 = \binom{3}{k} X^k (1 - X)^{3-k} \quad (3.3)$$

3.3.2 Implementierung



Abbildung 3.1: Übersicht über die Rasenmäher Station. Quelle: Eigene Aufnahme

In Abbildung 3.1 ist die Übersicht über den Aufbau der Station gegeben. Die Landschaft und der Rasenmäherroboter wurden von Magdalena Bienefeld entworfen. Mithilfe meines Strecken Skripts wurde dem Roboter leben eingehaucht.

Dazu kann mit den Ankerpunkten 2 und 3 die Streckenform angepasst werden und mit den Ankerpunkten 1 und 4 der Start- und Zielpunkt der Kurve festgelegt werden. Durch aneinanderreihen dieser Kurven fährt der Roboter die geschlossene Kurve ab. Dabei ist zu beachten, dass der Punkt 1 von Kurve A mit dem Punkt 4 von Kurve B und Punkt 4 von

Kurve A mit dem Punkt 1 von Kurve B zusammenfallen. Falls dies nicht beachtet wird, kommt es zu Sprüngen in der Strecke. Die Strecke wurde nach dem Vorbild aus einem Video entworfen. [11]

3.3.3 Problematik

Einer der ersten Entwürfe der Karte hatte über die gesamte Fläche Unebenheiten. Dies führte allerdings durch die implementierte Gestalt der Bézierkurve dazu, dass der Roboter durch diese Unebenheiten hindurch fährt.

In ersten Gedanken sollte das Hinzufügen einer Rigidbody Komponente und das Einschalten der Gravitation das Problem beheben. Es sollten die Koordinaten in der xz-Ebene mit der Kurve berechnet werden und der Roboter durch die Gravitation an den Boden gezogen werden. Ein Collider sollte verhindern, dass das Objekt durch die Landschaft fällt.

Jegliche Versuche führten allerdings nur zu dem Ergebnis, wie es in Abbildung 3.2 dargestellt ist. Das Problem wurde durch das Anpassen der Landschaft umgangen. Im neuesten Entwurf sieht die abgefahrene Strecke des Roboters nach einer gemähten Wiese aus.



Abbildung 3.2: Darstellung des Problems der unebenen Fläche. Quelle: Eigene Aufnahme

3.4 Station 8: Straße

3.4.1 Auto Design

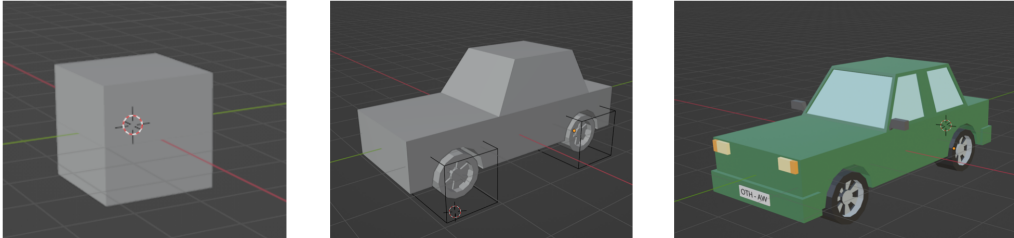


Abbildung 3.3: Entwicklungsprozess des Autos mit Blender. Links: Ausgangsobjekt, Mitte: Zwischenstand, Rechts: Fertige Objekt. Quelle: Eigene Aufnahme

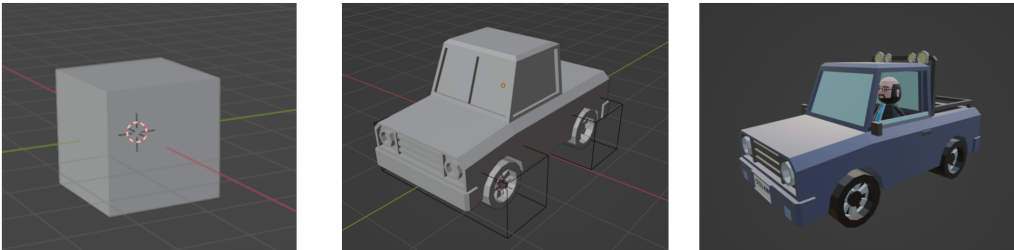


Abbildung 3.4: Entwicklungsprozess des Pickups mit Blender. Links: Ausgangsobjekt, Mitte: Zwischenstand, Rechts: Fertige Objekt. Quelle: Eigene Aufnahme

Mithilfe der 3D-Objektmodellierungssoftware Blender wurden die in Abbildung 3.3 und 3.4 abgebildeten Objekte erstellt. Ausgehend von dem Würfel sind die Autos Stückweise mit den Werkzeugen (Extrude, Cut, Scale, ...) von Blender erweitert worden. Als Abschluss werden die Materialien für die Texturen der Objekte erstellt.

Im Endprozess kann man mithilfe von Keyframes in Blender die Animationen festlegen. Die Reifen werden mit einer einfachen Rotationsbewegung über eine Zeitspanne verändert. Für die Wink-Bewegung des Charakters im Auto ist auf die Möglichkeiten der Knochen zurückgegriffen worden. Den Körper können eine beliebige Anzahl an Knochen gegeben werden. Diese fungieren wie Gelenke und können anschließend in den Keyframes zu den jeweiligen Zeitpunkten beliebig verändert werden.

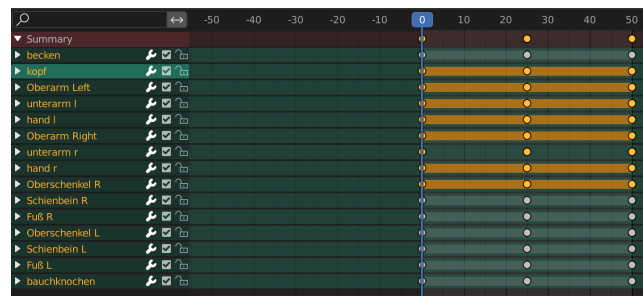


Abbildung 3.5: Keyframes Animation. Quelle: Eigene Aufnahme

Nach dem Exportieren des fertigen Objekts werden in Unity mithilfe von einem Animator die Animationen abgespielt. Für das gleichzeitige Abspielen der 4 Reifen wurde auf einen Blend-Tree zurückgegriffen. Dieser hat es ermöglicht, dass mit seinen festzulegenden Variablen alle Animationen eines Objektes zur selben Zeit abgespielt werden.

3.4.2 Implementierung

Beide Autos fahren auf der in Abbildung 2.1 unter Punkt 8 befindlichen Station Straße. Gelangt der Spieler zu dieser Station steht er vor einer Frage und muss diese aus 3 Antwortmöglichkeiten richtig beantworten. Wenn diese erfolgreich beantwortet wird, erhält der Spieler ein weiteres Blatt als Belohnung.

Als kleines Easter-Egg wurde in den Pickup das Asset des Professors Pirkel als Fahrer eingesetzt. Blickt man auf die Straße erkennt man, wie der Fahrer an einem vorbeifährt und dabei aus dem Auto winkt. Das Asset wurde netterweise von Dominik Schweiger zur Verfügung gestellt. Dieser ist angehöriger der Gruppe, die dieses für eine Projektarbeit erstellt haben.

3.5 Station 4: Zaun

Um das Blatt bei dieser Station zu erlangen, muss der Igel der im Zaun feststeckt befreit werden. Dies geschieht, indem der Spieler den Igel mit seinem Controller herauszieht. Für diese 3 unterschiedlichen Zustände existieren 3 Animationen. Abhängig von der Aktion muss eine von diesen ausgeführt werden. Die Schnittstelle zu dem Aufrufen der Aktionen sind in einem Skript implementiert worden. Johannes Horst musste nur noch eine der Funktionen abhängig von seiner ausgeführten Aktion mit dem Controller ausführen.

3.6 Spielende

Für die Registrierung des Spielendes wird ein Event ausgelöst, wenn alle 8 Blätter eingesammelt wurden. Dieses Event triggert eine Funktion und setzt ein Flag auf True. Mithilfe der Basisklasse der Station (Skript von Stefanie Hofmann entwickelt) kann die Position des Spielers abgefragt werden. Wenn dieser sich nach erfolgreichen Beenden aller Aufgaben im Haus wieder einfindet, ist das Spiel geschafft und die Szene wechselt zu den Credits.

Kapitel 4

Zusammenfassung und Ausblick

4.1 Zusammenfassung

Die Entwicklung eines Spiels ist ein komplexer Prozess. Der Entwurf bezieht nicht nur die Implementierung der Logik mit ein, sondern im Laufe der Entwicklung müssen noch weitere Aufgaben bearbeitet werden. Somit musste für das Implementieren der Logik die Programmiersprache C# erlernt werden. Für das Erstellen der Objekte habe ich einen Einblick in die 3D-Modellierung mit Blender bekommen. Besonders der Entwicklungsprozess der Gegenstände beansprucht abhängig vom Detailgrad der Modellierung ein hohes Maß an Zeit. Durch das Verwenden von Soundeffekten musste man sich zudem noch mit Lizenzen auseinandersetzen.

4.2 Ausblick

Das Spiel lässt sich in vielerlei Hinsicht weiterentwickeln. Zum einen können weitere Stationen implementiert werden, welche das Leben eines Igels noch weiter vertiefen oder die bereits bestehenden Stationen noch verbessert werden. Somit kann die Station des Rasenmähers auf die gesamte Spielwelt ausgeweitet werden und der Roboter mit einem Pathfinding Algorithmus ausgestattet werden. Dieser Algorithmus ermöglicht es den Roboter, unabhängig von der Position des Spielers, jagt auf diesen zu machen. Der Anwender müsste dann auf den Roboter reagieren und sich vor dem Objekt verstecken. Die Station der Straße könnte durch Collider erweitert werden, um einen Aufprall mit dem Igel zu registrieren. Zum erfolgreichen Abschließen der Station müsste es der Spieler auf die gegenüberliegende

Seite der Straße schaffen. Daraus könnte sich eine Art Minigame entwickeln, dass den Autos ausgewichen werden muss. Außerdem könnte das Spiel um weitere Level mit anderen Umgebungen erweitert werden. In dem Umfang, dass man sich zum einen in der Stadt zurechtfinden muss und zum anderen auf dem Land.

Literaturverzeichnis

- [1] *Creative Commons - CC0 1.0 Universal*. <https://creativecommons.org/publicdomain/zero/1.0/>, Abruf: 23. Januar 2022
- [2] *Freesound - Webseite*. <https://freesound.org>, Abruf: 29. Januar 2022
- [3] *GitHub*. <https://github.com/JohannesH1998/AR-VR-Igelschutz>, Abruf: 22. Januar 2022
- [4] *Igel - Beliebter Stachelritter liebt die Stadt*. <https://www.deutschewildtierstiftung.de/wildtiere/igel>, Abruf: 23. Januar 2022
- [5] *Igel überwintern: Die Igelrettungsstation in Fürth*. <https://www.youtube.com/watch?v=vP9LcVgXYTE>, Abruf: 23. Januar 2022
- [6] *OpenGameArt - Webseite*. <https://opengameart.org>, Abruf: 29. Januar 2022
- [7] *Pro Igel E. V.* <https://www.pro-igel.de>, Abruf: 23. Januar 2022
- [8] *Trello*. <https://trello.com>, Abruf: 22. Januar 2022
- [9] *Unity*. <https://unity.com>, Abruf: 22. Januar 2022
- [10] GLOBKE, Dipl.-Inform. W.: *Polynome im Einsatz: Bezier-Kurven im CAD*. <https://www.math.kit.edu/iag2/~globke/media/bezier.pdf>. – Vorlesungsskript
- [11] ZOTOV, Alexander: *Unity Tutorial How To Make Game Object Or Character Move Along Bezier Curve*. <https://www.youtube.com/watch?v=11ofnL0E8pw&t=169s>, Abruf: 23. Januar 2022

Abbildungsverzeichnis

2.1	Übersicht über die Spielkarte	3
2.2	Darstellung der Größenunterschiede	4
3.1	Übersicht über die Rasenmäher Station	7
3.2	Darstellung des Problems der unebenen Fläche	8
3.3	Entwicklungsprozess des Autos mit Blender	9
3.4	Entwicklungsprozess des Pickups mit Blender	9
3.5	Keyframes Animation	10