



universität
wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

Virtual Driving School: Eine Fahrsimulation in VR

verfasst von / submitted by

Ali Sina Iklil BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2020 / Vienna 2020

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

UA 066 921

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Informatik

Betreut von / Supervisor:

Univ.-Prof. Dipl.-Ing. Dr. Helmut Hlavacs

Abstract

Virtual Driving School ist ein Spiel, welches dem Spieler ermöglicht Autofahren zu erlernen und/oder seine Fahrfähigkeiten zu verbessern. Das Spiel wurde mit Unity entwickelt und es unterstützt die Oculus Rift S, ein VR-Headset und es kann ebenfalls das Logitech G29, ein Gaming-Lenkrad angeschlossen werden. Das Spiel kann jedoch auch mit Maus und Tastatur über den Bildschirm gespielt werden. Im Spiel kann der Spieler insgesamt 8 Lektionen absolvieren, jedoch kann der Spieler auch frei in der Spielwelt umherfahren. Dabei kann der Spieler mit einer automatischen oder manuellen Gangschaltung fahren. Die Spielwelt besteht aus zwei größeren Städten und zwei Dörfern und es gibt Fußgänger und Autofahrer, welche sich realistisch verhalten. Ebenfalls gibt es im Spiel Verkehrsschilder und neben Ortsstraßen gibt es auch Autobahnen und Freilandstraßen.

Inhaltsverzeichnis

1	EINLEITUNG	6
2	RELATED WORK.....	9
2.1	Virtual Reality	9
2.2	Einsatzfelder von VR zum Lernen	10
2.3	Fahrsimulation für Autonomes Fahren.....	11
2.4	Intelligentes Tutoring bei Fahrsimulation	12
2.5	Simulator für Menschen mit Rückenmarksverletzung	13
2.6	Auswirkung von Sicht und Eingabegerät	14
2.7	Barrieren und Potenzial im Lernen mit VR.....	15
2.8	Fahrsimulator in einem fahrenden Auto.....	16
2.9	Project Westdrive	17
3	IMPLEMENTIERUNG.....	18
3.1	Unity	18
3.2	Spielerauto	18
3.2.1	Armaturenbrett	19
3.2.2	Antrieb.....	20
3.2.3	Manuelle Gangschaltung.....	20
3.2.4	Automatische Gangschaltung.....	21
3.2.5	Lenkrad.....	21
3.2.6	Spiegel.....	21
3.2.7	Lichter	21
3.2.8	Weitere Features.....	22
3.3	Oculus Rift S	22
3.4	Logitech G29	23
3.5	Spielwelt	24
3.6	Verkehrsschilder	27
3.7	Fußgänger	27
3.7.1	Wegpunktsystem für Fußgänger	29
3.7.2	Pooling System für Fußgänger	30
3.8	Autofahrer.....	31
3.8.1	Wegpunktsystem für Autofahrer	31
3.8.2	Pooling System für Autofahrer	34
3.9	Menüs	34
3.9.1	Hauptmenü	34

3.9.2	Lektionen.....	35
3.9.3	Lektion Anfahren und Schalten.....	36
3.9.4	Lektion Parallelparken	37
3.9.5	Lektion Rückwärts einparken.....	37
3.9.6	Lektion Umkehren.....	38
3.9.7	Lektion Rechtsabbiegen	39
3.9.8	Lektion Linksabbiegen	39
3.9.9	Lektion Kreisverkehr.....	40
3.9.10	Lektion Spurwechsel	41
3.9.11	Steuerungsmenü	41
3.9.12	Pausemenü (Freie Fahrt)	42
3.9.13	Pausemenü (Lektion).....	43
3.9.14	Optionsmenü	43
3.10	Skripts.....	44
3.10.1	PlayerCar	44
3.10.2	Pedestrian	46
3.10.3	NPC_Car	46
3.10.4	Lights.....	47
3.10.5	Menu.....	47
4	PERFORMANCE.....	49
4.1	System	49
4.2	Methodik.....	49
4.3	Ergebnisse.....	50
4.3.1	Normaler Modus mit verschiedenen Grafikqualitäten	50
4.3.2	VR Modus mit verschiedenen Grafikqualitäten.....	54
4.3.3	Verschiedene Fußgängeranzahlen.....	55
4.3.4	Verschiedene Fahrzeuganzahlen	56
4.3.5	Verschiedene Sichtweiten	57
5	EVALUIERUNG.....	58
5.1	Ergebnisse.....	61
5.1.1	Allgemeine Fragen	61
5.1.2	Lektionen.....	61
5.1.3	Freie Fahrt Modus	62
5.1.4	Sonstige Fragen	62
5.1.5	Virtuelle Realität Fragen (falls VR getestet wurde).....	63

5.1.6	Logitech G29 Lenkrad Fragen (falls es getestet wurde)	63
6	DISKUSSION	65
6.1	Motion Feedback	65
6.2	Intelligent Tutoring System	65
6.3	Gefahren beim Autofahren in VR	66
6.4	Performance bei Virtual Reality	66
6.5	Nachtmodus	66
6.6	Wetter	67
6.7	Mehrere fahrbare Autos	67
6.8	Mehr Lektionen	68
7	ZUSAMMENFASSUNG	69
8	REFERENZEN	70
9	ANHANG	72

1 EINLEITUNG

Bei Virtual Driving School handelt es sich um eine Autofahrsimulation. Es soll dabei helfen Autofahren zu erlernen oder seine Fahrfähigkeiten zu verbessern. Da Fahrstunden in einer richtigen Fahrschule oftmals teuer sind, kann eine Fahrsimulation dabei helfen Geld zu sparen und das Risiko ist geringer als bei einem richtigen Auto. Als Fahranfänger kann man sich im Spiel bestimmte Grundlagen aneignen. Zum Beispiel kann man lernen, wie man ein Auto startet, wie man den Gang wechselt, wie man einparkt, wie man die Spur wechselt, wie man umkehrt, wie man links oder rechts in einer Kreuzung einbiegt und wie man sich im Kreisverkehr verhält. Das Spiel ist möglichst realistisch gehalten. Zum Beispiel muss man wie bei einem echten Auto mit manueller Schaltung beim Wechseln des Gangs die Kupplung gedrückt halten und darf sie nicht zu schnell loslassen beim Anfahren, damit der Motor nicht abstirbt. Es gibt jedoch auch die Option mit automatischer Gangschaltung zu fahren, also so, dass man nur beschleunigen und bremsen kann und der Gang automatisch ausgewählt wird. Der Motor kann in diesem Modus auch nicht absterben.



Abbildung 1: First-Person-View in Virtual Driving School

Das Spiel unterstützt die Oculus Rift S und das Logitech G29 Lenkrad mit den Pedalen (Kupplung, Bremse, Gas) und der Gangschaltung kann ebenfalls verwendet werden. Das Spiel kann jedoch auch nur mit Bildschirm und Tastatur sowie Maus gespielt werden. Hierbei wird die Maus zum Umsehen verwendet, während mit der Tastatur gelenkt und gefahren wird. Im Spiel kann man das Auto in der First-Person-Perspektive steuern, in welcher man ein detailliertes Cockpit sieht mit RPM- und Geschwindigkeitsanzeige sowie ein virtuelles Lenkrad, welches immer genauso rotiert ist wie das Logitech G29 Lenkrad, falls es verwendet wird. Ebenfalls kann der Spieler in den Rückspiegel und in die Seitenspiegel sehen. Der Spieler kann auch blinken, hupen, die Handbremse anziehen, das Licht einschalten, die Scheibenwischer aktivieren und die Warnblinkanlage einschalten. Wenn der Spieler bremst, dann leuchten die Bremslichter an der Hinterseite des Fahrzeugs auf und wenn der Spieler blinkt, dann leuchten die Blinker auf. Neben der First-Person-Perspektive kann man auch eine Third-Person-Perspektive verwenden, um sich einen besseren Überblick über das Geschehen zu verschaffen.



Abbildung 2: Third-Person-View in Virtual Driving School

Im Spiel kann der Spieler insgesamt 8 Lektionen auswählen und dadurch die Grundlagen des Autofahrens erlernen, jedoch gibt es auch einen Freie-Fahrt-Modus, wo der Spieler frei in der Spielwelt fahren kann. In der Spielwelt gibt es zwei größere Städte und zwei kleinere Dörfer, welche mit Freilandstraßen und einer Autobahn miteinander verbunden sind. Ebenso gibt es Verkehrsschilder sowie Ampeln für Fußgänger und Ampeln für Fahrzeuge. Die Spielwelt wird nicht zufallsgeneriert, sondern wurde manuell erstellt und bleibt somit gleich bei jedem Neustart des Spiels. Eine zufallsgenerierte Spielwelt hätte den Nachteil gehabt, dass die Spielwelt sich schnell wiederholen würde, wenn zu wenige Kacheln bzw. Tiles verwendet werden. Daher wurde entschieden, dass eine manuell erstellte Spielwelt besser geeignet ist, gerade auch weil die Spielwelt sehr heterogen ist, da es nicht nur Stadtgebiete gibt, sondern auch Autobahnen und Freilandstraßen.

Im Spiel gibt es, wie bereits erwähnt, Fußgänger und Fahrzeuge. Diese bewegen sich im Spiel mithilfe eines Wegpunktsystems. Damit die Performance des Spiels nicht zu stark leidet an zu vielen Fußgängern und Fahrzeugen, wurde ein Pooling-System implementiert. Das heißt, dass Fußgänger und Fahrzeuge ab einer bestimmten Entfernung zum Spieler, neu an einen Wegpunkt in der Nähe des Spielers platziert werden. Dabei wird jedoch darauf geachtet, dass der Spieler dies nicht bemerkt. Ebenfalls wird im Spiel bei vielen Objekten LOD (Level of Detail) eingesetzt, um die Performance zu verbessern. Das heißt, dass manche Objekte verschiedene Detailgrade besitzen und je nach Entfernung des Objekts zum Spieler wird ein passender Detailgrad ausgewählt. Ebenfalls ist das in Unity eingebaute Occlusion Culling aktiviert, um die Performance zu steigern. Dabei werden Objekte, die durch andere Objekte verdeckt werden, nicht gerendert, um die Performance zu verbessern.



Abbildung 3: Gesamtes Setup von Virtual Driving School

Die Anzahl der Fußgänger und Autofahrer kann im Optionsmenü angepasst werden. Dies ist wichtig, damit Spieler mit einem schwachen PC das Spiel trotzdem mit einer angenehmen Anzahl an Bildern pro Sekunde bzw. FPS (frames per second) spielen können. In den Optionen kann ebenfalls die Auflösung des Spiels sowie die Grafikqualität und die Sichtweite eingestellt werden. Auch kann im Optionsmenü ausgewählt werden ob zur Steuerung das Logitech G29 verwendet wird oder die Tastatur mit der Maus. Der Spieler kann ebenfalls einstellen ob er mit einer manuellen oder automatischen Gangschaltung fahren möchte. Für das Spiel wurde eine VR-Version erstellt und eine „normale“ Version, bei der kein VR verwendet wird. Zuerst wurde versucht beide Möglichkeiten in einen gemeinsamen Build einzubauen, sodass der Spieler in den Optionen einstellen kann, ob die Ausgabe über das VR-Headset oder über den Bildschirm erfolgen soll. Dies hatte jedoch zu Problemen geführt, da Unity ein Wechsel zwischen „normalem“ Modus und VR-Modus zur Laufzeit nicht angedacht hat bzw. nur schlecht unterstützt.

Im nächsten Kapitel „Related Work“ werden Arbeiten vorgestellt, die in Relation zu dieser Arbeit stehen und einige Erkenntnisse liefern in Bezug auf Virtual Reality, Lernen und Fahrsimulationen. Dann wird im dritten Kapitel die Implementierung von Virtual Driving School im Detail beschrieben und wichtige Designentscheidungen werden begründet. Im vierten Kapitel wird die Performance des Spiels getestet und wie diese sich ändert bei unterschiedlichen Grafikqualitäten, oder wenn VR verwendet wird oder wenn die Anzahl der Fußgänger und Fahrzeuge erhöht wird. Im fünften Kapitel werden die Methodik und die Ergebnisse der Evaluierung des Spiels beschrieben. Im sechsten Kapitel wird diskutiert, wie das Projekt verbessert werden könnte und im siebenten Kapitel folgt dann eine Zusammenfassung der wichtigsten Erkenntnisse.

2 RELATED WORK

2.1 Virtual Reality

Mit Virtual Reality bzw. VR bezeichnet man eine computergenerierte, meist interaktive virtuelle Umgebung. Von VR zu unterscheiden ist Augmented Reality bzw. AR, wo die tatsächliche Realität erweitert wird. Für VR wird ein sogenanntes VR-Headset bzw. eine VR-Brille verwendet, welche auf den Kopf aufgesetzt wird. Das Gerät erkennt die Bewegungen des Kopfes, wodurch man sich in der virtuellen Realität umsehen kann. Es gibt VR-Headsets, welche nur 3 Freiheitsgrade (bzw. 3 DOF) besitzen bzw. erkennen, sodass man nur den Kopf in der virtuellen Realität drehen aber nicht im Raum bewegen kann [1]. Ein Beispiel dafür ist die Oculus Go [1]. Es gibt aber auch VR-Headsets mit 6 Freiheitsgraden, wo die Rotation und Translation des Kopfes erkannt wird, wie z.B. bei der Oculus Rift S [1].

Bei fast allen VR-Headsets gibt es auch einen oder zwei Controller, welche mitgeliefert werden, mit denen man in der Spielwelt mit Objekten interagieren kann. Es gibt einfache Controller mit wenigen Buttons, mit denen man nur in eine Richtung zeigen kann, da diese nur 3 Freiheitsgrade besitzen, wie z.B. beim Controller der Oculus Go [1]. Es gibt aber auch komplexere Controller mit 6 Freiheitsgraden, welche einige Buttons besitzen und beide Hände in der virtuellen Realität abbilden, wie z.B. bei der Oculus Rift S [1]. In Fahrsimulationen kommen die Controller eines VR-Headsets oft nicht zum Einsatz, da man ein Lenkrad verwendet oder Maus und Tastatur.

Ebenfalls wichtig ist die Unterscheidung zwischen PCVR und mobilem VR, auch bekannt als Standalone-VR oder All-In-One-VR. Bei PCVR-Headsets muss ein Computer verwendet werden, auf welchem das Spiel gerendert wird, während bei mobilen VR-Headsets kein Computer benötigt wird. Außerdem sind PCVR-Headsets meist kabelgebunden, während mobiles VR kabellos ist, jedoch gibt es Möglichkeiten mit einem mobilen VR-Headset ein PCVR-Spiel zu spielen mit z.B. ALVR (Air Light VR) [2] oder Riftcat VRidge [3]. Bei beiden handelt es sich um Programme, welche ein PCVR-Spiel auf das mobile VR-Headset streamen und Eingaben vom mobilen VR-Headset entgegennehmen. Das Kabel bei kabelgebundenen VR-Headsets kann die Bewegungsfreiheit einschränken, selbst wenn das Kabel nicht am Boden liegt, sondern über die Decke geführt wird [4]. Daher sind oft kabellose VR-Headsets angenehmer, speziell in Spielen, wo der Spieler sich viel bewegen und drehen muss. Im Vergleich dazu stört das Kabel bei kabelgebundenen VR-Headsets weniger in Spielen, in welchen der Spieler sitzt, wie z.B. bei einer Fahrsimulation.



Abbildung 4: Oculus Rift S (ein PCVR-Headset) mit beiden Controllern

Die Systemanforderungen bei VR sind höher als bei Spielen, welche auf einem „flachen“ Bildschirm gespielt werden, da das Bild für beide Augen gerendert werden muss [5] und bei VR meist mehr FPS (frames per second) benötigt werden, da man sonst leichter Motion Sickness bzw. Bewegungskrankheit bekommen kann. Motion Sickness ist ein Phänomen, welches auftritt, wenn verschiedene Sinnesorgane widersprüchliche Informationen an das Gehirn liefern [4]. In vielen VR-Spielen gibt es die Möglichkeit sich über Teleportation zu bewegen, jedoch bevorzugen Spieler es oft, sich über einen der beiden Analogsticks (der Controller) in der virtuellen Welt fortzubewegen, obwohl dies leichter zu Motion Sickness führen kann [6]. Jedoch kann man mit der Zeit eine gewisse Toleranz aufbauen, sodass man länger spielen kann ohne Symptome wie Schwindel oder Übelkeit zu bekommen [6]. Das Aufbauen einer solchen Toleranz bezeichnen Spieler oft als „getting VR legs“ [6].

Mit VR-Headsets können nicht nur Spiele gespielt werden, sondern oft kann man auch im Internet surfen oder 3D-Videos oder normale Videos auf YouTube ansehen. Dabei zeigt man mit dem Controller, um zu klicken wie mit einer Maus und für Tastatureingaben gibt es eine virtuelle Tastatur, wo man auf die gewünschten Buchstaben zeigt und klickt.

Es gibt Spiele, welche (von offizieller Seite) nur mit VR-Headsets gespielt werden können, wie z.B. Half-Life: Alyx [7], jedoch gibt es sehr viele Spiele, welche einen VR-Modus unterstützen, jedoch auch ohne VR gespielt werden können wie z.B. City Car Driving [8]. Oft können auch ältere Spiele so modifiziert werden, dass man sie mit einem VR-Headset spielen kann.

Bei VR-Headsets gibt es neben den teureren Modellen wie der Oculus Quest, Oculus Rift S, HTC Vive oder Valve Index auch sehr günstige Preisklassen. Eine sehr preiswerte Möglichkeit VR zu erleben ist über ein sogenanntes Cardboard [9]. Dabei steckt man sein Smartphone in das Cardboard hinein, setzt dieses auf den Kopf auf und sieht sich z.B. ein 3D-Video auf YouTube an. Es gibt sehr günstige Cardboards welche aus Karton bestehen (und sogar selbst gebaut werden können) [9], aber auch welche die etwas aufwendiger gestaltet sind.

Bei VR muss man ebenfalls den sozialen Aspekt miteinbeziehen. Besonders bei mobilen VR-Headsets kann in normalen Umständen, nur die Person, welche das VR-Headset aufgesetzt hat, sehen was gerade passiert. Personen, die dabei sind, können oft leider nur fragen was in der virtuellen Realität gerade geschieht, was für Langeweile sorgen kann. Bei PCVR-Headsets ist es jedoch meist so, dass die virtuelle Realität auch auf dem Computerbildschirm gesehen werden kann. Das heißt wenn man jemandem zusehen möchte, der ein VR-Headset aufgesetzt hat, ist dies bei PCVR-Headsets leichter.

2.2 Einsatzfelder von VR zum Lernen

Virtual Reality kann in sehr vielen Feldern zum Lernen eingesetzt werden. In [10] konnten 4 praktische Vorteile von Lernen in VR identifiziert werden. Erstens können abstrakte Themen greifbarer gemacht werden [10]. Zum Beispiel kann der Aufbau und die Funktionsweise einer biologischen Zelle in VR leichter verstanden werden [10]. Zweitens hat VR den Vorteil, dass VR meist interaktiv ist im Vergleich zum passiven Observieren [10]. Drittens erlaubt VR Methoden die wünschenswert sind, jedoch in der Praxis zu aufwendig wären [10]. Zum Beispiel kann ein Lehrer in der Realität kaum historisch wichtige Orte auf der ganzen Welt mit seiner Klasse besuchen, in VR jedoch ist dies möglich [10]. Viertens erlaubt VR die Grenzen der Realität zu

sprengen und z.B. zu testen, was passieren würde, wenn man die Schwerkraft oder die Lichtgeschwindigkeit ändern würde [10]. In [11] wurde z.B. eine VR Fahrsimulation verwendet, um das Wissen der Schüler in technischen und physikalischen Themen zu verbessern. Die Schüler haben ein hohes Maß an Enthusiasmus gezeigt beim Lernen in VR und wurden ebenfalls motiviert selber STEM Karrieren (Science, Technology, Engineering und Mathematics) anzustreben [11].

Ebenfalls wird VR immer häufiger eingesetzt, um Feuerwehrmänner zu trainieren [12]. Der Vorteil beim Einsatz von VR in diesem Kontext ist, dass die Gefahren und die Kosten niedriger sind beim Üben von komplexen Gefahrensituationen [12]. Dies deckt sich mit den Erkenntnissen von [10], in dem empfohlen wurde, VR einzusetzen beim Erlernen von prozeduralem Wissen, also beim Üben von Bewegungsabläufen. In [13] wurde VR zum Beispiel eingesetzt, um Benutzern beizubringen, wie man einen PC zusammenbauen kann. Dabei muss der Benutzer virtuelle Komponenten auswählen und eine Balance zwischen dem Preis und den Spezifikationen des Systems finden [13]. Der Vorteil hierbei ist, dass man in der virtuellen Realität Fehler machen kann, welche in der Realität sehr teuer werden könnten [13].

2.3 Fahrsimulation für Autonomes Fahren

Fahrsimulationen die Virtual Reality einsetzen, werden auch verwendet, um Forschung im autonomen Fahren zu betreiben, da es weniger Gefahren gibt und der Aufwand niedriger sein kann [14]. In [14] wurde versucht eine möglichst immersive Fahrsimulation zu realisieren, indem VR in Verbund mit einem richtigen Auto eingesetzt wurde. Dazu wurde ein Audi S8 umgebaut und alle nicht notwendigen Teile wie z.B. der Motor und die Reifen wurden entfernt [14]. Ebenfalls wurden 4 Aktuatoren an den Ecken des Fahrzeuges installiert (wo vorher die Reifen waren), um Brems- und Beschleunigungskräfte zu simulieren [14]. An den Pedalen wurden Sensoren installiert, welche die ausgeübte Kraft messen und die Informationen an die Simulation senden [14]. Ebenfalls wurde Hand Tracking eingesetzt, damit erkannt wird, wo sich die Hände des Spielers befinden [14]. Um ein möglichst immersives Erlebnis zu erschaffen wurde ein 8x8km Areal im Bundesland Baden-Württemberg modelliert [14].



Abbildung 5: Gesamtkonzept der Fahrsimulation [14]

2.4 Intelligentes Tutoring bei Fahrsimulation

In [15] wurde eine Fahrsimulation erstellt, in der ein „Intelligent Tutoring System“ (ITS) das Fahrverhalten des Benutzers analysiert und ihm einen optimalen Weg durch die virtuelle Stadt vorschlägt, damit der Fahrer seine Fahrfähigkeiten verbessern kann. Um die physikalischen Kräfte während des Fahrens zu simulieren wurde eine Thruxim Pro 6 DOF Motion Plattform verwendet, welche Translation und Rotation in alle drei Richtungen erlaubt [15]. Ebenfalls wurde ein Force Feedback Lenkrad mit Pedalen und Gangschaltung verwendet und die Ausgabe erfolgte über drei Bildschirme oder über eine HTC Vive [15]. Die Bewegung der Motion Plattform muss vom VR-Headset subtrahiert werden, da sonst Bewegungen der Motion Plattform als Bewegungen des Kopfes gesehen werden würden [15].



Abbildung 6: Cockpit Mockup auf der Motion Plattform [15]

Als Game-Engine wurde Unity eingesetzt und ein aufwendiges Cockpit mit Geschwindigkeits- und RPM-Anzeige (siehe Abbildung 7) sowie Kontrollleuchten und Rück- und Seitenspiegel wurde manuell erstellt [15]. Für die Rück- und Seitenspiegel wurden Kameras vor den Spiegeln platziert, die nach hinten sehen und die Ausgabe der Kameras wird über Render Textures auf den Spiegeln dargestellt [15]. Um die Stadt zu erstellen wurde eine Software namens CityEngine verwendet und die fertige Stadt beinhaltet 9 Kilometer Straße, 519 Gebäude und 40 Autos, welche selbstständig durch die Stadt fahren können [15]. Für computergesteuerte bzw. NPC-Autos (non player character) mussten folgende Regeln implementiert werden: Autos dürfen nicht schneller als erlaubt fahren, Autos müssen genug Abstand zum Vorderauto halten, Autos müssen Vorrangregeln einhalten, Autos müssen bei Stoppschildern stoppen, Autos müssen bei rotem Licht warten [15].



Abbildung 7: Innenansicht im Auto [15]

In [15] wurde für das Intelligent Tutoring System (ITS) definiert welche Merkmale einen guten Fahrer ausmachen und wie diese im Spiel geprüft werden können. Zum Beispiel kann ein guter Fahrer spursicher auf geraden oder gekurvten Straßen fahren neben einigen anderen Merkmalen wie z.B., dass ein guter Fahrer bei Stoppschildern stehen bleibt oder beim Einbiegen die richtigen Aktionen in der richtigen Reihenfolge setzt [15]. Je nachdem wie gut der Fahrer fährt, wird ihm vom ITS eine Route vorgeschlagen, welche Situationen enthalten, die ihm helfen seine Fahrfähigkeiten zu verbessern [15]. Für die Evaluierung der Simulation wurde ein Experiment mit 17 Personen durchgeführt, wovon 4 aufgrund von Simulator Sickness vorzeitig abbrechen mussten, jedoch konnten die meisten Personen sich in der Simulation präsent fühlen und das Auto gekonnt durch die Stadt manövrieren [15].

2.5 Simulator für Menschen mit Rückenmarksverletzung

In [16] wurde eine Fahrsimulation entwickelt damit Menschen mit Rückenmarksverletzung in einer sicheren Umgebung Autofahren lernen können bzw. ihre Fahrfähigkeiten verbessern können. Dazu wird kein VR-Headset verwendet, sondern die Computer Assisted Rehabilitation Environment (CAREN) an der Universität in Südflorida, welche über einen 180° Projektionsschirm und eine Motion Base verfügt (siehe Abbildung 8), welche Translation und Rotation in alle Richtungen unterstützt [16]. Als Steuerung wurde ein Hebel verwendet, mit welchem Gas gegeben oder gebremst werden konnte, je nachdem in welche Richtung der Hebel wie weit gedrückt wurde und ein kleines Lenkrad, welches nur wenig Kraft zum Drehen benötigt [16]. Da Menschen mit Rückenmarksverletzung oftmals ihre Beine nicht oder nur eingeschränkt bewegen können, können zum Autofahren nur die Hände verwendet werden. Es gibt bei der Simulation auch die Möglichkeit ein normalgroßes Gaming Lenkrad mit Pedalen und Gangschaltung zu verwenden, um beide Steuerungsarten miteinander vergleichen zu können [16]. Mithilfe von Google SketchUp 3D wurde eine Highway Scene und eine City Scene erstellt mit Ampeln, Autos, Stoppschildern und Gebäuden [16]. Für die Evaluierung haben 5 gesunde Personen beide Steuerungsarten in beiden Scenes getestet [16]. Drei der 5 Subjekte hatten keine Symptome von Motion Sickness und die Simulation wurde gut bewertet, jedoch haben die Subjekte das Bremspedal als zu unempfindlich empfunden und die Steuerung für Menschen mit Rückenmarksverletzung war am Anfang ungewohnt [16].



Abbildung 8: Das CAREN System [16]

2.6 Auswirkung von Sicht und Eingabegerät

In [17] wurde untersucht wie sich verschiedene Sichten bzw. Views und Eingabegeräte auswirken auf das Fahrverhalten von Spielern. Mithilfe von Python wurde ein simples Rennspiel entwickelt mit einer 360m langen S-förmigen Rennstrecke [17]. Insgesamt wurden in [17] drei Studien durchgeführt, um zu sehen, wie schnell die Partizipanten die Rennstrecke beenden konnten mit verschiedenen Views und Eingabegeräten. Getestet wurde die Overhead-View, First-Person-View, Third-Person-High-View und eine Third-Person-Low-View (siehe Abbildung 9), um zu sehen bei welcher View die Partizipanten die Rennstrecke am schnellsten durchfahren können [17]. Für die Steuerung wurde entweder ein Gaming Lenkrad, eine Maus oder ein Thumbstick verwendet [17]. Bei der Overhead-View hatten die Partizipanten die schlechteste Performance, während bei der First-Person und Third-Person-View die Performance etwa gleich war [17]. Ebenfalls haben die Personen den Thumbstick und das Lenkrad der Maus gegenüber bevorzugt [17]. Bei der First-Person-View sollte erwähnt werden, dass es einen großen Unterschied machen kann, ob die Kamera vor der Windschutzscheibe platziert wird oder ob sich die Kamera im Auto auf dem Fahrersitz befindet. Wenn die Kamera sich vor der Windschutzscheibe befindet, kann der Spieler mehr von der Spielwelt sehen, jedoch kann man nicht mehr das Armaturenbrett samt Lenkrad und Rück- sowie Seitenspiegel sehen. In [17] wird Game Designern empfohlen dem Spieler die Möglichkeit zu geben, sich zwischen First und Third-Person entscheiden zu können.

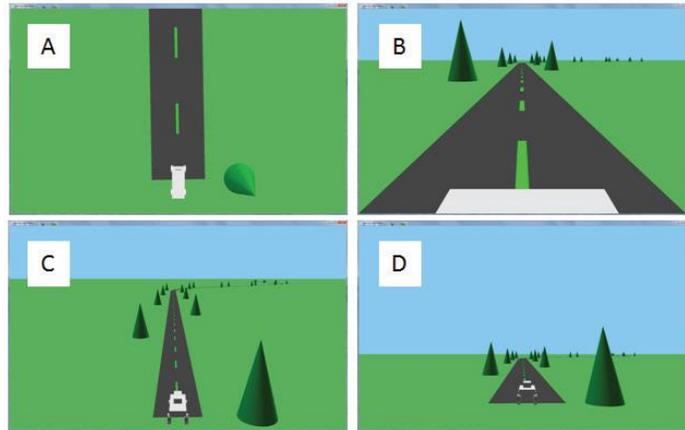


Abbildung 9: A) Overhead, B) First-Person, C) Third-Person-High, D) Third-Person-Low [17]

2.7 Barrieren und Potenzial im Lernen mit VR

In [10] wurde erforscht welche Vor- und Nachteile VR beim Lernen bietet. Dazu wurden 2 Studien durchgeführt: In der ersten Studie mussten die Partizipanten ein VR-Headset aufsetzen und in einer virtuellen Umgebung Briefe und Pakete sortieren und in die richtige Box ablegen [10]. In der zweiten Studie musste eine andere Gruppe von Partizipanten ein VR-Headset aufsetzen und sich ein 360° Video ansehen, welches in MinecraftVR entwickelt wurde und die einzelnen Schritte beim Brauen von Bier erklärt [10]. Die erste Studie war somit interaktiv da die Partizipanten zwei Controller hatten, um die Aufgabe zu lösen, während bei der zweiten Studie die Partizipanten nur eine Maus hatten, um das 360° Video zu starten bzw. zu stoppen [10]. Alle Partizipanten der zweiten Studie meinten, dass sie nicht die Möglichkeit sehen 360° Videos in VR zum Lernen zu verwenden, z.B. da Videos auch auf einem flachen Bildschirm angesehen werden können [10]. Bei 360° Videos gibt es ebenfalls den Nachteil, dass man als Zuseher manchmal nicht weiß wo man hinsehen soll. Zum Beispiel passiert gerade etwas Wichtiges, man schaut jedoch in die falsche Richtung und verpasst es. Ebenfalls ergibt sich die Schwierigkeit, dass Notizen machen schwierig ist, wenn man ein VR-Headset aufgesetzt hat, da man es immer zwischenzeitig absetzen müsste [10].



Abbildung 10: Interaktive Applikation und 360° Video [10]

In [10] wird empfohlen VR eher einzusetzen für implizites Lernen, also Wissen welches man sich unterbewusst aneignet. Ebenfalls empfiehlt [10] VR einzusetzen bei prozeduralem Wissen, also wenn bestimmte Bewegungsabläufe erlernt werden sollen. Außerdem wurde in [10] die Gefahr erkannt, dass man Realität und virtuelle Realität verwechseln könnte, besonders wenn man sich länger in der virtuellen Realität befindet. Zum Beispiel könnte man versuchen sich auf einen virtuellen Stuhl hinzusetzen und sich verletzen [10]. Ebenfalls gibt es bei VR besonders in kleinen Räumen mit vielen Hindernissen das Risiko mit dem Körper an etwas anzukommen und sich zu verletzen. Diese Gefahr kann minimiert werden, indem z.B. das VR-Headset den Spieler einen sogenannten Playspace definieren lässt wie z.B. bei der Oculus Rift S. Wenn der Spieler diesen sicheren Bereich verlässt, wird er vom VR-Headset gewarnt. Bei Fahrsimulationen gibt es auch das Risiko, dass man z.B. beim richtigen Auto eine andere Gangschaltung verwendet als in der Fahrsimulation, wodurch man beim Fahren in der Realität die Gänge durcheinanderbringen könnte. Ebenfalls könnte man sich in einer Fahrsimulation an die Empfindlichkeit des Lenkrads oder des Kupplungs-, Gas- oder Bremspedal gewöhnen, obwohl diese beim eigenen realen Auto anders eingestellt sind.

2.8 Fahrimulator in einem fahrenden Auto

In [18] wurde eine Fahrsimulation entwickelt, in der man sich mit einem VR-Headset auf dem Beifahrersitz befindet, während das reale Auto durch den Fahrer auf dem Fahrersitz gesteuert wird (siehe Abbildung 11). In der virtuellen Realität sitzt der Beifahrer jedoch auf dem Fahrersitz [18]. Vor dem Beifahrer befindet sich ein Gaming Lenkrad und auf dem VR-Headset wurde ein LEAP Motion Controller befestigt, welcher die Position der beiden Hände des Beifahrers trackt und in der virtuellen Realität repliziert [18]. Die Fahrsimulation wird im Paper als VR-OOM bezeichnet, eine Abkürzung für Virtual Reality On-rOad driving siMulation [18]. Die virtuelle Realität wurde mit Unity erstellt und läuft auf einem Laptop, welches sich während der Simulation auf dem Schoß einer Person auf dem Hintersitz befindet [18]. Durch ein sogenanntes Inertial Measurement Unit (IMU) kann die Orientierung und Bewegung des realen Fahrzeuges gemessen werden, sodass die Lage des Fahrzeuges in der virtuellen Realität berechnet werden kann [18]. Wenn der Beifahrer das Gaming Lenkrad dreht, wird es in der virtuellen Realität ebenfalls gedreht, da es am Laptop angeschlossen ist [18]. In [18] wird die virtuelle Realität ebenfalls auf ein Android Tablet gerendert, welches sich vor dem Fahrer befindet, damit dieser sehen kann, was der Beifahrer in der virtuellen Realität sieht.



Abbildung 11: Konzept von VR-OOM [18]

Dies ist nötig, damit der Fahrer dafür sorgen kann, dass die gefühlten Bewegungskräfte für den Beifahrer mit der virtuellen Realität korrespondieren [18]. Damit die Fahrsimulation gut funktioniert, wird in [18] ein möglichst großes offenes Gelände empfohlen, damit in der Fahrsimulation möglichst viele verschiedene Wege abgefahren werden können. Zur Evaluierung wurden 6 Partizipanten gebeten auf dem Beifahrersitz mit aufgesetztem VR-Headset Platz zu nehmen und das Fahrzeug autonom fahren zu lassen und nur dann einzugreifen, wenn sich in der virtuellen Realität eine gefährliche Situation ergibt [18]. Die Partizipanten wussten natürlich, dass das reale Fahrzeug vom Fahrer gefahren wird, sollten sich jedoch so verhalten als ob das Fahrzeug autonom fährt [18]. Die Partizipanten schätzten die Fahrsimulation als immersiv ein und hatten während der Fahrsimulation keine Beschwerden bezüglich Motion Sickness, jedoch hatten 4 Partizipanten etwa eine Stunde nach dem Experiment ein leichtes unbehagliches Gefühl [18]. Ebenfalls kommentierten manche Partizipanten, dass die Grafik der Fahrsimulation nicht realistisch genug aussieht und ein Partizipant hatte während der Fahrsimulation versucht das Bremspedal runterzudrücken, obwohl es auf dem Beifahrersitz keine Pedale gab, da in der Studie der Fokus auf die Tendenz der Partizipanten gelegt war, nach dem Lenkrad zu greifen [18].

2.9 Project Westdrive

In [19] wurde eine große, modulare Stadt in Unity entwickelt mit bis zu 150 computergesteuerten Autos und bis zu 655 Fußgängern. Das Projekt, welches den Namen Westdrive trägt, ist Open Source und wurde entwickelt, um Forschern zu ermöglichen, Studien für VR durchzuführen, ohne selbst erst eine Stadt bauen zu müssen [19]. Damit die Spielwelt realistisch wirkt, müssen auch sich realistisch verhaltende Fußgänger und Autos implementiert werden. Dabei muss immer darauf geachtet werden, dass die Performance nicht zu stark leidet. In [19] konnte auf einem leistungsstarken System eine Performance von 30 Bildern pro Sekunde erreicht werden, was für VR jedoch zu niedrig sein könnte. Das Ziel der Autoren von [19] ist unter anderem, das Projekt in Zukunft zu optimieren, sodass stabile 90 FPS (frames per second) erreicht werden können.



Abbildung 12: Szene in Project Westdrive [19]

3 IMPLEMENTIERUNG

3.1 Unity

Für die Entwicklung von Virtual Driving School wurde Unity ausgewählt. Unity ist eine sehr bekannte Game-Engine für Spiele aller Art. Mit Unity können Spiele für den PC entwickelt werden, aber auch für den Webbrowser oder für mobile Endgeräte wie z.B. für Android oder iOS. Neben 3D Spielen ist mit Unity auch die Entwicklung von 2D Spielen oder VR- bzw. AR-Spielen möglich. Beim Entwickeln werden eine oder mehrere Szenen erstellt, in die Spielobjekte platziert werden können. An Spielobjekte können Komponenten wie z.B. RigidBodies, Collider etc. angehängt werden aber auch Skripts. Diese können geschrieben werden in C#, Boo oder in JavaScript. Für Virtual Driving School wurde C# verwendet. Ein großer Vorteil bei Unity ist der große Asset Store, in dem sehr viele (oft kostenlose) Assets heruntergeladen werden können. Das Erstellen von Assets bei der Entwicklung von Spielen kann sehr zeitraubend sein. Zum Beispiel würde es sehr lange dauern ein Auto zu modellieren und zu texturieren. Ebenfalls ist das Modellieren und Texturieren von Spielobjekten eine Aufgabe, die künstlerisch sehr anspruchsvoll sein kann und einiges an Übung und Erfahrung erfordert. Es ist deutlich zeitsparender ein fertiges Auto aus dem Asset Store zu nehmen und im eigenen Spiel zu verwenden. Der Großteil der Assets bei Virtual Driving School kommt aus dem Asset Store, also alle Autos, Gebäude und Fußgänger, die im Spiel vorkommen. Einige Assets wie z.B. die Straßen, Gehsteige und Verkehrsschilder wurden jedoch selbst erstellt, da im Asset Store keine passenden Assets gefunden wurden. Ebenfalls wurden viele Assets angepasst bzw. erweitert. Zum Beispiel mussten das Armaturenbrett, die Spiegel, der Ganghebel sowie die Lichter beim Spielerauto selbst implementiert werden. Ohne Verwendung einer mächtigen Game-Engine wie Unity hätte ein Projekt wie Virtual Driving School um einiges mehr Zeit gekostet oder wäre in der Form nicht durchführbar gewesen.



Abbildung 13: Unity Editor

3.2 Spielerauto

Ganz am Anfang der Implementierung wurde im Asset Store nach einem passenden Auto gesucht, welches dann im Spiel gesteuert werden kann. Dabei musste darauf geachtet werden, dass es sich um ein Auto handelt, welches von innen modelliert wurde und von außen sowie innen einen relativ hohen Detailgrad aufweist. Ebenfalls sollte es sich nicht um ein Rennfahrzeug handeln, sondern es sollte ein relativ durchschnittliches Auto sein. Nachdem ein passendes Auto gefunden wurde, wurde dieses aus dem Asset Store heruntergeladen und in das Projekt importiert.

Es gab die Überlegung dem Spieler die Möglichkeit zu geben mehrere Autos selbst zu fahren. Dies hätte jedoch einiges an Arbeit erfordert, da die Spiegel, sowie das Armaturenbrett extra für jedes fahrbare Auto implementiert hätte werden müssen. Ebenfalls wäre es nötig gewesen ein Menü zu erstellen, in welchem der Spieler ein Auto auswählen kann. Daher wurde entschieden, den Fokus bei der Entwicklung auf wichtigere Themen zu legen.



Abbildung 14: Das Spielerauto

3.2.1 Armaturenbrett

Das Armaturenbrett des Spielerautos wurde manuell erstellt und zeigt dem Spieler folgendes an:

- Die Geschwindigkeit des Autos in km/h (Kilometer pro Stunde) auf der rechten Seite
- Die RPM des Motors (revolutions per minute bzw. Umdrehungen pro Minute) auf der linken Seite
- Den gewählten Gang in der Mitte
- Ob die Handbremse momentan betätigt ist
- Ob das Licht gerade eingeschalten ist

Das Armaturenbrett wird ebenfalls in der Third-Person-Perspektive angezeigt. Dies ist nötig, sonst müsste man, wenn man die Third-Person-Perspektive lieber verwendet, ständig in die First-Person-Perspektive wechseln, um z.B. seine momentane Geschwindigkeit zu sehen.



Abbildung 15: Das Armaturenbrett

3.2.2 Antrieb

Das Spielerauto hat einen Hinterradantrieb, also nur die hinteren Räder werden verwendet, um das Auto in Bewegung zu bringen. Ebenfalls werden nur die Hinterräder verwendet, um das Auto zu bremsen. Die Vorderräder werden nur zum Lenken verwendet. Wenn man aufs Gas drückt, dann wird auf die Hinterräder ein Drehmoment ausgeübt. Die Geschwindigkeit des Autos wird in km/h ausgerechnet und über die Anzahl der Umdrehungen pro Minute der Reifen kann über den eingelegten Gang und dessen Übersetzungsverhältnis die RPM des Motors berechnet werden. Bei einem realen Auto geschieht dies eigentlich umgekehrt. Dort wirkt sich das Drücken auf das Gaspedal auf die RPM des Motors aus und durch den eingelegten Gang findet durch das Getriebe eine Übersetzung auf die Drehzahl der Reifen statt. Bei der Implementierung wurde jedoch der erste Weg gewählt, da dies leichter zu bewerkstelligen war und es für den Spieler von außen gleich wirkt. Der Motor macht im Spiel natürlich, wenn er eingeschaltet ist, ein realistisches Geräusch und die Tonhöhe des Motors richtet sich nach der RPM. Das heißt bei niedriger RPM ist der Motorsound sehr tief, während der Motor bei einer hohen RPM aufheult wie bei einem richtigen Auto. Dadurch hat man im Spiel das charakteristische Geräusch, wo der Motor beim Beschleunigen lauter wird und dann beim Hochschalten wieder kurzzeitig leiser wird.

3.2.3 Manuelle Gangschaltung

Das Auto hat insgesamt 8 verschiedene Gänge. Es gibt die Gänge 1-6, den Rückwärtsgang (R) und einen neutralen Gang (N). Der Spieler darf beim Anfahren die Kupplung nicht zu schnell loslassen, sonst stirbt der Motor ab und muss wieder gestartet werden. Ebenfalls muss der Spieler beim Schalten die Kupplung gedrückt halten, sonst stirbt der Motor ab. Wenn der Spieler einen sehr hohen Gang verwendet und zu langsam damit fährt, dann beginnt der Motor zu „ruckeln“. Das bedeutet, dass wie bei einem richtigen Auto, der Motor anfängt stärker zu vibrieren und Geräusche zu machen. Wenn der Spieler noch langsamer fährt, dann kann der Motor auch absterben z.B., wenn im 3. Gang nur mit 15km/h gefahren wird. Ebenfalls sollte man im Spiel wie in einem echten Auto nicht mit einem zu niedrigen Gang zu schnell fahren. Wenn man z.B. im ersten Gang bereits 7000 RPM, also die maximale RPM im Spielerauto erreicht hat, dann kann das Auto nicht weiter beschleunigen und das Motorgeräusch wird sehr laut und hoch sein. Ebenfalls gibt es im Spielerauto einen Ganghebel, welcher manuell hinzugefügt wurde. Dieser ändert seine Orientierung passend zum momentan gewählten Gang.



Abbildung 16: Der Ganghebel

3.2.4 Automatische Gangschaltung

Wenn eine automatische Gangschaltung gewählt wurde, dann wird automatisch ein passender Gang gewählt. Der Spieler kann also nur beschleunigen und bremsen. Wenn der Spieler bremst, während das Auto steht, dann fährt das Auto zurück. Wenn das Auto zurückfährt kann das Auto abgebremst werden indem man aufs Gas drückt. Bei einer automatischen Gangschaltung wird im Spiel keine Kupplung benötigt. Ebenfalls kann der Motor bei einer automatischen Gangschaltung nicht absterben. Das Auto fährt sich mit automatischer Gangschaltung also so, wie man es gewohnt ist von üblichen Rennspielen, wenn eine automatische Schaltung verwendet wird.

3.2.5 Lenkrad

Wenn das Logitech G29 Lenkrad verwendet wird, dann ist das Lenkrad im Spielerauto immer genau gleich rotiert wie das physische Gaming-Lenkrad. Dies ist sehr wichtig, vor allem wenn man die Oculus Rift S verwendet, da man sonst die momentane Rotation des Gaming-Lenkrads aus den Augen verlieren würde, gerade beim Einparken, wo das Lenkrad oft maximal in eine Richtung gedreht wird. Wenn die Tastatur verwendet wird, dann wird das Lenkrad im Spielerauto ebenfalls nach links oder nach rechts gedreht und geht wieder selbstständig zurück in die Mitte, wenn keine Eingabe zum Rechts- oder Linksdrehen des Lenkrads gemacht wird.

3.2.6 Spiegel

Das Auto hat einen Rückspiegel und einen linken sowie rechten Seitenspiegel. Die Spiegel waren nicht vorhanden beim Spielerauto, sondern mussten selbst implementiert werden. Für jeden Spiegel wurde eine eigene Kamera platziert und das Bild wird auf eine Render Texture dargestellt an der Stelle, wo sich der Spiegel befindet. Die Kameras der Spiegel reagieren auf Kopfbewegungen. In der Realität ändert sich auch, was man auf einem Spiegel sieht, wenn man den Kopf bewegt. Dies merkt man jedoch nur wenn man die Oculus Rift S verwendet, da man dabei den Kopf nicht nur drehen, sondern auch im Raum bewegen kann.

3.2.7 Lichter

Beim Spielerauto gibt es mehrere Lichter. Zum einen gibt es die roten Bremslichter an der Hinterseite des Fahrzeugs, welche aufscheinen, wenn der Spieler die Bremse betätigt. Wenn der Spieler rückwärtsfährt, dann leuchten die Bremslichter nicht auf, wie bei einem richtigen Auto, außer natürlich, wenn beim Rückwärtsfahren gebremst wird. Ebenfalls gibt es die rechten und linken Fahrtrichtungsanzeiger bzw. Blinker, welche der Spieler betätigen kann. Dabei wird auch das typische Blinkergeräusch abgespielt. Ebenfalls kann der Spieler das vordere Abblendlicht einschalten, jedoch wurde im Spiel kein Nachtmodus implementiert, da hierfür die Zeit gefehlt hat.

3.2.8 Weitere Features

Im Spielerauto kann ebenfalls die Hupe betätigt werden, jedoch wurde nicht implementiert, dass Fußgänger oder andere Autofahrer darauf reagieren. Ebenfalls kann man beim Spielerauto die Scheibenwischer aktivieren. Dabei handelt es sich jedoch auch nur um ein kosmetisches Feature, da es im Spiel keinen Regen gibt oder allgemein ein Wettersystem. Jedoch könnten eingeschaltete Scheibenwischer dabei helfen, sich an das Fahren mit Scheibenwischern zu gewöhnen, da diese gerade bei Fahranfängern ablenkend wirken können. Im Spiel kann auch die Warnblinkanlage aktiviert werden. Diese Features wurden am Anfang der Implementierung eingebaut, und können das Fahrerlebnis realistischer wirken lassen, auch wenn sie keinen wichtigen Zweck erfüllen. Ebenfalls hat das Spielerauto sowie jedes Auto, welches vom Computer gesteuert wird, ein Nummernschild bzw. Kennzeichen vorne und hinten, wobei sich die Kennzeichen zwischen den Autos unterscheiden.

3.3 Oculus Rift S

Bei Virtual Driving School war sehr früh klar, dass das Spiel VR unterstützen soll, um ein möglichst realistisches Fahrerlebnis zu bieten. Es musste eine Entscheidung gefällt werden, welches VR-Headset verwendet werden soll bei der Implementierung. Zuerst wurde die Oculus Go ins Auge gefasst, da es sich um ein relativ preiswertes VR-Headset handelt, jedoch wurde schnell klar, dass mit einem mobilen VR-Headset die Entwicklung in Unity schwieriger werden würde, da man mithilfe von ALVR oder RiftCat VRidge das Spiel auf das VR-Headset streamen müsste. Ebenfalls hat die Oculus Go den Nachteil, dass die Oculus Go nur die Rotation und keine Bewegung des Kopfes wahrnimmt, da die Oculus Go nur 3 DOF (degrees of freedom) besitzt. Für die Implementierung fiel daher die Entscheidung auf die Oculus Rift S, vor allem, da es sich hierbei um ein PCVR-Headset handelt, welches am PC angeschlossen werden kann und somit sehr leicht verwendet werden kann, um den VR-Modus des Spiels in Unity zu testen. Ebenfalls erkennt die Oculus Rift S auch Bewegungen des Kopfes im Raum und nicht nur die Rotation, da das Gerät 6 Freiheitsgrade besitzt. VR-Headsets mit 6 DOF sind für Fahrsimulationen besser geeignet, da man sich im Auto leichter umsehen kann. In der Realität dreht man beim Autofahren nicht nur den Kopf, sondern bewegt ihn meistens, z.B. wenn man nach hinten sehen möchte beim Rückwärtsfahren. Ein Problem, welches bei Virtual Driving School aufgefallen ist, dass wenn man zurücksehen möchte, die Kopfstütze oft im Weg ist. Dies könnte gelöst werden indem man die Kopfstütze beim Fahrersitz entfernt, jedoch würde die Fahrsimulation dann an Realismus einbüßen, da ein richtiges Auto meist eine Kopfstütze hat. Ein allgemeines Problem bei VR ist, dass man sich durch virtuelle Hindernisse z.B. durch die Fahrertür bewegen kann.



Abbildung 17: Oculus Rift S mit beiden Controllern

Um das Spiel mit der Oculus Rift S zu spielen, muss das Headset eingerichtet werden und die Oculus Applikation installiert werden. Dann muss der VR-Build von Virtual Driving School gestartet werden und entweder eine Lektion oder der Freie-Fahrt-Modus ausgewählt werden. Die beiden mitgelieferten Controller der Oculus Rift S können im Spiel nicht verwendet werden, da die Steuerung nur über das Logitech G29 Lenkrad oder über Maus und Tastatur möglich ist. Eine Steuerung über die beiden Controller wäre zu ungenau gewesen und die Controller haben zu wenige Buttons für das Spiel. Die Controller können jedoch verwendet werden, um die Position des Headsets zurückzusetzen. Wenn z.B. die Oculus Rift S auf dem Tisch nach links orientiert ist, während das Spiel geladen wird, dann wird die Richtung vom Spiel als gerade gesehen. Um dies zu beheben, kann die View mithilfe der Controller gerade ausgerichtet werden.

Für die VR-Version des Spiels braucht man einen leistungsfähigeren PC als für die „normale“ Version des Spiels, da das Bild für beide Augen gerendert wird und man mehr FPS (frames per second) benötigt, um Motion Sickness zu vermeiden. Daher empfiehlt es sich, die Fußgänger- und Fahrzeugdichte sowie die Grafikkqualität niedriger zu setzen, wenn der VR-Modus verwendet wird. Ebenfalls sollte die Sichtweite entweder in den Optionen oder während des Spiels mit F11 verringert oder mit F12 erhöht werden, sodass ein guter Kompromiss zwischen Sichtweite und Performance gefunden werden kann. Neben der Oculus Rift S könnte Virtual Driving School auch mit der Oculus Go, Oculus Quest oder mit der alten Oculus Rift funktionieren, dies wurde jedoch nicht getestet.

3.4 Logitech G29

Neben einem VR-Headset musste ebenfalls entschieden werden, welches Gaming-Lenkrad für das Spiel verwendet werden soll. Von Anfang an war klar, dass neben Gas- und Bremspedal ebenfalls ein Kupplungspedal dabei sein sollte und eine Gangschaltung, damit das Autofahren im Spiel möglichst nah an die Realität herankommt. Dabei wurde das Logitech G29 Lenkrad ausgewählt, auch da Logitech ein Gaming SDK für Unity anbietet, sodass leicht erfasst werden kann wie das Lenkrad ausgerichtet ist, welche Tasten gedrückt werden oder wie tief die Pedalen gedrückt werden.



Abbildung 18: Logitech G29 Lenkrad mit Pedalen und Gangschaltung

Die Gangschaltung sowie die Pedalen werden am Lenkrad angeschlossen und das Lenkrad muss am Strom angeschlossen werden und per USB an den Computer. Das Lenkrad kann verwendet werden für die PS3/PS4 und für den PC. Es gibt ebenfalls das Logitech G920, dabei handelt es sich fast um dasselbe Lenkrad, jedoch kann dieses an die Xbox One und an den PC angeschlossen werden. Das Logitech G29 Lenkrad ist für den PC jedoch vorzuziehen, da es über mehr Buttons verfügt. Für Virtual Driving School wird nur das Logitech G29 Lenkrad unterstützt, da andere Lenkräder von Logitech oder Lenkräder von anderen Herstellern über weniger bzw. andere Buttons verfügen.

Das Logitech G29 Lenkrad unterstützt Force Feedback. Das heißt, dass das Lenkrad je nach Spiel so konfiguriert werden kann, dass es wieder zurück in die Ursprungsposition geht, wobei die Kraft hierfür eingestellt werden kann. Ebenfalls kann das Lenkrad so konfiguriert werden, dass es vibriert, wenn man z.B. über eine raue Oberfläche fährt. Außerdem gibt es die Möglichkeit, dass das Lenkrad abrupt in eine Richtung bewegt wird, wenn man z.B. in einem Spiel mit dem Auto einen Unfall verursacht und die eingelenkten Reifen durch ein Hindernis geradegestellt werden.

3.5 Spielwelt

Die Spielwelt von Virtual Driving School besteht aus einer großen Stadt, einer kleinen Stadt und zwei kleineren Dörfern (siehe Abbildung 19). Die Stadt im Süden wurde Südstadt genannt, die Stadt im Norden wurde Nordstadt genannt, das Dorf im Westen wurde Westdorf genannt und das Dorf im Osten wurde Ostdorf genannt. Die Orte benötigen Namen, da es im Spiel neben anderen Verkehrsschildern auch Ortsschilder gibt, auf denen der Name des Orts stehen muss. Ebenfalls stehen die Ortsnamen auf den Autobahnwegweisern. Die Autobahn verbindet die Südstadt mit der Nordstadt und es gibt etwa in der Mitte der Spielwelt zwei Autobahnauffahrten und zwei Autobahnausfahrten. Neben der Autobahn gibt es auch Freilandstraßen, welche die verschiedenen Orte miteinander verbinden. Das Terrain wurde mithilfe der Unity Terrain Tools erstellt. Das Terrain umfasst Wiesen, Bäume, Wälder, Seen, Hügel und Berge. Bei der Erstellung des Terrains wurde darauf geachtet, dass es möglichst natürlich aussieht. Der Boden ist an den meisten Stellen nicht eben, sondern geht leicht rauf und runter. Der Spieler kann nicht aus der Spielwelt herausfahren, da der Boden an den Grenzen abrupt um ein paar Meter erhöht ist. In vielen anderen

Spielen werden Spielweltgrenzen durch eine unsichtbare Wand realisiert oder dadurch, dass die Spielwelt sich auf einer Insel befindet. Jedoch werden unsichtbare Wände oft als faul angesehen und eine Begrenzung durch Wasser könnte auch problematisch sein, gerade wenn Autos sich unrealistisch verhalten, wenn man ins Wasser fährt.

In der Spielwelt wurden bewusst Höhenunterschiede gesetzt, damit die Spielwelt interessanter wirkt. In der Südstadt gibt es z.B. ein erhöhtes Plateau und manche Freilandstraßen sind an bestimmten Stellen erhöht im Vergleich zum Rest der Spielwelt. Steigungen und Gefälle sind wichtig, da das Befahren solcher auch von Fahranfängern gelernt werden sollte. In den Ortsgebieten wurde versucht möglichst viele verschiedene Gebäude zu platzieren, damit die Spielwelt abwechslungsreich wirkt. Es gibt im Spiel mehrere Hochhäuser, aber auch kleine Einfamilienhäuser sowie größere Gebäude. Ebenfalls wurde extra in der Südstadt und in der Nordstadt ein Fußballplatz kreiert. Ebenso gibt es Parks mit einem See, Bäumen und Blumen, um für mehr Abwechslung zu sorgen. Dadurch, dass die Spielwelt sich nicht stark wiederholt und relativ heterogen ist, kann man sich als Spieler leichter in der Spielwelt orientieren. Bei Straßen und Kreuzungen wurde ebenfalls darauf geachtet möglichst viel Variabilität zu haben. Es gibt Einbahnstraßen, einspurige, zweispurige und dreispurige Straßen (Spuren in beide Richtungen) und dazugehörige Kreuzungen. Ebenfalls wurden in den Ortsgebieten Straßenlampen platziert, wodurch die Orte realer wirken. Es gibt Fahrzeugampeln an Kreuzungen, welche sich realistisch verhalten und dieselben Farbzustände wie eine richtige Fahrzeugampel haben. Ebenfalls gibt es im Spiel Fußgängerampeln. Die Ampeln sind alle so eingestellt, dass sie zur richtigen Zeit den richtigen Farbzustand haben, sodass eine Richtung Grün hat, wenn die Querrichtung Rot hat. In der Südstadt gibt es am westlichen Ende der Stadt einen großen Übungsplatz, an dem die 8 Lektionen stattfinden. Ebenfalls gibt es im Spiel mehrere Tankstellen, um die Spielwelt nochmals etwas realistischer aussehen zu lassen, jedoch können Autos im Spiel nicht getankt werden. Eine heterogene Spielwelt hilft dem Spieler sich leichter zu orientieren, gerade über geographische Fixpunkte wie z.B. Fußballplätze, Parks, Tankstellen, Berge etc. Dies ist gerade in Virtual Driving School wichtig, da im Spiel dem Spieler keine Karte angezeigt wird. Es wurde überlegt im Pausemenü eine Karte der Spielwelt einzubauen und dem Spieler anzuzeigen wo er sich gerade befindet. Dazu hätte jedoch eine Karte erstellt werden müssen, auf der alle Straßen und Gebäude deutlich erkennbar sind und dies wäre zu aufwendig gewesen. Ebenfalls müsste auf der Karte sichtbar sein, wo sich der Spieler befindet, weswegen auf dieses Feature verzichtet wurde. Die Bäume in der Spielwelt wurden ebenfalls mithilfe der Unity Terrain Tools platziert. In der Spielwelt sind großteils einzeln Bäume, jedoch gibt es insgesamt drei Wälder, wo die Baumdichte besonders hoch ist. Die Bäume können in der Abbildung 19 nicht gesehen werden, da diese bei einer sehr großen Entfernung nicht mehr gerendert werden, um die Performance zu verbessern. Die Wälder befinden sich in der Spielwelt (siehe Abbildung 19) dort wo der Boden eine braune Farbe hat. In der Realität ist der Boden in einem Wald eher selten grün, sondern eher erdig.



Abbildung 19: Gesamte Spielwelt von Virtual Driving School

3.6 Verkehrsschilder

In der Spielwelt wurden sehr viele Verkehrsschilder an passenden Stellen platziert. Zum Beispiel wurden Ortsschilder platziert an Stellen, wo ein Ort beginnt bzw. aufhört. Ebenso Autobahnschilder, um anzuzeigen, wo die Autobahn beginnt und aufhört. An Fußgängerübergängen, die nicht geregelt sind, sind ebenfalls Schilder angebracht, die dem Fahrer mitteilen, dass sich dort ein Schutzweg befindet. Ebenso gibt es mehrere Gefahrenzeichen für z.B. scharfe Kurven, Verengungen oder Steigungen und Gefälle. Schilder wie für den Beginn und das Ende einer 30er-Zone sowie für die Geschwindigkeitsbegrenzungen 50 und 70 finden sich ebenfalls in der Spielwelt und werden auch von computergesteuerten Autofahrern eingehalten. Ebenfalls gibt es Vorrangschilder und passend dazu Vorrang-geben-Schilder. Es gibt im Spiel auch mehrere Einbahnen, die mit Einbahnstraßenschildern versehen sind, wobei in die Gegenrichtung auch Einfahrt-verboten-Schilder vorhanden sind. Es gibt im Spiel ebenfalls über den Autobahnen mehrere Schilder wie z.B. Vorwegweiser oder Geschwindigkeitsbegrenzungen. Stoppschilder lassen sich in der Spielwelt auch finden an Straßen, wo der Querverkehr schwer eingesehen werden kann und computergesteuerte Autofahrer halten sich auch an diese.

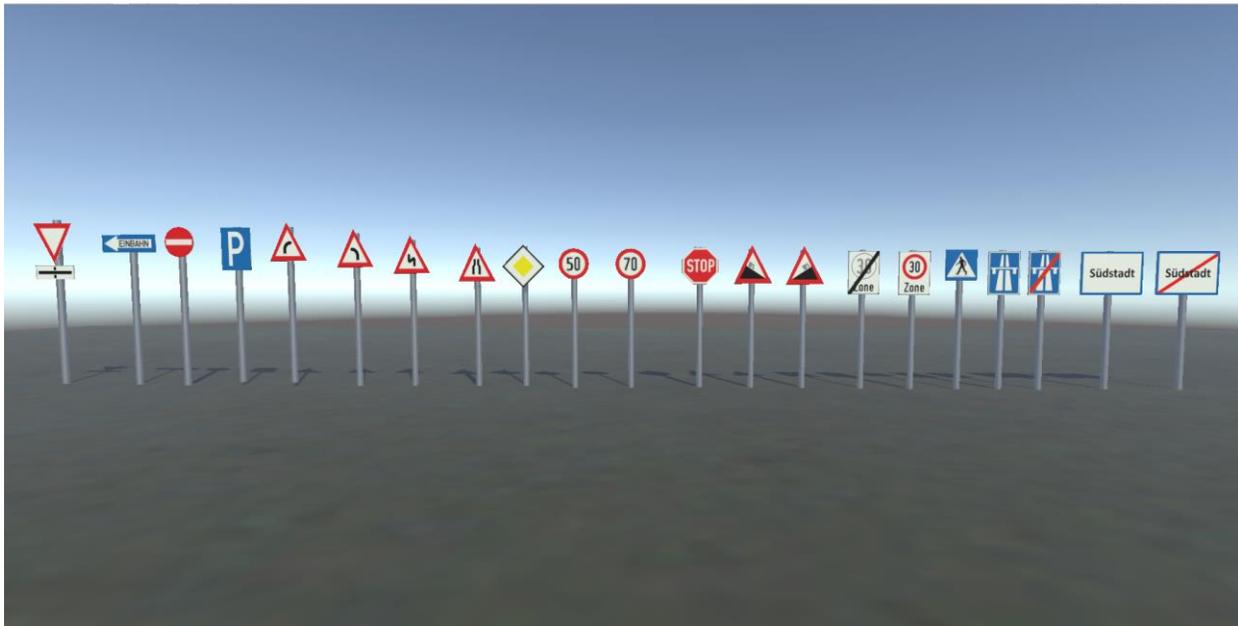


Abbildung 20: Schilder in Virtual Driving School

3.7 Fußgänger

Fußgänger sind in einer Fahrsimulation wichtig, da ohne Fußgänger das Verhalten des Fahrers an Fußgängerübergängen nicht geübt werden könnte. Als Autofahrer in der Realität muss nämlich vor einem nicht geregelten Schutzweg angehalten werden, damit einem Fußgänger das gefahrlose Überqueren ermöglicht werden kann. Außerdem lassen Fußgänger die Spielwelt realer und lebendiger wirken. Im Spiel gibt es die Möglichkeit im Optionsmenü die Anzahl der Fußgänger zwischen 0 und 20 einzustellen. Insgesamt gibt es 12 verschiedene Fußgängermodelle, davon 6 Männer und 6 Frauen. Alle Fußgänger wurden mit dem UMA (Unity Multipurpose Avatar) System erstellt [20]. Dabei handelt es sich um ein System, mit dem verschiedene Charaktere zur Laufzeit erstellt werden können. Bei diesen Avataren können neben Geschlecht auch andere Merkmale

wie z.B. Körpergröße angepasst werden. Ebenfalls kann man die Kleidung der Avatare auswählen und im Unity Asset Store kann man auch viele (oft kostenpflichtige) Bekleidungen finden. Neben Bekleidung können auch unter anderem Haare hinzugefügt werden. Die Gesamtheit aller hinzugefügten Objekte zu einem Avatar bezeichnet man als Recipe. Durch das UMA System werden zur Laufzeit die Meshes, Texturen und Bones eines Charakters und dessen Bekleidung kombiniert, wodurch eine bessere Performance erzielt und die Anzahl der Draw Calls minimiert werden kann [20]. Für alle 12 Fußgängermodelle, die in Virtual Driving School verwendet werden, musste die Bekleidung ausgewählt werden sowie die Frisur und die Haarfarbe. Es wurde darauf geachtet, dass die Bekleidungen der Fußgänger sowie die Frisuren sich möglichst wenig ähneln.



Abbildung 21: 6 männliche UMAs



Abbildung 22: 6 weibliche UMAs

3.7.1 Wegpunktsystem für Fußgänger

Damit die Fußgänger sich in der Spielwelt bewegen können, wurde ein Wegpunktsystem verwendet. Insgesamt wurden etwa 1000 Wegpunkte für Fußgänger per Hand manuell in der Spielwelt platziert. Um das Setzen der Wegpunkte zu erleichtern, wurde ein eigener benutzerdefinierter Editor für Unity aus [21] verwendet, mit dem Wegpunkte visualisiert werden können und einfacher gesetzt werden können. Ebenfalls wurde [21] als Hilfe verwendet beim Implementieren des Wegpunktsystems selbst, also wie Fußgänger sich zu einem Wegpunkt hinbewegen können und wie der nächste Wegpunkt ausgewählt werden kann.

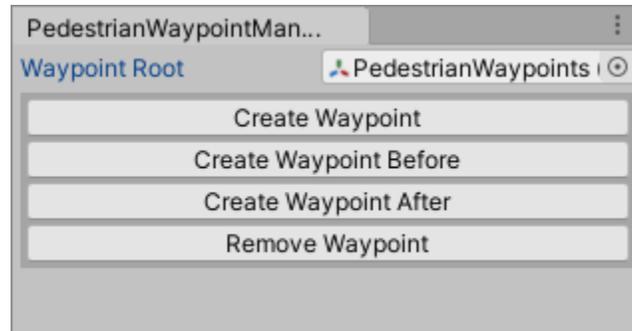


Abbildung 23: Editor zum einfachen Erstellen von Wegpunkten für Fußgänger

Mit dem Editor aus [21] können Wegpunkte erstellt werden. Ebenso kann ein Wegpunkt vor oder nach dem ausgewählten Wegpunkt erstellt werden. Ein ausgewählter Wegpunkt kann ebenso gelöscht werden.



Abbildung 24: Visualisierte Wegpunkte für Fußgänger

Die Wegpunkte werden in der Scene View von Unity als Würfel dargestellt und zwischen den Wegpunkten werden zwei Linien gezeichnet. Wegpunkte können in der Scene View ausgewählt

und verschoben werden. Verbindungen über Fußgängerübergänge sind mit blauen Linien versehen.

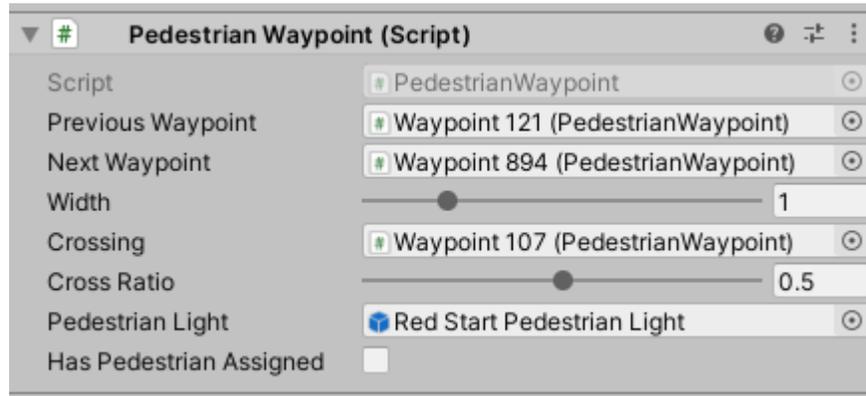


Abbildung 25: Aufbau eines Wegpunkts für Fußgänger

Jeder Wegpunkt kann einen vorherigen und nächsten Wegpunkt haben. Wegpunkte können zwischen Fußgängerübergängen verbunden werden über die Variable *crossing*. Dabei gibt es eine *crossRatio*, welche die Wahrscheinlichkeit darstellt, dass ein Fußgänger über den Fußgängerübergang geht, wenn er an einem Wegpunkt ankommt. Fußgänger warten, falls die Ampel rot ist oder wenn die Ampel grün blinkt. Außerdem wird die Variable *hasPedestrianAssigned* verwendet, damit nicht zwei Fußgänger am selben Wegpunkt spawnen, da diese sonst übereinanderstehen würden.

3.7.2 Pooling System für Fußgänger

Die Fußgängeranzahl kann im Optionsmenü auf einen Wert zwischen 0 und 20 gesetzt werden. Wenn man den maximalen Wert 20 auswählen würde und in der gesamten Spielwelt 20 Fußgänger an zufälligen Wegpunkten gespawnt werden würden, dann würde man als Fahrer im Spiel eher selten einen Fußgänger zu Gesicht bekommen und es würde unnötig Rechenleistung verschwendet werden für Fußgänger, die vielleicht auf der anderen Seite der Spielwelt sind. Um dies zu vermeiden wurde in Virtual Driving School ein Pooling System implementiert für Fußgänger. Das heißt, dass Fußgänger immer nur in der Nähe des Spielerautos sein können. Dazu wurde eine *nearZone* und eine *spawnZone* definiert. Der Wert der *nearZone* wurde auf 100 Meter gesetzt und der Wert der *spawnZone* auf 300 Meter. Am Anfang, wenn das Spiel gestartet wird (also wenn der Freie-Fahrt-Modus ausgewählt wird), werden die Fußgänger auf Wegpunkte gesetzt, die sich in der *spawnZone* befinden. Wenn der Spieler herumfährt und die Distanz zwischen Spieler und einem Fußgänger größer als 300 Meter wird, dann wird der Fußgänger an einen neuen Wegpunkt gesetzt, der sich in der *spawnZone* befindet. Damit der Fußgänger nicht zufällig direkt im Blickwinkel des Spielers „aufpoppt“, darf der Fußgänger nicht an einen Wegpunkt gesetzt werden, der sich zu nah am Spieler befindet. Dazu wird die Variable *nearZone* benötigt, damit Fußgänger nicht näher als 100 Meter zum Spieler gesetzt werden. Ebenfalls wird durch das Pooling System darauf geachtet, dass Fußgänger nicht an Wegpunkte gesetzt werden, wo sich schon ein Fußgänger befindet. Sonst könnte es manchmal passieren, dass Fußgänger zeitgleich an denselben Wegpunkt gespawnt werden und übereinanderstehen. Durch dieses Pooling System kann mit relativ wenigen Fußgängern trotzdem eine höhere Fußgängerdichte um den Spieler herum erreicht werden.

3.8 Autofahrer

Für eine Fahrsimulation sind Autofahrer, die vom Computer gesteuert werden, sehr wichtig. Ohne Autofahrer oder mit zu wenigen Autos würden die Straßen sehr leer wirken. Ebenfalls sind Autos auch notwendig, um bestimmte Situationen zu üben, z.B. das Halten des Sicherheitsabstands zum vorderen Auto oder Überholmanöver. Im Optionsmenü kann die Anzahl der Autofahrer zwischen 0 und 60 gesetzt werden. Insgesamt gibt es im Spiel 6 verschiedene Automodelle, jedes mit 5 unterschiedlichen Farben, sodass es insgesamt 30 verschiedene Prefabs gibt. Jedes Auto kann rechts oder links blinken. Ebenfalls leuchten die Bremslichter auf, wenn das Auto bremst. Ebenfalls besitzt jedes Automodell ein eigenes Nummernschild. Beim Automodell ganz links (siehe Abbildung 26) handelt es sich um dasselbe Automodell wie beim Spielerauto, jedoch hat das Spielerauto immer eine graue Farbe. Computergesteuerte Autos können bei einer roten Ampel stehen bleiben, genug Abstand zum vorderen Auto halten, bei Stoppschildern stehen bleiben, beim Linksabbiegen dem Gegenverkehr Vorrang geben, die Spur wechseln und die Blinker einschalten beim Spurwechsel oder beim Rechts- oder Linksabbiegen. In Virtual Driving School wurde entschieden, keine Personen in die Autos zu setzen, da dies die Performance verschlechtert hätte und sichtbare Personen in Autos eher als unwichtiges Feature gesehen wurde. Damit nicht sichtbar ist, dass sich in den Autos keine Personen befinden, wurden die Scheiben der Autos verdunkelt. Im Spielerauto befindet sich ebenfalls sowohl im First-Person- als auch im Third-Person-Modus keine Person im Auto.



Abbildung 26: Die 6 verschiedenen Automodelle

3.8.1 Wegpunktsystem für Autofahrer

Für die Autos, wurden insgesamt etwa 4000 Wegpunkte per Hand gesetzt. Damit dies leicht möglich ist und damit die Wegpunkte visualisiert werden können, wurde ein ähnlicher Editor wie bei den Fußgängern verwendet. Die Wegpunkte und das Wegpunktsystem für Autofahrer und

Fußgänger sind voneinander getrennt. Also es kann selbstverständlich kein Auto einen Wegpunkt für Fußgänger als Ziel verwenden oder umgekehrt.

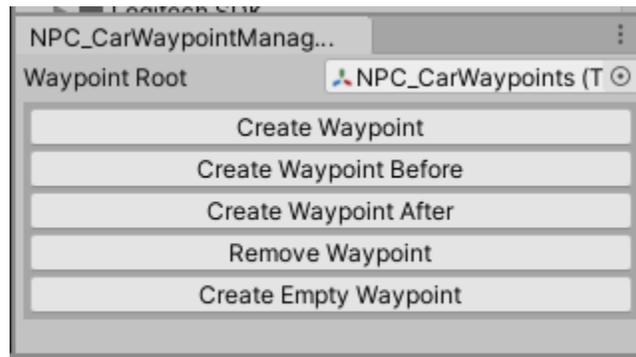


Abbildung 27: Custom Editor zum einfachen Erstellen von Wegpunkten für Autos

Wie bei den Fußgängern, können mit dem Editor ebenfalls Wegpunkte erstellt werden. Wenn ein Wegpunkt ausgewählt ist, kann ein Wegpunkt davor oder danach erstellt werden oder dieser kann gelöscht werden. Es können ebenfalls leere Wegpunkte erstellt werden, welche keinen vorherigen oder nächsten Wegpunkt besitzen.



Abbildung 28: Visualisierte Wegpunkte für computergesteuerte Autos

Die Wegpunkte für Autofahrer werden in der Scene View ebenfalls als Würfel dargestellt und zwischen den Wegpunkten werden auch zwei Linien gezeichnet. Wegpunkte können in der Scene View ausgewählt und verschoben werden. Wegpunkte können neben einem nächsten Wegpunkt auch einen Wegpunkt als *branch* haben. Dies ist nötig, z.B. wenn die rechte Spur bei einer Kreuzung zum Geradefahren und zum Rechtsabbiegen verwendet wird.

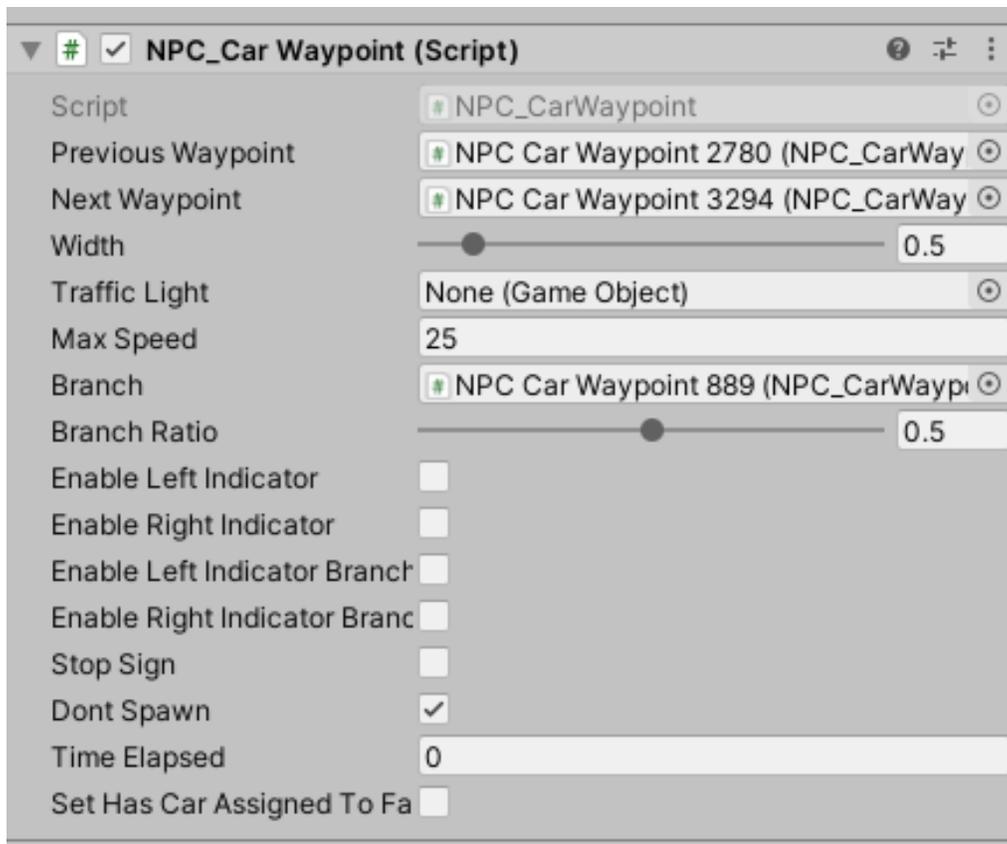


Abbildung 29: Aufbau eines Wegpunkts für NPC_Autos

Wegpunkte für Autofahrer können ebenso wie Wegpunkte für Fußgänger einen vorherigen und nächsten Wegpunkt haben. Ebenfalls können Wegpunkte, wie schon erwähnt, einen weiteren Wegpunkt in der Variable *branch* haben. Die *branchRatio* gibt an wie wahrscheinlich es ist, dass der Wegpunkt in *branch* als nächster genommen wird, statt dem Wegpunkt in *nextWaypoint*. Ebenfalls kann jedem Wegpunkt ein *trafficLight* zugewiesen werden. Dies wird verwendet, um zu wissen, ob die Fahrzeugampel auf Rot ist, damit das Auto stehen bleibt. Für jeden Wegpunkt kann auch eine Geschwindigkeit in der Variable *maxSpeed* gewählt werden. Dies ist nötig, da Autos z.B. auf Autobahnen deutlich schneller fahren können als z.B. in der 30er-Zone oder im Ortsgebiet ohne Geschwindigkeitsbegrenzung. Ebenfalls wird die *maxSpeed* beim Links- und Rechtssabbiegen bei Kreuzungen niedriger gesetzt, damit die Autos nicht zu schnell einbiegen. Die beiden Variablen *enableLeftIndicator* und *enableRightIndicator* können bei jedem Wegpunkt gesetzt werden, damit ein Auto, welches zu diesem Wegpunkt fährt, den entsprechenden Blinker setzen kann. Es gibt ebenfalls die beiden Variablen *enableLeftIndicatorBranch* und *enableRightIndicatorBranch*. Diese können auch bei jedem Wegpunkt gesetzt werden, falls z.B. ein Auto nur blinken sollte, wenn es gerade einen *branch* genommen hat. Dies ist notwendig beim Links- und Rechtsabbiegen, damit Geradefahrer nicht auch den Blinker setzen wie Abbieger. Die Variable *stopSign* kann gesetzt werden, wenn ein Auto zum Stillstand kommen muss am Wegpunkt und dann erst weiterfahren darf, also dann, wenn sich auf der Straße ein Stoppschild befindet. Die Variable *dontSpawn* ist nötig, da auf manchen Wegpunkten kein Auto durch das Pooling System gesetzt werden darf, z.B. auf Wegpunkten in Kreuzungsmitten, da es zu Unfällen führen könnte, wenn gerade die Querrichtung fahren darf. Sowie bei den Fußgängern, muss bei Autowegpunkten geachtet werden, dass nicht zwei Autos zur selben Zeit an denselben Wegpunkt gesetzt werden. Wenn dies passieren würde, dann befinden sich zwei Objekte im selben Raum und kollidieren

miteinander. Damit dies nicht passiert, können Wegpunkte über die private Variable *hasCarAssigned* reserviert werden, damit nicht noch ein Auto an denselben Wegpunkt gesetzt wird durch das Pooling System.

3.8.2 Pooling System für Autofahrer

Das Pooling System für Autofahrer funktioniert ähnlich wie das Pooling System für Fußgänger. Das Pooling System für Autofahrer und für Fußgänger ist jedoch strikt voneinander getrennt gehalten, da es ein wenig anders funktioniert. Es gibt ebenfalls eine *nearZone* und eine *spawnZone*. Der Wert der *nearZone* wurde auf 175 Meter gesetzt und der Wert der *spawnZone* wurde auf 500 Meter gesetzt. Wenn das Spiel gestartet wird (also, wenn eine Lektion oder der Freie-Fahrt-Modus ausgewählt wird), dann werden die Autos an Fahrzeugwegpunkte in der *spawnZone* platziert, also an Wegpunkten die nicht weiter als 500 Meter vom Spieler entfernt sind. Wenn ein Auto sich nicht mehr in der *spawnZone* befindet, wird es an einen anderen Wegpunkt gesetzt, der sich in der *spawnZone* befindet, aber außerhalb der *nearZone*. Dadurch kommt es nicht vor, dass Autos direkt im nahen Blickfeld des Spielers aufpoppen. Der Wert der *nearZone* wurde bei Autofahrern mit 175 Metern höher gesetzt als bei Fußgängern, wo der Wert lediglich 100 Meter beträgt. Dies ist deswegen so, da ein Auto weitaus größer ist als ein Fußgänger und ein großes Objekt, welches in der Nähe des Spielers aufpoppt, leichter gesehen werden würde. Wenn ein Auto an einen Wegpunkt gesetzt wurde, dann wird dieser Wegpunkt für 3 Sekunden reserviert, sodass nicht noch ein Auto an denselben Wegpunkt gesetzt werden darf. Dadurch kann es nicht dazu kommen, dass zwei Autos gleichzeitig auf den selben Wegpunkt gesetzt werden.

3.9 Menüs

3.9.1 Hauptmenü

Das Erstellen des Hauptmenüs war dank Unity eine sehr einfache Aufgabe. Für das Hintergrundbild des Hauptmenüs wurde ein Screenshot aus dem Spiel genommen und ein Gaußscher Weichzeichner angewendet, um die Schärfe zu verringern, um eine bessere Lesbarkeit im Menü zu bekommen. Im Hauptmenü gibt es ebenfalls eine Hintergrundmusik. An der linken unteren Seite steht entweder „Default (ohne VR)“ oder „VR“. Damit wird signalisiert, ob es sich bei der gestarteten Version um die normale Version oder um die VR-Version handelt. Wenn der Spieler auf „Lektionen“ klickt, wird er auf ein Menü weitergeleitet, in dem er eine Lektion auswählen kann. Wenn der Spieler auf „Freie Fahrt“ klickt, wird der Freie-Fahrt-Modus geladen, in dem er frei umherfahren kann. Mit einem Klick auf „Steuerung“ wird das Steuerungsmenü geöffnet, indem die Steuerung eingesehen werden kann, nicht jedoch geändert werden kann. Und mit einem Klick auf „Optionen“ wird das Optionsmenü geöffnet.



Abbildung 30: Hauptmenü

3.9.2 Lektionen

Insgesamt gibt es in Virtual Driving School 8 Lektionen:



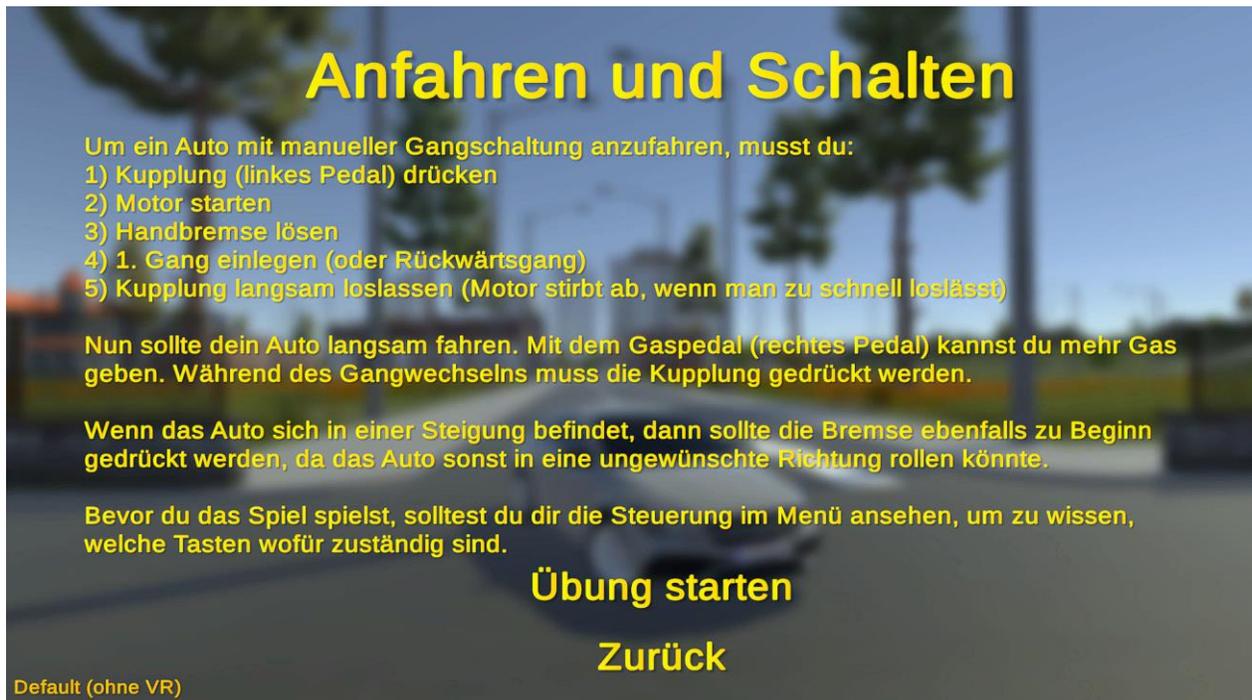
Abbildung 31: Lektionen, welche gespielt werden können

Die Lektionen können über das Hauptmenü ausgewählt werden. Der Spieler bekommt eine Erklärung wie die Übung durchgeführt werden muss und kann dann die Übung starten. Der Spieler

befindet sich dann im Spiel und kann selbst ausprobieren, was er gerade gelesen hat. Die Erklärung der Lektion kann auch im Pausemenü angesehen werden. Wenn der Spieler mit der Übung fertig ist, kann er selbstständig über das Pausemenü wieder ins Hauptmenü gelangen und dann eine andere Lektion auswählen oder den Freie-Fahrt-Modus spielen. Jemand der keine Erfahrung im Autofahren hat, wird nach dem Durchspielen aller Lektionen selbstverständlich nicht bereit sein für den Straßenverkehr. Wichtiges Grundwissen, wie z.B. die Vorrangregeln oder welche Geschwindigkeiten gefahren werden dürfen, wurden dem Spieler noch nicht vermittelt. Solche Lektionen wurden in Virtual Driving School nicht eingebaut, da die praktischen Übungen hierfür nicht gut funktioniert hätten, da es z.B. sehr viele Vorrangregeln gibt und nicht klar wäre, wie so eine Übung aussehen könnte. Die Übungen finden alle auf dem Übungsplatz statt, welcher sich westlich in der Südstadt befindet. Wenn der Spieler sich in einer Übung befindet, kann er nicht aus dem Übungsplatz herausfahren. Dabei handelt es sich um eine Designentscheidung, die so getroffen wurde, da der Spieler den Freie-Fahrt-Modus auswählen sollte, wenn er frei umherfahren möchte. Ebenfalls werden keine Fußgänger oder Autofahrer gespawnt, wenn der Spieler eine Lektion spielt, um die Performance zu verbessern.

3.9.3 Lektion Anfahren und Schalten

In dieser Lektion geht es darum, dass der Spieler lernt, wie man ein Auto startet und losfährt. Es wurde darauf geachtet, dass die Schritte möglichst einfach und kurz gehalten sind, damit der Spieler nicht überfordert wird. Dem Spieler wird auch gesagt, dass er sich die Steuerung im Menü ansehen soll, bevor er beginnt. Unabhängig davon ob Maus und Tastatur verwendet wird oder das Logitech G29, müsste der Spieler sonst lange rumprobieren bis er die richtigen Tasten findet. Für diese Lektion wird immer die manuelle Gangschaltung verwendet statt der automatischen Gangschaltung, da das Anfahren und Schalten mit einer automatischen Schaltung trivial ist, da man nur den Motor einschalten und Gas geben müsste, um loszufahren.



Anfahren und Schalten

Um ein Auto mit manueller Gangschaltung anzufahren, musst du:

- 1) Kupplung (linkes Pedal) drücken
- 2) Motor starten
- 3) Handbremse lösen
- 4) 1. Gang einlegen (oder Rückwärtsgang)
- 5) Kupplung langsam loslassen (Motor stirbt ab, wenn man zu schnell loslässt)

Nun sollte dein Auto langsam fahren. Mit dem Gaspedal (rechtes Pedal) kannst du mehr Gas geben. Während des Gangwechsels muss die Kupplung gedrückt werden.

Wenn das Auto sich in einer Steigung befindet, dann sollte die Bremse ebenfalls zu Beginn gedrückt werden, da das Auto sonst in eine ungewünschte Richtung rollen könnte.

Bevor du das Spiel spielst, solltest du dir die Steuerung im Menü ansehen, um zu wissen, welche Tasten wofür zuständig sind.

Übung starten

Zurück

Default (ohne VR)

Abbildung 32: Lektion Anfahren und Schalten

3.9.4 Lektion Parallelparken

In dieser Lektion kann der Spieler lernen bzw. üben wie man parallelparkt. Im Vergleich zu anderen Lektionen ist diese Lektion relativ schwierig, besonders wenn nur die First-Person-Sicht genutzt wird. Das parallele Einparken eines Autos ist auch im echten Leben ein eher schwieriges Manöver, besonders wenn die Parklücke sehr eng ist. Um die Lektion im Spiel zu veranschaulichen, werden ebenfalls Bilder zusätzlich zum Erklärungstext gezeigt.

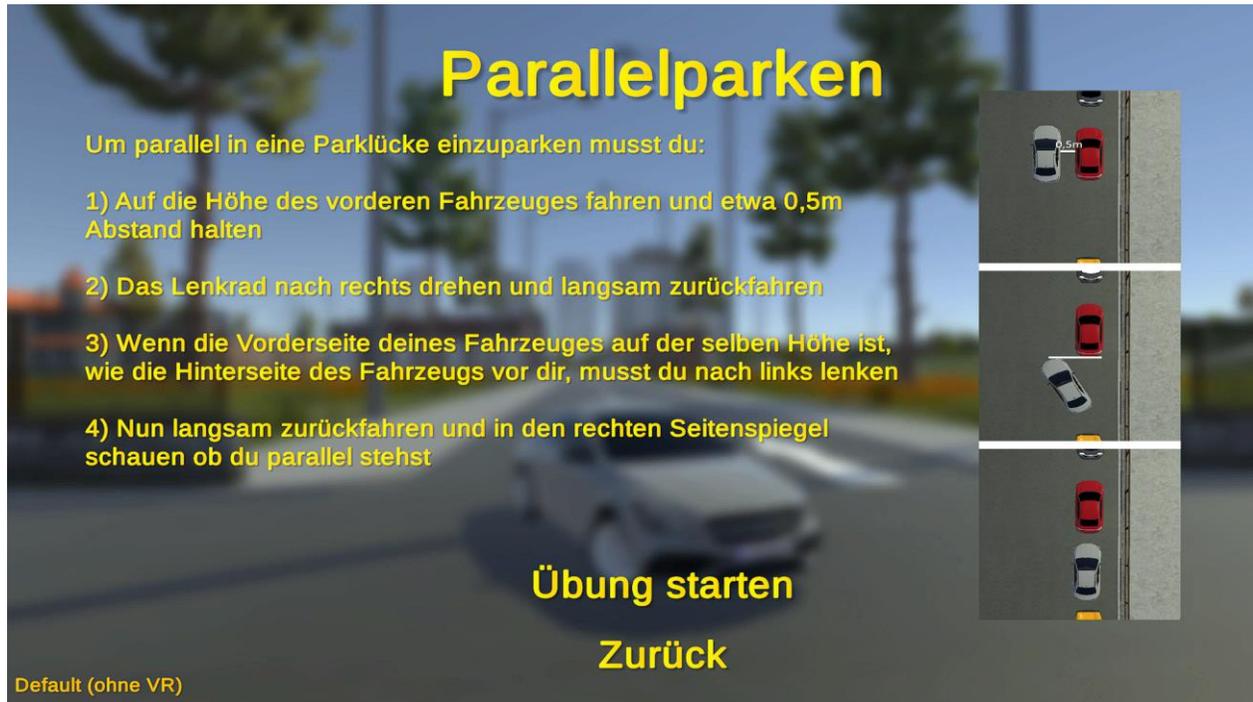


Abbildung 33: Lektion Parallelparken

3.9.5 Lektion Rückwärts einparken

In dieser Lektion kann der Spieler lernen wie man ein Auto rückwärts einparkt. Diese Lektion kann auch schwierig sein, wenn nur die First-Person-Perspektive verwendet wird. In solche Parklücken kann auch vorwärts eingeparkt werden, jedoch ist dies ein triviales Manöver. Im Erklärungstext wird auch erwähnt, dass der Spieler im echten Leben alternativ auch vorwärts einparken kann, dann aber jedoch rückwärts ausparken muss, wenn er das nächste Mal ins Auto steigt. Um zu veranschaulichen, welche Schritte in dieser Lektion gesetzt werden müssen, werden hier ebenfalls Bilder verwendet zusätzlich zum Erklärungstext (siehe Abbildung 34).

Rückwärts einparken

Um rückwärts einzuparken musst du:

- 1) Nach vorne fahren bis die Hinterseite deines Autos auf der Höhe des vorderen Autos ist, dabei etwa 1m Abstand halten
- 2) Lenkrad ganz nach rechts drehen und langsam rückwärts fahren
- 3) Wenn du parallel zu den anderen Autos stehst, das Lenkrad wieder geradestellen

Alternativ zum rückwärts einparken, kannst du auch vorwärts einparken, musst dann jedoch beim nächsten Mal rückwärts ausparken

Übung starten

Zurück

Default (ohne VR)

Abbildung 34: Lektion Rückwärts einparken

3.9.6 Lektion Umkehren

In dieser Lektion kann der Spieler lernen wie man mit einem Auto umkehrt bzw. reversiert. Dazu fährt der Spieler in eine durch Verkehrshüte markierte Lücke hinein und wieder heraus, um umzukehren. Um die einzelnen Schritte zu visualisieren, werden ebenfalls Bilder verwendet. In den Bildern werden Pfeile verwendet, um die Fahrtrichtung des Autos zu verdeutlichen.

Umkehren

Um umzukehren bzw. zu reversieren, musst du:

- 1) Einen passenden Ort finden auf der linken Seite, z.B. eine Einfahrt oder Parklücke (Achtung: Gegenverkehr hat Vorrang)
- 2) Lenkrad nach links drehen und in die Lücke hineinfahren
- 3) Lenkrad nach rechts drehen und herausfahren

Um die Fahrtrichtung zu ändern, eignet sich auch ein Kreisverkehr oder eine (durch Arm- oder Lichtzeichen) geregelte Kreuzung

Übung starten

Zurück

Default (ohne VR)

Abbildung 35: Lektion Umkehren

3.9.7 Lektion Rechtsabbiegen

In dieser Lektion kann der Spieler lernen welche Schritte gesetzt werden müssen, wenn man mit dem Auto rechts abbiegen möchte. Auf das Erwähnen trivialer Informationen, wie z.B., dass der Spieler anhalten muss, falls es eine rote Ampel gibt, wurde verzichtet, da der Erklärungstext sonst zu lange geworden wäre. Der Spieler lernt in dieser Lektion wie der 3-S Blick vor dem Blinken funktioniert und, dass vor dem Einbiegen nochmals ein Schulterblick gesetzt werden soll, um Fußgänger oder Radfahrer nicht zu gefährden. Ebenfalls wird erwähnt, dass in einem engen Bogen nach rechts eingebogen werden muss.

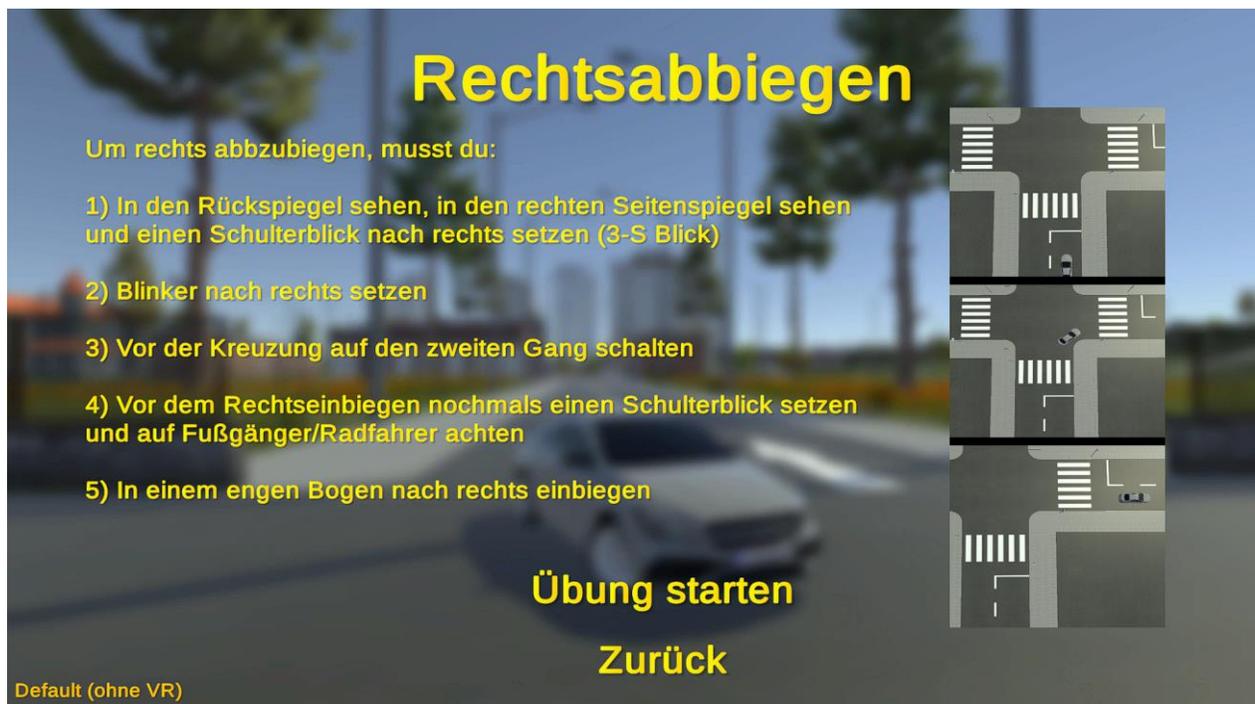


Abbildung 36: Lektion Rechtsabbiegen

3.9.8 Lektion Linksabbiegen

In dieser Lektion lernt der Spieler, wie er mit dem Auto korrekt links abbiegen kann. Hier wird ebenfalls im Erklärungstext darauf verzichtet, dem Spieler zu sagen, dass bei einer roten Ampel stehen geblieben werden muss, da dies trivial ist. Der Spieler lernt, dass er den Gegenverkehr zuerst fahren lassen muss und in einem weiten Bogen links einbiegen muss. Ebenfalls wird erwähnt, dass der Spieler auf Fußgänger und Radfahrer achten muss, welche die Straße überqueren möchten.

Linksabbiegen

Um links abzubiegen, musst du:

- 1) In den Rückspiegel sehen, in den linken Seitenspiegel sehen und dann einen Schulterblick nach links setzen (3-S Blick)
- 2) Nach links blinken
- 3) In den zweiten Gang schalten
- 4) Dem Gegenverkehr Vorrang geben, dann in einem weiten Bogen links einbiegen, dabei auf Fußgänger/Radfahrer achten

Übung starten

Zurück

Default (ohne VR)

Abbildung 37: Lektion Linksabbiegen

3.9.9 Lektion Kreisverkehr

In dieser Lektion lernt der Spieler wie er sich in einem Kreisverkehr zu verhalten hat. Dabei lernt der Spieler, dass das in den Kreisverkehr einfahrende Auto Vorrang hat, da die Rechtsregel gilt, außer es gibt ein Vorrang-geben-Schild. Der Spieler lernt auch, dass beim Ausfahren aus dem Kreisverkehr ebenfalls geblinkt werden muss.

Kreisverkehr

Bei einem Kreisverkehr musst du:

- 1) In den 2. Gang schalten (meistens, sonst 3. Gang)
- 2) In den Kreisverkehr nach rechts einfahren (dabei hat der Einfahrende Vorrang, da die Rechtsregel gilt, außer bei einem Vorrang-geben-Schild)
- 3) Vor der gewählten Ausfahrt nach rechts blinken
- 4) Auf Fußgänger/Radfahrer achten
- 5) Aus dem Kreisverkehr herausfahren

Übung starten

Zurück

Default (ohne VR)

Abbildung 38: Lektion Kreisverkehr

3.9.10 Lektion Spurwechsel

In dieser Lektion lernt der Spieler, wie er die Spur korrekt wechselt. Dazu muss ein 3-S Blick gesetzt werden, wonach geblinkt wird, wonach nochmals ein Schulterblick gesetzt werden muss, vor dem Wechseln der Spur. Der Schulterblick ist wichtig, da sonst Fahrzeuge im toten Winkel übersehen werden können, die einen z.B. gerade überholen möchten. Das Wechseln der Spur ist ein Manöver, welches sehr häufig gesetzt werden muss und sollte daher als Fahranfänger gut geübt werden. Besonders auf Autobahnen muss oft die Spur gewechselt werden und daher sollte das Wechseln der Spur gelernt sein.



Abbildung 39: Lektion Spurwechsel

3.9.11 Steuerungsmenü

Virtual Driving School kann mit dem Logitech G29 Lenkrad gesteuert werden oder mithilfe von Maus und Tastatur. Im Optionsmenü kann der Spieler auswählen, welche Art von Steuerung er verwenden möchte. Im Hauptmenü sowie im Pausemenü wird dem Spieler die Steuerung erklärt. Dies ist wichtig, da die Steuerung relativ komplex ist und es daher ein eigenes Menü geben sollte, welches die Steuerung erklärt. In der Mitte des Steuerungsmenüs (siehe Abbildung 40) sieht man die Aktion, während man auf der linken Seite die Eingabe für die Tastatur und Maus sieht und auf der rechten Seite sieht man die Eingabe für das Logitech G29 Lenkrad. Die Steuerung kann nicht geändert werden. Dies wurde so entschieden, da es relativ aufwendig gewesen wäre, dem Spieler die Möglichkeit zu geben, die Steuerung zu ändern und vor allem da gerade beim Logitech G29 Lenkrad ein Ändern der Steuerung sehr schwierig gewesen wäre, da das Lenkrad über viele Buttons verfügt, wovon jedoch die meisten Buttons schon für Virtual Driving School belegt sind. Wichtig zu erwähnen ist, dass keine anderen Lenkräder außer das Logitech G29 mit den dazugehörigen Pedalen und der Gangschaltung unterstützt werden.

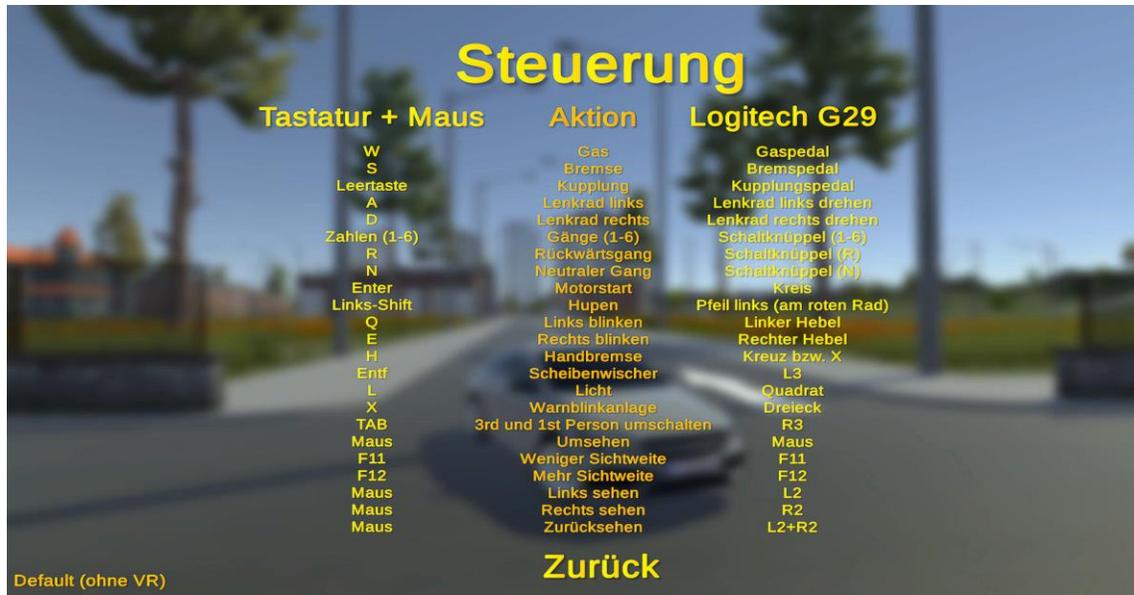


Abbildung 40: Steuerung von Virtual Driving School

3.9.12 Pausemenü (Freie Fahrt)

Das Pausemenü kann im Spiel geöffnet werden über die Escape-Taste. Im Pausemenü kann über Escape oder über einen Klick auf „Fortfahren“ das Spiel fortgefahren werden. Ebenfalls kann im Pausemenü die Steuerung angesehen werden. Dies ist wichtig, da die Steuerung für das Spiel doch relativ kompliziert sein kann, gerade wenn mit einer manuellen Schaltung gefahren wird. Wenn der Spieler nicht wüsste, wie z.B. die Handbremse deaktiviert wird oder wie der Motor eingeschalten werden kann, dann müsste er rumprobieren, bis er die richtige Taste findet. Der Spieler kann ebenfalls im Pausemenü zurück ins Hauptmenü gelangen. Während das Pausemenü offen ist, ist das Spiel pausiert.



Abbildung 41: Pausemenü (im Freien-Fahrt-Modus)

3.9.13 Pausemenü (Lektion)

Das Pausemenü während einer Lektion unterscheidet sich vom Pausemenü im Freien-Fahrt-Modus, dadurch, dass der Spieler ebenfalls den Erklärungstext der Lektion lesen kann, welchen er lesen konnte, bevor die Übung gestartet wurde. Dies ist wichtig, falls ein Spieler bei einer Lektion vergisst, wie die Übung genau ausgeführt werden muss. Dadurch muss er nicht extra wieder ins Hauptmenü, um nachzulesen was zu tun ist. Es finden sich hier dieselben Buttons wie im normalen Pausemenü.

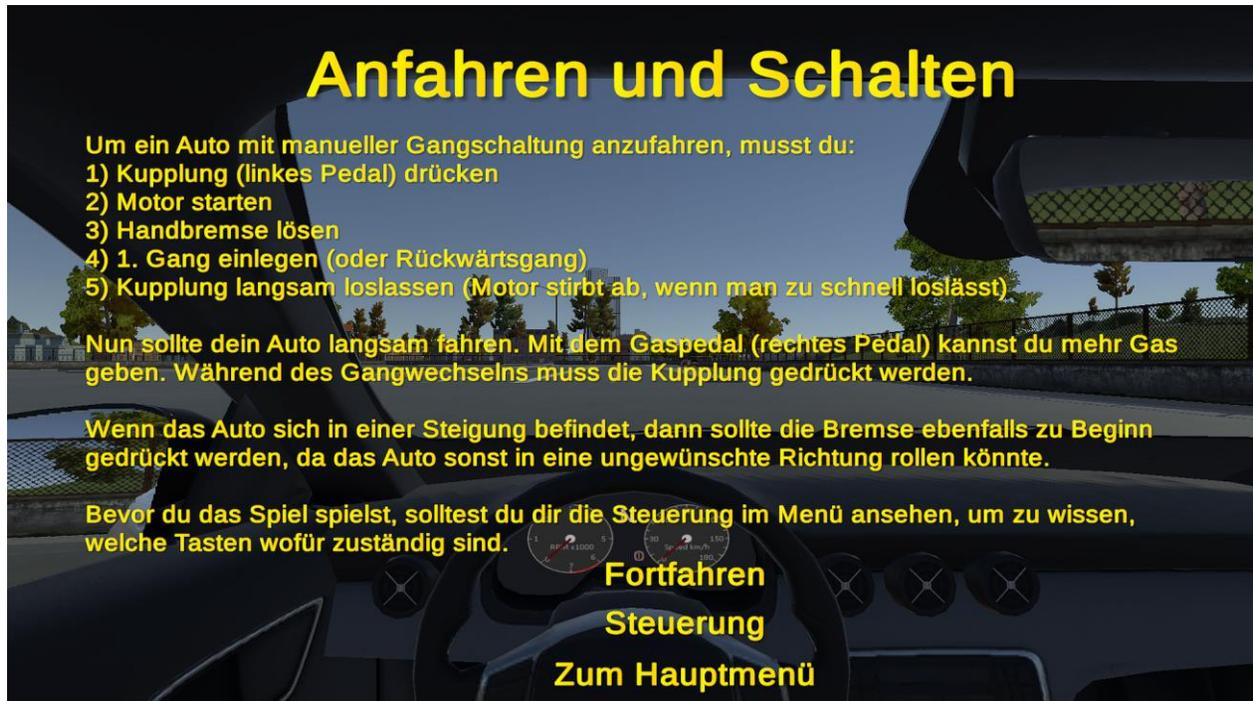


Abbildung 42: Pausemenü (während Lektion)

3.9.14 Optionsmenü

Im Optionsmenü (siehe Abbildung 43) kann unter anderem die Auflösung eingestellt werden. Die einstellbaren Auflösungen hängen ab vom verwendeten Computer und werden beim Starten des Spiels abgefragt. Unterstützt werden Auflösungen mit dem Verhältnis 16:9, 16:10 und 4:3. Ebenfalls kann die Grafikqualität eingestellt werden. Dabei gibt es die Optionen Sehr Niedrig, Niedrig, Mittel, Hoch, Sehr Hoch und Ultra. Ebenfalls kann im Optionsmenü die Art der Eingabe ausgewählt werden, also ob der Spieler mit Maus und Tastatur spielen möchte oder mit dem Logitech G29 Lenkrad. Die Anzahl der Fußgänger kann zwischen 0 und 20 gesetzt werden und die Anzahl der Autofahrer kann zwischen 0 und 60 gesetzt werden. Die Sichtweite kann zwischen 200 und 2000 gewählt werden, wobei die Sichtweite auch im Spiel per F11 verringert und per F12 erhöht werden kann. Dies wurde implementiert, da man eventuell bei Rucklern die Sichtweite schnell verringern möchte bzw. erhöhen möchte, wenn das Spiel flüssig läuft und man mehr von der Spielwelt sehen möchte. Ebenfalls kann ausgewählt werden, ob eine manuelle oder automatische Gangschaltung verwendet werden soll. Mit einem Klick auf „Auf Standard zurücksetzen“ werden alle Optionen auf die Standardeinstellungen zurückgesetzt. Diese sind so wie in der Abbildung 43 eingestellt, wobei für die Auflösung die höchstmögliche Auflösung des

Systems ausgewählt wird. Die Einstellungen werden mit Unity über PlayerPrefs in der Registry gespeichert und bleiben somit erhalten, wenn der Spieler das Spiel neustartet. Die Optionen können nicht geändert werden, während man sich in einer Lektion oder im Freie-Fahrt-Modus befindet. Dies ist eine Designentscheidung, die so getroffen wurde, da eine Umstellung der Anzahl der Fahrzeuge und Fußgänger während einer Lektion oder im Freien-Fahrt-Modus nicht funktionieren würde. Wenn der Spieler während des Spielens eine Option (mit Ausnahme der Sichtweite) ändern möchte, muss er zurück ins Hauptmenü und dann ins Optionsmenü.



Abbildung 43: Optionsmenü

3.10 Skripts

Die Skripts für Virtual Driving School wurden alle in C# geschrieben. Insgesamt handelt es sich um 26 Skripts, welche in mehrere Ordner (PlayerCar, Pedestrian, NPC_Car, Lights, Menu) aufgeteilt wurden, um die Übersichtlichkeit zu verbessern. In dieser Sektion wird eine kurze Übersicht gegeben über die Funktionen der jeweiligen Skripts. Die Überschriften geben den Ordner an, in dem sich die Skripts befinden.

3.10.1 PlayerCar

CarController: Hierbei handelt es sich um die umfangreichste Klasse im gesamten Projekt. In dieser Klasse wird ein Großteil der Logik für das Spielerauto implementiert. Zum Beispiel wird der Input des Spielers abgefragt um die Lenkung, die Beschleunigung und das Bremsen des Autos zu steuern. Ebenfalls werden die Räder des Spielerautos geupdated, sodass diese richtig orientiert sind. Ebenfalls wurde hier die manuelle und automatische Gangschaltung implementiert. Der Wechsel zwischen Third-Person-Sicht und First-Person-Sicht wurde hier ebenfalls programmiert sowie andere kleinere Sachen wie z.B. die Steuerung der Scheibenwischer.

GameInput: Diese Klasse hat nur 4 Variablen. Dabei handelt es sich um die float Variablen: *steering*, *gas*, *brake* und *clutch*. Die Variable *steering* gibt den Grad der Lenkung an und kann zwischen -1 und 1 sein, dabei steht -1 für ganz links und 1 für ganz rechts. Die Variablen *gas*, *brake* und *clutch* haben immer einen Wert zwischen 0 und 1 und geben Gas, Bremse und Kupplung an. Die Klassen **G29Input** und **KeyboardInput** erben von dieser Klasse.

G29Input: Diese Klasse erbt von **GameInput** und hier wird die Steuerung für das Logitech G29 Lenkrad abgefragt.

KeyboardInput: Diese Klasse erbt ebenfalls von **GameInput** und hier wird die Steuerung für Maus und Tastatur abgefragt.

CarLights: Über diese Klasse können die Fahrtrichtungsanzeiger bzw. Blinker, sowie die Bremslichter, die Warnblinkanlage und das Abblendlicht gesteuert werden. Ebenfalls werden hierüber die Zeichen auf dem Armaturenbrett für die Blinker, das Abblendlicht und die Handbremse gesteuert.

GearTextScript: Hierbei handelt es sich um ein sehr kleines Skript, welches die Aufgabe hat, den Text für den momentan gewählten Gang auf dem Armaturenbrett anzuzeigen. Hierfür wird der momentan gewählte Gang aus dem **CarController** genommen und dann wird der jeweilige Gang angezeigt. Es gibt die Gänge 1, 2, 3, 4, 5, 6, R (Rückwärtsgang) und N (Neutraler Gang).

RPMPointerScript: Über dieses kleine Skript wird lediglich der rote Zeiger auf der RPM Anzeige auf dem Armaturenbrett gesteuert. Die momentane RPM wird aus dem **CarController** genommen und dann wird der rote Zeiger so rotiert, dass er die richtige RPM auf dem Armaturenbrett anzeigt.

SpeedPointerScript: Hierüber wird, ähnlich wie beim **RPMPointerScript**, der rote Zeiger auf der Geschwindigkeitsanzeige auf dem Armaturenbrett gesteuert. Die momentane Geschwindigkeit wird aus dem **CarController** genommen und dann wird der rote Zeiger so rotiert, dass er die richtige Geschwindigkeit auf dem Armaturenbrett anzeigt.

SpeedTextScript: Diese Klasse kann die Geschwindigkeit als Text im Armaturenbrett anzeigen. Diese Klasse wurde jedoch nur am Anfang der Implementierung verwendet, als noch kein realistisches Armaturenbrett mit roten Zeigern implementiert war.

3.10.2 Pedestrian

PedestrianWaypoint: Diese Klasse wird verwendet für Wegpunkte für Fußgänger.

PedestrianWaypointNavigator: Diese Klasse wird verwendet, um den nächsten Wegpunkt auszuwählen, wenn ein Fußgänger einen Wegpunkt erreicht hat. Ebenfalls wird in dieser Klasse per Zufall die Entscheidung getroffen ob ein Fußgänger über einen Schutzweg geht, wenn er an einem vorbeikommt.

PedestrianNavigationController: Diese Klasse prüft unter anderem ob ein Fußgänger den Wegpunkt erreicht hat, welchen er als Ziel gesetzt bekommen hat. Ebenfalls findet in diesem Skript eine Translation sowie Rotation der Fußgänger statt und die Fußgänger werden ebenfalls animiert. Falls ein Fußgänger an einer Ampel warten muss, wird die Geschwindigkeit des Fußgängers auf 0 gesetzt.

PedestrianSpawner: Hier werden Fußgänger am Anfang des Spiels in der *spawnZone* gespawnt. Es werden so viele Fußgänger gespawnt, wie in den Optionen eingestellt wurde. Ebenfalls wird in diesem Skript jeder Fußgänger, welcher sich nicht mehr in der *spawnZone* des Spielerautos befindet, an einen neuen Wegpunkt gesetzt, welcher sich in der *spawnZone* befindet, nicht jedoch in der *nearZone*. Die *nearZone* wird benötigt, damit die Fußgänger nicht im nahen Blickfeld des Spielers aufpoppen. Die Klasse kümmert sich somit um das Spawnen der Fußgänger und um das Pooling, damit Fußgänger immer nur in der Nähe des Spielers zu sehen sind. Dadurch ist eine geringere Anzahl an Fußgängern notwendig, ohne, dass die Gehsteige in der Spielwelt zu leer wirken. Die Prüfung ob Fußgänger an einen anderen Wegpunkt gesetzt werden müssen, findet alle 5 Sekunden statt.

3.10.3 NPC_Car

NPC_CarWaypoint: Diese Klasse wird verwendet für Wegpunkte für computergesteuerte Autofahrer bzw. NPC Autos. Jeder Wegpunkt hat unter anderem eine maximale Geschwindigkeit, die gefahren werden darf und es kann auch eingestellt werden, dass der rechte oder der linke Blinker eingeschalten werden muss.

NPC_CarWaypointNavigator: Diese Klasse wird wie bei den Fußgängern verwendet, um den nächsten Wegpunkt auszuwählen, wenn ein Auto einen Wegpunkt erreicht hat. Ebenfalls wird in dieser Klasse per Zufall entschieden ob der Wegpunkt in der Variable *branch* genommen werden soll, falls der jetzige Wegpunkt über einen verfügt.

NPC_CarController: Diese Klasse ist etwas umfangreicher. Hier wird die Lenkung und die Beschleunigung der NPC Autos eingestellt. Die Lenkung wird so eingestellt, dass der nächste Wegpunkt erreicht werden kann. Ebenfalls wurde hier das Verhalten implementiert, dass NPC Autos stehen bleiben bei roten Ampeln oder wenn sich vor ihnen ein anderes Auto befindet. Ebenfalls bleiben NPC Autos stehen, wenn es am Zielwegpunkt ein Stoppschild gibt. Da es, wenn auch selten, dazu kommen kann, dass ein NPC Auto einen Unfall baut, da z.B. ein Wegpunkt zu knapp gesetzt ist, wird in diesem Skript auch geprüft, ob die Geschwindigkeit des Autos knapp 0 ist, obwohl das Auto nicht vor einer roten Ampel steht. Falls dies passiert, ist wahrscheinlich ein Unfall passiert und das Auto wird im NPC_CarSpawner an einen neuen Wegpunkt gesetzt, falls sich das Auto nicht in der *nearZone* des Spielers befindet.

NPC_CarLightScript: In diesem Skript werden die Bremslichter und die Blinker der NPC_Cars gesteuert.

NPC_CarSpawner: In dieser Klasse werden ebenfalls wie bei den Fußgängern, die NPC Autos gespawnt und gepoolt. Es werden so viele NPC Autos gespawnt, wie im Optionsmenü eingestellt. Die NPC Autos werden in der *spawnZone* gespawnt. Falls diese sich später nicht mehr in der *spawnZone* befinden, werden sie an einen anderen Wegpunkt gesetzt, welcher sich in der *spawnZone* befindet, nicht jedoch in der *nearZone*. Die *spawnZone* und die *nearZone* bewegt sich mit dem Spielerauto. Die Prüfung ob NPC Autos an einen neuen Wegpunkt gesetzt werden müssen, findet dabei alle 5 Sekunden statt.

3.10.4 Lights

PedestrianLightScript: Diese Klasse hängt an jeder Fußgängerampel. Es gibt im Spiel Fußgängerampeln, die mit Rot starten und welche die mit Grün starten. Diese wechseln sich immer ab. Wenn eine Richtung Rot hat, dann hat die Querrichtung Grün. Die Farben sind auch abgestimmt mit den Fahrzeugampeln. Bei Fußgängerampeln gibt es die Zustände Rot, Grün und Grünblinkend und der korrekte Wechsel zwischen den Zuständen wurde in diesem Skript implementiert.

TrafficLightScript: Diese Klasse hängt an jeder Fahrzeugampel. Im Spiel gibt es ebenfalls Fahrzeugampeln, die mit Rot starten und welche die mit Grün starten. Dabei wechseln sich diese immer ab. Wenn eine Richtung Rot hat, dann hat die Querrichtung Grün. Die Farben sind auch korrekt abgestimmt mit den Fußgängerampeln. Bei Fahrzeugampeln gibt es die Zustände Grün, Grünblinkend, Gelb, Rot und Rotgelb und der korrekte Wechsel zwischen den Zuständen wurde in diesem Skript implementiert. Hierbei ist wichtig zu wissen, dass Fahrzeugampeln in der Realität, wenn sie von Grün auf Rot schalten, dazwischen den Zustand Gelb haben. Wenn jedoch von Rot auf Grün geschaltet wird, dann kommt dazwischen der Zustand Rot mit Gelb gleichzeitig. Dies wurde im Spiel ebenfalls berücksichtigt.

StreetLightScript: Dieses Skript wird im Spiel nicht mehr verwendet. Im Spiel befinden sich Straßenlaternen an allen Straßen im Ortsgebiet. Dieses Skript hängte an jeder Straßenlaterne, um die Laterne ein oder auszuschalten. Jedoch gab es starke Performanceprobleme, wenn jede Straßenlaterne eingeschaltet wurde. Da im Spiel kein Nachtmodus implementiert wurde, wurde entschieden, dass Straßenlaternen immer ausgeschalten sind.

3.10.5 Menu

HauptmenüScript: Dieses Skript implementiert alle möglichen Funktionen, welche im Hauptmenü benötigt werden. Wenn das Spiel gestartet wird, dann werden am Anfang die Optionen über PlayerPrefs aus der Registry gelesen, falls gespeicherte Optionen in der Registry vorhanden sind. Die Optionen im Optionsmenü werden dann so eingestellt, dass sie mit den in der Registry gespeicherten Optionen übereinstimmen. Auch werden hier am Anfang die möglichen Auflösungen abgefragt, welche vom System unterstützt werden, um das Auflösungs-Dropdown im Optionsmenü mit den möglichen Auflösungen zu füllen. Ebenfalls findet sich hier die Funktion, welche aufgerufen wird, wenn im Optionsmenü alle Einstellungen wieder auf Standard zurückgesetzt werden. Auch wurden hier die Funktionen implementiert, die aufgerufen werden, wenn der Freifahrt-Modus oder eine Lektion gestartet wird.

PedestrianDensityValue: Diese Klasse sorgt nur dafür, dass die Zahl aktualisiert wird, welche rechts neben dem Slider für die Fußgängerdichte im Optionsmenü steht.

NPC_CarDensityValue: Dieses Skript sorgt ebenfalls nur dafür, dass die Zahl aktualisiert wird, welche rechts neben dem Slider für die Fahrzeugdichte im Optionsmenü steht.

ViewDistanceValue: Hierbei handelt es sich auch nur um eine Klasse, welche dafür sorgt, dass die Zahl aktuell gehalten wird, welche sich rechts neben dem Slider für die Sichtweite im Optionsmenü befindet.

PauseMenu: Über diese Klasse wird das Pausemenü implementiert. Wenn während des Spiels auf Escape gedrückt wird, dann wird das Spiel pausiert, der Sound wird gemutet und das Pausemenü wird angezeigt. Es kann fortgefahren werden, die Steuerung kann angesehen werden und ebenfalls kann der Spieler wieder ins Hauptmenü zurückkehren. Falls der Spieler sich nicht im Freifahrt-Modus, sondern in einer Lektion befindet, dann wird ihm im Pausemenü ebenfalls der Erklärungstext der Lektion angezeigt, welcher ihm ebenfalls vor dem Starten der Übung angezeigt wurde.

4 PERFORMANCE

In diesem Kapitel wird evaluiert wie die Performance des Spiels im Allgemeinen ist und wie diese sich ändert mit verschiedenen Grafikqualitäten, Fußgängeranzahlen, Fahrzeuganzahlen und Sichtweiten. Ebenfalls wird getestet wie stark sich die Performance des Spiels verschlechtert, wenn der VR-Modus verwendet wird. Mit Performance sind die Bilder pro Sekunde bzw. FPS (frames per second) gemeint. Eine hohe FPS ist besser, da das Spiel dann flüssiger läuft und angenehmer zu steuern bzw. zu spielen ist. Zuerst wird in diesem Kapitel das System, welches verwendet wurde, beschrieben, damit klar ist, welche Hardware verwendet wurde, um die Performance des Spiels zu testen. Dann wird darauf eingegangen wie die Daten erhoben wurden und dann werden die Ergebnisse dargestellt und diskutiert.

4.1 System

Das System, welches für das Testen der Performance verwendet wurde, verfügt über folgende Spezifikationen:

- Prozessor: Intel Core i5-4460 mit 4 Kernen mit jeweils 3.2 GHz
- Grafikkarte: AMD Radeon R9 380
- Arbeitsspeicher: 8192MB RAM
- Betriebssystem: Windows 10 Pro 64-Bit-Version

Die verwendete Grafikkarte liegt etwas unter den Mindestanforderungen für die Oculus Rift S [22]. In [22] wird als Mindestspezifikation unter anderem eine NVIDIA GTX 1050 Ti oder eine AMD Radeon RX 470 angegeben. Laut GPUBoss [23], wo Grafikkarten einfach miteinander verglichen werden können, ist die AMD Radeon R9 380 etwas weniger leistungsfähig als die AMD Radeon RX 470.

4.2 Methodik

Für das Erfassen der FPS wurde der fertige Build der normalen Version bzw. der VR-Version des Spiels verwendet. Dies ist wichtig zu erwähnen, da das Spiel in Unity merkbar langsamer läuft, als wenn es exportiert wurde. Um die FPS zu erfassen, wurde das Programm Fraps verwendet. Dabei wurde die Benchmark Funktion von Fraps eingesetzt, welche die Performance des Spiels nicht merkbar beeinflusst. Die Benchmark Funktion hat die momentane FPS 60 Sekunden lang in eine CSV-Datei geschrieben. Somit gibt es 60 Samples für jede Einstellung, bei der die Performance getestet wurde. Es wurde immer der Freie-Fahrt-Modus ausgewählt und die Kamera wurde nicht bewegt, sodass gleiche Bedingungen herrschen und die Ergebnisse vergleichbar sind. Beim Auto wurde in der Startposition immer der erste Gang eingelegt und dann Vollgas gegeben, sodass das Auto langsam gerade fuhr mit etwa 25km/h. Dabei wurde das Auto nicht nach rechts oder nach links gelenkt. Somit wurde immer dieselbe Strecke durchfahren für jede Einstellung, bei der die Performance getestet wurde. Wichtig zu erwähnen ist auch, dass im Optionsmenü die Slider für die Fußgängeranzahl, Fahrzeuganzahl und für die Sichtweite auch mit den Pfeiltasten gesteuert werden können, falls z.B. genau eine Sichtweite von 1000 ausgewählt werden muss. Diesen Wert allein mit der Maus auf dem Slider auszuwählen wäre deutlich schwieriger.

4.3 Ergebnisse

Insgesamt wurden 5 verschiedene Prüfungen durchgeführt. Die Fragen, welche interessant sind für das Testen der Performance des Spiels sind folgende:

- Wie wirken sich verschiedene Grafikqualitäten auf die FPS aus im normalen Modus?
- Wie wirken sich verschiedene Grafikqualitäten auf die FPS aus im VR-Modus?
- Wie wirken sich verschiedene Fußgängeranzahlen auf die FPS aus?
- Wie wirken sich verschiedene Fahrzeuganzahlen auf die FPS aus?
- Wie wirken sich verschiedene Sichtweiten auf die FPS aus?

Nach diesen Fragen sind die Ergebnisse ebenfalls strukturiert worden.

4.3.1 Normaler Modus mit verschiedenen Grafikqualitäten

Beim Erheben der FPS wurde die Anzahl der Fußgänger und die Anzahl der Fahrzeuge auf 0 gesetzt. Die höchste Auflösung wurde ausgewählt und die Sichtweite wurde auf 1000 gesetzt. Es werden hier nicht nur die Ergebnisse, sondern auch Screenshots gezeigt, damit die visuellen Änderungen der verschiedenen Grafikqualitäten gesehen werden können.



Abbildung 44: Grafikqualität auf Sehr Niedrig



Abbildung 45: Grafikqualität auf Niedrig



Abbildung 46: Grafikqualität auf Mittel



Abbildung 47: Grafikqualität auf Hoch



Abbildung 48: Grafikqualität auf Sehr Hoch



Abbildung 49: Grafikqualität auf Ultra

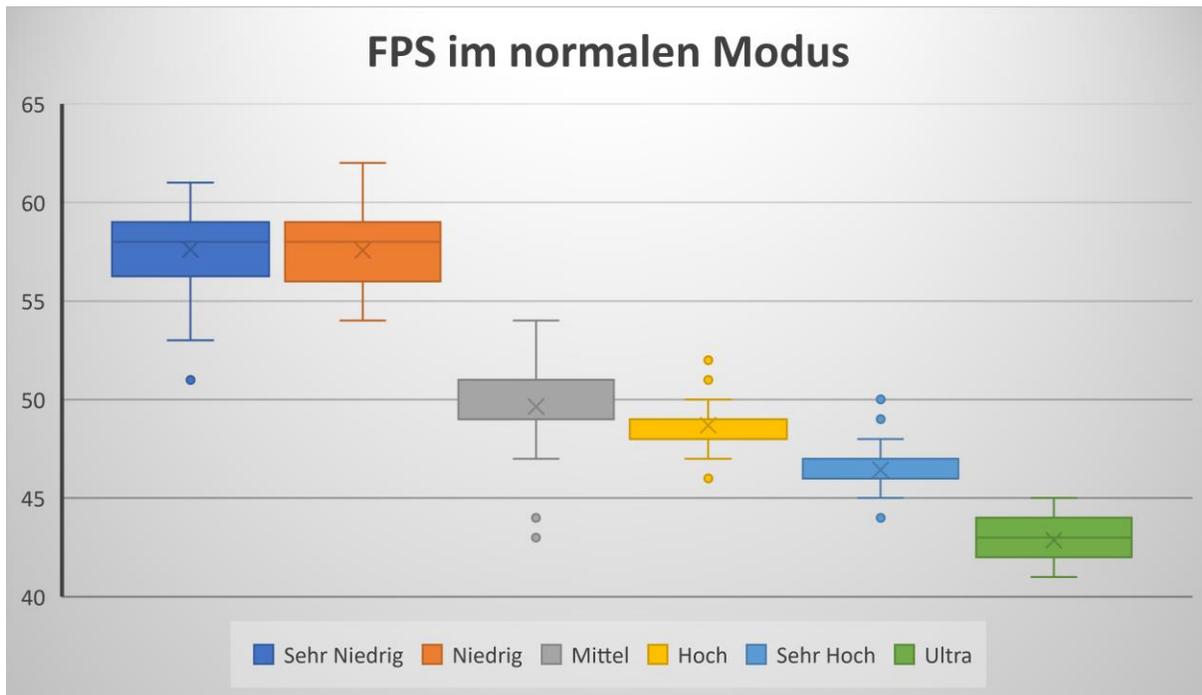


Abbildung 50: FPS im normalen Modus

Wie man sehen kann, werden die FPS niedriger je besser die Grafikqualität ist, was zu erwarten war. Bei genauem Hinsehen fällt auf, dass das Lenkrad und die linke Fahrertüre auf der Grafikqualität „Sehr Niedrig“ etwas unscharf wirken (siehe Abbildung 44). Interessant ist auch, dass bei der Einstellung „Sehr Niedrig“ das große Gebäude links nicht gerendert wurde, da der LOD

(level of detail) bias und der LOD Parameter bei dem Gebäude zu niedrig gesetzt wurde. Je besser die Grafikqualität ist, desto mehr Objekte werden dargestellt, da der LOD bias höher gesetzt ist, unabhängig von der Sichtweite. Die Sichtweite beschreibt im Spiel nur ab welcher Entfernung alles dahinter von der Kamera abgeschnitten wird. Bei der Grafikqualität „Hoch“ sieht man, dass im Hintergrund noch ein Hochhaus gerendert wird. Man sieht auch, dass mit besseren Grafikqualitäten immer mehr Straßenlaternen gerendert werden. Ab der Grafikqualität Mittel gehen die FPS merkbar nach unten, da ab dieser Grafikqualität Schatten dargestellt werden. Mit der Grafikqualität „Ultra“ hat das Spiel immer noch genug FPS, um angenehm spielbar zu sein. Die Schattendistanz wird auch immer größer ab der Grafikqualität „Mittel“. Eine zu niedrige Schattendistanz kann störend wirken, da der Spieler sieht wie die Schatten vor ihm plötzlich dargestellt werden. Dies merkt man besonders, wenn man an einem Ort fährt, der durch ein Hochhaus völlig im Schatten ist.

4.3.2 VR Modus mit verschiedenen Grafikqualitäten

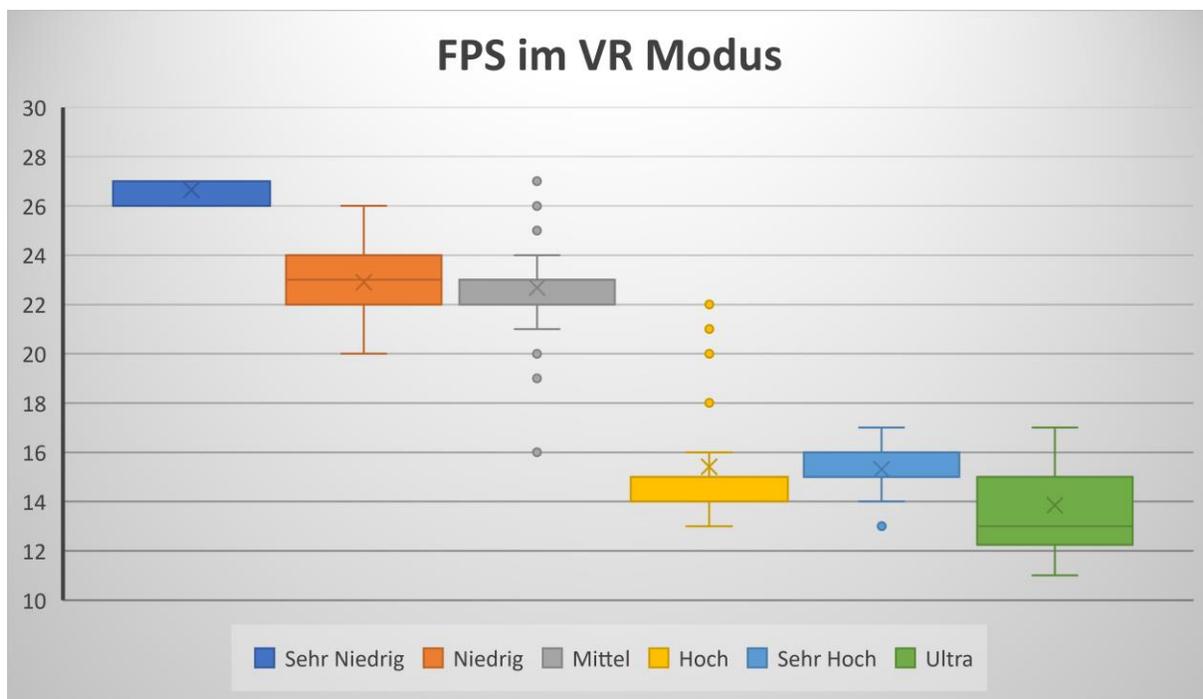


Abbildung 51: FPS im VR Modus

Um die verschiedenen Grafikqualitäten im VR-Modus zu testen, wurde die Anzahl der Fußgänger und die Anzahl Fahrzeuge auf 0 gesetzt und die Sichtweite auf 1000. Auffallend ist, dass die FPS im VR-Modus merkbar niedriger sind als im normalen Modus. Für Spiele in VR wird normalerweise eine höhere FPS benötigt als wenn auf einem flachen Display gespielt wird, da es bei VR sonst leicht zu Motion Sickness kommen kann. Der Grund, dass im VR Modus die FPS niedriger ist als im normalen Modus ist, dass das Spiel für beide Augen gerendert werden muss. Ab der Grafikqualität „Hoch“ ist die FPS zu niedrig, um angenehm spielbar zu sein im VR-Modus auf dem verwendeten System. Komisch ist, dass die FPS nicht bereits bei „Mittel“ nach unten gehen wie im normalen Modus, sondern erst ab „Hoch“. Es ist auch aufgefallen, dass die FPS im VR-Modus sich manchmal zufällig unterscheiden und beim neuerlichen Start diese dann plötzlich höher oder niedriger sind obwohl die Einstellungen nicht geändert wurden.

4.3.3 Verschiedene Fußgängeranzahlen

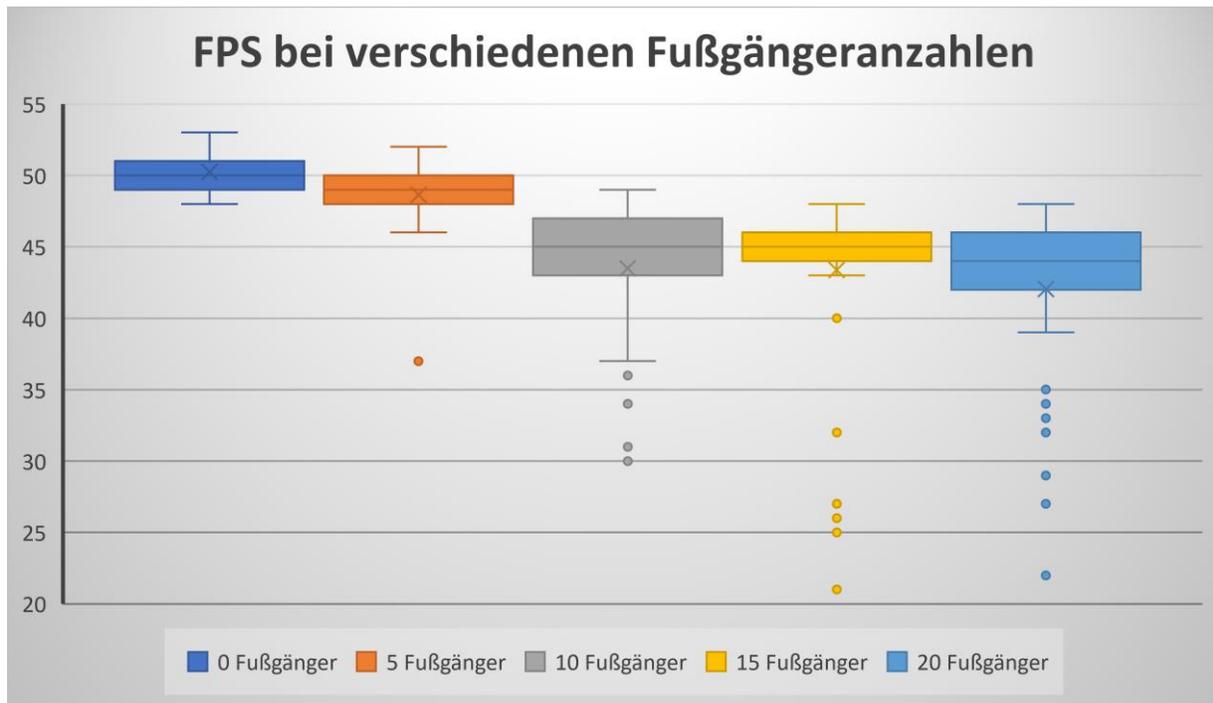


Abbildung 52: FPS bei verschiedenen Fußgängeranzahlen

Bei dieser Datenerhebung wurde die Grafikqualität auf „Mittel“ gesetzt und die Sichtweite auf 1000. Die Anzahl der Fahrzeuge wurde auf 0 gesetzt. Ebenfalls wurde nicht der VR-Modus verwendet, sondern der normale Modus. Die einzige Option, die verändert wurde, war die Fußgängeranzahl. Auffällig ist, dass es zu starken Frameeinbrüchen kommt, je höher die Anzahl der Fußgänger eingestellt war. Beim Implementieren ist dies ebenfalls aufgefallen, dass zu viele Fußgänger nicht nur insgesamt die FPS nach unten ziehen, sondern auch zu kurzfristigen starken Frameeinbrüchen führen können. Dies geschieht aber nicht alle 5 Sekunden, wie man vermuten könnte, da das Pooling System alle 5 Sekunden prüft, ob Fußgänger außerhalb der *spawnZone* sind, um die Fußgänger gegebenenfalls neu zu positionieren. Die starken Ruckler passieren häufiger als alle 5 Sekunden. Auf schwachen Systemen sollte daher eher eine niedrige Fußgängeranzahl gewählt werden oder ganz ausgeschaltet werden, wenn nicht geübt werden möchte, wie man sich z.B. als Autofahrer verhalten muss, wenn ein Fußgänger einen nicht geregelten Fußgängerübergang überqueren möchte. Eine niedrige Fußgängeranzahl oder gar keine Fußgänger könnten jedoch die Immersion schwächen, da die Gehsteige in der Spielwelt sehr leer wirken würden. Gerade da sich zu viele Fußgänger schlecht auf die Performance auswirken können, wurde das Pooling System implementiert, damit sich Fußgänger nicht an Stellen befinden wo der Spieler sie gar nicht zu Gesicht bekommt.

4.3.4 Verschiedene Fahrzeuganzahlen

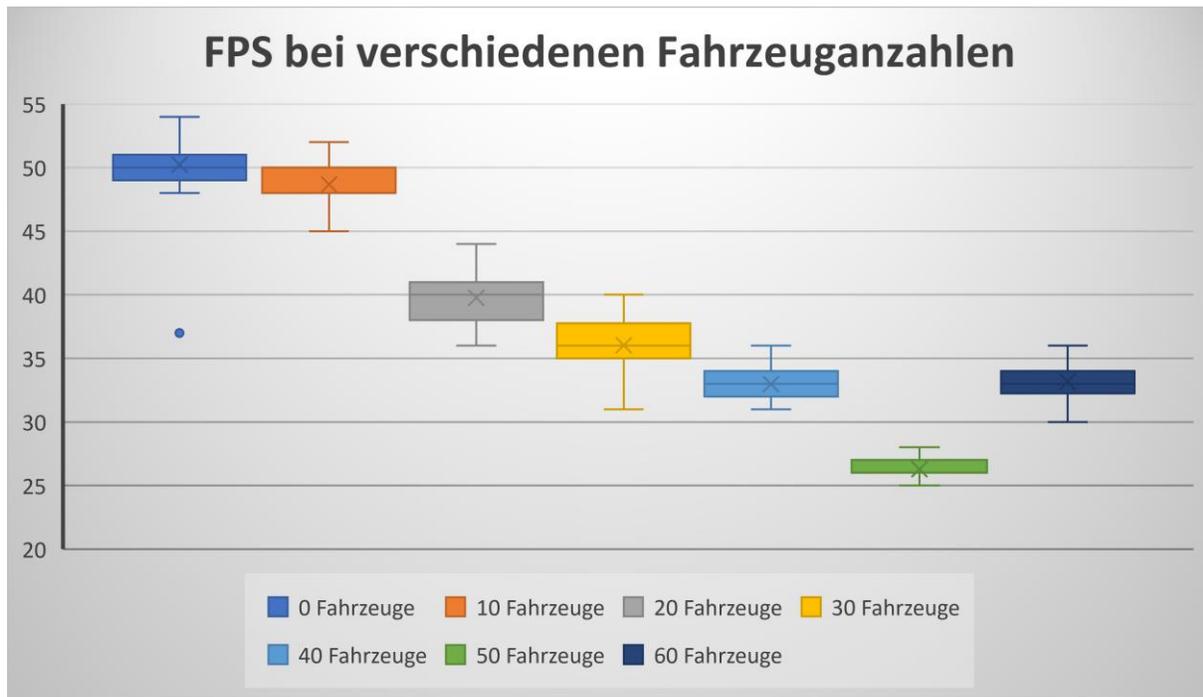


Abbildung 53: FPS bei verschiedenen Fahrzeuganzahlen

Beim Erheben dieser Daten wurde die Grafikqualität ebenfalls auf „Mittel“ gestellt und die Sichtweite auf 1000. Ebenfalls wurde nicht der VR-Modus verwendet, sondern der normale Modus. Die Anzahl der Fußgänger wurde immer auf 0 gesetzt. Die einzige Option, welche geändert wurde, war die Anzahl der Fahrzeuge. Wie zu erwarten gehen die FPS runter, je mehr Fahrzeuge sich in der Spielwelt befinden. Der Unterschied zwischen 0 und 60 Fahrzeugen ist jedoch kleiner als erwartet. Bei 60 Fahrzeugen wurden mehr FPS gemessen als bei 50 Fahrzeugen, jedoch ist nicht klar warum dies passiert ist. Es könnte sein, dass beim Überprüfen der FPS in diesem Fall weniger Fahrzeuge im Blickwinkel des Spielers waren, sodass diese nicht gerendert werden mussten, was für eine bessere Performance sorgte. Es werden nämlich Objekte nicht gerendert, wenn diese sich nicht im Blickfeld des Spielers befinden oder wenn diese von anderen Objekten z.B. von Häusern verdeckt werden. Ebenfalls ist aufgefallen, dass die FPS eher konstant bleiben und es nicht manchmal zu starken Frameeinbrüchen kommt wie beim Testen verschiedener Fußgängeranzahlen. Es ist davon auszugehen, dass ein Fahrzeug weniger Performance kostet als ein Fußgänger, da ein Fahrzeug nicht animiert ist, da sich nur die Reifen drehen können und das Bremslicht und die Blinker eingeschaltet werden können. Fußgänger hingegen sind animiert, wobei zu erwähnen ist, dass diese nicht extra animiert werden mussten, sondern, dass dies bereits durch das UMA System geschieht. Eine hohe Anzahl an Fahrzeugen ist in einer Fahrsimulation wichtiger als eine hohe Anzahl an Fußgängern, da Fahrzeuge beim Üben verschiedener Situationen, wie z.B. beim sicheren Wechsel der Spur öfter benötigt werden. Es ist auch aufgefallen, dass gerade in der Südstadt besonders viele Fahrzeuge benötigt werden, damit die Straßen nicht leer wirken. Dies liegt daran, dass es in der Südstadt besonders viele Straßen gibt und ebenso viele der Straßen zwei- oder dreispurig in beide Richtungen sind. Die anderen Orte der Spielwelt wirken weniger leer, da dieselbe Anzahl an Fahrzeugen auf weniger Straßen verteilt wird.

4.3.5 Verschiedene Sichtweiten

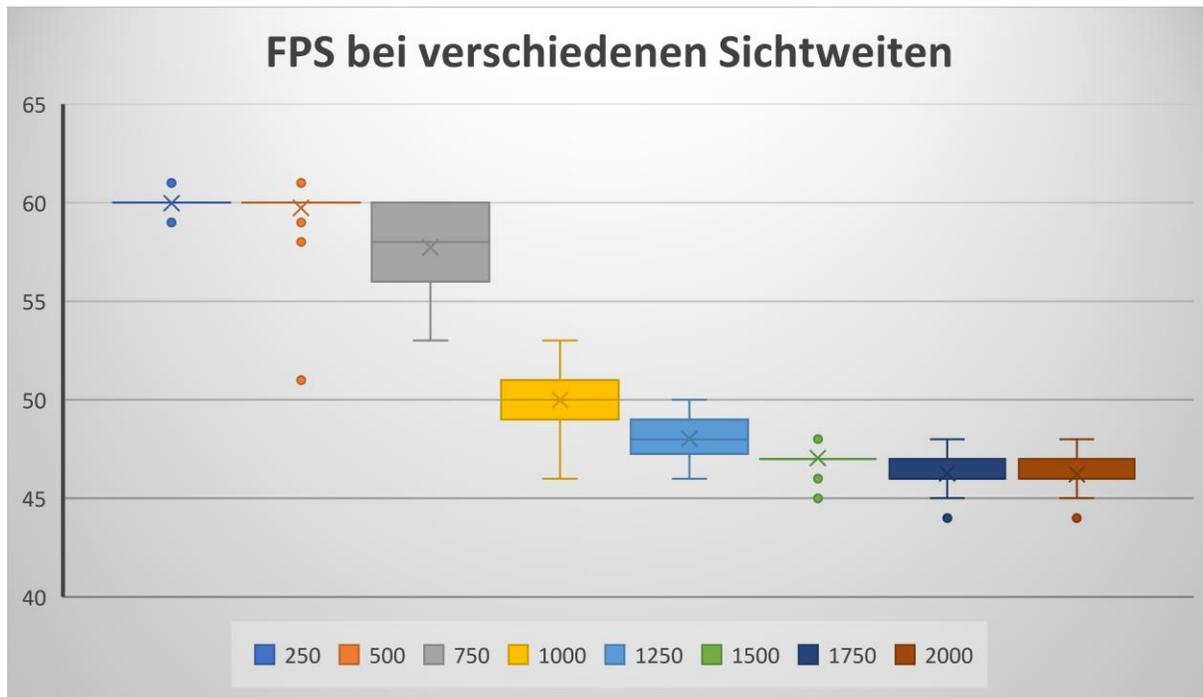


Abbildung 54: FPS bei verschiedenen Sichtweiten

Die Sichtweiten sind in Metern angegeben. Bei der Erhebung dieser Daten wurde die Grafikqualität auf „Mittel“ gesetzt. Die Anzahl der Fußgänger und die Anzahl der Fahrzeuge wurde auf 0 gesetzt. Die einzige Option, welche geändert wurde, war die Sichtweite. Es wurde ebenfalls nicht der VR-Modus, sondern der normale Modus verwendet. Wie zu erwarten, geht die FPS nach unten je höher die Sichtweite eingestellt ist. Interessant ist, dass ab einer Sichtweite von 1500 Metern die FPS sich nicht mehr stark verschlechtert haben. Vermutlich liegt es daran, dass ab 1500 Metern bereits die ganze Stadt in der Sichtweite war und bei noch höheren Sichtweiten, nur leeres Terrain gerendert werden musste, welches weniger Rechenleistung benötigt. Die Sichtweite kann nicht nur im Optionsmenü geändert werden, sondern ebenfalls mit der Taste F11 verringert oder mit der Taste F12 erhöht werden. Besonders auf der Autobahn oder auf Freilandstraßen ist aufgefallen, dass eine Erhöhung der Sichtweite oft möglich ist, da die Performance dort sehr gut ist, da es weniger Objekte gibt als z.B. in Ortsgebieten. Es ist auch aufgefallen, dass die Spielwelt realer wirkt, wenn die Sichtweite höher gesetzt wird, besonders an Orten wo die hohe Sichtweite ausgenutzt werden kann, also wo keine Gebäude die Sicht versperren.

5 EVALUIERUNG

Für die Evaluierung von Virtual Driving School wurden insgesamt 9 Personen befragt. Davon haben 3 Personen das Spiel sowohl im normalen Modus als auch im VR-Modus mit dem Logitech G29 Lenkrad getestet, während die anderen 6 Personen nur den normalen Modus mit Maus und Tastatur getestet haben. Alle Personen mussten einen Fragebogen ausfüllen, welcher allgemeine Fragen zur Person beinhaltet, Fragen zu den Lektionen und zum Freie-Fahrt-Modus sowie sonstige Fragen. Der Fragebogen beinhaltet sowohl Skalenfragen als auch offene Fragen. Bei den Skalenfragen konnten die Antworten 1 bis 6 gegeben werden, wobei 1 die positivste Antwort und 6 die negativste Antwort ist. Zum Beispiel steht 1 für Sehr gut, Sehr oft, Sehr professionell usw. während 6 für Sehr schlecht, Sehr selten oder Sehr unprofessionell usw. steht. Die Skala wurde bewusst von 1 bis 6 gewählt, damit keine neutrale Antwort möglich ist. Im Fragebogen werden bewusst nur die Antworten 1 und 6 benannt. Was die Antworten dazwischen bedeuten, mussten die teilnehmenden Personen selbst entscheiden. Falls der VR-Modus getestet wurde, wurden Fragen gestellt, welche aus [24] stammen. Ebenfalls wurden Fragen zur Steuerung mit dem Logitech G29 gestellt, falls es verwendet wurde. Der gesamte Fragebogen sieht wie folgt aus:

Allgemeine Fragen

Wie alt bist du?

Bist du männlich oder weiblich?

Hast du einen Führerschein?

Wenn ja, seit wie vielen Jahren?

Wie oft fährst du mit dem Auto auf einer Skala von 1 - 6? (1=Sehr oft, ..., 6=Sehr selten)

Wie würdest du deine Fahrfähigkeiten beschreiben auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

Lektionen

Welche Lektionen hast du im Spiel ausprobiert?

Wie findest du sind die Lektionen gemacht auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

Was findest du gut bzw. schlecht an den Lektionen?

Konntest du was Neues lernen bei den Lektionen? Falls ja, was?

Findest du es gut, dass man als Spieler selbst die Lektionen über das Hauptmenü beenden muss (also, dass es nicht von selbst passiert)?

Freie Fahrt Modus

Wie findest du den Freie-Fahrt-Modus auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

Welche Gangschaltung hast du verwendet? Automatisch, manuell oder hast du beide ausprobiert?

Wie findest du sind die anderen Autofahrer gemacht auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

Wie findest du sind die Fußgänger gemacht auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

Wie findest du ist die Spielwelt gemacht auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

Was findest du gut bzw. schlecht?

Sonstige Fragen

Hast du dich im Menü zurechtgefunden?

Hast du dich mit der Steuerung zurechtgefunden?

Wieviel Spaß macht das Spiel auf einer Skala von 1 - 6? (1=Sehr viel, ..., 6=Sehr wenig)

Wie professionell wirkt das Spiel auf einer Skala von 1 - 6? (1=Sehr professionell, ..., 6=Sehr unprofessionell)

Wie realistisch findest du das Fahrverhalten des eigenen Autos auf einer Skala von 1 - 6? (1=Sehr realistisch, ..., 6=Sehr unrealistisch)

Wie findest du die Grafik des Spiels auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

Wie flüssig und ruckelfrei lief das Spiel auf einer Skala von 1 - 6? (1=Sehr flüssig, ..., 6=Sehr ruckelig)

Wie stark ist dein Computer auf einer Skala von 1 - 6? (1=Sehr stark, ..., 6=Sehr schwach)

Wie lange waren die Ladezeiten des Spiels bei dir ungefähr?

Wie könnte man das Projekt deiner Meinung nach verbessern?

Denkst du das Spiel eignet sich, um damit Autofahren zu lernen bzw. zu üben? Warum?

Hast du das Spiel mit der Maus und Tastatur gespielt oder mit dem Logitech G29 Lenkrad oder beides?

Hast du das Spiel mit dem normalen Bildschirm gespielt oder mit einem VR-Headset oder beides?

Anmerkungen?

Virtuelle Realität Fragen (falls VR getestet wurde)

In der computererzeugten Welt hatte ich den Eindruck, dort gewesen zu sein... (1=Sehr stark, ..., 6=Überhaupt nicht)

Ich hatte das Gefühl, dass die virtuelle Umgebung hinter mir weitergeht. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

Ich hatte das Gefühl, nur Bilder zu sehen. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

Ich hatte nicht das Gefühl, in dem virtuellen Raum zu sein. (1=Hatte das Gefühl, ..., 6=Hatte nicht das Gefühl)

Ich hatte das Gefühl, in dem virtuellen Raum zu handeln statt etwas von außen zu bedienen. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

Ich fühlte mich im virtuellen Raum anwesend. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

Wie bewusst war Ihnen die reale Welt, während Sie sich durch die virtuelle Welt bewegten (z.B. Geräusche, Raumtemperatur, andere Personen etc.)? (1=Extrem bewusst, ..., 6=Unbewusst)

Meine reale Umgebung war mir nicht mehr bewusst. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

Ich achtete noch auf die reale Umgebung. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

Meine Aufmerksamkeit war von der virtuellen Welt völlig in Bann gezogen. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

Wie real erschien Ihnen die virtuelle Umgebung? (1=Vollkommen real, ..., 6=Gar nicht real)

Wie sehr glich Ihr Erleben der virtuellen Umgebung dem Erleben einer realen Umgebung? (1=Vollständig, ..., 6=Überhaupt nicht)

Wie real erschien Ihnen die virtuelle Welt? (1=Nicht zu unterscheiden von der realen Welt, ..., 6=Wie eine vorgestellte Welt)

Die virtuelle Welt erschien mir wirklicher als die reale Welt. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

Logitech G29 Lenkrad Fragen (falls es getestet wurde)

Hast du dich mit der Lenkradsteuerung zurechtgefunden?

Bevorzugst du das Logitech G29 oder Maus und Tastatur?

Wie realistisch ist die Lenkradsteuerung gemacht auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

Was gefällt dir bzw. was gefällt dir nicht an der Lenkradsteuerung?

5.1 Ergebnisse

Hier werden die wichtigsten Ergebnisse aus der Evaluierung zusammengefasst. Gegliedert werden die Ergebnisse wie im Fragebogen auch in die verschiedenen Teile: Allgemeine Fragen, Lektionen, Freie Fahrt Modus, Sonstige Fragen, Virtuelle Realität Fragen und Logitech G29 Lenkrad Fragen.

5.1.1 Allgemeine Fragen

Die Teilnehmer waren 15, 55, 57, 27, 30, 27, 26, 25 und 34 Jahre alt. Es wurden 7 Männer und 2 Frauen befragt. Von den befragten Personen hatten 3 keinen Führerschein, während die anderen 6 Personen einen Führerschein hatten. Die Personen die einen Führerschein hatten, hatten diesen im Durchschnitt seit etwa 10 Jahren.

5.1.2 Lektionen

Die teilnehmenden Personen haben nicht alle Lektionen ausprobieren müssen, sondern nur so viele wie sie wollten, da der Zeitaufwand zu hoch gewesen wäre alle Lektionen auszuprobieren und im Detail zu bewerten. Im Fragebogen wurde nachgefragt, welche Lektionen ausprobiert wurden und im Durchschnitt wurden etwa 4 Lektionen ausprobiert. Bei der Frage wie die Lektionen gemacht sind auf einer Skala von 1 bis 6, ergibt der Durchschnitt aller Antworten 2,5 mit einer Standardabweichung von etwa 0,94. Bei der Frage was die teilnehmenden Personen gut bzw. schlecht an den Lektionen fanden, gab es verschiedene Antworten. Zum Beispiel war einem Teilnehmer nicht bewusst in welcher Reihenfolge die Lektionen zu machen sind, da sie nicht nummeriert sind. Hier muss jedoch erwähnt werden, dass es keine wirkliche Reihenfolge gibt, da die Lektionen großteils unabhängig voneinander sind mit Ausnahme, dass zuerst gelernt werden sollte wie man ein Auto anfährt und schaltet bevor die übrigen Lektionen gespielt werden. Von zwei teilnehmenden Personen, welche keinen Führerschein besitzen, wurde angegeben, dass im Spiel nicht erwähnt wird, wie man ein Auto mit manueller Schaltung anhält ohne, dass der Motor abstirbt. Eine Lektion, welche das Anhalten eines Autos erklärt, wäre somit sinnvoll gewesen oder es hätte in der ersten Lektion „Anfahren und Schalten“ erklärt werden müssen. Von einem Teilnehmer wurde kritisiert, dass man während einer Lektion nicht aus dem Übungsplatz herausfahren kann. Dabei handelt es sich jedoch um eine bewusste Designentscheidung, da während einer Lektion im Pausemenü der Erklärungstext der Lektion steht und da während einer Lektion keine Fußgänger oder fahrenden Autos in der Spielwelt sind, um die Performance zu verbessern. Falls man in der Spielwelt umherfahren möchte, muss man den Freie-Fahrt-Modus auswählen. Positiv fanden mehrere Teilnehmer, dass die Instruktionen einfach gehalten sind und während der Lektion im Pausemenü nachgelesen werden können. Es wurde aber auch gesagt, dass es besser wäre, wenn die einzelnen Schritte während einer Lektion angezeigt werden würden und wenn das Spiel einem Feedback geben würde, was richtig oder falsch gemacht wurde. Dies ist während der Implementierung bewusst gewesen, jedoch wurde darauf verzichtet, da der Aufwand als zu groß eingeschätzt wurde und damit stattdessen mehr Lektionen eingebaut werden können. Bei der Frage, ob und was die Teilnehmer lernen konnten bei den Lektionen, haben die Teilnehmer ohne Führerschein geantwortet, dass sie sich das durch die Lektionen vermittelte Wissen aneignen konnten. Die Teilnehmer mit Führerschein wussten das Meiste natürlich schon, jedoch hat ein Teilnehmer geantwortet, dass die Bewegungsabläufe wieder bewusster geworden sind. Es wurde auch gefragt ob die Teilnehmer es gut finden, dass die Lektionen selbst über das Hauptmenü

beendet werden müssen. Darauf haben die meisten mit Ja geantwortet, jedoch haben zwei Personen gesagt, dass es besser wäre, wenn man eine Rückmeldung bekommen würde, ob man die Übungen richtig macht.

5.1.3 Freie Fahrt Modus

Bei der Skalenfrage wie die Teilnehmer den Freie Fahrt Modus finden, ergibt der Durchschnitt der Antworten etwa 1,9 mit einer Standardabweichung von etwa 0,6. Bei der Frage welche Gangschaltung verwendet wurde, haben 4 Personen mit „Manuell“ geantwortet, eine Person mit „Automatisch“ und 4 Personen mit „Beide“. Bei der Skalenfrage wie die Autofahrer gemacht sind, ergibt der Durchschnitt der Antworten 2 mit einer Standardabweichung von 1. Bei der Skalenfrage wie die Fußgänger gemacht sind, ergibt der Durchschnitt der Antworten etwa 1,9 mit einer Standardabweichung von etwa 0,93. Bei der Skalenfrage wie die Spielwelt gemacht ist, ergibt der Durchschnitt der Antworten etwa 1,7 mit einer Standardabweichung von etwa 0,87. Bei der Frage, was als gut oder als schlecht empfunden wurde im Freie-Fahrt-Modus gab es verschiedene Antworten. Kritisiert wurde, dass manche Straßen nicht nahtlos ineinander übergehen oder, dass das eigene Auto manchmal umkippt. Wenn das Auto im Spiel umkippt und sich nicht mehr aufrichtet, dann kann man nur den Freie-Fahrt-Modus neustarten. Falls man aber an einer bestimmten Stelle in der Spielwelt z.B. auf der Autobahn üben möchte, müsste man extra wieder dorthin fahren. Eine Lösung dieses Problems wäre, dass das Auto, wenn es auf dem Kopf oder auf der Seite liegt, nach einigen Sekunden vom Spiel wieder richtig orientiert wird. Kritisiert wurde ebenfalls das Fußgänger mit dem Auto nur „verschoben“, nicht jedoch überfahren werden können. Kritisiert wurde auch, dass Autofahrer und Fußgänger mit der Umwelt wenig interagieren, also, dass z.B. nichts passiert bei einem Unfall. Ebenfalls hat ein Teilnehmer erwähnt, dass die Lenkung bei hohen Geschwindigkeiten mit der Tastatur zu stark ist und schwächer sein sollte. Positiv hervorgehoben wurde von einem Teilnehmer, dass das Auto viele Details hat, also, dass es z.B. eine Hupe gibt, Scheibenwischer und eine Warnblinkanlage. Ein Teilnehmer hat erwähnt, dass er während des Fahrens (im Freie-Fahrt-Modus aber auch bei Lektionen) gerne Hintergrundmusik gehört hätte.

5.1.4 Sonstige Fragen

Bei der Frage, ob sich die Teilnehmer im Menü zurechtgefunden haben, haben alle Teilnehmer mit Ja geantwortet. Bei der Frage ob sich die Teilnehmer mit der Steuerung zurechtgefunden haben, haben die meisten Teilnehmer mit Ja geantwortet. Ein Teilnehmer hat im Fragebogen selbst die Frage gestellt, wie die Kupplung langsam losgelassen wird, wenn man die Tastatur zum Steuern verwendet. Dabei zeigt er eine Schwäche der Tastatursteuerung auf, nämlich, dass die Kupplung, welche mit der Leertaste betätigt wird, auf einer Tastatur nur gedrückt oder nicht gedrückt sein kann. Die Kupplung kann somit nicht langsam losgelassen werden, außer es wird das Logitech G29 verwendet mit den dazugehörigen Pedalen. Wenn die Tastatursteuerung verwendet wird, dann kann der Motor nicht absterben durch zu schnelles Loslassen der Kupplung. Ein Teilnehmer hat vorgeschlagen, dass die Steuerung mithilfe eines Tutorials erklärt werden sollte. Ebenfalls wurde kritisiert, dass das Starten des Motors mit der Eingabetaste belegt ist und diese Taste somit bei den meisten Menschen mit der rechten Hand betätigt werden muss, welche im Spiel zum Umsehen mit der Maus verwendet wird. Ein Teilnehmer meinte, die Steuerung mit manueller Gangschaltung sei zu kompliziert und die Lenkung zu stark bei hohen Geschwindigkeiten, wodurch man leicht die Kontrolle über das Auto verliert.

Bei der Skalenfrage wie viel Spaß das Spiel macht, ergibt der Durchschnitt der Antworten etwa 2,1 mit einer Standardabweichung von etwa 0,78. Bei der Skalenfrage wie professionell das Spiel wirkt, ergibt der Durchschnitt der Antworten etwa 2,2 mit einer Standardabweichung von etwa 1,2. Bei der Skalenfrage wie realistisch das Fahrverhalten des eigenen Autos empfunden wurde, ergibt der Durchschnitt der Antworten etwa 2,7 mit einer Standardabweichung von etwa 0,87. Bei der Skalenfrage wie die Grafik des Spiels empfunden wurde, ergibt der Durchschnitt der Antworten etwa 2,2 mit einer Standardabweichung von etwa 0,83. Bei der Skalenfrage wie flüssig und ruckelfrei das Spiel lief, ergibt der Durchschnitt der Antworten etwa 1,6 mit einer Standardabweichung von etwa 0,73. Bei der Skalenfrage wie stark der eigene Computer eingeschätzt wurde, ergibt der Durchschnitt der Antworten etwa 2,6 mit einer Standardabweichung von etwa 0,88.

Bei der Frage wie lange die Ladezeiten des Spiels ungefähr waren, wurde meistens mit wenigen Sekunden geantwortet, wobei eine Person angegeben hat, dass die Ladezeiten beim ersten Starten einer Lektion oder des Freie-Fahrt-Modus länger sind als beim wiederholten Starten. Dies ist während der Implementierung ebenfalls aufgefallen. Bei der Frage, wie das Projekt verbessert werden könnte gab es verschiedene Antworten. Zum Beispiel könnte die Grafik verbessert werden. Ebenfalls gab es die Antwort, dass man während des Spielens interaktives Feedback bekommen soll, z.B. falls der Motor abwürgt, warum dies passiert ist. Es gab von einer Person auch den Verbesserungsvorschlag mehr Soundeffekte einzufügen z.B., wenn man gegen etwas fährt oder, dass man Punkte sammelt, wenn man richtig fährt und Punkte verliert, wenn man Fehler macht. Ebenfalls kam die Antwort, dass beim Rückwärtsfahren die „S“-Taste manchmal etwas länger gedrückt werden muss, bevor das Auto beginnt zurückzufahren. Bei der Frage, ob die Personen denken, dass das Spiel sich eignet, um damit Autofahren zu üben gab es verschiedene Antworten. Viele Teilnehmer sagten, es sei ganz gut, um sich Basiswissen anzueignen, jedoch ist es besser, wenn ein Lenkrad mit Pedalen und Gangschaltung verwendet wird statt Maus und Tastatur. Bei der Frage welche Methode der Eingabe verwendet wurde, haben 6 Personen mit „Maus und Tastatur“ geantwortet, 2 Personen mit „Logitech G29“ und eine Person hat beides ausprobiert. Bei der Frage welche Methode der Ausgabe verwendet wurde, haben 6 Personen mit „Bildschirm“ geantwortet und 3 Personen mit „Beides“ (Bildschirm und VR-Headset). Bei der Frage ob es noch Anmerkungen gibt, hat eine Person erwähnt, dass es gut wäre, wenn man im VR Modus virtuelle Beine sehen würde, wenn man nach unten sieht und wenn man seine eigenen Hände in der Spielwelt sehen würde.

5.1.5 Virtuelle Realität Fragen (falls VR getestet wurde)

Der VR-Modus wurde von 3 Personen getestet und es wurden insgesamt 14 Fragen gestellt. Es wurden nur Skalenfragen gestellt und die Fragen stammen aus [24]. Insgesamt kann man sagen, dass den 3 Teilnehmern der VR-Modus gefallen hat. Die Teilnehmer hatten den Eindruck in der computererzeugten Welt gewesen zu sein und fühlten sich im virtuellen Raum anwesend. Auf die Antworten zu jeder einzelnen Frage wird hier nicht näher eingegangen, da bei Interesse alle Antworten im Anhang am Ende dieser Arbeit nachgelesen werden können.

5.1.6 Logitech G29 Lenkrad Fragen (falls es getestet wurde)

Die Steuerung mit dem Logitech G29 wurde von 3 Personen getestet. Bei der Frage, ob die Personen sich mit der Lenkradsteuerung zurecht gefunden haben kamen gemischte Antworten. Eine Person sagte, dass es schwierig war das Logitech G29 gemeinsam mit dem VR-Headset zu

verwenden, da man die Tasten auf dem Logitech G29 Lenkrad nicht sehen kann, um z.B. den Motor einzuschalten oder um die Handbremse zu lösen. Bei der Frage, ob das Logitech G29 oder die Steuerung per Maus und Tastatur besser ist, hat eine Person geantwortet, dass es egal sei, während die anderen beiden Personen meinten, das Lenkrad sei besser, da es realer ist. Bei der Skalenfrage wie realistisch die Lenkradsteuerung gemacht ist auf einer Skala von 1 bis 6, kamen die Antworten 4, 1 und 1. Bei der Frage was den Teilnehmern an der Lenkradsteuerung gefällt oder nicht gefällt kam die Antwort, dass es besser wäre, wenn das Lenkrad sich zur Ausgangsposition zurückdrehen würde. Ebenfalls wurde kritisiert, dass das Lenkrad sowie das Gaspedal zu unempfindlich sind. Auch fand ein Teilnehmer, dass das Kupplungspedal sehr langsam losgelassen werden muss, damit das Auto nicht abstirbt.

6 DISKUSSION

In diesem Kapitel wird diskutiert, wie Virtual Driving School verbessert werden könnte.

6.1 Motion Feedback

Das Simulieren von Autofahrten unter Einsatz von VR-Headsets ist ein Problem, welches bereits öfter behandelt wurde. Erkennbar ist, dass viele Fahrsimulationen versuchen die Immersion durch Motion Feedback zu erhöhen. Fahrsimulatoren ohne Motion Feedback können leichter zu Symptomen von Motion Sickness führen [25]. Wenn das Motion Feedback genau richtig kalibriert ist, dann würde man in der Fahrsimulation weniger Motion Sickness haben, da der Körper die Bewegungskräfte spürt, welche er erwartet, wenn man in einem Auto fährt. Motion Plattformen können auch als Privatperson sehr leicht erworben werden, haben jedoch einen hohen Preis und benötigen viel Platz. Ebenfalls könnte es ein Problem geben, wenn man eine Motion Plattform zusammen mit einem VR-Headset verwendet, da die Bewegungen der Motion Plattform vom VR-Headset subtrahiert werden müssen [15]. Sonst würden Bewegungen der Motion Plattform von der Fahrsimulation als Bewegung des Kopfes wahrgenommen werden. Deswegen sind VR-Headsets oft nicht geeignet für den Gebrauch in z.B. einem Bus oder Flugzeug. Virtual Driving School so zu ergänzen, dass es zusammen mit einer Motion Plattform verwendet werden kann wäre zu aufwendig, wäre jedoch eine Möglichkeit das Projekt zukünftig zu verbessern. Es gibt jedoch die Möglichkeit bei Virtual Driving School das Force Feedback System einzusetzen, welches vom Logitech G29 Lenkrad unterstützt wird. Somit hätte das Spiel ebenfalls haptisches Feedback, was die Immersion erhöhen könnte. Jedoch sollte es nicht übertrieben werden, da es den Spieler stören könnte und es sollte per Option abschaltbar sein. Zum Beispiel könnte das Lenkrad leicht vibrieren, wenn man auf einem Boden fährt, der uneben ist. Außerdem könnte z.B. das Lenkrad etwas schwerfälliger zu drehen sein, wenn das virtuelle Auto steht aufgrund der stärkeren Reibung an den Reifen. Aufgrund von Zeitmangel und da Force Feedback schnell störend wirken könnte, wurde jedoch darauf verzichtet. Jedoch wäre es eine Möglichkeit, das Projekt zu verbessern.

6.2 Intelligent Tutoring System

In [15] wurde ein ITS (Intelligent Tutoring System) entwickelt und in der Fahrsimulation eingesetzt, welches die Fahrfähigkeiten des Spielers misst und ihm anhand dessen eine Route vorschlägt, in der Situationen vorkommen, die ihm dabei helfen, möglichst optimal zu lernen. Dies wäre sicher auch bei Virtual Driving School eine gute Idee gewesen. In Virtual Driving School gibt es neben den Lektionen nur einen Freie-Fahrt-Modus, in dem der Spieler frei herumfahren kann, ohne, dass ihm Tipps gegeben werden oder eine Route vorgeschlagen wird. Das Implementieren eines solchen Systems würde wohl machbar sein, würde jedoch einiges an Aufwand bedeuten. Ebenfalls könnte ein solches System von manchen Spielern schnell als störend empfunden werden, weswegen es abschaltbar sein sollte.

6.3 Gefahren beim Autofahren in VR

Eine Gefahr bei Fahrsimulationen ist, dass der Spieler sich zu sehr an die Fahrsimulation gewöhnen könnte und dabei bestimmte Automatismen wie z.B. beim Gangschalten entwickelt, welche dann im realen Auto ebenfalls angewendet werden, obwohl das reale Auto anders funktioniert. Dies könnte man lösen, indem man den Spieler z.B. in den Optionen entscheiden lässt, wie die Gänge angeordnet sein sollen, damit diese wie beim eigenen richtigen Auto sind. Dem Spieler die Möglichkeit zu geben, die Steuerung mit Maus und Tastatur oder dem Lenkrad an die eigenen Bedürfnisse anzupassen, wäre ein Feature, welches das Spiel verbessern könnte. Ebenfalls gibt es die Gefahr, dass man das Autofahren in der Realität mit dem Autofahren in VR verwechselt und sich in Gefahr bringt. Man könnte argumentieren, dass dies bei einer sehr realistisch aussehenden Fahrsimulation leichter passieren würde als bei einer Simulation mit schlechter Grafik.

6.4 Performance bei Virtual Reality

Eine wichtige Erkenntnis ist auch, dass für eine angenehme Fahrsimulation in VR möglichst viele Bilder pro Sekunde besser sind, um die Gefahr von Motion Sickness zu verringern. Leider sind die Anforderungen in VR um einiges höher als wenn die Ausgabe auf einem flachen Bildschirm erfolgt. Bei Virtual Driving School sind die Bilder pro Sekunde im VR-Modus auch deutlich niedriger als im „flachen“ Modus. Eine Möglichkeit das Projekt zu verbessern, wäre indem das Projekt noch stärker optimiert wird, um mehr Performance rauszuholen, damit das Spiel auch im VR-Modus spielbar ist auf Systemen, welche weniger leistungsfähig sind.

6.5 Nachtmodus

Bei Virtual Driving School wurde kein Nachtmodus implementiert, da der Fokus auf anderen Themen lag und die Zeit nicht gereicht hätte. Das Spielerauto hat zwar zwei Abblendlichter, beim Ein- und Ausschalten der Abblendlichter gab es jedoch vor allem im VR-Modus starke Ruckler. Daher wurden die Lichter entfernt. Stattdessen wird nur die Textur der Abblendlichter weiß gefärbt, wenn das Abblendlicht eingeschalten wird. Im Spiel gibt es ebenfalls Straßenlaternen an jeder Straße im Ortsgebiet, jedoch gab es auch hier das Problem, dass das Spiel ruckeln würde, wenn diese eingeschalten werden, da es dann zur gleichen Zeit sehr viele Lichtquellen geben würde. Dies liegt vermutlich daran, dass die Berechnung für jedes Licht im Spiel während der Laufzeit stattfindet. In Unity gibt es die Möglichkeit Light Baking zu verwenden. Hierbei wird Licht, welches sich nicht ändert, in die Textur „hineingebacken“. Dies geschieht über das Erstellen von Light Maps. Dieser Vorgang hat jedoch aufgrund der großen Spielwelt in Virtual Driving School sehr lange gedauert und blieb ab einem gewissen Punkt hängen und musste abgebrochen werden. Light Baking hätte verwendet werden können für statische Lichter wie z.B. Straßenlaternen, nicht jedoch für z.B. fahrende Autos. Da die Entscheidung getroffen wurde, keinen Nachtmodus zu implementieren, wurden die Lichtquellen wieder aus den Straßenlaternen entfernt. Falls man einen Nachtmodus implementieren würde, müssten ebenfalls die NPC Autos über Abblendlichter verfügen, welche eingeschalten werden können. Die NPC Autos würden durch das Licht zwar nicht besser „sehen“, da die NPC Autos nur Wegpunkten folgen, jedoch würden NPC Autos ohne Abblendlicht vom Spieler schlechter gesehen werden, vor allem z.B. in Freilandstraßen, wo es keine Straßenlaternen gibt. Auf den Autobahnen und Freilandstraßen wurden keine Straßenlaternen aufgestellt, da es an solchen Straßen in der Realität auch selten Straßenlaternen gibt. Außerdem hätte das Aufstellen von Straßenlaternen an Autobahnen und Freilandstraßen viel

Zeit gekostet und ebenfalls hätte die Performance darunter gelitten, da die Spielwelt sowieso schon über sehr viele Objekte verfügt. Falls man einen Nachtmodus implementieren würde, sollten die Gebäude im Spiel auch über Beleuchtung verfügen. Wenn man in der Realität z.B. in der Nacht fährt, dann sieht man neben dem Licht der Straßenlaternen und Ampeln ebenfalls oft leuchtende Reklamen oder Fenster, in denen das Licht eingeschalten ist. Dadurch würde das Spiel realistischer wirken, jedoch könnte man dadurch auch als Fahrer üben wichtige Lichter wie z.B. Ampellichter und Lichter von Autos von anderen Lichtern zu unterscheiden. Falls es einen Nachtmodus geben würde, dann wäre es wohl am besten, dies den Spieler über die Optionen einstellen zu lassen. Eventuell könnte man sogar neben Tag und Nacht verschiedene Tageszeiten einbauen wie z.B. Früh, Mittag, Nachmittag und Abend. Dazu müsste das Sonnenlicht im Spiel stärker oder schwächer gestellt werden und ebenfalls müsste die Farbe des Lichts passend geändert werden für jede wählbare Tageszeit. Zum Beispiel würde in der Früh eine leicht orange Lichtfarbe gewählt werden und am Abend eine leicht dunkelblaue Lichtfarbe.

6.6 Wetter

In Virtual Driving School wurde kein Wetter implementiert. Dem Spieler die Möglichkeit zu geben das Wetter einstellen zu lassen, könnte deutlich aufwendiger sein als das Implementieren verschiedener Tageszeiten. Als Wetterzustände könnte z.B. Sonne, Bewölkt, Regen und Schnee eingebaut werden. Man könnte sogar einbauen, dass der Grad an Bewölkung oder an Regen oder Schnee eingestellt werden kann. Dazu müsste die Lichtfarbe des Sonnenlichts passend geändert werden. Zum Beispiel müsste bei Schneefall die Lichtfarbe eher weiß sein und bei einem klaren Himmel müsste die Lichtfarbe eher gelb sein. Für Regen und Schnee wären natürlich Regen- oder Schneepartikel notwendig, welche vom Himmel herunterfallen. Eine sehr einfache und performancesparende Möglichkeit wäre, wenn man bei Regen oder Schnee einfach ein Video-Overlay über das Spiel legen würde, dies würde jedoch unecht wirken, sodass dreidimensionale Partikel vorzuziehen wären. Ebenfalls sollte implementiert werden, dass die Partikel sich auf die Spielwelt auswirken. Zum Beispiel sollten Regen- und Schneepartikel auf den Scheiben der Autos zu finden sein und wegewischt werden, wenn der Scheibenwischer verwendet wird. Außerdem sollten auch die Straßen nass bzw. mit Schnee bedeckt werden. Das Fahrverhalten des eigenen Autos und von NPC Autos könnte sich ebenfalls durch Regen und Schnee ändern. Ebenfalls könnte sich das Befahren der Straßen auf den nassen oder schneebedeckten Boden auswirken, so dass z.B. Rinnen entstehen, jedoch könnte dies sehr aufwendig zum Implementieren sein. Unabhängig von Wetter, könnte man das Spiel auch verbessern, wenn Autos auf verschiedenen Oberflächen ein anderes Fahrverhalten aufweisen würden. Zum Beispiel würden sich Autos auf Asphaltstraßen anders fahren als auf Grasböden oder Schotterstraßen. Das Implementieren eines Wettersystems in einer Fahrsimulation könnte je nach Grad des gewünschten Realismus sehr aufwendig sein. Jedoch würde es die Fahrsimulation verbessern, da das Fahren bei Regen und bei Schnee geübt werden könnte. Ebenfalls kann das Verwenden der Scheibenwischer geübt werden, da diese gerade bei Fahranfängern ablenkend wirken können. Außerdem kann geübt werden, wie die Geschwindigkeit an die Wetterbedingungen angepasst werden muss und das Beachten des Verkehrs mit schlechten Sichtbedingungen durch Regen und Schnee könnte geübt werden.

6.7 Mehrere fahrbare Autos

Eine andere Möglichkeit das Projekt zu verbessern wäre, indem man dem Spieler die Möglichkeit gibt mehrere Autos zu fahren. Neben verschiedenen Automodellen könnte man dem Spieler auch

die Möglichkeit geben, die Farbe des zu fahrenden Autos zu ändern. Damit die Auswahl zwischen verschiedenen Autos nicht nur ein kosmetisches Feature wäre, welches für visuelle Abwechslung sorgen würde, sollten die Autos sich auch anders steuern lassen. Zum Beispiel könnte sich das Fahrverhalten der Autos unterscheiden und wie schnell diese sich beschleunigen und bremsen lassen oder wie stark gelenkt werden muss im Vergleich zu anderen Autos. Interessant wäre auch, wenn andere fahrbare Autos z.B. einen Vorderrad- oder Allradantrieb hätten. Das Spielerauto momentan hat nämlich einen Hinterradantrieb. Autos mit Hinterradantrieb sind bekannt dafür, dass sie leichter übersteuern, während Autos mit Vorderradantrieb leichter untersteuern. Untersteuern tritt auf, wenn ein Auto in einer Kurve zu schnell fährt und verstärkt sich, wenn das Auto auch noch in der Kurve gebremst wird, wodurch die Vorderräder an Haftung verlieren und das Auto aus der Kurve nach außen schiebt. Übersteuern hingegen tritt auf, wenn ein Auto in einer Kurve zu schnell unterwegs ist und die Hinterräder an Haftung verlieren und das Heck des Autos ausbricht, also wenn es anfängt zu schleudern.

6.8 Mehr Lektionen

Wie in der Evaluierung festgestellt, wäre eine Lektion sinnvoll gewesen, welche dem Spieler erklärt wie er ein Auto mit manueller Schaltung zum Anhalten bringen kann. Also, dass, die Kupplung gedrückt werden muss, damit der Motor nicht abstirbt. Eine Lektion, welche dem Spieler erklärt was Über- und Untersteuern ist, wie es vermieden werden kann und was zu tun ist, falls es auftritt, wäre auch eine gute Erweiterung gewesen, da gerade dieses Wissen in Notfallsituationen lebensrettend sein könnte.

7 ZUSAMMENFASSUNG

Zusammengefasst kann man nach der Evaluierung sagen, dass Virtual Driving School sich gut eignet, um sich Basiswissen anzueignen und die Bewegungsabläufe beim Autofahren kennenzulernen. Unter anderem kann man lernen, wie ein Auto gestartet wird, wie geschaltet wird und welche Blicke wann gesetzt werden müssen oder wie man richtig einparkt. Selbstverständlich können solche Bewegungsabläufe besser mit einem Lenkrad mit Pedalen und Schaltknüppel geübt werden als mit Maus und Tastatur. Mit einem VR-Headset können zusätzlich die Blicke besser geübt werden als mit einem flachen Bildschirm. Um das Spiel mit einem VR-Headset zu spielen, sollte jedoch ein besonders starkes System verwendet werden, da das Spiel so flüssig wie möglich laufen sollte bei gleichzeitig hoher Grafikqualität und Sichtweite sowie Fußgänger- und Fahrzeugdichte, damit die Spielwelt realer wirkt. Eine wichtige Erkenntnis ist, dass bei Fahrsimulationen Motion Feedback eingesetzt werden kann, um das Fahrgefühl näher an die Realität zu bringen. Dabei kann zum Beispiel das Force Feedback System des Lenkrads genutzt werden, oder eine Motion Plattform könnte eingesetzt werden, um Bewegungskräfte zu simulieren, die auf den Fahrer in einem echten Auto wirken würden. Jedoch muss darauf geachtet werden, dass Personen bestimmte Bewegungsabläufe, die sie in der Simulation gelernt haben, unabsichtlich in der Realität verwenden könnten, was gefährlich werden könnte, wenn z.B. die Gänge anders angeordnet sind. Ebenfalls muss gerade bei Personen, die wenig Erfahrung mit VR-Headsets haben, darauf geachtet werden, dass nicht zu lange gespielt wird, da sonst Symptome von Motion Sickness auftreten können. Jedoch ist es möglich mit der Zeit eine Toleranz für VR aufzubauen. Eine andere wichtige Erkenntnis ist, dass Fahrsimulatoren speziell Menschen mit Behinderung helfen können. Zum Beispiel können Menschen mit Rückenmarksverletzung, welche ihre Beine nicht bewegen können, in einer Fahrsimulation das Fahren mit einer alternativen Steuerung üben, bevor dies in einem echten Auto geübt wird.

8 REFERENZEN

- [1] „Oculus,“ Oculus VR, [Online]. Available: <https://www.oculus.com/compare/>. [Zugriff am 10 05 2020].
- [2] „Air Light VR,“ [Online]. Available: <https://github.com/polygraphene/ALVR>. [Zugriff am 10 05 2020].
- [3] „RiftCat VRidge,“ RiftCat, [Online]. Available: <https://riftcat.com/vridge>. [Zugriff am 10 05 2020].
- [4] N. Winkler, K. Röthke, N. Siegfried und A. Benlian, „Lose Yourself in VR: Exploring the Effects of Virtual Reality on Individuals’ Immersion,“ in *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.
- [5] „Oculus Developer,“ Oculus, [Online]. Available: <https://developer.oculus.com/documentation/native/pc/dg-render/>. [Zugriff am 10 05 2020].
- [6] M. Kraus, „On the Preference for Travel by Steering in a Virtual Reality Game,“ in *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2020.
- [7] „Half-Life: Alyx,“ Valve, [Online]. Available: https://store.steampowered.com/app/546560/HalfLife_Alyx/. [Zugriff am 10 05 2020].
- [8] „City Car Driving,“ Forward Development Ltd, [Online]. Available: https://store.steampowered.com/app/493490/City_Car_Driving/. [Zugriff am 10 05 2020].
- [9] „Google Cardboard,“ Google, [Online]. Available: https://arvr.google.com/intl/de_de/cardboard/get-cardboard/. [Zugriff am 10 05 2020].
- [10] A. Schwarze, H. Kampling, O. Heger und B. Niehaves, „Is Virtual Reality the Future of Learning? A Critical Reflection,“ in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.
- [11] P. Häfner, V. Häfner und J. Ovtcharova, „Experiencing Physical and Technical Phenomena in Schools Using Virtual Reality Driving Simulator,“ in *International Conference on Learning and Collaboration Technologies*, 2014.
- [12] F. Corelli, E. Battagazzorre, F. Strada, A. Bottino und G. P. Cimellaro, „Assessing the Usability of Different Virtual Reality Systems for Firefighter Training,“ in *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020)*, 2020.
- [13] D. Slovikosky, J. Davis, D. Sanchez und L. Bozgeyikli, „PC Builder Hero: An Immersive Computer Building Workshop Experience in Virtual Reality,“ in *ACM Designing Interactive Systems*, 2019.
- [14] H. Lê, T. Long Pham und G. Meixner, „A Concept For A Virtual Reality Driving Simulation In Combination With A Real Car,“ in *Adjunct Proceedings of the 9th International ACM Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI ’17)*, Oldenburg, 2017.
- [15] S. Ropelato, F. Zünd, S. Magnenat, M. Menozzi und R. W. Sumner, „Adaptive Tutoring on a Virtual Reality Driving Simulator,“ in *Proceedings of 1st Workshop on Artificial Intelligence Meets Virtual and Augmented Worlds (AIVRAR)*, Bangkok, 2017.

- [16] S. Tudor, S. Carey und R. Dubey, „Development and Evaluation of a Dynamic Virtual Reality Driving Simulator,“ in *PErvasive Technologies Related to Assistive Environments*, Corfu, 2015.
- [17] S. Bateman, A. Doucette, R. Xiao, C. Gutwin, R. L. Mandryk und A. Cockburn, „Effects of View, Input Device, and Track Width on Video Game Driving,“ in *Proceedings of the Graphics Interface*, 2011.
- [18] D. Goedicke, J. Li, V. Evers und W. Ju, „VR-OOM: Virtual Reality On-rOad driving siMulation,“ in *ACM Conference on Human Factors in Computing Systems*, 2018.
- [19] F. N. Nezami, M. A. Wächter, G. Pipa und P. König, „Project Westdrive: Unity City With Self-Driving Cars and Pedestrians for Virtual Reality Studies,“ in *Frontiers in ICT*, 2020.
- [20] „UMA 2 - Unity Multipurpose Avatar,“ [Online]. Available: <https://assetstore.unity.com/packages/3d/characters/uma-2-unity-multipurpose-avatar-35611>. [Zugriff am 10 06 2020].
- [21] „Building a Traffic System in Unity,“ [Online]. Available: <https://www.youtube.com/watch?v=MXCZ-n5VyJc>. [Zugriff am 10 06 2020].
- [22] „Oculus Mindestanforderungen,“ [Online]. Available: https://support.oculus.com/248749509016567/?locale=de_DE. [Zugriff am 29 07 2020].
- [23] „GPUBoss,“ [Online]. Available: <http://gpuboss.com/gpus/Radeon-RX-470-vs-Radeon-R9-380>. [Zugriff am 29 07 2020].
- [24] „igroup presence questionnaire (IPQ),“ [Online]. Available: <http://www.igroup.org/pq/ipq/download.php#German>. [Zugriff am 21 06 2020].
- [25] Q. C. Ihemedu-Steinke, D. Sirim, R. Erbach, P. Halady und G. Meixner, „Development and Evaluation of a Virtual Reality Driving Simulator,“ in *Mensch und Computer*, 2015.

9 ANHANG

Hier kann der Fragebogen mit allen Antworten nachgelesen werden. Die Antworten von allen Fragebögen wurden in einen Fragebogen zusammengefasst, damit die Antworten leichter gelesen werden können.

Fragebogen Virtual Driving School:

Allgemeine Fragen

Wie alt bist du?

- 15
- 55
- 57
- 27
- 30
- 27
- 26
- 25
- 34

Bist du männlich oder weiblich?

- Männlich
- Weiblich
- Männlich
- Männlich
- Weiblich
- Männlich
- Männlich
- Männlich
- Männlich

Hast du einen Führerschein?

- Nein
- Nein
- Ja
- Nein
- Ja
- Ja
- Ja
- Ja
- Ja

Wenn ja, seit wie vielen Jahren?

- /
- /
- 9
- /
- 12 Jahre
- seit 8 Jahren
- seit 9 Jahren
- 7 Jahre
- 16

Wie oft fährst du mit dem Auto auf einer Skala von 1 - 6? (1=Sehr oft, ..., 6=Sehr selten)

- Nie
- Nie
- 1
- Nie
- 4
- 5
- 3
- 2
- 2

Wie würdest du deine Fahrfähigkeiten beschreiben auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

- 6
- 4
- 1
- 6
- 2
- 3
- 2
- 2
- 2

Lektionen

Welche Lektionen hast du im Spiel ausprobiert?

- Anfahren und Schalten, Rechtsabbiegen, Linksabbiegen, Parallelparken, Rückwärts einparken, Spurwechsel
- Anfahren und Schalten
- Anfahren und Schalten, Parallelparken
- Anfahren und Schalten, Rechtsabbiegen, Links abbiegen, Parallel Parken
- A) Anfahren und Schalten, B) Rückwärtseinparken, C)Linksabbiegen, D)Kreisverkehr
- Anfahren und Schalten, Rückwärts einparken, Umkehren, Spurwechsel
- Alle
- Spurwechsel, Rückwärtseinparken, Parallelparken
- Anfahren und Schalten, Rechtsabbiegen, Parallelparken, Kreisverkehr, Spurwechsel

Wie findest du sind die Lektionen gemacht auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

- 4
- 2
- 2
- 2
- A) 1, B) 3, C) 1, D) 1
- 4
- 2
- 2
- 3

Was findest du gut bzw. schlecht an den Lektionen?

- Lektionen nicht nummeriert, fundamentale Manöver nicht beigebracht wie zum Beispiel das Bremsen ohne dem Absterben des Motors, Bei keiner Lektion ist es möglich herauszufahren aus dem Übungsplatz, Bei der Abbildung zum Rückwärts einparken erkennt man nicht wo sich die Vorderseite oder Hinterseite des Fahrzeugs befindet
- /
- Steuerung ist schwierig mit Lenkrad
- Gut: Es ist relativ klar, worum es geht und es beginnt langsam. Schlecht: Bremsen! Ich glaube da wäre eine Lektion sinnvoll, weil sich der Motor ausschaltet beim bremsen und ich nicht weiß wieso.
- A) ich finde es sehr gut umgesetzt, es ist nahe an der Realität, B) eventuell ist es mit einem Gamepad einfacher, aber umsehen und einparken ist mit Maus und Tastatur verhältnismäßig schwierig, C) & D) Gut erklärt und auch gut umgesetzt von der Steuerung
- Gut: Einfache Erklärung. Schlecht: schwierige Umsetzung mit der Tastatur als Steuerung
- Die Lektionen sind gut gemacht und alles funktioniert, wie es soll. Es wäre gut, wenn die Lektionsziele während des Fahrens eingeblendet werden würden und man diese „erfüllen“ könnte. Somit könnte man eine Lektion „abschließen“.
- Gut: Kurze Ladezeiten, man kann das Auto sehr detailliert steuern, große Auswahl und gute Beschreibung. Nicht so gut: Keine Rückmeldung ob man etwas richtig oder falsch macht, wobei das sicher schwer messbar ist
- Gut: Mir gefällt das am Anfang einer Lektion Instruktionen gegeben werden und diese auch während der Lektion zur Verfügung stehen. Schlecht: Ich hätte gerne Musik in den Lektionen gehabt.

Konntest du was Neues lernen bei den Lektionen? Falls ja, was?

- Gangschaltung, Motor anmachen, Blinker,
- Ja, wie man anfährt und schaltet
- Nein
- Ja, wie man ein Auto startet und losfährt und Gänge wechselt mit Kupplung. Einparken ist schwierig.
- Nein
- ja, es sind automatische Bewegungsabläufe wieder bewusster geworden.
- Nein
- Eigentlich nicht aber das Spiel war interessant zu testen. VR hätte ich gerne getestet
- Nein, war mir schon bekannt, nachdem ich ja bereits einen Führerschein habe.

Findest du es gut, dass man als Spieler selber die Lektionen über das Hauptmenü beenden muss (also, dass es nicht von selbst passiert)?

- Ja
- Ja
- Ja
- Ja
- Ja
- Ich finde es schlecht, dass man die Lektionen selbst beenden muss, so weiß man nicht wirklich ob man die Lektion geschafft hat oder nicht.
- Ich empfinde dies nicht als störend. Einen Modus in dem die Lektionen nacheinander/durcheinander geladen werden wäre gut.
- Einerseits gut aber wie schon erwähnt keine Rückmeldung ob man die Übung richtig macht
- Stört mich nicht.

Freie Fahrt Modus

Wie findest du den Freie-Fahrt-Modus auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

- 2
- 3
- 2
- 1
- 2
- 2
- 2
- 1
- 2

Welche Gangschaltung hast du verwendet? Automatisch, manuell oder hast du beide ausprobiert?

- Beide ausprobiert
- Manuell
- Manuell
- Manuell
- Beide
- Manuell
- beide ausprobiert
- Automatisch
- Ich habe beide ausprobiert. Automatisch ist angenehmer zu fahren, wie ein Arcade Mode bei einem Rennspiel. Allerdings geht hierbei dann der Lerneffekt verloren, in welcher Reihenfolge bzw. Kombinatorik die Fahrzeugbestandteile miteinander agieren sollten. Die manuelle Steuerung war sehr gewöhnungsbedürftig allerdings, da man hier sehr stark auf die besagte Reihenfolge achten soll und mir dadurch öfter das Auto abgestorben ist.

Wie findest du sind die anderen Autofahrer gemacht auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

- 2
- 1
- 1
- 2
- 2
- 4
- 3
- 2
- 1

Wie findest du sind die Fußgänger gemacht auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

- 2
- 1
- 1
- 3
- 1
- 2
- 3
- 3
- 1

Wie findest du ist die Spielwelt gemacht auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

- 3
- 1
- 1
- 2
- 1
- 2
- 3
- 1
- 1

Was findest du gut bzw. schlecht?

- Straßen gehen nicht nahtlos in einander herein
- /
- Lenkrad ist nicht empfindlich genug
- Gut: Rumpfahnen macht Spaß. Schlecht: Auto kippt manchmal um
- Es macht Spaß im freien Modus die Welt zu erkunden und auszutesten wo die Grenzen sind, leider lassen sich Fußgänger nur verschieben, besser wäre es gewesen, wenn sie umfallen sollte man sie versehentlich anfahren. An das Schalten gewöhnt man sich recht schnell und wird dann recht intuitiv – die Steuerung (zumindest mit Maus und Tastatur) wird nur ziemlich schwierig bei höheren Geschwindigkeiten – es reagiert sehr extrem. Ich finde die vielen Details sehr gut, zum Beispiel die Hupe, die Scheibenwischer und auch die Warnblinkanlage – das macht es sehr viel realistischer.
- Schlecht: Es gab zu wenig andere Autofahrer um wirklich Dinge auszuprobieren, wie z.B. überholen. Gut: Die Sicht in der Spielwelt war gut, es war erkenntlich was die einzelnen Objekte darstellen sollten. Die Ampeln waren übersichtlich.
- die anderen Autofahrer und die Fußgänger interagieren wenig mit der Umwelt. Beispielsweise passiert nichts bei einem Unfall.
- Weder gut noch schlecht, cool wäre wenn das Auto im freie Fahrt Modus schneller wäre als wir bei den Lektionen. Wirkt für mich eher wie ein cooler Bonus zu den Lektionen der aber auch Spaß macht.
- Mir gefällt das Fahrzeug, es wurde mit sehr viel Liebe zum Detail konstruiert. Die einzelnen Lektionen sind sehr abwechslungsreich und vor allem lehrreich gemacht. Der Freie Fahrt Modus macht besonders mit dem Automatik Modus Spaß und hat schon etwas von einem kleinen Arcade Rennspiel. Ich hätte mir grundsätzlich gerne mehr Musik im Hintergrund gewünscht, was aber für einen Fahrsimulator sicher keine Priorität hat.

Sonstige Fragen

Hast du dich im Menü zurechtgefunden?

- Ja
- Ja
- Ja
- Ja
- Ja
- ja, war einfach
- ja
- Ja
- Ja, ist sehr übersichtlich.

Hast du dich mit der Steuerung zurechtgefunden?

- Ja
- ja, aber Gangschaltung ungewohnt
- Ja, es war gut
- Jain, wie lässt man die Kupplung langsam los? Mehrmals drücken?
- hat etwas gedauert, aber ja
- Es wäre einfacher mit Hilfe eines Tutorials die Steuerung kennenzulernen. Und die Default-Tastaturbelegung fand ich nicht so gut, da man mit der „Maushand“ auch den Motor starten musste, da die Entertaste zu weit von der linken Hand entfernt ist und auch die Gangschaltung notwendig ist.
- Ja
- Ja
- Nein, die manuelle Steuerung ist sehr kompliziert mit Maus und Tastatur. Außerdem ist die Steuerung über WASD zu ungenau, da meiner Meinung nach beim links- bzw. rechtsfahren das Auto sehr schnell zu driften beginnt und man den Lenkradeinschlag zu ungenau definieren kann (auch in der Automatik Einstellung ein Problem).

Wieviel Spaß macht das Spiel auf einer Skala von 1 - 6? (1=Sehr viel, ..., 6=Sehr wenig)

- 2
- 1
- 2
- 2
- 2
- 4
- 2
- 2
- Im Automatik Modus: 1 / Im Manuellen Modus: 3

Wie professionell wirkt das Spiel auf einer Skala von 1 - 6? (1=Sehr professionell, ..., 6=Sehr unprofessionell)

- 4
- 2
- 1
- 1 (solche Software sieht oft nicht anders aus)
- 2
- 4
- 2
- 1
- 3

Wie realistisch findest du das Fahrverhalten des eigenen Autos auf einer Skala von 1 - 6? (1=Sehr realistisch, ..., 6=Sehr unrealistisch)

- 3
- 2
- 1
- 3 (Auto schleudert nicht)
- 3
- 2
- 3
- 3
- 4

Wie findest du die Grafik des Spiels auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

- 2
- 1
- 1
- 3
- 3
- 3
- 2
- 3
- 2

Wie flüssig und ruckelfrei lief das Spiel auf Skala von 1 - 6? (1=Sehr flüssig, ..., 6=Sehr ruckelig)

- 2
- 2
- 2
- 1
- 1
- 1
- 1
- 1
- 1
- 3

Wie stark ist dein Computer auf einer Skala von 1 - 6? (1=Sehr stark, ..., 6=Sehr schwach)

- 3
- 3
- 3
- 2
- 2
- 1
- 2
- 3
- 4

Wie lange waren die Ladezeiten des Spiels bei dir ungefähr?

- 10 Sekunden beim Ersten Mal, 5 bei den Nächsten Versuchen
- 5 Sek
- 2-3 Sekunden
- Nicht sehr lange, 3-5 Sekunden
- ein paar Sekunden
- 3 Sekunden
- unter 10 Sekunden
- Sehr kurz
- Kaum zu spüren, vielleicht 1-2 Sekunden.

Wie könnte man das Projekt deiner Meinung nach verbessern?

- Lektionen nummerieren, Spiel ruckelt am Anfang, fundamentale Grundlagen erklären, Abbildung verbessern,
- Gar nicht aber Grafik könnte realistischer sein
- Lenkrad und Gaspedal empfindlicher machen
- Interaktives Feedback, z.B. wenn man den Motor abwürgt → wieso ist das passiert?
- Mit ein paar Sounds (dass es mehr kracht, wenn man irgendwo anfährt zum Beispiel). Eventuell im Freien Modus dass es Punkte zum Sammeln bzw. verlieren gibt (wenn eine rote Ampel überfahren wird -> Abzug, wenn blinken beim Abbiegen -> Punkte) oder eine Art Auszeichnungssystem für jede erfolgreich gemeisterte Lektion
- Lektionen sollten ein Ziel haben, Tastaturbelegung ändern, Vogelperspektive fürs Rückwärts-parken, da man sonst zu wenig sieht.
- Essenzielle Tastaturbelegungen könnten während der Fahrt/Lektionen eingeblendet werden. Der Rückwärtsgang im manuellen Modus funktioniert erst bei mehrmaligem Drücken der S-Taste. Wenn das Auto rückwärts rollt, muss es erst komplett zum Stehen kommen ehe man wieder vorwärtsfahren kann. Es wäre gut, wenn andere Autotypen ausgewählt werden könnten. Aus meiner Sicht geraten die Autos zu schnell ins Schleudern (subjektive Ansicht)
- Meiner Meinung wäre es am sinnvollsten das Fahrverhalten noch zu verbessern und vielleicht auch wenn sicher schwierig das irgendwie Rückmeldung geben ob die Übung richtig gemacht wird.
- Ja insbesondere eine Verfeinerung der Lenkradeinschlag wäre sinnvoll, da vor allem bei höheren Geschwindigkeiten im Moment eine sehr hohe Unfallwahrscheinlichkeit gegeben ist.

Denkst du das Spiel eignet sich, um damit Autofahren zu lernen bzw. zu üben? Warum?

- Es eignet sich um ein gewisses Basiswissen anzueignen, doch dieses Stück Software ist und wird nie ein Ersatz für das Fahren im echten Leben sein
- Ja, sehr gut für Anfänger
- Ja, sehr
- Zum Üben ja, zum Lernen fehlt ein bisschen das live Feedback was man tun soll.
- Ich denke es hat auf jeden Fall Potential, es zeigt sehr gut die Abläufe die notwendig sind, um den Straßenverkehr zu meistern, auch wenn die Steuerung mit Maus und Tastatur gewöhnungsbedürftig sind.
- Lernen nein, üben teilweise. Lernen nein, weil es doch zu weit von der echten Welt entfernt ist, wenn man auf die Tastatur klickt anstatt ein echtes Lenkrad in der Hand zu halten. Üben ja, da man Abläufe wie Schulterblick nochmals verinnerlichen und visualisieren kann.
- aus meiner Sicht eignet sich das Programm, um damit Autofahren zu lernen. jedoch nur wenn die Software mit einem Lenkrad zusammen genutzt wird. Die Steuerung mit Tastatur und mause ist aus meiner Sicht zu weit entfernt von realen Bedingungen.
- Ist sicher hilfreich um spielerisch damit einfache Sachen zu erlernen bevor man mit einem richtigen Auto fährt.
- Eher nicht, außer im VR Modus mit einem Lenkrad, Pedalen und einer Kupplung. Autofahren lernt man meiner Meinung nach nur in einem Auto oder einem Simulator, der ein Auto 1:1 imitiert. Die Instrumente, Fahrzeugkomponenten (wie z.B. Lenkrad, Pedale und Kupplung, Lichtmaschine etc.) sind über Maus und Tastatur unmöglich zu replizieren.

Hast du das Spiel mit der Maus und Tastatur gespielt oder mit dem Logitech G29 Lenkrad oder beides?

- Beides
- G29
- Logitech G29
- Maus & Tastatur
- Maus und Tastatur
- Maus und Tastatur
- Maus + Tastatur
- Nur mit Maus und Tastatur
- Maus und Tastatur.

Hast du das Spiel mit dem normalen Bildschirm gespielt oder mit einem VR-Headset oder beides?

- Beides
- Beides
- Beides
- Normaler Bildschirm
- Bildschirm
- Normalen Bildschirm
- normaler Bildschirm
- Nur normaler Monitor
- Normaler Bildschirm.

Anmerkungen?

- /
- Man sieht seine Beine nicht und seine Hände nicht in VR leider
- /
- Ich finde die Idee ziemlich gut
- /
- An sich eine gute Umsetzung eines Autofahr-Simulators mit Potenzial zur Verbesserung.
- /
- Bei mir war es manchmal schwierig den Rückwärtsgang hineinzubekommen. Wenn etwas geladen wird würde ich statt „Spiel ladet“ → „Spiel lädt“ schreiben ist aber nur ein Vorschlag.
- /

Virtuelle Realität Fragen (falls VR getestet wurde)

Quelle: <http://www.igroup.org/pq/ipq/download.php#German>

In der computererzeugten Welt hatte ich den Eindruck, dort gewesen zu sein... (1=Sehr stark, ..., 6=Überhaupt nicht)

- 2
- 1
- 1

Ich hatte das Gefühl, dass die virtuelle Umgebung hinter mir weitergeht. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

- 3
- 1
- 1

Ich hatte das Gefühl, nur Bilder zu sehen. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

- 5
- 2
- 6

Ich hatte nicht das Gefühl, in dem virtuellen Raum zu sein. (1=Hatte das Gefühl, ..., 6=Hatte nicht das Gefühl)

- 5
- 1
- 1

Ich hatte das Gefühl, in dem virtuellen Raum zu handeln statt etwas von außen zu bedienen. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

- 1
- 5
- 1

Ich fühlte mich im virtuellen Raum anwesend. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

- 2
- 1
- 1

Wie bewußt war Ihnen die reale Welt, während Sie sich durch die virtuelle Welt bewegten (z.B. Geräusche, Raumtemperatur, andere Personen etc.)? (1=Extrem bewusst, ..., 6=Unbewusst)

- 3
- 1
- 2

Meine reale Umgebung war mir nicht mehr bewußt. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

- 3
- 1
- 2

Ich achtete noch auf die reale Umgebung. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

- 4
- 3
- 5

Meine Aufmerksamkeit war von der virtuellen Welt völlig in Bann gezogen. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

- 3
- 3
- 2

Wie real erschien Ihnen die virtuelle Umgebung? (1=Vollkommen real, ..., 6=Gar nicht real)

- 3
- 2
- 1

Wie sehr glich Ihr Erleben der virtuellen Umgebung dem Erleben einer realen Umgebung? (1=Vollständig, ..., 6=Überhaupt nicht)

- 3
- 2
- 1

Wie real erschien Ihnen die virtuelle Welt? (1=Nicht zu unterscheiden von der realen Welt, ..., 6=Wie eine vorgestellte Welt)

- 4
- 2
- 2

Die virtuelle Welt erschien mir wirklicher als die reale Welt. (1=Trifft völlig zu, ..., 6=Trifft gar nicht zu)

- 6
- 3
- 2

Logitech G29 Lenkrad Fragen (falls es getestet wurde)

Hast du dich mit der Lenkrad Steuerung zurechtgefunden?

- Ja
- Es war schwierig, weil man Tasten am Lenkrad und Gangschaltung nicht sieht in VR. Es wäre besser wenn das Spiellenkrad und das G29 genau gleich wäre und wenn man seine Hände sieht in VR.
- Es war ein bisschen schwierig

Bevorzugst du das Logitech G29 oder Maus und Tastatur?

- Egal
- G29
- Lenkrad da realer, aber Tastatur wurde nicht ausprobiert

Wie realistisch ist die Lenkradsteuerung gemacht auf einer Skala von 1 - 6? (1=Sehr gut, ..., 6=Sehr schlecht)

- 4
- 1
- 1

Was gefällt dir bzw. was gefällt dir nicht an der Lenkradsteuerung?

- Lenkrad geht nicht zur Ausgangsposition
- Es war sehr gut, aber ganz neu
- Lenkrad und Gas was zu unempfindlich. Kupplungspedal muss sehr langsam losgelassen werden, damit Auto nicht abstirbt