

PokerTool

Entwicklung und Implementierung einer AR-Android-Anwendung für Wahrscheinlichkeitsberechnungen bei Texas Holdem Poker

Masterarbeit

im Fachgebiet Computervisualistik



UNIVERSITÄT
KOBLENZ · LANDAU



COMPUTERVISUALISTIK

vorgelegt von: Christoph Thul

Matrikelnummer: 207200109

Erstgutachter: Prof. Dr. Stefan Müller

Zweitgutachter: M.Sc. Gerrit Lochmann

© 2013

Erklärung

Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbständig verfasst wurde und ich keine anderen als die angegebenen Hilfsmittel - insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen - benutzt habe und die Arbeit von mir vorher nicht in einem anderen Prüfungsverfahren eingereicht wurde. Die eingereichte schriftliche Fassung entspricht der auf dem elektronischen Speichermedium (CD-Rom).

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

.....
(Ort, Datum)

.....
(Unterschrift)



Aufgabenstellung für die Masterarbeit
Christoph Thul

(Mat. Nr.207 200 109)

**Thema: PokerTool - Entwicklung und Implementierung einer AR-Android-Anwendung
für Wahrscheinlichkeitsberechnungen bei Texas Holdem Poker**

Durch die Verbreitung von Smartphones und der damit einhergehenden erhöhten Aufmerksamkeit an Augmented Reality Anwendungen ist es spannend, einen wirklichen Mehrwert durch eine solche Anwendung zu generieren. Hierbei ist es wichtig, die Grundfunktionen eines Computers mit den möglichen Vorteilen einer AR-Anwendung zu einer hilfreichen Applikation zu verbinden. Es muss die Rechenkraft eines Computers benutzt werden, um die reale Welt sinnvoll mit hilfreichen, virtuellen Informationen anzureichern. Wichtig hierbei ist es, eine einfache Bedienung und einen schnellen Umgang mit einer solchen Anwendung zu gewährleisten.

Ein Poker-Assistent soll alle Vorteile einer AR-Anwendungen einsetzen. Die schwierige Wahrscheinlichkeitsberechnung von Gewinnchancen, sowie die schnelle automatisierte Kartenerfassung sollen hierbei im Vordergrund stehen.

AR-Anwendungen auf Smartphones bilden den Grundstein für ein zukünftiges AR-unterstütztes Leben. In diesem Kontext soll eine der Kernfragen dieser Masterarbeit die Frage nach dem wirklichen Mehrwert für Nutzer dieser App im Vergleich zu nicht unterstützten Spielern sein. Im Hinblick auf die Zukunft soll diskutiert werden, wie eine sinnvolle AR-Unterstützung aussehen sollte.

Die inhaltlichen Schwerpunkte der Arbeit sind:

1. Recherche
2. Konzeption einer AR-Poker Wahrscheinlichkeits-App
3. Prototypische Umsetzung
4. Evaluation hinsichtlich des Mehrwertes gegenüber einem nicht unterstützten Spieler
5. Dokumentation der Ergebnisse und Ausblick

Koblenz, den 23.01.2013

– Christoph Thul –

– Prof. Dr. Stefan Müller –

Zusammenfassung

Durch die Verbreitung von Smartphones und der damit einhergehenden, erhöhten Aufmerksamkeit für Augmented Reality Anwendungen, ist es spannend einen wirklichen Mehrwert durch eine solche Anwendung zu generieren. Es ist wichtig, die Grundfunktionen eines Computers mit den möglichen Vorteilen einer AR-Anwendung zu einer hilfreichen App zu verbinden. Es muss die Rechenkraft eines Computers mit hilfreichen, virtuellen Informationen in der realen Welt sinnvoll angereichert werden. Wichtig ist zudem, dass eine einfache Bedienung und schneller Umgang mit einer solchen Anwendung gewährleistet sein.

Ein Poker-Assistent kann all diese Stärken des Computers und einer AR-Anwendung einsetzen. Die schwierige Wahrscheinlichkeitsberechnung von Gewinnchancen sowie die schnelle automatisierte Kartenerfassung sollen hierbei im Vordergrund stehen.

AR-Anwendungen auf Smartphones bilden den Grundstein für ein zukünftiges AR-bestimmtes Leben. In diesem Kontext soll eine der Kernfragen dieser Masterarbeit die Frage nach dem wirklichen Mehrwert für Nutzer dieser App im Vergleich zu nicht unterstützten Spielern sein. Wobei auch geklärt werden soll, ob dieses Anwendungsbeispiel eine sinnvolle AR-Unterstützung benötigt und wie, im Hinblick auf die Zukunft und zukünftige Technologien, eine sinnvolle AR-Unterstützung aussehen sollte.

Abstract

Today you can find smartphones everywhere. This situation created a hype for Augmented Reality and AR Apps. The big question is: Do these applications provide a real added value? To make AR practically it is important to add the computational power of a computer to the advantages of AR. An easy and fast way of interaction is essential.

A Poker-Assistance-Software is an ideal test area for an AR Application with real added value. The estimation of the winning probability and a fast automated tracking of the playing cards is the perfect field of investigation.

In this discussion it is interesting to evaluate the added value of AR Applications in common.

"Ja, lach du nur, du dämliches Pelzvieh."
-Han Solo

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Ziel der Masterarbeit	1
1.3. Aufbau der Masterarbeit	2
2. Texas Holdem Poker	3
2.1. Begriffserklärung	3
2.2. Spielablauf	5
2.3. Hände	7
2.4. Wahrscheinlichkeitsberechnung	7
2.4.1. Grundlage	7
2.4.2. Odd-Berechnung	8
2.4.3. Gewinnchanceberechnung	9
2.5. Poker Apps	10
3. Kartenerkennung	12
3.1. Metaio SDK	12
4. Augmented Reality	14
4.1. Begriffserklärung und Definition	14
4.2. Mehrwert	15
4.3. Beliebte Augmented Reality Applikationen	16
4.4. Besondere Anforderungen durch Augmented Reality	17
4.4.1. Hardware und Software	18
5. PokerTool	21
5.1. Problemstellung	21
5.2. Vision	21
5.3. Konzeption	22
5.3.1. Anforderungsdefinition	23
5.3.2. Plattform	24
5.4. Graphical User Interface	24
5.4.1. Grundsätze der Dialoggestaltung	25
5.4.2. Android Design Principles and Styleguide	26

5.4.3. PokerTool Design	27
5.5. Implementierung	29
5.5.1. Aktivitätsdiagramm	30
5.5.2. Startbildschirme	31
5.5.3. Laden des Contents	33
5.5.4. Kartendetektion und Augmented Reality Feedback	35
5.5.5. Ermittlung der Wahrscheinlichkeiten	37
5.5.6. Ermittlung des Ranges	43
5.5.7. Gestensteuerung	53
6. Evaluation	55
6.1. These	55
6.2. Aufbau	55
6.3. Ergebnisse	56
6.3.1. Fragebogen	57
6.3.2. Log-Daten	62
6.4. Bewertung	79
7. Fazit und Aussicht	81
7.1. Fazit	81
7.2. Aussicht	82
Literaturverzeichnis	83
A. Anhang	i
A.1. CD	i
Abbildungsverzeichnis	ii
Tabellenverzeichnis	iv
Verzeichnis der Listings	v

1. Einleitung

PokerTool - Entwicklung und Implementierung einer AR-Android-Anwendung für Wahrscheinlichkeitsberechnungen bei Texas Holdem Poker soll zuerst auf die Motivation, das Ziel und den Aufbau der Masterarbeit eingehen.

1.1. Motivation

Augmented Reality hat durch die starke Performance und die Hardware-Ausstattung von Smartphones einen großen Anwendungsbereich gefunden und viel Fürsprache erfahren. Immer mehr Applikationen versuchen dem Benutzer durch AR-Bestandteile eine neue Art der Hilfestellung oder Sicht auf die Umwelt zu geben, erweisen sich jedoch meistens als unpraktisch und besitzen keinen wirklichen Mehrwert. Die Motivation dieser Masterarbeit liegt darin, dem Benutzer ein besonderes AR-Erlebnis mit wirklichem Mehrwert zu verschaffen. Für diese Problemstellung ist die Grundidee eines AR-Poker-Assistenzprogrammes ideal, da sie alle Vorteile einer Augmented Reality Anwendung vereint und einen wirklichen Mehrwert durch Information bietet, die ein Mensch nur sehr schwer selbst generieren kann. Im Gegenzug können durch die klar definierte Spielsituation, die Nachteile, wie z.B. ein beschränktes Sichtfeld oder unpraktische Handhabung, einer Smartphone-Nutzung minimiert werden.

1.2. Ziel der Masterarbeit

Das Ziel dieser Masterarbeit soll die Entwicklung und Implementierung einer sinnvollen Augmented Reality Anwendung sein, die dem Benutzer einen klaren Mehrwert verschafft und alle Vorteile eines computergestützten Assistenzprogrammes vorweisen kann. Das entstehende Programm *PokerTool* soll durch die Konzeption ein durchdachtes Design erlangen, das den Benutzer akkurat, robust und praktisch unterstützt. Das Design sowie die Funktionsmechanik, sollen nach den aktuellsten Gestaltungsgrundsätzen erstellt werden. *PokerTool* soll unter diesen Berücksichtigungen eine AR-Anwendung schaffen, die der Benutzer beim Pokern benutzen kann, um während des Spiels bessere Entscheidungen zu treffen. Dafür soll der Spieler das Spiel mit der Smartphone-Kamera verfolgen können. Die Anwendung soll die Karten automatisch erfassen und dem Benutzer die

Wahrscheinlichkeiten für einen Gewinn oder eine mögliche, bessere Hand errechnen. Die Implementation soll prototypisch auf Basis des Android-Betriebssystems geschehen und evaluiert werden. Hierbei sind die allgemeinen Fragen von Interesse, ob Augmented Reality überhaupt beim aktuellen Stand der Technik praktisch ist oder einen wirklichen Mehrwert bietet. Diese sollen durch diese Masterarbeit erörtert werden.

1.3. Aufbau der Masterarbeit

Die Arbeit soll den kompletten Schaffungsprozess von *PokerTool* erfassen. Zu Beginn soll hierfür auf das Einsatzgebiet Poker und insbesondere auf die Variante Texas Holdem eingegangen werden. Dort sollen alle Grundlagen zum Spiel und die Wahrscheinlichkeitsberechnung dahinter erklärt werden. Im Anschluss sollen die Problemlösungen für die Kartenerkennung und das Thema Augmented Reality im Allgemeinen diskutiert werden. Nach der Betrachtung der Grundlagen soll die Konzeption sowie die prototypische Implementation von *PokerTool* beschrieben werden. Zum Abschluss wird das entstandene Programm evaluiert. Dabei sollen auch Rückschlüsse auf einen Mehrwert durch Augmented Reality im Allgemeinen geschlossen werden. Zuletzt erfolgt eine Diskussion über die Zukunft von AR.

2. Texas Holdem Poker

Dieser Abschnitt befasst sich mit der Pokervariante Texas Holdem. Es sollen zuerst die benötigten Begriffe eingeführt und erklärt werden. Im Anschluss soll der Spielablauf und die Reihenfolge der Gewinnhände beschrieben werden. Mit diesem Hintergrundwissen kann auf die Wahrscheinlichkeitsberechnung eingegangen werden.

2.1. Begriffserklärung

Pokern, mit seiner Variante Texas Holdem, hat eine lange Tradition. Dieser Tradition ist es zu schulden, dass viele Begrifflichkeiten entstanden sind, die für Außenstehende unter Umständen wenig Sinn machen. Auf diese Begriffe soll hier eingegangen werden, um im Folgenden den Spielablauf vollständig zu verstehen. (Gambke [2010])

- **Farbe** Die Farbe in Kartenspielen beschreibt, ob es sich um Karo (♠), Pik (♣), Kreuz (♥) oder Herz (♦) handelt.
- **Deck** Das Deck ist das vollständige Kartenspiel. Es enthält beim Pokern in der Regel 52 Karten. Die Werte einer *Farbe* erstrecken sich von 2 bis 10, Bube, Dame, König und Ass. Jeder Wert kommt einmal pro *Farbe* vor. Das vollständige Deck wird pro Runde gemischt.
- **Hole Cards/Starthand** Die Hole Cards sind im Poker Texas Holdem die zwei persönlichen Karten, die nur der Spieler selbst sehen und kennen sollte. Im deutschen werden sie auch *Starthand* genannt. Diese beiden Karten sind die erste Informationsquelle des Spielers und die einzige Möglichkeit, die *Community Cards* zu verbessern, um so die bestmögliche *Hand* zu erreichen.
- **Community Cards/Board** Die Community Cards oder auch das Board sind die für jeden Mitspieler offen ausgelegten Karten im *Flop*, *Turn* und *River*.
- **Preflop** Das Austeilen der *Starthände* und die erste *Setzrunde* geschieht im sogenannten Preflop.
- **Flop** Der Flop besteht aus den ersten drei, offen ausgelegten Karten der *Community Cards*.
- **Turn** Der Turn ist die vierte, offen ausgelegte *Community Card*.

- **River** Der River ist die fünfte und letzte offen ausgelegte *Community Card*.
- **Hand/Rang** Die Hand oder der Rang ist die beste Kombination, die sich aus den *Community Cards* und den *Hole Cards* für den jeweiligen Spieler ergibt. Insgesamt hat der Spieler zum Schluss sieben Karten, aus denen er die beste Kombination bestimmt. Hierfür kann er seine beiden, eine oder auch keine *Hole Card* einbringen. Wenn ein Spieler keine *Hole Card* einbringen kann, spielt er das sogenannte *Board*. Auf die Reihenfolge bzw. Wertigkeit der Hände wird im Abschnitt 'Hände' näher eingegangen.
- **Suited/Off-Suited** Haben die beiden *Hole Cards* die gleiche *Farbe* wird dieses Paar als *Suited* bezeichnet (Bsp. ♠ Ass und eine ♠ 10). Im Gegenzug bezeichnet man die *Hole Cards* mit verschiedenen Farben als *Off-Suited*.
- **Connectors** Sind die Werte der *Starthand* aufeinanderfolgend, erhalten sie das Merkmal *Connected* (Bsp. 9 ♣ und 10 ♠). Das Merkmal *Suited* (bzw. *Off-Suited*) kann mit der Eigenschaft *Connectors* verbunden werden.
 - Beispiel: Die Starthand Ass ♠ und König ♠ erhält das Attribut *Suited-Connectors* und ist mit besseren Gewinnchancen ausgestattet als *Off-Suited* oder keine *connected Cards*.
- **Overcards** Alle Karten, die einen höheren Wert als die höchste *Hole Card* des Spielers besitzen, bezeichnet man als *Overcards*.
- **Gutshot** Die spezielle *Hand Straße* besteht aus im Wert fünf aufeinanderfolgenden Karten. Ist die Straße nicht vollständig, weil eine Karte in der 'Mitte' fehlt, bezeichnet man diese Situation als *Gutshot*.
 - Beispiel: Nach dem *Flop* hält der Spieler die Starthand 10 ♣ und Bube ♠. Der *Flop* besteht aus 3 ♠, Dame ♣ und Ass ♡. Bei dieser Konstellation liegt ein *Gutshot* vor, weil dem Spieler ein König fehlt, um eine Straße zu bilden.
- **Open Ended** Besitzt der Spieler eine *Hand*, bei der vier Kartenwerte aufeinanderfolgen und nur noch eine Karte 'unterhalb' oder 'oberhalb' der Zahlenfolge zu einer *Straße* fehlt, wird dies als *Open Ended* bezeichnet.
- **Draw** Kann der Spieler mit den bisherigen *Community Cards* vier Karten einer *Farbe* vorweisen, spricht man von einem *Draw*.
- **Outs** Als *Outs* bezeichnet man alle Karten, die der eigenen *Hand* fehlen, um eine bessere *Hand* zu erreichen.
 - Beispiel: Mit den *Community Cards* hat ein Spieler 5 ♡, 10 ♠, 10 ♣, König ♠ und Ass ♠. Er spielt zur Zeit ein 10er Pärchen. Um aus diesem Pärchen

einen *Drilling* zu machen, fehlen ihm entweder die 10 ♡ oder die 10 ◇. Er hat somit zwei Outs auf die Chance einen *Drilling* zu erreichen.

- **Odds** Die Wahrscheinlichkeiten eine bestimmte *Hand* im nächsten oder übernächsten Zug zu erhalten, werden mit den Odds ausgedrückt. Diese Prozentangabe ergibt sich aus den *Outs*.
- **Showdown** Der Showdown wird erreicht, wenn nach offenlegen aller fünf *Community Cards* mindestens zwei Spieler, ihre eigene *Starthand* aufdecken, um zu ermitteln, wer gewonnen hat.
- **Pott** Der Pott bezeichnet das gesamte Geld bzw. Chips, das während allen Setzrunden gesammelt wurde. Dieses Geld geht an den Spieler, der zuletzt im Spiel ist oder nach dem Showdown die beste *Hand* hat.
- **Splitpott** Haben beim *Showdown* mindestens zwei Spieler eine gleichstarke *Hand*, wird der Pott unter diesen Spielern aufgeteilt.
- **Nuts** Die bestmögliche *Hand* in einer Runde wird als Nuts bezeichnet. Niemand kann diese *Hand* in dieser Runde schlagen.
- **Bluff** Gibt ein Spieler vor, durch z.B. Setzen eines hohen Geldbetrages, eine gute *Hand* zu besitzen, obwohl er eigentlich eine schlechtere *Hand* hält, mit der nur eine geringe Aussicht auf den Gewinn besteht, so nennt man diesen Vorgang bluffen.
- **Kicker/Beikarte** Als Kicker oder Beikarte wird die Karte bezeichnet, welche, wenn zwei oder mehr Spieler den gleichen Rang einer Hand halten (Bsp. jeweils ein Königspärchen), das Spiel entscheidet. Hier gewinnt die höchste Beikarte.

2.2. Spielablauf

In diesem Abschnitt soll auf den Spielablauf der Pokervariante Texas Holdem nach Gambke [2010] eingegangen werden. Der grundsätzliche Ablauf ist bei allen Pokervarianten derselbe. Der Spieler setzt Geld auf seine Chance zu gewinnen. Bei Texas Holdem entsteht durch die *Community Cards* eine besondere Situation, weil ein Teil der Information, die alle Spieler erhalten, derselbe ist. Diese Information muss der Spieler nutzen, um seine eigenen Gewinnchancen einzuschätzen. Es sollen nun ausführlich die maximal neun Phasen des Spielablaufes erklärt werden, um im Abschnitt 5: 'PokerTool' die Vorgehensweise der Software zu verstehen und wichtige Entscheidungen zu begründen.

1. **Preflop** Zu Beginn jeder Runde erhält jeder Spieler zwei verdeckte Karten, die *Hole Cards* bzw. *Starthand*. Diese Karten darf nur der jeweilige Spieler kennen. Anhand von ihnen entscheidet jeder Spieler für sich, ob er diese Runde mitspielen möchte.

Es folgt die Setzrunde, die immer gleich abläuft. Im Preflop gibt es jedoch den Sonderfall eines vordefinierten Mindesteinsatz, die *Blinds*. Die Blinds sollen einen Spieler zwingen, den Spielverlauf nicht ohne Verlust 'auszusitzen'. In allen anderen Setzrunden gibt es keinen Mindesteinsatz, dieser wird erst von dem ersten Spieler, der Geld setzt, vorgegeben.

2. **Setzrunde** Der Spieler, der an der Reihe ist, hat jeweils fünf Möglichkeiten zu reagieren:
 - **Fold** Entweder folded er. D.h. er möchte diese Runde nicht spielen, weil seiner Meinung nach die Karten keine Aussicht auf Erfolg haben oder der Preis, um weiterzuspielen zu hoch ist.
 - **Call** Die zweite Möglichkeit ist callen. Wenn ein Einsatz bereits gesetzt wurde, muss die gleiche Summe gezahlt werden, um mitzuspielen. Wenn dieser Einsatz gezahlt wird, nennt man diese Handlung Callen.
 - **Bet** Ist kein Mindesteinsatz gezahlt (dies kann auf Grund der Blinds im Preflop nicht vorkommen), kann der Spieler, der an Reihe ist, einen Mindesteinsatz definieren.
 - **Raise** Die vierte Möglichkeit lautet Raise. Von jedem Spieler, dessen Zug ansteht, kann der Mindesteinsatz für alle Spieler erhöht werden.
 - **Checken** Die letzte Möglichkeit ist das Checken. Hat der Spieler, der an der Reihe ist, den geforderten Einsatz bereits erfüllt, kann er checken und somit nichts tun.

Wenn alle im Spiel befindlichen Spieler den gleichen Betrag bezahlt haben, ist die Setzrunde beendet.

3. **Flop** Sind mindestens zwei Spieler noch im Spiel, beginnt die nächste Phase mit der Bekanntgabe des Flops. Es werden drei Karten für alle offen aufgedeckt. Nun hat der jeweilige Spieler neue Informationen und die neue Setzrunde beginnt.
4. **Setzrunde** Jeder noch im Spiel befindliche Spieler hat wieder die fünf Möglichkeiten des Foldens, Checkens, Bettens, Callens und Raisens. Es gilt die gleiche Abfolge wie bereits im Preflop, jedoch gibt es nun keine Blinds mehr.
5. **Turn** Sind nun weiterhin mindestens zwei Spieler im Spiel, wird die Turn-Karte aufgedeckt.
6. **Setzrunde** Wie im Preflop und nach dem Flop hat jeder im Spiel befindliche Spieler seine fünf Aktionsmöglichkeiten.
7. **River** Wenn weiterhin mehr als ein Spieler im Rennen ist, wird mit dem River die letzte Karte aufgedeckt.

8. **Setzrunde** Ein letztes Mal kommt es zu einer Setzrunde. Zu diesem Zeitpunkt hat der Spieler alle zur Verfügung stehenden Informationen erhalten und seine endgültige Hand steht fest.
9. **Showdown** Ist die Setzrunde beendet und mindestens zwei Spieler haben ihre Karten noch nicht gefolded, kommt es zum Showdown und der Gewinner wird ermittelt.

2.3. Hände

Um den Gewinner beim Showdown zu ermitteln, gibt es eine Rangfolge an Kartenkombinationen (Risk [2009]). Diese Kombinationen sollen in der folgenden Tabelle 2.1. eingeführt werden. Dabei wird hier die schwächste Hand zuerst genannt und erhält die Rangnummer 1. Sie wird von allen höheren Rangnummern geschlagen.

2.4. Wahrscheinlichkeitsberechnung

Pokern ist ein *Glückspiel*, das sich aus den Kombinationsmöglichkeiten eines vollständigen Decks zusammensetzt. Die Wahrscheinlichkeitsberechnung kann hierbei zwei verschiedene Szenarien berechnen. Zum Einen kann die Gewinnwahrscheinlichkeit der Hand in dieser Runde berechnet werden, zum Anderen die Wahrscheinlichkeit im nächsten Zug eine bestimmte Hand zu erreichen. Die zweite Wahrscheinlichkeit wird in der Fachsprache auch als *Odd* bezeichnet. Obwohl beide Wahrscheinlichkeiten miteinander verknüpft sind, werden sie verschieden berechnet. Beide Informationen sollen in der Applikation *PokerTool* dem Benutzer zur Verfügung stehen.

2.4.1. Grundlage

Bei der Berechnung der Wahrscheinlichkeiten gibt es mehrere Ansätze. Beiden Szenarien liegt das gleiche Konstrukt der Kombinatorik zu Grunde und soll hier zuerst erklärt werden. Insgesamt gibt es

$$\binom{52}{5} = 2.598.960 \text{ verschiedene mögliche Hände} \quad (2.1)$$

beim *Showdown*, die in ihrer Wertigkeit in einer Hierarchie eingeordnet sind. Durch die Besonderheit von sieben Karten, die ein Spieler bei der Spielvariante Texas Holdem im Showdown zur Verfügung hat, ergeben sich bereits 133.784.560 verschiedene Kombinationen nach dem River. Um jede Kombination zu berechnen, die ein Spieler in einer

Runde gegenüberstehen kann, müssen die jeweiligen Möglichkeiten multipliziert werden. Es ergeben sich

$$\binom{52}{2} \times \binom{50}{3} \times \binom{47}{1} \times \binom{46}{1} = 56.189.515.200 \text{ Kombinationen.} \quad (2.2)$$

Da diese Anzahl von Möglichkeiten der Ausgangspunkt für die Berechnung der jeweiligen Gewinnchancen ist, zeigt sich bereits hier, dass ein assistierendes Programm sinnvoll ist. ([Podeszwa 2010])

2.4.2. Odd-Berechnung

Die Odds zeigen die Chance, eine bestimmte Hand im nächsten bzw. übernächsten Zug zu erhalten und berechnen sich aus den *Outs*. Hierbei wird die Wahrscheinlichkeit des Erreichens für jeden einzelnen Rang ermittelt. Um die Wahrscheinlichkeit zu berechnen, muss in zwei Schritten vorgegangen werden:

1. Outs bestimmen: Wie viele Karten gibt es, um eine bestimmte Hand zu erreichen.

Beispiel: Der Spieler hält nach dem Flop folgende Karten: Ass \heartsuit , Ass \diamondsuit , König \spadesuit , 7 \diamondsuit und 2 \heartsuit . Um den ersten Schritt vollständig abzuschließen, müssten für jeden Rang die Outs ermittelt werden. Beispielhaft soll dies an der Hand Drilling geschehen. Um einen Drilling im nächsten Zug bzw. übernächsten Zug zu erreichen, benötigt der Spieler das Ass \clubsuit oder das Ass \spadesuit . Daraus ergeben sich zwei Outs.

2. Odds berechnen: Die Wahrscheinlichkeit eine Hand zu erreichen wird berechnet.

Der zweite Schritt ist, die jeweiligen Outs für die Hände in Wahrscheinlichkeiten umzurechnen. Hierbei wird folgende Rechnung benötigt:

$$P_t = 1 - \frac{47 - \text{outs}}{47} \quad (2.3)$$

$$P_r = 1 - \frac{46 - \text{outs}}{46} \quad (2.4)$$

$$P_{t+r} = 1 - \left(\frac{47 - \text{outs}}{47} \times \frac{46 - \text{outs}}{46} \right) \quad (2.5)$$

Die Wahrscheinlichkeit, dass der Spieler seine Hand durch den Turn verbessert, wird durch die Gleichung P_t gezeigt. Eine Verbesserung durch die River-Karte wird durch die Gleichung P_r ermittelt. Um zu berechnen, ob die Hand im Turn oder River erlangt wird, gibt es die Gleichung P_{t+r} . (Poneta [2011])

Rang	Name	Beschreibung	Beispiel
1	High Card	Höchste Karte; Hat niemand mindestens ein <i>Pair</i> gewinnt derjenige mit der höchsten Karte. Haben zwei Spieler die gleiche höchste Karte, entscheidet die zweithöchste usw.	High Card Ace 
2	One Pair	Zwei Karten desselben Wertes; Halten mindestens zwei Spieler ein Paar mit demselben Wert, entscheidet die höchste Beikarte	One Pair Aces 
3	Two Pair	Zwei Pärchen	Two Pairs of Aces and Tens 
4	Three of a Kind	Drei Karten eines Wertes	Three of a Kind of Aces 
5	Straight	Fünf Karten, deren Wert aufeinander folgt; Das Ass kann hierbei als der Wert 'Eins' oder als höchster Wert hinter dem König gewertet werden; Die kleinste Straight beginnt mit einem Ass, die größte endet mit einem Ass	Straight 
6	Flush	Fünf Karten derselben Farbe; Können mindestens zwei Spieler einen Flush aufzeigen, gewinnt derjenige mit der höchsten Karte des Flushs	Flush 
7	Full House	Ein <i>Three of a Kind</i> und ein <i>Pair</i> ; Besitzt mehr als ein Spieler einen Flush, gewinnt derjenige mit dem höchsten <i>Three of a Kind</i> . Haben mindestens zwei Spieler einen gleichwertigen <i>Three of a Kind</i> , gewinnt das höhere <i>Pair</i>	Full House 
8	Four of a Kind	Alle vier Karten eines Wertes in den jeweiligen Farben	Four of a Kind of Aces 
9	Straight Flush	Eine Straight, in der die fünf Karten darüber hinaus auch dieselbe Farbe besitzen	Straight Flush 
10	Royal Flush	Ein Straight Flush, in der die fünf Karten darüber hinaus von 10 bis zum Ass reichen	Royal Flush 

Tabelle 2.1.: Rangfolge der nach unten besser werdenden verschiedenen Pokerhände

2.4.3. Gewinnchanceberechnung

Die Gewinnchance eines Spielers ändert sich im Spiel mehrere Male. Diese Änderungen kommen durch den ständigen Informationsgewinn zustande. Grundlegend bestimmen zwei große Faktoren die Gewinnwahrscheinlichkeit.

1. Welche Karten besitzt der Spieler.
2. Wie viele Gegenspieler sind beteiligt.

Eine schnelle und genaue Möglichkeit, diese Faktoren einzubeziehen, ergibt sich aus einer empirischen Messung. Die Chancen ergeben sich aus dem Sammeln der Häufigkeit des Siegens im Showdown mit einer jeweiligen Starthand bei einer bestimmten Anzahl von Gegnern.

Im Preflop spielt bei dieser Messung die Farbe keine Rolle. Es ist nur von Bedeutung, ob die Starthand *suited* oder *off-suited* vorliegt. Somit entstehen 169 Startkombinationen, für die für einen bis neun Gegenspieler in mehreren Millionen Durchläufen der empirische Wert ermittelt wurde. Da die Startformation immer die gleiche ist, gibt es somit nur 1521 Startwahrscheinlichkeiten. In Tabelle 2.2. wird ein Auszug von Thema-Poker.com [2013] der Wahrscheinlichkeiten mit der jeweiligen Starthand bei drei Gegner gezeigt. Alle vollständigen Tabellen können auf [Thema-Poker.com 2013] eingesehen werden.

Rang	Hand	Anzahl	Gewinnchance	Splitchance
1	AA	6	63,9%	0,4%
2	KK	6	58,2%	0,4%
3	QQ	6	53,5%	0,4%
4	JJ	6	49,2%	0,5%
5	TT	6	45,3%	0,5%
6	AKs	4	41,7%	1,0%
7	99	6	41,1%	0,5%
8	AQs	4	40,1%	1,1%
9	AK	12	38,9%	1,0%
10	AJs	4	38,7%	1,2%
11	KQs	4	38,3%	1,1%
12	88	6	37,6%	0,5%
13	ATs	4	37,6%	1,3%
14	AQ	12	37,1%	1,1%
15	KJs	4	37,0%	1,1%

Tabelle 2.2.: Gewinnwahrscheinlichkeit im Preflop bei drei Gegenspielern (Thema-Poker.com [2013])

Durch die Offenlegung des Flops entstehen, wie bereits oben gezeigt, zu viele verschiedene Kombinationen, um diese in Tabellen zu sammeln. Eine schnelle Lösung nach TEO-FILO (Teofilo [2011]) ist die dynamische, empirische Ermittlung der Handstärke. Hierbei

wird die jetzige entstandene Hand gegen eine ausreichend große Stichprobe von gegnerischen Händen evaluiert und wieder die Häufigkeit der Siege ermittelt. Die entstandene Wahrscheinlichkeit ist eine Momentaufnahme des aktuellen Zustandes. Um die Entwicklung der Hand einzuberechnen, müssen alle *möglichen Hände des Spielers* gegen alle *entstehenden, möglichen Hände der Gegner* einbezogen werden. Der Einbezug der Entwicklung der Hand muss im späteren Verlauf in einer Kosten-Nutzen-Rechnung (Performance vs. Genauigkeit der Wahrscheinlichkeit) entschieden werden. Die Evaluation muss nach dem River und Turn jeweils erneut durchgeführt werden, da neue Informationen einbezogen werden müssen und um eine gute Approximation der Wahrscheinlichkeiten zu berechnen.

2.5. Poker Apps

Für diese Problemstellung wurde bereits von verschiedenen Entwicklern versucht ein Tool den Benutzern zur Seite zu stellen. Im Android Markt *Play Store* ist die beliebteste App der *Poker Odd Calculator* mit über 100.000 Downloads. Dieses Programm approximiert in Echtzeit die Wahrscheinlichkeit gegen jede Anzahl von Gegnern zu gewinnen.

Abbildung 2.1. zeigt das Interface und seine Benutzung. Der Spieler muss wie in B gezeigt, jede Karte einzeln eingeben und erhält in Echtzeit die Chancen. Die sofortige Angabe der Approximation wird zeitgleich weiterhin verbessert. Diesen Fortschritt kann man in C in einem Fortschrittsbalken erkennen. In diesem Beispiel besagt die Anzeige, dass bisher ungefähr die Hälfte der Berechnung beendet worden ist.

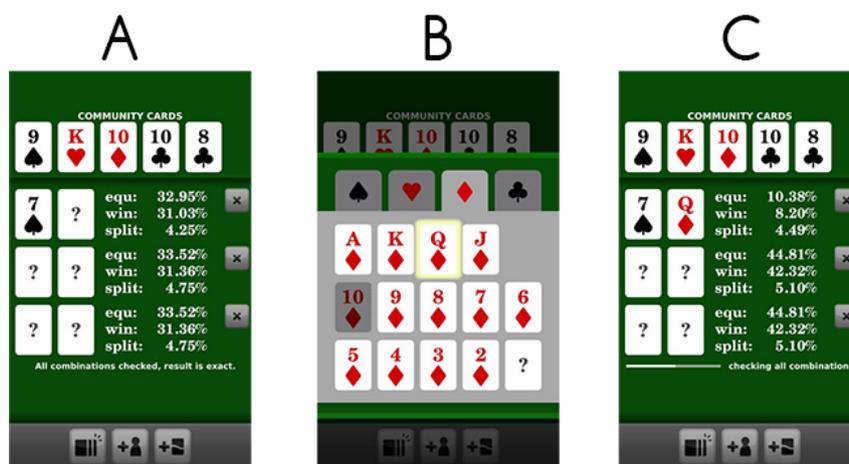


Abbildung 2.1.: Poker Odd Calculator: Android Anwendung zur Bestimmung der Gewinnwahrscheinlichkeiten

Das *Graphical User Interface* ist aufgeräumt und konzentriert sich auf seine Aufgabenstellung. Die Benutzung ist simpel, jedoch verlangt das Programm durch die Eingabe der

POKERTOOL

2. *Texas Holdem Poker*

Werte die Aufmerksamkeit des Benutzers. Genau dieses Problem soll *PokerTool* lösen. Durch eine automatisierte Kartenerkennung soll das Smartphone mehr zu einem Begleiter und einer dritten Karte werden, die der Spieler in der Hand hält, als ein technischer Fremdkörper, mit dem der Benutzer interagieren muss.

3. Kartenerkennung

Die Grundidee von *PokerTool* setzt sich aus Wahrscheinlichkeitsrechnung mit automatisierter Kartenerkennung zusammen. Die Kartenerkennung ist ein sehr interessantes Feld aus der Bildverarbeitung und kann eine eigenständige Masterarbeit ausfüllen. Diese Arbeit soll sich auf das Augmented Reality-Erlebnis sowie den Mehrwert von AR-Anwendungen konzentrieren. Aus diesem Grund wurde bei der Kartenerkennung auf das Metaio SDK zurückgegriffen. Im Folgenden soll das Framework und seine Aufgabe erläutert werden.

3.1. Metaio SDK

Das Metaio SDK ist ein Augmented Reality Entwicklerframework, welches die Tracking und Rendering Pipeline für Smartphones übernimmt. Das Tracking geschieht hierbei entweder über *ID Marker* oder *Picture Marker*. Für *PokerTool* ist die Konfiguration via *Picture Marker* interessant, da hier jedes erdenkliche Bild als Marker definiert werden kann. Die Lösung, die jeweiligen Spielkarten an sich als Marker zu bestimmen, führt so zu einer robusten und akkuraten Methode die Karten zu tracken. Die Tracking Konfiguration wird in einer XML vorgenommen und definiert dort:

- das Referenzbild
- die Position des Koordinatensystems in Relation zum Referenzbild
- die Anzahl der gleichzeitig zu trackenden Objekte
- den Schwellenwert der Ähnlichkeit des Referenzbildes
- die Anzahl der Detektion für ein Referenzbild bis ein *Match* festgestellt wird

Die letzten drei Punkte definieren hier die wichtigsten Stellschrauben um eine robuste und akkurate Kartenerkennung zu erzeugen.

Den zweiten wichtigen Part, der durch das Metaio SDK übernommen wird, ist die Anreicherung des Kamerabildes mit digitalen Inhalten. Durch die Verknüpfung der Koordinatensysteme der Referenzbilder mit 3D Modellen, kann der gewünschte digitale Content auf dem Smartphonebildschirm erscheinen. Hierfür muss ein Model erstellt werden, das

auch eine Animation aufweisen kann. *PokerTool* möchte die Kartenerkennung durch eine Augmented Reality Anzeige lösen und dem Benutzer somit ein intuitives Feedback geben, ob eine Karte erkannt wurde. ([metaio 2013])

4. Augmented Reality

Der Themenbereich *Augmented Reality* ist ein sehr aktuelles und spannendes Thema. In diesem Abschnitt soll aber einmal der oft benutzte Begriff AR definiert und die Theorie dahinter erklärt werden.

4.1. Begriffserklärung und Definition

Nach MILGRAM von 1994 (Paul Milgram [1994]) ist *Augmented Reality (AR)* ein Teilgebiet der *Mixed Reality*. AR ist, wie Abbildung 4.1. zeigt, der Schritt zwischen der realen Umgebung und einer virtuellen Umwelt.

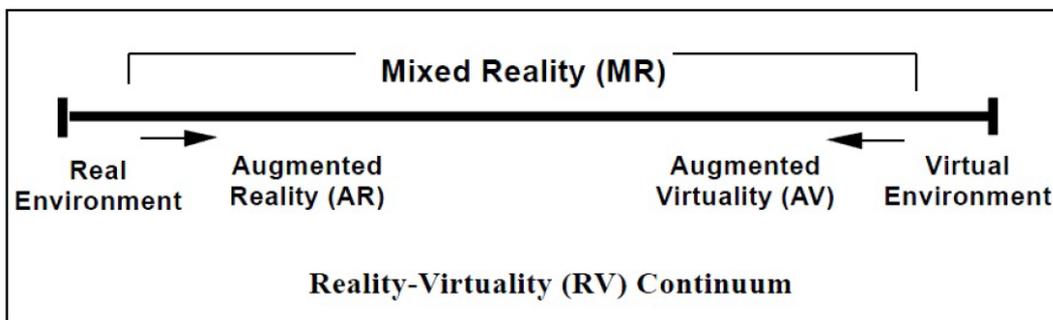


Abbildung 4.1.: Vereinfachte Einordnung der Begriffe Augmented Reality im Kontext der Virtual Reality Continuums nach MILGRAM ([Paul Milgram 1994])

Der wichtigste Punkt von Mixed Reality ist, dass die reale Welt mit virtuellen Inhalten vermischt wird. Dies sollte nicht mit *Virtual Reality* verwechselt werden, in der der Benutzer komplett in eine virtuelle Welt abtaucht. Mixed Reality und Augmented Reality werden beide im ähnlichen Kontext benutzt, jedoch hat AR nach AZUMA drei eindeutige Charakteristiken, die bei MR nicht vollständig erfüllt werden müssen ([Azuma 1997]):

1. Kombiniert reale Welt mit virtuellen Inhalten
2. Echtzeit-Interaktion
3. Abhängiger 3D Bezug zwischen realer Welt und virtuellen Inhalten

AZUMA war es hierbei wichtig, dass AR nicht von einem speziellen Trägermedium wie *Head-Up-Displays (HUD)* abhängig ist. Somit ist die Definition von 1994 20 Jahre später mit der Verbreitung von Smartphones noch aktuell. AR kann in seinem Teilgebiet trotzdem nach MILGRAM und AZUMA über die technische Schnittstelle definiert werden. Hierbei unterscheiden sich *durchsichtige Displays* und *monitorbasierte Displays*. Während bei den durchsichtigen Display fensterartig auf dem realen Bild gearbeitet wird, steht beim monitorartigen Vorgang eine Kamera im Fokus, die das anzureichernde Bild zuerst erfasst. In dieser Arbeit wird die prototypische Umsetzung via Smartphone geschehen und auf die zweite Kategorie zurückgegriffen. Die Umsetzung mit Hilfe eines Smartphones hat vor allem die Vorteile der aktuellen Machbarkeit, Mobilität und der Transformation der 3-dimensionalen Welt auf eine 2-dimensionale Ebene.

4.2. Mehrwert

Der Mehrwert von AR-Anwendungen ist es, dem Benutzer Informationen zu einem bestimmten realen Ort in Echtzeit zu vermitteln. Diese Informationen können durch *Hinzufügen* von digitalen Inhalten zur realen Welt oder durch *Löschen* von realen existierenden Inhalten durch digitale Inhalte gewonnen werden. (Azuma [1997]) Die Sicht des Benutzers wird transformiert, da die zusätzlichen Informationen im besten Falle nur durch die AR-Anwendung vermittelt werden können.

Ob nun ein wirklicher Mehrwert hinter einer AR Anwendung steht, ist ein viel diskutiertes Thema in der Fachwelt (Bachfischer [2011]). Laut BACHFISCHER ist AR zur Zeit ein „innovatives und spannendes Marketing Tool“. Der bisherige Vorteil im Bereich Marketing lässt ihn die Frage stellen, ob überhaupt genügend Mehrwert für den Benutzer erzeugt wird, da sonst AR-Anwendungen in mehr Bereichen des alltäglichen Lebens zu finden wären. Er findet keine Antwort auf die Frage, prognostiziert jedoch, dass die Zukunft die Fragen beantwortet.

KATIER der Universität Amsterdam kommt zu dem Schluss, dass Augmented Reality den Mehrwert eines Visualisierungstools inne hält. AR kann eine intuitive Brücke zwischen virtueller und realer Welt schlagen und eine Webbasierte Sicht auf unsere Umwelt schaffen. Die intuitive Brücke kann soweit reichen, dass eine Erklärung der Handhabung nicht mehr benötigt wird und der Benutzer sich anhand des realen Kontextes durch virtuelle Daten navigiert. Dieser Kontext kann alle visuellen Merkmale der Welt betreffen. (Katier [2011])

Die genannte anhaltende Diskussion soll ein Schwerpunkt dieser Masterarbeit sein. Es soll betrachtet werden, ob ein Mehrwert durch PokerTool geschaffen werden kann und ob sich diese Erkenntnisse auf das gesamte Gebiet der Augmented Reality übertragen lassen.

4.3. Beliebte Augmented Reality Applikationen

Die beliebteste Augmented Reality Anwendung weltweit ist mit über 30 Millionen Downloads *Layar*. *Layar* ist dabei ein Marker-Erkenner und blendet, wie in Abbildung 4.2. gezeigt, die jeweiligen Informationen ein. Die Informationen stammen vom *Layar* Anbieter selbst oder von Drittanbietern, die *Layar* aufgrund der weiten Verbreitung als Wiedergabemedium nutzen. Es gibt zwei große Anwendungsgebiete von *Layar*: Zum Einen in den Printmedien, da die zu detektierenden Marker in Zeitschriften oder auf Plakaten zu finden sind. Durch dieses Konzept sind die anzuzeigenden Inhalte sehr verschieden, konzentrieren sich jedoch meist auf den Marketingbereich. Zum Anderen bietet *Layar* die sogenannten *Geo-Ebenen*. Hier zeigt die Kamera die Umgebung an und reichert diese mit Geoinformationen an. Diese Informationen werden meist als eine Art Kompass dargestellt und zeigen dem Benutzer den Weg zu einem ausgewählten Ziel. Die Ziele umfassen großteils gewerbliche Objekte wie Supermärkte, Kinos oder Restaurants. (Lay [2013])

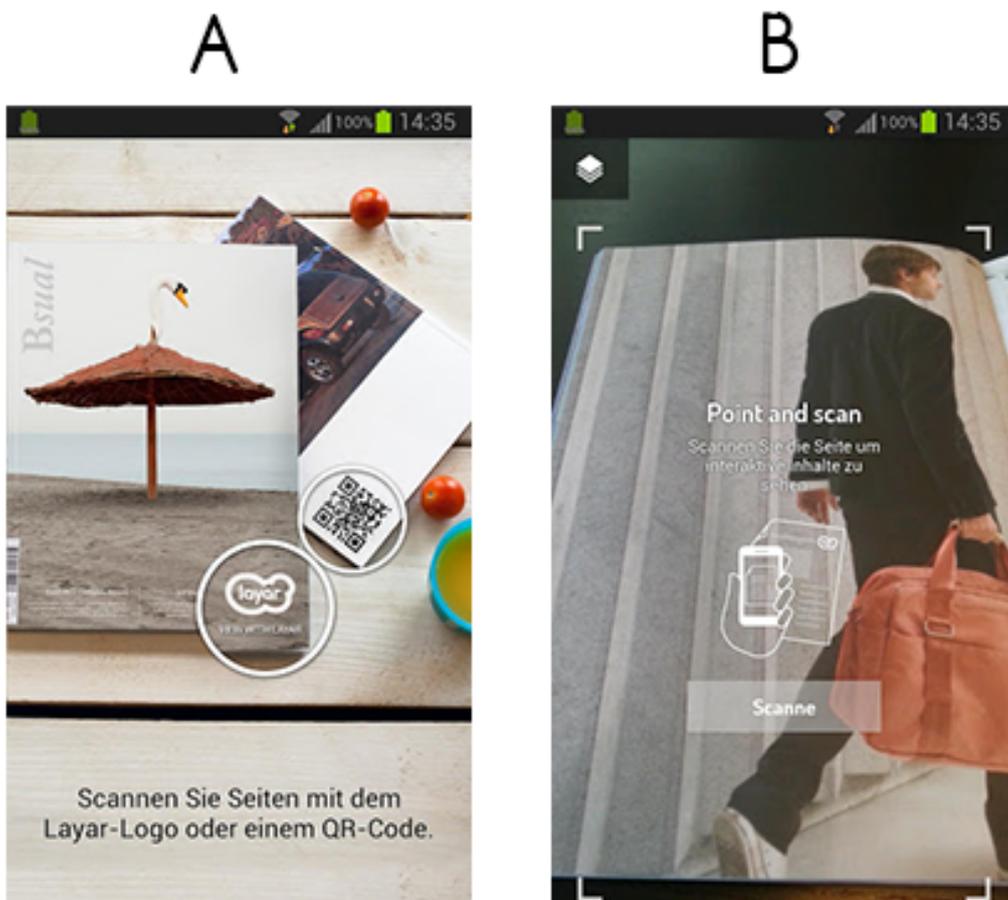


Abbildung 4.2.: Screenshots von Layar: Marker-Detektion in Printmedien

Eine weitere beliebte AR Anwendung ist mit mehr als drei Millionen Downloads der Augmented Reality Browser *Wikitude*. Wikitude konzentriert sich fast vollkommen, wie in Abbildung 4.3. zu sehen, auf die Anreicherung der Umwelt mit Geodaten. Kleine Nebenfunktionen beinhalten AR-Spiele, in der das Spiel über die Bewegungssensoren sowie der Kamera, den Spielern die Illusion suggeriert, über das Smartphone in der realen Welt mit digitalen Gegenständen zu interagieren. Die Hauptfunktion ist, wie im erweiterten Konzept von *Layar*, das Anzeigen von *Orten von Interesse* für den Benutzer. Die Orte sind hier zum Einen Sehenswürdigkeiten und zum Anderen gewerbliche Objekte wie Restaurants. (Wik [2013])



Abbildung 4.3.: Screenshots von Wikitude: (A) Anzeigen von Geodaten, (B) AR Spiel, (C) Währungsrechner

Beide Produkte bieten durch ihre offene Gestaltung Platz für innovative Ideen. Eine Idee z.B. von *Wikitude* besteht darin, fremdländisches Geld zu filmen und in eine gewünschte Währung umzurechnen. Diese neuen Konzepte lassen das Potential von Augmented Reality im Alltag erahnen und auf Weiterentwicklung hoffen.

4.4. Besondere Anforderungen durch Augmented Reality

Zusammenfassend besteht eine Augmented Reality Anwendung nur aus den drei in *Begriffserklärung und Definition* beschriebenen Charakteristiken: die Kombination der realen Welt mit virtuellen Inhalten, der Interaktion in Echtzeit und der abhängige 3D-Bezug

zwischen realer Welt und virtuellen Inhalten. Diese Definition schafft somit die für eine Smartphone Anwendung folgende Anforderungen an die Hardware und Software.

4.4.1. Hardware und Software

Im Kontext der Smartphone-Entwicklung bestehen besondere Anforderungen nur an eine Kamera, ein Display und eine ausreichende Rechenleistung, um in Echtzeit digitale Inhalte zu präsentieren. Diese drei grundlegenden Merkmale sind mit dem heutigen Stand der Technik bereits erreicht. Darüber hinaus zeigt nur die softwaretechnische Umsetzung die Anforderung an die Hardware auf. Durch Frameworks wie das *Metaio SDK* werden leicht erreichbare Hardwareanforderungen geschaffen. (metaio [2013])

Die besonderen Anforderungen an die Software mit eingebundenem Augmented Reality Teil sind ein interessantes Themengebiet. Bei der Umsetzung von AR Anwendungen stellt sich die Frage, ob durch die besonderen Merkmale auch eine besondere Konzeption notwendig ist. Nach OLIVEIRA liegen drei verschiedene Schnittstellen vor, wie AR eingebunden wird ([Allan Oliveira 2012]):



Abbildung 4.4.: Verschiedene Arten der Anzeige einer Augmented Reality Schnittstelle; A - Static Tracked; B - Dynamic Tracked; C - Dynamic with Vision (Allan Oliveira [2012])

- *Static Tracked*: Der digitale Inhalt besitzt eine feste Position und Größe im Verhältnis zum Benutzer. Ein Beispiel dafür könnte ein Kompass wie in Abbildung 4.4. sein, der dem Benutzer immer die gesuchte Richtung weist, jedoch seine Größe und Position im Verhältnis zum Benutzer nicht ändert.
- *Dynamic Tracked*: Der digitale Inhalt setzt die Position des Benutzers ins Verhältnis zu jedem Objekt von Interesse. Dieses Verfahren bringt virtuelle Inhalte in die reale Umgebung. Wie in B in Abbildung 4.4. könnte z.B. eine digitale Route in die reale Welt eingeblendet werden.

- *Dynamic with Vision*: Dynamische Inhalte werden, basierend aus der Bilderkennung mit angepassten Eigenschaften, in reale Welt hineingerechnet. Die angepassten Eigenschaften beruhen auf der erkannten Szenen. C in Abbildung 4.4. zeigt ein digitales Objekt, welches dynamisch im Bezug auf den erkannten Marker mit der Umwelt agiert.

Die unterschiedlichen Arten der Schnittstelle belegen, dass eine besondere Software-technische Umsetzung nicht notwendig ist. Obwohl neue Möglichkeiten durch eine AR Implementation geschaffen wurden, ergeben sich keine neuen Besonderheiten in der Umsetzung auf einem Monitor bzw. Smartphone-Display. Die bisherigen Grenzen, die im Kontext der Smartphone-Entwicklung entstanden sind, werden nicht angetastet. Die komplette Interaktion mit dem User findet wie gehabt auf einem 2-dimensionalen Bildschirm statt. Die Konzeption unterliegt weiterhin den gewohnten Styleguides der App-Entwicklung. Die Richtlinien der Schnittstelle bzw. der GUI sollen im Abschnitt 5: PokerTool erarbeitet werden.

5. PokerTool

Anhand der diskutierten Grundlagen soll im folgenden Kapitel die entstandene Applikation *PokerTool* im Detail erläutert werden. Aus diesem Grund soll zuerst die Vision und die Konzeption, dann die Besonderheit des Graphical User Interfaces und zum Schluss auf die Implementierung eingegangen werden.

5.1. Problemstellung

Wie schon im Abschnitt 2: *Poker Texas Holdem* ausführlich beschrieben, steht ein Pokerspieler in jeder Runde maximal vier Mal vor einer Entscheidung. Jede Entscheidung beruht auf der Einschätzung von neuen Informationen. Im Preflop evaluiert der Spieler seine beiden Karten, im Flop kommen die ersten drei Community-Cards dazu. Mit dem River und dem Turn erhält er jeweils noch eine Board-Karte und damit weitere Informationen. Jede Entscheidung, die der Benutzer für sich trifft, beruht auf diesen Informationen, sowie dem Handeln seiner Mitspieler. Die jeweiligen Aktionen der Mitspieler muss der Spieler deuten, indem er ein *Check*, *Bet*, *Call*, *Raise* oder *Fold* des Gegners als Aussage über die gegnerische Hand sieht. Diese Informationen haben viel mit Psychologie zu tun und kann nur anhand bekannter Karten für einen Computer nicht trivial ermittelt werden. Auf der anderen Seite kann der Computer anhand der bekannten Karten bestimmte Wahrscheinlichkeiten schnell aufstellen. Diese Wahrscheinlichkeiten sollen dem Benutzer bei seiner Entscheidung helfen.

Die Erfassung und Visualisierung der Daten ist hierbei der zweite Teil des Problems. Wie kann der Benutzer, ohne in seiner Spielweise eingeschränkt zu sein, die Informationen an ein Gerät übermitteln und anschließend aussagekräftige Daten zurückerhalten? Im Folgenden soll zuerst die Vision einer möglichen Anwendung beschrieben werden.

5.2. Vision

Der Grundgedanke war eine Augmented Reality Applikation zu entwickeln, die dem Benutzer in einer komfortablen Handhabung einen wirklichen Mehrwert bringt. *PokerTool* ist für diesen Zweck eine ideale Lösung. Durch die klar definierte Situation des Pokerspiels, einer intuitiven Steuerung und einem Informationsgewinn, der nur dynamisch und

von einem Computer hergestellt werden kann, können alle Vorteile einer AR Smartphone Anwendung ausgeschöpft und im Gegenzug die Nachteile minimiert werden.

Die Vision der Applikation war folgendermaßen definiert: Der Spieler erhält bei der Spielvariante Poker Texas Holdem seine Starthand. Er kann sich nicht entscheiden, ob er diese Karten spielen möchte. Er nimmt sein Smartphone und startet die Applikation. Er erfasst schnell seine Karten und das Smartphone liefert ihm umgehend die Wahrscheinlichkeit diese Runde zu gewinnen. Diese Entscheidungshilfe soll zu jedem Zeitpunkt des Spiels möglich sein und so viele Informationen wie möglich an den Benutzer liefern. Im Falle von *PokerTool* belaufen sich die Informationen auf:

- Rückmeldung, welche Karte einbezogen wurde
- Wahrscheinlichkeit zu gewinnen
- Wahrscheinlichkeit, den Pott zu splitten
- Wahrscheinlichkeit durch weitere mögliche Karten eine bessere Hand zu erreichen

Der Spieler erhält durch *PokerTool* mehr Sicherheit und kann seine erforderlichen Aktionen besser einschätzen. Die Benutzung darf den Spieler nicht stören und soll sein Spiel verbessern.

5.3. Konzeption

Um die Vision zu erfüllen, müssen alle Funktionalitäten sorgfältig abgewogen werden. Denn nicht jede Funktion ist in einer AR Anwendung förderlich für die Nutzbarkeit. Um diesen Umstand zu erfüllen, müssen im Vorfeld zuerst die wichtigsten Eigenschaften einer Smartphone Applikation genannt werden. Diese stimmen überein mit den Anforderungen der gängigen Styleguides für die Konzeption von Smartphone Apps wie von Apple bzw. Android [Alliance [2013]]. Mit besonderer Hinsicht auf den AR Anteil, die *PokerTool* erfüllen muss, sind die wichtigsten Eigenschaften:

- Mehrwert erzielen
- Einfachheit
- Hoher Grad an Automation in der Bedienung
- Intuitive Steuerung
- Feedback und Kommunikation
- Robustheit

Eine der elementarsten Fragen dieser Arbeit ist die Suche nach einem wirklichen Mehrwert durch eine AR Anwendung. Dieser Konzeptionspunkt wurde bereits bei der Grundidee verfolgt und darf bei der weiteren Definition einer sinnvollen Anwendung nicht in den Hintergrund treten. Um den Mehrwert herauszustellen, soll ein besonderes Augenmerk auf die Einfachheit gerichtet und keine unnötigen Features implementiert werden. PokerTool soll einen möglichst hohen Grad an Automatismen aufweisen, um dem Benutzer als nicht störender Assistent zur Seite zu stehen. Die Automatismen müssen aber durch Benutzer kontrollierbar sein, deswegen muss auf eine intuitive Steuerung geachtet werden. Dies setzt auch voraus, dass der Benutzer jederzeit ein Feedback erhält und eine daraus resultierende Kommunikation über das Geschehene entsteht.

5.3.1. Anforderungsdefinition

Durch Bestimmen der wichtigsten Eigenschaften, die die Konzeption erfüllen muss, lassen sich sinnvolle funktionale und nicht-funktionale Anforderungen ermitteln. Die zu erfüllenden Anforderungen auf funktionaler Ebene lauten demnach:

1. PokerTool muss ein komplettes Poker-Kartendeck detektieren können.
2. PokerTool muss aus der Detektion jeder Karte des Kartendecks den korrekten Wert extrahieren.
3. PokerTool muss bei falschem Extrahieren eine Karte neu detektieren.
4. PokerTool muss ein visuelles Feedback geben, ob eine Karte erkannt wurde.
5. PokerTool muss ein visuelles Feedback geben, welche Karte erkannt wurde.
6. PokerTool muss die Wahrscheinlichkeiten in Echtzeit ausrechnen.
7. PokerTool muss die Wahrscheinlichkeit, ob der Spieler gewinnt, ausgeben.
8. PokerTool soll die Wahrscheinlichkeit, ob der Spieler eine bessere Hand erreicht, ausgeben.
9. PokerTool muss das Ändern der Anzahl der Gegenspieler zulassen.
10. PokerTool muss für eine neue Runde alle Angaben auf einmal zurücksetzen können.
11. PokerTool muss alle Informationen verstecken können, um keinem Gegner aus Versehen einen Vorteil zu geben.
12. PokerTool soll Feedback geben, was mit der nächsten einzulesenden Karte geschieht.
13. PokerTool soll durch Gestensteuerung einfache Aufgaben, die sonst durch Knopfdruck realisiert werden müssen, übernehmen.

14. PokerTool muss beendet werden können.

Die nicht-funktionalen Anforderungen sind:

1. PokerTool soll mit möglichst wenig Touchinteraktion durch den Benutzer verwendet werden können.
2. PokerTool muss zu 99.9% der Zeit Absturz-frei laufen.
3. PokerTool muss genügend Platz für die Anzeige des Kamerabildes zulassen.
4. PokerTool muss ein aufgeräumtes Design besitzen.
5. PokerTool muss eine Einführung in die Programmfeatures geben.
6. PokerTool muss die Wahrscheinlichkeiten in einer angemessenen Schriftgröße anzeigen.
7. PokerTool muss alle dargestellten Informationen anzeigen, ohne dass der Hintergrund dies beschwert.
8. PokerTool muss große und eindeutige Symbole für die erkannten Karten anzeigen.
9. PokerTool soll ein eigenes Logo verwenden.
10. PokerTool soll ein schlichtes, aber ansprechendes Design erhalten.

5.3.2. Plattform

Als Plattform wurde das Android Betriebssystem gewählt. Aus Entwicklersicht ist die Wahl zwischen den beiden größten mobilen Betriebssystemen Apple IOS und Android gleichwertig. Beide Systeme bieten alle benötigten Bestandteile und Workarounds für eine erfolgreiche Lösung der Problemstellung. Jedoch muss sich die Konzeption durch Eigenarten der jeweiligen Plattform auf eine Plattform für die Prototypentwicklung entscheiden. Der weitere Verlauf zeigt, dass das entstandene Programm sich ohne Einschränkungen auf Apple IOS integrieren lassen würde.

5.4. Graphical User Interface

Ein wichtiger Teil der Konzeption liegt in der Gestaltung der Graphischen Schnittstelle mit dem Benutzer. Es ist die einzige visuelle Interaktionsmöglichkeit des Spielers mit der Anwendung. Durch automatisiertes Erkennen des Kamerabildes und automatisierte Visualisierungen, die auf der dynamischen Umgebung resultieren, stellt sich zuerst die Frage, ob Augmented Reality Applikationen einen besonderen Stellenwert bei *GUI* Implementierung besitzt. Wie bereits in Abschnitt 4: *Augmented Reality* beschrieben, ist

AR ein Visualisierungstool. Es arbeitet im Fall einer Smartphone-App auf dem Kamerabild. Alle Aktionen und Interaktionen werden durch das Display gestaltet. Daraus lässt sich schlussfolgern, dass einer AR Anwendung keine besondere Behandlung gegenüber einer herkömmlichen Applikation zuteil werden muss. Sie unterliegt den Grundsätzen der Dialoggestaltung und den Styleguides für Smartphone-Entwicklung. Im Folgenden soll zuerst auf die Norm EN ISO 9241-110, anschließend auf die Styleguides für Android-Entwicklung eingegangen werden. Abschließend soll eine Konzeption der graphischen Oberfläche für PokerTool vorgestellt werden.

5.4.1. Grundsätze der Dialoggestaltung

Die Grundsätze der Dialoggestaltung stellen den internationalen Standard EN ISO 9241-110 für Interaktion zwischen Mensch und Maschinen dar. Er beinhaltet sieben Gesetze die wenn möglich eingehalten werden sollen, um eine leichte und benutzerfreundliche Arbeitsumgebung zu erschaffen. Teilweise werden die sieben *Gesetze* bereits von den Vorgaben der gesamten Konzeption abgedeckt. Aus dieser Überschneidung lässt sich eine Hierarchie der Wichtigkeit der Grundsätze ableiten.

1. **Aufgabenangemessenheit** Der Dialog soll dem Aufgabenfeld angemessen sein. Es sollen keine unnötigen Extras oder Features implementiert werden. Das Programm soll dahingehend konzipiert sein, dass es die gestellten Probleme effizient und effektiv löst. (ISO 9241-10, S.4; §3.2)
2. **Selbstbeschreibungsfähigkeit** Jeder einzelne Dialog zwischen der Software und seinem Benutzer sollen ohne zusätzliche Hilfe selbsterklärend sein. (ISO 9241-10, S.5; §3.3)
3. **Erwartungskonformität** Die Aktion und Reaktionen des Programmes sollen sich mit den Erwartungen des Benutzers decken. Diese Erwartungen sollen auch teilweise durch Altwissen des Spielers erreicht werden. (ISO 9241-10, S.6; §3.5)
4. **Lernförderlichkeit** Dem Benutzer soll sich das Programm nach möglichst kurzer Einarbeitungszeit erschließen. Die selbst geschaffenen Strukturen sollen in allen Bereichen der Software konsistent sein. (ISO 9241-10, S.9; §3.8)
5. **Fehlertoleranz** Die Software muss mit Fehlern rechnen und umgehen können. Es muss für ihre Beseitigung eine Strategie ausgearbeitet haben und dem Benutzer gegebenenfalls anbieten. Darüberhinaus darf kein Fehler das gesamte Programm zum Absturz führen. (ISO 9241-10, S.7; §3.6)
6. **Steuerbarkeit** Der Benutzer hat jederzeit die Kontrolle über das Programm und steuert dies. Die Benutzergeschwindigkeit gibt der User vor. (ISO 9241-10, S.6; §3.4)

7. **Individualisierbarkeit** Der Benutzer kann das Programm nach seinen Vorstellungen an seine Bedürfnisse anpassen. Es sollen so viele Möglichkeiten zur Individualisierung vorhanden sein wie möglich. (ISO 9241-10, S.8; §3.7)

Für die Konzeption von PokerTool sind vor allem die ersten sechs Grundsätze zu beachten. Die *Aufgabenangemessenheit* geht einher mit dem konzeptionellen Grundsatz der Einfachheit. PokerTool soll sich auf seine Hauptaufgabe konzentrieren und zusätzliche Eigenschaften, die nicht relevant sind, aussparen. Auch die Lernförderlichkeit wird durch diese Vorgabe vereinfacht, da die Handlungsmöglichkeiten und eine Vielzahl an verschiedenen Features vermieden wird. Die *Steuerbarkeit* sowie die *Erwartungskonformität* und *Selbstbeschreibungsfähigkeit* sind in Anbetracht einer AR Anwendung, deren gesamter Dialog auf einem Hauptbildschirm geschieht, besonders hervorzuheben. Das zu konzipierende Programm muss, um einen erkennbaren Mehrwert zu erzielen, diese Grundsätze einhalten. Da es sich um eine Prototypische Umsetzung handelt, sollte der Punkt der *Individualisierbarkeit* der *Aufgabenangemessenheit* untergeordnet werden. Im Abschnitt 7: Aussichten soll auf Verbesserungsvorschläge eingegangen werden.

5.4.2. Android Design Principles and Styleguide

Zur erfolgreichen Entwicklung eines ansprechenden und selbsterklärenden Designs hat das *Android User Experience Team* einige Design Prinzipien in einem Styleguide zusammengefasst. Ihre Vorgaben sollen die Interessen der Benutzer widerspiegeln und verstehen sich als Ratschläge. Die für PokerTool wichtigsten Prinzipien sollen im folgenden Abschnitt vorgestellt werden [Alliance [2013]]:

- **Halte es kurz** Benutze kurze Sätze mit einfachen Wörtern. Die meisten Benutzer überspringen lange Sätze.
- **Erfreue mich in überraschender Weise** Eine schöne Oberfläche, eine bedächtig platzierte Animation oder ein zeitlich gut abgestimmter Soundeffekt ist eine Freude zu erleben. Unterschwellige Effekte verstärken das Gefühl ein mächtiges Werkzeug in den Händen zu halten.
- **Echte Objekte machen mehr Spaß als Knöpfe und Menüs** Erlaube es den Benutzern, Objekte anzufassen und direkt zu verändern. Es reduziert die Anstrengungen einer Aufgabe, wenn man den Benutzer emotional erfreut.
- **Bilder sind schneller als Worte** Achte darauf Bilder zu verwenden, um Ideen zu transportieren. Sie erregen Aufmerksamkeit und sind effizienter als Worte.
- **Entscheide für mich, aber lass mich das letzte Wort haben** Nimm eher möglichst viel, nach deinem besten Wissen und Gewissen, an als danach zu fragen. Zu viel Auswahl und Entscheidungen machen die Benutzer unglücklich. Aber wenn du

daneben liegst, erlaube deine falsche Annahme durch den Benutzer korrigieren zu lassen.

- **Zeige nur das, was ich benötigte, wenn ich es benötige** Benutzer werden überfordert, wenn sie zu viele Informationen bzw. Auswahlmöglichkeiten auf einmal sehen. Brich Aufgaben und Informationen in kleinere, verständliche Stücke. Verstecke Optionen, die zu einem bestimmten Zeitpunkt unwichtig sind und unterrichte die Benutzer, wie sie dorthin gelangen, wenn sie sie brauchen.
- **Ich sollte immer wissen, wo ich bin** Gib den Benutzern die Selbstsicherheit, dass sie wissen, wo sie sich im Programm befinden. Biete bei Aufgaben Rückmeldungen an.
- **Wenn es gleich aussieht, sollte es sich gleich verhalten** Hilf den Benutzern verschiedene Funktionen durch nicht nur subtil veränderte Bilder zu unterscheiden. Vermeide Situationen, in denen Orte ähnlich aussehen, aber sich trotz gleicher Eingabe verschieden verhalten.
- **Mach wichtige Dinge schnell** Nicht alle Aktion sind gleichwertig. Entscheide, welche die meist wichtigste für deine App ist und mach sie einfach auffindbar und schnell zu benutzen. Denke an den Auslöser einer Kamera oder den Pause-Knopf eines Music-Players.

5.4.3. PokerTool Design

Die vorangegangene ausführliche Einführung in Designprinzipien ermöglicht es, eine gut durchdachte Oberfläche zu schaffen. Wie bereits beschrieben, soll die Anwendung sich auf die Hauptaufgabe der Kartenerkennung konzentrieren. Daraus erschließt sich ein Hauptkamerafenster, welches den Großteil des Bildschirms einnimmt. Ein aufgeräumtes Design mit einer intuitiven Bedienung ist somit oberste Voraussetzung. Um ein universelles Verständnis zu erzeugen, soll die Prototypische Umsetzung in englischer Sprache geschehen. Die Angaben der Wahrscheinlichkeiten sollen leicht verständlich und einfach dargestellt werden. Durch die Touchscreenbedienung müssen die Tasten ausreichend groß, einfach benannt und selbsterklärend sein. Die automatisierte Kartenerkennung setzt einen Indikator voraus, der anzeigt, wo die nächste zu scannende Karte gespeichert wird. Um die Lernförderlichkeit zu erhöhen, sollte diese detektierte Karte jeweils für die eigene Hand des Benutzers, den Flop, den Turn und den River feste Positionen auf dem Bildschirm haben, auf die der Indikator zeigt. Um die wichtigen Informationen schnell zu extrahieren, muss der Hintergrund neutral und schlicht gehalten werden. Es stellen sich aus der Anforderungsanalyse drei Hauptknöpfe heraus. Der Beenden-, Neue-Runde- und Verstecken-Knopf. Der Beenden-Knopf soll auf Altwissen aufbauen und durch ein X

gekennzeichnet werden. Der Neue-Runde-Button soll ein schnelles Löschen aller detektierten Karten hervorrufen, um ohne Verzögerung eine neue Runde zu starten. Verkürzt und prägnant soll dieser Button nur mit *new* gekennzeichnet werden.

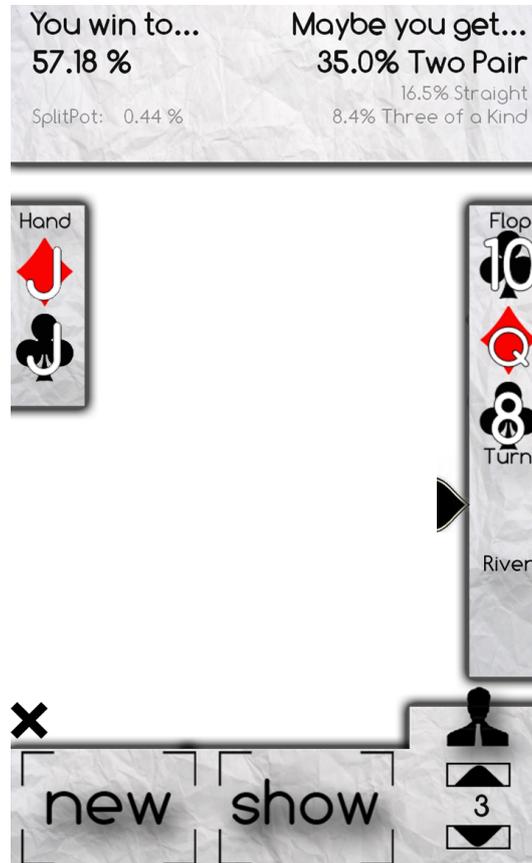


Abbildung 5.1.: PokerTool: Design des Hauptbildschirms

Der wichtigste Knopf ist der Verstecken-Knopf. Dieser soll verhindern, dass wenn das Smartphone auf dem Tisch abgelegt wird oder die Community Karten gefilmt werden, die bisher erzeugten (nur für den Benutzer bestimmten) Informationen nicht preisgegeben werden. Da ein Knopf, der design ist wie ein Lichtschalter und das Ein- und Ausschalten der Informationen per Knopfdruck veranlasst, vergessen werden könnte, ist die Konzeption ähnlich eines *Totmannschalters* vorgesehen. Nur dann, wenn der Knopf gedrückt wird, werden die Informationen angezeigt. Die Informationen sind hierbei die Wahrscheinlichkeiten und die bereits detektierten Karten. Die Funktion dieses Knopfes soll mit *show* benannt werden. Die Information, wie viele Mitspieler beteiligt sind, kann immer angezeigt und eingestellt werden. Das entstandene Design wird in Abbildung 5.1. gezeigt. Der weiße Hintergrund wird während der Anwendungsausführung mit dem Kamerabild gefüllt. In diesem Beispiel wurden bereits der Bube \diamond , der Bube \clubsuit , sowie die 10 \clubsuit , Dame \diamond und 8 \clubsuit erkannt. Aus dieser Konstellation und der Angabe von drei Ge-

genspielen unten rechts, konnten die oben angezeigten Wahrscheinlichkeiten bestimmt werden.

Eine weitere zu lösende Designfrage war die Anzeige der Kartenwerte. Diese sollten nicht nur als Werte in Textform angezeigt werden, sondern multimodal durch Text und Bild eine optimale kognitive Aufnahmefähigkeit bewerkstelligen. Die Farbe der Karte wird durch ihre tatsächliche Farbe repräsentiert. Der Wert als Text hinzugefügt. Einen Auszug der Kartensymbole ist in Abbildung 5.2. zu finden. In Abbildung 5.1. kann darüberhinaus auch die Einbettung in den Hauptbildschirm wahrgenommen werden. Die Einfachheit und eine schnelle menschliche Auffassung wurden hier als wichtigste Voraussetzung gewertet.

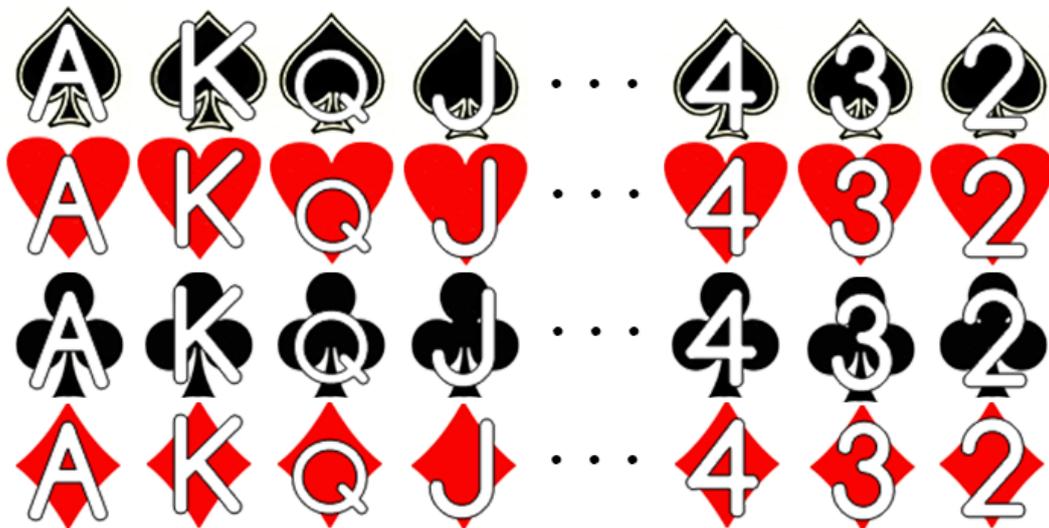


Abbildung 5.2.: PokerTool: Design der Kartenwert-Symbole

5.5. Implementierung

Im folgenden Abschnitt soll auf die Implementierung von PokerTool eingegangen werden. Zuerst wird ein Aktivitätsdiagramm in den Programmablauf einführen, bevor auf besondere Bausteine der Implementierung eingegangen wird.

5.5.1. Aktivitätsdiagramm

Die Vorgehensweise des Programmes kann anhand des Aktivitätsdiagrammes in Abbildung 5.3. nachvollzogen werden. Das Programm ist optimiert für seinen speziellen Einsatzort und liefert, je mehr Karten der Benutzer der Software zur Verfügung stellt, mehr bzw. genauere Informationen zurück.

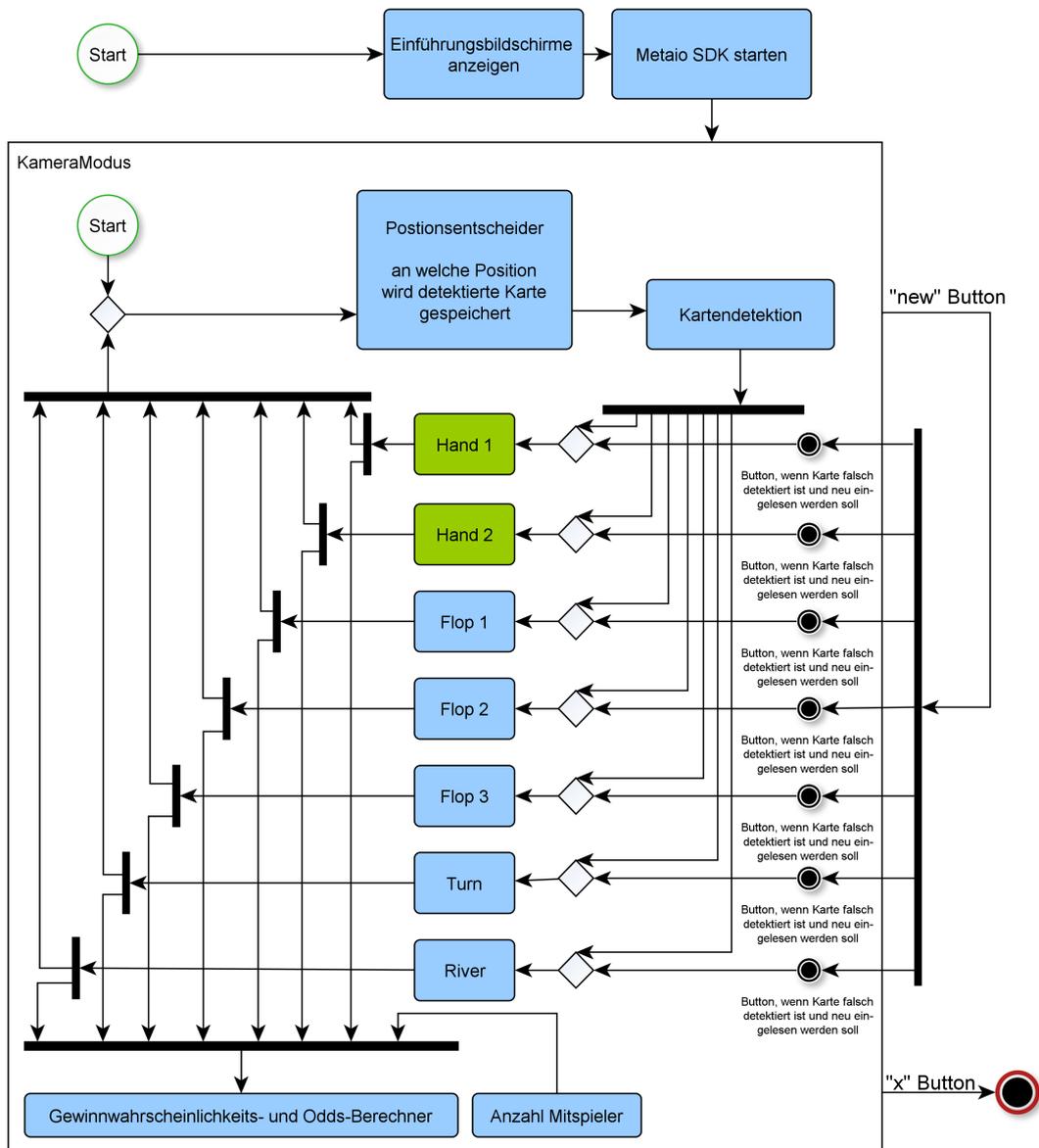


Abbildung 5.3.: PokerTool: Aktivitätsdiagramm des allgemeinen Programmablaufes

PokerTool startet mit einer Einführung, die dem Benutzer den Umgang mit der Anwendung erläutert. Im Anschluss startet das Metaio SDK, welches die Detektierung und weiteren Augmented Reality Bestandteile vorbereitet. Es folgt das eigentliche Programm im *KameraModus*, welches den eigentlichen Ablauf des Tools durchführt. Die fortlaufende Anwendung befindet sich durch eine Schleife in diesem Abschnitt bis das Programm beendet wird. Die Anwendung beginnt, indem es entscheidet, an welcher Position eine mögliche nächste detektierte Karte abgespeichert werden soll. In diesem Schritt setzt PokerTool auch das visuelle Feedback für den Benutzer, welches mit dem nächsten Speicherplatz übereinstimmt. Danach *wartet* die Software in der Kartendetektion auf eine zu identifizierende Karte. Wenn eine Karte erkannt wird, wird ihr Wert an den vorher bestimmten Slot weitergegeben und dort gespeichert. Ist eine Karte abgespeichert, erfolgen zwei Schritte zeitgleich:

1. Wahrscheinlichkeits- und Oddberechnung:

Zum Einen wird überprüft, ob eine Wahrscheinlichkeits- sowie eine Oddberechnung vorgenommen werden kann. Die Voraussetzung für die Wahrscheinlichkeitsrechnung ist, dass mindestens die eigenen Hole Cards abgespeichert sind. Um diese Berechnung so genau wie möglich zu machen, kann der Benutzer die Anzahl der Mitspieler eintragen.

Um die Odds zu ermitteln, muss darüber hinaus dem Programm der Flop bereits bekannt sein.

2. Positionsbestimmung

Zum Anderen wird wieder entschieden, an welcher Stelle eine mögliche nächste Karte gespeichert werden soll. Der Kreislauf beginnt erneut von vorne.

Die Positionsbestimmung ist wichtig, da zu jedem Zeitpunkt ein Abspeichern einer falschen Karte geschehen könnte. Diese Karte kann über Gestensteuerung oder Knopf Eingabe gelöscht werden. Die entstandene *Lücke* muss erkannt und der weitere Verlauf von PokerTool angepasst werden.

Alle Karten können auch auf einmal über die Betätigung des *new*-Buttons geleert werden. Dabei wird intern jeder einzelne Kartenspeicherort angesteuert und gelöscht.

Zu jedem Zeitpunkt kann darüber hinaus PokerTool via des Beenden-Buttons beendet werden. Der genaue Ablauf der jeweiligen Programmabschnitte soll in den folgenden Abschnitten erklärt werden.

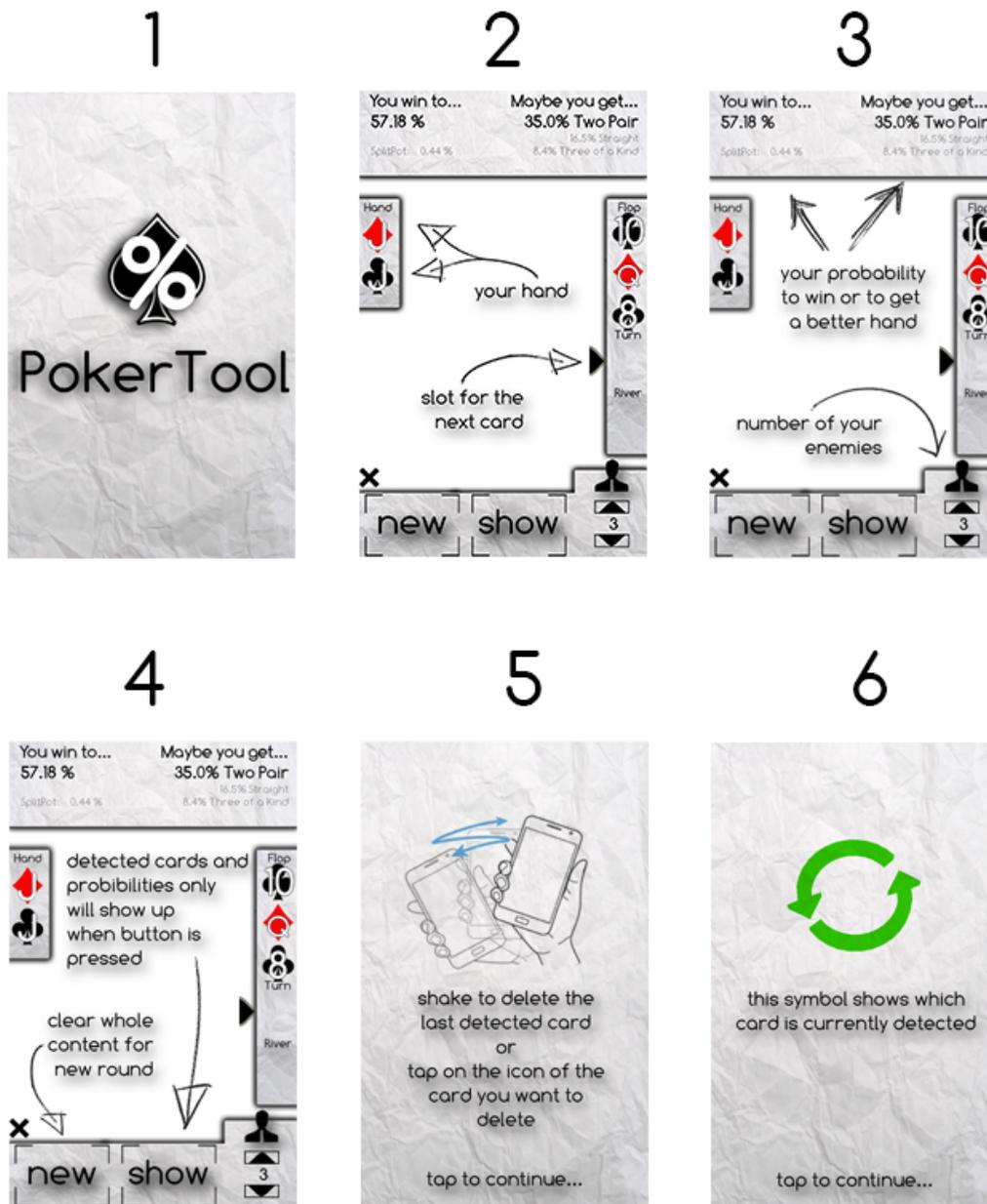


Abbildung 5.4.: PokerTool: Reihenfolge der Startbildschirme

5.5.2. Startbildschirme

Um die Anzeige und Steuerung von PokerTool schnell und einfach zu erlernen, werden beim Start die in Abbildung 5.4. aufgeführten Startbildschirme der Reihe nach gezeigt. Die Einführung läuft selbstständig ab, kann aber auch durch Berühren des Touchscreens übersprungen werden. Bildschirm zwei bis vier erläutern die Eigenschaften der graphischen Benutzerschnittstelle. Bildschirm fünf erklärt dem Benutzer, dass über eine *Shake*-Geste die letzte detektierte Karte gelöscht werden kann. Der letzte Bildschirm zeigt, dass eine aktuell detektierte Karte mit der AR-Visualisierung eines grünen kreisförmigen Pfeiles erkannt wird.

5.5.3. Laden des Contents

Für die richtige Verwendung des Metaio SDK ist der erste Schritt für ihre AR-Lösung das Laden des anzuzeigenden Contents und der Tracking Daten. Für jedes darzustellende Modell müssen, wie in Listing 5.1 gezeigt, die folgenden Schritte bewältigt werden:

Nach der Deklaration der vom Metaio SDK definierten *IGeometry*, wird in der Methode `loadContent` zuerst die Tracking Data in Form einer XML geladen und der Pfad bestimmt. Ab Zeile 18 wird zuerst der Pfad der Geometry zu den Assets definiert und anschließend wird das Model geladen und skaliert. Dieser Schritt muss im Falle von PokerTool für jede Karte einzeln vorgenommen werden, obwohl jede Karte das gleiche Modell erhält. Dies ist nötig, da jedem festgestellten Marker ein eigenes Modell zugewiesen werden muss.

```
1 // declare gobal IGeometry varibals
2 private IGeometry mAceSpade;
3 private IGeometry mAceHeart;
4 private IGeometry mAceDiamond;
5 private IGeometry mAceCross;
6
7 protected void loadContent()
8 {
9     try
10    {
11        // Load desired tracking data for planar marker tracking
12        trackingConfigFile = AssetsManager.getAssetPath("Assets1/TrackingData_MultiTargets.xml")
13        ;
14        MetaioDebug.log("Tracking Config path = "+trackingConfigFile);
15
16        boolean result = metaioSDK.setTrackingConfiguration(trackingConfigFile);
17        MetaioDebug.log("Markerless tracking data loaded: " + result);
18
19        // loading Path to each Model
20        String aceCrossModel = AssetsManager.getAssetPath("Assets1/check.zip");
21        String aceDiamondModel = AssetsManager.getAssetPath("Assets1/check.zip");
22        String aceHeartModel = AssetsManager.getAssetPath("Assets1/check.zip");
23        String aceSpadeModel = AssetsManager.getAssetPath("Assets1/check.zip");
24        ...
25
26        //create Geometry in Geometry-Container
27        mAceSpade = metaioSDK.createGeometry(aceSpadeModel);
28        mAceHeart = metaioSDK.createGeometry(aceHeartModel);
29        mAceDiamond = metaioSDK.createGeometry(aceDiamondModel);
30        mAceCross = metaioSDK.createGeometry(aceCrossModel);
31        ...
32
33        // Set geometry properties
34        mAceSpade.setScale(new Vector3d(100.0f, 100.0f, 100.0f));
35        mAceHeart.setScale(new Vector3d(100.0f, 100.0f, 100.0f));
36        mAceDiamond.setScale(new Vector3d(100.0f, 100.0f, 100.0f));
37        mAceCross.setScale(new Vector3d(100.0f, 100.0f, 100.0f));
38        ...
39    }
40 }
```

Listing 5.1: PokerTool: Ausschnitt aus der Methode loadContent der Java-Klasse CamMode

5.5.4. Kartendetektion und Augmented Reality Feedback

Die Kartendetektion und Visualisierung wird größtenteils abgenommen durch das Metaio Framework. Listing 5.2 zeigt, wie zuerst die erkannten Tracking-Daten in den *poses* gespeichert werden. Ist eine Pose erkannt, wird dieses Koordinatensystem mit einem Modell verknüpft und auf dem Display an der korrekten Position angezeigt.

```
1 public void onDrawFrame()
2 {
3     super.onDrawFrame();
4     if (metaioSDK != null)
5     {
6         // get all detected poses/targets
7         TrackingValuesVector poses = metaioSDK.getTrackingValues();
8         //if we have detected one, attach our metaio man to this coordinate system Id
9         if (poses.size() !=0){
10            // poses give all detected poses at the same time back
11            // it made a connection between the model and the coordinates
12            // if coordinates are detected, model will be shown
13            mAceSpade.setCoordinateSystemID(1);
14            mAceHeart.setCoordinateSystemID(2);
15            mAceDiamond.setCoordinateSystemID(3);
16            mAceCross.setCoordinateSystemID(4);
17            ...
18        }
19    }
}
```

Listing 5.2: *PokerTool*: Ausschnitt aus der Methode *onDrawFrame* zur Erkennung der Marker

Anschließend wird in Listing 5.3 die erkannte Karte für *PokerTool* identifiziert und an den richtigen Speicherort weitergeleitet. Bei diesem Vorgang wird auch das gesamte visuelle Feedback für den Benutzer sowie die Wahrscheinlichkeitsberechnung veranlasst. Die Einordnung in den Speicherort wird über eine *if*-Bedingung gelöst, da so auch mehrere erkannte Marker gleichzeitig erfasst werden können. Bei der prototypische Umsetzung von *PokerTool* wurde jedoch entschieden, nur eine Karte in einem Bild zu erkennen, um der Anwendung das höchste Maß an Robustheit, Effizienz und Schnelligkeit zukommen zu lassen. Die Methode *onDrawFrame* sucht in jedem gefilmten Frame nach einem Referenzbild in Form einer bekannten Karte. Die entstandene Augmented Reality Realisierung kann in Abbildung 5.5 betrachtet werden. Der AR-Bestandteil in *PokerTool* ist somit eine Verbindung aus *Static Tracked* und *Dynamic with Vision*. Zum Einen werden die Karten getrackt und erhalten eine dynamische Visualisierung, durch das animierte Symbol dass eine Karte erkannt wurde. Zum Anderen werden anhand der aufgenommenen Realität Informationen, in Form von Wahrscheinlichkeiten und Anzeigen der erkannten Karten, geschaffen. Die Berechnung der Gewinnchancen soll im nächsten Abschnitt erklärt werden.



Abbildung 5.5.: PokerTool: Augmented Reality Feedback und Wahrscheinlichkeitsrechnung

```

1   for (int i = 0; i < poses.size(); i++) {
2       // get the values of the detected card
3       curCard = deck.getDetectedCardFrom(poses.get(i).getCoordinateSystemID());
4       // save lastDetectedCard for shake gesture
5       lastDetectedCard=curCard;
6
7       // The decision which slot is choosen, is made through an
8       // if condition to set multiple Cards at the same time
9
10      // FIRST CARD
11      // hand.getCounter shows the place of the slot to set
12      // 0= first detected card, 10= first slot is free AND min one other slot is full
13      if ((hand.getCounter() == 0 && !hand.isTheCardAlreadyDetected(curCard)) |
14          (hand.getCounter() == 10 && !hand.isTheCardAlreadyDetected(curCard))
15          ) {
16          hand.setFirst(curCard);          // -> set current card to the the first slot
17          hand.setCounterToRightIndex();  // -> set next slot
18          setIndicator();                 // -> Indicator: which is the next slot
19          updateCardView(curCard, "FirstCard"); // -> show current detected card
20          updateProbabilty();            // -> compute probability
21      } else {
22          //SECOND CARD
23          if ((hand.getCounter() == 1 && !hand.isTheCardAlreadyDetected(curCard)) |
24              (hand.getCounter() ==11 && !hand.isTheCardAlreadyDetected(curCard))
25              ){
26              hand.setSec(curCard);
27              updateCardView(curCard, "SecondCard");
28              hand.setCounterToRightIndex();
29              setIndicator();
30              updateProbabilty();
31          }
32          ...

```

Listing 5.3: PokerTool: Ausschnitt aus der Methode onDrawFrame zur Einordnung in die korrekte Speicherstelle

5.5.5. Ermittlung der Wahrscheinlichkeiten

Die Ermittlung der Gewinnwahrscheinlichkeiten und der Odds ist das Kernstück von PokerTool. Abbildung 5.6 zeigt den Ablauf der Berechnung. Um diese Probleme zu lösen, muss zuerst bestimmt werden, in welcher Situation der Spieler ist. Es liegen verschiedene Berechnungsweisen vor, je nachdem ob der Spieler sich im Preflop, Flop, Turn oder River befindet. Die Preflop-Berechnung kann nur die allgemeinen Gewinnchancen angeben. Diese liegen, wie im Kapitel 2: Poker Texas Holdem bereits beschrieben, in Tabellen vor. Die empirischen Werte müssen nur anhand der Hole Cards des Spielers nachgeschlagen werden.

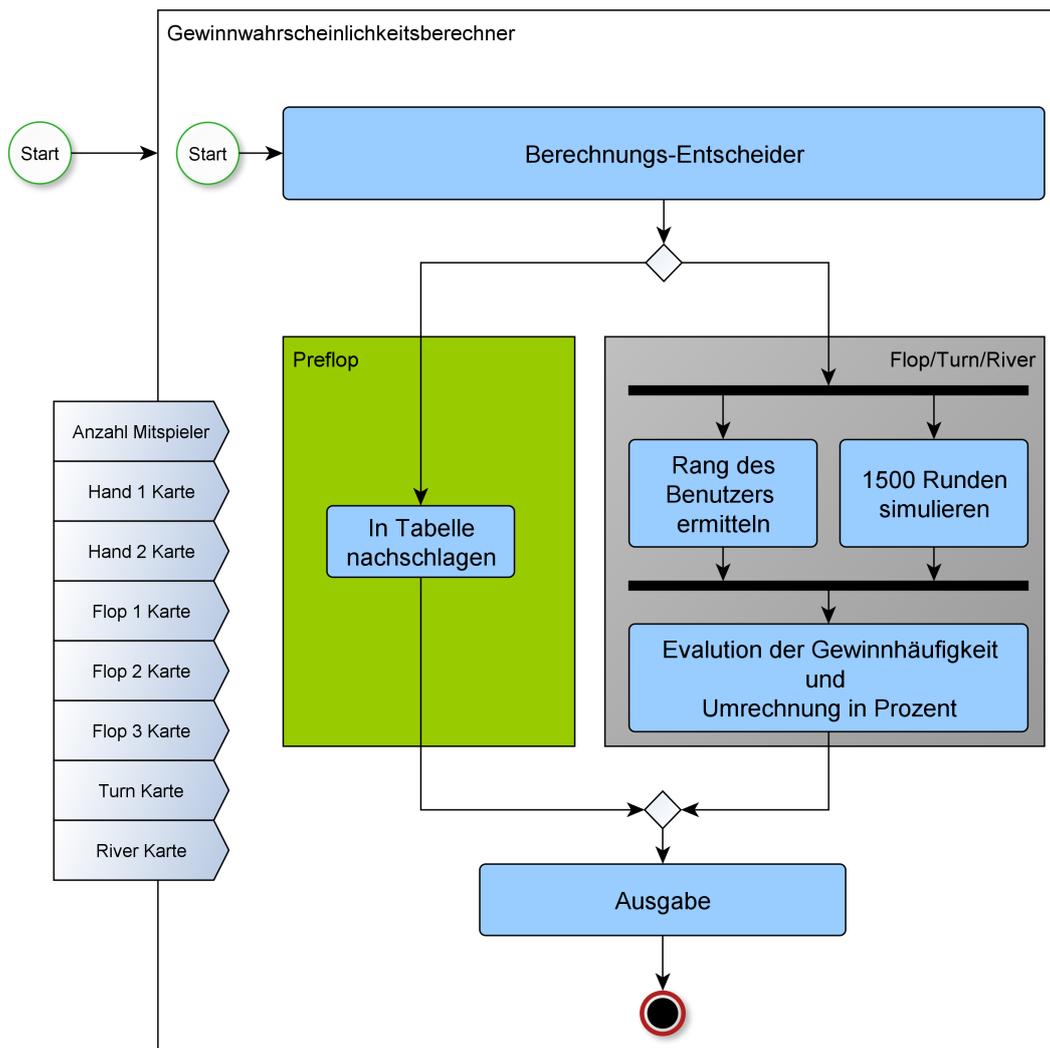


Abbildung 5.6.: PokerTool: Aktivitätsdiagramm zur Gewinnwahrscheinlichkeitsberechnung

Listing 5.4 und 5.5 zeigen die genauen Entscheidungswege von PokerTool. In Listing 5.4 kann die Voraussetzung für das Preflop Spiel und das Nachschlagen der empirischen Werte aus der Tabelle betrachtet werden. Nach dem Aufdecken der ersten Flop-Karte muss ein anderer Algorithmus verwendet werden, um die Chance auf einen Gewinn zu berechnen.

```
1 // Probability Preflop
2 // if rang is >2, then this slot is full
3 if( hand.handArray[0].getRang()>1 &&
4     hand.handArray[1].getRang()>1 &&
5     hand.handArray[2].getRang()<2 &&
6     hand.handArray[3].getRang()<2 &&
7     hand.handArray[4].getRang()<2 &&
8     hand.handArray[5].getRang()<2 &&
9     hand.handArray[6].getRang()<2
10    ){
11
12     // search in Hashtables
13     String key= hTableForProbs.translateToKey(hand.handArray[0], hand.handArray[1]);
14     probabilityToWinFloat=hTableForProbs.getProb(players, key);
15
16     //show messages
17     resultProb= ""+probabilityToWinFloat+" %";
18     betterHandText="Maybe you get...";
19     resultForBetterHand1= "waiting...";
20     resultForBetterHand2= "";
21     resultForBetterHand3= "";
22 }
```

Listing 5.4: PokerTool: Ausschnitt aus der Methode `updateProbability` für die Berechnung im Preflop

Ab dem Moment, in dem zusätzlich zu den Hole Cards, die drei Flop Karten bekannt sind, können zu den Gewinnwahrscheinlichkeiten auch die Odds bestimmt werden. Listing 5.5 zeigt, wie zuerst die Methode `handStrength` aufgerufen wird. Diese Methode berechnet die genaue Gewinnwahrscheinlichkeit anhand der gesichteten Karten. Ab diesem Zeitpunkt entsteht die Ermittlung aus einer Berechnung. Wie genau die Berechnung stattfindet, soll später mit der Erläuterung der Methode `handStrength` genauer vermittelt werden. `HandStrength` überprüft, selbstständig wie viele Karten für die Berechnung zur Verfügung stehen und muss deswegen in keine weiteren Bedingungen mehr eingeschlossen werden. Die Odds werden bestimmt, indem in der Methode `calcOdds` die Outs pro möglicher Hand gezählt werden. Dieser Wert wird dann in die Odds-Wahrscheinlichkeit umgerechnet. Zum Abschluss der Methode werden dann die korrekten Wahrscheinlichkeiten an die Anzeige übergeben.

```

1  //After Preflop
2  if( hand.handArray[0].getRang()>1 &&
3     hand.handArray[1].getRang()>1 &&
4     ( hand.handArray[2].getRang()>1 |
5     hand.handArray[3].getRang()>1 |
6     hand.handArray[4].getRang()>1 |
7     hand.handArray[5].getRang()>1 |
8     hand.handArray[6].getRang()>1 )
9  ){
10 // compute Probability with "handStrength"-algorithm
11 if(probabilityToWinFloat!=100){
12 handStrength();
13 }
14
15 //decision which way of Odd-Computation
16 //After Flop
17 if( hand.handArray[2].getRang()>1 &&
18     hand.handArray[3].getRang()>1 &&
19     hand.handArray[4].getRang()>1 &&
20     hand.handArray[5].getRang()<1 &&
21     hand.handArray[6].getRang()<1 ){
22 hand.calcOuts(hand, hTableForProbs, 1); // ->Odd computation
23 setTopOdds();
24 }
25 // After Turn
26 if( hand.handArray[2].getRang()>1 &&
27     hand.handArray[3].getRang()>1 &&
28     hand.handArray[4].getRang()>1 &&
29     hand.handArray[5].getRang()>1 &&
30     hand.handArray[6].getRang()<1 ){
31 hand.calcOuts(hand, hTableForProbs, 2); // ->Odd computation
32 setTopOdds();
33 }
34 //After River
35 if( hand.handArray[2].getRang()>1 &&
36     hand.handArray[3].getRang()>1 &&
37     hand.handArray[4].getRang()>1 &&
38     hand.handArray[5].getRang()>1 &&
39     hand.handArray[6].getRang()>1 ){
40 hand.myRankIs(); // -> no more Computation
41 betterHandText="Your Rank is...";
42 resultForBetterHand1=myRank.getName();
43 resultForBetterHand2="it will not getting";
44 resultForBetterHand3="better";
45 }
46 //set probabily values
47 resultProb= ""+probabilityToWinFloat+" %";
48 resultSplitPott= ""+probabilitySplitPott+" %";
49 }
50 else {
51     resultProb= "Show Cards!";
52 }
53 }
54 //show probabilities and the top 3 chance to reach a better hand
55 updateTextView(resultProb, "ProbabilityToWin");
56 updateTextView(betterHandText, "BetterHandTEXT");

```

```
57 updateTextView(resultForBetterHand1, "BetterHand1");  
58 updateTextView(resultForBetterHand2, "BetterHand2");  
59 updateTextView(resultForBetterHand3, "BetterHand3");  
60 updateTextView(resultSplitPott, "SplitPott");
```

Listing 5.5: PokerTool: Ausschnitt aus der Methode `updateProbability` für die Berechnung nach dem Preflop

Die genaue Berechnung der Gewinnchance wird aus einer dynamischen, empirischen Evaluation gezogen. Der `handStrength`-Algorithmus für diese Berechnung stammt von [Teofilo 2011] und berechnet die aktuelle Gewinnwahrscheinlichkeit. Es werden über 1000 Runden gegen die angegebene Anzahl von Mitspielern simuliert und daraus berechnet, wie oft der Benutzer gewonnen hätte. In Listing 5.6 wird der Ablauf der Berechnung nachgezeichnet. Zu Beginn werden die benötigten Variablen initiiert. Darunter auch die Anzahl der simulierten Runden. Je mehr Mitspieler an der zu testenden Runde teilnehmen, desto weniger Runden müssen durchgespielt werden. Dies beruht auf der Tatsache, dass nur

$$\binom{52}{2} = 1326 \text{ verschiedene, mögliche Starthände} \quad (5.1)$$

zur Verfügung stehen. Ein Durchlauf in der `while`-Schleife ab Zeile 20 repräsentiert eine mögliche Runde. Hierbei werden dem entsprechenden Mitspieler zufällig Karten zugewiesen und der beste Gegner dieser Runde wird verglichen mit der Hand des Benutzers. Dieses Verfahren evaluiert die Gewinnchance in dieser Situation. Da jedoch nur bei mehreren Mitspielern der Pool an möglichen Starthänden gleich bleibt, kann die Zahl der Durchläufe verringert werden und die Performance dadurch verbessert werden.

Der Algorithmus berechnet keine möglichen zukünftigen Runden. Diese Entscheidung liegt einem *Trade-Off* zu Grunde. Eine Einbeziehung würde dem Spieler zwar genauere Prozentangaben verschaffen, jedoch auch die Performance zu ihrem Nachteil beeinträchtigen. Die Entscheidung fiel zu Gunsten der Geschwindigkeit aus, da die Verbesserungen nur in Ausnahmefällen zu signifikanten Änderungen der Chancen führen.

```

1 public void handStrength(){
2
3     // reset variables
4     ahead=tied=behind=0;
5     int monteCarlo = 0;
6     float HowLong=(float) 1225.0-(enemies*30);
7
8     // estimate the remaining cards
9     remainingCard = (ArrayList<Card>) fullDeckList.clone();
10    for(int i=0; i<hand.handArray.length; i++){
11        remainingCard.remove(hand.handArray[i]);
12    }
13    // save remaining cards for a reset
14    remainingCardReset= (ArrayList<Card>)remainingCard.clone();
15
16    //estimate the rank of the user
17    myRank = hand.myRankIs();
18
19    // empiric evaluation of the winning probability
20    while(monteCarlo<HowLong){
21
22        // estimate the best rank of a random round
23        // with various players
24        Rank bestOppRank = new Rank();
25        bestOppRank.setRankID(0);
26
27        for(int i=0; i<enemies; i++){
28            possibleHand.setCardsToThisHand(hand);
29            fillOpp(possibleHand, remainingCard);
30            oppRank = possibleHand.oppRankIs();
31
32            //save the best possible hand of an enemy
33            // in this random round
34            if(oppRank.isBetterThan(bestOppRank)==1){
35                bestOppRank.setOne(oppRank.getOne());
36                bestOppRank.setTwo(oppRank.getTwo());
37                bestOppRank.setThree(oppRank.getThree());
38                bestOppRank.setFour(oppRank.getFour());
39                bestOppRank.setFive(oppRank.getFive());
40                bestOppRank.setRankID(oppRank.getRankID());
41            }
42        }
43
44        // evaluate if the user would win in this situation
45        int winner =myRank.isBetterThan(bestOppRank);
46        //store who wins
47        if(winner==1){
48            ahead++;
49        } else {
50            if (winner== 2) {
51                behind++;
52            } else
53                tied++;
54        }
55    }
56

```

```

57     monteCarlo++;
58     remainingCard= (ArrayList<Card>) remainingCardReset.clone();
59 }
60
61 // calculate handstrength
62 handStrength= (float)(ahead + tied)/ (ahead + tied + (float)(behind));
63 // calculate chance for SplitPott
64 probabilitySplitPott= (float)(tied)/(ahead + tied + behind);
65
66 // some adjustance
67 if(handStrength==probabilitySplitPott){handStrength=(float) 0.0;}
68 if((tied)==monteCarlo){handStrength=1;}
69
70 //transform in percentage
71 probabilitySplitPott=probabilitySplitPott*100;
72 probabilityToWinFloat= handStrength*100;
73
74 //if the user already was in the same situation
75 //the percentage get more accurate
76 String key= hTableForProbs.translateToKeyForBetterResults(enemies, hand.
77     savedSortedHandArray);
78 int newNormalized=1;
79 if( hTableForProbs.hForBetterResults.containsKey(key)){
80     int normalized =hTableForProbs.saved.get(key);
81     newNormalized = normalized+1;
82     probabilityToWinFloat=(float)( (probabilityToWinFloat+
83         (normalized*hTableForProbs.hForBetterResults.get(key)))
84         /newNormalized);
85 }
86 hTableForProbs.hForBetterResults.put(key, probabilityToWinFloat);
87 hTableForProbs.saved.put(key, newNormalized);
88 probabilityToWinFloat = (float) (((int) (probabilityToWinFloat * 100)) / 100.0);
89 probabilitySplitPott = (float) (((int) (probabilitySplitPott * 100)) / 100.0);
90 }

```

Listing 5.6: PokerTool: handStrenght-Algorithmus

5.5.6. Ermittlung des Ranges

Die Ermittlung des Ranges oder der Hand aus fünf, sechs oder sieben Karten ist nicht trivial. Eine Bestimmung der verschiedenen möglichen Hände kann durch eine Mischung aus generischen Formeln und Behandlung von Ausnahmeregeln gelöst werden. Listing 5.7 zeigt die Vorbereitungen, die getroffen werden müssen, um auf die einzelnen Ränge zu testen. Um den Rang zu bestimmen, benötigt *PokerTool* folgende Elemente:

- Einen Ausgabe Rang, der zu Beginn auf Null gesetzt wird und je nach Testergebnis den richtigen Wert zugewiesen bekommt.
- boolean-Variablen zum Test auf verschiedene Hände (Full House, Drilling,...)
- Alle erfassten Karten in einem sortierten Array dem Wert nach absteigend.

- Alle erfassten Karten in einem sortierten Array dem Wert nach absteigend, ohne das doppelte Vorkommen von Werten, jedoch mit dem Erhalt der dominierenden Farbe.
 - Beispiel: Erfasste Karten sind: Ass ♠, Ass ♥, Bube ♣ und 10 ♠.

Nach Eliminierung der doppelten Werte erhält man: Ass ♠, Bube ♣ und 10 ♠. Zu beachten ist hier, dass das Ass ♥ aus dem Array genommen wurde. Es wurde die Farbe erhalten, die am meisten vorkommt. Dies ist wichtig bei der Bestimmung von Flushs.

```

1 public Rank myRankIs(){
2     Rank rank= new Rank();
3     rank.setRankID(0);
4     boolean three= false;
5     boolean two = false;
6
7     //sort HandArray from high to low value: ace, king, queen, jack,..., 3, 2
8     Card[] sortedHandArray= handArray.clone();
9     sortiere(sortedHandArray);
10    savedSortedHandArray= sortedHandArray.clone();
11
12    // elimiate the double values in the sortedHandArray
13    // but the most occuring color will stay
14    // -> AceHeart,AceSpade, KingHeart,JackCross will be AceHeart,KingHeart,JackCross,
15    Card[] sortedHandArrayWithoutDoubles = sortedHandArray.clone();
16    elimiateDoubles(sortedHandArrayWithoutDoubles);
17    savedSortedHandArrayWithoutDoubles= sortedHandArrayWithoutDoubles.clone();
18    ...
19
20    //estimate rank
21 }

```

Listing 5.7: PokerTool: Ausschnitt aus Java-Klasse Hand: Rang bestimmen

Es wird beim Test der bestmöglichen Hand begonnen. Wenn die Bedingungen für sie nicht stimmen, wird der nächst schlechtere Rang getestet. Ist ein Rang positiv getestet, bricht die Überprüfung ab und gibt diesen Rang aus. Es sollen nun die Überprüfungen für die einzelnen Hände vorgestellt werden. Listing 5.8 beginnt mit der Überprüfung auf die bestmögliche Hand, ein Straight Flush. Bei diesem Test wird zuerst überprüft, ob in dem Array ohne doppelte Werte mindestens 5 Karten enthalten sind, ansonsten kann es zu keiner Straße oder Flush kommen. Die nächste Bedingung fragt ab, ob die Karten aufeinander folgen und die gleiche Farbe besitzen. Darüber hinaus muss noch eine besondere Abfrage erfolgen, da das Ass den Beginn einer Straße oder das Ende einer Straße markieren kann. Am Ende wird der Rang, wenn er positiv getestet wurde, korrekt abgespeichert und die komplette Überprüfung beendet.

```

1 // test the rank of the hand
2 //Straight Flush
3 if(sortedHandArrayWithoutDoubles[4].getRang()!=0){
4 for(int i=0; i<3; i++){
5     if( ( sortedHandArrayWithoutDoubles[i].getRang() ==
6         sortedHandArrayWithoutDoubles[(i+1)].getRang()+1 &&
7         sortedHandArrayWithoutDoubles[i+1].getRang() ==
8         sortedHandArrayWithoutDoubles[(i+2)].getRang()+1 &&
9         sortedHandArrayWithoutDoubles[i+2].getRang() ==
10        sortedHandArrayWithoutDoubles[(i+3)].getRang()+1 &&
11        sortedHandArrayWithoutDoubles[i+3].getRang() ==
12        sortedHandArrayWithoutDoubles[(i+4)].getRang()+1 &&
13
14        sortedHandArrayWithoutDoubles[i].getColor() ==
15        sortedHandArrayWithoutDoubles[(i+1)].getColor() &&
16        sortedHandArrayWithoutDoubles[i+1].getColor() ==
17        sortedHandArrayWithoutDoubles[(i+2)].getColor() &&
18        sortedHandArrayWithoutDoubles[i+2].getColor() ==
19        sortedHandArrayWithoutDoubles[(i+3)].getColor() &&
20        sortedHandArrayWithoutDoubles[i+3].getColor() ==
21        sortedHandArrayWithoutDoubles[(i+4)].getColor() &&
22        sortedHandArrayWithoutDoubles[i].getRang()!=0) |
23 //special case Ace low
24 ( sortedHandArrayWithoutDoubles[0].getRang() == 14 &&
25 sortedHandArrayWithoutDoubles[i].getRang() == 5 &&
26 sortedHandArrayWithoutDoubles[i].getRang() ==
27 sortedHandArrayWithoutDoubles[(i+1)].getRang()+1 &&
28 sortedHandArrayWithoutDoubles[i+1].getRang()==
29 sortedHandArrayWithoutDoubles[(i+2)].getRang()+1 &&
30 sortedHandArrayWithoutDoubles[i+2].getRang()==
31 sortedHandArrayWithoutDoubles[(i+3)].getRang()+1 &&
32
33 sortedHandArrayWithoutDoubles[0].getColor() ==
34 sortedHandArrayWithoutDoubles[i].getColor() &&
35 sortedHandArrayWithoutDoubles[i].getColor() ==
36 sortedHandArrayWithoutDoubles[(i+1)].getColor() &&
37 sortedHandArrayWithoutDoubles[i+1].getColor() ==
38 sortedHandArrayWithoutDoubles[(i+2)].getColor() &&
39 sortedHandArrayWithoutDoubles[i+2].getColor() ==
40 sortedHandArrayWithoutDoubles[(i+3)].getColor() )
41 ){
42 //set the actual rank with the best five cards
43 if (sortedHandArrayWithoutDoubles[i].getRang() == 5) {
44     rank.setOne(sortedHandArrayWithoutDoubles[i]);
45     rank.setTwo(sortedHandArrayWithoutDoubles[i + 1]);
46     rank.setThree(sortedHandArrayWithoutDoubles[i + 2]);
47     rank.setFour(sortedHandArrayWithoutDoubles[i + 3]);
48     rank.setFive(sortedHandArrayWithoutDoubles[0]);
49 } else {
50     rank.setOne(sortedHandArrayWithoutDoubles[i]);
51     rank.setTwo(sortedHandArrayWithoutDoubles[i + 1]);
52     rank.setThree(sortedHandArrayWithoutDoubles[i + 2]);
53     rank.setFour(sortedHandArrayWithoutDoubles[i + 3]);
54     rank.setFive(sortedHandArrayWithoutDoubles[i + 4]);
55 }
56 rank.setRankID(9); // 9 = straight flush

```

```

57     rank.setName("Straight Flush");
58     return rank; // end of the test
59 }
60 }
61 }

```

Listing 5.8: PokerTool: Ausschnitt aus Java-Klasse Hand: Straight Flush bestimmen

Der nächste Test in Listing 5.9, ist die Überprüfung auf einen Vierling. Hierbei dürfen, anders als bei der vorherigen Prüfung, nur maximal vier verschiedene Karten in dem Array ohne Doppelte vorliegen, da sonst nicht genug Karten gleicher Farbe aufgedeckt sein können. Anschließend wird nur überprüft, ob in dem normal sortierten Array mit Doppelten vier Mal der gleiche Wert vorkommt. Zum Schluss wird wie immer der Rang korrekt gesetzt und die Überprüfung abgebrochen. Dies soll nicht mehr in den weiteren Erläuterungen erklärt werden.

```

1 //Four of a Kind
2 if(sortedHandArrayWithoutDoubles[4].getRang()==0){
3     for(int x=0; x<4; x++){
4         if( sortedHandArray[x].getRang() == sortedHandArray[(x+1)].getRang() &&
5             sortedHandArray[x+1].getRang() == sortedHandArray[(x+2)].getRang() &&
6             sortedHandArray[x+2].getRang() == sortedHandArray[(x+3)].getRang() &&
7             sortedHandArray[x].getRang()!=0
8         ){
9
10            rank.setOne(sortedHandArray[x]);
11            rank.setTwo(sortedHandArray[x+1]);
12            rank.setThree(sortedHandArray[x+2]);
13            rank.setFour(sortedHandArray[x+3]);
14            // set Kicker
15            if(x==0) rank.setFive(sortedHandArray[x+4]);
16            else rank.setFive(sortedHandArray[0]);
17            rank.setRankID(8); // 8= four of a kind
18            rank.setName("Four of a Kind");
19            return rank;
20        }
21    }
22 }

```

Listing 5.9: PokerTool: Ausschnitt aus Java-Klasse Hand: Vierling bestimmen

Es folgt in Listing 5.10 der Test auf ein Full House. Für ein Full House muss ein Drilling und Pärchen vorliegen. Der Reihe nach wird dieses im Array mit möglichen vorkommenden Dopplungen geprüft. Wenn ein Drilling oder ein Pärchen vorliegt, wird dies in der Boolean-Variablen abgespeichert, auf die spätere Überprüfungen zurückgreifen können.

```
1 //Full House
2 for(int i=0; i<5; i++){
3     if( sortedHandArray[i].getRang() == sortedHandArray[(i+1)].getRang() &&
4         sortedHandArray[i+1].getRang() == sortedHandArray[(i+2)].getRang()&&
5         sortedHandArray[i].getRang() != 0 )
6     {
7         rank.setOne(sortedHandArray[i]);
8         rank.setTwo(sortedHandArray[i+1]);
9         rank.setThree(sortedHandArray[i+2]);
10        three =true;
11    }
12 }
13 for(int i=0; i<6; i++){
14     if( sortedHandArray[i].getRang() == sortedHandArray[(i+1)].getRang() &&
15         sortedHandArray[i].getRang() != rank.getOne().getRang() &&
16         sortedHandArray[i].getRang() != 0) {
17         rank.setFour(sortedHandArray[i]);
18         rank.setFive(sortedHandArray[i+1]);
19         two=true;
20     }
21 }
22 if(three && two){
23     rank.setRankID(7); //7= full house
24     rank.setName("Full House");
25     return rank;
26 }
```

Listing 5.10: PokerTool: Ausschnitt aus Java-Klasse Hand: Full House bestimmen

Der nächste Rang ist der Flush und wird in Listing 5.11 überprüft. Als erstes wird dabei getestet, ob eine Farbe im Array mindestens fünf Mal vorkommt. Ist dies der Fall, wird der korrekte Flush abgespeichert.

```

1 // Flush
2 if(sortedHandArrayWithoutDoubles[4].getRang()!=0){
3     int heart = 0; int spade = 0;
4     int cross = 0; int diamond = 0;
5     //which flush
6     int winnerWinnerChickenDinner=0;
7     for (int i = 0; i < sortedHandArray.length; i++) {
8         if (sortedHandArray[i].getColor() == 1)
9             spade++;
10        else {
11            if (sortedHandArray[i].getColor() == 2)
12                heart++;
13            else {
14                if (sortedHandArray[i].getColor() == 3)
15                    diamond++;
16                else {
17                    if (sortedHandArray[i].getColor() == 4)
18                        cross++;
19                }}}}
20    if (spade > 4)
21        winnerWinnerChickenDinner = 1;
22    else {
23        if (heart > 4)
24            winnerWinnerChickenDinner = 2;
25        else {
26            if (diamond > 4)
27                winnerWinnerChickenDinner = 3;
28            else {
29                if (cross > 4)
30                    winnerWinnerChickenDinner = 4;
31            }}}}
32    if (winnerWinnerChickenDinner > 0) {
33        // which Card in rank
34        int rankCard = 0;
35        for (int i = 0; i < sortedHandArray.length; i++) {
36            if (sortedHandArray[i].getColor() == winnerWinnerChickenDinner) {
37                rankCard++;
38                switch (rankCard) {
39                    case 1:
40                        rank.setOne(sortedHandArray[i]);
41                    case 2:
42                        rank.setTwo(sortedHandArray[i]);
43                    case 3:
44                        rank.setThree(sortedHandArray[i]);
45                    case 4:
46                        rank.setFour(sortedHandArray[i]);
47                    case 5:
48                        rank.setFive(sortedHandArray[i]);
49                }}}}
50        rank.setRankID(6); //6= flush
51        rank.setName("Flush");
52        return rank;
53    }}

```

Listing 5.11: PokerTool: Ausschnitt aus Java-Klasse Hand: Flush bestimmen

Nach dem Ausschließen eines Flushs wird als nächstes auf eine Straße geprüft. Listing 5.12 zeigt, dass der Test identisch zu dem Test einer Straight Flush ist, jedoch die Farbe der Karten egal ist. Zu diesem Zeitpunkt muss nicht mehr mit einem Straight Flush gerechnet werden, weil die dafür vorgesehene Abfrage negativ beantwortet wurde.

```

1  //Straight
2  if(sortedHandArrayWithoutDoubles[4].getRang()!=0){
3      for(int i=0; i<3; i++){
4          if( (sortedHandArrayWithoutDoubles[i].getRang() == sortedHandArrayWithoutDoubles[(i
5              +1)].getRang()+1 &&
6              sortedHandArrayWithoutDoubles[i+1].getRang() == sortedHandArrayWithoutDoubles[(i
7              +2)].getRang()+1 &&
8              sortedHandArrayWithoutDoubles[i+2].getRang() == sortedHandArrayWithoutDoubles[(i
9              +3)].getRang()+1 &&
10             sortedHandArrayWithoutDoubles[i+3].getRang() == sortedHandArrayWithoutDoubles[(i
11             +4)].getRang()+1 )|
12             //special case Straight with an Ace low
13             (sortedHandArrayWithoutDoubles[0].getRang() == 14 &&
14             sortedHandArrayWithoutDoubles[i].getRang() == 5 &&
15             sortedHandArrayWithoutDoubles[i].getRang() == sortedHandArrayWithoutDoubles[(i+1)
16             ].getRang()+1 &&
17             sortedHandArrayWithoutDoubles[i+1].getRang() == sortedHandArrayWithoutDoubles[(i
18             +2)].getRang()+1 &&
19             sortedHandArrayWithoutDoubles[i+2].getRang() == sortedHandArrayWithoutDoubles[(i
20             +3)].getRang()+1 )
21         ){
22             if(sortedHandArrayWithoutDoubles[i].getRang() == 5){
23                 rank.setOne(sortedHandArrayWithoutDoubles[i]);
24                 rank.setTwo(sortedHandArrayWithoutDoubles[i+1]);
25                 rank.setThree(sortedHandArrayWithoutDoubles[i+2]);
26                 rank.setFour(sortedHandArrayWithoutDoubles[i+3]);
27                 rank.setFive(sortedHandArrayWithoutDoubles[0]);
28             }
29             else{
30                 rank.setOne(sortedHandArrayWithoutDoubles[i]);
31                 rank.setTwo(sortedHandArrayWithoutDoubles[i+1]);
32                 rank.setThree(sortedHandArrayWithoutDoubles[i+2]);
33                 rank.setFour(sortedHandArrayWithoutDoubles[i+3]);
34                 rank.setFive(sortedHandArrayWithoutDoubles[i+4]);
35             }
36             rank.setRankID(5); // 5= straight
37             rank.setName("Straight");
38             return rank;
39         }
40     }
41 }

```

Listing 5.12: PokerTool: Ausschnitt aus Java-Klasse Hand: Straight bestimmen

Ob ein Drilling vorliegt, wird in Listing 5.13 erfragt. Als erstes kann auf die Überprüfung auf ein Full House zurückgegriffen werden. Liegt ein Drilling vor, muss nur noch dieser herausgesucht werden.

```
1 //Three of a Kind
2 if(three){
3   for(int i=0; i<5; i++){
4     if( sortedHandArray[i].getRang() == sortedHandArray[(i+1)].getRang() &&
5        sortedHandArray[i+1].getRang() == sortedHandArray[(i+2)].getRang() &&
6        sortedHandArray[i].getRang()!=0)
7     {
8       rank.setOne(sortedHandArray[i]);
9       rank.setTwo(sortedHandArray[i+1]);
10      rank.setThree(sortedHandArray[i+2]);
11      if(i==0){
12        rank.setFour(sortedHandArray[3]);
13        rank.setFive(sortedHandArray[4]);
14      }
15      if(i==1){
16        rank.setFour(sortedHandArray[0]);
17        rank.setFive(sortedHandArray[4]);
18      }
19      if(i==2){
20        rank.setFour(sortedHandArray[0]);
21        rank.setFive(sortedHandArray[1]);
22      }
23
24      rank.setRankID(4); // 4= three of a kind
25      rank.setName("Three of a Kind");
26      return rank;
27    }
28  }
29 }
```

Listing 5.13: PokerTool: Ausschnitt aus Java-Klasse Hand: Drilling bestimmen

Als vorletzter Test in Listing 5.14, wird nach zwei Pärchen gesucht. Das sortierte Array wird zuerst auf ein Pärchen getestet. Ist dies der Fall, wird in den restlichen Karten nach einem weiteren gesucht. Hier kann nicht auf die vorherige Detektion auf ein Pärchen im Full House zurückgegriffen werden, weil durch das weitere Durchsuchen der restlichen Karten die Prüfung noch einmal von Beginn starten muss.

```

1  //Two Pair
2  for (int i = 0; i < sortedHandArray.length-1; i++) {
3      if (sortedHandArray[i].getRang() == sortedHandArray[(i + 1)].getRang() &&
4          sortedHandArray[i].getRang() != 0) {
5          rank.setOne(sortedHandArray[i]);
6          rank.setTwo(sortedHandArray[i + 1]);
7
8          for (int j = i+1; j < sortedHandArray.length-1; j++) {
9              if (sortedHandArray[j].getRang() == sortedHandArray[(j + 1)].getRang() &&
10                 sortedHandArray[j].getRang() != 0) {
11                  rank.setThree(sortedHandArray[j]);
12                  rank.setFour(sortedHandArray[j + 1]);
13
14                  if (i == 0 && j == 2)
15                      rank.setFive(sortedHandArray[4]);
16                  if (i != 0)
17                      rank.setFive(sortedHandArray[0]);
18                  if (i == 0 && j != 2)
19                      rank.setFive(sortedHandArray[2]);
20
21                  rank.setRankID(3); //3= two pair
22                  rank.setName("Two Pair");
23                  return rank;
24              }
25          }
26      }}

```

Listing 5.14: PokerTool: Ausschnitt aus Java-Klasse Hand: Zwei Pärchen bestimmen

Als letzte Abfrage erscheint die Prüfung auf ein Paar. In Listing 5.15 wird gezeigt, dass zuerst auf ein Paar im Array getestet wird. Liegt dies vor, muss anhand der Wertigkeit die Lage im Array ermittelt werden. Die besten Beikarten bestimmen das endgültige Blatt der Hand.

Schlagen alle Überprüfungen fehl und die Methode ist noch nicht durch einen *return*-Aufruf beendet worden, kann lediglich eine High Card vorliegen. In diesem Fall repräsentiert das sortierte Array die Hand, die der Benutzer spielen muss.

```
1      //One Pair
2      for (int i = 0; i < savedSortedHandArray.length-1; i++) {
3          if (savedSortedHandArray[i].getRang() == savedSortedHandArray[(i + 1)].getRang() &&
4              savedSortedHandArray[i].getRang() != 0) {
5              rank.setOne(savedSortedHandArray[i]);
6              rank.setTwo(savedSortedHandArray[i + 1]);
7              switch(i){
8                  case 0:
9                      rank.setThree(savedSortedHandArray[2]);
10                     rank.setFour(savedSortedHandArray[3]);
11                     rank.setFive(savedSortedHandArray[4]);
12                     rank.setRankID(2); // 2= one pair
13                     rank.setName("One Pair");
14                     return rank;
15                 case 1:
16                     rank.setThree(savedSortedHandArray[0]);
17                     rank.setFour(savedSortedHandArray[3]);
18                     rank.setFive(savedSortedHandArray[4]);
19                     rank.setRankID(2);
20                     rank.setName("One Pair");
21                     return rank;
22                 case 2:
23                     rank.setThree(savedSortedHandArray[0]);
24                     rank.setFour(savedSortedHandArray[1]);
25                     rank.setFive(savedSortedHandArray[4]);
26                     rank.setRankID(2);
27                     rank.setName("One Pair");
28                     return rank;
29                 case 3: case 4: case 5:
30                     rank.setThree(savedSortedHandArray[0]);
31                     rank.setFour(savedSortedHandArray[1]);
32                     rank.setFive(savedSortedHandArray[2]);
33                     rank.setRankID(2);
34                     rank.setName("One Pair");
35                     return rank;
36             }
37         }
38     }
39     // High Card
40     rank.setOne(sortedHandArray[0]);
41     rank.setTwo(sortedHandArray[1]);
42     rank.setThree(sortedHandArray[2]);
43     rank.setFour(sortedHandArray[3]);
44     rank.setFive(sortedHandArray[4]);
45     rank.setRankID(1); // 1= high card
46     rank.setName("High Card");
47     return rank;
```

Listing 5.15: PokerTool: Ausschnitt aus Java-Klasse Hand: Pärchen bestimmen oder niedriger

5.5.7. Gestensteuerung

Um dem Benutzer die größtmögliche Freiheit bei der Eingabe zu verschaffen und die Bedienung am einfachsten zu gestalten, sollten Bedienelemente über Gestensteuerung möglich sein. Während der Konzeptions- und Implementierungsphase konnte ein Szenario festgestellt werden, für das die Gestensteuerung ideale Voraussetzungen bietet. Bei der Detektion der Spielkarten kann es zu *Falschdetektionen* kommen. Diese *falschen Karten* werden durch den Benutzer erkannt und aus der Anwendung via Knopfdruck (auf das Feld der falschen Karte) wieder herausgenommen. Dieser Vorgang wurde als umständlich empfunden. Die Lösung ist eine Schüttelgeste, die die letzte detektierte Karte löscht. Wie in Abbildung 5.7. zu sehen, kann aber auch jede andere Karte über dieses Schütteln gelöscht werden. Das häufigste, vorkommende Szenario, auf die der Algorithmus konzipiert wurde, ist eine Falsch-Detektion, das *sofortige Erkennen* durch den Benutzer und seinen *sofortigen Wunsch auf Korrektur*. Der Wunsch der direkten Korrektur kann dann via Schütteln vorgenommen werden. Wenn diese Karte gelöscht wurde, wird durch weiteres Schütteln die davor zuletzt eingescannte Karte gelöscht und so weiter. Bei jedem Löschen werden auch die Wahrscheinlichkeiten korrigiert.

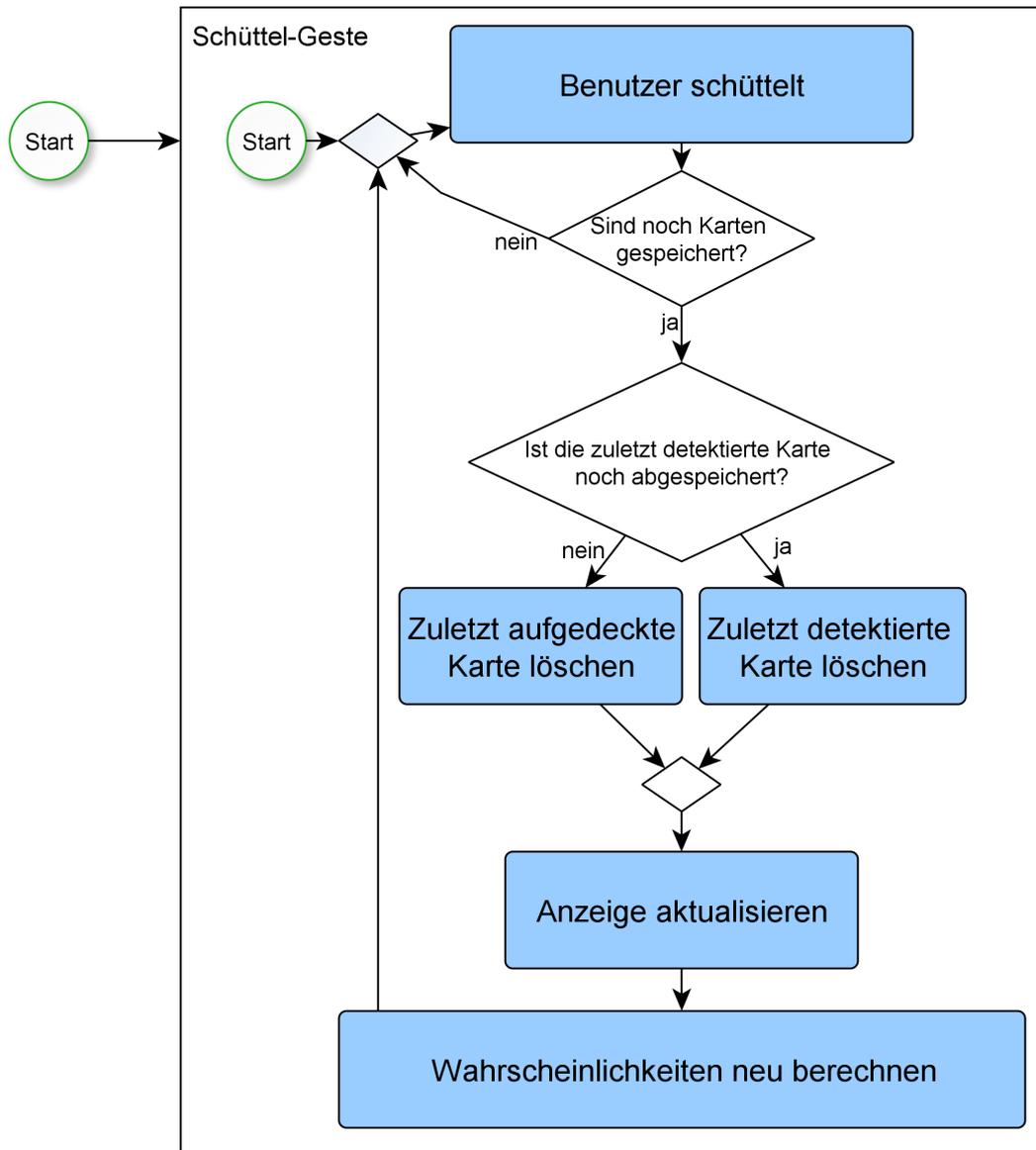


Abbildung 5.7.: PokerTool: Aktivitätsdiagramm zur Schüttelgeste

6. Evaluation

Dieses Kapitel befasst sich mit der Evaluation des entstandenen Prototyps von *PokerTool*. Es soll zuerst die aufgestellte These hinter der Evaluation vorgestellt werden. Anschließend sollen der Ablauf sowie die Ergebnisse aufgezeigt werden und eine Bewertung stattfinden.

6.1. These

PokerTool ist ein Assistenztool mit Augmented Reality Bestandteil. Es wurde ausführlich auf die Konzeptions- und Entwicklungsentscheidungen eingegangen, um die folgende These zu überprüfen.

Durch gut durchdachte Augmented Reality Smartphone Anwendungen erfährt der Benutzer einen Mehrwert, jedoch ist dieser Mehrwert nicht groß genug, um den Nachteil der unpraktischen Handhabung zu egalisieren, der durch das Trägermedium Smartphone entsteht. Je weniger der Benutzer mit einem AR Gerät direkt interagieren muss, desto mehr überwiegen die Vorteile.

AR ist, wie BACHFISCHER (Bachfischer [2011]) bereits angemerkt hat, eine neue Art der Visualisierung und kann den Benutzer bei einer guten Konzeption sinnvoll unterstützen. Das Diskutieren der bestehenden Styleguides sowie der Gestaltungsgrundsätze EN ISO 9241-110, sollte die Voraussetzung für ein gutes *Fundament* von *PokerTool* schaffen, um die genannte These zu überprüfen. Die Evaluation kann jedoch nur Aussagen über *PokerTool* im speziellen und den Gedanken bzw. die Theorie hinter AR beleuchten. Es ist schwer, eine allgemeingültige Aussage zu schaffen, da die Theorie nur anhand *PokerTool* getestet wurde. Auf der anderen Seite können jedoch Prognosen aufgestellt werden, die den richtigen Weg aufzeigen.

6.2. Aufbau

PokerTool wurde unter realen Bedingungen einer Pokerrunde evaluiert. Die Benutzung von Hilfsmitteln ist normalerweise verboten, jedoch ist die prototypische Umsetzung

geschaffen worden, um das AR Erlebnis zu testen. Der verbotene Anwendungsfall kann vernachlässigt werden, um die These zu diskutieren.

Die Evaluationsgruppe bestand aus sieben Personen. Die Größe ist mit einer normal stattfindenden Pokerrunde übereinstimmend. Gespielt wurden zwei Turniere. Der Turniermodus besagt, dass jeder Spieler mit der gleichen Anzahl an Chips bzw. Geld startet und der Gewinner derjenige ist, der zum Schluss das gesamte im Umlauf befindliche Geld erspielt hat. Hat eine Person kein Geld mehr, scheidet sie aus dem Turnier aus.

Sechs Spieler besaßen eine installierte Version von *PokerTool* auf ihrem Smartphone. Nach einer anonymen Befragung vor der ersten Verwendung der Software, erfolgte eine ausführliche Einführung in die Anwendung. Nach einer Eingewöhnungszeit startete das erste Turnier. Alle Spieler waren angewiesen, *PokerTool* für eine halbe Stunde bei jeder Handlung zu nutzen. Nach dieser Zeit war es den Probanden freigestellt, das Tool weiter zu nutzen. Jeder durfte für sich entscheiden, ob er die Hilfe von *PokerTool* annehmen möchte. Nach dem Beenden der ersten Spielrunde, wurde mit den gleichen Regeln und Startvoraussetzungen ein zweites Turnier gestartet. Der einzige Unterschied war, dass die Spieler direkt von Beginn selbstverantwortlich über die Nutzung von *PokerTool* entscheiden durften.

Die gesamte Nutzung des Tools wurde ohne Wissen der Spieler in einem Log auf dem jeweiligen Smartphone gespeichert. Durch ein Nicht-Wissen der gespeicherten Nutzungsdaten konnte ein natürliches Spielverhalten vorausgesetzt werden. Geloggt wurde die Zeit, in der die Spieler *PokerTool* wirklich genutzt haben. Dies wurde gemessen über die Verwendung des *Show* Buttons. Nur wenn dieser gedrückt ist, erlangt der Benutzer die gewünschten Informationen.

Im Anschluss an die zweite Runde wurde allen Spieler ein zweiter Fragebogen vorgelegt, in dem die Erfahrungen der Verwendung des Assistenztools einfließen.

6.3. Ergebnisse

Die Ergebnisse der Evaluation ergeben sich aus zwei unterschiedlichen Arten der Überprüfung. Zuerst soll auf die direkte Befragung der Testpersonen eingegangen werden. Im zweiten Teil wird das Hauptaugenmerk auf die unbemerkt gespeicherten Log-Daten gelegt.

6.3.1. Fragebogen

Die homogene Gruppe bestand aus sieben männlichen Probanden im Alter von 24 bis 26 Jahren. Es wurde in den Fragebögen vor und nach der Verwendung von PokerTool eine Skala mit sieben Antwortmöglichkeiten verwendet:

- stimme zu (1)
- stimme eher zu (2)
- stimme ein wenig zu (3)
- neutral (4)
- stimme ein wenig nicht zu (5)
- stimme eher nicht zu (6)
- stimme nicht zu (7)

Vor der Benutzung der Anwendung wurden Fragen über die Erfahrung mit dem Spiel Poker und der Nutzung von Smartphones gestellt. Die Verteilung der Antworten kann in Abbildung 6.1. gesehen werden. Ein heterogenes Auftreten besaß die Gruppe in der eigenen Einschätzung der Pokererfahrung. Zwei Personen gaben an, sehr viel Erfahrung zu besitzen, eine Person stufte sich eher ein, gar keine zu besitzen. Jedoch stufen sich alle Testpersonen mit einer geringen Varianz von 1.06 im Median mit viel Erfahrung in der Smartphonebedienung ein.

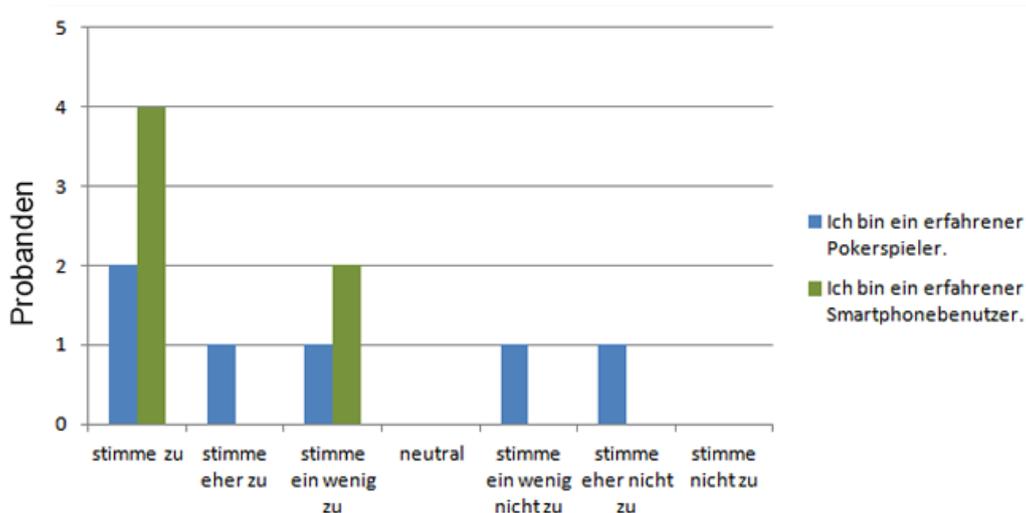


Abbildung 6.1.: Evaluationsergebnisse: Vor Benutzung von PokerTool über Erfahrung mit Poker und Smartphone-Nutzung

Eine interessante Gegenüberstellung ergab die Einstellung zu Augmented Reality. AR wurde von den befragten Personen mit einem Median von 2 als sinnvoll empfunden und

von den Personen mit Mehrwert wahrgenommen. In Abbildung 6.2. kann jedoch auch eine geschlossene Erfahrung zur Häufigkeit in der Verwendung von AR Anwendungen erkannt werden. Die Probanden gaben im Median von 5,5 mit einer Varianz von 0,3 an, eher gar keine Applikationen mit AR-Bestandteil zu nutzen. Die Gründe hierfür waren:

- es gibt wenig Angebot
- die meisten AR sind unpraktisch
- es gibt keine sinnvollen Angebote

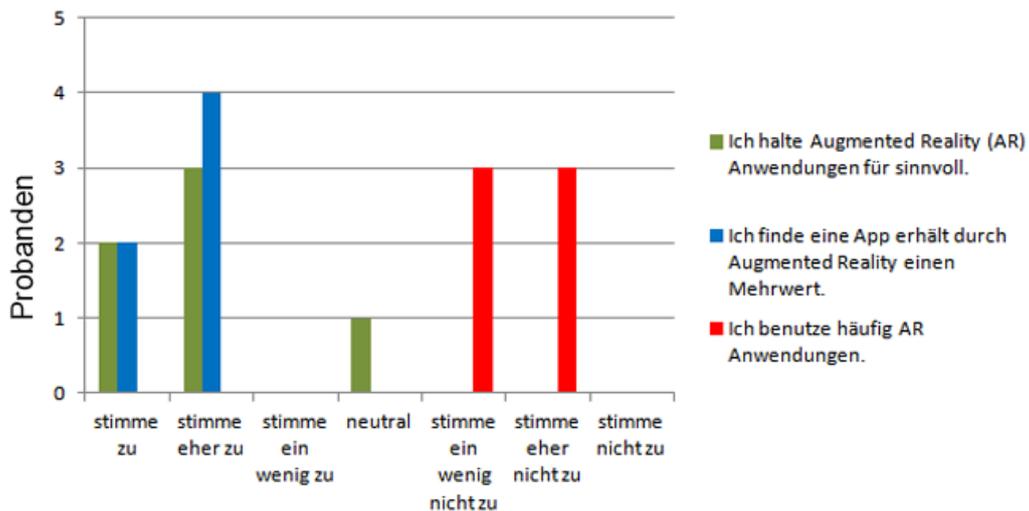


Abbildung 6.2.: Evaluationsergebnisse: Vor Benutzung von PokerTool über Einstellung zu AR

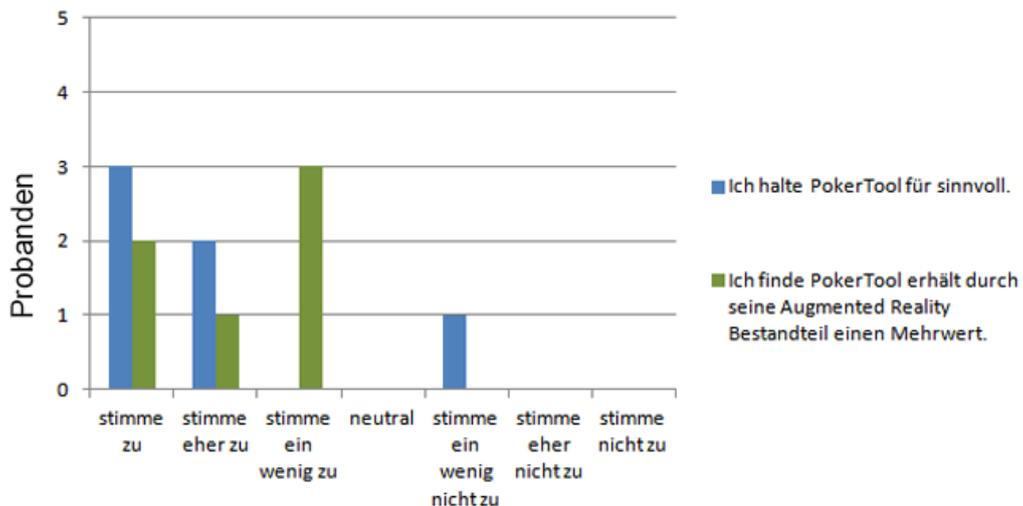


Abbildung 6.3.: Evaluationsergebnisse: Einstellung zu PokerTool vor der Benutzung nach der Sinnhaftigkeit (1) und Mehrwert (2)

Nach der Einführung in das Konzept von PokerTool wurde nach der Einstellung gegenüber dem zu testenden Programm gefragt. Wie Abbildung 6.3. zeigt, wurde die Frage nach der Sinnhaftigkeit und dem zu erwartenden Mehrwert von PokerTool gestellt. Mit einem Median von 1,5 empfand man die Software als sinnvoll und nahm mit einem Median von 2,5 auch an, dass diese AR Anwendung einen Mehrwert besitzt. Die wichtigsten Eigenschaften die PokerTool besitzen sollte, sind laut den Probanden eine *schnelle, verzögerungsfreie, einfache* und *intuitive* Bedienung. Darüber hinaus soll es *robust* und *korrekt* die Wahrscheinlichkeiten ermitteln.

Nach der Nutzung gaben die Testpersonen an, wie in Abbildung 6.4. zu sehen, ein positives Bild von der Konzeption von PokerTool zu besitzen. Mit einem Median von 2 empfanden die Probanden die Steuerung intuitiv sowie das Design für gelungen. Noch besser schnitt nur die Augmented Reality Feedbackanzeige ab. Bis auf eine Person sagten alle Spieler, dass sie die Anzeige und somit den AR-Bestandteil als sinnvoll hielten.

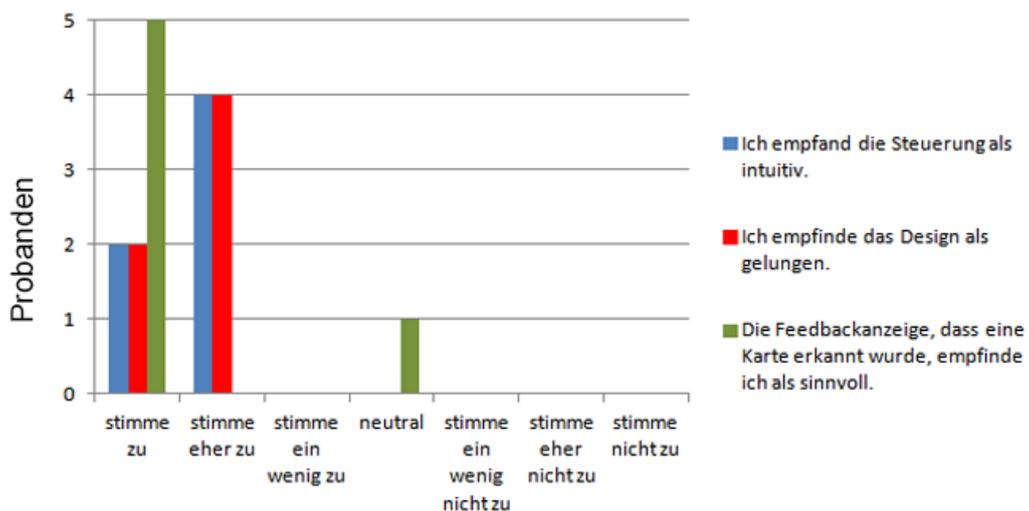


Abbildung 6.4.: Evaluationsergebnisse: Einstellung zur Benutzung und Design von PokerTool

Die Informationen, die den Spielern vermittelt wurden, gaben ihnen im Median von 3 jeweils eher Sicherheit und halfen bei der Entscheidungsfindung. Ein Grund für diese Aussage war, dass die Benutzer noch lernen mussten, die Prozentangaben einzuordnen. Die Verteilung kann in Abbildung 6.5. betrachtet werden.

Auf die Frage, ob die Spieler *PokerTool* als gelungene AR Anwendung wahrnehmen, ist die Meinung mit einem Median von 2 und einer geringen Varianz von 0,3 einheitlich positiv. Im Gegensatz dazu steht, dass der Median bei der Frage, ob die Personen die Anwendung wiederverwenden würden, bei 3 mit einer Varianz von 1,7 liegt. Abbildung 6.6. zeigt, dass die Personen einen Mehrwert sehen und definieren *PokerTool* als gelungen, würden diese App jedoch nur vielleicht wiederverwenden. Ein oft genannter Grund war,

dass die Handhabung zu unpraktisch ist, obwohl die Meinung bestand, dass *PokerTool* für die komplexe Aufgabenstellung Poker eine gelungene Applikation darstellt.

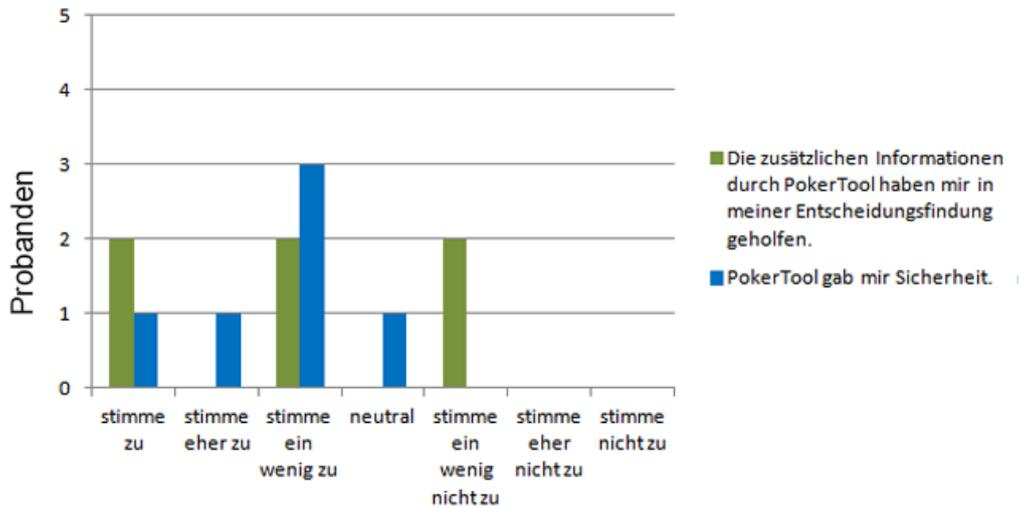


Abbildung 6.5.: Evaluationsergebnisse: Einstellung zur Hilfestellung durch PokerTool

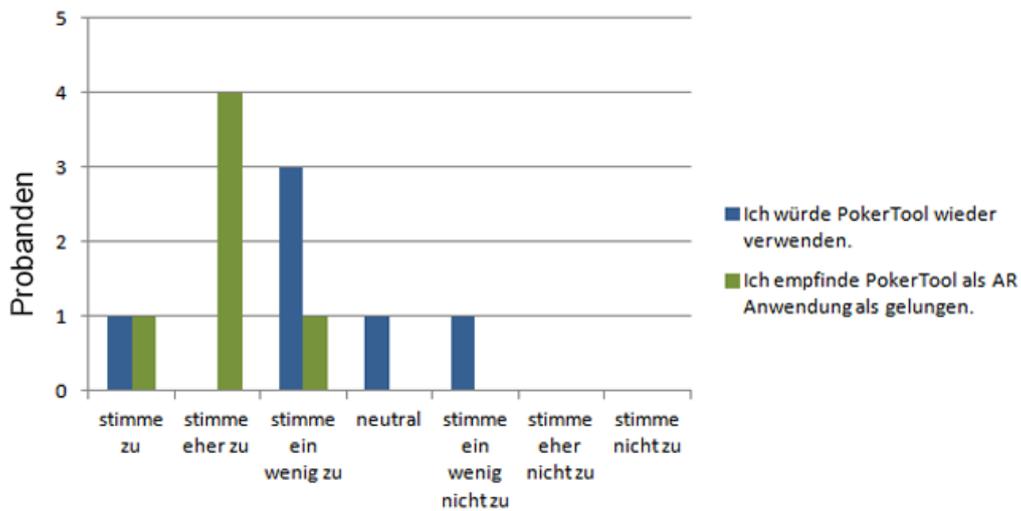


Abbildung 6.6.: Evaluationsergebnisse: Einstellung zur Ganzheit und Wiederverwendbarkeit von PokerTool

Eine Antwort für die Frage, warum die Spieler die Applikation nicht wirklich wiederverwenden wollen, liegt auch darin begründet, ob sie *PokerTool* gestört hat. Abbildung 6.7. zeigt, dass im Median die Benutzer gespalten sind über den Störfaktor der Benutzung. Die Hälfte der Spieler gab an, dass sie die Verwendung störte, die andere Hälfte eher nicht. Auf der anderen Seite sagten die Probanden, dass sie es im Median eher nicht gestört hat, als die Mitspieler die Anwendung nutzten.

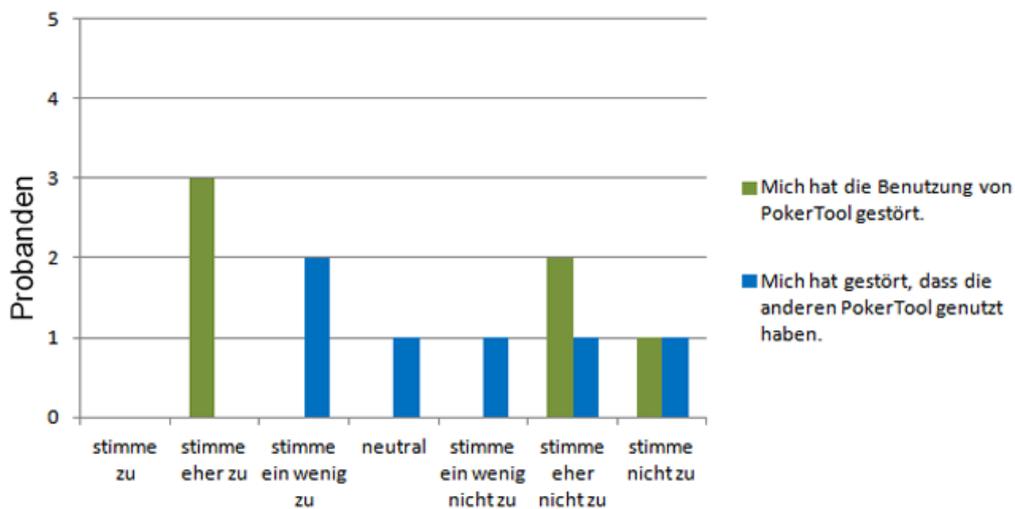


Abbildung 6.7.: Evaluationsergebnisse: Einstellung über den Störfaktor durch PokerTool

Obwohl die Nutzung von *PokerTool* den Spieler selbst eher weniger störte, sagten jedoch alle, dass durch die Nutzung der Spielfluss beeinträchtigt wurde. Ob diese Meinung, wie in Abbildung 6.8. gezeigt, auch im Spiel messbar war, sollen die Log-Daten zeigen.

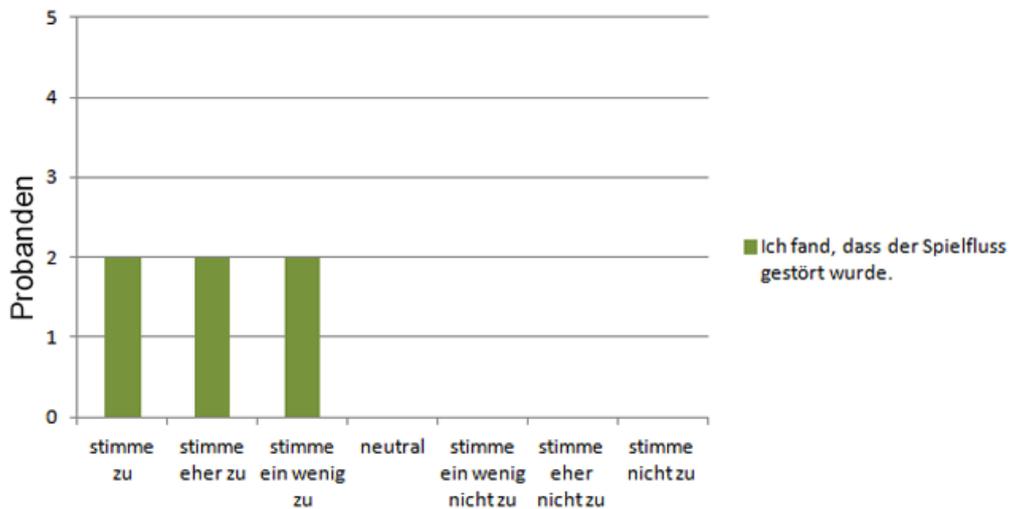


Abbildung 6.8.: Evaluationsergebnisse: Beeinträchtigung des Spielflusses durch PokerTool

Zum Schluss wurden die Spieler nach der Meinung zu einer perfekten Umsetzung von *PokerTool* im speziellem und Augmented Reality im Allgemeinen gefragt. Wenn *PokerTool* in einem für den Interaktionsgedanken perfekten Setup konstruiert wäre, würden sie es dann nutzen? Das perfekte Setup sieht vor, dass der Benutzer kein Device in den Händen halten muss und sich komplett auf sein Spiel konzentrieren kann wie z.B. bei einer AR-Brille. Über die AR-Brille würde hierbei dem Benutzer jederzeit die gleichen

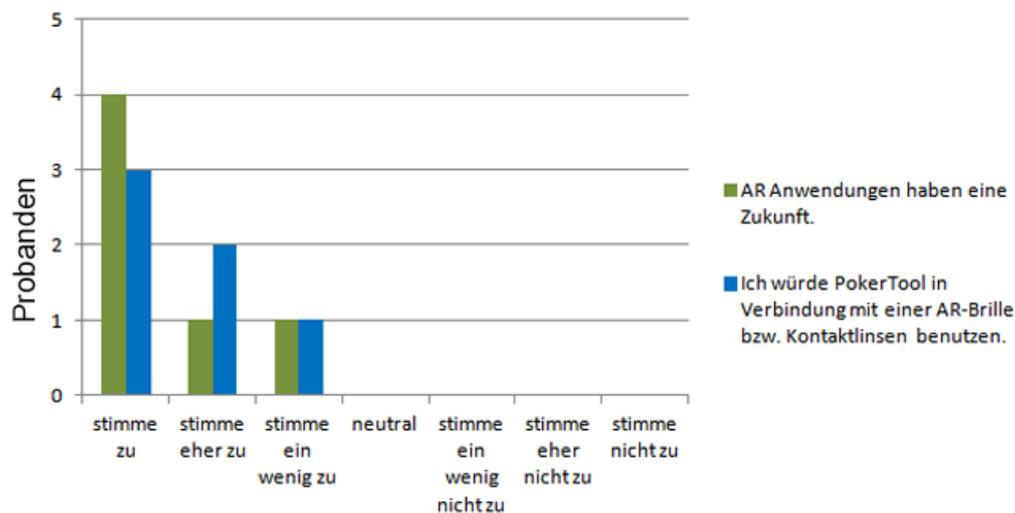


Abbildung 6.9.: Evaluationsergebnisse: Einstellung zur Zukunft von PokerTool und AR im Allgemeinen

Informationen wie durch PokerTool zur Verfügung gestellt. Die Erfassung der Karten soll vollkommen selbstständig ablaufen. Bei einem solchen Szenario, zeigt Abbildung 6.9., dass die Wiedernutzung auf einen Wert von 1,5 im Median ansteigt. Die Gruppe sagt darüber hinaus auch mit einem Median von 1, dass Augmented Reality eine Zukunft haben wird, ein Potential wird hier bereits gesehen.

6.3.2. Log-Daten

Die Log-Daten wurden von jedem PokerTool-Benutzer individuell erfasst. Für jeden Spieler konnte so ein persönliches Nutzungsprofil angelegt werden. Abbildungen 6.10. bis 6.14. zeigen die einzelnen Profile im ersten Turnier. Auf der X-Achse ist der komplette Spielverlauf abgebildet. Auf der Y-Achse findet man die akkumulierte Nutzungsdauer in Millisekunden. Die geloggte Nutzung entsteht, wenn der Spieler den *show*-Button betätigt. Es wurden nur diese Daten erfasst, um den Probanden nicht zu stören und dadurch ein natürliches Spielverhalten zu erzeugen. Steigt die Kurve, so spiegelt das die Nutzung von *PokerTool* zu der gegebenen Zeit wieder. Stagniert die Kurve, wurde das Assistenztool zu dieser Zeit nicht benutzt. Die individuelle Nutzung wurde angereichert durch die Informationen, wann eine Hand von dem Benutzer gewonnen (grüner Punkt) und wann eine Hand von einem Gegner gewonnen wurde (roter Punkt). Darüber hinaus kann das Ausscheiden aus dem Turnier an der gestrichelten Linie festgestellt werden.

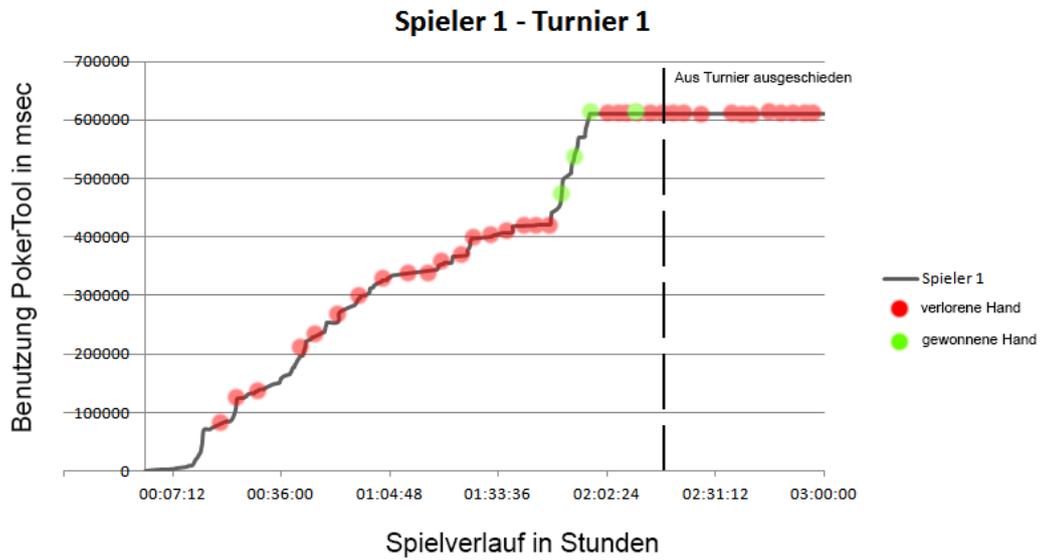


Abbildung 6.10.: Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 1 im ersten Turnier

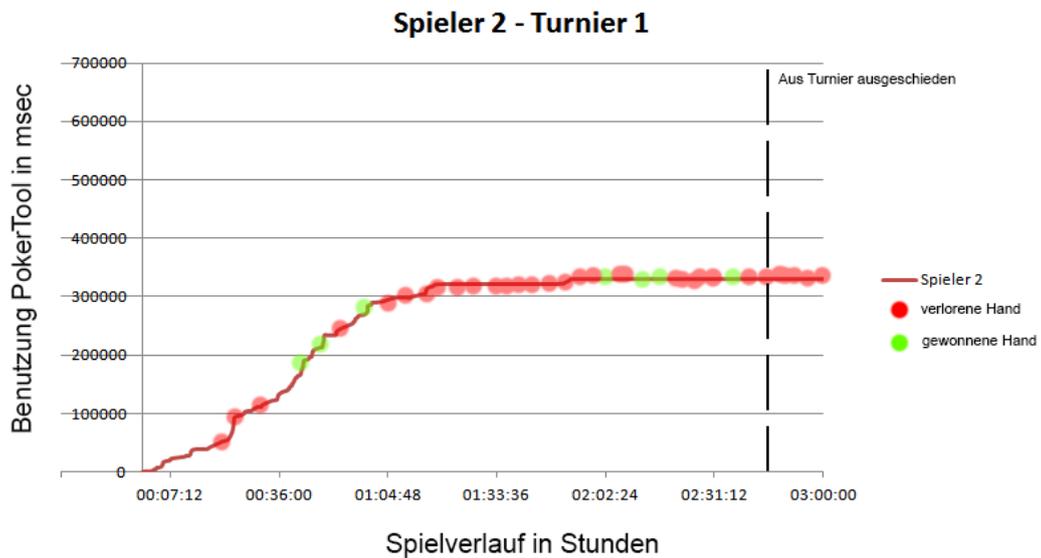


Abbildung 6.11.: Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 2 im ersten Turnier

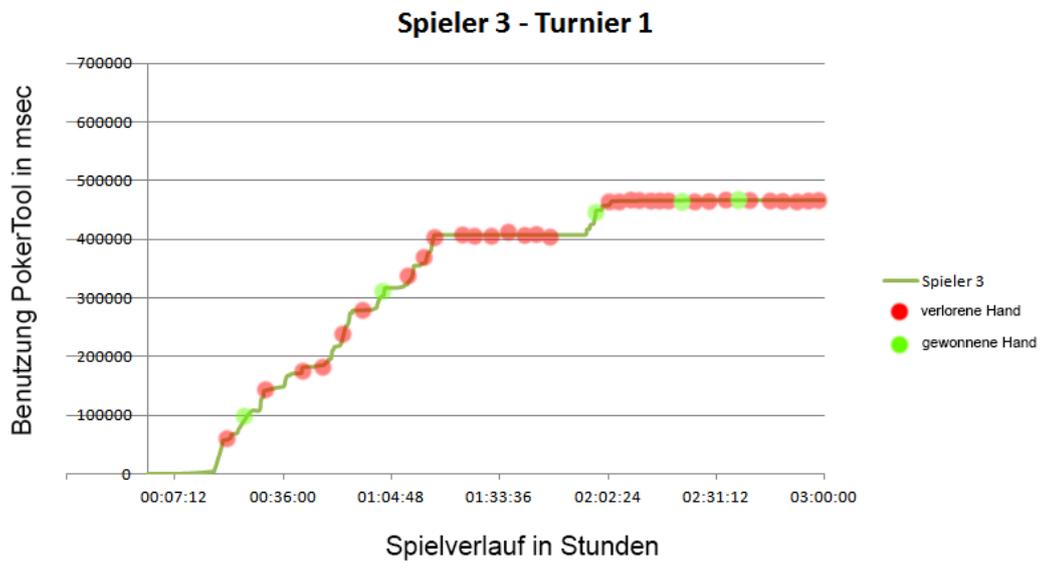


Abbildung 6.12.: Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 3 im ersten Turnier

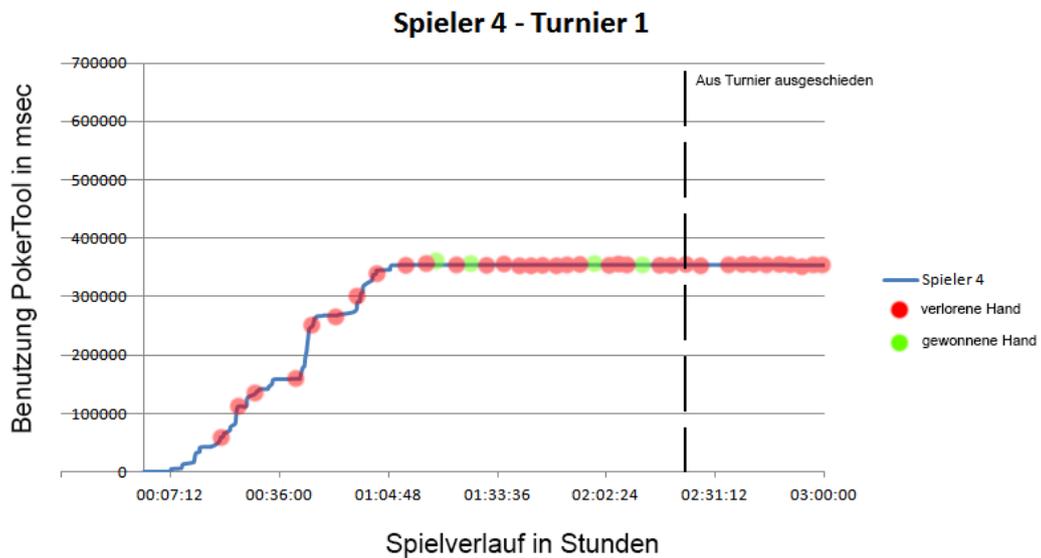


Abbildung 6.13.: Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 4 im ersten Turnier

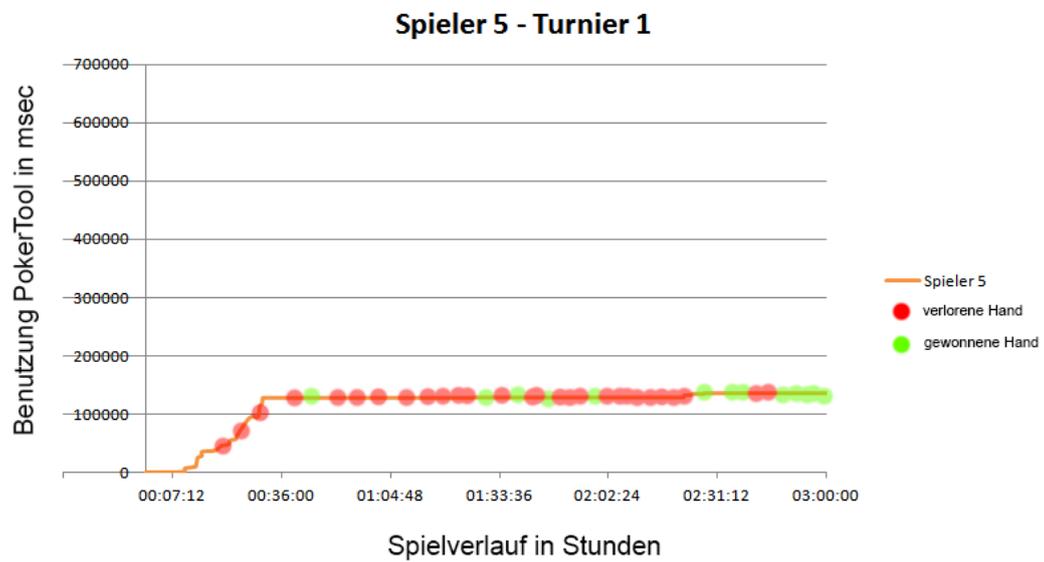


Abbildung 6.14.: Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 5 im ersten Turnier

Im zweiten Turnier nutzten Spieler 2, 4 und 5 PokerTool nicht weiter. Es liegen nur Daten von Spieler 1 und 3 vor. Über die Nutzung der Anwendung konnten die Spieler diesmal von Beginn an selbst entscheiden. Die Nutzungsprofile von Turnier 2 sind in Abbildung 6.15. zu finden.

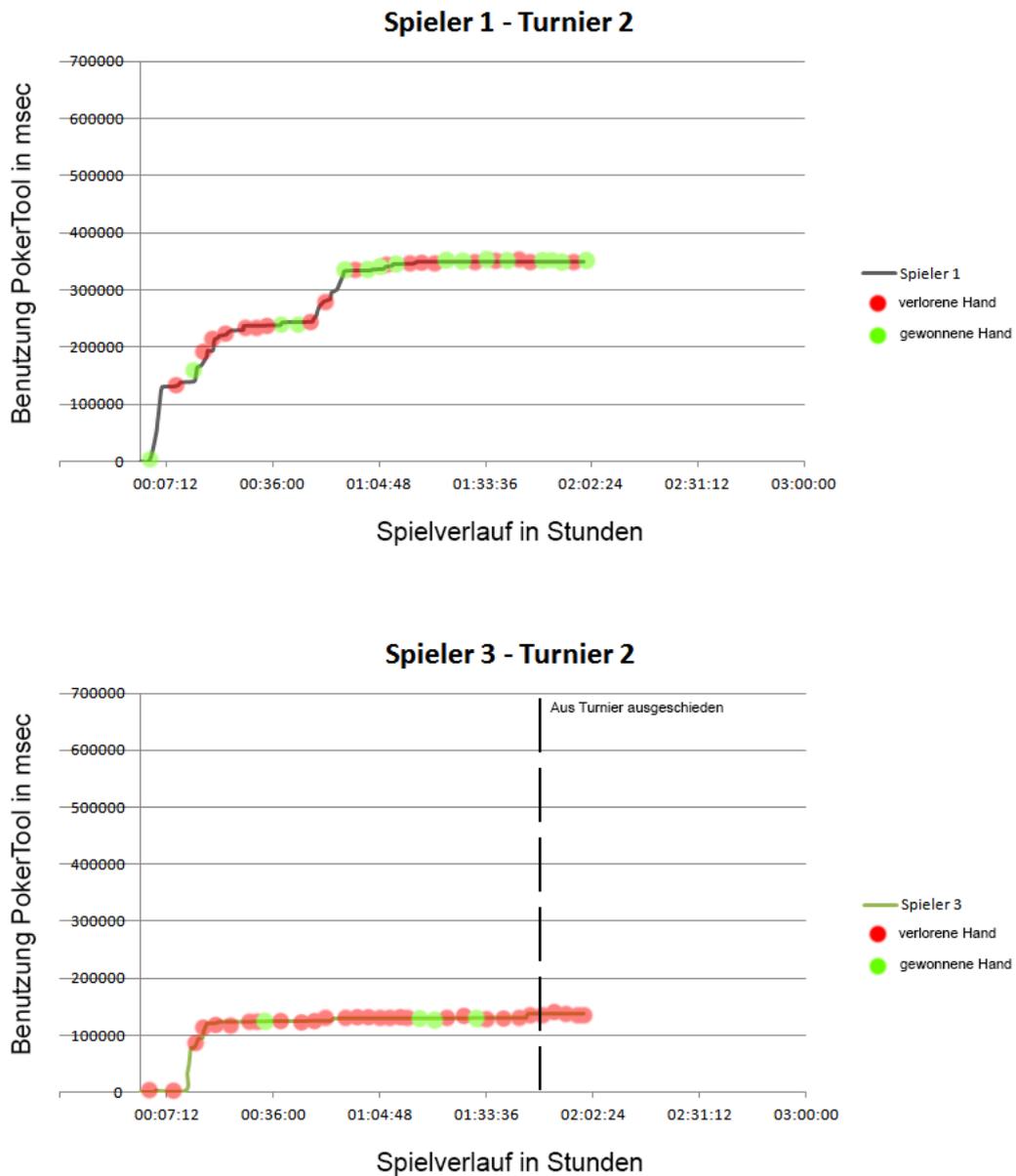


Abbildung 6.15.: Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 1 und 3 im zweiten Turnier

Aus diesen Profilen lassen sich im Speziellen Informationen über *Spielfluss*, *Nutzung*, *Einsatzrate* und *Erfolgsrate* extrahieren.

▪ **Spielfluss**

Der Spielfluss kennzeichnet die Geschwindigkeit des Spielverlaufes. Es soll untersucht werden, ob *PokerTool* einen Einfluss darauf genommen hat.

In der ersten halben Stunde wurden die Spieler angewiesen, bei jeder ihrer Aktionen *PokerTool* zu benutzen. Danach durften die Probanden selbstständig über den Einsatz des Tools bestimmen. In den ersten 30 Minuten wurden so drei von insgesamt 41 Runden gespielt. Zum Vergleich kann der Zeitraum ab Minute 120 genommen werden, da von da an bis zum Schluss keiner Hilfe durch die Software angenommen hat. In diesem Abschnitt wurden innerhalb einer Stunde 18 Hände gespielt. Es wurden somit drei mal so viele Runden beendet, wie im selben Zeitraum mit *PokerTool*. Da jedoch in der letzten Stunde des Spiels Spieler aus dem Turnier ausschieden und so die kleinere Anzahl darüber hinaus das Spiel beschleunigt, kann zum Vergleich auch der Spielabschnitt von Minute 60 bis 120 betrachtet werden. Zu diesem Zeitpunkt ist kein Spieler ausgeschieden, jedoch nutzten die Probanden nur in Einzelfällen die App. In dieser Zeit wurden 15 Runden gespielt. Das sind 2,5 mal mehr Hände als bei vollständiger Nutzung von *PokerTool*.

Im Vergleich zum zweiten Spiel mit derselben Anzahl an Mitspielern kann eine weitere Abschätzung vorgenommen werden. Insgesamt wurde im zweiten Turnier eine kürzen Turnierzeit von zwei Stunden erreicht (zum Vergleich drei Stunden im ersten Turnier). Der einzige Unterschied zwischen den Turnieren lag bei der Verwendung von *PokerTool*. Nur zwei Personen nutzten die AR-Anwendung in speziellen Fällen. Insgesamt wurden in zwei Stunden 44 Hände gespielt. Im Durchschnitt wurden so pro Stunde 13,6 Hände im ersten und 22 Hände pro Stunde im zweiten Turnier gespielt.

Ein Vergleich der beiden Turniere zeigt das Indiz, dass ein Einfluss durch *PokerTool* sichtbar ist. Jedoch muss beachtet werden, dass es sich nicht um *identische* Turniere handelte, bei dem in einem Turnier auf *PokerTool* verzichtet wurde. Um identische Turniere zu schaffen, müsste eine identische Abfolge der ausgeteilten Karten erfolgen.

Trotzdem lässt sich folgende Aussage treffen: Eine Verlangsamung des Spielflusses ist eindeutig feststellbar.

▪ **Nutzung**

Die Nutzung soll Aufschluss auf die Extraktionseffizienz der Informationen und die Häufigkeit des Einsatzes von *PokerTool* geben.

Im Fragebogen wurden die Probanden nach ihrer Einschätzung über die Benutzung von *PokerTool* befragt. Es sollte angegeben werden, wie viel sie das Programm *subjektiv* beim Spielen benutzt haben. Nach eigenen Angaben war dies im Durchschnitt bei jeder 5. Hand. Diese eigene Annahme kann durch die Log-Daten überprüft werden. Die wirkliche Nutzung wird in Tabelle 6.1. gezeigt. Im Durchschnitt überprüften die Spieler ungefähr 35% ihrer Hände. Ein Indiz dafür, dass die Nutzung nicht so auffällig war, wie die Erinnerung daran.

Spieler	gespielte Hände mit PokerTool	Gesamte Anzahl an gespielten Händen	Prozentuale Nutzung
Turnier 1			
Spieler 1	21	28	75%
Spieler 2	12	35	34%
Spieler 3	13	41	31,7%
Spieler 4	8	30	26%
Spieler 5	4	41	9,7%
Turnier 2			
Spieler 1	15	44	34%
Spieler 3	6	34	17,6%

Tabelle 6.1.: Gespielte Hände mit PokerTool

Tabelle 6.2. zeigt weiter die Nutzungsdauer von *PokerTool* durch die Spieler. Im Durchschnitt wurde das Programm im ersten Turnier insgesamt 379,229 Sekunden genutzt. Im zweiten Turnier setzten die Spieler das Programm gezielter ein und erzielten so eine durchschnittliche Nutzung von 244,375 Sekunden. Schaut man sich die Dauer im einzelnen an, erkennt man, dass eine durchschnittliche Anzeige im ersten wie im zweiten Turnier bei ungefähr 2,2 bis 2,3 Sekunden liegt. Dieser Wert lässt mehrere Schlüsse zu:

- Die Art der Nutzung bleibt über längere Zeit gleich.
- Die Extraktion der Information ist in kurzer Zeit zu bewältigen.
- Das visuelle Checken, ob die Karten richtig detektiert wurden, ist in kurzer Zeit zu bewältigen.

Hier zeigen sich die durchdachte Konzeption und die Vorteile der AR-Anwendung als Visualisierungstool.

Allgemein lässt sich weiter aus diesen Daten lesen, dass 3 von 5 Spielern, denen *PokerTool* zur Verfügung stand, im zweiten Turnier komplett auf seinen Einsatz verzichteten. Die Gründe hierfür wurden bereits oben erwähnt und besagen, dass die Interaktion als unpraktisch angesehen wurde. Die kurze Dauer des Anzeigens

Spieler	Nutzungsdauer	Informationen abgerufen	Durchschnittliche Anzeige
Turnier 1			
Spieler 1	609,576 Sek	209	2,917 Sek
Spieler 2	329,961 Sek	174	1,896 Sek
Spieler 3	466,197 Sek	192	2,428 Sek
Spieler 4	353,579 Sek	146	2,421 Sek
Spieler 5	136,834 Sek	72	1,900 Sek
Turnier 2			
Spieler 1	350,583 Sek	104	3,370 Sek
Spieler 3	138,167 Sek	134	1,031 Sek

Tabelle 6.2.: Nutzungsdauer von PokerTool

hingegen spricht für ein gut durchdachtes AR-Design, in dem die wichtigsten Informationen schnell vermittelt werden können.

Als Schlussfolgerung kann gezogen werden, dass PokerTool effizient in der Informationsextraktion designt wurde. Die Informationsvermittlung beruht, wie in der Konzeptphase beschrieben, auf allen Grundsätzen einer AR-Anwendung. Die Spieler bleiben in ihrem Nutzungsverhalten gleich, setzen das Tool aber im Verlauf der Turniere immer gezielter ein. Der Benutzer entschließt sich, dass eine ständige Anzeige der Wahrscheinlichkeiten für ihn nicht notwendig ist.

▪ Einsatzrate

Die Einsatzrate soll evaluieren, wann *PokerTool* von den Testpersonen eingesetzt wurde. Hierfür soll das Hauptaugenmerk auf die eingezeichneten gewonnen bzw. auch auf die verlorenen Hände gerichtet werden. Bei allen Kurven ist eine bestimmte Situation vergleichbar. Als Vorzeigebeispiel soll hierfür das Spiel von Spieler 1 im ersten Turnier in Abbildung 6.10. betrachtet werden. In der Zeit von 1:40 Stunde bis 1:55 Stunde konnte Spieler 1 drei Hände in Folge für sich entscheiden. Diese Erfolge sind im Diagramm mit einem starken Anstieg der Benutzung von PokerTool verbunden. Dies lässt den Schluss zu, dass der Spieler in einer Situation, in der er vermeintlich gute Karte hält, diese erfasst und die Wahrscheinlichkeiten zu gewinnen oder eine bessere Hand überprüft.

Die Situationen lassen sich in allen Diagramm wiederfinden. Ein weiteres Beispiel ist auch in Abbildung 6.12. im Zeitraum von 1:00 Stunde bis 1:15 Stunde zu sehen. Hier verliert Spieler 3 innerhalb von 15 Minuten drei Hände. Jedoch ist ein starker Anstieg in der Nutzung von PokerTool erkennbar. Nach jeweiligem Anstieg folgt eine kurze Stagnation in der Kurve, in der die jeweilige Entscheidung vorliegt. Wieder lässt sich die anstehende Entscheidung durch die Benutzung der Software erkennen.

Als Schluss lässt sich daraus ziehen, dass sich die Einsatzrate von PokerTool in entscheidenden Situationen häuft. Die Benutzer möchten jede Information des AR-Systems in ihre Entscheidungsfindung einfließen lassen, um das bestmögliche Resultat zu erzielen. In diesen Situationen überwiegen die gewonnenen Informationen den Nachteilen in der Handhabung einer Smartphonebenutzung.

▪ Erfolgsrate

Nach der Einsatzrate soll auch die Erfolgsrate von PokerTool untersucht werden. Hierbei soll im Vordergrund stehen, ob AR-unterstützte Spieler ein besseres Ergebnis erzielen. Dieser Evaluationspunkt ist komplex und setzt sich aus mehreren Faktoren zusammen. Denn der Erfolg von *PokerTool* ist nicht nur darin zu sehen, ob eine Person mit *PokerTool* mehr gewinnt, sondern auch, ob in schwierigen Situationen die Angaben durch die Software in eine Fold-Entscheidung münden und damit dem Benutzer einen höheren Verlust erspart haben.

Als erste Aussage kann festgehalten werden, dass das AR-System in der Endphase nicht eingesetzt wurde. Die Gründe liegen darin, dass bei einer geringen Anzahl von Mitspielern, sich der Spielfluss erhöht. In dieser Zeit ist die Handhabung des AR-Systems von PokerTool den Benutzern zu aufwändig.

Als zweite Aussage kann geschlossen werden, dass der erste Turnier-Gewinner (Spieler 5) früh von der Benutzung und der Handhabung absah. Er konzentrierte sich auf seine eigenen Erfahrungen. Der zweite Turnier-Gewinner (Spieler 1) hat PokerTool in beiden Turnieren genutzt und hat, wie bereits oben erwähnt, den späteren Einsatz gezielt auf spezielle Situationen angewendet. Hier könnte ein Erfolg durch PokerTool naheliegen, da im zweiten Durchlauf nur zwei von sieben Spielern die Hilfestellung nutzten. Im ersten Durchgang wurde der Vorteil durch die Chancengleichheit minimiert.

Um genauer auf den Erfolg durch ein AR-System zu schließen, sollen noch einmal die gewonnen Hände jedes geloggtten Spielers verglichen werden. Tabelle 6.3. zeigt die Anzahl der Hände, die ein Spieler gewonnen hat und nachweislich in der Zeit die Anwendung genutzt hat. In der letzten Spalte steht dem gegenüber, wie viele Hände der Spieler insgesamt in dem Turnier gewonnen hat. Spieler 1 bis 3 haben im Durchschnitt in fast 60% der Hände, die sie gewonnen haben, das Programm um Hilfe gefragt. Spieler 4 und 5 können aus dieser Rechnung herausgenommen werden, da sie relativ früh für sich entschieden haben, auf diese Hilfestellung zu verzichten. Der Verzicht des Assistenten kann jedoch nicht als Nachteil angesehen werden, da Spieler 5 nachweislich Erfolg hatte.

Bei Turnier 2 zeigt Tabelle 6.3. für Spieler 1 eine Erfolgsrate mit *PokerTool* von knapp 47% an. Der an die Spielsituation angepasste Prozentsatz liegt, bei den

Händen, in denen PokerTool mit Erfolg beteiligt war, höher. Angepasst deswegen, weil wenn die Ausgangslage betrachtet wird, fällt auf, dass Spieler 1 nach einer Stunde und 15 Minuten nicht mehr auf die AR-Anwendung zurückgreift. Zu diesem Zeitpunkt hat er in 6 von 8 gewonnenen Händen PokerTool benutzt. Der Prozentsatz liegt hier bei 75%. Nach 75 Minuten wurde durch den erhöhten Spielfluss die Handhabung nach Angaben des Benutzers zu aufwändig.

Spieler	gewonnene Hände mit Pokertool	gewonnene Hände insgesamt
Turnier 1		
Spieler 1	3	4
Spieler 2	3	7
Spieler 3	3	5
Spieler 4	0	4
Spieler 5	0	13
Turnier 2		
Spieler 1	8	17
Spieler 3	0	4

Tabelle 6.3.: Gewinnstatistik unter Benutzung von PokerTool

Zusammenfassend kann für die Erfolgsrate von *PokerTool* ein positives Urteil gefällt werden. Jedoch müssten mehrere Evaluationen diese These bestätigen. Die AR-Anwendung wurde in entscheidenden Situationen eindeutig genutzt. Nach Angaben der Probanden im Fragebogen sahen sie die Informationen als Hilfestellung an, was den Verdacht auf eine positive Erfolgsrate zulässt.

6.4. Bewertung

Die Bewertung der Ergebnisse soll unter der Beachtung der aufgestellten These erfolgen. Im Kern soll diskutiert werden, ob Augmented Reality Smartphone-Anwendungen einen Mehrwert liefern, jedoch deren Handhabung zu umständlich ist.

Zu diesem Zweck sollen nun alle Ergebnisse im Gesamtkontext gesehen und evaluiert werden. Die Aufstellung der Gruppe bietet einen guten Ausgangspunkt für die Bewertung. Die Gruppe war Smartphone-erfahren, jedoch wiesen sie verschiedene Stadien der Pokererfahrung auf. Dies ist Voraussetzung, um zu zeigen, dass die Nutzung des Smartphones den Spielern sehr vertraut ist. Fehler durch Falscheingaben sowie das Falsch-Erfassen der Smartphone-typischen Funktionalitäten sind nur in einer geringen Zahl zu erwarten. Die Benutzer sind die Handhabung gewohnt und mit ihr vertraut.

Durch das unterschiedliche Pokerkwissen konnte auf eine unterschiedliche Nutzung gesetzt werden. Unerfahrene Spieler konnten durch das System in ihrer Entscheidung unterstützt werden, als erfahrene. Diese Eigenschaft schafft ein gutes Einordnen in das Gesamtbild.

Aus dem Fragebogen ging hervor, dass die Spieler subjektiv einen Mehrwert erkannt haben und das Design als gelungen empfinden. Es half ihnen, die Informationen schnell zu erfassen und zu verarbeiten. Die Sinnhaftigkeit ist *subjektiv* gegeben. Weiter kann durch die eigene Unterschätzung der Häufigkeit des Anwendens auf eine intuitive und unauffällige Aufgabenbewältigung durch *PokerTool* geschlossen werden.

Die Einschätzung, dass ein Mehrwert sowie ein intuitives Design vorliegt, ließ sich auch aus der Analyse der Log-Daten schließen. Die **Nutzung** konnte als effizient in der Extraktion und der Vermittlung der Informationen bezeichnet werden. Die **Einsatzrate** ließ den Schluss zu, dass in wichtigen, entscheidenden Situationen das Tool bedient wurde. Die **Erfolgsrate** zeigt eine positive Auswirkung von *PokerTool* an. Alle diese Kriterien belegen, dass die AR-Anwendung gut durchdacht ist und dadurch dem Benutzer einen Mehrwert bringt.

Der erste Teil der These scheint erfüllt. Um den zweiten Teil zu bestätigen oder zu widerlegen, sollen wieder zuerst die Ergebnisse des Fragebogens einbezogen werden. Hier gaben die Spieler an, dass sie die eigene Nutzung subjektiv störte und sie eher weniger das Tool wiederverwenden würden. Diese Aussagen lassen sich auch durch die Log-Daten untermauern. Der verlangsamte **Spielfluss** zeigt, dass durch die Anwendung mehr Zeit benötigt wird. Weiter wurde das Visualisierungstool im Verlauf der Turniere gezielter eingesetzt. Eben dann, wenn der Vorteil durch *Mehr-Informationen* dem Nachteil der *Handhabung* überwog. Diese Argumentation bekräftigt die Aussage, dass selbst eine gut durchdachte AR-Anwendung durch ihre Benutzung ein negatives Auftreten hat.

Trotz dieser Bewertung in der Benutzung, empfanden die Probanden Augmented Reality als positive Erscheinung. Ihr Wunsch nach mehr sinnvollen Applikationen und durch die Verbreitung von AR-Anwendungen als Marketingtool zeigt, dass diese Art der Visualisierung dem Benutzer gefällt. Die Verbesserungsvorschläge an *PokerTool* weisen auch hier in die Richtung, dass weniger direkte Interaktion wünschenswert ist. Je weniger die Personen in ihrem normalen Handeln eingeschränkt werden, dabei aber mehr Informationen über ihre Umwelt erfahren, desto mehr würden die Testpersonen ein Tool nutzen. Als Ergebnis steht hier fest: *Die Anwendung soll sich dem Benutzer anpassen und nicht umgekehrt.*

Diese Folgerung stützt sich auch auf die Aussage, dass die Testpersonen *PokerTool* in einer optimalen Umsetzung, z.B. mit Hilfe einer AR-Brille, nutzen würden. Der Schritt in Richtung einer AR-Brille zeigt, dass das Handeln der Personen nicht eingeschränkt, sondern dass ihr *normales* Sichtfeld mit digitalen Informationen angereichert werden soll. Die heutige Situation, das Aufgabenfeld mit Hilfe eines mobilen Device zu lösen, verursacht das Problem der Abschätzung zwischen Nutzen und Kosten.

Die These:

Durch gut durchdachte Augmented Reality Smartphone-Anwendungen erfährt der Benutzer einen Mehrwert, jedoch ist dieser Mehrwert nicht groß genug, um den Nachteil der unpraktischen Handhabung zu egalisieren, der durch das Trägermedium Smartphone entsteht. Je weniger der Benutzer mit einem AR-Gerät direkt interagieren muss, desto mehr überwiegen die Vorteile.

gilt in dieser Argumentation als bewiesen.

7. Fazit und Aussicht

Zum Abschluss dieser Arbeit soll nun das Fazit und eine Aussicht auf künftige Themen erarbeitet werden.

7.1. Fazit

Diese Arbeit hatte das Ziel, durch eine durchdachte Konzeption und ein gutes Design eine Augmented Reality Anwendung zu schaffen, die einen Mehrwert hervorbringt.

Der klar definierte Bereich der Pokervariante Texas Holdem ist das ideale Umfeld, um die Aufgabe zu bearbeiten. Der computergenerierte Mehrwert anhand der Wahrscheinlichkeitsberechnung sowie die Eigenschaften des Visualisierungstools durch Augmented Reality sind hierbei eine gut geeignete Kombination für einer AR-Anwendung.

In einer ausführlichen Einführung in das Thema Poker, wurden in dieser Arbeit die Grundlagen und der Ablauf des Spiels erklärt. Anhand des Wissens konnte auf die Gewinnchancen sowie die Odds eingegangen werden, die mittels des Computers bestimmt werden können. In diesem Rahmen wurde auch ein Programm präsentiert, das ohne AR-Lösung auskommt.

Im Folgenden wurde das Framework *Metaio SDK* vorgestellt, mit deren Hilfe das Tracken der Karten sowie die Anzeige der digitalen Inhalte in der realen Welt bewerkstelligt wurde. Mit diesem Workaround konnten die Grundlagen geschaffen werden, um einen optimalen Interaktionsdialog zwischen Mensch und Maschine zu erschaffen. Es sollte gezeigt werden, dass eine AR-Anwendung zwar eine neue Möglichkeit der Visualisierung bietet, jedoch allen Gestaltungsgrundsätzen unterliegt, denen herkömmliche Anwendungen auch untergeordnet sind. Für diesen Zweck wurde das Trägermedium Smartphone betrachtet und anschließend die Grundsätze der Dialoggestaltung und die Prinzipien der Designgestaltung erörtert.

Die Implementation diskutierte eine Möglichkeit auf die gestellten Probleme, wie das Erfassen der Karten, die Berechnung der Wahrscheinlichkeiten und das Anzeigen aller erstellten Informationen, sinnvoll und effizient zu reagieren.

Abschließend prüfte die Evaluation die These, dass eine gut durchdachte AR-Smartphone-Anwendung zwar einen Mehrwert generieren kann, jedoch die unpraktische Handhabung

eines Smartphones dadurch nicht egalisiert. Dafür wurde die Anwendung in realistischer Umgebung getestet. Das Nutzungsverhalten der Testpersonen sowie ihre Meinung zu dem Thema wurden erfasst. Die Analyse wertet ihre Einstellungen sowie die Nutzung, Einsatzrate, Erfolgsrate und den Einfluss von *PokerTool* auf den Spielfluss aus. Die anschließende Argumentation bestätigte, dass je weniger die Benutzer mit einem AR-Gerät direkt interagieren müssen, desto eher überwiegt der Nutzen der zu erhaltenden Informationen.

7.2. Aussicht

Im letzten Abschnitt soll noch einmal auf die Aussichten eingegangen werden, die diese Arbeit geschaffen hat. Zum Einen ergab die Evaluation, dass *PokerTool* zwar eine durchdachte Lösung ist, jedoch konnten auch hier Verbesserungen angebracht werden. Die meisten Anmerkungen richteten sich an das Tracken der Karten. Dieses lief robust, akkurat und effizient unter guten Lichtverhältnissen. Es traten jedoch bei Reflexionen und Dunkelheit Probleme auf. Diese Anmerkungen zeigen auf, dass Augmented Reality noch weiteres Entwicklungspotenzial besitzt.

Die zukünftige Arbeitsweise mit AR wird jedoch durch Entwicklungen wie *GoogleGlass*¹ gesichert. Das Aufgabenfeld erweitert sich durch die neuen technischen Möglichkeiten. AR wird somit immer mehr in unser alltägliches Leben einfließen. Dieser Gedankengang bringt jedoch eine neue These hervor, die es zu evaluieren gilt. Wir leben heute in einer Welt, in der wir die Umwelt auf unsere visuelle Sicht angepasst haben. Alle Schilder, die uns führen, sind für eine schnelle Informationsvermittlung *ohne* technische Hilfsmittel gestaltet worden. Dies lässt die Frage aufkommen, ob eine Entwicklung von Augmented Reality in manchen Umgebungen sinnvoll ist. Es stehen nur die alltäglichen Situationen im Fokus. Dass ein Augmented Reality gestützter Pilot nur Vorteile aus der vorhandenen Technik bezieht, ist hier nicht Gegenstand der Diskussion.

Augmented Reality zeigt jedoch durch seine beliebte Anwendung in allen Bereichen Lebens, dass seine zukünftige Entwicklung vorangetrieben wird und das vollständige Potenzial von AR noch nicht ausgeschöpft ist.

¹<http://www.google.com/glass/>

Literaturverzeichnis

Lay 2013

Layar. <http://www.layar.com/>. Version: 2013 4.3

Wik 2013

Wikitude. <http://www.wikitude.com/app/>. Version: 2013 4.3

Allan Oliveira 2012

ALLAN OLIVEIRA, Regina B. A.: *Creation and Visualization of Context Aware Augmented Reality Interfaces*. University of Sao Carlos, 2012 4.4.1, 4.4, A.1

Alliance 2013

ALLIANCE, Open H.: *Android Design Principles*. <http://developer.android.com/design/get-started/principles.html>. Version: 2013 5.3, 5.4.2

Azuma 1997

AZUMA, Ronald T.: *A Survey of Augmented Reality*. Hughes Research Laboratories, August 1997 4.1, 4.2

Bachfischer 2011

BACHFISCHER, Dr. N.: *Augmented Reality Marketing: Hype or Added Value?* <http://www.aquarius.biz/en/2011/10/25/augmented-reality-marketing-hype-or-added-value/>. Version: 2011 4.2, 6.1

Gambke 2010

GAMBKE, Thomas: *Spielanleitung Poker Texas Holdem*. Spielbank Bad Neuenahr, Juni 2010 2.1, 2.2

Katier 2011

KATIER, Bendert: *Mobile Augmented Reality*, University of Amsterdam - Business Studies, Diplomarbeit, Juni 2011 4.2

metaio 2013

METAIO: *Metaio SDK Requirements*. <http://dev.metaio.com/sdk/getting-started/android/setting-up-the-development-environment/>. Version: 2013 3.1, 4.4.1

Paul Milgram 1994

PAUL MILGRAM, Akira Utsumi Fumio K. Haruo Takemura T. Haruo Takemura: *Augmented Reality: A class of displays on the reality-virtuality continuum*. 1994 4.1, 4.1, A.1

Podeszwa 2010

PODESZWA, Lubosz: *PokAR: Spielkarten als Marker*. Universität Koblenz, Januar 2010 2.4.1

Poneta 2011

PONETA, Olaf: *Bildbasierte Integration von Software am Beispiel der Entwicklung eines Expertensystems für Online-Poker-Plattformen*. Universität Koblenz, Dezember 2011 2

Risk 2009

RISK, Nicholas A.: *USING COUNTERFACTUAL REGRET MINIMIZATION TO CREATE A COMPETITIVE MULTIPLAYER POKER AGENT*, University of Alberta, Diplomarbeit, 2009 2.3

Teofilo 2011

TEOFILO, Luis F.: *Estimating the Probability of Winning for Texas Holdem Poker Agents*, University of Porto, Diplomarbeit, 2011 2.4.3, 5.5.5

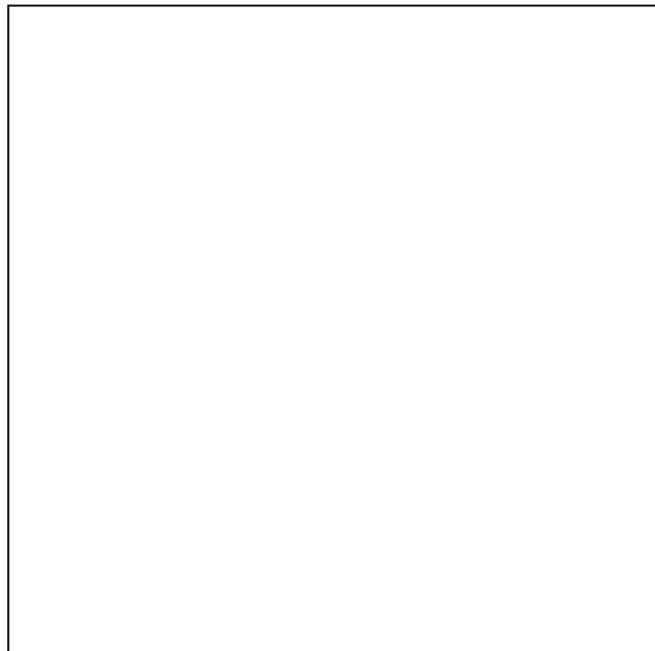
Thema-Poker.com 2013

THEMA-POKER.COM: *RANGLISTE - STARTHAENDE - 3 GEGNER*. <http://www.thema-poker.com/starthaende/rangliste/3-gegner>. Version: 2013 2.4.3, 2.2, A.1

A. Anhang

A.1. CD

Die CD enthält den vollständiger Code zu *PokerTool* sowie die komplette Masterarbeit in elektronischer Form.



Abbildungsverzeichnis

2.1. Poker Odd Calculator: Android Anwendung zur Bestimmung der Gewinnwahrscheinlichkeiten	11
4.1. Vereinfachte Einordnung der Begriffe Augmented Reality im Kontext der Virtual Reality Continuum nach MILGRAM ([Paul Milgram 1994]) . . .	15
4.2. Screenshots von Layar: Marker-Detektion in Printmedien	17
4.3. Screenshots von Wikitude: (A) Anzeigen von Geodaten, (B) AR Spiel, (C) Währungsrechner	18
4.4. Verschiedene Arten der Anzeige einer Augmented Reality Schnittstelle; A - Static Tracked; B - Dynamic Tracked; C - Dynamic with Vision (Allan Oliveira [2012])	19
5.1. PokerTool: Design des Hauptbildschirms	28
5.2. PokerTool: Design der Kartenwert-Symbole	29
5.3. PokerTool: Aktivitätsdiagramm des allgemeinen Programmablaufes . . .	30
5.4. PokerTool: Reihenfolge der Startbildschirme	32
5.5. PokerTool: Augmented Reality Feedback und Wahrscheinlichkeitsrechnung	36
5.6. PokerTool: Aktivitätsdiagramm zur Gewinnwahrscheinlichkeitsberechnung	38
5.7. PokerTool: Aktivitätsdiagramm zur Schüttelgeste	54
6.1. Evaluationsergebnisse: Vor Benutzung von PokerTool über Erfahrung mit Poker und Smartphone-Nutzung	58
6.2. Evaluationsergebnisse: Vor Benutzung von PokerTool über Einstellung zu AR	59
6.3. Evaluationsergebnisse: Einstellung zu PokerTool vor der Benutzung nach der Sinnhaftigkeit (1) und Mehrwert (2)	60
6.4. Evaluationsergebnisse: Einstellung zur Benutzung und Design von PokerTool	61
6.5. Evaluationsergebnisse: Einstellung zur Hilfestellung durch PokerTool . .	62
6.6. Evaluationsergebnisse: Einstellung zur Ganzheit und Wiederverwendbarkeit von PokerTool	63
6.7. Evaluationsergebnisse: Einstellung über den Störfaktor durch PokerTool	64

6.8. Evaluationsergebnisse: Beeinträchtigung des Spielflusses durch PokerTool	65
6.9. Evaluationsergebnisse: Einstellung zur Zukunft von PokerTool und AR im Allgemeinen	66
6.10. Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 1 im ersten Turnier	67
6.11. Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 2 im ersten Turnier	68
6.12. Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 3 im ersten Turnier	69
6.13. Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 4 im ersten Turnier	70
6.14. Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 5 im ersten Turnier	71
6.15. Evaluationsergebnisse: Nutzung von PokerTool durch Spieler 1 und 3 im zweiten Turnier	73

Tabellenverzeichnis

2.1. Rangfolge der nach unten besser werdenden verschiedenen Pokerhände	9
2.2. Gewinnwahrscheinlichkeit im Preflop bei drei Gegenspielern (Thema-Poker.com [2013])	10
6.1. Gespielte Hände mit PokerTool	75
6.2. Nutzungsdauer von PokerTool	76
6.3. Gewinnstatistik unter Benutzung von PokerTool	78

Verzeichnis der Listings

5.1. PokerTool: Ausschnitt aus der Methode loadContent der Java-Klasse CamMode	34
5.2. PokerTool: Ausschnitt aus der Methode onDrawFrame zur Erkennung der Marker	35
5.3. PokerTool: Ausschnitt aus der Methode onDrawFrame zur Einordnung in die korrekte Speicherstelle	37
5.4. PokerTool: Ausschnitt aus der Methode updateProbabilty für die Berechnung im Preflop	39
5.5. PokerTool: Ausschnitt aus der Methode updateProbabilty für die Berechnung nach dem Preflop	40
5.6. PokerTool: handStrenght-Algorithmus	42
5.7. PokerTool: Ausschnitt aus Java-Klasse Hand: Rang bestimmen	44
5.8. PokerTool: Ausschnitt aus Java-Klasse Hand: Straight Flush bestimmen	45
5.9. PokerTool: Ausschnitt aus Java-Klasse Hand: Vierling bestimmen	46
5.10. PokerTool: Ausschnitt aus Java-Klasse Hand: Full House bestimmen	47
5.11. PokerTool: Ausschnitt aus Java-Klasse Hand: Flush bestimmen	48
5.12. PokerTool: Ausschnitt aus Java-Klasse Hand: Straight bestimmen	49
5.13. PokerTool: Ausschnitt aus Java-Klasse Hand: Drilling bestimmen	50
5.14. PokerTool: Ausschnitt aus Java-Klasse Hand: Zwei Pärchen bestimmen	51
5.15. PokerTool: Ausschnitt aus Java-Klasse Hand: Pärchen bestimmen oder niedriger	52