

# Dragon's Lair

## Ein selbsterklärendes Gesellschaftsspiel auf Basis eines TUIs

### BACHELORARBEIT

zur Erlangung des akademischen Grades

### Bachelor of Science

im Rahmen des Studiums

### Medieninformatik und Visual Computing

eingereicht von

**Lukas Pichlhöfer**

Matrikelnummer 0527644

an der Fakultät für Informatik  
der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Ing. Dr.techn. Hilda Tellioglu

Wien, 2. Mai 2016

---

Lukas Pichlhöfer

---

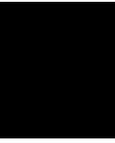
Hilda Tellioglu



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Zielsetzung . . . . .	2
<b>2 Grundlagen</b>	<b>3</b>
2.1 Tangible User Interfaces . . . . .	3
2.2 Das Phänomen “Spiel” . . . . .	4
2.3 Usability im Kontext von Computerspielen . . . . .	9
<b>3 Spielkonzept</b>	<b>15</b>
3.1 Begriffsdefinitionen . . . . .	15
3.2 Spielsteine . . . . .	17
3.3 Spielregeln . . . . .	18
<b>4 Design und Entwicklung</b>	<b>19</b>
4.1 Hardwarebasis . . . . .	19
4.2 Konstruktion des Tisches . . . . .	21
4.3 Software . . . . .	24
<b>5 User Interface</b>	<b>33</b>
5.1 Analyse . . . . .	33
5.2 Design der physischen Komponenten . . . . .	33
5.3 Design der virtuellen Oberfläche . . . . .	35
<b>6 Evaluierung des Prototypen</b>	<b>39</b>
6.1 Methode . . . . .	39
6.2 Aufbau des Tests . . . . .	40
6.3 Ergebnisse . . . . .	40
6.4 Fazit . . . . .	43
<b>7 Analyse und Ausblick</b>	<b>45</b>
7.1 Resümee . . . . .	45
7.2 Ausblick . . . . .	46

<b>Abbildungsverzeichnis</b>	<b>48</b>
<b>Glossary</b>	<b>49</b>
<b>Literaturverzeichnis</b>	<b>51</b>



# Introduction

Seit der Entwicklung der ersten Computerspiele hat diese Form der interaktiven Unterhaltung stetig an Interesse und Akzeptanz in den unterschiedlichsten Bevölkerungsgruppen gewonnen. Während das gemeinsame Spiel mehrerer Personen schon in den Anfängen dieser Entwicklung ein wichtiger Aspekt darstellte (vgl. Pong, Tennis for Two) und bis heute durch vernetztes Spielen mit einer Vielzahl von Mitspielern, zum Beispiel in Form von Massively Multiplayer Online Game (MMOG) und anderen online basierten Multiplayer-Titeln, an Bedeutung gewonnen hat, geraten soziale und kommunikative Aspekte, wie sie zum Beispiel bei Brettspielen zum Tragen kommen, durch die Art des gemeinsamen Spielens, über räumliche Grenzen hinweg, unter Umständen ins Hintertreffen. Im Gegensatz zu der direkten Interaktion, in die Mitspieler, die sich an einem Tisch gegenüber sitzen, auch abseits des Spielgeschehens treten können, fehlt diese soziale Komponente bei Online-Spielen und es wird somit ein für Gesellschaftsspiele wichtiger Teilaspekt des Spielerlebnisses eingebüßt. Einer der Aspekte in denen Computerspiele traditionellen Brett- und Kartenspielen hingegen überlegen sind, ist der zumeist niederschwellige Zugang, der es ermöglicht in das Spielgeschehen einzusteigen, ohne sich vorher lange in eine Spielanleitung einlesen zu müssen. Da das Erlernen vieler klassischer Brettspiele – speziell mit komplexeren Regeln – ein oft langwieriges Studium der Spielanleitung voraussetzt, was gerade für Neueinsteiger oder allgemein während der ersten Spielrunden eine Hürde und speziell für Kinder eine regelrechte Geduldsprobe darstellt, soll versucht werden diesen Aspekt mithilfe von Computertechnologie möglichst zu minimieren. Um dies zu untersuchen soll der Prototyp eines computergestützten Brettspiels entwickelt werden, das zum Ziel hat die sozialen Komponenten von Gesellschaftsspielen mit dem niederschweligen Zugang von Computerspielen zu verbinden. Es gilt zu untersuchen, ob sich die Vorteile von Computerspielen in ein Setting übertragen lassen, das dem eines klassischen Gesellschaftsspiels – konkret eines Brettspiels – entspricht, ohne dass dadurch die menschliche Interaktion negativ beeinflusst wird. Vielmehr soll die Kommunikation der teilnehmenden Spieler untereinander gefördert werden, indem gewisse Verwaltungsauf-

gaben, die sonst vom Spieler zu erledigen wären – zum Beispiel das Zählen von Punkten – automatisch vom System übernommen werden.

### 1.1 Zielsetzung

Es soll untersucht werden inwieweit das vom klassischen Gesellschaftsspiel bekannte Spielerlebnis sich verändert, wenn Computertechnologie als Teil des Spielprinzips eingeführt wird. Konkret ist von besonderem Interesse, ob es möglich ist Prinzipien, die in Computer- und Videospiele genutzt werden um einen möglichst niederschweligen Zugang zum Spielerlebnis zu gewähren und damit das Studium einer Spielanleitung überflüssig zu machen, in die Domäne eines Tabletop-Spieles zu übertragen. Durch den Einsatz eines TUI soll sich der Umgang mit dem Spiel möglichst intuitiv gestalten und durch die Ausgestaltung als Tabletop-System ein ähnliches Spielerlebnis, wie es von Brettspielen bekannt ist, vermitteln und auch Spielern, die wenig Erfahrung mit Computertechnologie haben, einen niederschweligen Zugang ermöglichen werden.

# Grundlagen

## 2.1 Tangible User Interfaces

Unter einem Tangible User Interface (TUI) versteht man eine Art einer Benutzerschnittstelle. Im Gegensatz zu gängigen grafischen Benutzeroberflächen, die eine klare Trennung zwischen Eingabegeräten – wie Maus und Tastatur – und der grafischen Ausgabe vorsieht, besteht diese eindeutige Unterscheidung bei einem TUI nicht (vgl. [UI00]). Vielmehr dienen hier physische Objekte sowohl der Repräsentation, als auch als Mittel zur Manipulation der zugrunde liegenden Datenmodelle.

Während Graphical User Interfaces (GUIs) Information hauptsächlich in einer flüchtigen visuellen Gestalt präsentieren, wird bei TUIs die physische Repräsentation mit ihrer digitalen Form gekoppelt, sodass Systeme entstehen, die zwar computergestützt funktionieren, jedoch nicht auf den ersten Blick als Computer erkennbar sind.

### 2.1.1 Charakteristika von TUIs

Als zentrales Merkmal wird die Kopplung der physischen Repräsentation von Daten an die zugrunde liegenden Daten- und Rechenmodelle angesehen. Es können zum Beispiel die Position und Ausrichtung von Objekten, oder auch die relative Lage der einzelnen Objekte zueinander erfasst werden. Eine Änderung der Position hat direkten Einfluss auf das Datenmodell, auf dem das System basiert. So dienen die physische Repräsentationen der Daten auch gleichzeitig als Eingabemethode.

Im Gegensatz zu den grafischen Elementen in klassischen GUIs, können diese physischen Objekte nicht einfach im laufenden Betrieb erzeugt und wieder verworfen werden. Sie sind persistent und verkörpern einen Teil des Zustands, in welchem sich das System befindet und werden als solcher oft sowohl von dem/der BenutzerIn als auch vom System selbst “gelesen”.

Obwohl die physischen Elemente in TUIs eine zentrale Rolle spielen, so beruhen sie meist auf der Kombination dieser mit digitalen Repräsentationen, die meist den Hauptteil der

dynamischen Information darstellt, die vom zugrunde liegenden Computersystem zur Verfügung gestellt wird. (vgl. [UI00]).

### 2.2 Das Phänomen “Spiel”

Der Begriff des Spiels ist nur schwierig zu definieren und abzugrenzen. Das Phänomen des Spielens findet sich schon im Tierreich, wenngleich nur bei den am höchsten entwickelten Arten, den Wirbeltieren – hier vorwiegend bei Säugetieren – aber auch Vögeln. Es kann bei Tieren als ein Merkmal der Intelligenz verstanden werden. Die Fähigkeit eines Tieres zu spielen ist ausgeprägter, je intelligenter ein Tier ist. Auch wird bei manchen Tieren die Spielfähigkeit durch die Domestikation vergrößert.

Bei Tieren ist das Spiel dabei hauptsächlich eine Erscheinung der Jugendzeit. Von Jungtieren aller höherer Tierarten, die zu ihren Jungen eine Pflegebeziehung pflegen, kennt man eine Jugendzeit, in der die Jungen von den Aufgaben der eigenen Existenzsicherung, wie der Feindesabwehr oder Nahrungsbeschaffung, weitgehend entlastet sind. In dieser Zeit nehmen sich die Elterntiere respektive die Herden dieser Aufgaben an und dem jugendlichen Tier wird ein Spielraum gewährt, in dem es nicht nur spielen kann, sondern spielen muss. Diese Spielräume schwinden erst mit der Geschlechtsreife des Tiers (vgl. [Bal66])

Beim Menschen hat das Spiel eine ausschlaggebende Bedeutung für die Entstehung der Kultur. Da es auch in der Tierwelt auftritt, ist es älter als die Kultur selbst. Johann Huizinga ([Huj38]) stellt die Thesen auf, Kultur entstehe in Form von Spiel und echte Kultur können nicht ohne einen gewissen Spielgehalt bestehen. Demnach ist jede Gesellschaft und jede Form der Kultur vom Spiel geprägt. Die Dichtung, also das Experimentieren mit der Sprache sowie die bildenden und auch darstellenden Künste sind nichts anderes als ein Spiel. Huizinga definiert den Begriff des Spiels folgendermaßen:

Spiel ist eine freiwillige Handlung oder Beschäftigung, die innerhalb gewisser festgesetzter Grenzen von Zeit und Raum nach freiwillig angenommen, aber nach unbedingt bindenden Regeln verrichtet wird, ihr Ziel in sich selber hat und begleitet wird von einem Gefühl der Spannung und Freude und einem Bewußtsein [sic] des “Andersseins” als das “gewöhnliche Leben”.

Johan Huizinga [Huj38, 34]

Spielen dient also keinem weiteren Zweck, sondern wird um seiner selbst willen, freiwillig getan, allein weil es Vergnügen bereitet. Es hebt sich allein dadurch schon vom gewöhnlichen Leben ab, das von einem Prozess der unmittelbaren Befriedigung von Notwendigkeiten bestimmt ist.

Auch die Tatsache, dass es sich auf einen bestimmten Ort und einen festgelegten Zeitraum beschränkt und auf einem Rahmen aus Regeln basiert, die nur in diesem Zusammenhang ihre Bedeutung haben, zeigt, wie das Spiel vom gewöhnlichen Leben abgegrenzt ist. Beim Spielen wird so “getan als ob”. Es wird kein direkter Nutzen aus dem Spiel gezogen. Es

dient keinem materiellem Interesse und kann dennoch den/die SpielerIn vollkommen ausfüllen.

### 2.2.1 Was macht ein gutes Spiel aus?

Es ist nicht einfach zu analysieren oder zu benennen, was ein gutes Spiel ausmacht. Es herrscht allerdings weitgehend Einigkeit darüber, dass der zentrale Fokus auf die Konzeption und Balance der Spielmechanik gelegt werden sollte. Dies gilt für sämtliche Arten von Spielen. Daneben existieren allerdings noch andere Faktoren, die das ihrige zum Erleben des Spiels und der Einbindung der SpielerInnen beitragen.

#### Charakteristika von Spielen

Computerspiele, wenn sie denn gut sind, vereinigen in sich zwölf Faktoren, die maßgeblich dazu beitragen, dass es sich bei jenen um den Wahrscheinlich einnehmendsten Zeitvertreib überhaupt handelt [Pre01].

1. Spaß – Sie bereiten uns Vergnügen und Freude
2. Spiel (vgl. englisch *play*) – Eine starke leidenschaftliche Einbindung
3. Regeln – Sie stellen eine Struktur bereit
4. Ziele – Sorgen für Motivation
5. Interaktivität – gibt uns die Möglichkeit zu Handeln
6. Adaptivität – sorgt für Flow
7. Folgen und Feedback – ermöglicht uns zu lernen
8. Gewinnsszenarien – geben uns Selbstbestätigung
9. Konflikt/Konkurrenz/Herausforderung – sorgen für Adrenalin
10. Problemlösung – regt die Kreativität an
11. Interaktion – stellt soziale Gruppen her
12. Darstellung und Geschichten – erzeugen Emotionen

Während Spaß und Spiel einerseits Charakteristika eines Spiels darstellen, ergeben sie sich andererseits aus der richtigen Mischung der zehn übrigen Faktoren. Sie sind also gleichsam Charakteristika eines Spiels, wie auch Ziel des Designprozesses. Sie sorgen dafür, dass sich die SpielerInnen entspannen und motivieren sie. Spielen ist außerdem ein effektiver Weg des Lernens, laut Diane Ackerman sogar die bevorzugte Art des menschlichen Gehirns neue Dinge zu lernen [Ack11]. Sich Aufgaben spielerisch zu widmen kann die Konzentration erhöhen sowie das Durchhaltevermögen steigern. Gleichzeitig

empfinden Personen, die spielerische Aufgaben erledigen Vergnügen dabei.

Der Grund, warum Spiele uns dermaßen einnehmen ist einerseits, weil sie eine Form von Spiel und Spaß darstellen, andererseits durch weitere strukturelle Eigenschaften charakterisiert werden, die zu ihrer Faszination beitragen.

**Regeln** Sie sind ein definierendes Element eines Spiels. Ohne Regeln handelt es sich lediglich um freies spielen, nicht jedoch um ein Spiel.

**Ziele oder Vorgaben** Die Erreichung eines definierten Ziels stellt in Spielen einen wichtigen Aspekt der Motivation dar. Dies ist wichtig, weil der Mensch eine ergebnisorientierte Spezies ist, die im Gegensatz zu den meisten anderen Tieren, einen zukünftigen Zustand antizipieren kann und Strategien für dessen Erreichung entwickelt.

**Folgen und Feedback** Feedback ist ein Element, das vor allem Computerspielen zu eigen ist. Es entsteht, wenn es im Spiel eine Veränderung gibt, ausgelöst durch etwas, dass der/die SpielerIn getan hat. Den SpielerInnen wird unmittelbar mitgeteilt, ob ihre Handlungen positive oder negative Auswirkungen haben und ob sie sich innerhalb der Regeln bewegen oder diese brechen. Außerdem gibt es Hinweis darauf, ob man sich der Erreichung des Ziels annähert oder davon entfernt.

Eine Form des Feedbacks ist der Spielstand oder die Punktezahl. Feedback kann aber auch in grafischer oder auditiver Form vorliegen. Sogar taktiles Feedback ist möglich, wie es zum Beispiel durch Kraftrückkopplung in Form von Vibrationen in Joysticks und Gamepads genutzt wird. Das Feedback erlaubt den SpielerInnen zu lernen. Sie werden dadurch entweder belohnt, wenn sie etwas richtig gemacht haben, wie eine Aufgabe zu meistern, erhalten aber auch Hinweise darauf, wenn sie ein Ziel verfehlen. Die Dosierung dieser Hinweise ist dabei von größter Wichtigkeit, da ein Zuviel oder zu Zuwenig an Feedback bei den SpielerInnen rasch zu Frustration führen kann.

**Konflikt/Konkurrenz/Herausforderung** Die Herausforderung in einem Spiel ist eine Aufgabe oder Problem, das die Spieler zu lösen haben. Die meisten Menschen haben Spaß an Herausforderungen, besonders wenn sie den Schwierigkeitsgrad selbst bestimmen können.

Die allermeisten Spiele enthalten ein Element der Herausforderung oder des Konflikts, auch wenn diese teilweise durch Kooperation der SpielerInnen gelöst werden. Ein Spiel selbst ist *sicher*, im Sinne von "den Spieler nicht gefährdend". Es handelt sich eben nicht um reale Bedrohungen oder Konflikte, die im Spiel ausgefochten werden. Dies gilt allerdings nicht für die erzeugten Emotionen der Spielenden, die dabei oft sehr real sind.

**Interaktion** Es gibt für Computerspiele zwei Aspekte der Interaktion. Einerseits die Interaktion des Spielers mit dem System, die über das Interface stattfindet, andererseits die Interaktion von Spielern untereinander, die den sozialen Aspekt des

Spieles darstellen. Spiel fördert die Formierung sozialer Gruppen. Die Tendenz, die mittlerweile besteht, möglichst allen Spielen einen Multiplayer-Modus zu geben, ist ein Indiz dafür, dass das Spiel mit anderen meist kurzweiliger ist, als wenn alleine gespielt wird.

**Darstellung** Darstellung bedeutet, das Spiel handelt von *etwas*, es wird also zum Beispiel eine Tätigkeit *dargestellt*. Die Darstellung beinhaltet alle Geschichten und Schilderungen, in die das Spielgeschehen verwoben ist, um diesem einen Rahmen zu geben.

### Dem Spiel Bedeutung geben

Wie bereits festgestellt, gibt es zahlreiche Definitionen dafür, was ein Spiel ausmacht. Eine Definition, die sich besonders für Gesellschafts- aber auch für Computerspiele aufdrängt ist jene von Katie Salen und Eric Zimmerman.

A game is a system in which players engage in artificial conflict, defined by rules, that results in a quantifiable outcome.

Katie Salen und Eric Zimmerman [SZ04]

Sie heben in dieser Definition die Wichtigkeit des Regelsystems, auf dem ein Spiel basiert, klar hervor. Ein jedes Spiel ist also durch seine Regeln definiert, denen sich die SpielerInnen unterwerfen. Sie besagten was im Rahmen des Spiels erlaubte Aktionen darstellt und was verboten ist und stecken den SpielerInnen Ziele, die es für sie zu erreichen gilt.

Gleichzeitig führen sie den Begriff des *meaningful play* als das vorrangige Ziel des Designprozesses von Spielen ein. Es soll dem Spielgeschehen also eine Bedeutung gegeben werden. Damit ist das Zusammenwirken der Aktionen, die von den SpielerInnen gesetzt werden, und den Konsequenzen, die sich im durch das Spiel dargestellten, System daraus ergeben, gemeint. Die Bedeutung einer Aktion entsteht also in ihrer Beziehung zu den sich daraus ergebenden Folgen.

Dabei wird zwischen den direkten Folgen einer Aktion unterschieden, die unmittelbar aus ihr resultieren, und jenen Konsequenzen, die sich aus den einzelnen Entscheidungen, die die SpielerInnen treffen, für den Gesamtverlauf ergeben.

Die Bedeutung erhält das Spiel im Zuge des Designprozesses. Sie resultiert aus den den SpielerInnen vom Spieldesigner zur Verfügung gestellten Entscheidungsmöglichkeiten und den vorgesehenen Reaktionen des Systems darauf.

### Unberechenbarkeit

Während das Ziel des Spiels vorher definiert sein muss, so sollte der Ausgang des Spiels selbst nicht vorhersehbar sein. Es sollte also möglichst zu keinem Zeitpunkt im Verlauf eines Spiel der Ausgang desselben berechenbar sein. [AD12]

Natürlich ist der Ausgang des Spiels durch die Regeln determiniert, sodass zwei Spiele, die dem selben Ablauf folgen auch zum selben Ergebnis führen müssen. Es ist also

genau jener Ablauf, der nicht voraussagbar sein darf. Dies erreicht man zum Beispiel durch die Nutzung einer Zufallskomponente. In Brettspielen wird dies oft durch das Werfen eines Würfels gelöst, in Kartenspielen erfüllt das Abheben von Karten von einem zufällig durchmischten Stapel den gleichen Zweck. Eine weitere Möglichkeit ergibt sich daraus, den Entscheidungen der SpielerInnen Einfluss auf den Spielverlauf für die anderen MitspielerInnen zu geben. So wirkt sich etwa bei dem Kartenspiel *Uno* die zuletzt abgelegte Karte stark auf die Handlungsmöglichkeiten des/derjenigen aus, der/die als nächstes am Zug ist.

### Muster und Strukturen

Based on my reading, the human brain is mostly a voracious consumer of patterns, a soft pudgy gray Pac-Man of concepts. Games are just exceptionally tasty patterns to eat up.

Raph Koster [Kos13]

Die Faszination, die von Spielen ausgeht, führt Raph Koster hauptsächlich auf die natürliche Tendenz des menschlichen Gehirns zurück, Muster und Strukturen zu erkennen. Als Beispiel dafür, dass der Mensch quasi von Natur aus darauf “programmiert” ist, Muster zu erkennen, nennt er zum Beispiel die Tatsache, dass Menschen dazu tendieren in den verschiedensten leblosen Objekten Gesichter wahrzunehmen.

Demnach seien im Grunde alle Spiele Rätsel, die die Routinen des Gehirns zur Musterkennung in Anspruch nehmen. Somit stellen Spiele – so abstrakt und fern von der tatsächlichen Lebensrealität sie auch sein mögen – für das Gehirn reale Probleme dar, die genauso wie die entsprechenden Herausforderungen, auf die man im normalen Tagesablauf stoßen würde, von ihm behandelt werden. Sie unterscheiden sich vom realen Leben nur durch die Tatsache, dass die SpielerInnen im Regelfall nicht mit entsprechenden Konsequenzen für ihr Leben rechnen müssen. Spiele sind also für das Gehirn bereits aufbereitete Aufgaben in Form formaler Systeme, die sich, durch die Abstraktion und das Weglassen von ablenkenden nicht essentiellen Details, ideal als Hilfsmittel zum Erlernen bestimmter Fähigkeiten eignen. Lernt man etwas neues oder meistert eine Aufgabe, werden im Gehirn Endorphine freigesetzt, die wie körpereigene Drogen wirken und so für das erhebende Gefühl verantwortlich sind, das man dabei bekommt.

Das führt allerdings auch dazu, dass Spiele nur solange interessant bleiben, wie man sie nicht durchschaut beziehungsweise perfekt beherrscht. Als Beispiel dient hier das simple Spiel “Tic Tac Toe”. Hat man einmal die stark beschränkten Möglichkeiten des Spielverlaufs durchschaut, so verliert das Spiel jeglichen Reiz, da es berechenbar wird. Dabei bedeutet das nicht notwendigerweise, dass das Gehirn durch völlig neue Erfahrungen entsprechend stimuliert wird. Vielmehr erzeugen neue Facetten von bereits Bekanntem die beschriebenen Reaktionen.

## 2.3 Usability im Kontext von Computerspielen

### 2.3.1 Affordance

*Affordance*, zu deutsch in etwa “Angebotscharakter”, ist ein Konzept, das von James J. Gibson eingeführt wurde, der sich mit der Wahrnehmungstheorie beschäftigte. Gibson verortet die Wahrnehmung auf der Ebene von Medien, Oberflächen und Materialien, anstatt auf der Ebene von Partikeln und Atomen. Insbesondere gehe es bei der Wahrnehmung darum, was die Umwelt einem Individuum  *anbietet*, also welche Interaktionsmöglichkeiten sie ihm zur Verfügung stellt [Gib79].

Donald Norman übernahm Gibsons Konzept und popularisierte es im HCI-Kontext, wich aber von Gibsons ursprünglicher Definition ab [Nor88]. So bezieht sich Norman nur auf wahrgenommene *Affordances*, also jene, die dem Individuum bekannt sind, während Gibson auch jene *Affordances* betrachtet, die nicht offensichtlich sind. So kann ein Buch einerseits gelesen, umgeblättert und zugeschlagen werden (wahrgenommene *Affordances*), andererseits lässt es sich auch werfen, zerreißen oder darauf stehen.

Für Gibson stellt *Affordance* die Aktionsmöglichkeit eines Akteurs da, unabhängig davon ob dieser sie selbst wahrnimmt. Sie ist zwar von den Möglichkeiten des Akteurs abhängig – zum Beispiel kann ein Tisch eine Person tragen, eine zweite schwerere Person nicht – nicht aber von dessen Erfahrungen oder Vorkenntnissen. Der Akteur dient als Bezugsrahmen, der für Gibson gemeinsam mit der Umwelt ein untrennbares Paar bildet. Gibson konzentrierte seine Arbeit auf direkte Wahrnehmung, die dann möglich ist, wenn eine *Affordance* existiert und gleichzeitig in der Umgebung Information vorlegt, die eindeutig diese *Affordance* definiert. Es gibt Fälle in denen eine *Affordance* vorliegt, aber keine Informationen um auf diese zu schließen. So ist bietet eine Geheimtür die Möglichkeit des Durchschreitens, für einen Akteur existiert jedoch kein Hinweis darauf, dass diese Möglichkeit gegeben ist.

Norman weicht von Gibsons Verständnis ab, indem er die wahrgenommenen und tatsächlichen Eigenschaften eines Dings als *Affordances* bezeichnet und impliziert damit, dass es wahrgenommene Eigenschaften geben kann, die nicht tatsächlich existieren. Diese werden von Norman allerdings dennoch als *Affordances* bezeichnet. Er koppelt *Affordances* an das Vorwissen und die Erfahrungen der Akteure. Laut Norman geben *Affordances* Hinweise darauf wie Dinge bedient werden und welche Möglichkeiten gegeben sind. Wenn Designer sich also ihrer bedienen, wissen die BenutzerInnen durch bloßes Hinsehen was zu tun ist. Dabei unterscheidet er die tatsächlich existenten *Affordances* und jene, die lediglich wahrgenommenen werden, jedoch nicht wirklich existieren. Er legt den Fokus darauf, wie dem/der BenutzerIn die Funktionen vermittelt werden und betont, dass die wahrgenommenen *Affordances* ausschlaggebend für die Benutzbarkeit sind [Nor98][123]. McGrenere und Ho, die versuchen eine einheitliche *Affordance*-Definition zu etablieren, stellen im Bezug auf HCIs, neben den genannten, die Arbeit von Gaver heraus [MH00]. Gavers Arbeit beruht auf den Ideen, die von Gibson formuliert wurden, die er versucht, im HCI-Kontext einzubringen. Während Norman davon spricht, dass die *Affordance* die Handlungsmöglichkeit suggeriert, ist es bei Gaver das Design, dass die *Affordance*

nahe legt [Gav91]. Anhand einer Tür erklärt er *Nested Affordances*, also *Affordances*, die ineinander verschachtelt beziehungsweise voneinander abhängig sind – eine Tatsache, die Gibson zwar impliziert, aber nicht ausführt. So ist das Ziehen an einem Türgriff verschachtelt im Öffnen der Tür und beide stellen *Affordances* der Tür dar. Gaver schlug eine Trennung der *Affordances* und der darüber verfügbaren Information von der Art, in welcher diese dann tatsächlich erkannt werden, vor. Ein weiteres Konzept, das Gaver einführt, sind *Sequential Affordances*. Dabei handelt es sich um eine Erklärung dafür, wie verschiedene *Affordances* erst mit der Zeit entdeckt werden.

McGrenere und Ho stellen abschließend den Unterschied zwischen dem Nutzwert eines Objekts und dessen Benutzbarkeit heraus und stellen fest, dass im HCI-Umfeld größeres Augenmerk auf Usability als auf den eigentlichen Nutzen gelegt wird. Ihrer Meinung nach müssen DesignerInnen den BenutzerInnen die Aktionen zur Verfügung stellen, die diese für die Erledigung ihrer Aufgaben benötigen [MH00]. Diese stellen die *Affordances* des Designs dar. Die Frage ob die gegebenen Möglichkeiten dem Entsprechen, was die BenutzerInnen erreichen wollen, entscheidet über den Nutzwert der Software. Die Usability einer Software wiederum, kann dadurch verbessert werden, klar wahrnehmbare Information zur Verfügung zu stellen, die ebendiese *Affordances* beschreiben. Eine zugrundeliegende *Affordance* kann unabhängig davon existieren, ob sie von den BenutzerInnen korrekt interpretiert wird oder überhaupt als existent wahrgenommen werden. Als Beispiel dient eine Schaltfläche, die zwar selbst durch ihr erhaben wirkendes Aussehen die *Affordance* der Klickbarkeit vermittelt, aber von den BenutzerInnen nicht um deren selbst willen geklickt werden, sondern um die damit verknüpfte Funktionalität aufzurufen.

### 2.3.2 Ästhetik

Schon in der Antike beschäftigten sich Philosophen wie Platon und Aristoteles mit der Ästhetik, die meist als Theorie des Schönen oder als Philosophie der Kunst verstanden wird. Dabei kann der Begriff Ästhetik einerseits die objektiven Eigenschaften eines Stimulus meinen (wie zum Beispiel Form oder Farbe), andererseits auch die subjektiven Reaktionen, die diese Eigenschaften auslösen. Mittlerweile hat das Interesse für die Ästhetik in die verschiedensten wissenschaftlichen Disziplinen Einzug gehalten. Seit relativ kurzer Zeit wird nun der Einfluss und die Wirkung der Ästhetik eines Systems auf den/die BenutzerIn in der HCI untersucht. Norman schreibt zum Beispiel 1988 in *The psychology of everyday things* [Nor88], Designer würden oft den Fehler machen die Ästhetik an erste Stelle zu setzen und dahinter die Funktionalität zu vergessen. Er legt den Fokus ausschließlich auf Usability und missachtet dabei die Bedeutung der ästhetischen Aspekte von Design. Diese Herangehensweise war und ist weit verbreitet und spiegelt sich auch im Designleitsatz “Form follows function” wider.

Diese Form der Vernachlässigung der Ästhetik im Bereich von Design wird mittlerweile von zahlreichen Designern kritisiert. Auch Norman selbst behandelt in seiner jüngeren Arbeit *emotionales Design* und geht auf die Tatsache eine, dass Dinge, wie auch Personen, in Menschen Emotionen wecken könne, die sich auf die Funktionsweise des Gehirn auswirken. Positive Emotionen lösen mentale Vorgänge aus, die zu einer größeren Offenheit und Toleranz führen. Menschen in einem derartigen Zustand sehen eher das große Ganze,

als sich auf auf kleinere Schwierigkeiten zu versteifen [Nor05].

Dieser Umstand ist eine mögliche Erklärung, für die Ergebnisse verschiedener Studien, die den Zusammenhang zwischen der Ästhetik und der Empfundene Usability eines Systems untersucht haben und eine Korrelation zwischen den beiden Faktoren feststellen konnten. Kurosu und Kashimura stellten in ihrem Experiment, in dem sie Testpersonen die Benutzeroberfläche eines Geldautomaten nach den Kriterien der wahrgenommenen Usability und Ästhetik bewerten ließen, einen starken Zusammenhang zwischen diesen beiden Kriterien fest [KK95]. Tractinsky wiederholte den Versuch, der ursprünglich in Japan durchgeführt worden war, mit leichten Anpassungen des Aufbaus in Israel. Ursprünglich ging er davon aus, das Ergebnis könne auf die Japanische Kultur zurückgeführt werden, in der Ästhetik ein hoher Wert beigemessen wird [Tra97]. Die Ergebnisse stützten die Erkenntnisse des ursprünglichen Experiments. Die Korrelation zwischen der Ästhetik und der wahrgenommenen Usability stellte sich als noch höher dar, als im ursprünglichen Versuch. Tractinsky et al. wiederholten den Versuch später ein weiteres Mal. Zusätzlich zu den vorangegangenen Versuchen, die sich darauf beschränkten zu untersuchen, wie die Ästhetik die Einschätzung der Usability beeinflusst bevor die BenutzerInnen das System genutzt hatten, sollte ermittelt werden, ob die Einschätzungen der Ästhetik und Usability auch nach einer längeren Nutzung des Systems fortbestehen und wie sie von der tatsächlichen Usability abhängen. Es stellte sich heraus, dass die erste Einschätzung der Usability auch nach dem Umgang mit dem System stark von der Ästhetik der Oberfläche abhing, was den Autoren zufolge Grund gibt, die uneingeschränkte Empfehlung der HCI-Literatur, der Usability einen höheren Stellenwert als der Ästhetik beizumessen, infrage zu stellen [TKI00].

Während sich die meisten Studien auf die subjektiv empfundene Usability stützen, gibt es nur wenige Studien, die objektiv messbare Kriterien, wie Effizienz, heranziehen. Die Ergebnisse dieser Studien sind dabei nicht konsistent. So konnten Ben-Bassat et. al. eine Herabsetzung der Effizienz durch eine ansprechendere Gestaltung der Oberfläche feststellen [BBMT06]. Zu ähnlichen Erkenntnissen kamen auch Sauer und Sonderegger in ihrer Studie, bei dem das Aussehen eines Mobiltelefons experimentell manipuliert wurden [SS09]. Andere Studien wiederum konnten keinen Zusammenhang zwischen der Bearbeitungszeit und der Ästhetik feststellen [HSDA07, TM07]. Sonderegger und Sauer, die in einer weiteren Studie eine gesteigerte Effizienz durch eine höhere wahrgenommene Attraktivität feststellen konnten, weisen darauf hin, dass sich die Diskrepanzen zwischen den Ergebnissen der verschiedenen Studien möglicherweise durch zwei verschiedene Phänomene erklären lassen. Einerseits komme es durch eine ästhetischer wahrgenommene Oberfläche zu einem Effekt der "Ausdehnung der angenehmen Erfahrung", wodurch sich die herabgesetzte Effizienz erklären ließe, andererseits fanden sie in ihrer Studie auch Resultate, die den Effekt der "erhöhten Motivation" stützen [SS10], wonach ästhetisch Ansprechende Produkte die BenutzerInnen beruhigen [Lin07] oder ihnen helfen in einen Flow-Zustand zu gelangen [Flo97].

### 2.3.3 Learnability

Learnability als ein wichtiger Teilaspekt der Usability von Software, bezieht sich grob darauf, wie einfach und effizient ein/e BenutzerIn, der/die noch nie mit der betreffenden Software zu tun hatte, deren Benutzung erlernen kann, um eine vorgegebene Aufgabe zu erfüllen.

Trotz dieses relativ simplen Ansatzes gibt es eine Vielzahl unterschiedlicher Definitionen des Begriffs. Ebenso herrscht Uneinigkeit darüber wie Learnability evaluiert und anhand welchen Kriterien sie gemessen werden soll. Die verschiedenen Methoden zur Ermittlung von Learnability unterscheiden sich einerseits darin, welche Kriterien zur Evaluierung herangezogen werden, andererseits in Bezug auf die Zeitperiode, die im Zuge der Evaluierung betrachtet wird, wie Grossman et. al. in ihrer Überblicksstudie festgestellt haben [GFA09]. So definiert Nielsen Learnability als die Wahrnehmung der Lernkurve eines Neulings beim ersten Kontakt mit einer Software [Nie94]. Dieser Aspekt, also der Fokus auf die Erfahrungen eines Benutzers in der ersten Phase des Erlernens – auch *Initial Learning* – des Umgangs mit der Software ist dabei üblich, wenn auch nicht selbstverständlich. Es existieren ebenso Tendenzen ein Augenmerk auf das erweiterte Lernen (*extended learning*), das nach längerer Nutzung eines Programmes auftritt, wenn die Kenntnisse vertieft werden und die BenutzerInnen zum Beispiel effizientere Wege finden ihre Aufgaben zu erfüllen, zu legen. Mit ihrer Definition als “die Klarheit, mit der neue Benutzer einen effektive Umgang beginnen können und eine maximale Leistung erreichen”, weisen Dix et. al. auf die Aspekte des erweiterten Lernens hin [DFAB03].

In der ISO/IEC 9126 wird Learnability definiert als “[d]ie Fähigkeit eines Softwareprodukts, einem Benutzer das Erlernen der Anwendung leicht zu machen” [ISO01]. Diese Definition geht wiederum nicht näher auf den zu betrachtenden Zeitabschnitt der Benutzung einer Software ein. Grossman et. al. haben eine Taxonomie der Learnability entwickelt, bei der einerseits nach dem untersuchten Zeitbereich unterschieden wird, andererseits auch die Erfahrung und Vorkenntnisse der BenutzerInnen eingegangen wird [GFA09]. So werden Benutzer anhand folgender Kennwerte unterschieden:

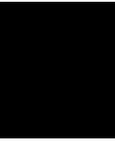
- Ihrer Erfahrung im Umgang mit Computern
- Die Erfahrung mit dem fraglichen Interface
- Der Umfang ihres Fachwissens
- Vorerfahrungen mit vergleichbarer Software

Im Rahmen dieses Projekts ist hauptsächlich die erste Phase des Kontakts mit dem System, bei dem die BenutzerInnen die Grundregeln des Spiels erlernen sollen, von Interesse. Weil es sich bei der zu implementierenden Software um ein Spiel handelt, dessen Interface für die BenutzerInnen keine Möglichkeiten einer Effizienzsteigerung bei der Eingabe – zum Beispiel durch die Verwendung alternativer Eingabemethoden (vgl. die Nutzung von Tastenkombinationen anstatt Befehle in Menüs aufzurufen in klassischen WIMP-Applikationen) bietet – beschränkt sich die Evaluierung auf das *Initial Learning*.

### **Evaluierung von Learnability**

Für die Evaluierung der Learnability von Userinterfaces werden unterschiedliche Ansätze verfolgt. Einerseits ist eine heuristische Evaluierung möglich, bei der ExpertInnen das System anhand eines Katalogs bekannter Richtlinien beurteilen. Jede nicht beachtete Richtlinie ist dabei ein mögliches Problem mit der Learnability. Eine weitere Methode ist die Learnability anhand messbarer Kriterien zu bewerten. Dazu gehören einerseits zum Beispiel die Durchführungszeit einer Vorgegebenen Aufgabe, die Zahl der dabei gemachten Fehler oder die Anzahl der zur Erfüllung der Aufgabe nötigen durchgeführten Operationen. Santos und Badre schlagen eine Methode vor, die auf dem automatisierten Mitschneiden der Benutzerinteraktion beruht. Sie betrachten dabei die Dauer und Häufigkeit der Pausen, die die BenutzerInnen bei der Interaktion mit dem System einlegen [SB95]. Diese Pausen erlauben Rückschlüsse auf die Art wie BenutzerInnen die einzelnen Komponenten ihrer mentalen Prozesse organisieren. Bei der Interaktion mit dem Computer halten sich die BenutzerInnen normalerweise zunächst ihr Ziel vor Augen und formulieren dann eine Strategie um dieses zu erreichen bevor sie deren Umsetzung mithilfe des Computers planen. In dieser Erfassungsphase kommt es üblicherweise zu keiner physischen Interaktion mit dem System. Sobald die Erfassungsphase abgeschlossen ist fährt der/die BenutzerIn mit der Ausführungsphase fort, in der er/sie die nötigen Schritte ausführt um sein/ihr Ziel mit dem Computer zu erreichen. Sobald das Ziel erreicht wurde oder der/die BenutzerIn daran gescheitert ist es zu erreichen, wiederholt sich dieser Kreis und der/die BenutzerIn geht wieder in die Erfassungsphase über. Während der Ausführung ist der Ausführungsplan im Kurzzeitgedächtnis gespeichert und wird als Chunk bezeichnet. Dabei handelt es sich um Gruppierungen von Komponenten mentaler Vorgänge. Anhand eines Modells werden Pausen in der Eingabe automatisch festgestellt und dort wo eine Pause detektiert wird, nimmt man an die Grenze eines Chunks an. Es hat sich gezeigt, dass die Chunks mit zunehmender Erfahrung der BenutzerInnen größer werden. So kann man die Entwicklung der Größe der Chunks als einen Indikator für die Learnability heranziehen. Eine Methode, die auf die subjektiven Eindrücke der Benutzer eingeht, ist der Thinking-Aloud-Test. Dabei wird eine oder mehrere Zielgruppen zusammengestellt aus denen jeweils vier bis fünf Testpersonen ausgewählt werden. Diese haben dann auf einem Testsystem eine oder mehrere ihnen gestellte Aufgaben zu lösen. Sie werden aufgefordert dabei laut mitzusprechen und werden dabei gefilmt. Eine Evaluierung der Effizienz des Testsystems wird dabei durch die Tatsache, dass das laute Mitdenken Verzögerungen verursacht, verunmöglicht. Ein Problem, das sich bei einem Thinking-Aloud-Test stellen kann, ist dass die Testperson mit der Zeit verstummt, weil sie nicht gewohnt ist, das was sie denkt zu kommentieren [MYJ94]. Diese Einschränkung lässt sich zum Beispiel durch die gleichzeitige Evaluierungen von zwei Testpersonen umgehen, die gebeten werden untereinander ihre Gedanken auszutauschen.





# Spielkonzept

Die Entwicklung des Prototypen fußt auf dem Konzept eines eigens entwickelten Legespiels für zwei Personen, das mit nur wenigen simplen Regeln auskommt.

Die TeilnehmerInnen ziehen abwechselnd per Zufallsprinzip jeweils einen Spielstein und legen diesen auf dem Spielfeld an die bereits abgelegten Spielsteine an.

Für den/die SpielerIn gilt es dabei ihren jeweils gezogenen Spielstein so an die bereits gelegten Steine anzulegen, dass sich dadurch ein *Raum* ergibt. Ein *Raum* ist in diesem Kontext durch die Eigenschaft gekennzeichnet, auf allen Seiten von einer *Wand* umgeben zu sein. Je größer ein *Raum* ist, desto mehr Punkte erhält der/diejenige SpielerIn, der/die den letzten Stein anlegt, mit dem ein Bereich von Spielsteinen vollständig von *Wänden* umgeben ist, wodurch ein *Raum* entsteht.

Die Tatsache, dass die Punkteanzahl steigt, wenn die *Räume* größere Ausmaße annehmen, soll dazu motivieren, möglichst große *Räume* zu konstruieren. Die Regel, die besagt, dass dem/derjenigen SpielerIn sämtliche Punkte für einen abgeschlossenen *Raum* zukommen, der/die den letzten Spielstein zur Komplettierung der umgebenden *Wand* angelegt hat, führt hingegen dazu, dass die SpielerInnen mit der Möglichkeit taktieren müssen, dass der/die GegenspielerIn die Punkte einheimst, die er/sie sich selbst versprochen hat.

## 3.1 Begriffsdefinitionen

### 3.1.1 Wand

Eine *Wand*, die auf den Spielsteinen entweder durchgehend (vgl. 3.1: Spielstein E) oder aber mit einer *Tür* (vgl. 3.1: Spielstein F) dargestellt wird, existiert an maximal einer Seite des Spielsteins. Sie erlaubt es einen beliebigen anderen Spielstein – unabhängig von dessen Farbe – an diese Seite anzulegen. Dabei gilt, dass die *Wandseite* eines Steins nicht direkt an die *Wandseite* eines anderen Steins stoßen darf.

#### 3.1.2 Erreichbarkeit

Die Erreichbarkeit bezieht sich in diesem Kontext auf die Beziehung zwischen zwei Spielsteinen zueinander. Ein Spielstein ist dann von einem anderen Spielstein aus erreichbar, wenn zwischen diesen beiden Steinen ein Pfad existiert. Das bedeutet es ist möglich ausgehend von Stein A zum Stein B zu gelangen, indem man beliebig oft von einem Nachbarn des Steins A, zu welchem eine Verbindung besteht (das heißt es befindet sich keine *Wand* zwischen den Steinen oder, falls eine *Wand* zwischen den Steinen existiert, befindet sich eine *Tür* in dieser *Wand*) zu einem an diesen Nachbarn angrenzenden Stein “springen” kann. Von diesem ausgehend springt man nun wieder zu einem von dessen Nachbarn (es gelten genannte Bedingungen) usw. bis man bei Stein B angelangt ist. Ist dies nicht möglich, so ist Stein B von Stein A aus nicht erreichbar. Diese Beziehung ist bidirektional.

In dem in Abbildung 3.1 dargestellten Beispiel, sind sowohl der Stein B als auch der Stein E von A aus erreichbar (neben C, F und G), nicht jedoch der Stein D. Die Erreichbarkeit des Steins E von Stein A ist durch folgenden Pfad, der zwischen den Steinen existiert, gegeben:  $A \rightarrow B \rightarrow C \rightarrow G \rightarrow F \rightarrow E$ .

#### 3.1.3 Raum

Ein *Raum* ist eine zusammenhängende, das heißt aneinander angrenzende, Anordnung von Spielsteinen der gleichen Farbe. Jeder Spielstein eines *Raumes* muss dabei von jedem anderen Stein, der zum selben *Raum* gehört, erreichbar sein, ohne dass dabei eine *Wand* überwunden werden muss.

Ist diese Anordnung auf allen Seiten von einer *Wand* umgeben, das heißt z.B. für den nördlichen Rand eines *Raumes*, dass alle Randsteine entweder mit ihrer *Wand*-Seite nach Norden ausgerichtet sein müssen, oder aber mit ihrer nach Norden ausgerichteten Seite jeweils an die *Wand*-Seite eines anderen Steins stoßen. Dies gilt jeweils entsprechend für die übrigen Seiten des *Raumes*, so das man vereinfacht sagen kann, dass der *Raum* von *Wänden* umgeben ist. Es ist also nicht möglich von einem Stein eines *Raumes* zu einem angrenzenden, nicht zum *Raum* gehörenden Stein zu gelangen, ohne dabei auf eine *Wand* zu stoßen. Ist gleichzeitig die Bedingung erfüllt, dass jeder Stein, der zum *Raum* gehört an allen offenen Seiten an einen anderen Spielstein angrenzt, so ist dieser *Raum* vollständig.

Der/diejenige SpielerIn, der/die den letzten Stein angelegt hat, der dafür sorgt, dass genannte Bedingungen erfüllt werden, erhält die Punkte für den *Raum* gutgeschrieben.

Steine F und G in 3.1 bilden gemeinsam einen abgeschlossenen *Raum*. Weitere *Räume* werden jeweils durch die Steine A, B und C, die Steine D und H, sowie die Steine I, J und K gebildet. Der Stein E ist nicht Teil des *Raumes*, der von I, J und K gebildet wird, obwohl er daran angrenzt und die gleiche Farbe hat, da sich eine *Wand* zwischen E und I befindet.

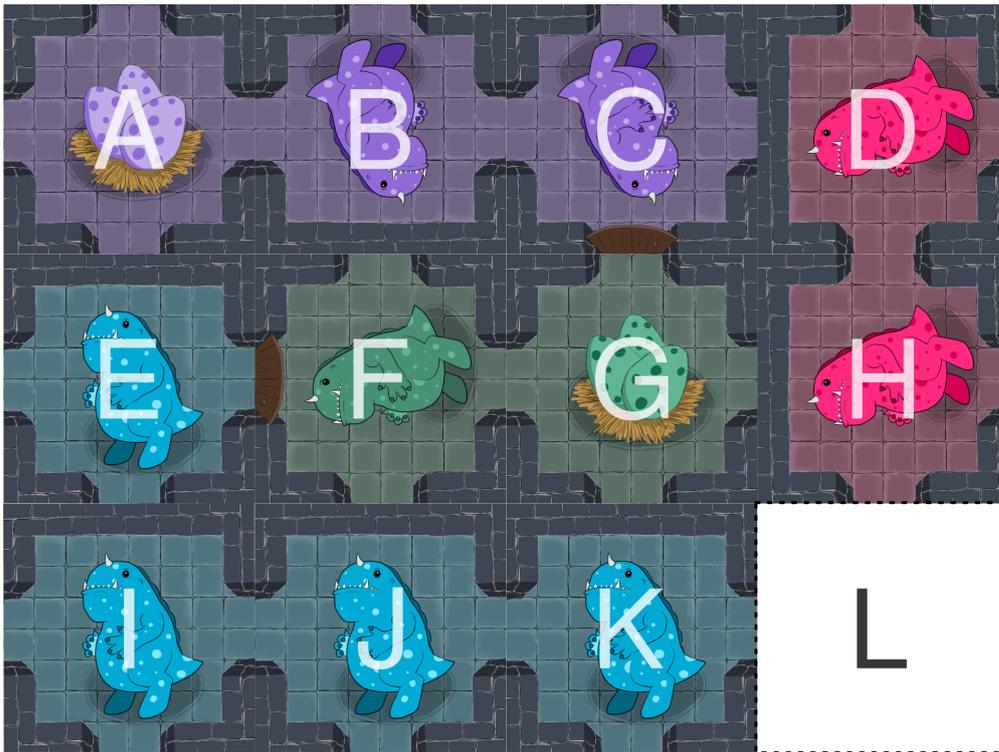


Abbildung 3.1: Eine mögliche gültige Anordnung von Spielsteinen

## 3.2 Spielsteine

Zwölf unterschiedliche Arten von quadratischen, jeweils gleich großen, Spielsteinen bilden die Grundlage des Spiels, wobei drei Typen unterschieden werden, die in jeweils 4 unterschiedlichen Farben vorliegen.

Die Eigenschaften der unterschiedlichen Spielsteintypen werden im Folgenden angeführt.

**Wandsteine** Sie sind an 3 von 4 Seiten offen, während sich an der vierten Seite eine *Wand* befindet (vgl. 3.1: z.B. Steine B und D).

**Türsteine** Wie die *Wandsteine* sind sie an 3 Seiten offen und haben an der vierten Seite eine *Wand*, in der sich jedoch eine *Tür* befindet. Diese *Tür* ermöglicht nach den Prinzipien des Spiels ein *Durchgehen*, womit ein an der *Türseite* platzierter Spielstein vom aktuellen Spielstein aus *erreichbar* ist (vgl. 3.1: Steine C und F)

**Bodensteine** Diese Spielsteine sind an allen 4 Seiten offen (vgl. 3.1: Steine A und G).

## 3.3 Spielregeln

### 3.3.1 Spielziel

Das Spielziel ist es nach einer vorher festgelegten Zeit mehr Punkte zu erhalten als der/die GegenspielerIn. Dies wird erreicht indem man die zufällig gezogenen Spielsteine so anordnet, dass sich abgeschlossene *Räume* bilden. Jene/r SpielerIn, der/die den letzten zur Vervollständigung eines *Raumes* notwendigen Spielstein angelegt hat, erhält sämtliche Punkte für den entsprechenden *Raum* gutgeschrieben. Die Anzahl der erhaltenen Punkte berechnet sich dabei aus der Anzahl der zum *Raum* gehörenden Spielsteine sowie deren Typ. Dabei haben offene Steine den höchsten Punktwert, Steine mit *Tür* den nächst niedrigeren und Steine mit durchgängiger *Wand* den geringsten. Die Punktwerte der Steine werden addiert und schlussendlich wird geprüft, ob die Mitte des Spielfelds von dem *Raum* aus erreichbar ist. Ist dies der Fall, so werden die Punkte des Raumes verdoppelt. Wer nun am Ende des Spiels die meisten Punkte erreicht hat ist der/die SiegerIn.

### 3.3.2 Möglichkeiten Spielsteine anzulegen

Beim Platzieren der Spielsteine gelten folgende Einschränkungen:

1. Ein Spielstein muss immer an einen anderen Spielstein angrenzend abgelegt werden.
2. Grenzt ein Spielstein mit einer *Wand*-Seite an einen anderen Spielstein, so darf diese nicht direkt an die *Wand*-Seite des zweiten Steines stoßen.
3. Grenzt ein Spielstein mit einer offenen Seite an eine offene Seite eines anderen Spielsteins, so müssen beide Steine die gleiche Farbe haben.

# Design und Entwicklung

Dieses Kapitel vermittelt einen detaillierten Überblick über die einzelnen Planungs- und Entwicklungsschritte, die zur Implementierung des Prototypen notwendig waren. Dabei wird sowohl auf die verwendete Software, als auch auf die zu Grunde liegende Hardware-Basis eingegangen.

## 4.1 Hardwarebasis

Ursprünglich bestand die Absicht, die entwickelte Software auf einem bereits existenten Prototypen eines Touch-Tisches, der im Zuge eines Diplomprojekts – des MusicTheory-Table (vgl. [Kah11, 64ff]) – gebaut worden war, zum Einsatz zu bringen. Das System nutzte zwei Sony PlayStation Eye Kameras im Werkszustand für das Tracken von Touch-Eingaben und Fiducial Markern, während zwei Pico-Beamer zur Projektion des darzustellenden Inhalts auf die Tischplatte dienten. Bei dieser handelte es sich um eine 745x485mm große, satinierte Acrylglasplatte mit einer Dicke von 3 mm. Die gesamte Konstruktion basierte auf einem Rahmengerüst aus Aluminiumprofilen.

Bei der ersten praktischen Evaluierung der Machbarkeit, auf Basis der genannten bestehenden Hardware, zeigten sich mehrere Probleme mit dem Aufbau. Einerseits stellte sich heraus, dass die Fiducial Marker durch die satinierte Plexiglasplatte zu stark verschwammen und somit von der Tracking-Software nicht mehr erkannt werden konnten. Gleichzeitig ergab sich aus der Verwendung von unmodifizierten Kameras und der sich daraus ergebenden Notwendigkeit die Tischplatte von unten mit Licht aus dem sichtbaren Spektrum zu beleuchten, das Problem, dass die Beleuchtung das Bild des Beamers überstrahlte.

Aufgrund genannter Probleme und Einschränkungen wurde der Bau einer eigenen Hardware-Basis entschieden, die folgende Anforderung erfüllen sollte:

- Tracking von Fiducial Markern mit einer Größe von ca. 4-6 cm Kantenlänge

- Eine Rechteckige Tischplatte, die groß genug ist um gleichzeitig mindestens 100 Spielsteine darzustellen
- Ausreichend detailreiche grafische Darstellung der Benutzeroberfläche
- Gute Erkennbarkeit, der dargestellten grafischen Inhalte

Um die Wiederverwendbarkeit und den Nutzwert insgesamt zu heben, wurden noch zusätzliche Anforderungen berücksichtigt:

- Die Möglichkeit Touch-Eingaben zu erfassen (Multitouch)
- Stabilität der Konstruktion
- Wartbarkeit/Wiederverwendbarkeit

Anhand der erwähnten Anforderungen wurde ein Konzept für einen Touch-Tisch mit einer nutzbaren Oberfläche von mindestens 900 x 600 mm entwickelt, der im Stehen zu verwenden ist. Die Erfassung der Touch-Eingaben sowie das Fiducial Tracking findet dabei im Infrarot-Spektrum statt, während ein lichtstarker Kurzstanz-Projektor unterhalb der Tischplatte für die Darstellung der grafischen Ausgaben verantwortlich ist. Bei der Recherche bestehender Hardwareprojekte mit vergleichbaren Anforderungen stellte sich die Verwendung von modifizierten PlayStation Eye Kameras als eine gängige und erprobte Methode heraus das optische Tracking zu realisieren. Als Projektor kommt der BenQ MW870UST zur Verwendung, der, durch seine extrem kurze Wurfdistanz, ohne die Notwendigkeit eines Umlenkspiegels verbaut werden kann und durch die maximale Helligkeit von 3000 Lumen auch bei Tageslicht ein erkennbares Bild produziert. Im Folgenden sind die technischen Spezifikationen der verbauten Hardware aufgelistet.

##### 4.1.1 Sony Playstation Eye

**Optik (werksseitig):** Fixfokus-Optik mit zwei Einstellungen, die ein Umschalten zwischen 56° und 75° Blickfeld erlaubt, Infrarot-Block-Filter

**Optik (modifiziert):** Fixfokus-Optik mit Brennweite 3,6mm, Infrarot-Pass-Filter (Wellenlänge 850 nm)

**Auflösung:** 640 x 480 bei 60 Bildern/Sekunde, 320 x 240 bei 120 Bildern/Sekunde

**Audio:** Anordnung von 4 Mikrofonkapseln mit einer Sampling-Rate von 16-bit bei 48 kHz, SNR 90 dB

**Schnittstellen:** USB 2.0

**Abmessungen (HxBxT):** 55 x 80 x 65 mm

### 4.1.2 BenQ MW870UST

**Projektionstechnologie:** DLP

**Kontrastverhältnis:**

**Helligkeit:** 3000 Lumen

**Auflösung:** 1280x800 (nativ)

**Video-Modi:** 720p, 1080i, 1080p/60, 576i, 576p, 480p, 480i

**Lautsprecher:** 10.0 W x 2

**Video-Eingänge:** VGA, S-VIDEO, HDMI 1.3

**Abmessungen (HxBxT):** 15 x 33 x 28 cm

**Wurfdistanz:** 0.3–2.4 m

**Bildgröße:** 107–758 cm

**Projektionsverhältnis:** 0.37:1

### 4.1.3 LED-Module

**Strahlleistung:** 1051 mW

**Lichtfarbe:** Infrarot 850 nm

**Anzahl der LEDs:** 3 je Modul, 16 mm Abstand

**Abstrahlwinkel:** 120°

**Abmessungen:** 44 x 18 x 9 mm

**Betriebsspannung:** 12V DC

## 4.2 Konstruktion des Tisches

Durch die Rechteckige Form, der angepeilten nutzbaren Tischoberfläche und der Projektionscharakteristik des Verwendeten Beamers, ergab sich die Notwendigkeit eines Rahmens an einer der Längsseiten der Projektionsfläche. Der Projektor musste versetzt außerhalb der Projektionsoberfläche befestigt werden. Es fiel die Entscheidung das Design entsprechend symmetrisch anzupassen und den Rahmen auf beiden Längsseiten einzuplanen, da er beim Benutzen des Tisches einerseits zum Ablegen der Hände bzw. Unterarme dienen kann, andererseits auch verwendet werden kann um Fiducials dort abzulegen, wenn diese gerade nicht in Gebrauch sind. So ergibt sich eine annähernd quadratische Grundfläche

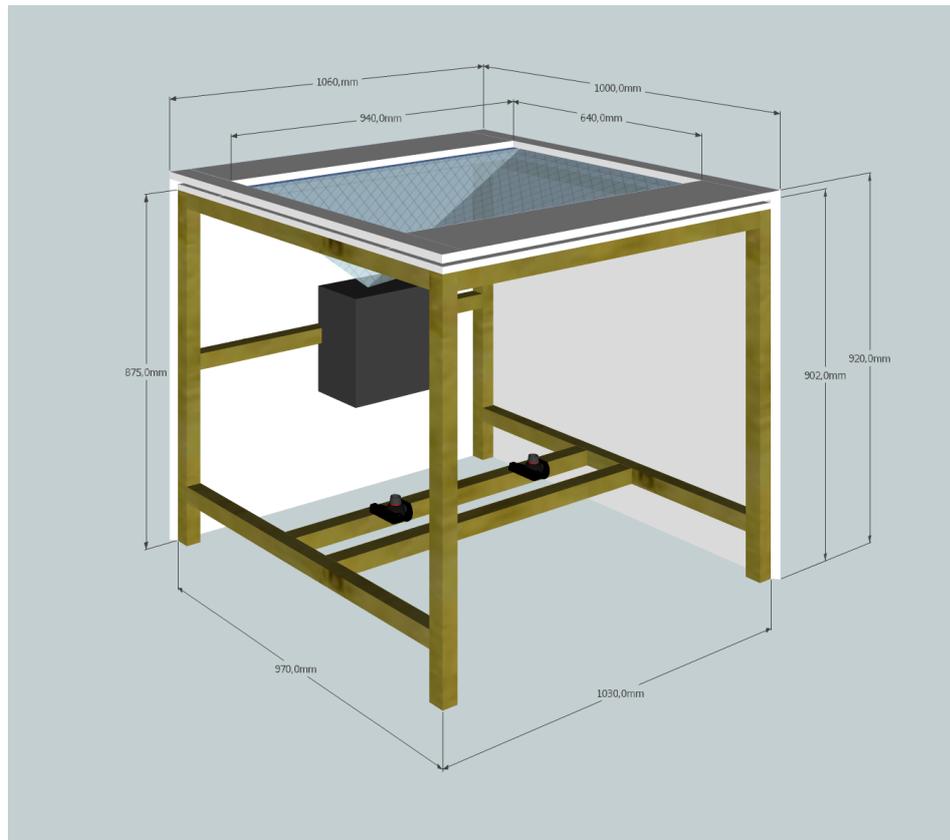


Abbildung 4.1: 3D-Modell der Rahmenkonstruktion



Abbildung 4.2: Rahmenkonstruktion vor der Installation der Elektronik

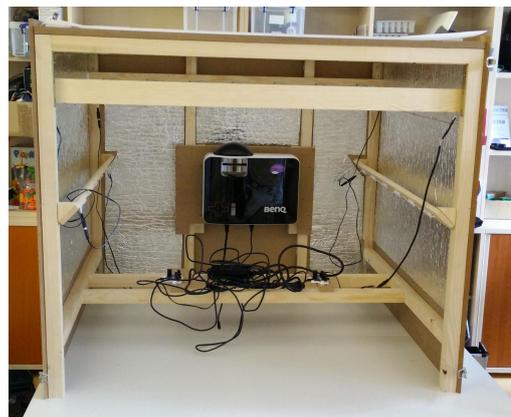


Abbildung 4.3: Tisch mit installierter Hardware

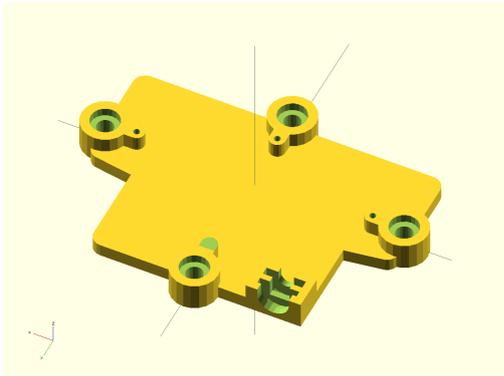


Abbildung 4.4: CAD-Modell der Halterung für die Kameraplatine

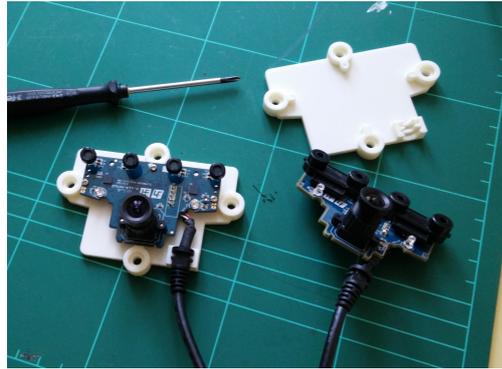


Abbildung 4.5: 3D gedruckte Halterung mit installierter Platine

der Konstruktion.

Aufgrund vorhergehender Erfahrungen und der relativ einfachen Verarbeitung sowie der vergleichsweise günstigen Kosten, wurden 35 x 35 mm Fichtenleisten für die Rahmenkonstruktion gewählt, die mit Stuhlwinkeln verbunden wurden. Die notwendige Verwindungssteifigkeit erhält die Konstruktion durch die Verkleidung des Rahmens mit 10 mm MDF-Platten. Diese sind an drei Seiten des Rahmens jeweils mit Schrauben befestigt. Um eine möglichst einfache Wartbarkeit zu ermöglichen, wurden Gewindemuffen im Rahmen montiert. Diese ermöglichen ein Lösen und der Schraubverbindungen ohne das Material zu verschleifen. An der vierten Seite ist die Verkleidung mit Truhenverschlüssen angebracht, was einen schnellen und einfachen Zugang zum Inneren der Konstruktion ermöglicht.

Als Projektions- und Eingabefläche dient eine 12 mm starke klare Acrylglasplatte die auf den Rahmen aufgelegt ist. Die Materialstärke gewährleistet die nötige Steifigkeit um zu verhindern, dass die Tischplatte nachgibt, wenn Eingaben getätigt werden. Auf der Acrylglasplatte ist eine Projektionsschicht angebracht. Dabei handelt es sich um eine transluzente reflexionsfreie Polyesterfolie, die in der Regel von Architekten zum Zeichnen von Plänen eingesetzt wird. Sie bieten einen guten Kompromiss zwischen der Fähigkeit zur Projektion und der klaren Darstellung der Fiducials und ist dabei sehr preisgünstig. Es wäre auch die Verwendung einer Folie, die speziell für die Projektion ausgelegt ist möglich. Durch diese wird die Darstellung der Fiducials allerdings in der Regel zu unscharf, um eine entsprechende Erkennung zu gewährleisten.

#### 4.2.1 Beleuchtung

Die LED-Module wurden in gleichmäßigen Abständen an allen vier Seiten der Konstruktion am Rahmen angebracht. Dabei wurden die Module im Winkel von  $90^\circ$  zur Plexiglasplatte an der Längsseite im Abstand von ca. 10 cm von der Oberseite der Rahmenkonstruktion befestigt. Durch den Rahmen an der Längsseite konnten direkte Reflexionen der LED-Module an der Längsseite, die auf dem von dem Webcams

aufgenommenen Bildern als Hotspots sichtbar wären und so die Bilderkennung stören würden, vermieden werden. An der Breitseite wurden die Module zur Vermeidung dieses Störfaktors im Winkel von  $135^\circ$  zur Oberfläche im vertikalen Abstand von ca. 40 cm angebracht, sodass das abgestrahlte Licht möglichst nicht direkt an der Oberfläche reflektiert wird. Um die Helligkeit an der Oberfläche zu optimieren und eine möglichst gleichmäßige Ausleuchtung zu gewährleisten, wurde die Innenseite der Tischkonstruktion mit infrarot-reflektierender Folie ausgekleidet.

### 4.2.2 Kameras und Projektor

Die Linsen, mit der die Sony Playstation Eye-Kameras werksseitig ausgestattet waren, wurden inklusive Infrarotfilter entfernt und es wurde eine Gewindehalterung mit M12-Gewinde installiert, in die das 3,6 mm-Fixfokus-Objektiv mit dem Infrarot-Pass-Filter eingesetzt wurden. Nach der experimentellen Ermittlung der idealen Positionen der Kameras zueinander sowie der optimalen Entfernung zur Oberfläche, wurden diese fix installiert (vgl. Abbildungen 4.4 und 4.5). Um ein Verrutschen der Kameras zu verhindern, wurden die Kameragehäuse entfernt und die Platinen auf eigens konstruierte, im 3D-Druckverfahren hergestellte Halterungen installiert, die an der Holzrahmenkonstruktion festgeschraubt wurden.

Der Projektor wurde ebenfalls mit der Rahmenkonstruktion fest verschraubt.

## 4.3 Software

### 4.3.1 reacTIVision und TUIO

Bei reacTIVision handelt es sich um ein Toolkit für maschinelles Sehen, das speziell für den Einsatz in Systemen mit Multitouch-Oberflächen und die Erkennung von Fiducial Markern entwickelt wurde. Ursprünglich war der hauptsächliche Anwendungsbereich eine interaktive musikalische Installation namens reacTable (vgl. [JGAK07]), ein modularer Synthesizer mit einem TUI auf Basis einer interaktiven Tischoberfläche. Das zugrundeliegende Protokoll TUIO wurde jedoch bald von verschiedenen Open Source Projekten zur Implementierung großer Multitouch-Oberflächen übernommen (vgl. [Kal09]) und hat sich so zu einem Quasi-Standard entwickelt.

#### Das TUIO-Protokoll

Der unmittelbare Zweck des TUIO-Protokolls war eine Abbildung der Lage von Pointern (Fingern) und Tokens (sonstige physischen Objekte) auf einer zweidimensionalen Tischoberfläche. Während für die Pointer nur die normalisierten kartesischen Koordinaten erfasst wurden, war es für Tokens zusätzlich möglich eine Kennung und Information zur ihrer Rotation zu übermitteln. Die Implementierung des TUIO Protokolls stützt sich auf Open Sound Control, ein Nachrichten-basiertes Netzwerkprotokoll, dessen Haupteinsatzgebiet die Echtzeitverarbeitung von Sound in Netzwerken ist.

Die Nachrichten, die in OSC verwendet werden, bestehen dabei lediglich aus einer null-terminierten Zeichenkette, die die Adresse des zu ändernden Parameters spezifiziert und, darauf folgend, den einzelnen Werten, die der jeweilige Parameter annehmen soll. Die einzelnen Nachrichten werden wiederum zu sogenannten Bündeln zusammengepackt und können dann versendet werden. Obwohl OSC unabhängig vom Transportprotokoll ist, wird in der Regel UDP zur Übertragung der OSC-Bündel verwendet. Auch TUIO setzt auf diese Art der Übermittlung der Bündel, was die nötige latenzarme Kommunikation ermöglichte, die für das erste Anwendungsszenario, bei dem mehrere `reactTables` in verschiedenen Städten über eine Internetleitung miteinander verbunden waren, Voraussetzung war. Um die Auswirkungen möglicher Übertragungsfehler, die durch die Verwendung von UDP entstehen können, zu minimieren, wird ein Zustandsmodell anstelle eines Event-basierten Ansatzes verwendet. Dabei werden in jedem Nachrichtenbündel alle derzeit aktiven Pointer und Tokens übermittelt, sodass der Verlust von Datenpaketen kompensiert werden kann.

Durch die weite Verbreitung des TUIO-Protokolls, existieren mittlerweile zahlreiche Implementierungen von Clients in diversen Programmiersprachen. Das Spektrum reicht dabei von C++ und Java bis Flash und Processing.

### **reactTIVision**

ReactTIVision dient als Engine und übernimmt sämtliche Bildverarbeitungsschritte, die nötig sind um auf einem Kamerabild die Fiducials zu erkennen sowie Finger-Tracking zu ermöglichen. Die Kamerakoordinaten werden dann in Applikationskoordinaten transformiert und die erkannten Pointer und Tokens werden in Form von TUIO-Nachrichten an die Applikation weitergeleitet. Dabei ist eine Kalibrierung mithilfe eines Koordinatennetzes möglich, um eine eventuelle Verzerrung durch die Kameralinse auszugleichen. Dabei wird ein Ausgedrucktes Gitternetz auf die zu erfassende Oberfläche aufgelegt und ein entsprechend großes Netz wird als Overlay eingeblendet. Die Einzelnen Schnittpunkte des eingeblendeten Netzes werden nun auf ihre jeweilige Entsprechung auf dem aufgedruckten Netz geschoben.

Die Tracking-Software ist in C++ implementiert und steht für Linux, Windows und MacOS zur Verfügung. Sie unterstützt im unmodifizierten Zustand jedoch nur die Erfassung von nicht mehr als einer Kamera.

## **4.3.2 Auswahl der Tracking-Software**

### **reactTIVision als erster Ansatz**

In der Anfangsphase des Projekts war noch nicht klar, dass die Umsetzung nach einer Lösung mit mehr als einer Kamera verlangte. Die ersten Versuche erfolgten deshalb mit `reactTIVision` in der Version 1.4 auf einem Linux System. Die Software lieferte dabei sehr gute Ergebnisse bei der Erkennung von Fiducials ab einer gewissen Größe bei gleichzeitig guter Performance. War die Kamera allerdings soweit von der Tischfläche entfernt, wie zur Erfassung der gesamten Oberfläche nötig war, reichte die Auflösung für

eine verlässliche Erkennung der Fiducials nicht mehr aus.

Es wäre in diesem Fall nötig gewesen die Symbole auf ein Maß zu vergrößern, die eine sinnvolle Umsetzung des Spielkonzepts nicht mehr möglich gemacht hätte. Aus diesem Grund wurde das System um eine zweite Kamera erweitert, was auch bei der Software eine Adaptierung bedingte. Nach entsprechender Recherche fiel die Entscheidung vorerst darauf, die ReactIVision-Software selbst um die Möglichkeit eine zweite Kamera einzusetzen, zu erweitern. Dieses Vorhaben war bereits für MacOS in Form des ReactIVision-Forks StitchRV (vgl. [WBA10]), dessen Sourcecode zur Verfügung stand, umgesetzt worden. Anhand des dort verfolgten Ansatzes wurde eine entsprechende Anpassung für den Linux-Zweig von ReactIVision durchgeführt. Die Adaptionen beschränkten sich dabei auf die Implementierung einer eigenen Klasse, die ein Interface implementiert, über welches jede Kameraquelle in ReactIVision angesteuert wird, und intern das Bild von zwei unterschiedlichen Playstation-Kameras jeweils um 90° dreht und zu einem einzigen Bild zusammenfügt, bevor es an die Bilderkennungsrountinen weitergereicht wird. Die Evaluierung dieser Lösung ergab eine gute Erkennung von Fiducials, wobei es allerdings zu Problemen im Bereich der Stoßkante der zwei Bilder kam, da die von reactIVision mitgebrachte Methode der Kalibrierung für die Verwendung von mehr als einer Kamera nicht geeignet war.

Da im Laufe der Umsetzung des Projekts die Version 1.5 der reactIVision-Software veröffentlicht worden war, wurde auch für die neue Version der Software die entsprechende Anpassung umgesetzt, die allerdings ebenfalls keine zufriedenstellenden Ergebnisse im Bereich der Stoßkanten lieferte. Eine Anpassung an der Kalibrierungsroutine gestaltete sich als zu aufwendig, sodass die Entscheidung fiel die Tracking-Software auszutauschen.

### **Wechsel zu CCV**

Die freie Tracking-Lösung Community Core Vision stellte sich dabei als beste Alternative heraus, da sie frei verfügbar war, performant arbeitete und seit der Version 1.5 das Tracking für Systeme mit mehreren Kameras unterstützte. Sie baut dabei auf gängige Frameworks für Bildverarbeitung wie OpenCV und openFrameworks auf. Gleichzeitig ist sie plattformübergreifend lauffähig, wobei allerdings zum aktuellen Zeitpunkt die Multi-Kamera-Unterstützung nur für Windows-Systeme verfügbar ist. Diese machte mit dem Wechsel der Tracking-Lösung auch einen Plattformwechsel zu Windows 7 notwendig. Im Gegensatz zu Linux ist die Hardwareunterstützung für die verwendeten Kameras unter Windows nur kommerziell in Form des CL-Eye-Treibers [CL16] verfügbar. Die Verwendung von mehr als einer Kamera auf dem selben System, erfordert dabei zusätzlich den Einsatz der dazugehörigen Management-Software, da die Identifikation der beiden Kameras anhand ihrer Hardware-ID durch CCV ansonsten nicht gewährleistet ist.

Waren die oben genannten Voraussetzungen erfüllt, stellte sich CCV als adäquate Lösung heraus, mit welcher eine verlässliche Erkennung von Fiducials auf der gesamten Oberfläche möglich war.

### 4.3.3 Entwicklung des Spiels

Wie bereits erwähnt existiert eine große Auswahl an Implementierungen von Clients für das TUIO-Protokoll in den verschiedensten Programmiersprachen. Nach einer ersten Recherche der verfügbaren Bibliotheken wurde mit einer ersten Implementierung in Processing unter Zuhilfenahme der Processing TUIO Client API begonnen. Nach kurzer Zeit stellte sich allerdings die mangelhafte IDE-Unterstützung als problematisch dar und auch die gewünschte Modularisierung der Anwendung konnte mit Processing nicht zufriedenstellend erreicht werden, weswegen eine Portierung nötig wurde. Aufgrund entsprechender Vorerfahrung und der Plattformunabhängigkeit, wurde ein Wechsel zu Java beschlossen.

Für Java steht mit MT4j eine Multitouch-Entwicklungsplattform zur Verfügung, die auf TUIO-Eingaben aufsetzt und gleichzeitig den Java-Unterbau von Processing nutzt. Es ermöglicht eine hardwarebeschleunigte Bildschirmausgabe und transformiert das Statusbasierte Prinzip des TUIO-Protokolls in ein im Java-Kontext geläufiges Event-System.

#### MT4j

MT4j (Multi-touch for Java) [LRZ10] ist ein plattformunabhängiges Framework, das viele der Aufgaben übernimmt, die für die Entwicklung von Anwendungen für Multitouch-Systeme häufig benötigt werden. Dazu zählen unter Anderem die Verarbeitung von Touch-Gesten und das Zurverfügungstellen von Widgets, speziell für die Nutzung mit Berührungseingabe. Die Basis für die Erstellung einer Anwendung mit MT4j bildet die abstrakte Klasse `AbstractMTApplication`. Sie stellt Funktionalität für die visuelle Ausgabe sowie der Eingabebehandlung zur Verfügung. Dazu dient die Methode `processInputEvent`, die aufgerufen wird, sobald ein TUIO-Bündel von der Tracking-Software empfangen wurde und die für die Behandlung der Reaktionen auf Benutzereingaben zuständig ist. Dies erfolgt über ein Event-System, wobei zwischen Ereignissen für Fingereingaben und Fiducials unterschieden wird.

Für Fiducials dient die Ereignisklasse `MTFiducialInputEvt`, die sich, zur Kodierung der Eingaben, in einem der drei folgenden Status befinden kann:

**INPUT\_STARTED** bedeutet, dass ein Token gerade auf die Oberfläche gelegt wurde.

Die Besonderheit hierbei ist, dass CCV im ersten Bündel nicht die tatsächlichen Koordinaten der Position des Objektes sendet, sondern einen Standardwert außerhalb der Oberfläche.

**INPUT\_UPDATED** wird genutzt wann immer ein Token seine Position oder Rotation ändert.

**INPUT\_ENDED** bedeutet, dass ein zuvor erfasster Token mit einer bestimmten ID nicht mehr detektiert wurde und somit als nicht mehr auf der Oberfläche befindlich gilt, er also entfernt wurde.

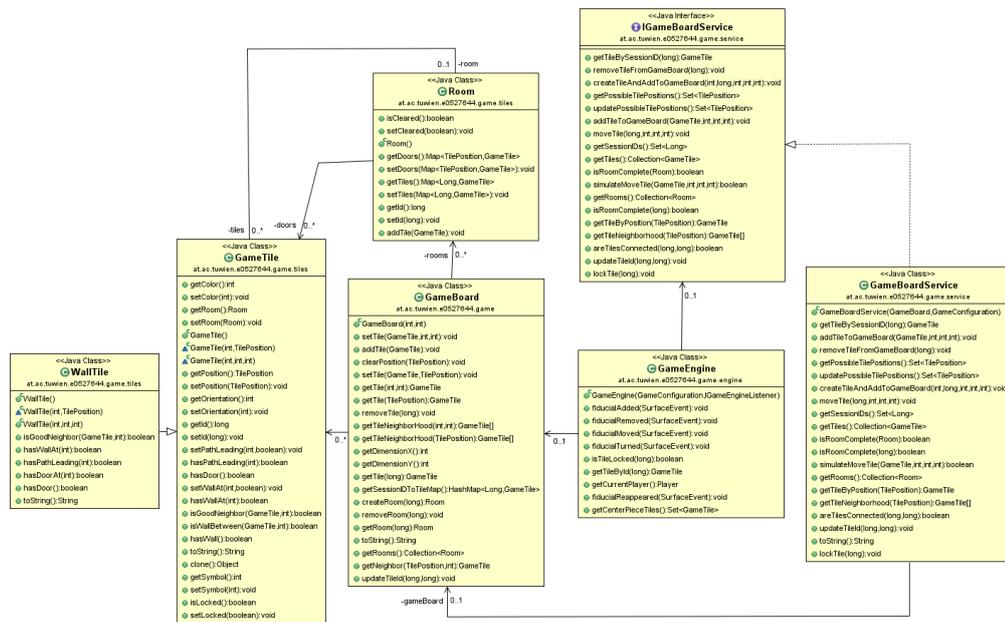


Abbildung 4.6: Klassendiagramm der Kernkomponenten

Für die Bildschirmausgabe stehen diverse Widgets und Grafikprimitive zur Verfügung, die die direkte Manipulation mittels Berührungseingaben unterstützen, wobei diese Funktionen im Rahmen des in dieser Arbeit behandelten Projekts nicht zum Einsatz gebracht wurden.

### 4.3.4 Implementierung

Die Software ist nach dem MVP-Prinzip strukturiert. Die *Ansicht* stellt die Klasse *GameScene* dar, die für die Behandlung der Benutzereingaben und der Realisierung der graphischen und auditiven Ausgaben zuständig ist. Sie ist von der bereits genannten *AbstractMTApplication* aus der *MT4j*-Plattform abgeleitet. Die *GameEngine* übernimmt die Aufgaben des *Repräsentators*, also in diesem Fall das Verwalten des Spielm odells und die Abbildung der Spielregeln. Die Kommunikation der beiden Komponenten basiert auf einem Eventsystem. Dabei stehen verschiedene Ereignisse zur Verfügung um Information von der Benutzerschnittstelle an die *Engine* weiter zu reichen. Die in Reaktion auf die entsprechenden Events aufgerufenen Methoden werden im Folgenden näher beschrieben:

**fiducialAdded** Diese Methode wird aufgerufen, wenn ein neuer Spielstein auf der Oberfläche detektiert wurde.

**fiducialRemoved** Ist die Methode, die aufgerufen wird, wenn ein Spielsteine von der Spielfläche entfernt wurde.

**fiducialMoved** Die Methode wird genutzt um einen Positionswechsel eines Spielsteins an die Engine zu melden.

**fiducialTurned** Ändert sich die Ausrichtung eines Spielsteins, wird diese Methode genutzt.

**fiducialReappeared** Ist ein Spielstein, der kurzzeitig als entfernt gegolten hat – etwa weil der Fiducial-Marker nicht erkannt wurde – wieder detektiert worden, so wird dies mithilfe dieser Methode an die Engine weitergegeben.

Informationen wie die Koordinaten, der Rotationswinkel, die eindeutige Identifikationsnummer und die Kennung des Fiducial-Markers werden so an die Engine weitergegeben, wo die Koordinaten auf eine Position im Datenmodell abgebildet und dann unter Berücksichtigung der Ausrichtung und Farbe des entsprechenden Spielsteins auf Validität im Kontext der Spielregeln geprüft werden. Abhängig vom Ergebnis dieser Prüfung wird dann wiederum über ein eigenes Set von Events an die Benutzeroberfläche zurückgemeldet, ob ein Spielzug gültig war oder einen regelwidrigen Zustand erzeugt hat. Folgende Methoden stehen dafür zur Verfügung:

**adjacentWallErrorOccured** Ein Spielstein wurde so positioniert, dass seine *Wandseite* an die *Wandseite* eines anderen Spielsteins stößt. Dies widerspricht den Spielregeln.

**wrongColorErrorOccured** Ein Spielstein wurde mit einer offenen Seite an eine jeweils offene Seite eines Spielsteins anderer Farbe gelegt. Dabei handelt es sich um eine Regelwidrigkeit.

**noNeighborsErrorOccured** Ein Spielstein wurde abseits eines markierten Felds abgelegt, wobei es sich um einen regelwidrigen Zug handelt.

**updatedPossiblePositions** Ein Spielstein wurde an eine valide Position gelegt, daraus ergeben sich neue Positionierungsmöglichkeiten für den nächsten Spielzug.

**updatedRooms** Ein Spielstein wurde an eine valide Position gelegt, die im Datenmodell hinterlegten *Räume* haben sich dadurch geändert.

**updatedLockedTiles** Ein Spielstein wurde an eine valide Position gelegt, was als erlaubter Spielzug registriert wurde. Dieser Spielstein kann daraufhin nicht weiter bewegt werden.

**roomCompleted** Mit dem letzten Spielzug wurde ein *Raum* abgeschlossen.

Die genannten Events lösen in der Benutzerschnittstelle jeweils entsprechende Auditive und grafische Effekte, wie das Einblenden von Fehlersymbolen oder das Aktualisieren der Hinweistexte und der Punktestände der SpielerInnen aus.

Die existenz einer Fehlersituation wird geprüft, bevor ein Spielzug als vollendet gilt.

Solange also ein/e SpielerIn seinen/ihren Spielstein nicht gültig positioniert hat, wird nach jedem Positions- oder Orientierungswechsel des Spielsteins erneut geprüft ob die aktuelle Position gültig ist. Dies basiert auf einem Modell, wonach ein jeder Spielstein, neben anderen, über folgende Eigenschaften verfügt:

**Position** Die Position auf dem Spielfeld, das in ein Raster aus Spalten und Zeilen unterteilt ist. Die Position des Spielstein wird ermittelt indem geprüft wird in welchem Kästchen des Rasters der Mittelpunkt des Spielsteins sich befindet.

**Orientierung** Ein Spielstein kann nur eine von vier Orientierungen aufweisen, wobei diese von 0 beginnend im Uhrzeigersinn nummeriert sind und 0 für Norden, 1 für Osten usw. steht. Der Rotationswinkel des Spielsteins wird folgender Formel entsprechend auf eine der vier Orientierungen abgebildet:  $\lfloor \frac{\alpha}{90} \rfloor \bmod 4$

**Wände** Jeder Spielstein verfügt über ein Array mit der Länge 4, das die binäre Information darüber enthält, an welcher Seite des Spielsteins sich eine *Wand* befindet. Dabei wird wie bei der Orientierung vorgegangen und von 0 für Norden, im Uhrzeigersinn gezählt.

**Pfade** Wie die Information über die *Wände* eines Spielsteins, wird auch die Information über die Pfade in einem gleichsam aufgebauten Array gespeichert. Ein Pfad existiert dann, wenn es entweder keine *Wand* an der entsprechenden Seite des Spielsteins gibt, oder aber sich an der *Wandseite* auch eine *Tür* befindet.

**Farbe** Die Farbe wird durch einen Integerwert repräsentiert und dient dazu festzustellen ob zwei Spielsteine die gleiche Farbe haben oder nicht.

Ob zwei Spielsteine regelkonform nebeneinander positioniert wurden wird anhand des Algorithmus in Auflistung 4.1 festgestellt. Dies passiert für jeden Stein, der an den aktuell ausgespielten Spielstein angrenzt.

Ebenso wird ermittelt ob der aktuelle Spielzug einen *Raum* vervollständigen würde. Dafür wird zuerst mithilfe des Algorithmus in Auflistung 4.2 festgestellt ob jeder zum *Raum* gehörige Spielstein über eine *Wand* an der Außenseite verfügt oder aber, wenn nicht, mit dieser Seite an die *Wand* eines anderen Steins angrenzt. Dazu werden die Steine eines *Raumes* jeweils nach Spalte gruppiert und dann nach Zeile geordnet, sodass geprüft werden kann, dass der Stein mit der niedrigsten und derjenige mit der Höchsten Zeilennummer pro Spalte an der Außenseite jeweils eine *Wand* hat. Gleiches gilt für die einzelnen Zeilen.

Listing 4.1: Methode um festzustellen ob zwei Spielsteine regelkonform zueinander platziert wurden

```

private boolean areGoodNeighbors(GameTile a, GameTile b, int direction)
throws InvalidTilePositionException {
    if (a == null || b == null) {
        return true;
    }
    if (a.hasWallAt(direction)) {
        // A has a wall facing B
        if (b.hasWallAt((direction + 2) % 4)) {
            // B has a wall facing A in the opposite direction
            // which leads to a double wall
            throw new AdjacentWallsException(a, b);
        } else {
            // B has no wall facing A in the opposite direction
            // There is a single wall between the two tiles
            return true;
        }
    } else {
        // In case there is no wall on this tile in the specified direction
        if (b.hasWallAt((direction + 2) % 4)) {
            // B has a wall facing A in the opposite direction
            // There is a single wall between the two tiles
            return true;
        } else {
            // B has no wall facing A in the opposite direction
            if (b.getColor() == a.getColor()) {
                // The tiles are the same color
                return true;
            } else {
                // The tiles are of different color
                throw new WrongColorException(a, b);
            }
        }
    }
}
return true;
}

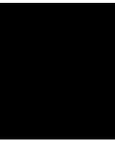
```

Listing 4.2: Methode um zu ermitteln, ob ein *Raum* an allen Seiten durchgehend von einer *Wand* umgeben ist

```
public boolean isRoomComplete(long roomID) {
    Room room = gameBoard.getRoom(roomID);
    Collection<GameTile> tiles = room.getTiles().values();
    Map<Integer, Vector<GameTile>> groupedByCol = GameTileHelper.groupTilesByCol(tiles);
    groupedByCol = GameTileHelper.sortByRow(groupedByCol);
    Map<Integer, Vector<GameTile>> groupedByRow = GameTileHelper.groupTilesByRow(tiles);
    groupedByRow = GameTileHelper.sortByCol(groupedByRow);
    for (Vector<GameTile> gameTiles : groupedByCol.values()) {
        if (!tileHasWallAtDirection(gameTiles.firstElement(), GameTile.NORTH))
            return false;
        if (!tileHasWallAtDirection(gameTiles.lastElement(), GameTile.SOUTH))
            return false;
    }
    for (Vector<GameTile> gameTiles : groupedByRow.values()) {
        if (!tileHasWallAtDirection(gameTiles.firstElement(), GameTile.WEST))
            return false;
        if (!tileHasWallAtDirection(gameTiles.lastElement(), GameTile.EAST))
            return false;
    }
    return true;
}

private boolean tileHasWallAtDirection(GameTile tile, int direction) {
    GameTile neighbor = gameBoard
        .getNeighbor(tile.getPosition(), direction);
    boolean neighborHasWall = false;
    if (neighbor != null) {
        neighborHasWall = neighbor.hasWallAt((direction + 2) % 4);
    } else if (isTileNeighborBoardEdge(tile.getPosition(), direction)) {
        neighborHasWall = true;
    }
    return (tile.hasWallAt(direction) || neighborHasWall);
}

private boolean isTileNeighborBoardEdge(TilePosition position, int direction) {
    if (position.getCol() <= 0 && direction == GameTile.WEST)
        return true;
    if (position.getRow() <= 0 && direction == GameTile.NORTH)
        return true;
    if (position.getCol() >= gameBoard.getDimensionX() - 1
        && direction == GameTile.EAST)
        return true;
    if (position.getRow() >= gameBoard.getDimensionY() - 1
        && direction == GameTile.SOUTH)
        return true;
    return false;
}
```



# User Interface

Im Folgenden wird die Konzeption und Entwicklung der Benutzeroberfläche beschrieben, die sich von ersten Designskizzen bis zu einem funktionstüchtigen Prototypen erstreckt. Dabei gliedert sich die Gestaltung der Benutzeroberfläche in zwei Komponenten, die ineinander greifen. Einerseits wurden anhand der in Kapitel 3 beschriebenen Regeln Spielsteine konzipiert, die in physischer Form vorliegen, andererseits wurde mit der graphischen Oberfläche, die auf den Tisch projiziert wird, auch die virtuelle Komponente der Benutzerschnittstelle entwickelt.

## 5.1 Analyse

Die Anforderung an das System, den BenutzerInnen das Spiel zu vermitteln, ohne dass diese Vorkenntnisse über dessen Regeln haben, setzt voraus, dass die Benutzerschnittstelle intuitiv erfassbar sein muss. Gleichzeitig sollte sie gegenüber Fehleingaben von BenutzerInnen tolerant sein und diese mit dem nötigen Feedback darauf hinweisen, wenn sie regelwidrig spielen. Im Gegensatz zu Software, die für einen produktiven Einsatz konzipiert ist, ist es in diesem Kontext nicht nötig, dass Benutzereingaben rückgängig gemacht werden können, da kein reeller Schaden entsteht, wenn zum Beispiel ein Spielstein nicht optimal platziert wurde. Das Konzept des Spiels wird dabei – abgesehen vom digitalen Feedback – zu einem guten Teil durch die Gestaltung der Spielsteine vermittelt.

## 5.2 Design der physischen Komponenten

Sobald ein grobes Spielkonzept feststand, wurden erste Entwürfe für das Aussehen der der Spielsteine erstellt. Bei den ersten Versionen ist nur die abstrakte Form der Steine gegeben. Es handelte sich um Kärtchen mit quadratischer Grundfläche, die vollfarbig ausgemalt waren. Laut den Erkenntnissen der Gestaltpsychologie werden Objekte gleicher Farbe als zusammengehörig wahrgenommen (vgl. [RP90]). Dieser Umstand wird bei der farblichen

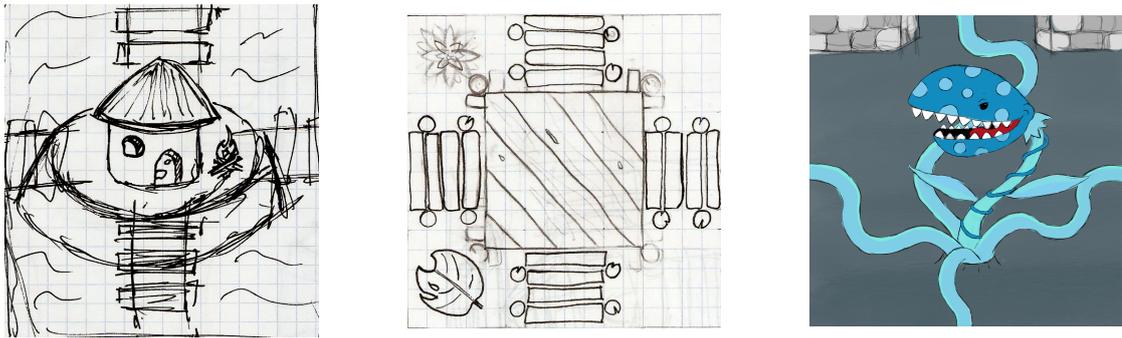


Abbildung 5.1: Alternative Gestaltungsentwürfe der Spielsteine

Gestaltung der Spielsteine genutzt, da dadurch intuitiv erkennbar wird, dass Spielsteine mit gleicher Farbe aneinander gereiht werden sollen. Die konzeptuelle *Wand* wurde durch einen einfachen schwarzen Balken repräsentiert. In dieser Entwicklungsphase gab es nur einen Typ Spielsteine, nämlich solche mit einer durchgehenden *Wand*, diese allerdings in sechs verschiedenen Farben. Es stellte sich bald heraus, dass durch die Beschränkung auf einen Typ Spielsteine, *Räume* nur eine längliche Form mit Maximalausmaßen von zwei Steinen an der jeweils kürzeren Seite erreichen konnten, was eine starke Einschränkung darstellt und den Spielverlauf zu wenig abwechslungsreich darstellte. Aus diesem Grund wurden die sogenannten offenen Spielsteine eingeführt, mit deren Hilfe es möglich wird einen *Raum* in beide Dimensionen beliebig auszudehnen. Sie erlauben auch *Räume* in unterschiedlichen Formen anzuordnen, während es vor ihrer Einführung nur möglich war rechteckige *Räume* zu konstruieren. Diese zwei Typen von Steinen wurden in jeweils sechs Farben ausgearbeitet und prototypisch ausgedruckt. Das Aussehen der ersten Prototypen war dabei noch relativ abstrakt gehalten und wurde von Bodenmosaiken inspiriert. Das überarbeitete Spielkonzept stellte sich als tauglich heraus, sodass mit der Konzeption der digitalen Repräsentation des Spiels begonnen werden konnte.

Schon der zu diesem Zeitpunkt aktuelle Entwurf des Spiels sah vor, dass es einen speziellen zentralen Spielstein geben sollte, von welchem ausgehend die SpielerInnen ihre jeweils gezogenen Spielsteine anlegen sollten. Diese Idee ist angelehnt an das Strategiespiel *Dungeon Keeper*, bei dem es ebenfalls einen zentralen Raum – das Dungeonherz – gibt, von welchem ausgehend die SpielerInnen ihren Dungeon aufbauen müssen.

Im nächsten Schritt, wurde ein neuer Typ von Spielstein eingeführt, der über eine *Wand* verfügt, welche jedoch nicht durchgängig abgeschlossen ist. Mit diesem neuen Spielstein fand das Verbinden von abgeschlossenen *Räumen* und die Regel, die besagt, dass eine Einheit derjenigen Person, die sie vervollständigt hat, mehr Punkte bringt, wenn sie von der Mitte aus erreichbar ist, Einzug in das Spiel. Um dieses Konzept, des Verbindens der abgeschlossenen *Räume* und auch das Spielkonzept insgesamt besser zu vermitteln, wurde ein Narrativ eingeführt und die abstrakte Gestaltung der Spielsteine durch eine ikonische ersetzt. Mehrere Entwürfe, die in Abbildung 5.1 zu sehen



Abbildung 5.2: Visualisierung der Bewegungsmöglichkeiten im Spiel *The Banner Saga*

sind, wurden entwickelt, wovon eines vorsah die Spieler sollen Dörfer aus Stelzenhäusern bauen, wobei verschiedene Stämme jeweils unterschiedliche Formen von Plattformen und Brücken benutzten. Ein weiteres basierte auf der Idee ein Geflecht von Pflanzen zu erstellen, wobei Pflanzen unterschiedlicher Art nur an bestimmten Punkten miteinander verbunden werden sollten. Schlussendlich stellte sich eine Variante, die der ursprünglichen der Gestaltung der Spielsteine relativ ähnlich war, als am geeignetsten heraus.

Es wurde nun also ein Narrativ eingeführt, wonach Drachen in einer Höhle ihre Eier ausbrüten und dafür *Räume* brauchen. Auf den Spielsteinen, die entweder eine *Wand* oder eine *Tür* aufwiesen, befindet sich jeweils ein Drache, auf den offenen ein Nest mit Eiern. Außerdem wurden die *Wände* als durchgehende Balken zusätzlich um Elemente in den Ecken der Spielsteine ergänzt, die nun ebenfalls auf offenen Spielsteinen zu finden sind. Diese Änderung wurde nötig, da in bestimmten Situationen für die SpielerInnen nicht erkennbar war, dass Spielsteine, die zusammen laut Spielregel eine durchgehende *Wand* bildeten, auch tatsächlich zusammen ein solche formierten.

Nach ausgiebigen Tests der Hardwareplattform, stellte sich heraus, dass die Fiducials einen gewissen Detailgrad nicht übersteigen sollten, um eine akzeptable Detektion zu garantieren. Dies führte zur Auswahl des Fiducialsets "Amoeba Mini" aus dem reacTIVision-Toolkit, das 12 unterscheidbare Fiducials enthält. Dadurch ergab sich die Reduktion der möglichen Farben der Spielsteine auf vier mit jeweils drei unterschiedlichen Typen von Steinen, wie sie in Kapitel 3 beschrieben wurden.

### 5.3 Design der virtuellen Oberfläche

Der erste Entwurf der virtuellen Komponente der Benutzerschnittstelle sah vor, dass der zentrale Stein quadratisch sein und eine Seitenlänge von jeweils zwei gewöhnlichen

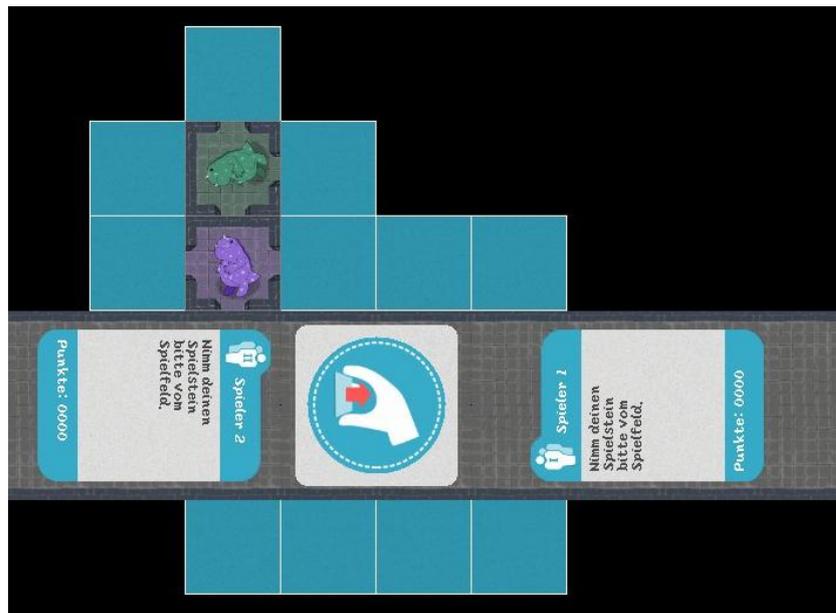


Abbildung 5.3: Visualisierung der gültigen Positionen eines Spielsteins

Spielsteinen aufweisen sollte. Dieser Stein sollte nur virtuell existieren, also nur auf der Tischfläche eingeblendet werden, um zu verhindern, dass er beim Spielen verschoben wird. Gleichzeitig ermöglicht die Tatsache, dass der Stein nur als Grafik existiert, die Möglichkeit an dieser Stelle für die SpielerInnen relevante Informationen, wie visuelle Hinweise, einzublenden.

Da die SpielerInnen keinerlei Vorkenntnisse der Spielregeln besitzen, muss ihnen vermittelten werden, wo sie die Spielsteine jeweils platzieren können. Da dies – den Spielregeln gemäß – nicht an beliebiger Stelle erlaubt ist, sondern ein Spielstein nur neben einem bereits auf dem Spielfeld befindlichen Stein platziert werden darf, wurde entschieden die gültigen Positionen eines Spielzugs zu signalisieren. Dazu werden an den Stellen, an denen ein Spielstein grundsätzlich positioniert werden kann – unabhängig von dessen Orientierung – Felder in Größe des Spielsteins eingeblendet. Dieser Ansatz, die möglichen Spielzüge hervorzuheben, findet auch in Computerspielen Anwendung. In rundenbasierten Rollen- oder Strategiespielen etwa, werden die Felder auf dem Spielplan, auf welche die Spielfigur bewegt werden kann, farbig hervorgehoben (vgl. Abbildung 5.2). Durch die Größe und Form der Hinweiskfelder, die jener der Spielsteine entspricht, wird eine zusätzliche Affordance erzeugt, ähnlich wie Kinderspielen, bei denen unterschiedlich geformte Spielsteine in Vertiefungen unterschiedlicher Form gesteckt werden müssen. Die Umsetzung in einer fortgeschrittenen Version des Prototypen ist Abbildung 5.3 zu entnehmen.

Zusätzlich dazu sollten die Möglichkeiten einen Spielstein zu Platzieren auch über visualisiertes Schlüssel-Schloss-Prinzip vermittelt werden. An jeder Seite der bereits auf

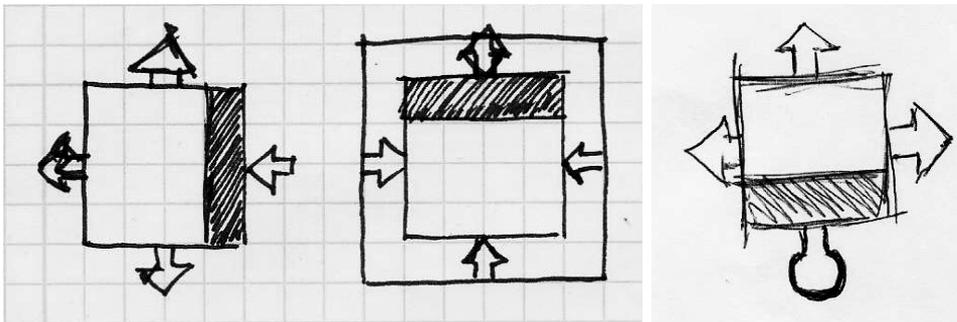


Abbildung 5.4: Entwurf visueller Hinweise in Form eines Schlüssel-Schloss-Prinzips

dem Spielfeld befindlichen Spielsteine, an der ein Anlegen möglich wäre, wird dabei eine *Buchse* einer Bestimmten Form angezeigt. Wird der zu platzierende Spielstein nun auf die Spielfläche gelegt, werden an den entsprechenden Seiten – je nachdem wie und wo er angelegt werden kann – die jeweilig passenden *Schlösser* der entsprechenden Form eingeblendet. In Abbildung 5.4 ist ein Entwurf dieses Konzepts zu sehen, das jedoch verworfen wurde, da es einerseits erst dann möglich ist, die entsprechenden Informationen zur Darstellung der *Schlösser* und *Schlüssel* zu ermitteln, wenn bereits bekannt ist welcher Stein gezogen wurde – wenn er also bereits auf die Spielfläche gelegt wurde, und andererseits die gültige Platzierung von der Ausrichtung des Spielsteins abhängt.

Aus diesem Grund fiel die Entscheidung, die SpielerInnen mit Symbolen darüber zu informieren, wenn sie einen Spielstein regelwidrig positioniert haben. Dabei Wird unterschieden welche Regel konkret gebrochen wird und ein entsprechendes Symbol eingeblendet. Gleichsam erhalten die SpielerInnen einen visuellen Hinweis, wenn sie einen Spielstein korrekt platziert haben. Durch zusätzliches auditives Feedback in Form von Soundeffekten, die abgespielt werden, wenn ein Fehler auftritt beziehungsweise, wenn ein Spielzug erfolgreich beendet wurde, wird dies noch unterstützt. Die dafür eingesetzten Symbole sind der Abbildung 5.6 zu entnehmen. Die verwendeten Piktogramme basieren dabei auf einer abstrahierten Darstellung der betreffenden Spielsituation auf rötlichem Hintergrund, die mit einem abgerundeten roten Rahmen umgeben und ebenfalls rot durchgestrichen ist. Bei Situationen in denen es um das Zusammenstoßen zweier Spielsteine geht, sind die relevanten Stoßkanten durch Pfeile markiert.

Zusätzlich zu den Symbolen, die auf regelwidrige Spielzüge hinweisen sollen, gibt es Symbole, die Hinweise darauf geben, was die Spieler jeweils als nächstes tun sollten, wie sie in Abbildung 5.7 zu sehen sind. So gibt es ein Symbol, dass die SpielerInnen darauf hinweist, ihren Spielstein von der Tischoberfläche zu nehmen, sowie ein Symbol, dass die SpielerInnen darauf hinweist, einen neuen Spielstein zu ziehen. Auch werden Symbole verwendet um zu signalisieren welche/r der SpielerInnen als nächste/r an der Reihe ist. Schließlich zeigt ein Symbol an, wenn ein *Raum* abgeschlossen wurde.

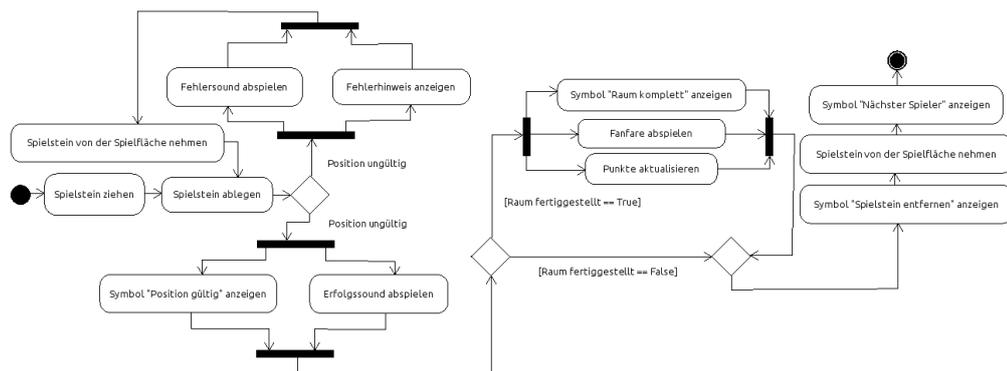


Abbildung 5.5: Darstellung eines Spielzugs in Form eines Aktivitätsdiagramms

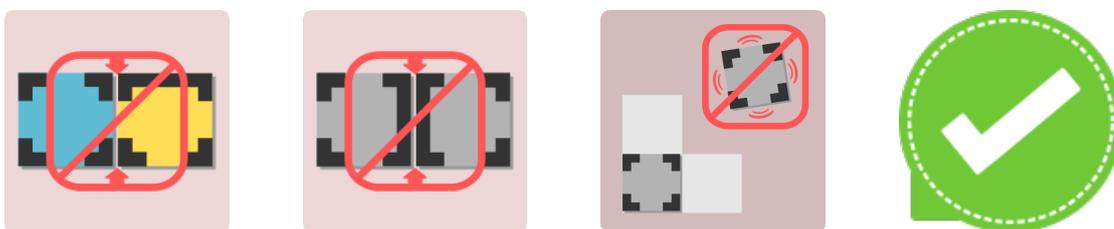


Abbildung 5.6: Symbole für die Situationen v.l.n.r.: Zwei Spielsteine unterschiedlicher Farbe stoßen an ihren offenen Seiten aneinander, ein Spielstein stößt mit der *Wandseite* direkt an eine *Wand*, der Spielstein grenzt nicht an einen zweiten Spielstein an, der aktuelle Spielzug ist gültig



Abbildung 5.7: Symbole für die Situationen v.l.n.r.: SpielerIn 1 ist an der Reihe, der aktuelle Spielstein soll entfernt werden, ein neuer Spielstein soll gezogen werden, ein *Raum* wurde fertiggestellt

Der mögliche Ablauf eines Spielzugs und die sich daraus ergebenden Rückmeldungen des Systems ist in Abbildung 5.5 dargestellt. Je nachdem welche Fehlersituation auftritt, unterscheidet sich dabei das angezeigte Symbol. Ein Spielzug gilt erst dann als beendet, wenn der Spielstein regelkonform angelegt wurde. In diesem Fall wird noch unterschieden ob durch den Spielzug ein *Raum* vervollständigt wird und falls dies zutrifft den BenutzerInnen wiederum entsprechendes Feedback gegeben.

# Evaluierung des Prototypen

Das Ziel bei der Konzipierung des Systems war es, den BenutzerInnen, durch auditives und graphisches Feedback, die Information zur Verfügung zu stellen, die es ihnen ermöglicht die Spielregeln selbst zu ermitteln, ohne vor Beginn des Spiels auf die konkreten Ziele oder die erlaubten Spielzüge hingewiesen worden zu sein. Dabei wurde versucht nicht einfach die Spielregeln in Form eines Tutorials zu vermitteln, in dem den SpielerInnen vorgeschrieben wird, welche Aktionen sie als nächste zu setzen haben, sondern diese hauptsächlich durch die Gestaltung der Oberfläche erfasst, beziehungsweise durch Ausprobieren herausgefunden werden können.

## 6.1 Methode

Grundsätzlich kann man sich mehrerer verschiedener Methoden zur empirischen Evaluierung der Benutzerschnittstelle eines Systems bedienen. Die unter anderem von Jakob Nielsen entwickelte heuristische Evaluierung [NM90], bei der eine Oberfläche von Interfaceexperten geprüft und anhand eines festgelegten Kriterienkatalogs bewertet wird, hat den Vorteil, dass damit vor allem in frühen Designstadien effizient Probleme mit der Benutzerschnittstelle identifiziert werden können. Allerdings sind dafür im Normalfall mehrere Experten notwendig, die das Design evaluieren und nach einer möglichst adäquaten Heuristik bewerten.

Als Alternative hat sich vor allem die Methode des Usability Testing bewährt. Dabei werden Versuchspersonen, die möglichst repräsentativ für die späteren Zielgruppe des System sein sollten, dazu angehalten, Aufgabenstellungen mit der zu testenden Oberfläche zu lösen, die später im Realbetrieb auch vorkommen. Dabei soll festgestellt werden an welchen Stellen Probleme bei der Benutzung auftreten.

Die Versuchspersonen werden zum Lauten Denken angehalten [Lew82]. Das Ziel dabei ist es, ein Bild von den kognitiven Prozessen zu erhalten, die bei der Bearbeitung einer

Aufgabe ablaufen. Dazu ist es wichtig, dass die Testpersonen möglichst alle Gedanken verbalisieren und nicht nur jene, die sie für relevant befinden. Die Äußerungen werden dabei durch die Versuchsleitung möglichst objektiv protokolliert. Auch eine zusätzliche Videoaufzeichnung ist üblich.

### 6.2 Aufbau des Tests

Zuerst wurden die TeilnehmerInnen kurz über ihre Gewohnheiten im Bezug auf Computer- und Videospiele sowie auch Gesellschaftsspiele befragt, um festzustellen ob zumindest Grundkenntnisse beider Domänen beziehungsweise ein Interesse für zumindest eine der Domänen vorhanden ist.

Die Versuchspersonen erhielten eine kurz gehaltene verbale Einführung in den Versuch. Sie wurden angewiesen sich an den gegenüberliegenden Seiten des Tisches zu platzieren. Weiters wurde einem/r der SpielerInnen ein mit Spielsteinen gefüllter Beutel gegeben und die Anweisung erteilt, die VersuchsteilnehmerInnen sollen abwechselnd blind Steine aus dem Beutel ziehen. Es wurde ihnen die Aufgabe erteilt, dass Spiel gemeinsam zu spielen und anhand der Hinweise, die sie durch die Benutzeroberfläche erhalten herauszufinden, wie es funktioniert. Abschließend wurden die TeilnehmerInnen gebeten ihre Gedanken zu verbalisieren und ihre Aktionen zu kommentieren, sowie sich miteinander darüber auszutauschen. Im Anschluss begannen die Testpersonen mit der Lösung der ihnen gestellten Aufgabe. Sie wurden dabei beobachtet sowie gefilmt. Abhängig vom Verlauf dauerte eine Evaluierungssession zwischen 30 und 45 Minuten. Im Anschluss wurden die Testpersonen noch nach ihrer Meinung zum Spiel selbst sowie zu den einzelnen Hinweisen und verwendeten Symbolen befragt. Weiters wurden ihnen Fragen zu speziellen Situationen, die während des Spielens aufgefallen waren, gestellt.

#### 6.2.1 TeilnehmerInnen

Die Gesamtheit der TeilnehmerInnen der Evaluierung setzte sich aus zehn aktiven und ehemaligen StudentInnen zusammen, die alle über Erfahrung mit Computern beziehungsweise Computerspielen verfügten. Bis auf eine Person gaben alle TeilnehmerInnen an zumindest gelegentlich Computerspiele zu spielen. Gleiches gilt für Gesellschaftsspiele, wobei diese von den meisten TeilnehmerInnen seltener gespielt wurden. Eine Versuchsperson gab an aktuell weder regelmäßig Computerspiele noch Gesellschaftsspiele zu spielen, hatte aber in der Vergangenheit dennoch Erfahrungen damit gemacht.

### 6.3 Ergebnisse

Es hat sich gezeigt, dass alle Evaluierungsgruppen das Spielziel ermitteln konnten. Es dauerte zwischen 11 und 25 Minuten bis jeweils die erste Testperson pro Gruppe Punkte erzielen konnte. Während die Grundregeln der möglichen Platzierung von Steinen in der Regel schnell verstanden wurden und anhand der Symbole, die den BenutzerInnen

angezeigt wurden, wenn diese einen Stein an einer ungültigen Stelle platziert hatten, und der dazugehörigen kurzen schriftlichen Erläuterung erfolgreich vermittelt werden konnten, wurden in den meisten Spielen die ersten Punkte eher zufällig erzielt. Das Spielziel wurde den Testpersonen erst in dem Moment bewusst, in dem der erste *Raum* von einem/r SpielerIn fertiggestellt wurde und dieser dafür Punkte erhielt.

### 6.3.1 Probleme

Im Zuge der Evaluierung wurden einige Probleme identifiziert, die im Folgenden ausgeführt werden. Weiters sollen spezielle Anregungen der Testpersonen hier Erwähnung finden.

#### Farbgebung

Die Farbgebung der Spielsteine stellte sich als problematisch heraus, da sich die Farben der virtuellen Repräsentationen der Spielsteine – abhängig vom Umgebungslicht – zum Teil deutlich von denen der physischen Steine unterschieden. So fiel es vielen Testpersonen schwer grüne und blaue Spielsteine zu unterscheiden. Sowohl grüne als auch blaue physische Spielsteine wurden virtuellen blauen Steinen zugeordnet. Ähnliche Verwechslungen ergaben sich auch zwischen den pinken und violetten Steinen. Dabei ist anzumerken, dass die Testpersonen keine Probleme die verschiedenen Farben in ihrer jeweiligen Domäne auseinander zu halten. Die Schwierigkeiten ergaben sich erst durch die Kombination der virtuellen und physischen Repräsentationen.

#### Informationsfelder

Die Informationsfelder, von denen sich eines an jeder der einander jeweils gegenüberliegenden Seiten des Spielplans, an ein/e SpielerInnen platziert ist, befindet, wurden von manchen Testpersonen kaum beachtet. Es war für sie nicht ersichtlich, wann sich der Inhalt dieses Feldes geändert hatte, weshalb sie nicht nachsahen, ob dort eventuell für ein Hinweis zu finden wäre, der für sie hilfreich sein könnte. Sie nahmen die Inhalte des Felds eher als statisch wahr und bemerkten erst spät, dass sie ihm unter Umständen nützliche Information entnehmen hätten können. Auch wurde der Wunsch geäußert auf Abruf kontextabhängige Hinweise erhalten zu können, indem man in etwa den aktuellen Spielstein auf ein speziellen Bereich des Spielfelds legt.

#### Tipps

Die Hinweistexte, die den SpielerInnen dann in den Informationsfeldern angezeigt wurden, wenn diese gerade nicht an der Reihe waren, führten manche BenutzerInnen in die Irre. Es ließ sich beobachten, dass diejenigen Testpersonen, die die Hinweistexte beachtetten, sich sehr auf die darin erhaltenen Hinweise fokussierten. Die Hinweistexte waren allerdings hauptsächlich darauf ausgerichtet, zu vermitteln, darauf zu achten, dass es möglich sein soll von einem *Raum* aus in die Mitte zu gelangen und nicht das eigentlich Ziel, nämlich *Räume*, die an allen Seiten von *Wänden* umgeben sein sollten, zu bauen. So versuchten die betreffenden Testpersonen vorwiegend herauszufinden was es mit diesem Hinweis auf

sich hatte. Neben dem Hinweis darauf was es bedeutet, wenn ein *Raum* abgeschlossen ist, wurde auch eine Information dazu, was es bedeutet, Drachen einzusperren – nämlich dass der *Raum* über keinen Zugang zur Mitte des Spielplans verfügt – eingefordert.

### **Mögliche Positionen der Spielsteine**

Im Allgemeinen hatten die Versuchsperson keine Probleme herauszufinden, dass die Spielsteine auf den markierten Feldern zu platzieren waren. Ein/e TeilnehmerIn äußerte sich allerdings dahingehend, nicht zu wissen, ob die Steine auf den Feldern abzulegen seien, oder aber auf dem Rest des Spielfelds. Mehrere SpielerInnen versuchten auch ihre Spielsteine auf bereits besetzten Feldern abzulegen, was keinerlei Reaktion des Systems auslöste. Aus diesem Grund war ihnen nicht klar, ob es sich dabei um ein Fehlverhalten ihrerseits handelte oder ob die nötigen Voraussetzungen für eine entsprechende Reaktion nicht gegeben waren.

### **Symbole bei Fehlplatzierungen**

Die Symbole die verschiedene Fehlersituationen signalisierten wurden im Allgemeinen gut verstanden. Als problematisch stellte sich dabei heraus, dass die Ausrichtung der Symbole nicht immer der Position der Person, welche gerade am Zug war, entsprach. Dadurch standen die Symbole teilweise Kopf und waren entsprechend schlechter zu erkennen. Auch wurde der Wunsch geäußert, die Symbole mögen genauer der aktuellen Spielsituation entsprechen – zum Beispiel indem man die Ausrichtung des Symbols der tatsächlichen Lage der Spielsteine zueinander auf dem Spielfeld entsprechend ausführt – da es so einfacher sei die Darstellung, die nicht notwendigerweise der Ausrichtung der Tatsächlichen Spielsituation auf die tatsächliche Spielsituation umzulegen.

### **Narrativ**

Einige Versuchspersonen stießen auf Ungereimtheiten im Narrativ des Spiels. Zum Beispiel wurde erwähnt, dass Eier in *Räumen* ohne Drachen nicht ausgebrütet werden könnten und es gleichsam in *Räumen* ohne Eier nichts auszubrüten gäbe. Andere entwickelten Theorien darüber, wie das Verhältnis von Eiern zu Drachen pro *Raum* auszusehen hätte. Beide dieser Aspekte werden durch die eigentlichen Spielregeln nicht abgedeckt.

### **Punktevergabe**

Die Punktevergabe war für den Großteil der TeilnehmerInnen nicht durchschaubar. Sie konnten das Berechnungsschema nicht nachvollziehen, nach welchem die Punkte für einen *Raum* vergeben wurden. Zusätzlich fiel auf, dass der Wunsch geäußert wurde im Nachhinein noch nachvollziehen zu können, welche/r SpielerIn einen bestimmten *Raum* fertiggestellt hatte.

### **Erkennung der Spielsteine**

Es zeigte sich, dass die Erkennung der Fiducials durch das System teilweise kleiner Probleme machte. Vereinzelt kam es zu Situationen, in den ein Spielstein erkannt und dann sofort wieder verloren wurde. Dieses Verhalten führte dann zu einer Schleife des Erkennens eines neuen Steines und des Entfernen desselben durch die Tracking-Software, bis dieser neu positioniert wurde.

### **Spielprinzip**

Es wurde der Vorschlag eingebracht, wonach man den gezogenen Spielstein nicht notwendigerweise sofort ablegen müsse, sondern sich einen kleinen Vorrat an Spielsteinen anlegen könne und bei Bedarf dann daraus einen Stein ausspielen dürfe.

## **6.4 Fazit**

Im Zuge der Evaluierung konnten diverse Schwachstellen aufgedeckt werden, durch deren Beseitigung ein effizienteres Erlernen der Spielregeln verhindert wurde. Gleichzeitig hat sich gezeigt, dass sämtliche Testpersonen das Spielziel in Erfahrung bringen konnten und die Grundregeln der Positionierung der Spielsteine in der Regel sehr gut anhand der eingeblendeten Hinweise vermittelt werden konnten. Das Spiel selbst wurde – einmal verstanden – von den meisten Testpersonen durchweg positiv bewertet.



# Analyse und Ausblick

## 7.1 Resümee

Ziel der Arbeit war es ein System (vgl. Abbildung 7.1) zu entwickeln, das den Rahmen eines klassischen Gesellschaftsspiels, bei dem die SpielerInnen einander gegenüber sitzen oder stehen und in direkte persönliche Interaktion treten, mithilfe von Computertechnologie soweit zu erweitern, dass die Vorbereitungszeit für die SpielerInnen, die bei herkömmlichen Gesellschaftsspielen vor allem für das Erlernen der Spielregeln notwendig ist, weitgehend eliminiert werden kann.

Dies wurde versucht mithilfe eines Tangible User Interfaces zu erreichen, das den BenutzerInnen hauptsächlich mittels Rückmeldungen in Reaktion auf deren Handlungen die

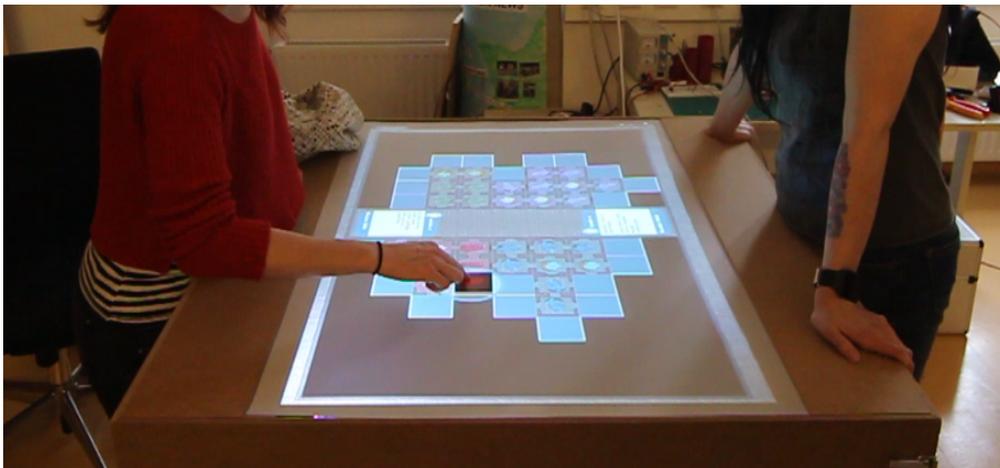


Abbildung 7.1: Finale Version des Prototypen wie er bei der Evaluierung zum Einsatz kam

Regeln und Ziele des Spieles vermitteln sollte. Dabei wurde sowohl von grafischem wie auch auditivem Feedback Gebrauch gemacht.

Die in Kapitel 6 beschriebene Evaluierung hat gezeigt, dass es allen Versuchsgruppen möglich war das Spielziel in weniger als 30 Minuten zu ermitteln. Die Vorlaufzeit, die bis zum Erreichen der ersten Punkte nötig war, unterschied sich dabei von Gruppe zu Gruppe teils stark. In den meisten Fällen reichten die den SpielerInnen zur Verfügung gestellten Hinweise nicht aus um das Spielziel eindeutig zu vermitteln. Vielmehr wurde es in aller Regel zufällig entdeckt. Im Gegensatz dazu konnten die Regeln für die Positionierung der Spielsteine durch entsprechendes Feedback erfolgreich vermittelt werden. Für gewöhnlich machten die Testpersonen, sobald eine Fehlersituation ein oder zweimal im Spiel aufgetreten war, denselben Fehler kaum wieder.

Als eine potenzielle Fehlerquelle stellten sich die Diskrepanzen in der wahrgenommenen Farbgebung der Spielsteine und ihrer digitalen Repräsentationen heraus, die dazu führten dass SpielerInnen die Farben des aktuellen Spielsteins nicht eindeutig der entsprechenden Farbe auf der projizierten Oberfläche zuordnen konnten. Als eine weitere Ursache für Unklarheiten stellten sich die textuellen Hinweise heraus, die von den Testpersonen teilweise nicht wahrgenommenen wurden, gleichzeitig aber nicht klar genug formuliert beziehungsweise falsch priorisiert waren und somit ihren Zweck den Testpersonen zu helfen das Spielziel herauszubekommen, verfehlten.

Auch die Vergabe der Punkte war für praktisch alle Testpersonen nicht durchschaubar und minderte somit das Spielerlebnis etwas.

Trotz der genannten Probleme konnte gezeigt werden, dass die Integrierung von Computertechnologie in Gesellschaftsspiele Potenzial hat das Spielerlebnis zu bereichern und den zeitlichen Aufwand, der für das Erlernen der Spielregeln im Kontext von klassischen, nicht technologiegestützten Gesellschaftsspielen notwendig ist, zu reduzieren.

### 7.2 Ausblick

Im Zuge der Evaluierung wurden zahlreiche Verbesserungsmöglichkeiten aufgezeigt, die die Erlernbarkeit des Spiels erhöhen können. So bietet eine Überarbeitung der textuellen Hinweise – einerseits durch präzisere Formulierungen, andererseits durch prominentere Platzierung – Potential den SpielerInnen ihre eigentliche Aufgabe effizienter zu vermitteln. Dies ist auch durch einen zusätzlichen Kontextbezug der Formulierungen möglich. Weiters sollte den SpielerInnen klar signalisiert werden, wenn sich die Hinweistexte geändert haben. Die Abwesenheit von Feedback, wenn Spielsteine auf bereits besetzten Feldern platziert werden, lässt sich ebenfalls beheben.

Auch die Farbgebung der Spielsteine hat sich als verbesserungswürdig herausgestellt. Durch klarer unterscheidbare Farben und Anpassung der Druckfarben, damit diese den Farben der virtuellen Oberfläche besser entsprechen, kann die Zuordnung der Spielsteine potenziell vereinfacht werden.

Um die Punktevergabe transparenter zu gestalten, könnten entsprechende Animationen eingeführt werden, die nach dem Vervollständigen eines *Raumes* für jeden einzelnen Spielstein die entsprechende Punktezahl sowie die Bonuspunkte, die man erhält, wenn eine Verbindung zur Mitte des Spielplans besteht, signalisieren. Weiters könnte man die einzelnen abgeschlossenen Räume entsprechend dem/der SpielerIn markieren, der/die ihn vervollständigt hat.

Auch das Spielprinzip bietet noch Raum für weitere Entwicklungen. So könnte man es zu einer Voraussetzung machen, dass ein Raum mindestens einen Drachen und ein Nest aufweisen muss, damit das Brüten überhaupt möglich ist. Auch eine Variante, bei der die SpielerInnen am Anfang einen Vorrat von mehreren Steinen erhalten und diese dann je nach situativer Gegebenheit ausspielen können, ist denkbar.

Die Probleme beim Tracking der Fiducials, die dazu führten, dass die Tracking-Software teilweise die Symbole kurzzeitig nicht detektieren konnte, lassen sich wahrscheinlich durch eine Feinabstimmung der Parameter der Bildverarbeitungskette minimieren. Auch eine spezielle Anpassung der Client-Software, wonach ein Fiducial erst nach einer bestimmten Zeit als abwesend gilt und das gleiche Fiducial, so es an der selben Stelle, an der gerade ein Fiducial mit der selben ID nicht mehr erkannt wurde, als ident gilt und somit kein entsprechender Event ausgelöst wird.

# Abbildungsverzeichnis

3.1	Eine mögliche gültige Anordnung von Spielsteinen . . . . .	17
4.1	3D-Modell der Rahmenkonstruktion . . . . .	22
4.2	Rahmenkonstruktion vor der Installation der Elektronik . . . . .	22
4.3	Tisch mit installierter Hardware . . . . .	22
4.4	CAD-Modell der Halterung für die Kameraplatine . . . . .	23
4.5	3D gedruckte Halterung mit installierter Platine . . . . .	23
4.6	Klassendiagramm der Kernkomponenten . . . . .	28
5.1	Alternative Gestaltungsentwürfe der Spielsteine . . . . .	34
5.2	Visualisierung der Bewegungsmöglichkeiten im Spiel <i>The Banner Saga</i> . . . .	35
5.3	Visualisierung der gültigen Positionen eines Spielsteins . . . . .	36
5.4	Entwurf visueller Hinweise in Form eines Schlüssel-Schloss-Prinzips . . . . .	37
5.5	Darstellung eines Spielzugs in Form eines Aktivitätsdiagramms . . . . .	38
5.6	Symbole für die Situationen v.l.n.r.: Zwei Spielsteine unterschiedlicher Farbe stoßen an ihren offenen Seiten aneinander, ein Spielstein stößt mit der <i>Wandseite</i> direkt an eine <i>Wand</i> , der Spielstein grenzt nicht an einen zweiten Spielstein an, der aktuelle Spielzug ist gültig . . . . .	38
5.7	Symbole für die Situationen v.l.n.r.: SpielerIn 1 ist an der Reihe, der aktuelle Spielstein soll entfernt werden, ein neuer Spielstein soll gezogen werden, ein <i>Raum</i> wurde fertiggestellt . . . . .	38
7.1	Finale Version des Prototypen wie er bei der Evaluierung zum Einsatz kam .	45

# Glossary

**digitale Repräsentation** Im Unterschied zu den physischen Komponenten eines TUI auf eher klassischem Weg erzeugte Ausgaben aus einem Computersystem (z.B. grafische Ausgaben oder Audio) (vgl. [UI00]. 3

**Fiducial Marker** Eine Objekt, dass im Gesichtsfeld eines Bildverarbeitungssystems platziert wird und dort als Referenzpunkt, dessen Position im Raum vom System erfasst wird, dient.. 19

**Flow** Flow bezeichnet das Gefühl eines mentalen Zustands vollkommener Konzentration, das als beglückend empfunden wird.. 5

**WIMP** Das englische Akronym steht meist für “Windows”, “Icons”, “Menus” und “Pointer” und bezeichnet das dominierende Konzept grafischer Benutzerschnittstellen.. 12



# Literaturverzeichnis

- [Ack11] Diane Ackerman. *Deep play*. Vintage, 2011.
- [AD12] Ernest Adams and Joris Dormans. *Game mechanics: advanced game design*. New Riders, 2012.
- [Bal66] Gustav Bally. *Vom Spielraum der Freiheit*. Schwabe, 1966.
- [BBMT06] Tamar Ben-Bassat, Joachim Meyer, and Noam Tractinsky. Economic and subjective measures of the perceived value of aesthetics and usability. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 13(2):210–234, 2006.
- [CL16] Inc. Code Laboratories. CL-Eye Platform Driver camera access in third party applications, 2016.
- [DFAB03] Alan Dix, Janet E Finlay, Gregory D Abowd, and Russell Beale. Human-computer interaction. 2003.
- [Flo97] Finding Flow. The psychology of engagement with everyday life. *Mihaly Csikszentmihalyi*, 1997.
- [Gav91] William W Gaver. Technology affordances. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 79–84. ACM, 1991.
- [GFA09] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. A survey of software learnability: metrics, methodologies and guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 649–658. ACM, 2009.
- [Gib79] James J Gibson. The ecological approach to visual perception. 1979.
- [HSDA07] Jan Hartmann, Alistair Sutcliffe, and Antonella De Angeli. Investigating attractiveness in web user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 387–396. ACM, 2007.
- [Huj38] Johan Hujzinga. *Homo ludens*. Stuttgart: Rowohlt Taschenbuch, 1938.

- [ISO01] Software Product Evaluation — Quality Characteristics and Guidelines for the User. Standard, International Organization for Standardization, Geneva, CH, 2001.
- [JGAK07] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146. ACM, 2007.
- [Kah11] Agnes Kahofer. Musictheorytable : Ein tangible user interface zur unterstützung des musikkundeunterrichts an musikschulen. Master’s thesis (Diplomarbeit), Vienna University of Technology, 2011.
- [Kal09] Martin Kaltenbrunner. reactivation and tuio: a tangible tabletop toolkit. In *Proceedings of the ACM international Conference on interactive Tabletops and Surfaces*, pages 9–16. ACM, 2009.
- [KK95] Masaaki Kurosu and Kaori Kashimura. Apparent usability vs. inherent usability: experimental analysis on the determinants of the apparent usability. In *Conference companion on Human factors in computing systems*, pages 292–293. ACM, 1995.
- [Kos13] Raph Koster. *Theory of fun for game design*. Ö’Reilly Media, Inc.", 2013.
- [Lew82] Clayton Lewis. Using the “thinking-aloud” method in cognitive interface design (ibm research rep. no. rc 9265 [# 40713]). *Yorktown Heights, NY: IBM Thomas J. Watson Research Center*, 1982.
- [Lin07] Gitte Lindgaard. Aesthetics, visual appeal, usability and user satisfaction: what do the user’s eyes tell the user’s brain? *Australian Journal of Emerging Technologies & Society*, 5(1), 2007.
- [LRZ10] Uwe Laufs, Christopher Ruff, and Jan Zibuschka. Mt4j-a cross-platform multi-touch development framework. *arXiv preprint arXiv:1012.0467*, 2010.
- [MH00] Joanna McGrenere and Wayne Ho. Affordances: Clarifying and evolving a concept. In *Graphics Interface*, volume 2000, pages 179–186, 2000.
- [MYJ94] WS Maarten, FB Yvonne, and AC Jacobijn. The think aloud method: a practical guide to modelling cognitive processes, 1994.
- [Nie94] Jakob Nielsen. *Usability engineering*. Elsevier, 1994.
- [NM90] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256. ACM, 1990.
- [Nor88] Donald A Norman. *The psychology of everyday things*. Basic books, 1988.

- [Nor98] Donald A Norman. *The invisible computer: why good products can fail, the personal computer is so complex, and information appliances are the solution*. MIT press, 1998.
- [Nor05] Donald A Norman. *Emotional design: Why we love (or hate) everyday things*. Basic books, 2005.
- [Pre01] Marc Prensky. Fun, play and games: What makes games engaging. *Digital game-based learning*, 5:1–05, 2001.
- [RP90] Irvin Rock and Stephen Palmer. Gestalt psychology. *Sci Am*, 263:84–90, 1990.
- [SB95] Paulo J Santos and Albert Badre. Discount learnability evaluation. 1995.
- [SS09] Juergen Sauer and Andreas Sonderegger. The influence of prototype fidelity and aesthetics of design in usability tests: Effects on user behaviour, subjective evaluation and emotion. *Applied ergonomics*, 40(4):670–677, 2009.
- [SS10] Andreas Sonderegger and Juergen Sauer. The influence of design aesthetics in usability testing: Effects on user performance and perceived usability. *Applied ergonomics*, 41(3):403–410, 2010.
- [SZ04] Katie Salen and Eric Zimmerman. *Rules of play: Game design fundamentals*. MIT press, 2004.
- [TKI00] Noam Tractinsky, Adi S Katz, and Dror Ikar. What is beautiful is usable. *Interacting with computers*, 13(2):127–145, 2000.
- [TM07] Manfred Thüring and Sascha Mahlke. Usability, aesthetics and emotions in human–technology interaction. *International Journal of Psychology*, 42(4):253–264, 2007.
- [Tra97] Noam Tractinsky. Aesthetics and apparent usability: empirically assessing cultural and methodological issues. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 115–122. ACM, 1997.
- [UI00] B. Ullmer and H. Ishii. Emerging frameworks for tangible user interfaces. *IBM Systems Journal*, 39:915–931, 2000.
- [WBA10] Sijie Wang, Allen Bevans, and Alissa N. Antle. Stitchrv: multi-camera fiducial tracking. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 287–290. ACM, 2010.