

Explorative Datenanalyse am Beispiel von Spielerklassifikation für Augmented-Reality-Spiele

CHRISTIAN WINTER



MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im September 2016

© Copyright 2016 Christian Winter

Diese Arbeit wird unter den Bedingungen der *Creative Commons Lizenz Namensnennung–NichtKommerziell–KeineBearbeitung Österreich* (CC BY-NC-ND) veröffentlicht – siehe <http://creativecommons.org/licenses/by-nc-nd/3.0/at/>.

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Hagenberg, am 26. September 2016

Christian Winter

Inhaltsverzeichnis

Erklärung	iii
Kurzfassung	vii
Abstract	viii
1 Einleitung	1
1.1 Motivation	2
1.2 Problemstellung	2
1.3 Zielsetzung	3
1.4 Aufbau dieser Arbeit	3
2 Vancouver Maneuver	4
2.1 Voraussetzungen	4
2.2 Gameplay	7
2.2.1 Steuerung	7
2.2.2 Spielmechaniken	8
2.2.3 Levelstruktur	8
2.2.4 Spielelemente	9
3 Explorative Datenanalyse	12
3.1 Einfache statistische Maßzahlen	13
3.1.1 Mittelwert	13
3.1.2 Streuung	14
3.1.3 Anscombe-Quartett	15
3.2 Wertetabelle	15
3.3 Histogramm	16
3.4 Diagramm	17
3.4.1 Streudiagramm (scatter plot)	17
3.4.2 3D-Diagramm	19
3.5 Betrachtung von Teilmengen	21
3.6 Versteckte Variablen	22
3.7 Regression	22
3.7.1 Regressionsanalyse	22

3.7.2	Korrelation	22
3.7.3	Lineare Regression	23
3.8	Kreisförmige Daten	23
3.8.1	Eulerwinkel	24
3.8.2	Quaternion	24
3.8.3	Summen von Kreisförmigen Daten	24
4	Testumgebung	27
4.1	Versuchsaufbau	27
4.2	Dateiformat	27
4.3	Beurteilung der Spielerfähigkeiten	28
4.3.1	Bewertungskriterien	28
5	Konkreter Einsatz in Vancouver Maneuver	30
5.1	RStudio	30
5.1.1	Verwendete Pakete	32
5.2	Welche Daten stehen zur Verfügung	33
5.2.1	Anzahl der Objekte im Sichtfeld	33
5.2.2	Objektposition am Bildschirm	33
5.2.3	Objektdistanz zum Fokuspunkt des Spielers	34
5.3	Aufbereitung der Daten	34
5.3.1	Verwendung der Daten in <i>RStudio</i>	34
5.3.2	Probleme	35
5.3.3	Ausreißer	36
5.4	Vorgehensweise	37
5.5	Analyse der Spieldurchläufe	37
5.5.1	Tabelle	37
5.5.2	Standardabweichung und Varianz	38
5.5.3	Streudiagramm	38
5.5.4	3D Diagramme	39
5.6	Ansätze und Ideen	39
5.6.1	Nähe zum Spielfeld als Indikator für gute Spieler	39
5.6.2	Übermäßige Rotation als Indikator für schlechte Spieler	40
5.7	Analyse der gesamten Daten	42
5.7.1	Ergebniss	42
5.8	Aufgetretene Probleme und Fehlerquellen	42
5.8.1	Balkenanzahl von Histogrammen	42
5.8.2	Vergleich von Rotationen	43
5.8.3	Absolute und relative Werte	44
5.8.4	Anzahl der Testdaten	45
6	Schlussbemerkungen	49
6.1	Zusammenfassung	49
6.2	Fazit	49

6.3	Ausblick	50
A	Inhalt der CD-ROM/DVD	51
A.1	PDF-Dateien	51
A.2	Bilder	51
A.3	Literatur	51
A.4	Online-Quellen	51
A.5	Data	51
A.6	R-Skripte	52
	Quellenverzeichnis	53
	Literatur	53
	Online-Quellen	53

Kurzfassung

In dieser Arbeit wird gezeigt worum es sich bei explorativer Datenanalyse handelt und wie bei einer solchen vorgegangen wird. Das ganze wird anhand von Daten gezeigt, die mit Hilfe eines Videospiels erstellt werden. Da solche Verfahren relativ weit verbreitet sind und für viele Spiele bereits weitestgehend erforscht sind, wird in dieser Arbeit versucht diese Methoden für neuartigen Informationen zu verwenden. Die behandelten Daten werden von einem *Augmented-Reality*-Videospiel erstellt. Bei den Daten handelt es sich lediglich um Informationen, die anhand des Sichtfeldes der Spieler generiert werden. Die Analyse soll in der Lage sein, aufgrund dieser Daten, auf das Können und das Geschick der Spieler schließen zu können. Das Ziel ist es, die Spieler anhand dieser Beurteilung in fünf Klassen unterschiedlichen Könnens einzuteilen. Es wird gezeigt wie die Daten erzeugt, aufbereitet und verarbeitet werden. Der Arbeitsprozess mit der Analysesoftware wird gezeigt. Alle Schritte der Analyse werden gezeigt. Sämtliche aufgetretenen Probleme und die Lösung derer werden beschrieben. Das Ergebnis der Analyse war, dass es nicht möglich ist das Können der Spieler mit deren Sichtfeld in Verbindung zu bringen.

Abstract

This thesis shows what exploratory data analysis is and how it is applied to data collected from a video game. Because these methods are already researched a lot in different fields this thesis will use novel and uncommon data. The collected data is generated from an augmented reality game and only consists of field-of-view specific information. The main goal is to determine the player's skill and classify the players into five different groups based on it. It is shown which data was extracted from the field of view, how it was cleaned up and used in an analysis software. The workflow of this process is documented. The progress of the analysis is shown and the problems that occurred are listed and displayed how these were solved. The result of this data analysis is that the attempt did not work out as intended. It was not possible to link a player's skill to the player's field of view.

Kapitel 1

Einleitung

Basierend auf Handlungen die getätigt werden, lassen sich bestimmte Aussagen über Spieler treffen. Informationen über Verhaltensweisen können dazu verwendet werden, um das Spielerlebnis möglichst gut auf die Spieler zuzuschneiden. Dabei ist es wichtig möglichst viel Wissen über diese Personen zu sammeln. Je mehr Daten hierbei zu Verfügung stehen desto besser. Spieler erklären sich jedoch nur selten dazu bereit, sich langwierig durch Formulare zu schlagen, um Spieleherstellern dabei zu helfen, Informationen zu erlangen. Deswegen ist es oftmals besser, diese Informationen direkt während des Spielens zu sammeln. Diese Aufzeichnungen können anschließend an einen Server geschickt werden, der diese Daten auswertet oder es kann ebenfalls sein, dass diese Daten direkt im Spiel verarbeitet werden. Es gibt einige *Frameworks* die dem Hersteller helfen an diese Daten zu gelangen. Ein Beispiel in der Web- und App-Entwicklung ist *Google Analytics* [13] und bei der Verwendung der *Game Engine Unity* gibt es unter anderem *Unity Analytics* [22]. So lässt sich bei in Entwicklung befindlicher Software leicht feststellen, in welchen Bereichen die Benutzer die meiste Zeit verbringen, wie diese beim Navigieren vorgehen, welche Aktionen getätigt werden bevor das Spiel verlassen wird und noch vieles mehr. Mit all diesen Informationen kann die Spielerfahrung für sämtliche Spieler anspruchsvoll gestaltet werden. Spieler, welche Probleme haben bestimmte Bereiche zu bewältigen, werden mit helfenden Gegenständen unterstützt oder es werden Passagen mit leichteren Gegnern freigeschaltet. Im Gegensatz dazu, können Gegner stärker gemacht werden, sollte es sich um besonders gute Spieler handeln, um auch diesen Spielern eine Herausforderung zu bieten. Bei Videospielen gibt es immer wieder Vertreter die aufgrund von Handlungen in der Spielwelt, Daten über den Spieler sammeln und die Spielwelt dementsprechend anpassen. Ein gutes Beispiel für solch eine Änderung ist ein dynamischer Schwierigkeitsgrad, der sich an dem Verhalten der Spieler orientiert. Werden diese Methoden jedoch zu offensichtlich eingebaut kann es dazu führen, dass diese Anpassung ausgenutzt wird, um schneller durch das Spiel zu schreiten. So kann

es sein, dass Spieler kurz vor einer schweren Passage absichtlich schlechter spielen um schwächeren Gegnern gegenüber zu stehen. Adams schreibt genauer über die Vor- und Nachteile eines adaptiven Schwierigkeitsgrades [7]. Die Implementierung einer solchen Maßnahme würde jedoch den Rahmen dieser Arbeit sprengen und ist somit kein Teil davon. In dieser Arbeit geht es lediglich darum, die Ausgangsbasis für solch eine mögliche Implementierung zu liefern. Genauer gesagt sollen die Fähigkeiten der Spieler (von hier an als *Skill* bezeichnet) beurteilt werden.

1.1 Motivation

In der Videospielbranche gibt es momentan einen sehr starken Trend in Richtung *Virtual-Reality* (VR) und *Augmented-Reality* (AR). Obwohl es diese Bereiche schon länger gibt, findet aktuell und in den letzten Jahren ein sehr starkes Wachstum statt. Gründe dafür sind unter anderem, dass Game-Engine Hersteller vermehrt VR-Funktionalität als Standardfeature zur Verfügung stellen [23] [12]. VR und AR wird somit für immer mehr Entwickler zugänglich gemacht. Große Fortschritte in der Entwicklungen mobiler Geräte führten zu Smartphones die mittlerweile leistungsstark genug sind, um VR und AR Spiele darauf laufen zu lassen. Dieser Trend ist auch an dem Erfolg von Augmented-Reality Spielen wie z. B. *Pokemon Go* erkennbar [8] [10]. Da es sich für diesen neuen Markt um ein noch relativ unerforschtes Gebiet handelt, entstehen neue Möglichkeiten zur Datengewinnung. Es werden neuartige Daten mit noch unbekanntem Potential generiert. Diese Daten können dazu verwendet werden, um Produkte dem Zielmarkt besser anzupassen. Mit dem wachsenden Markt in VR und AR werden Unternehmen nicht daran vorbei kommen, auch diese Daten auszuwerten und zu analysieren. Entwickler werden daher in Zukunft vermehrt versuchen auch diese Geräte dazu zu nutzen, um Daten durch das Verhalten der Benutzer zu erlangen. Da Analysen der Daten aus diesen Gebieten noch relativ unerforscht sind, widmet sich diese Arbeit der explorativen Analyse von Daten die aus einem dieser Gebiete stammen, nämlich *Augmented Reality*. Es werden altbewährte Methoden verwendet um neuartige Daten zu erforschen.

1.2 Problemstellung

Um den *Skill* von Spielern zu analysieren, werden üblicherweise Aktionen und Handlungen, die der Spieler ausführt, ausgewertet. Diese Arbeit widmet sich bewusst und gezielt nur den Daten, die durch die Nutzung von *AR* entstehen. In dem konkreten Fall ist das die virtuelle Kamera durch die der Spieler die virtuelle Spielwelt betrachtet. Die gesammelten Daten welche im Abschnitt 5.2 genauer beschrieben sind, werden dazu verwendet, um den *Skill* sämtlicher Spieler bzw. sämtlicher gespielten Spiele festzustellen.

Der *Skill*, der für ein gespieltes Level erstellt wird, soll nach dem Schulnotensystem zu fünf Gruppen zuordenbar sein. Es werden zunächst sämtliche Werte für die einzelnen Spiele analysiert, um zu sehen welche Daten es gibt und bei welchen Informationen es sich lohnt diese mit anderen Spielen zu vergleichen. Anschließend wird versucht einen Zusammenhang zwischen allen Spielen herzustellen. Die explorative Datenanalyse ist ein großer Teil der Arbeit, da noch nicht bekannt ist ob ein Zusammenhang zwischen den verfügbaren Daten und dem *Skill* besteht.

1.3 Zielsetzung

Das Spiel *Vancouver Maneuver* (siehe Kapitel 2) diene nicht nur für diese Arbeit als Grundlage. Kertesz beschäftigt sich in seiner Arbeit mit dem prozeduralen Generieren von Leveln für dieses Spiel [5]. Der fertige Algorithmus der referenzierten Arbeit, soll mit der Vorgabe eines Wertes, der den Schwierigkeitsgrad darstellt, Level erstellen können. Das Ziel ist es, dass mit Hilfe dieser Arbeit eine Methode entwickelt wird, die in das Spiel eingebaut werden kann, welche das Spiel zur Laufzeit analysiert und somit den *Skill* berechnet. Dieser Wert kann nach dem Abschluss eines jeden Levels dazu verwendet werden, um die darauffolgenden, generierten Level an den Spieler entsprechend anzupassen.

1.4 Aufbau dieser Arbeit

Nach diesen einleitenden Worten, die zu verstehen geben sollen, wieso diese Arbeit zustande gekommen ist, wird in Kapitel 2 zunächst das Videospiel *Vancouver Maneuver* beschrieben, welches ein maßgeblicher Teil dieser Arbeit war. Damit wird ein Überblick über die vorhandenen Spielelemente verschafft, um zu verstehen welche Daten für die Analyse zur Verfügung stehenden. Darauf folgt eine Erklärung wie üblicherweise bei explorativer Datenanalyse vorgegangen wird in Kapitel 3. Es wird beschrieben welche Methoden es gibt und welche Eigenschaften diese aufweisen. In Kapitel 4 wird auf die Rahmenbedingungen der Tests eingegangen und wie sichergestellt wurde, dass die Daten vergleichbar bleiben. Den Hauptteil der Arbeit stellt der konkrete Einsatz der Exploration dar, welcher in Kapitel 5 beschrieben wird. Es wird gezeigt welche Software verwendet wurde und welche Daten auf welche Weise verarbeitet wurden. Es wird auf die besten Ergebnisse und auf aufgetretene Probleme eingegangen. Die Arbeit endet mit Kapitel 6, welches das Resümee beinhaltet. Es wird beschrieben welche Schritte gut und welche eher schlecht funktioniert haben und es gibt einen Einblick in zukünftige Entwicklungen.

Kapitel 2

Vancouver Maneuver

Die statistische Analyse des Spielverhaltens wird anhand des Videospiels *Vancouver Maneuver* ermittelt. Das Spiel wurde als mehrsemestriges Projekt während des Masterstudiums umgesetzt. Es diente als Grundlage für die Auswertungen, da es sämtliche Daten aufzeichnete und zur Verfügung stellte. Die Entwicklung wurde mit der *Game-Engine Unity* [21] umgesetzt. Die AR-Funktionalität wurde mit Hilfe der Bibliothek *Vuforia* [25]¹ implementiert.²

2.1 Voraussetzungen

Bei dem Spiel handelt es sich um ein *Augmented-Reality*-Videospiel. Voraussetzungen um spielen zu können, sind der Spielplan und ein Smartphone, auf welchem die entsprechende App installiert ist. Bei dem Spielplan handelt es sich um ein *Image-Tracker* (siehe Abb. 2.1). Das bedeutet, dass mit Hilfe einer entsprechender Software, die Ausrichtung und Position des *Trackers* erkannt werden kann, um virtuelle Objekte darauf abzubilden. Das Erkennen geschieht mittels Betrachtung durch die Kamera des Handys in Echtzeit. Die virtuelle Welt kann somit unmittelbar auf dem Bildschirm des Handys dargestellt werden. Bei einem *Tracker* handelt es sich um eine Grafik welche vom Entwickler beliebig gestaltet werden kann. Das Bild, welches zur Verfügung gestellt wird, sollte jedoch viele markante Punkte beinhalten. Diese werden als *Feature Points* bezeichnet und sind notwendig, um die Ausrichtung des *Trackers* zu erkennen. Auf der Website von *Vuforia* kann überprüft werden, ob ein Bild viele dieser Punkte aufweist. Die Grafik wird analysiert

¹ Zum Zeitpunkt der Entwicklung des Spiels wurde Vuforia noch von der Firma Qualcomm[16] angeboten, dem ursprünglichen Entwickler. Vuforia wurde mittlerweile von PTC (Parametric Technology Corporation) [15] aufgekauft.

² *Vuforia* kann für die Plattformen Android, iOS und WindowsPhone benutzt werden. Es gibt außerdem ein *Unity-Package* welches in der *Unity-Engine* verwendet werden kann [24].

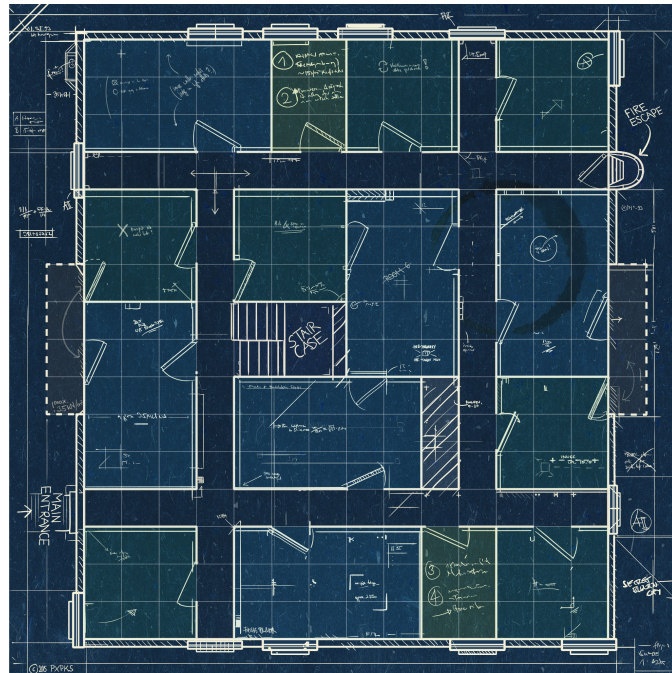


Abbildung 2.1: Die Grafik, die das Spielfeld darstellt und ausgedruckt werden kann.

und erhält eine Wertung von einem bis zu fünf Sternen, wobei ein Stern schlecht und fünf Sterne sehr gut bedeutet. Je besser die Wertung ist, desto besser funktioniert das *Tracking*. Das heißt unter anderem, dass der *Tracker* schneller erkannt wird und das *Tracking* auch bei hektischen Bewegungen noch gut funktioniert. Ebenso kann eine korrekte Berechnung der Orientierung durchgeführt werden, selbst wenn nur ein kleiner Ausschnitt der Grafik sichtbar ist. Der *Tracker* kann ausgedruckt werden und dient ohne weitere Aufbereitung als Spielbrett. Der *Tracker* kann in beliebiger Größe ausgedruckt werden. Für ein gutes Spielerlebnis sollte der Ausdruck jedoch möglichst groß sein. Im besten Falle sollte das ausgedruckte Exemplar mindestens füllend auf einem DIN A1³ platziert werden.

Das Spiel wurde für die Plattform *Android* entwickelt. Sobald die App auf dem Smartphone installiert und gestartet wurde, kann das virtuelle Spielfeld durch das Handy betrachtet werden, indem die Kamera auf den *Tracker* gerichtet und dieser erkannt wird (siehe Abb. 2.2). Obwohl die Software auf allen Smartphones laufen sollte, gilt es zu beachten, dass dieses Spiel einen erheblichen Rechenaufwand verursacht. Eine dementsprechend starke Hardware ist deswegen für ein flüssiges Spielerlebnis notwendig. Der Grund dafür ist, dass zu jedem Zeitpunkt das aktuelle Bild, welches durch die laufende

³ Das entspricht $59,4 \times 84,1$ cm [19].



Abbildung 2.2: Die virtuelle Welt kann durch die Kamera des Smartphones betrachtet werden. Die Spielelemente werden auf dem ausgedruckten Spielplan dargestellt.

Kamera geliefert wird, nach den *Feature Points* durchsucht werden muss. Die Lagesensoren des Handys werden ebenfalls sehr stark beansprucht, um die projizierte Spielwelt möglichst passgenau darstellen zu können. Zusätzlich gibt es noch die Elemente die für jedes Videospiel notwendig sind, wie z. B. die Darstellung sämtlicher 3D-Modelle. Selbst bei einer kleinen Spielwelt, wie es bei Vancouver Maneuver der Fall ist, wird der Prozessor und der Grafikchip des Smartphones sehr stark belastet. Zusätzlich wird die gesamte Spiellogik auf dem Handy berechnet. Zudem wird für den Mehrspielermodus noch eine aktive WLAN-Verbindung benötigt, welche auch gewisse Kapazitäten benötigt. All dies führt zu einem sehr hohen Stromverbrauch. Der Akku des Smartphones wird dadurch erheblich schneller entladen, als dies bei vielen anderen Apps oder Spielen der Fall ist.

2.2 Gameplay

Bei dem Spiel handelt es sich um ein Puzzle-Game. Es gilt die Levels des Spiels möglichst schnell durchzuspielen. Die Spielfiguren können weder verletzt noch getötet werden. Es gibt auch keine anderweitigen Gefahren, die ein vorzeitiges Beenden des Spiels mit sich bringen würden. Die Aufmerksamkeit soll vollständig der Lösung der Rätsel gewidmet werden. Die Spieler versuchen zu dem Zielobjekt, einem Safe, zu gelangen und diesen zu öffnen. Dieser Safe lässt sich jedoch nur mit einem bestimmten Schlüssel öffnen. Um sich diesen Schlüssel zu verschaffen, muss der Spieler mehrere kleine Zwischenrätsel lösen. Diese gestalten sich je nach Schwierigkeitsgrad des Levels von relativ offensichtlich bis hin zu sehr knifflig. Solche Rätsel sind Kombinationen aus klassischen Schalter- und Laser-Rätseln. Ein einfaches Beispiel für ein Schalter-Rästel wäre wie folgt: Ziel ist es, durch eine Tür zu gelangen. Die Tür ist verschlossen, wird jedoch geöffnet so lange sich der Spieler auf einem Trittschalter befindet. Sobald der Spieler den Schalter verlässt, schließt sich die Tür wieder. Die Passage wird gelöst indem eine Kiste auf dem Schalter platziert wird. Bei einem Laser-Rästel muss ein Lichtstrahl, durch Platzieren von Spiegeln zu einem dafür vorgesehenes Element gelenkt werden. Die Rätsel werden jedoch um eine eigene Komponente erweitert. Die wesentlichste Änderung bezieht sich darauf, dass sämtliche Räume im Spiel verschoben und rotiert werden können (siehe Abschnitt 2.2.3). In jedem Level stehen zwei Spielfiguren zur Verfügung, die abwechselnd gesteuert werden können. Für einfache Rätsel ist dies nicht unbedingt erforderlich. Wird jedoch versucht den Level möglichst schnell zu lösen, so ist dies die empfohlene Strategie, damit lange Laufwege vermieden werden. Um komplexere Rätsel zu bewältigen, müssen jedoch beide Spielfiguren bewegt und koordiniert eingesetzt werden. Es gibt einen Einspieler- und einen Zweispielermodus. Als Einzelspieler werden beide Spielfiguren rundenbasiert gesteuert. Der Spieler kann die Figuren nach belieben einsetzen und zwischen den Figuren zu jedem Zeitpunkt wechseln. Der Zweispielermodus kann nur mittels funktionierender WLAN-Verbindung gespielt werden. Zu Beginn des Spiels wird jedem Spieler eine Spielfigur zugewiesen. Es wird rundenbasiert abgewartet bis der Mitspieler seinen Zug beendet.

2.2.1 Steuerung

Während des Spielens sieht der Benutzer ein Fadenkreuz, welches mittig am Bildschirm platziert ist. Sämtliche Interaktionen im Spiel werden mit Hilfe dieses Fadenkreuzes getätigt. Dies bedeutet, dass sich der Spieler in der realen Welt bewegen muss um verschiedene Gegenstände in der virtuellen Welt anzuvisieren.

Buttons

Der Bildschirm wird durch eine gedachte senkrechte Linie in zwei Tasten unterteilt, welche für die Eingaben im Spiel verantwortlich sind. Der rechte Teil dient als Interaktions-Button. Mit diesem lassen sich vorrangig die Figuren in der Spielwelt navigieren. Durch betätigen des Buttons wird die Spielfigur auf das Feld unter dem Fadenkreuz geschickt. Befindet sich ein Objekt auf einem der benachbarten Felder der Figur, so wird eine Interaktion mit dem Spielobjekt ausgeführt. Hierbei kann es sich z. B. um einen Schrank handeln der geöffnet werden kann, oder um einen Gegenstand der verschoben werden kann. Der linke Button wird dazu verwendet, um die Spielwelt zu verändern. Genauer gesagt können damit Räume angehoben und neu angeordnet werden. Der Spieler muss dafür mittels Fadenkreuz den entsprechenden Raum anvisieren und die linke Hälfte des Bildschirms des Smartphones berühren. Solange der Finger den Kontakt mit dem Bildschirm nicht verliert, kann der Raum bewegt werden. Dies geschieht durch Bewegen in der realen Welt. Der Raum wird an der Stelle positioniert, an der mit Hilfe des Fadenkreuzes der Raum hinprojiziert wird. Solange der Button gedrückt ist, kann auf der gegenüberliegenden Seite des Bildschirms gewischt werden, um den Raum zu drehen. Sobald die Wisch-Geste erkannt wird, dreht sich der Raum um 90° . Wird von oben nach unten gewischt entspricht dies einer Rotation im Uhrzeigersinn. Ein Wischen in die entgegengesetzte Richtung führt entsprechend eine Rotation gegen den Uhrzeigersinn aus.

2.2.2 Spielmechaniken

Die Spielfiguren bewegen sich auf einem festgelegten Raster und können mit Gegenständen in der virtuellen Welt interagieren, sollten sie sich direkt auf einem der anliegenden Felder befinden. Um das Level zu lösen gibt es keine Zeitvorgabe und auch die einzelnen Spielzüge sind zeitlich nicht begrenzt. In jedem Zug können beliebig viele Felder betreten werden. Es können ebenfalls beliebig viele Interaktionen durchgeführt und somit Aktionen in der Spielwelt ausgelöst werden.

2.2.3 Levelstruktur

Das virtuelle Level setzt sich aus mehreren Räumen zusammen. Die genaue Anzahl hängt jedoch vom jeweiligen Schwierigkeitsgrad ab. Das Einführungslevel, welches dazu dient den Spieler mit den Spielelementen vertraut zu machen, besteht beispielsweise aus drei Räumen. Das komplexeste Rätsel welches designet wurde, wird mit neun Räumen gebildet. In der aktuellen Implementierung hätten theoretisch bis zu 49 Räume Platz. Genau genommen könnten somit 48 Räume verwendet werden, da ein Slot frei bleiben muss, um das Verschieben der Räume zu ermöglichen. Diese Räume können zu jedem Zeitpunkt durch den Spieler, der gerade an der Reihe

ist, verschoben werden. Es gibt keine Beschränkungen welche Räume bewegt werden können. Jeder Raum kann beliebig auf dem Raster platziert werden, solange sich kein anderer Raum auf dem Feld befindet. Während des Verschiebe-Vorgangs kann der sich momentan in Bewegung befindliche Raum rotiert werden. Die genaue Steuerung ist im Abschnitt 2.2.1 beschrieben. Sämtliche Positionierungs- und Rotationsaktionen können beliebig oft ausgeführt werden.

Raumaufbau

Jeder Raum ist in 25 Spielfelder unterteilt, welche quadratisch auf einem Raster aufgeteilt sind. Sämtliche Räume müssen über mindestens eine Tür verfügen, welche dazu dient einer Spielfigur das Betreten und Verlassen möglich zu machen. Türen können sich an jedem äußeren Feld eines Raumes befinden. Liegen zwei Räume direkt aneinander und passen sämtliche Türen der anliegenden Kanten beider Räume zusammen, so können die Spielfiguren durch eine der Türen den jeweilig anderen Raum betreten. Dies wird dem Spieler jedoch noch dahingehend erschwert, dass es Türen gibt, die versperrt sind. Diese können durch unterschiedliche Mechanismen entsperrt werden. Eine Variante der Türen lässt sich mittels im Boden befindlicher Schalter öffnen und eine andere durch Lichtsensoren.

2.2.4 Spielelemente

Bewegliche Gegenstände

Bewegliche Gegenstände sind beispielsweise Kisten. Bei den Kisten gibt es je nach Umgebung visuelle Unterschiede. Die visuelle Repräsentation wird zum Zeitpunkt des Startens des Levels festgelegt. Bei der gewöhnlichen Kiste handelt es sich um eine Holzkiste. Sollte die Kiste jedoch aus einem Badezimmer stammen, so wird diese als Waschmaschine angezeigt. Funktionstechnisch unterscheiden sich diese Kisten jedoch nicht. Beide Varianten können von der Spielfigur durch eine Aktion gezogen oder geschoben werden. Ein weiterer Gegenstand, der gezogen werden kann, ist der Spiegel. Dieser kann zudem, während sich die Spielfigur im Zieh-Modus befindet, durch ein erneutes Antippen gedreht werden. Der Spiegel dient hauptsächlich dazu Laserstrahlen umzuleiten, kann jedoch ebenfalls auf einem Trittschalter platziert werden, um diesen auszulösen.

Lichtschraken

Das Hauptelement eines Laser-Rätsels ist der Lichtstrahl. Im konkreten Fall bei *Vancouver Maneuver* ist dieser Strahl ein Laser, der bestimmte elektronische Geräte aktiviert bzw. deaktiviert. Sobald der Lichtstrahl unterbrochen wird, ändert sich der Zustand des betroffenen Gerätes. Der Strahl ist nicht

tödlich für Spielfiguren und kann somit durchschritten werden. Es wird lediglich für die Dauer des Aufenthalts bzw. des Durchschreitens der Stromkreislauf unterbrochen. Sämtliche Gegenstände unterbrechen den Strahl, sollte dieser einen Sensor aktiviert haben, wird dieser Sensor demnach für die Zeit der Unterbrechung deaktiviert.

Sicherheitskamera

Eine Sicherheitskamera hindert den Spieler daran bestimmte Felder zu betreten bzw. diese zu durchschreiten. Kameras können durch Schalter bzw. Lichtschranken ihren Aktivitätszustand wechseln. Sollte eine Figur ein Feld betreten haben und die Kamera wird anschließend aktiviert, sodass der Spieler nun innerhalb des Sichtfeldes der Kamera steht, kann dieser Spieler keine Aktionen mehr ausführen und somit auch nicht mehr das Feld verlassen. Diese Spielfigur muss von dem Mitspieler befreit werden, indem dieser die Kamera wieder entsprechend deaktiviert. Die Kamera schwenkt im aktiven Zustand immer wieder über den überwachten Bereich. Visuell werden die überwachten Felder durch eine Rot gestrichelte Textur gekennzeichnet. Um den Zustand der Felder weiter hervorzuheben, sind die betroffenen Felder zudem animiert, die Textur wird sozusagen über die Felder geschoben.

Schrank

Als Behälter für Gegenstände werden in dem Spiel Schränke verwendet. Mit diesen Schränken kann interagiert werden, wenn die Spielfigur auf dem Feld davor steht. Schränke können den Schlüssel enthalten, der benötigt wird um den Safe zu öffnen. In schwierigeren Rätseln kann es vorkommen, dass mehrere Schränke im Level platziert sind. In diesem Fall ist der Schlüssel aber nur in einem Schrank. Das bedeutet, dass der Schrank mit dem Schlüssel zuerst gefunden werden muss. Beide Spieler teilen sich ein Inventar. Somit muss der Tresor nicht unbedingt von der Spielfigur geöffnet werden, die den Schlüssel gefunden hat.

Schalter

Schalter werden in dem Spiel dazu verwendet, um den Aktivitätszustand von bestimmten Geräten in der Spielwelt zu ändern. Eine Variante von Schaltern sind Trittschalter. Diese befinden sich am Boden und bedecken ein ganzes Spielfeld. Sie werden aktiviert, indem sich ein Objekt darauf befindet. Dabei kann es sich um eine der beiden Spielfiguren oder um einen beweglichen Gegenstand (siehe Abschnitt 2.2.4) handeln. Ein weiterer Schalter, der im Spiel vorkommt, ist ein Lichtsensor. Dieser ist aktiv solange ein Laserstrahl direkt darauf gerichtet ist.

Tresor

Der Tresor stellt das Ziel eines jeden Levels dar. Dieser kann nur mit dem entsprechenden Schlüssel geöffnet werden. Bei einer Interaktion mit dem Tresor ohne Schlüssel erhalten die Spieler einen Hinweis, dass dieser benötigt wird.

Kapitel 3

Explorative Datenanalyse

Bei der explorativen Datenanalyse wird versucht Zusammenhänge zwischen vorgegebenen Daten zu ermitteln und diese zu verstehen. Es sind noch keinerlei Charakteristiken oder Merkmale der Datenmengen bekannt. Diese Analyse stellt somit den ersten Schritt dar, um sich ein Bild von den Daten zu machen. Um herauszufinden welche Variablen dabei verwendet werden können, stehen eine Vielzahl von möglichen Methoden zur Verfügung. Die verschiedenen Methoden lassen dabei unterschiedliche Verhaltensmuster der Wertmengen zum Vorschein kommen. Die explorative Datenanalyse ist oftmals keine genaue Wissenschaft. Sie hilft jedoch dabei einen bestimmten Erwartungswert zu ermitteln. Dieser sollte so gut wie möglich, die verwendeten Daten abdecken, um zukünftige Werte möglichst genau vorherzusagen. Wie Tukey [6, S. 13f] schreibt:

The most important maxim for data analysis to heed, and on which many statisticians seem to have shunned, is this: „Far better an approximate answer to the *right* question, which is often vague, than an *exact* answer to the wrong question, which can always be made precise. “Data analysis must progress by approximate answers, at best, since its knowledge of what the problem really is will at best be approximate.

Sinngemäß übersetzt bedeutet dies folgendes: Bei der Datenanalyse ist es wichtig zu beachten, dass eine angenäherte Antwort auf die *richtige* Frage, welche oftmals unspezifisch ist, besser ist, als eine *genaue* Antwort auf die falsche Frage, welche immer genau formuliert werden kann. Es muss mit angenäherten Ergebnissen gearbeitet werden, da das Wissen über das Problem, im besten Falle, nur ungefähr ist.

Zur Analyse können verschiedene Verfahren herangezogen werden. Es gibt visuelle Verfahren, welche grafisch den Sachverhalt widerspiegeln und es möglich machen viel Information schnell zu interpretieren, und nicht-visuelle Verfahren, welche ein berechnetes Ergebnis liefern. Für eine gute Explorati-

on sind beide Typen wesentlich. Da es eine große Anzahl an verschiedenen Verfahren gibt ist es nicht möglich, jede im Detail durchzunehmen. Daher wird in diesem Kapitel vor allem auf jene genauer eingegangen, die bei der explorativen Analyse dieser Arbeit Verwendung fanden und interessante Ergebnisse lieferten.

3.1 Einfache statistische Maßzahlen

Zu den nicht-visuellen Verfahren zählen vor allem die einfachen statistischen Maßzahlen. Diese geben einen groben Überblick über die vorhandenen Daten. Es reicht jedoch nicht aus, sich lediglich auf diese Resultate zu verlassen, da es vorkommen kann, dass sich Daten ähneln, die jedoch komplett unterschiedliche Eigenschaften aufweisen (siehe Abschnitt 3.1.3).

3.1.1 Mittelwert

Eine der wohl einfachsten Methoden sich den ersten Überblick über Daten zu verschaffen, ist der Durchschnitt. Es gibt unter anderem, das arithmetische Mittel. Um diese von einer Wertemenge $\{x_i\}$ mit der Anzahl n an Werten zu berechnen

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i.$$

Median

Eine weiterer Mittelwert ist der Median. Dieser hat den Vorteil gegenüber dem arithmetischem Mittel, dass Ausreißer das Ergebnis nicht im gleichen Maße beeinflussen. Bei einer sortierten Mengen an Zahlen wird dieser für eine ungerade Anzahl an Elementen folgendermaßen berechnet:

$$\tilde{x} = x_{(n+1)/2}$$

und für eine gerade Anzahl an Elementen:

$$\tilde{x} = \frac{x_{n/2} + x_{n/2+1}}{2}.$$

Der Median einer Datenmenge mit einer ungeraten Anzahl an Elementen stellt außerdem sicher, dass der berechnete Wert auch in der Datenmenge vorhanden ist.

Gestutztes Mittel (trimmed mean)

Eine weitere resistente Methode ist es die Datensätze zu trimmen. Dies bedeutet, dass gleich viele der niedrigsten und der höchsten vorkommenden

Daten verworfen werden. Die Anzahl wird üblicher Weise in Prozent angegeben. Bei einem um 5% gestutztem Mittel werden 5% der niedrigsten und 5% der höchsten Werte verworfen. Nach dem Trimmen wird der Datensatz behandelt als ob er vollständig wäre. Diese Methode ist dazu gedacht Verteilungen mit langen Enden ähnlicher einer normalen Verteilung zu machen. Handelt es sich jedoch bereits um eine Normalverteilung, bringt diese Methode einen Verlust an Effizienz mit sich. Besitzt die Verteilung sehr lang gezogene Enden so funktioniert das Trimmen ebenfalls nicht perfekt, so Tukey [6, S. 19].

Winsorisiertes Mittel (Winsorized mean)

Ein Spezialfall des Trimmens ist das winsorisierte Mittel. Die getrimmten Daten werden nicht verworfen, sondern die Variablen erhalten den Wert des nächsten nicht abgeschnittenen Elements. Für normale Daten, ist das winsorisierte Mittel stabiler als das gestutzte Mittel. Bei jeglicher Art des Trimmens eines Datensatzes sollte der Trimmvorgang an die Daten entsprechend angepasst werden. Wie Tukey schreibt, [6, S. 19], ist der größte Vorteil des gestutzten Mittels und des winsorisierten Mittels jener, dass der Standarderror kleiner ist, als bei einem ungetrimmten Durchschnitt. Selbst beim Trimmen mit einer fixen Anzahl werden bereits bessere Ergebnisse sichtbar. Diese verbessern sich noch weiter wenn der Trimmvorgang an die Daten entsprechend angepasst wird. Dies bedarf jedoch einer separaten Studie.

3.1.2 Streuung

Es gibt Datensätze bei denen Mittelwerte nicht sinnvoll miteinander verglichen werden können. Es kann der Fall sein, dass die Streuung hingegen Informationen über die Daten preisgibt. Die Varianz gibt den quadratischen Abstand zum Erwartungswert an. Die Standardabweichung entspricht der Quadratwurzel der Varianz.

Trimmen

Die Streuung kann gleichermaßen, wie es bei den Mittelwerten der Fall ist, getrimmt werden. Hierbei kann ebenfalls auf einfaches Werwerfen der Daten, durch Stutzen oder auf Abändern der Werte durch Winsorieren, zurückgegriffen werden.

Spannweite

Eine weitere Eigenschaft die verglichen werden kann ist die Spannweite. Diese bildet sich aus der Subtraktion des kleinsten Wertes der Datenmenge von dem größten Wert der Datenmenge.

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Tabelle 3.1: Die vier Wertemengen (I, II, III und IV) des Anscombe-Quartetts. Jede Menge setzt sich aus elf Wertpaaren mit den Komponenten x und y zusammen.

3.1.3 Anscombe-Quartett

Das Anscombe-Quartett wurde im Jahre 1973 von dem englischen Statistiker Francis Anscombe entwickelt [1]. Es veranschaulicht die Wichtigkeit der visuellen Darstellung bei der Datenanalyse und den Effekt von Ausreißern auf das Ergebnis. Das Quartett besteht aus vier Mengen von Datenpunkten. Jede Menge wird dabei aus elf Wert-Paaren (x, y) gebildet, die in der Tabelle 3.1 angeführt sind. Sämtliche Mengen verfügen über nahezu identische einfache statistische Maßzahlen. Der Mittelwert von x jeder Datenmenge ergibt genau 9. Die Varianz von x ist genau 11. Der Mittelwert von y jeder Datenmenge ergibt genau 7,50. Die Varianz von y ist genau 4,122 oder 4,127. Die Korrelation zwischen x und y ist für sämtliche Datensätze 0,816. Die lineare Regression entspricht für alle Datensatzpaare der Gleichung $y = 3,00 + 0,500x$. Bei der Betrachtung der Visualisierungen der Wertmengen, werden deren Charakteristiken jedoch sofort erkennbar (siehe Abb. 3.1).

3.2 Wertetabelle

Die Daten direkt in einer Tabelle zu betrachten, kann sich manchmal ebenfalls als sehr nützlich herausstellen. Exakte Korrelationen lassen sich auf diese Weise zwar vermutlich nicht bestimmen, es kann jedoch sein, dass bestimmte auffällige Muster auftreten und erkannt werden. Werte die sich nicht oder nur sehr wenig ändern fallen leicht auf. Es kann sein, dass sich Werte stets innerhalb einer bestimmten Bandbreite bewegen. Möglicherweise

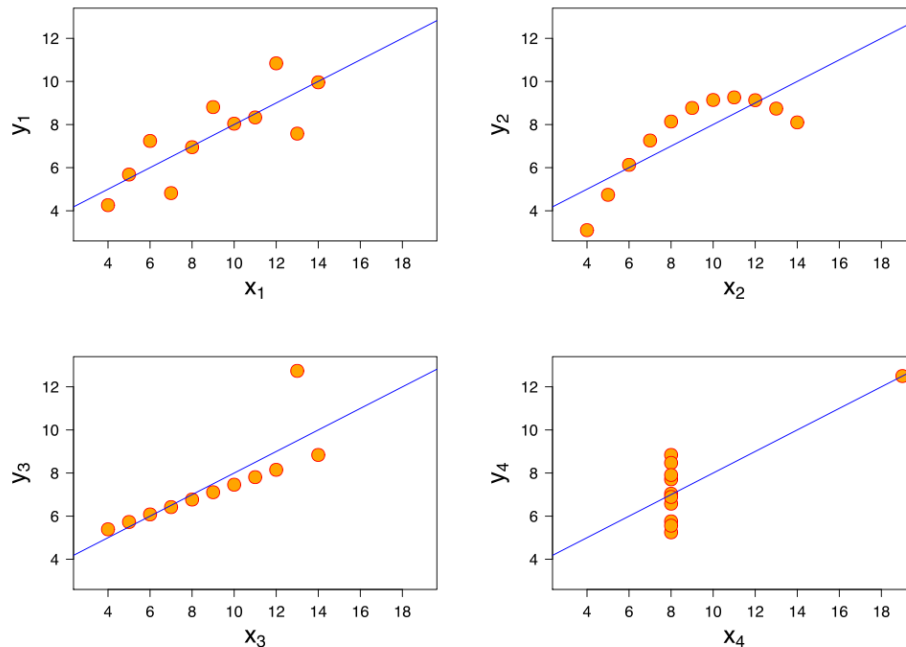


Abbildung 3.1: Die grafisch dargestellten Datensätze des Anscombe-Quartetts zeigen eindeutige Charakteristiken, welche bei dem Einsatz einfacher statistischer Maßzahlen nicht erkennbar werden. Grafik aus [14].

ist es der Fall, dass gewisse Werte immer in Kombination mit den gleichen Folgewerten auftreten. Ein gutes Beispiel für das Erkennen solcher Merkmale bieten die Wertmengen des Anscombe-Quartetts (siehe Abb. 3.1). Es zeigt sich, dass es sogar bei einem relativ kleinen Datensatz schwer fällt einen klaren Zusammenhang zu erkennen. Das wiederholte Vorkommen der selben Zahl, in den x -Werten des Datensatzes IV sticht sofort ins Auge. Ebenso fällt der Ausreißer auf, der dieses Muster zerstört. Je größer die Datensätze sind, desto schwerer wird dieser Ansatz. Daher ist es ratsam auf Visualisierungen der Daten zurückzugreifen.

3.3 Histogramm

Mit Hilfe eines Histogramms wird ebenfalls einiges an Information ersichtlich. Es veranschaulicht die Verteilung und die Häufigkeit von Werten. Es wird gezeigt mit welcher Frequenz bestimmte Werte auftreten. Auf diese Weise können Charakteristiken der Verteilung bestimmt werden. Dabei können ebenfalls Muster auftreten. Es kann sein, dass ein Wert besonders oft auftritt oder dass es immer wieder bestimmte Spitzen gibt. Es muss jedoch

darauf geachtet werden, dass die Balkenanzahl des Histogramms entsprechend angepasst wird. Diese muss den Daten entsprechen. Andernfalls kann es sein, dass Information verloren geht (siehe Abschnitt 5.8).

3.4 Diagramm

Die vermutlich hilfreichsten Mittel sich relativ schnell einen Überblick über Charakteristiken von Daten zu verschaffen, bieten Diagramme. Hierbei gibt es die verschiedensten Möglichkeiten Datensätze zu visualisieren und zu vergleichen. Bei den folgenden handelt es sich in keinster Weise um eine vollständige Aufzählung, sondern nur um jene Diagramme, die sich bei dieser Exploration als am wertvollsten herausstellten.

3.4.1 Streudiagramm (scatter plot)

Eine einfache und sehr hilfreiche Methode, um eine mögliche Korrelation zwischen zwei Werten zu sehen, ist das Streudiagramm. Dieses entspricht einem kartesischen Koordinatensystem und besitzt somit eine x - und eine y -Achse. Auf jeder dieser Achsen wird eine der Datenreihen eingetragen, um zu sehen ob diese in einem Zusammenhang stehen. Aufgrund der Punktwolke, die sich daraus ergibt, lassen sich gegebenenfalls bestimmte Charakteristiken erkennen. Es kann sein, dass sich die Punkte in eine bestimmte Richtung neigen. Liegen diese Punkte größtenteils auf einer Geraden, so gibt es vermutlich einen linearen Zusammenhang (siehe Abschnitt 3.7.3). Es kann ebenfalls sein, dass sich aus den Punkten eine Kurve ergibt oder andere Muster, die mit bloßem Auge erkennbar sind. Dazu zählen unter anderem immer wiederkehrende Schwankungen. Ein weiteres Merkmal, welches auftreten kann, sind kleine Gruppierungen, die als Cluster bezeichnet werden. Die Punkte innerhalb eines Clusters haben meist eine Eigenschaft gemeinsam, die der Grund für dieser Gruppenbildung ist.

Oftmals ist es hilfreich, anstelle von identischen Punkten, bestimmte Eigenschaften der Wertmengen darzustellen. Dieses Erkennungsmerkmal hilft dabei, sich ein besseres Bild von der Verteilung machen zu können. Wird z. B. eine Datenmenge der Bevölkerung betrachtet, hilft es vielleicht bei einer Zuordnung zu einer gewissen Eigenschaft, je nach Altersgruppe die Punkte in unterschiedlichen Farben einzuzichnen. Somit wird der zweidimensionale Plot sozusagen um eine dritte Dimension erweitert. Sollte die Punktwolke sehr dicht, und es somit nicht klar erkennbar sein, wo die dichtesten Punkteansammlungen (Hotspots) liegen, so hilft es die Punkte transparent darzustellen (siehe Abb. 3.2). Die Transparenz sollte je nach Anzahl der sich überlappenden Punkte angepasst werden.

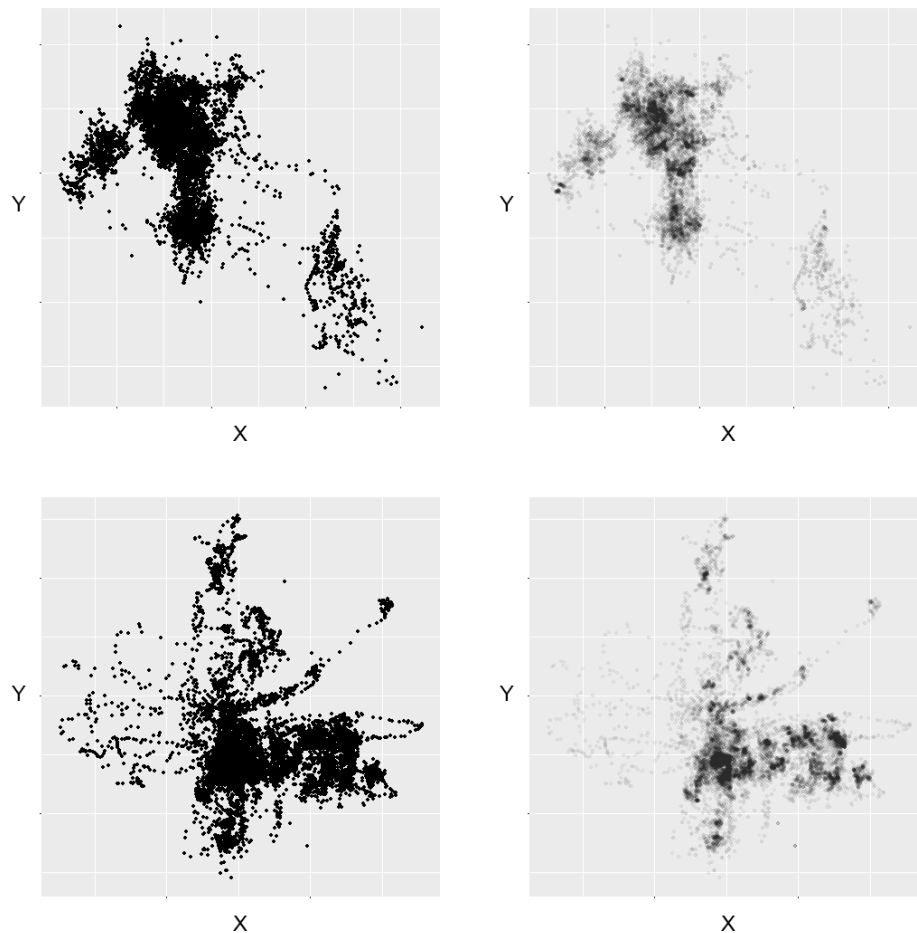


Abbildung 3.2: Der Vorteil von transparenten Punkten wird deutlich. In den rechten Bildern werden die dichten Stellen (Hotspots) leicht erkennbar.

Liniendiagramm (line graph)

Ein Liniendiagramm, kann auf die gleiche Weise wie das Streudiagramm verwendet werden. Der Unterschied dabei ist, dass die Punkte durch eine Linie verbunden sind (siehe Abb. 3.3). Daher macht es recht wenig Sinn diese Darstellung zu wählen, wenn es sich um eine Punktwolke handelt. Diese Form der Visualisierung bietet sich unter anderem bei Daten an, die über eine Dauer gemessen wurden. Es wird veranschaulicht wie stark eine zeitliche Komponente die Werte verändert. Möglicherweise werden Schwankungsbereiche oder wiederkehrende Muster ersichtlich. Ein stetiges Wachstum oder ein andauernder Abfall sind mit diesem Diagramm ebenfalls sehr leicht erkennbar.

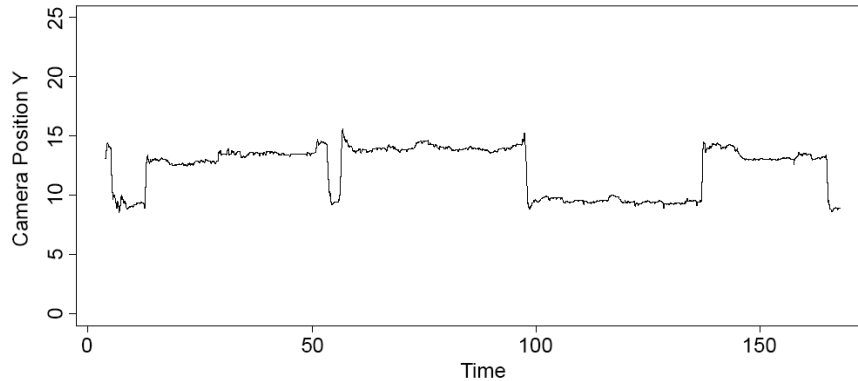


Abbildung 3.3: Dieses Beispiel eines Liniendiagramms zeigt, wie sich der Wert einer Variable, über eine bestimmte Dauer hinweg verhält. Der Wert ändert sich nur innerhalb einer gewissen Spannweite, tut dies aber nur kurzfristig, stabilisiert sich anschließend wieder und geht dann wieder zum Ursprungswert zurück. Es wird ersichtlich, dass der Wert nach einer Änderung, immer wieder am selben Minimum und Maximum zum Stehen kommt. Wieso sich der Wert ändert, ist nicht ersichtlich, es kann jedoch überprüft werden, ob sich andere Variablen im gleichen Zeitraum ähnlich verhalten.

3.4.2 3D-Diagramm

Wie bereits in Abschnitt 3.4.1 beschrieben, ist es auch möglich mit einem Streudiagramm eine zusätzliche Dimension zu betrachten, indem bestimmte Eigenschaften an Stelle der Punkte verwendet werden. Manchmal reicht dies jedoch nicht aus bzw. wird der Informationsgehalt des Diagramms nicht erhöht. Sollte es dennoch erforderlich oder gewünscht sein drei Dimensionen in Kombination zu visualisieren, so können dreidimensionalen Diagrammen verwendet werden. Eine Möglichkeit ist es, die Daten lediglich in einer 3D-Grafik auf ein 2D-Bild zu projizieren (siehe Abb. 3.4). Dies hat den Vorteil, dass diese Bilder keine zusätzlichen Programme zum Betrachten benötigen und sie können z. B. einfach in Dokumente eingebunden werden. Das Problem ist jedoch, dass es vorkommen kann, dass Werte hinter anderen liegen und somit versteckt werden. Es muss darauf geachtet werden, dass sämtliche Daten aus dem dargestellten Blickwinkel erkennbar sind. Andernfalls müssen mehrere Bilder angefertigt werden, um die Daten aus verschiedenen Perspektiven zu zeigen. Bolker [2, S. 18] beschreibt, dass es schwer ist, einen wirklich guten 3D-Plot zu erstellen und dass für einen Ausdruck ein guter Betrachtungswinkel ausgesucht werden muss. Da dies für die Exploration jedoch keine wesentliche Rolle spielt, bietet sich eine andere Methode an. Es ist ebenso möglich, diese Datenmenge in einem dreidimensionalen Diagramm anzuzeigen, welches es ermöglicht, dieses interaktiv zu rotieren.

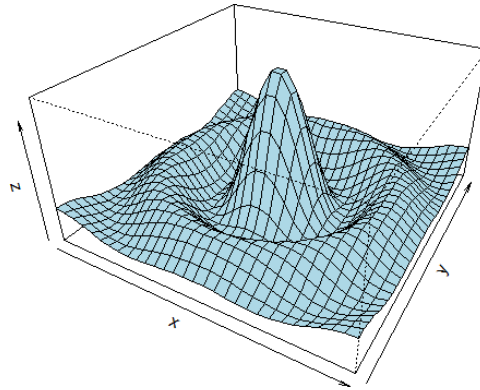


Abbildung 3.4: Ein Beispiel für eine dreidimensionale Visualisierung. Aufgrund des Blickwinkels könnte es sein, dass Informationen hinter der Spitze nicht erkennbar sind.

Somit können die Daten um einiges genauer unter die Lupe genommen werden. Der Nachteil hierbei ist natürlich, dass diese interaktive Version nicht in Dokumente eingebunden werden kann. Außerdem wird ein eigenes Programm oder Werkzeug benötigt, um das Betrachten der Daten möglich zu machen. Beide Nachteile können bei der explorativen Datenanalyse jedoch vernachlässigt werden.

Um das bestmögliche Resultat zu erzielen, sollten die Vorteile der beiden Welten kombiniert werden. Während der explorativen Phase sollte daher ein interaktives Diagramm verwendet werden, um die wesentlichsten Merkmale der Daten zu erhalten. Hierbei kann ein guter Betrachtungswinkel gefunden werden. Für eine anschließende Präsentation der Daten sollte das erlangte Wissen dazu genutzt werden, um eine ausreichende zweidimensionale Projektion der Daten möglich zu machen. Diese kann anschließend ohne Weiteres in Dokumente eingebunden werden. Der Vorteil, den ein dreidimensionales Diagramm bringt, wird in Abbildung 3.5 ersichtlich. Werden jeweils nur zwei Achsen des dreidimensionalen Vektors ausgegeben, fällt es schwer sich ein Bild von den Daten zu machen, selbst wenn sämtliche Kombinationen zur Verfügung stehen. Im dreidimensionalen Diagramm hingegen, wird die Struktur sofort ersichtlich, selbst wenn es sich dabei nur um einen Blickwinkel handelt. Bei einer dreidimensionalen Darstellung ist es gleichermaßen möglich, eine zusätzliche Datenmenge zu betrachten, indem die Punkte eingefärbt oder durch bestimmte Werte ersetzt werden, wie dies bereits bei einem zweidimensionalen Streudiagramm der Fall ist.

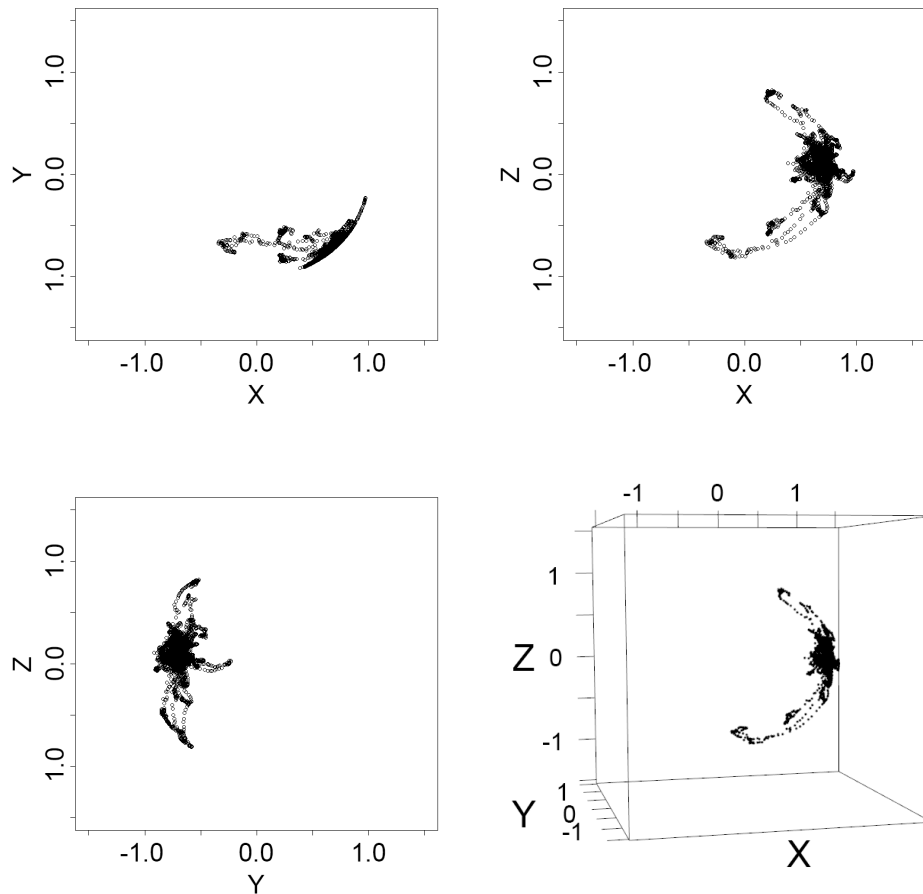


Abbildung 3.5: Die Punkte stellen sämtliche Blickrichtungen einer Kamera über den Spielverlauf hinweg dar. Es wird eine zweidimensionale Sicht auf diese dreidimensionale Punktwolke aus jeder Achsenkombination und zusätzlich ein 3D-Diagramm der Daten gezeigt.

3.5 Betrachtung von Teilmengen

Um sich ein Bild von einer großen Datenmenge zu machen, ist es oftmals hilfreich, die Daten in kleinere Gruppen zu unterteilen und diese zu betrachten. Je nach Größe der Originalgruppe und gewünschtem Ergebnis reicht es oftmals schon aus, die Daten in nur wenige Subgruppen zu teilen. Dies sollte immer im Bezug zu einer Variable ausgeführt werden und nicht willkürlich. So könnten Ergebnisse für Frauen und Männer separat betrachtet werden, um zu sehen ob es einen sichtbaren Unterschied unter der Berücksichtigung der Geschlechter gibt. Unterschiedliche Altersgruppen könnten ebenso einen

Rückschluss auf bestimmte Verhaltensmuster zulassen. Es ist jedoch immer sehr stark von den Daten abhängig, wie bei der Unterteilung vorgegangen wird.

3.6 Versteckte Variablen

Es ist nicht immer der Fall, dass eine explorative Datenanalyse das Verhalten einer bestimmten Variable erklärt. Selbst wenn das gewünschte Ergebnis nicht erzielt wird, ist es möglich, dass ein Zusammenhang zwischen Daten auftritt, der sich auf etwas anderes zurückführen lässt. So kann es vorkommen, dass die Korrelation demografische Ursachen hat. Möglicherweise verhalten sich junge Personen anders als ältere. Vielleicht führen Frauen andere Aktionen aus als Männer. Es könnte sein, dass die Daten Rückschlüsse auf die Bildung oder die Erfahrungen der Testpersonen zulassen.

Riepl schreibt über Scheinkorrelation [17]. Er führt das Beispiel an, dass in einer Region mit vielen Störchen, viele Kinder geboren werden. Der Grund liegt in einer gemeinsamen Ursache der betrachteten Variablen, welche in diesem Fall z. B. der Industrialisierungsgrad der betroffenen Region ist. Daher ist es bei der Erstellung der Daten wichtig, personenbezogene Information und andere Daten die erhalten werden können, mit in die Datenbank zu speichern. Somit ist es möglich bei gemeinsamen Auftreten von Verhaltensmerkmalen auf solche Werte zu prüfen. Es gilt jedoch zu beachten, dass nicht zu viele unwesentliche Daten verwendet werden, da mehr verfügbare Daten auch bedeuten, dass mehr Rauschen erzeugt wird. Daher sollte bei der Selektion der Daten vorsichtig vorgegangen werden. Andernfalls kann es dazu führen, dass wesentliche Merkmale nicht mehr erkannt werden können.

3.7 Regression

3.7.1 Regressionsanalyse

Regressionsanalyse bezeichnet einen Prozess der dazu dient die Beziehung zwischen einer abhängigen und einer oder mehrerer unabhängigen Variablen herauszufinden. Diese Regression kann dann dazu genutzt werden um zukünftige Erwartungswerte zu ermitteln. Das Ziel ist es die Summe der quadrierten Residuen aller Werte (Sum of Squared Errors (SSE)) zu minimieren. Residuen bezeichnen den Fehler zwischen dem erwarteten und dem tatsächlichen Wert. Einerseits werden durch die Quadrierung sämtliche Zahlen positiv und andererseits werden große Abweichungen hervorgehoben.

3.7.2 Korrelation

Korrelation bezeichnet die Beziehung zwischen zwei statistischen Variablen. Der Korrelationskoeffizient gibt an, wie gut die Beziehung zwischen den zwei

Werten ist. Ein positiver Koeffizient bedeutet, dass bei einer Steigerung von x ebenso y steigt. Somit ergibt sich, dass bei einem negativen Koeffizienten eine Steigerung von x einen Abfall von y bedeutet. Der Wert des Korrelationskoeffizienten reicht von -1 bis $+1$. Der Wert von 1 bzw. -1 bedeutet, dass eine perfekte Beziehung besteht und jeder Wert durch ein lineares Modell erklärt werden kann. Bei einem Wert von 0 besteht keine Beziehung zwischen den Variablen. Liegt der absolute Wert bei $0,3$ handelt es sich um eine schwache Beziehung, bei $0,5$ ergibt dies eine mittelmäßige Beziehung und bei $0,7$ wird von einer guten Beziehung gesprochen.

3.7.3 Lineare Regression

Bei der linearen Regression wird versucht den linearen Zusammenhang zwischen einer (einfache lineare Regression) oder mehreren verschiedenen Variablen (multiple lineare Regression) zu erkennen. Letzten Endes soll es möglich sein, aus diesen einen oder diesen mehreren unabhängigen Variablen, die abhängige Variable zuverlässig vorherzusagen. Der absolute Wert des Korrelationskoeffizienten soll demnach möglichst hoch sein. Eine lineare Regressionsgleichung wird in der Form

$$y = \beta_0 + \beta_1 x + \epsilon$$

angegeben. Dabei beschreibt β_0 den Achsenabschnitt (Intercept) und β_1 die Steigung (Slope). Bei ϵ handelt es sich um den Error, also um den Unterschied zwischen den Erwartungswerten und den tatsächlichen Werten. Bei der Analyse kann es vorkommen, dass mehrere unabhängige Variablen immer zusammen auftreten und gleichermaßen mit der abhängigen Variable korrelieren. Dies bedeutet, dass beide Werte gleichermaßen in Beziehung zu der abhängigen Variable stehen. Daher kann eine der beiden Variablen verworfen werden, da diese keinen Mehrwert für die Analyse mit sich bringt. Erwähnenswert ist noch, dass das lineare Modell nicht unbedingt für jeden Wert Sinn ergeben muss. Es handelt sich lediglich um das am besten angepasste Modell, welches versucht die meisten Werte so gut wie möglich zu erklären. Nähere Details zur linearen Regression in [3].

3.8 Kreisförmige Daten

Es gibt verschiedene Möglichkeiten eine Rotation darzustellen. Horn beschreibt diese ausführlich [4, S. 632]. Im Videospielbereich werden hauptsächlich zwei davon verwendet, einerseits eulersche Winkel und andererseits Quaternionen.

3.8.1 Eulerwinkel

Der Vorteil der Eulerwinkel ist, dass diese ein einfach zu verstehendes Konzept darstellen. Eine Rotation wird durch einen dreidimensionalen Vektor mit den Elementen x , y und z dargestellt. Jedes Element bezieht sich auf eine Achse. Jeder Wert gibt die Rotation um die entsprechende Achse an. Ein Problem, welches hierbei jedoch auftreten kann ist eine *kardanische Blockade* (*Gimbal Lock*). Daher wird in den vielen Fällen auf eine andere Methode der Rotation zurückgegriffen.

3.8.2 Quaternion

In *Game-Engines* ist es üblich Rotationen als Quaternionen zu speichern. Oftmals werden dem Benutzer zwar vorrangig Euler Winkel als Schnittstelle zugänglich gemacht, diese werden jedoch intern auf Quaternionen umgerechnet und abgespeichert. Der Grund dafür liegt darin, dass das Konzept der Quaternionen nicht ganz trivial ist. Horn [4, S. 632ff] beschreibt dieses ausführlich. Eine wesentliche Eigenschaft von einer Quaternion ist, dass sie aus den vier Elementen x , y , z und w besteht. Der Vorteil von Quaternionen ist unter anderem, dass eine kardanische Blockade nicht auftreten kann. Um dies zu erreichen könnte zwar ebenfalls eine Drehmatrix verwendet werden, bei einem Einsatz von Quaternionen ist der Speicherbedarf jedoch geringer. Pro Rotation müssen nur vier Werte gespeichert werden, anstelle von neun, wie es bei einer dreidimensionalen Rotationsmatrix der Fall wäre.

3.8.3 Summen von Kreisförmigen Daten

Die Berechnung von Mittelwerten ist bei kreisförmigen Daten nicht ohne Weiteres möglich. Dies wird anhand eines vereinfachten Beispiels in einem zweidimensionalen Raum veranschaulicht. Dafür werden ein Kreis und zwei Zahlen, die den Winkel und somit den Punkt auf dem Kreis angeben, benötigt. Ein voller Kreis entspricht 360° . Wird der Durchschnitt beispielsweise zwischen 165° und 195° gebildet, ergibt dies 180° . Das ist in diesem Fall korrekt (siehe Abb. 3.6). Rotiert ein Objekt jedoch stattdessen um den 0° -Winkel entstehen Probleme. So würde der Durchschnitt von 345° und 15° ebenfalls zu 180° führen. Das gewünschte Ergebnis wäre jedoch 0° (siehe Abb. 3.6). Eine Möglichkeit dieses Problem zu lösen wäre, dass die entsprechende Rotation auf einen Positionsvektor in einem kartesischen Koordinatensystem umgerechnet wird (siehe Abb. 3.8). Der Mittelwert dieser Vektoren kann nun ohne Weiteres berechnet werden. Der Ergebnis-Vektor kann anschließend noch normalisiert werden, um einen Punkt auf dem Kreis oder der Oberfläche der Kugel zu ergeben oder er kann verwendet werden, um diesen zurück auf Winkelwerte zu rechnen. Ein Problem tritt jedoch auf, wenn der berechnete Durchschnitt genau am Ursprung liegt. Somit ergibt sich ein Vektor der nicht normalisiert und auch nicht auf Winkel zurück-

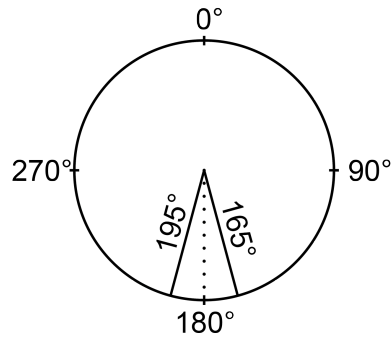


Abbildung 3.6: Wird der Mittelwert von Winkeln berechnet, funktioniert dies solange die Werte weniger als 180° auseinander liegen. In der Grafik wird aus 165° und 195° ein Durchschnitt von 180° gebildet.

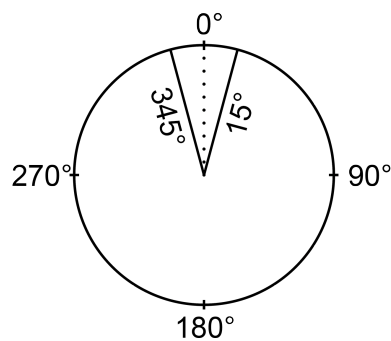


Abbildung 3.7: Der Durchschnitt der beiden Winkelwerte 15° und 345° sollte 0° ergeben, wie dies in der Grafik ersichtlich wird. Das arithmetische Mittel der beiden Werte ergibt jedoch 180° .

gerechnet werden kann. Handelt es sich bei den Rotationsdaten eines dreidimensionalen Raumes, wird ein Ursprungsvektor angenommen und dieser wird entsprechend der gegebenen Rotation gedreht. Das Ende des Vektors entspricht nun einem dreidimensionalen Punkt in einem kartesischen Koordinatensystem. Für einen praktischen Einsatz und Beispiele aufgetretener Fehler, siehe Abschnitt 5.8.2.

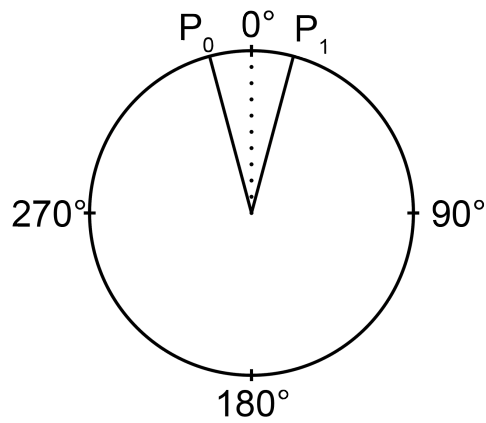


Abbildung 3.8: Die beiden Punkte P_0 und P_1 sind zweidimensionale Vektoren mit den Elementen x und y . Die entsprechenden Werte eines Winkels α werden mittels $x = \cos(\alpha)$ bzw. $y = \sin(\alpha)$ berechnet. Der Mittelwert aus P_0 und P_1 kann nun korrekt berechnet werden. Der Winkel des Ergebnis-Vektors β wird mit $\beta = \text{atan2}(\bar{y}, \bar{x})$ berechnet.

Kapitel 4

Testumgebung

4.1 Versuchsaufbau

Bei sämtlichen Versuchen war ein Teammitglied anwesend um bei Fragen zur Verfügung zu stehen. Der Spielplan wurde immer in der gleichen Größe zur Verfügung gestellt und bei sämtlichen Spielen wurde mit dem gleichen Handy getestet. Dies stellte sicher, dass sämtliche Bedingungen für alle Testpersonen gleich waren und hierbei keinerlei Verfälschung der Ergebnisse auftrat. Es gab ein Einführungslevel, in dem die Testpersonen die Steuerung und sämtliche Spielelemente kennenlernen konnten. Anschließend sollten die Spieler versuchen die folgenden Level so schnell wie möglich zu lösen. Es gab drei verschiedene Level, welche einen zunehmenden Schwierigkeitsgrad aufwiesen. Die Tester konnten die Level mehrmals durchspielen und versuchen ihre Zeit zu verbessern. Das wiederholte Durchspielen sollte dabei helfen, Veränderungen vom Spielverhalten bei verbessertem Können zu erkennen. Mit dem Beenden jedes Levels wurde automatisch eine *Replay*-Datei gespeichert. In dieser Datei wurden alle wichtigen Informationen gespeichert, damit das Spiel originalgetreu rekonstruiert werden kann. Die Kameraposition und -rotation wurde in jedem Frame gespeichert, um die Bewegungen der Testperson genau nachbilden zu können. Bei sämtlichen anderen Ereignissen wurde lediglich das betroffene Event und ein Zeitstempel gespeichert, da dies vollkommen ausreicht um es korrekt wiedergeben zu können. Diese *Replay*-Dateien wurden binär gespeichert um eine möglichst kleine Dateigröße zu erzielen. Dies bedeutet jedoch auch, dass diese Dateien nur mit dem geeigneten Programm verwendet werden können.

4.2 Dateiformat

Bei dem Programm, welches die *Replay*-Dateien abspielen kann, handelt es sich um eine modifizierte Version des Spiels. Diese Version war aus praktischen Gründen für den PC ausgelegt. Die wichtigste Modifikation war, dass

es nicht möglich war Benutzer-Eingaben zu verarbeiten. Sämtliche Befehle, die das Spiel vorantreiben, werden durch Events ausgelöst, die aus den *Replay*-Dateien ausgelesen werden. Das Abspielen erfolgt in Echtzeit. Es wäre zwar theoretisch möglich diese Wiederholungen schneller abzuspielen, damit ginge jedoch ein Verlust an Genauigkeit der Daten einher. Während des Abspielens wird eine CSV-Datei erstellt, welche sämtliche Informationen enthält, die für die statistische Auswertung notwendig sind. CSV steht für *Comma Seperated Values* und ist ein übliches Format um tabellarisch angeordnete Daten lesbar abzuspeichern, wie Bolker schreibt [2, S. 3]). Ein großer Vorteil von CSV liegt darin, dass keine speziellen Programmen notwendig sind um diese Dateien zu öffnen. Da es sich um einfache Textdateien handelt, können solche Dateien mit jedem Texteditor betrachtet und bearbeitet werden. Aus Gründen der Zeitersparnis wurde das Programm dahingehend angepasst, dass es möglich ist, beliebig viele *Replays* automatisch hintereinander abzuspielen. Dadurch werden die CSV-Dateien für sämtliche somit geladenen Spiele generiert und es entfällt die Notwendigkeit den Prozess für jede Datei einzeln zu starten. Der Grund für die getrennte Erstellung von der binären *Replay*-Datei und der CSV-Datei liegt darin, dass es im Nachhinein noch möglich sein soll, Daten die eventuell nicht von Anfang an berücksichtigt worden sind, zugänglich zu machen. Aufgrund der Tatsache, dass das Spiel originalgetreu nachbildet werden kann, können sämtliche Informationen aus dem Abspielen der Wiederholung gewonnen werden, die auch aus dem echten Spiel hätten extrahiert werden können. Aus den CSV-Dateien lässt sich weder das Spiel rekonstruieren, noch lassen sich zusätzliche Daten gewinnen.

4.3 Beurteilung der Spielerfähigkeiten

Die Bewertung der Spielerfähigkeiten (*Skill*) der einzelnen Spiele wird nicht automatisch berechnet sondern erfolgt händisch. Alle *Replays* werden nochmals abgespielt und betrachtet. Dabei wird jedes Spiel nach bestimmten Kriterien bewertet. Wichtig bei der Beurteilung des *Skills* war, dass lediglich spiel-relevante Aktionen gezählt wurden. Dazu zählt nicht wie der Spieler mit der Kamera umgeht oder worauf er seinen Blick richtet.

4.3.1 Bewertungskriterien

Ein guter Spieler zeichnet sich dadurch aus, dass dieser den Level in möglichst wenigen Zügen beendet. Lange Laufwege werden vermieden indem die beiden Spielfiguren in unterschiedliche Bereiche des Levels geschickt werden. Dadurch können schneller unterschiedliche Aktionen in verschiedenen Bereichen der Spielwelt ausgeführt werden. Durch abwechselndes Benutzen der Spielfiguren kann das Level somit schneller beendet werden. Unnötig ausgeführte Schritte rücken den *Skill* in schlechtere Bereiche. Dazu zählt

z. B. Kisten zu verschieben, wenn dies nicht erforderlich ist, oder das Platzieren von Gegenständen in Räumen, in denen diese nicht gebraucht werden. Ebenso fällt es negativ ins Gewicht, wenn Spieler unnötig lange Wege auf sich nehmen. Handlungen die getätigt werden, nur um diese kurz darauf wieder rückgängig zu machen, sind einer guten Bewertung ebenfalls abträglich. Hierbei muss jedoch beachtet werden, dass es vorkommen kann, dass kleine Fehler aufgrund der ungewohnten Steuerung auftreten. So kann es beispielsweise passieren, dass Spieler nur ein Feld zu weit laufen, weil sie beim Zielen Probleme hatten. Dies wurde beim Ansehen der Wiederholung jedoch eindeutig klar, da die Spieler den Fehler sofort korrigierten. Da die Beherrschung der Steuerung ebenfalls Kennzeichen eines guten Spielers sind, wirkt sich dies auf die Bewertung zwar negativ aus, der Einfluss ist jedoch weniger gravierend als eine falsche Herangehensweise an ein Rätsel.

Kapitel 5

Konkreter Einsatz in Vancouver Maneuver

Das Ziel der Exploration ist es, den *Skill* der Spieler aus vorhandenen Daten bestimmen zu können. Als Vergleichsbasis dient der bereits festgelegte Wert.

5.1 RStudio

Zur Auswertung wurde die Programmiersprache *R* [11] und das Programm *RStudio* [18] verwendet. Bei *R* handelt es sich um eine Programmiersprache für statistische Berechnungen. *R* wurde als Open-Source Gegenstück zu der Sprache *S* entwickelt. Bereits mit der Standardfunktionalität lässt sich eine Vielzahl mathematischen und statistischen Operationen durchführen. So lassen sich komplexe Operationen mit recht einfachen Funktionsaufrufen durchführen. Unter anderem ist es möglich Visualisierungen von Daten anzufertigen. Die Möglichkeiten der Sprache werden zudem durch zusätzliche Pakete erweitert und ergänzt.

RStudio ist eine kostenlose und quelloffene IDE.¹ Das Programm verfügt über einen Texteditor, einen Interpreter, einen Debugger und eine integrierte Hilfe, welche die Dokumentation zu den Standardfunktionen liefert. Des weiteren gibt es die Möglichkeit Erweiterungen zu durchsuchen und zu installieren. Die Oberfläche teilt sich in vier Hauptbereiche (siehe Abb. 5.1). Bei dem ersten Bereich handelt es sich um den Texteditor. Dieser dient dazu den Code von geöffneten Skripten zu betrachten und diesen gegebenenfalls zu bearbeiten. Der Texteditor verfügt über Syntax-Highlighting² und einfaches Autocomplete.³ Es besteht die Möglichkeit die geschriebenen

¹IDE steht für *Integrated Development Environment* und bezeichnet ein Programm welches sämtliche Funktionalität liefert die zum Entwickeln der Software gebraucht werden

²Bestimmte Schlüsselwörter werden in einer bestimmten Textfarbe dargestellt um diese leichter zu erkennen.

³Sobald die ersten Buchstaben getippt wurden, werden sämtliche Variablen und Befehle

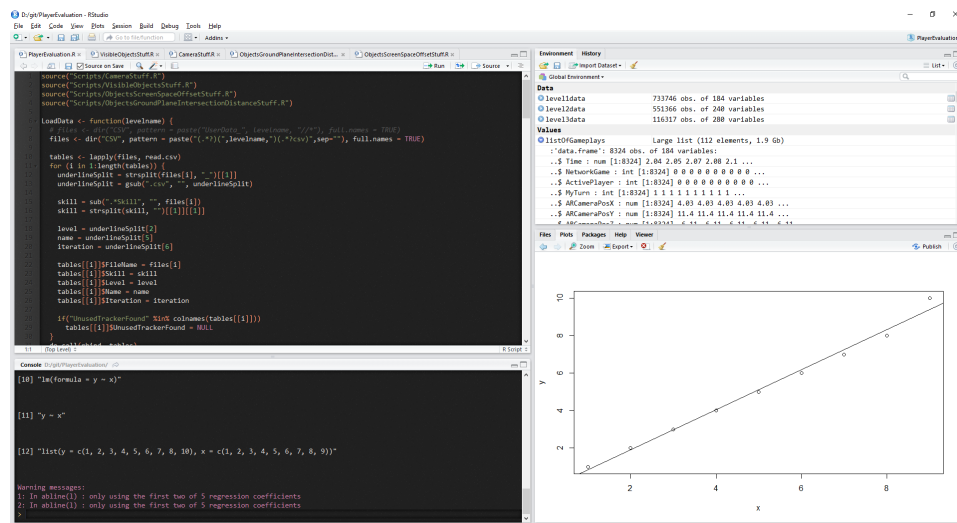


Abbildung 5.1: Das Bild zeigt die Oberfläche von *RStudio*. Links oben befindet sich der interne Texteditor mit einem geöffneten R-Skript. Darunter wird die Konsole angezeigt, die gerade Werte und entsprechende Warnungen ausgibt. Rechts oben befindet sich der Reiter *Environment*, welcher alle aktuellen Variablen enthält. Unterhalb wird ein Scatterplot dargestellt.

Skripte sofort nach jeder Speicheraktion automatisch zu importieren. Sollte ein Skript mit Syntax-Fehler geladen werden, gibt *RStudio* beim Importieren einen Error in der Konsole aus. Darunter befindet sich eine Konsole. Mit dieser können Befehle ausgeführt werden. Aus praktischen Gründen sollten sich diese eher kurz halten. Ansonsten wäre es angebracht ein eigenes Skript dafür anzulegen. Dieses kann dann importiert und der darin enthaltene Code ausgeführt werden. In der Konsole stehen sämtliche Variablen und Funktionen zu Verfügung die in dem Variablen-Viewer sichtbar sind. Diese können somit zusätzlich zu den Standardfunktionen einfach benutzt und ausgeführt werden. Der Reiter *Environment* zeigt sämtliche erstellten Daten. Dies beinhaltet importierte Funktionen, erstellte Variablen und erstellte *DataFrame*-Objekte⁴. Letztere können durch einen Doppelklick in einer eigenen Ansicht genauer betrachtet werden. Der letzte Hauptbereich wird dazu verwendet um angefertigte Grafiken anzuzeigen. Wird ein Bild mittels Code erstellt, scheint dieses sofort hier auf. Werden mehrere Visualisierungen angefertigt, so können diese mittels Schaltflächen durchgeblättert werden. Alle Grafiken bleiben so lange erhalten bis diese gelöscht werden oder das Programm geschlossen wird. Es besteht auch die Möglichkeit sämtliche visuellen Repräsentationen, mittels R-Skript, direkt als Bilddateien, auf die

aufgelistet die mit den entsprechenden Buchstaben beginnen.

⁴*DataFrame* stellt ein *R*-eigenes Format dar und entspricht tabellarisch angeordneten Datensätze mit denen in *RStudio* schnell und einfach gearbeitet werden kann.

Programm 5.1: Dieser Code zeigt wie die `plot()`-Funktion mit den wichtigsten Parametern in R verwendet werden kann. Dieser Code kann auch ohne Zeilenumbrüche in eine einzige Zeile geschrieben werden. Um den Code übersichtlich zu halten, erweist sich diese Weise jedoch als hilfreich.

```
1 plot(  
2   x = data$x,  
3   y = data$y,  
4   xlim = c(0, 10),  
5   ylim = c(0, 5),  
6   type = 'l',  
7   main = "Title",  
8   xlab = "Width",  
9   ylab = "Height"  
10 )
```

Festplatte zu speichern. Des weiteren gibt es noch Bereiche wie z. B. *History* in dem sämtliche Befehle aufgezeichnet werden, um diese schnell wieder ausführen zu können oder um diese schnell in ein R-Skript zu importieren. Die restlichen Bereiche sind ähnlich wie in vielen anderen Programmen. So gibt es z. B. noch einen *Help*-Reiter in dem Informationen zu den Funktionen bereit gestellt werden oder einen *Files*-Reiter in dem die Dateien des Projekte aufscheinen und organisiert bzw. geöffnet werden können.

5.1.1 Verwendete Pakete

Für die ersten Schritte wurde die standardmäßige Diagramm-Funktionalität von *RStudio* verwendet. Diese Funktion ist leicht zu bedienen und liefert gute Ergebnisse für einfache Visualisierungen. Sie wird mit dem Befehl `plot()` aufgerufen und benötigt als Parameter zumindest die Datensätze für die *x*- und *y*-Achse. Der Bereich der Darstellung kann durch Beschränkungen der Achsen angepasst werden. Beschränkungen werden stets als zweidimensionaler Vektor angegeben, die den kleinst- und größtmöglichen Wert darstellen. Ein Vektor wird mittels `c(x,y)` erstellt. Um den angezeigten Bereich zu begrenzen, kann `xlim=` bzw. `ylim=` verwendet werden. Es kann auch noch die Art der Visualisierung geändert werden. So lässt sich z. B. an Stelle des standardmäßigen Streudiagramms mit Kreisen, ein Liniendiagramm erstellen, indem der Parameter `type=` mit `'l'` befüllt wird. Selbstverständlich lassen sich die einzelnen Achsen, sowie die gesamte Grafik beschriften. Standardmäßig wird andernfalls der Name des verwendeten Datensatzes verwendet. Im Programm 5.1 wird gezeigt wie all dies in Kombination aussieht. Es gibt noch ein paar andere Einstellungen die vorgenommen werden können, die erwähnten waren jedoch für die Datenanalyse am wichtigsten. Trotz dieser Einstellungsmöglichkeiten gelangen die Standardplots an ihre Grenzen.

Im Verlauf dieser Arbeit wurde somit auf das Paket *ggplot2* [26] gesetzt, um besser aufbereitete Grafiken zu ermöglichen. Damit gestaltet es sich leichter die Visualisierungen mit Farben zu versehen. Es war möglich bestimmte Datenpunkte einzufärben oder mit zusätzlichen Informationen zu versehen. So konnte z. B. an Stelle eines Kreises, eine andere Information aus dem Datensatz verwendet werden. All dies erleichterte es erheblich Zusammenhang in den Daten ersichtlich zu machen und zu erkennen. Für einen bestimmten Fall war ein dreidimensionales Diagramm erforderlich. Dabei wurde auf eine interaktive Visualisierung gesetzt. Um diese zu erstellen wurde das Paket *plot3d* [20] verwendet.

5.2 Welche Daten stehen zur Verfügung

Bei der Auswertung der Daten wurde darauf Wert gelegt, lediglich Daten zu verwenden, die aufgrund der Nutzung von *Augmented-Reality* vorhanden sind. Konkret handelt es sich hierbei um den Sichtbereich des Spielers im Spiel. Das Sichtfeld in der Spielwelt bewegt sich synchron mit den Bewegungen des Smartphones in der realen Welt. Hierbei werden sämtliche Positions- und Rotationsänderungen unmittelbar in die Spielwelt übertragen. Die wichtigsten Daten sind vor allem die Position im dreidimensionalen Raum, welche als Vektor mit den drei Elementen x , y und z angegeben wird und die Rotation, die im Gegensatz dazu, als Quaternion (siehe Abschnitt 3.8.2) mit den vier Dimensionen x , y , z und w abgespeichert wird. In der CSV-Datei wird jedes Element von mehrdimensionalen Werten in einer eigenen Spalte gespeichert. Dies erleichtert den Zugriff darauf und vereinfacht den Arbeitsprozess.

5.2.1 Anzahl der Objekte im Sichtfeld

Durch das gegebene Sichtfeld des Spielers kann festgestellt werden, ob ein bestimmtes Spielobjekt im Augenblick sichtbar ist oder nicht. Aufgrund der Tatsache, dass die Kameraposition und -rotation ununterbrochen aufgezeichnet wird, lässt sich dieser Wert somit präzise über die gesamte Spieldauer hinweg ermitteln. Dieser Wert kann für jedes Objekt in der Spielwelt erstellt werden. Dadurch kann bestimmt werden, wie viele Objekte zu einem beliebigen Zeitpunkt gleichzeitig sichtbar sind.

5.2.2 Objektposition am Bildschirm

Bei jedem Objekt wird zudem noch die Position am Bildschirm aufgezeichnet. Diese Information steht als zweidimensionaler Vektor (x, y) zur Verfügung. Die Achsen werden in gewohnter Manier separat in der CSV-Datei gespeichert. Der Wertebereich der beiden Achsen reicht von -1 bis $+1$. Wo bei -1 die linke bzw. obere und $+1$ die rechte bzw. untere Kante darstellt.

Durch diese Konvention muss bei der Auswertung der aufgezeichneten Daten keine Rücksicht auf die unterschiedlichen Bildschirmauflösungen verschiedener Smartphones genommen werden. Dadurch bleiben sämtliche Daten unabhängig vom verwendeten Gerät und somit vergleichbar. Dieser Wert wird einerseits durch die Kamerarotation und andererseits auch durch die Position beeinflusst. Durch eine Drehbewegung des Spielers um seine eigene Achse wandern die Objekte sozusagen aus dem Bildschirm hinaus. Bei einer Positionsveränderung bei der sich die Distanz zum *Tracker* ändert, werden diese Werte ebenfalls verändert. Mit zunehmendem Abstand rücken die Objekte näher in Richtung Bildschirmmitte. Mit abnehmender Distanz hingeben, verteilen sich die Spielobjekte großzügiger über den Bildschirm.

5.2.3 Objektdistanz zum Fokuspunkt des Spielers

Der Fokuspunkt bezeichnet den Schnittpunkt der aus dem Vorwärtsvektor der Kamera und dem Spielfeld berechnet wird. Die Richtung des Vektors verläuft stets in die Blickrichtung der Kamera und startet in der Mitte des Bildschirms. Für jedes Objekt wurde zu jedem Zeitpunkt die Distanz zwischen der Objektposition und diesem Fokuspunkt gespeichert. Die Entfernung wird hierbei in Spielwelt-Einheiten angegeben. Die Idee hinter diesen Aufzeichnungen war, dass dieser Wert unabhängig von der Distanz zwischen dem Spieler und dem Spielfeld in der realen Welt ist. Diese Werte bleiben unverändert solange der Blick auf den selben Punkt gerichtet ist, ungeachtet der Position und der Rotation. Diese Werte stellten sich jedoch als weitestgehend uninteressant heraus.

5.3 Aufbereitung der Daten

5.3.1 Verwendung der Daten in *RStudio*

Bolker schreibt wie Daten in *RStudio* geladen und verwendet werden können [2, S. 3ff]. Sämtliche CSV-Dateien werden zunächst importiert und in das *R*-Format *DataFrame* umgewandelt. Bei diesem Prozess wurden einige Spalten in die Tabellen eingefügt. Der Name einer jeden CSV-Datei enthält folgende Informationen: Präfix, Levelname, Datum und Uhrzeit des Testspiels, Skill, Name und gespielte Iteration des Levels und sieht folgendermaßen aus:

`UserData_Alex4Rooms1_2016-03-4-21-31-40_Skill13_HP_1.`

Der Präfix `UserData` diene hierbei lediglich als Unterscheidungsmerkmal zwischen den Replay-Dateien und den CSV-Dateien. Der Levelname bezieht sich auf die interne Benennung, wie sie bei der Entwicklung des Spiels entstanden ist. Die Verwendung des Dateinamens zur Speicherung von gewissen Informationen bietet folgende Vorteile: Da die originale *Replay*-Datei binär ist und somit nach dem Erstellen nicht mehr ohne weiteres verändert werden kann, ermöglicht diese Lösung eine schnelle und einfache Möglichkeit

zusätzliche Informationen zu den Dateien zu speichern. Diese Informationen könnten zwar in die CSV-Datei geschrieben werden, diese Daten würden jedoch verloren gehen, sollten die *Replays* erneut konvertiert werden. Außerdem müsste ein spezielles Programm verwendet werden, um Daten in die CSV-Dateien zu schreiben bzw. müssten die Dateien stets importiert, bearbeitet und dann wieder exportiert werden. Der Arbeitsaufwand steht jedoch in keinem Verhältnis zu dem Resultat, da ein Vielfaches der Zeit benötigt wird um lediglich eine Information z. B. den *Skill* mit den Daten zu verknüpfen. Es würde noch die Möglichkeit geben, eine zusätzliche Datei anzulegen welche die Werte mit den Dateien in Verbindung bringt. Die Folge davon wäre, dass diese Datei immer aktuell gehalten werden muss. Dies bedeutet, dass es neben einem Mehraufwand auch noch eine potentielle Fehlerquelle gibt. Informationen in den Dateinamen zu integrieren erwies sich somit als die beste Methode.

5.3.2 Probleme

Nachdem schon einige *Replays* erstellt waren, wurde durch ein anderes Problem bedingt, entwicklungsintern etwas an der PC-Version des Spiels geändert. Es wurde ein zusätzliches Kamera-Objekt erstellt, welches für die Arbeit mit der generativen Levelerstellung benötigt wurde. Im Zuge dessen wurde das originale Kamera-Objekt umbenannt. Die *Replays* die ab diesem Zeitpunkt aufgezeichnet wurden, hatten zudem zusätzliche Spalten, die Informationen der zusätzliche Kamera enthielten. Die Folge war, dass die alten und neuen *Replay*-Formate nicht mehr kompatibel waren. Dieses Problem wurde ebenfalls beim Import behoben, indem nur die Spalten verwendet wurden, die für diese Arbeit gebraucht wurden. Außerdem wurden die Namen der betroffenen Spalten vereinheitlicht.

Während des Spielens kann es vorkommen, dass der *Tracker* nicht korrekt erkannt wird. Dies kann der Fall sein wenn der Blickwinkel zu flach ist und somit zu wenig *Feature*-Punkte zum Berechnen vorhanden sind. Das gleiche Problem kann auftreten, wenn sich die Kamera dem Spielfeld zu weit nähert und somit der betrachtete Ausschnitt zu wenig Daten für die Berechnungen liefert. Das Papier welches für das Spielfeld benutzt wird, beeinflusst ebenfalls die Qualität des *Trackings*. Reflexionen und Glanzlichter, die beim Verwenden von Fotopapier auftreten können, stören den Algorithmus außerordentlich. Sollte dies der Fall sein versucht die Software das Spielfeld mit den wenigen Daten, trotzdem noch möglichst gut abzubilden. Da dies leider nur sehr selten zu einem guten Ergebnis führt, werden Daten nicht verwendet die zu Zeitpunkten auftreten, in denen der *Tracker* nicht von der Software erkannt wird.

5.3.3 Ausreißer

Hintergrundwissen über Ausreißer hilft bei der Entscheidung ob diese entfernt oder behalten werden sollen. Kristen DiCerbo schreibt [9], dass Informationen über den zu Grunde liegenden Prozess wichtig sind. Es gilt herauszufinden, ob der Vorgang der zu den Extremwerten führt, jener ist der erforscht wird oder ob es sich um das Einwirken eines anderen Prozesses handelt. Bei Spielen können viele verschiedene Dinge ungewollt für Ausreißer verantwortlich sein: Die Anleitung wurde nicht verstanden, die Umgebung wird erforschen, es werden eigene Ziele verfolgen, der Aufgabe wird keine Aufmerksamkeit geschenkt oder es kann sich um geräte- oder datenspezifische Fehler handeln.

This context and background can help us determine how to deal with outliers. Do we keep them or pull them out? The fundamental question to ask is: Do we know something about these observations that suggest they come from a different process than the process we are seeking to understand? In games, numerous unintended processes could lead to outlying values: failure to understand instructions, exploring the environment, following their own goals, failure to pay attention to the task, or equipment or data failures.

Bei den Testdaten entstanden Ausreißer durch Fehler des AR-*Trackings*, welche erst während der Analyse bekannt wurden (siehe Abschnitt 5.5.1). Diese somit verfälschten Daten wurden deshalb aus der Tabelle in *RStudio* gelöscht. Die originalen CSV-Dateien blieben unverändert, um die Originale in einem reinen Zustand zu lassen. Eine weitere Art von Ausreißern waren Spieler, die testeten wie gut das *Tracking*-Verhalten ist und versuchten die Grenzen der AR-Technologie auszureizen. Spieler versuchten so weit wie möglich vom *Tracker* weg zu gehen und trotzdem noch Spielen zu können. Es war auch der Fall, dass Spieler so nahe wie möglich zu dem *Tracker* gingen, um zu sehen wie sich das Spiel dabei verhält oder einfach nur um die 3D-Modelle aus der Nähe zu betrachten. Diese Daten wurden jedoch weiterverarbeitet und somit nicht aus der Tabelle gelöscht. Der Grund dafür ist, dass diese Daten genau so Teil des Spiels sind und vielleicht gute von schlechten Spielern trennen. Dieses Verhalten war oftmals nur bei dem ersten Durchgang der Fall, da es für viele Tester die erste Gelegenheit war diese neue Technologie auszuprobieren. So könnte auch darauf geschlossen werden, dass sich schlechtere Spieler leichter durch solche Dinge ablenken ließen und geübte Spieler, denen ein schnelles Durchspielen des Levels wichtiger ist, nicht.

5.4 Vorgehensweise

Wie in Kapitel 3 beschrieben, ist noch kein Zusammenhang zwischen den Daten bekannt. Deswegen wird eine Vielzahl an Methoden verwendet, um sich ein Bild von den verfügbaren Daten zu verschaffen. Ziel dieser Exploration ist es, gewisse Variablen zu verwenden, um diese letzten Endes zu kombinieren. Mit dem Ergebnis wird für jeden Spieler ein skalarer Wert berechnet, um zu sehen ob dieser in einer Relation zu dem zuvor bereits festgelegten *Skill* steht. Genauer gesagt soll aus dem Verhalten des Spielers (eine Kombination aus unterschiedlichen unabhängigen Variablen) auf dessen *Skill* (die abhängige Variable) geschlossen werden können (siehe Abschnitt 3.7.3).

5.5 Analyse der Spieldurchläufe

Um herauszufinden, bei welchen Daten es überhaupt möglich wäre eine Korrelation zu finden, werden sämtliche Dateien zunächst einzeln und für sich stehend betrachtet und analysiert. Eine Datei entspricht einem gespielten und beendeten Level. Zu Beginn werden einfache statistische Maßzahlen für jede Spieldatei ausgegeben. Es wird versucht herauszufinden, ob es gewisse Muster gibt, die immer wieder auftreten. Sollten diese Muster auftreten werden andere Variablen betrachtet, um zu sehen, ob andere Werte ebenfalls zusammen mit diesem Verhalten in Erscheinung treten. Es gab eine Liste der Spieldaten, wobei jedes Element einer Datei entsprach. Jeder Eintrag hat bereits einen zugeordneten *Skill*. Dieser wurde dazu genutzt um die Liste danach zu ordnen. Dies hat den Vorteil, dass Visualisierungen, die für sämtliche Listeneinträge erstellt werden, in dieser Reihenfolge durchgeblättert werden können. Auf diese Weise lassen sich potentielle Kandidaten für eine Korrelation zu dem *Skill* leichter erkennen. Wurde ein potentieller Kandidat gefunden wird dieser Wert, für jeden Spieler, auf einem Plot eingezeichnet, um eventuell auch dort Muster zu erkennen (siehe Abschnitt 5.7).

5.5.1 Tabelle

Aufgrund der enormen Menge an Daten konnte der Ansatz mit der tabellarischen Auflistung (siehe Abschnitt 3.2) nicht wirklich angewandt werden. Eine Einsicht in diese konnte jedoch helfen bestimmte Auffälligkeiten zu klären. Während der ersten Analysen, mit visuellen Ausgaben, fiel auf, dass Positions- und Rotationswerte der Kamera sporadisch sehr stark umher sprangen. Es fiel schwer lediglich aus der grafischen Ausgabe Schlüsse zu ziehen. Ein Blick in die Tabelle der Daten zeigte jedoch recht schnell, dass der Grund im Zusammenhang mit dem *Tracking* der AR-Kamera stand. Die Tracking-Software versucht stets aus allen gegebenen Feature-Punkten das bestmögliche Ergebnis zu berechnen. Mit einer Abnahme der erkannten Fea-

tures wurde das Ergebnis ungenauer. Sollte der *Tracker* komplett aus dem Sichtfeld verloren werden so wird mit den zuletzt vorhandenen Daten noch versucht die virtuelle Welt darzustellen. Dadurch kam es oft zu Ergebnissen die in keinsten Weise verwertbar waren. Somit war klar, dass diese Daten bereinigt werden mussten (siehe Abschnitt 5.3.3).

5.5.2 Standardabweichung und Varianz

Es gibt Werte, welche sich nicht ohne Weiteres vergleichen lassen. Bei Variablen, die sehr stark von ihrem Anfangswert abhängen und sich im Laufe der Zeit nur leicht verändern, ist dies der Fall. Um diese Werte vergleichbar zu machen, kann beispielsweise auf die Streuung zurückgegriffen werden. In *Vancouver Maneuver* fällt es beispielsweise schwer die Kameraposition der Spiele zu vergleichen, da die Spieler von jeder beliebigen Position aus starten und das Spielfeld betrachten können. Sie können sich die gesamte Zeit über frei in der realen Welt bewegen. In diesem Fall bietet sich die Varianz an, da sich die Streuung sehr wohl und sehr gut vergleichen lässt. Bei vielen Spielern ist es der Fall, dass sie sich zu Beginn am meisten bewegen. Im Verlauf des Spiels ändern sie ihre Position nur noch relativ selten und bewegen sich auch kaum von ihrem Startpunkt weg. Beim betrachten der Rotation zeichnet sich ein ähnliches Bild ab. Der Grund dafür liegt in der Natur des Spiels. Da Werte nur verwertbar aufgezeichnet werden, wenn ein *Tracker* erkannt wird, muss die Rotation somit immer zum Spielfeld hin ausgerichtet sein. Daher ergibt sich, dass sich die Rotation nur in einem bestimmten Bereich befinden kann solange die Position stagniert.

Spannweite

Es kann sein, dass bei Werten der Schwankungsbereich mehr Aussagekraft hat, als z. B. die Varianz. Bei allen Testdaten, die mittels Varianz analysiert wurden, wurde ebenso die Spannweite betrachtet, um zu sehen, welche Methode bessere Ergebnisse liefert. Es war der Fall, dass sich bei gewissen Werten bei der Varianz ein Bild abzeichnete und dieses, mittels Betrachtung der Bandbreite, noch erheblich stärker zum Vorschein kam. Diese führt in der Gesamtanalyse (siehe Abschnitt 5.7) letzten Endes zu einer Cluster-Bildung. Bei dem betroffenen Wert handelt es sich um den Rotationswert der Kamera. Der Grund für diese Cluster lag jedoch bei einem Fehler (siehe Abschnitt 5.8.2 und Abschnitt 3.8.3).

5.5.3 Streudiagramm

Das Streudiagramm hilft sehr gut um Zusammenhänge zwischen zwei Variablen zu erkennen (siehe Abschnitt 3.4.1). Eine besonders gute Hilfestellung leistet diese Visualisierung bei der Verwendung der Zeit auf der x -Achse. Die Grafik stellt somit einen Funktionsgraphen einer Zeit/Wert-Funktionen

dar. Es wurde ein klares Bild vermittelt, wie sich ein bestimmter Wert über die gesamte Spieldauer hinweg verhält. Hierbei kann die Visualisierung der Werte als Punkte oder als Linie erfolgen. Auf diese Weise können Schwankungsmuster leicht erkannt werden. Der Schwankungsbereich lässt sich gut eingrenzen und Ausreißer fallen ebenso stark auf. Ein merkwürdiges Verhalten zeigte sich z.B. bei Rotationsdaten (siehe Abschnitt 5.8.2) oder bei Positionsdaten wenn der *Tracker* nicht gefunden wurde (siehe Abschnitte 5.5.1 und 5.3.3).

5.5.4 3D Diagramme

Wie in Abschnitt 3.8.3 beschrieben, fällt es schwer Rotationsdaten zu vergleichen. Ebenso ist es schwer diese gut zu visualisieren. Somit wurde für die grafische Darstellung ein interaktives dreidimensionales Streudiagramm verwendet. Ein Null-Rotations-Vektor wird definiert in der Form

$$\vec{v} = (0, 0, 1)$$

und stellt den Vorwärtsvektor der Kamera, in nicht rotiertem Zustand, dar. Der Vektor wurde um die Quaternion gedreht, um die Blickrichtung der Kamera in einem dreidimensionalen kartesischen Koordinatensystem darstellen zu können. Der Vektor wurde anschließend als Punkt in dem 3D Diagramm gezeichnet. Dies wurde über die gesamte Datenmenge gemacht und somit ergab sich eine dreidimensionale Punktwolke, die die Blickrichtung der Kamera widerspiegelte (siehe Abb. 3.5).

5.6 Ansätze und Ideen

Nach der Analyse der Daten und den ersten Ergebnissen gab es bereits Anwärter für potentiell erfolgreiche Methoden. In Kombination mit Beobachtungen der Testspieler ergaben sich ein paar Ansätze, die weiter verfolgt wurden.

5.6.1 Nähe zum Spielfeld als Indikator für gute Spieler

Bei der Beobachtung von Spielern fiel auf, dass sich die Verhaltensweise mit dem Umgang des Smartphones nach mehrmaligem Spielen ein wenig änderte. Die Spieler begannen das Handy mehr zu bewegen und sich damit dem *Tracker* weiter zu nähern. Dadurch konnten genauere Arbeitsschritte präziser und schneller ausgeführt werden. Unter der Prämisse, dass sich Spieler durch wiederholtes Spielen verbessern, kann davon ausgegangen werden, dass ein Zusammenhang zwischen der Nähe zum *Tracker* und dem *Skill* vorhanden ist. Einerseits wäre es möglich lediglich die Distanz zum Zentrum zu messen, andererseits wird bei der Analyse klar, dass hierbei noch andere Faktoren eine Rolle spielen. Durch die Analyse der Histogramme (siehe

Abb. 5.2) wurde ersichtlich, dass die Anzahl der Objekte, die gleichzeitig zu sehen waren, mehr Aussagekraft hatte. Die Anzahl der sichtbaren Objekte variiert stärker bei guten Spielern. Dies muss der Fall sein, wenn sich der Spieler näher zum Spielfeld bewegt, da das Sichtfeld nicht mehr alle Objekte enthalten kann. Ein weiterer Aspekt, der hierbei ins Gewicht fällt ist, dass gute Spieler die Räume nicht weiträumig verteilen, sondern diese immer dicht zusammen halten, damit die Spielfiguren keine weiten Wege zurücklegen müssen und sich der Spieler selbst nicht weit bewegen muss. Somit muss eine niedrige Anzahl an gleichzeitig sichtbaren Objekten, bei guten Spielern, mit der Distanz zum Spielfeld zu tun haben. Werden diese beiden Annahmen kombiniert, lässt sich dies auf die Anzahl der sichtbaren Objekte zu jedem gegebenen Zeitpunkt zurückführend. Nähert sich ein Spieler dem *Tracker*, wie die Annahme eines guten Spielers das macht, engt dieser sein Sichtfeld ein und es sind weniger Objekte gleichzeitig sichtbar. Es wurde mit einem Vektor weitergerechnet, dessen Länge der Anzahl der aufgezeichneten Frames entsprach. Jedes Element stellt die Anzahl der aktuell sichtbaren Objekte dar. Es wurden unterschiedliche Werte berechnet, deren Grundlage diese Zahlenmenge war. Letztenendes war der vielversprechendste Anwärter die Varianz dieser Werte.

5.6.2 Übermäßige Rotation als Indikator für schlechte Spieler

Ein anderer Ansatz, der bei Beobachtungen von den Spielern aufgekommen ist, ist jener, dass Spieler die öfter gespielt haben oder schon mit dem Projekt vertraut waren, bei dem genaueren Zielen mit dem Smartphone näher an das Spielfeld herantraten, anstelle aus der Ferne zu versuchen durch Rotation ihr Ziel anzuvisieren. Das Handy wurde nur sehr begrenzt rotiert und stattdessen wurde die Position des Handys verändert. Die Handhaltung wurde nicht verändert. Den Großteil der Bewegungen übernahmen die Arme, sodass sich die Position des Smartphones stets parallel zu den vorigen Positionen befand. Im Gegensatz dazu bewegten sich Spieler, die zum ersten Mal spielten, während spielrelevanter Phasen so gut wie gar nicht. Die Position der Smartphones blieb starr und das Zielen fand lediglich über die Rotation des Handys statt. Je öfter diese Personen jedoch spielen, desto mehr fingen sie an sich zu bewegen und Rotationen auf ein notwendiges Minimum zu beschränken. Da auch bereits bei dem vorigen Ansatz, lediglich die Position zu benutzen, kein gutes Ergebnis lieferte, wurde hier ebenfalls eine zweite Komponente berücksichtigt. Daher wurde die Rotationsdaten verwendet um diese im weiteren Verlauf zu analysieren. Es wurde mit einer Liste weitergearbeitet, die für jeden Zeitpunkt im Spiel die Rotation der Kamera, als Quaternion und somit mit vier Elementen, enthält. Die interessantesten Resultate wurden erzielt indem die Varianz und die Spannweite dieser Zahlen berechnet wurden.

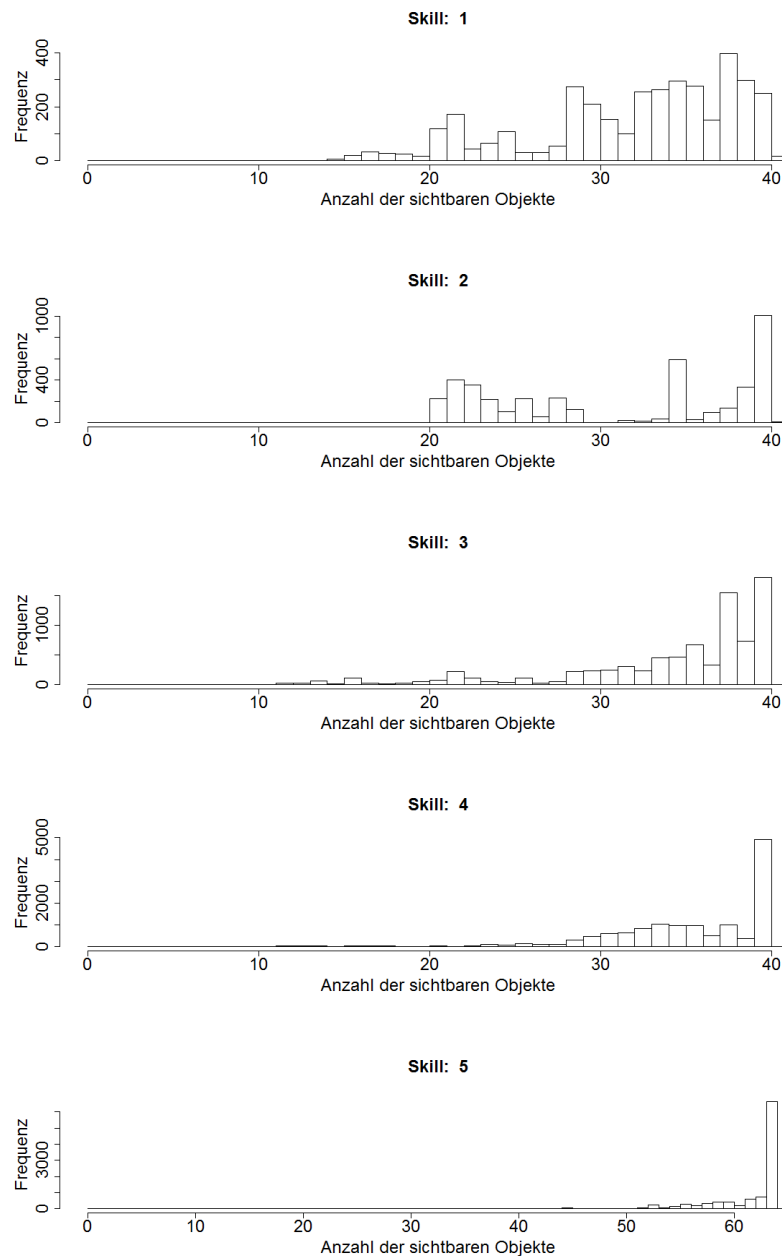


Abbildung 5.2: Jede Grafik bezieht sich auf Daten von Spielern unterschiedlichen *Skills*. Die Histogramme zeigen die Häufigkeit, der Anzahl der gleichzeitig sichtbaren Objekte, über die jeweilige Spieldauer hinweg. Diese Histogramme weisen die Tendenz auf, dass sie breiter gefächert sind, je besser der Spieler ist. Das bedeutet, dass sich bei guten Spielern des Öffteren weniger Objekte im Sichtbereich befinden.

5.7 Analyse der gesamten Daten

Die eben beschriebenen Methoden liefern einen skalaren Wert pro Spiel, der nun in Verbindung mit dem *Skill* gebracht werden soll. Diese Information wird für sämtliche Spiele auf einem Streudiagramm aufgetragen. Auf der x -Achse wird der *Skill* angegeben und die y -Achse entspricht dem Wert der Spiele. Gibt es eine Korrelation zwischen den Daten und dem *Skill*, so muss dieses Verhalten ebenfalls in der Abbildung in Erscheinung treten. Die grafische Ausgabe hilft hierbei Tendenzen und Muster zu erkennen. Gleichzeitig wird eine lineare Regressionslinie auf dem Diagramm eingezeichnet. Sollte eine gewisse Neigung oder Ausrichtung der Punktwolke erkennbar sein, so muss dieses Verhalten auch durch die Regressionslinie widerspiegelt werden.

5.7.1 Ergebniss

Die Exploration führte zu zwei potentiellen Kandidaten. Einerseits handelte es sich dabei um die Varianz der Anzahl der Objekte im Sichtfeld des Spielers über die Spieldauer hinweg (siehe Abschnitt 5.6.1). Es war eine leichte Tendenz über die *Skills* hinweg ersichtlich. Andererseits wurde bei der Betrachtung der Varianz bzw. der Spannweite der Rotation der Kamera eine Cluster-Bildung festgestellt. Es wurde vermutet, dass bei den Datensätzen innerhalb eines Clusters, Gemeinsamkeiten auftreten, die bei der weiteren Analyse helfen können. In beiden Fällen stellte sich jedoch heraus, dass es sich dabei um Fehler handelte (siehe Abschnitte 5.8.3 und 5.8.2).

5.8 Aufgetretene Probleme und Fehlerquellen

5.8.1 Balkenanzahl von Histogrammen

Bei Histogrammen ist es wichtig, die Anzahl der Balken richtig festzulegen. Die automatische Berechnung führte in manchen Fällen Werte zusammen und verfälschte das Ergebnis. Konkret aufgetreten ist das Problem bei der Verwendung der Anzahl der Spielobjekte im Sichtbereich des Spielers. Hierbei ist es über den Großteil der Spiele hinweg üblich, dass die letzten Balken erheblich größer sind als die anderen. Es ist auch oft der Fall, dass lediglich die letzten Balken Information enthielte. Diese wurde bei automatischer Balkenanzahl jedoch oft zusammengeführt und dies verursachte falsche Ergebnisse (siehe Abb. 5.3). In dem beschriebenen Fall handelte es sich jedoch um diskrete Zahlen, und somit wurde die maximale Anzahl der Levelobjekte als Balkenstufenanzahl verwendet, wodurch dieses Problem gelöst war.

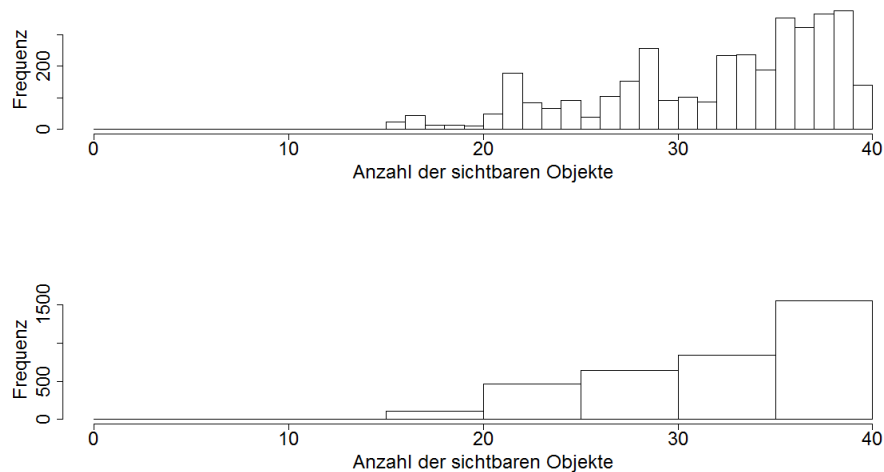


Abbildung 5.3: Das Bild zeigt zwei Histogramme des selben Datensatzes. Sie unterscheiden sich lediglich in der Balkenanzahl. Bei der oberen Version wurde die Balkenanzahl den Daten (Anzahl der Objekte im Level) entsprechend gesetzt. Bei dem unteren Histogramm wurde diese Anzahl automatisch angepasst. Wie zu erkennen ist, bleibt das generelle Bild zwar erhalten, aber kleinere und vielleicht auch essentielle Details verschwinden. Vereinzelt auftretende hohe Frequenzen gehen dadurch verloren, indem diese mit anderen Werte zusammen genommen werden. Dadurch kann es vorkommen, dass eine wesentliche Charakteristik verschwindet.

5.8.2 Vergleich von Rotationen

Bei dem Vergleich der Varianz und der Spannweite der Kamerarotation kam es zur Cluster-Bildung. Zunächst wurde von Spielereigenheiten ausgegangen, da immer die selben Spieler betroffen waren. Wie sich jedoch bei weiterer Analyse der Rotationsdaten zeigte, war der Grund dafür ein Fehler, der bei der Berechnung von Werten von kreisförmigen Daten auftrat. Das Problem zeigte sich, als versucht wurde die einzelnen Elemente des Quaternions auf einem Streudiagramm abzubilden. Da es sich um Kreisförmige Daten handelt, können diese nicht ohne weiteres verglichen werden (siehe Abschnitt 3.8.3). Bei der Darstellung der Werte der einzelnen Elemente des Quaternions (siehe Abb. 5.4), wird ersichtlich wieso sich diese Cluster bilden. Die Grafik stellt den x -Wert des Rotations-Quaternions dar. Dieser Wert schwankt nur sehr leicht. Da die Schwankungen jedoch genau an der Grenze auftreten wo aus einem positiven Wert ein negativer Wert wird (dies tritt auf wenn der Wert 0.5 überschritten bzw. -0.5 unterschritten wird) springt der Wert sofort über die gesamte Spannweite. Dies erklärt somit

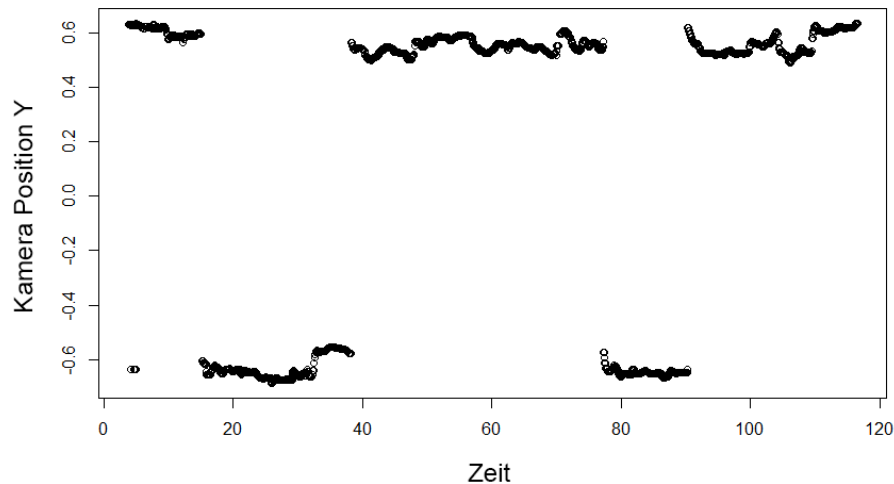


Abbildung 5.4: Das Bild zeigt das Verhalten des y -Wertes des Rotationsquaternions der Kamera über den gesamten Verlauf des Spiels. Es ist erkennbar, dass sich der Wert stets nur leicht verändert, zeitweise jedoch über die gesamte Spannweite springt.

auch, wieso diese Cluster bei der Verwendung der Spannweite (siehe Abb. 5.5) ausgeprägter sind, als dies bei der Verwendung der Varianz der Fall ist (siehe Abb. 5.6). Bei der Spannweite, ist der maximale Wert erreicht, sobald einmal die Grenze überschritten wurde, die das Vorzeichen des Wertes umkehrt. Bei der Varianz hingegen wird die Cluster-Bildung deutlicher je öfter diese Grenze überschritten wird. Das Problem wurde wie in Abschnitt 3.8.3 beschrieben, behoben. Anstelle von Winkeln wurde die Rotationsinformation des Quaternion in ein kartesisches Koordinaten System gebracht, um die Koordinaten dadurch vergleichbar zu machen. Auf dem korrigierten Diagramm waren keine Cluster mehr erkennbar (siehe Abb. 5.7).

5.8.3 Absolute und relative Werte

In dem Spiel gab es drei verschiedene Level. Da es für die Analyse jedoch unerheblich sein sollte, um welchen Level es sich handelt, wurden allesamt mit den selben Methoden analysiert. Das Problem trat auf, als die Berechnung eines Wertes die Anzahl der Levelobjekte verwendete. Diese Anzahl war aufgrund der Struktur und des Aufbaus sämtlicher Level unterschiedlich. Dies führte dazu, dass die Ergebnisse bei einem Vergleich, eine scheinbare Tendenz und somit falsche Resultate vorbrachten (siehe Abb. 5.8). Dieses Problem wurde dadurch gelöst, dass diese Werte nun stets relativ im Ver-

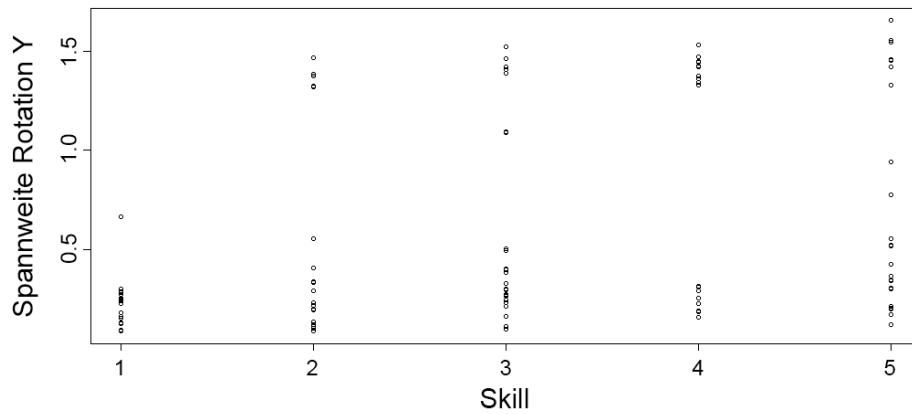


Abbildung 5.5: Das Streudiagramm zeigt die Spannweiten der y -Achse des Rotationsquaternions sämtlicher Spiele im Verhältnis zum *Skill*. Es werden Cluster erkennbar.

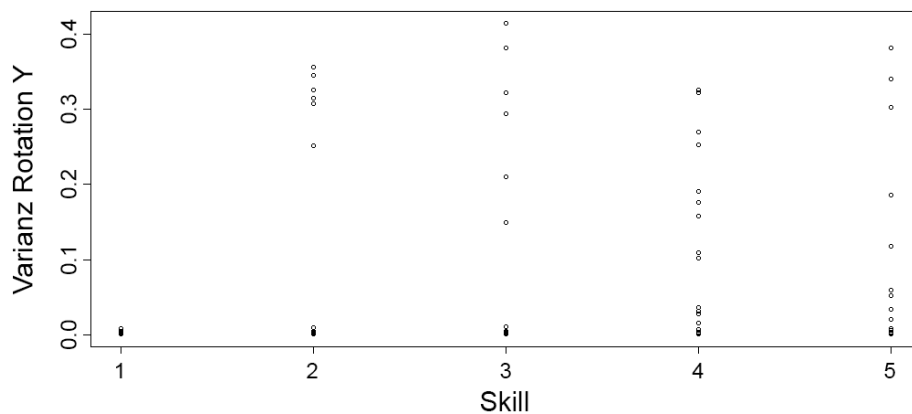


Abbildung 5.6: Das Streudiagramm stellt den Rotationswert der y -Achse dem *Skill* gegenüber. Jeder Punkt entspricht der Varianz der y -Achse der Rotation eines Spieles über die Spieldauer hinweg. Im Gegensatz zu der Verwendung der Spannweite, ist die Cluster-Bildung jedoch nicht allzu stark ausgeprägt (vergleiche mit Abb. 5.5).

hältnis zur maximalen Anzahl angegeben wurde (siehe Abb. 5.9).

5.8.4 Anzahl der Testdaten

Der letzte Punkt bezieht sich nicht nur auf diese spezielle Implementierung, sondern richtet sich an alle Datenanalysen. Die Anzahl der Testdaten ist

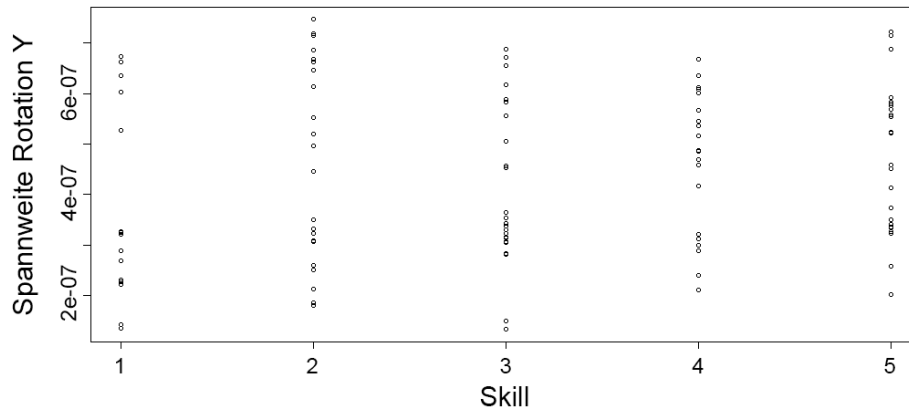


Abbildung 5.7: Das Diagramm stellt die Spannweite der Rotation dem *Skill* gegenüber. Sämtliche Fehler wurden behoben indem die Spannweite in einem kartesischen Koordinatensystem berechnet wird. Auf dem Streudiagramm des korrigierten Ergebnisses sind keine Cluster mehr vorhanden (vergleiche mit Abb. 5.5 und 5.6).

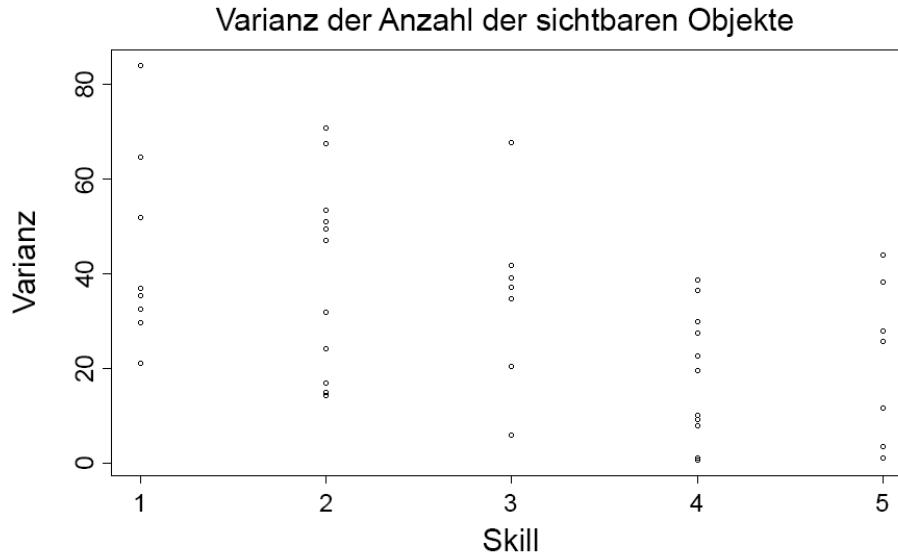


Abbildung 5.8: Dieses Streudiagramm zeigt einen potentiellen Wert für jedes Spiel entsprechend zu dem dazugehörigen *Skill*. Der Wert entspricht der Varianz der Anzahl der Objekte im Sichtfeld über die Spieldauer hinweg. Es ist eine leicht fallende Tendenz zu erkennen. Es wurde jedoch nicht berücksichtigt, dass es in den verschiedenen Levels unterschiedlich viele Objekte gibt.

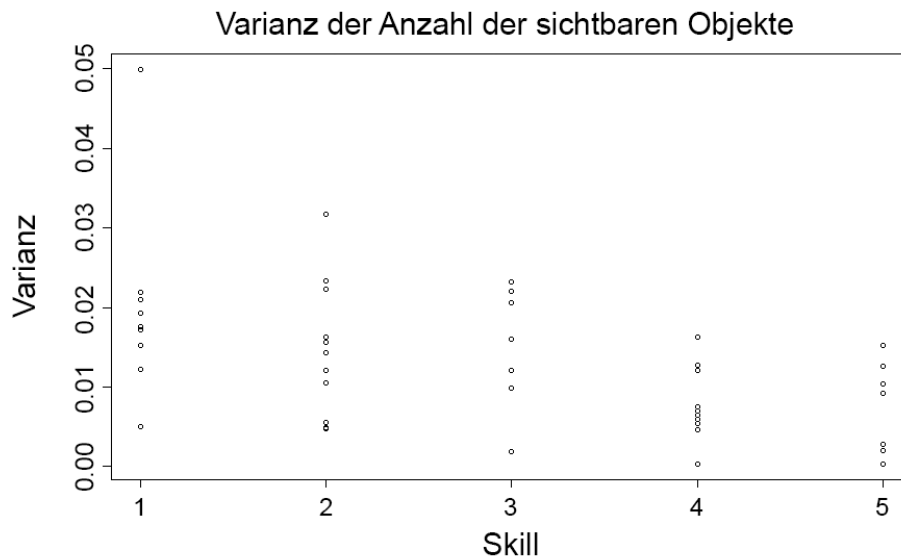


Abbildung 5.9: Die Grafik zeigt die Varianz der Anzahl der sichtbaren Objekte über die Spieldauer hinweg an und zeigt diese zugehörig zu dem entsprechenden Skill an. Die ohnehin lediglich nur schwach vorhandene fallende Tendenz wurde durch die Korrektur des Fehlers sehr stark abgeschwächt. Die Objektanzahl wird nun in Prozent angegeben (vergleiche Abb. 5.8).

maßgeblich für eine gute Analyse. Mit der explorativen Analyse wurde bereits begonnen als die ersten Daten verfügbar waren. Dies führte dazu, dass eine Korrelation, aufgrund einer leichten Tendenz der Datenpunkte, vermutet wurde (siehe Abb. 5.8). Im Laufe der Zeit wurden immer mehr Testdaten durch Spieler generiert. Dies führte dazu, dass die zuvor erwähnte Tendenz immer undeutlicher wurde und letzten Endes verschwand (siehe Abb. 5.10). Der Grund dafür war, dass es zu wenig unterschiedliche Daten gab und die Resultate durch die wenigen Spieler eine zufällig aufgetretene Tendenz aufwiesen. Zu wenig vorhanden Daten können außerdem zu einem weiteren Problem führen. Es könnte sein, dass die Methoden und Daten eine Korrelation aufweisen würden, aber aufgrund zu weniger Daten wird diese nicht ersichtlich, da sich diese Stichprobe zu stark verteilt.

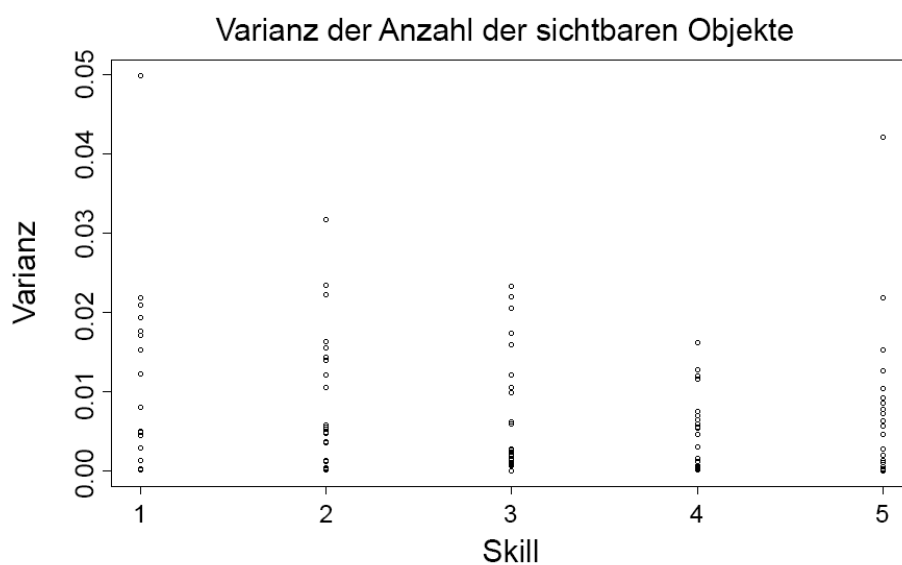


Abbildung 5.10: Im Streudiagramm wird die Varianz der Anzahl der sichtbaren Objekte über den Spielverlauf hinweg, für jede Spieldatei, dem *Skill* gegenüber gestellt, so wie dies bereits in Abbildung 5.9 der Fall ist. Jegliche Tendenz eines scheinbaren Zusammenhangs ist mit einer erhöhten Anzahl an Testdaten komplett verschwunden. Die Werte erstrecken sich über sämtliche *Skill*-Level hinweg über die gleichen Bereiche, mit einigen wenigen Ausreißern.

Kapitel 6

Schlussbemerkungen

6.1 Zusammenfassung

Es wurde gezeigt worum es bei explorativer Datenanalyse geht und wie diese anhand eines Spiels angewandt wird. Anfänglich gab es ein paar potentielle Kandidaten die auf eine Korrelation zu dem *Skill* hinwiesen, da diese erkennbare Muster und leichte Tendenzen aufwiesen. Da sich diese Annahme jedoch als falsch herausstellt, wurde zunächst vermutet, dass es sich dabei um Spielereigenheiten handelte. Nach weiteren Analysen wurde jedoch herausgefunden, dass diese Muster aufgrund von Fehlern aufgetreten sind.

6.2 Fazit

Die Analyse endete mit dem Ergebnis, dass sich der *Skill* nicht aus dem Umgang mit dem Smartphone erkennen lässt. Einige Spieler bemerkten, dass das Spielen von *Vancouver Maneuver* auf Dauer zu einer anstrengenden Handhaltung führt. Es wurde ebenfalls angemerkt, dass das Spielen im Sitzen nicht gut möglich ist, da Bewegung ein Teil des Spiels ist, aber Stehen mit der Zeit anstrengend ist. Dies könnte dazu führen, dass sich die Spieler im Verlaufe des Spiels immer weniger bewegen und somit die Daten und dadurch das Ergebnis dieser Analyse stark beeinflussen. Es fiel auf, dass Spieler in den ersten Durchgängen, als sie noch nicht allzu vertraut mit AR waren, sich nur sehr minimal bewegten. Sie hielten das Smartphone wie sie es von einem Controller gewohnt waren und bewegten es nur sehr wenig. Sie versuchten immer die gesamte Spielwelt im Sichtfeld zu behalten und rotieren das Handy beim Spielen nur langsam. Einerseits aufgrund der noch nicht vorhandenen Erfahrung mit AR und andererseits um präziser zu sein. Mit mehr Erfahrung begannen die Spieler sich selbst als Teil der Spielerfahrung zu sehen und begannen sich mit dem Smartphone über den Tisch und den *Tracker* zu beugen um näher am Geschehen zu sein. Da sich das Spielverhalten in relativ kurzer Zeit bereits änderte, ist es vielleicht auch möglich,

dass sich diese Verhaltensmuster durch einen vermehrten Einsatz von AR-Spielen, noch weiter prägen lassen. Vielleicht werden dadurch Daten die aus AR-Applikationen gewonnen werden in Zukunft aussagekräftiger, wenn die Benutzer mehr mit der Materie vertraut sind. Möglicherweise lässt sich die gescheiterte Analyse jedoch auch auf die Art des Spiel zurückführen. Der Spieler stellt eine, sich über dem Spielfeld schwebende Entität dar, die nicht wirklich in der Spielwelt vorhanden ist. Vielleicht ist der Versuch diese Daten in Verbindung mit dem *Skill* zu bringen unangebracht.

Die Verwertbarkeit der Resultate, ändert jedoch nur wenig an der Herangehensweise der Datenanalyse. Ungeachtet dessen, ob die Exploration Ergebnisse hervorbringt oder nicht, muss gleich vorgegangen werden. Lediglich das Endresultat ist ein anderes. Bei den Analysen und vor allem bei den Fehlern die auftraten, wurden einige Erfahrungen gesammelt, die in zukünftige Analysen einfließen werden.

6.3 Ausblick

Der Fehlschlag der Analyse, bedeutet jedoch nicht, dass es unmöglich ist, Ergebnisse in diesem Bereich zu erzielen. Die Ergebnisse könnten eventuell verbessert werden, wenn die AR-Daten mit spielrelevanten Elementen kombiniert werden. Möglicherweise liefern andere Anwendungsbereiche unterschiedliche Ergebnisse. Die Forschungen in diesem Bereich könnten sich vorerst an besser zugeschnitten Videospiel-Genre richten. Spiele in denen der Spieler ein Teil der Welt ist, wie dies z. B. bei First-Person-Spielen der Fall ist. Es ist möglich, dass sich die Aussagekraft solcher Daten ändert, sobald sich die breite Masse an AR-Technologien gewöhnt hat. Da es sich bei AR um einen wachsenden Markt handelt, ist davon auszugehen, dass in Zukunft noch viel Forschung in diese Richtung betrieben wird. Mit neuen AR-Applikationen ist auch damit zu rechnen, dass neuartige Daten entstehen und als Folge dessen neue Methoden entwickelt werden, um diese zu verarbeiten und zu analysieren.

Anhang A

Inhalt der CD-ROM/DVD

A.1 PDF-Dateien

Pfad: /

Winter_Christian_2016.pdf Masterarbeit

A.2 Bilder

Pfad: /images

*.png Original Rasterbilder

A.3 Literatur

Pfad: /literatur

*.pdf Literaturquellen

A.4 Online-Quellen

Pfad: /online

*.html Index Dateien der Online-Quellen

/* Daten der Online-Quellen

A.5 Data

Pfad: /data

VancouverManeuver.apk Original .apk-Datei zur Installation
von *Vancouver Maneuver* auf einem
Android-Smartphone

Replays.zip	ZIP-Archiv der binären <i>Replay</i> -Dateien
CSVCreator.exe	Ausführbares Programm zum Konvertieren der <i>Replay</i> -Dateien
CSVCreator_Data . . .	Notwendige Dateien zum Ausführen von CSVCreator.exe
CSV.zip	ZIP-Archiv der CSV-Dateien aller konvertierten <i>Replay</i> -Dateien

A.6 R-Skripte

Pfad: /scripts

*.r	Original R-Skripte
---------------	--------------------

Quellenverzeichnis

Literatur

- [1] Francis John Anscombe. „Graphs in statistical analysis“. *The American Statistician* 27.1 (1973), S. 17–21 (siehe S. 15).
- [2] Ben Bolker. *Exploratory data analysis and graphics*. Princeton University Press, 2007 (siehe S. 19, 28, 34).
- [3] Andrew Gelman und Jennifer Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press, 2006 (siehe S. 23).
- [4] Berthold K. P. Horn. „Closed-form solution of absolute orientation using unit quaternions“. *Journal of the Optical Society of America A* 4.4 (1987), S. 629–642 (siehe S. 23, 24).
- [5] Moritz Kertesz. „Prozedurale Levelgenerierung für rasterbasierte Mehrspieler-Rätselspiele“. Masterarbeit. Hagenberg, Austria: University of Applied Sciences Upper Austria, Interactive Media, Okt. 2016 (siehe S. 3).
- [6] John W. Tukey. „The Future of Data Analysis“. *Annals of Mathematical Statistics* 33.1 (März 1962), S. 1–67 (siehe S. 12, 14).

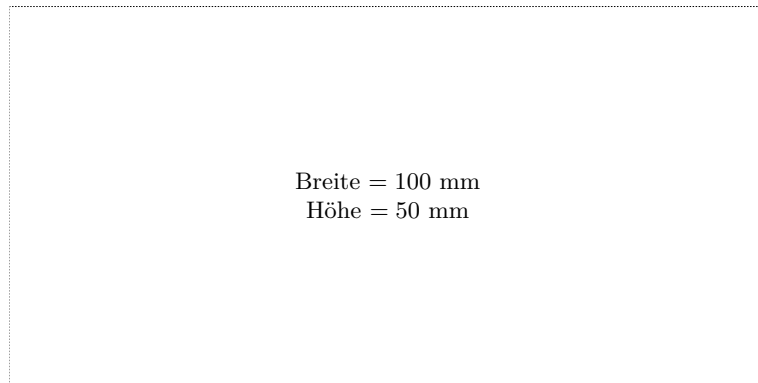
Online-Quellen

- [7] Ernest Adams. *The Designer’s Notebook: Difficulty Modes and Dynamic Difficulty Adjustment*. 2016. URL: http://www.gamasutra.com/view/feature/132061/the_designers_notebook_.php?print=1 (siehe S. 2).
- [8] Robbie Allan. *Pokemon Go Statistics*. 2016. URL: <https://www.surveymonkey.com/business/intelligence/pokemon-go-usage-statistics/> (siehe S. 2).

- [9] Kristen DiCerbo. *Exploratory Data Analysis and Game Data: Resistance*. 2014. URL: <http://researchnetwork.pearson.com/digital-data-analytics-and-adaptive-learning/exploratory-data-analysis-game-data-resistance> (siehe S. 36).
- [10] Darrell Etherington. *Pokemon Go adds 9B to Nintendos value*. 2016. URL: <https://techcrunch.com/2016/07/11/pokemon-go-adds-9b-to-nintendos-value-global-rollout-continues-this-week/> (siehe S. 2).
- [11] The R Foundation. *R*. 2016. URL: <https://www.r-project.org/> (siehe S. 30).
- [12] EPIC GAMES. *VR - Unreal Engine*. 2016. URL: <https://www.unrealengine.com/vr> (siehe S. 2).
- [13] Google. *Google Analytics*. 2016. URL: <https://www.google.com/analytics/> (siehe S. 1).
- [14] Robert Kosara. *Anscombes Quartett*. 2011. URL: <https://eagereyes.org/media/2011/anscombe.png> (siehe S. 16).
- [15] Parametric Technology Corporation (PTC). *PTC*. 2016. URL: <http://www.ptc.com/> (siehe S. 4).
- [16] Inc. (QTI) Qualcomm Technologies. *Qualcomm*. 2016. URL: <https://www.qualcomm.com/> (siehe S. 4).
- [17] Wolf Riepl. *Scheinkorrelation vs. intervenierende Variable*. 2012. URL: <http://statistik-dresden.de/archives/1089> (siehe S. 22).
- [18] RStudio. *R-Studio*. 2016. URL: <https://www.rstudio.com/> (siehe S. 30).
- [19] Ralf Sachs. *DIN Formate*. 2016. URL: <http://www.din-formate.de/reihe-a-din-groessen-mm-pixel-dpi.html> (siehe S. 5).
- [20] Karline Soetaert. *plot3D*. 2016. URL: <https://cran.r-project.org/web/packages/plot3D/index.html> (siehe S. 33).
- [21] Unity Technologies. *Unity*. 2016. URL: <https://unity3d.com/> (siehe S. 4).
- [22] Unity Technologies. *Unity Analytics*. 2016. URL: <https://unity3d.com/services/analytics> (siehe S. 1).
- [23] Unity Technologies. *VR - Unity*. 2016. URL: <https://docs.unity3d.com/Manual/VROverview.html> (siehe S. 2).
- [24] Vuforia. *Develop.Vuforia.com/Download/SDK*. 2016. URL: <https://developer.vuforia.com/downloads/sdk> (siehe S. 4).
- [25] Vuforia. *Vuforia*. 2016. URL: <http://www.vuforia.com/> (siehe S. 4).
- [26] Hadley Wickham. *ggplot2*. 2013. URL: <http://ggplot2.org/> (siehe S. 33).

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —