

Spiele als Multiuser-Anwendungen für Medienfassaden

Forschungsprojekt 2

Autor: Robin Schlegel
Matrikelnummer: 522402
Fachbereich: Wirtschaftswissenschaften II
Studiengang: Angewandte Informatik

Berlin, den 30. September 2012

Inhaltsverzeichnis

1. Einleitung	1
1.1. Einführung in Medienfassaden	1
1.2. Forschungsziele	2
1.3. Hintergrund: Errichtung des Forschungs- und Weiterbildungszentrum für Kultur und Informatik	2
2. Spiele für Medienfassaden	3
2.1. Definition: Computerspiel	3
2.2. Spielgenres	5
2.3. Besonderheiten der Mehrspielerinteraktion	7
2.3.1. Darstellung der Inhalte für mehrere Spieler	7
2.3.2. Mehrspieler-Steuerung	9
2.3.3. Kommunikation zwischen mehreren Geräten	9
2.4. Besonderheiten bei der Visualisierung an einer Medienfassade	9
2.5. Klassifikation von Spielen für eine Medienfassade	10
3. Technische Umsetzung	12
3.1. Auswahl des Spiels	12
3.2. Anforderungen	12
3.3. Benötigte Infrastruktur	13
3.3.1. Client: Eingabe	13
3.3.2. Server: Ausgabe	15
3.4. Aufbau der Medienfassade des FKI	16
3.4.1. Testlabor für die Medienfassade	17
3.5. Entwurf des Prototyps	18
3.5.1. Authentifizierung	18
3.5.2. Simple Directmedia Layer	19
3.5.3. Farbgebung	20
3.5.4. Benutzeroberfläche Server	20
3.5.5. Benutzeroberfläche Client	21
3.5.6. Kommunikation zwischen Client und Server	22
3.5.7. Programmstruktur	25

4. Ergebnis	26
4.1. Ausblick	26
A. Kommunikations-Protokoll	I

Abbildungsverzeichnis

1. Aufteilung des Bildschirms im Splitcreen-Modus	8
2. Benötigte Infrastruktur für ein Spiel mit Client-Server Architektur an der Medienfassade	14
3. Aufbau der Medienfassade des FKI-Gebäudes	17
4. Aufbau des Testlabors mit Projektionsleinwand und LED-Kacheln	18
5. Entwurf der Benutzeroberfläche des Servers	21
6. Benutzeroberfläche des Servers auf den LED-Kacheln im Testlabor	21
7. Startbildschirm auf dem Android-Client	23
8. Spielbildschirm auf dem Android-Client	23
9. Kommunikation zwischen Client und Server je Spielfigur	24

1. Einleitung

In den letzten Jahren sind die medialen Fassaden (Medienfassaden) sowohl für den Architekten als auch für den Betrachter immer attraktiver geworden. Es sind unzählige neue Medienfassaden mit farbenreichen und riesigen Visualisierungen entstanden. Das Stadtbild wird mehr und mehr von Medienfassaden geprägt.

Besonders interessant werden Medienfassaden jedoch erst, wenn eine Möglichkeit der Interaktion besteht. Dies bietet häufig einen Mehrwert für die Fassade. Besonders Spiele bieten sich aufgrund ihrer Charakteristika für Medienfassaden an.

Ziel dieser Arbeit ist es einen Überblick über das Potenzial von Spielen für Medienfassaden zu erstellen. Es soll gezeigt werden, inwiefern sie sich als Interaktionsmöglichkeit für eine Medienfassade anbieten und welche Art von Spielen hierfür besonders geeignet ist. Dabei liegt der Schwerpunkt auf Spielen, welche die Möglichkeit der Mehrspielerinteraktion bieten. Auf der Theorie aufbauend soll prototypisch ein Spiel umgesetzt werden, um die verschiedenen Problematiken bei Medienfassaden zu verdeutlichen.

Die Arbeit beginnt mit einer kurzen Einführung in Medienfassaden, den Zielen der Arbeit, sowie dem Hintergrund der Arbeit: Die Errichtung des Forschungs- und Weiterbildungszentrum für Kultur und Informatik auf dem Campus der HTW-Berlin. Im Abschnitt 2 folgt eine Analyse des Begriffs Computerspiel, eine Auflistung der Spielgenre und darauf aufbauend eine Klassifikation für Spiele für eine Medienfassade. Dabei werden die Besonderheiten der Mehrspielerinteraktion sowie einer Medienfassade erläutert. Auf der Theorie aufbauend folgt die Beschreibung der technischen Umsetzung eines Prototyps für die Medienfassade. Dabei werden Entwicklungsprobleme analysiert und Lösungsansätze erarbeitet. Abschließend folgt im Abschnitt 4 ein Fazit sowie ein kurzer Ausblick.

1.1. Einführung in Medienfassaden

Unter dem Begriff *Medienfassade* werden alle Arten von Fassaden verstanden, die mit Hilfe von computergesteuerten Displays oder Leuchtmitteln wandelbar sind. Es ist dabei ein wichtiges Merkmal, dass die Medienfassade ein integraler Bestandteil der Architektur ist und in ihrem Ausdruck der Nutzung und der Architektur entsprechen [Sau04].

Insbesondere die Interaktion mit einer Medienfassade rückt immer mehr in den Mittelpunkt. Laut Sauter werden vier Interaktionsstufen einer Medienfassade unterschieden [Sau04]: Grundlegend ist der Zustand *Autoaktiv*. Dieser Zustand liegt vor, wenn nur Bewegtbilder oder vergleichbares auf einer dynamischen Fassade abgespielt werden. Eine

1. Einleitung

direkte Interaktion mit dem Betrachter findet nicht statt. Die nächste Stufe ist *Reaktiv*, bei der die Fassade mit Hilfe von verschiedenen Sensoren auf ihr Umfeld reagiert. Des Weiteren gibt es den Zustand *Interaktiv*. Hierbei tritt der Mensch mit der Fassade in einen Dialog. Zuletzt gibt es noch *Partizipativ*. Bei dieser Art wird die Fassade durch eine interessierte Öffentlichkeit bespielt. Dadurch wird erreicht, dass die Öffentlichkeit sich mit der Fassade beschäftigt und identifiziert.

Computerspiele finden sich primär in der Stufe *Interaktiv*, aber auch bei *Partizipativ* oder *Reaktiv* sind kleinere Spiele vorstellbar.

1.2. Forschungsziele

Zum Abschluss dieser Forschungsarbeit sollen verschieden Ziele erreicht werden:

- Untersuchung der Mehrspielerinteraktion auf Probleme und Besonderheiten. Dabei soll insbesondere die Steuerung, die Spielfelddarstellung und der Datentransfer untersucht werden.
- Klassifikation von Spielen für die Medienfassade anhand von Merkmalen.
- Besonderheiten der Medienfassade bezüglich der verwendeten Technologien herausarbeiten.
- Implementierung eines Prototyps der als Grundlage für weitere Projekte dienen kann.

1.3. Hintergrund: Errichtung des Forschungs- und Weiterbildungszentrum für Kultur und Informatik

Hintergrund der Arbeit ist die Errichtung des Forschungs- und Weiterbildungszentrum für Kultur und Informatik (FKI) auf dem Campus der HTW-Berlin. Der Bau des Gebäudes hat im Dezember 2011 begonnen und wird voraussichtlich im Dezember 2012 abgeschlossen sein.

Bestandteil des Neubaus ist die Planung um Umsetzung einer Medienfassade. Der aktuelle Stand sieht dabei eine Kombination verschiedener Technologien für die Umsetzung einer Medienfassade vor. Es ist geplant, eine Verbindung von LED-Kacheln, Rückprojektion und Ambient-Light umzusetzen. Der detaillierte Aufbau der Medienfassade ist im Unterabschnitt 3.4 beschrieben. Die Medienfassade soll dabei auf verschiedene Einsatzmöglichkeiten geprüft werden, wobei in dieser Arbeit der Schwerpunkt auf Computerspielen, insbesondere welche mit Mehrspielerkomponente, liegt.

2. Spiele für Medienfassaden

Prinzipiell bieten Medienfassaden eine Vielzahl von Einsatzmöglichkeiten. Insbesondere wenn es eine Interaktionsmöglichkeit mit der Medienfassade vorhanden ist, kann diese für mehr als nur die Visualisierung von Daten genutzt werden. Naheliegender ist somit auch der Einsatz von Computerspielen. Klassiker wie Tetris oder Pong werden immer wieder zur Demonstration verwendet. Eine der ersten Umsetzungen von Spielen auf einer Medienfassade fand im Jahr 2001 in Berlin statt. Hierbei wurde im Projekt Blinkenlights Pong für eine Medienfassade umgesetzt [Bli].

Bevor über die verschiedenen Möglichkeiten von Spielen auf Medienfassaden eingegangen wird, soll zunächst geklärt werden, wie der Begriff *Computerspiel* definiert ist, dessen eindeutige Definition nicht einfach ist. Darauf aufbauend werden im Unterabschnitt 2.2 die gängigen Spielgenres vorgestellt und auf die Nutzung mit einer Medienfassade geprüft. Dabei wird auf die Besonderheiten der Mehrspielerinteraktion und der Nutzung einer Medienfassade als Display für ein Spiel eingegangen. Letztendlich werden im Unterabschnitt 2.5 Merkmale spezifiziert, die ein Computerspiel auszeichnen, welches besonders gut für die Umsetzung auf einer Medienfassade geeignet ist.

2.1. Definition: Computerspiel

Der Begriff *Computerspiel* setzt sich aus den Wörtern *Computer* und *Spiel* zusammen. Eine Klärung des Begriffs *Spiel* ist daher grundlegend für den Begriff *Computerspiel* notwendig. Der Philosoph Bernard Suits definiert ein Spiel wie folgt:

„Playing a game is the voluntary effort to overcome unnecessary obstacles.“
[SZ04, S. 76]

Greg Costikyan, ein Spieledesigner, definiert ein Spiel hingegen wie folgt:

„A game is a form of art in which participants termed players, make decisions in order to manage resources through game tokens in the pursuit of a goal.“
[SZ04, S. 78]

Diese unterschiedlichen Definitionen zeigen bereits, wie verschieden der Fokus bei der Definition eines Spiels liegen kann, wenn die Autoren aus so unterschiedlichen Bereichen stammen.

2. Spiele für Medienfassaden

Katie Salen und Eric Zimmerman haben unterschiedliche Definition auf ihre Kernaspekte hin untersucht und dabei folgende Merkmale herausgearbeitet [SZ04, S. 52]:

System: Ein System besteht aus vielen Bestandteilen, die untereinander in Verbindung stehen und zusammen ein komplexes Ganzes bilden. Dabei gibt es viele unterschiedliche Arten von Systemen, wie zum Beispiel ein soziales System. Allerdings haben alle Arten von Systemen vier Elemente: Objekte als Teile des Systems, Eigenschaften der Objekte und des Systems, Beziehungen zwischen den Objekten und eine Umgebung die das System umgibt.

Spieler: Die Person, die mit dem System interagiert. Dies kann eine einzelne Person sein, aber auch mehrere Personen.

Künstlich: Spiele grenzen sich von der realen Welt ab. Obwohl Spiele oft auch in der realen Welt auftreten, ist das Künstliche ein entscheidendes Merkmal eines Spiels.

Konflikt: Konflikte sind ein zentraler Punkt in Spielen. Dieser Konflikt kann sich in vielen verschiedenen Formen ausprägen, wie soziale Konflikte, Herausforderungen oder Kooperationen.

Regeln: Regeln bilden die Struktur eines jedes Spiels und sind somit ebenso ein zentrales Merkmal.

Abgeleitet aus diesen Punkten, ist die folgende Definition von dem Begriff *Spiel* entworfen worden, welche auch als Basis dieser Arbeit dient:

„A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome.“ [SZ04, S. 81]

Darauf aufbauend kann die Begrifflichkeit *Computerspiel* weiter definiert werden. Logischerweise muss irgendeine Art von Computer für die Nutzung des Spiels genutzt werden. Dabei gibt es jedoch unzählige Arten von Computern, die für Spiele genutzt werden können. Beispielhaft wären hier Spielkonsolen, Desktop-PCs, Handhelds, Smartphones oder Tablets zu nennen. Teilweise verschwimmen allerdings die Grenzen zwischen einem normalen und einem digitalem Spiel. Ist zum Beispiel ein Brettspiel mit einem kleinen Computer als Bestandteil des Spiels schon ein Computerspiel? Oder wie schaut es aus, wenn ein physischer elektronischer Würfel verwendet wird, der seine Ergebnisse per Bluetooth an den Computer übermittelt (siehe [Don12])? Herkömmliche Spiele und Computerspiele verschmelzen mehr und mehr. Es gibt jedoch laut Katie Salen und

2. Spiele für Medienfassaden

Eric Zimmerman vier Merkmale, die in Computerspielen oft stärker ausgeprägt sind und somit bei einer Abgrenzung helfen können [SZ04, S. 87-89]:

Unmittelbare aber begrenzte Interaktion: Ein großer Vorteil der digitalen Technik ist, dass ein unmittelbares und interaktives Feedback gegeben werden kann. Das Spiel kann direkt und dynamisch auf die Entscheidungen des Spielers entscheiden.

Informations-Handhabung: Digitale Spiele können Daten speichern und verarbeiten, wovon normalerweise auch im großen Umfang Gebrauch gemacht wird. Dabei können verschieden Daten, wie Text-, Video-, Bilder-, Audio- oder auch 3D-Daten verwendet werden.

Automatisierte komplexe Systeme: Es ist ein weiteres prägendes Merkmal von digitalen Spielen, dass sie die Möglichkeit bieten, komplexe Abläufe zu berechnen, die ohne Nutzung eines Computers nicht ohne größeren Aufwand berechnet werden könnten. In den meisten nicht-digitalen Spielen liegt es an den Spielern das Spiel auf Basis der definierten Regeln schrittweise vorwärts zu führen. In einem digitalen Spiel kann dies vom Computer berechnet und somit übernommen werden.

Kommunikation über ein Netzwerk: Eine Kommunikation der Teilnehmer über ein Netzwerk, ist das letzte Merkmal von digitalen Spielen. Viele, jedoch nicht alle digitalen Spiele, weisen dieses Merkmal auf.

2.2. Spielgenres

Inzwischen sind eine Vielzahl von verschiedenen Arten von Spielen entwickelt worden. Die Autoren Andrew Rollings und Ernest Adams geben in ihrem Buch „Game Design“ einen guten Überblick über die geläufigsten Spielgenres [RA03, S. 287 ff.]:

Actionspiele: Actionspiele zählen zu dem ältesten Genre, wobei es inzwischen eine Menge unterschiedlicher Stile gibt. Eines haben sie allerdings alle gemeinsam: Es geht darum beim Spieler die Reaktionszeit und die Hand-Auge-Koordination zu testen. Actionspiele weisen dafür meistens eine geringere Komplexität auf, da das Gehirn nur eine begrenzte Anzahl an Information pro Zeiteinheit verarbeiten kann. Es kann generell gesagt werden: Je schneller ein Spiel ist, desto geringer sollte seine Komplexität sein.

Strategiespiele: Strategiespiele haben ihren Ursprung in den Brettspielen. Allerdings haben die digitalen Versionen einen essentiellen Vorteil: Der Computer kann die

2. Spiele für Medienfassaden

Berechnung der komplexen Regeln übernehmen. Beim diesem Genre wird zwischen dem rundenbasierten Strategiespielen und den Echtzeitstrategiespielen unterschieden. Besonders die rundenbasierten Varianten haben im Vergleich zu anderen Spielgenres recht lange Spielzeiten.

Rollenspiele: Rollenspiele haben ihren Ursprung in den Pen-&-Paper-Rollenspielen. Die beiden wichtigsten Merkmale sind, dass der Spielcharakter konfigurierbar ist und an Erfahrung gewinnt, sowie eine langer Handlungsstrang. Häufig benötigen diese Spiele eine längere Eingewöhnungszeit bis der Spieler die komplette Komplexität des Spiels erfassen kann.

Sportspiele: Sportspiele sind direkte Umsetzungen der Sportveranstaltungen der Realität und decken die unterschiedlichsten Sportarten ab. Im Gegensatz zu vielen anderen Genres gibt es bei Sportspielen meistens keine unbekannte virtuelle Welt, sondern es wird die für den Spieler bereits bekannte Welt simuliert. Dies ergibt für den Spieldesigner die Situation, dass der Spieler genau weiß, wie zum Beispiel realer Fußball auszusehen hat, und somit das Spiel möglichst nahe an diese Realität kommen muss. Es gibt bei den Sportspielen allerdings auch oft Spiele, die mit einer bewussten Vereinfachung arbeiten. Meistens ist das dann eher Sammlung von Kurzspielen, die sich inhaltlich an Olympia oder anderen Sportveranstaltungen orientiert.

Fahrzeugsimulation: Bei Fahrzeugsimulationen wird versucht das Gefühl zu erzeugen, der Spieler würde ein reales Fahrzeug fahren oder fliegen. Dabei ist das Spektrum der Fahrzeuge sehr groß und kann vom Rennauto, über eine Straßenbahn oder einem Schiff bis zum Flugzeug reichen. Ähnlich wie bei Sportspielen wird auch bei diesem Genre versucht die Realität möglichst exakt abzubilden.

Aufbau- und Managementsimulationen: Bei Aufbau- und Managementsimulationen geht es um den Fortschritt, denn der Spieler baut etwas in einem fortschreitendem Prozess. Je besser der Spieler die Prozesse versteht, desto erfolgreicher wird sein Bauvorhaben sein. Diese Art von Spielen weist mit über die höchste Komplexität auf und hat dementsprechend häufig auch eine lange Spielzeit.

Abenteuerspiele: Abenteuerspiele sind eine interaktive Geschichte über einen Charakter, der durch den Spieler kontrolliert wird. Das Ziel ist es nicht, etwas aufzubauen oder zu besiegen, sondern den Charakter erfolgreich durch die Geschichte zu begleiten. Typische Merkmale sind das Erkunden, Sammeln oder Manipulieren von

2. Spiele für Medienfassaden

Objekten, Lösen von Puzzeln und vergleichsweise wenig Action- oder Kampfelemente.

Künstliches Leben: Das Genre Künstliches Leben beinhaltet die Modellierung biologischer Prozess, oft um den Lebenszyklus von lebenden Dingen zu simulieren. In diese Kategorie zählen Spiele in denen zum Beispiel ein Haustier gehalten wird oder die Verwaltung der Population einer Kreatur über die Zeit.

Puzzlespiele: Die meisten Computerspiele beinhalten in irgendeiner Form Puzzles. Abgrenzend dazu sind Puzzlespiele solche Spiele, bei denen nicht nur nebenbei kleinere Puzzle enthalten sind, sondern bei denen es primär um das Lösen von Puzzeln geht.

MMORPGS: Eines der jüngeren Genres bei Computerspielen sind die MMORPGs (massively multi-player online roleplaying games). Diese zeichnen sich besonders durch eine persistente Welt aus, an der Tausende von Spielern gleichzeitig teilnehmen. Die Spiele sind häufig äußerst komplex und haben für den Spieler eine sehr lange Spielzeit.

Bei der Betrachtung der verschiedenen Spielgenres wird schnell klar, dass einige sich besser als andere für die Umsetzung an einer Medienfassade eignen. Im Unterabschnitt 2.5 wird dieser Punkt näher untersucht.

2.3. Besonderheiten der Mehrspielerinteraktion

Besonders interessant sind Spiele mit einem Mehrspielermodus, bei dem mehrere Spieler gegen- oder miteinander spielen. Dabei wird unterschieden, ob am gleichen Gerät oder auf mehrere verteilt, die über ein Netzwerk miteinander kommunizieren, gespielt wird. Der Mehrspielermodus ist inzwischen ein häufiges Qualitätskriterium für ein Spiel und erfreut sich einer großen Beliebtheit unter den Spielern. Einerseits der gegenseitige Vergleich der Spielerfertigkeiten, andererseits das gemeinsame Spielerlebnis, wirken einen großen Reiz auf viele Menschen aus.

Im Gegensatz zum Einzelspielermodus, gibt es beim Mehrspielermodus neue Aspekte bei der Gestaltung eines Spiels zu beachten:

2.3.1. Darstellung der Inhalte für mehrere Spieler

Sobald mehrere Personen gleichzeitig an einem Gerät aktiv werden, stellt sich die Frage, inwiefern der Bildschirm die Inhalte für welchen Spieler darstellt.

2. Spiele für Medienfassaden

Eine Möglichkeit ist der sogenannte *Splitscreen*. Bei diesem wird, wie in Abbildung 1 dargestellt, der Bildschirm je nach Spieleranzahl in 2 oder 4 Bereiche aufgeteilt. Dadurch erhält jeder Nutzer einen Bereich des Bildschirms, der nur für ihn ist. Dies hat den Nachteil, dass durch die geringere Größe der Bereiche, weniger Informationen dargestellt werden können und jeder Spieler immer sehen kann, was gerade die anderen Spieler machen. Des Weiteren müssen die Nutzer zum Start des Spiels sich oft erst kurz orientieren, welcher Bereich denn nun ihrer ist. Der Vorteil ist dagegen, dass die Spielbereiche gleichzeitig für alle Spieler dargestellt werden und völlig unabhängig voneinander Inhalte darstellen können.

Des Weiteren gibt es den sogenannten *Hot-Seat-Modus*. Hierbei wechseln die Spieler sich ab und wechseln sozusagen den Stuhl vor dem Gerät. Somit hat jeder Spieler wieder den kompletten Bildschirm zur Verfügung, wodurch mehr und komplexere Informationen dargestellt werden können. Nachteilig ist hingegen, dass der andere Spieler zu diesem Zeitpunkt nichts anderes übrig bleibt, als abzuwarten bis die gerade aktive Person seinen Spielzug beendet hat. Durch die bessere Vernetzung wurde der Modus allerdings immer mehr durch den Netzwerk-Modus (siehe Unterunterabschnitt 2.3.3) verdrängt. Besonders bei rundenbasierten Strategiespielen ist dieser Modus manchmal jedoch noch zu finden.

Bei weniger umfangreichen Spielen mit einer überschaubaren Spielfläche und Komplexität, werden die Spieler teilweise auf einem Bildschirm gleichzeitig ohne Aufteilung des Bildschirms dargestellt. Dabei gibt es entweder ein Spielfeld das vom Bildschirm fest angezeigt wird, auf dem sich alle Spieler bewegen, oder es gibt ein sich bewegendes Spielfeld, von dem immer ein in der Größe gleicher Bestandteil dargestellt wird. Die Bewegung des Spielfelds ist dabei entweder in einer festen Geschwindigkeit oder von der Bewegung des Spielers abhängig. Diese Methode ist recht simpel für die Implementierung eines Mehrspielermodus, eignet sich aber nur für einige einfache Arten von Spielen, wie zum Beispiel die Umsetzung von Brettspielen.

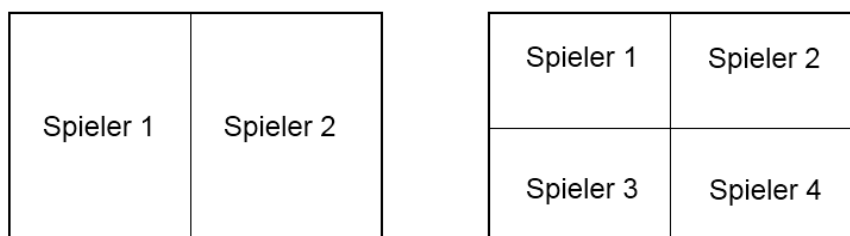


Abbildung 1: Aufteilung des Bildschirms im Splitscreen-Modus

2. Spiele für Medienfassaden

2.3.2. Mehrspieler-Steuerung

Beim Einzelspielermodus ist stets klar, zu wem eine Spielfigur oder ein Zeiger eines Eingabegeräts gehört. Sobald jedoch mehrere Spieler einen Bildschirm in einem Mehrspielermodus gleichzeitig nutzen, ist die Zuteilung nicht mehr so einfach. Damit es keine Orientierungsprobleme gibt, muss jedem Spieler stets klar sein, welches seine Spielfigur ist. Meistens wird dies durch eine farbliche Zuordnung oder durch die Verteilung optisch verschiedener Charaktere erreicht.

Bei Zeigern wird dies allerdings noch etwas unübersichtlicher. Eine alleinige farbliche Hervorhebung könnte unzureichend zu unterscheiden sein und sich als große Hürde für Leute mit Farberkennungsdefizite herausstellen. Eine neuere Lösung für diese Problematik ist die Nutzung von mobilen Geräten und Video See-Through. Hierbei wird das eine primäre Display durch Augmented Reality auf den mobilen Geräten, für jeden einzelnen Nutzer um zusätzliche Informationen und Eingabemöglichkeiten erweitert. Details dazu sind in der Arbeit „Multi-User Interaction on Media Facades through Live Video on Mobile Devices“ zu finden [BGW⁺].

2.3.3. Kommunikation zwischen mehreren Geräten

Eine immer häufiger verbreitete Variante ist, dass jeder Spieler sein eigenes Gerät für den Mehrspielermodus verwendet, welches den Vorteil hat, dass die eben beschriebenen Probleme mit der Bildschirmaufteilung und der Steuerung entfallen. Hierbei müssen die Geräte über ein Netzwerk kommunizieren und den Spielstatus auf allen Systemen synchron halten. Je nach Umfang des Spiels, kann dies einen erheblichen Mehraufwand für die Implementierung bedeuten. Insbesondere bei Actionspielen, bei denen eine gute Reaktion wichtig ist, darf die Verzögerung durch die Kommunikation über das Netzwerk nicht spürbar sein. Schon eine Verzögerung von 100 *ms* kann dazu führen, dass ein Spieler auf dem Gerät an einer Position dargestellt wird, auf der er sich eigentlich nicht mehr befindet. Des Weiteren werden Features, wie das Speichern des Spielstandes oder das Pausieren des Spiels, deutlich komplexer.

2.4. Besonderheiten bei der Visualisierung an einer Medienfassade

Eine Medienfassade kann über viele unterschiedliche Technologien realisiert werden, wobei besonders häufig LED-basierte Lösungen verwendet werden. Medienfassaden unterscheiden sich in vielen Punkten vom herkömmlichen Display. Einige der Technologien sind nur im Dunkeln nutzbar, andere haben einen sehr großen Pixelabstand. Des Wei-

2. Spiele für Medienfassaden

teren sollte berücksichtigt werden, dass der Nutzer in einer gewissen Entfernung zum Display steht. Eine detaillierte Übersicht über die verschiedenen Technologien für Medienfassaden und der Funktionsweise sind in der Arbeit „Medienfassade - Möglichkeiten von Visualisierung und Interaktion“ von Stephan Bergemann und Robin Schlegel zu finden [BS12, S. 3 ff.]. Besonders interessant sind Medienfassaden, bei denen mehrere Technologien kombiniert werden. Darüber lassen sich neue Ansätze verfolgen und neue Ideen umsetzen.

Im Falle der Medienfassade des FKI-Gebäudes wird eine Kombination der Technologien LED-Kacheln, Rückprojektion und Ambient-Light genutzt. Im Unterabschnitt 3.4 ist der exakte Aufbau der Fassade dargestellt. Alle drei Technologien haben dabei ihren eigenen Vor- und Nachteile. Die LED-Kacheln kommen einem herkömmlichen Bildschirm vom Aufbau her am nächsten. Damit können zentrale Elemente eines Spiels oder Zusatzinformationen dargestellt werden. Die Rückprojektion bietet die höchste Auflösung, ist allerdings vom Aufbau her etwas verteilt und nur bei Nacht einsatzbereit. Sie kann für visuelle Gimmicks oder auch für Zusatzinformationen für ein Spiel verwendet werden. Ebenso ist ein Spiel über die verschiedenen Rückprojektionsflächen verteilt vorstellbar. Entweder die einzelnen Flächen werden ähnlich wie in einem Splitscreen-Modus auf die Spieler verteilt oder eine zusammenhängende Spielfläche wird dargestellt. Das Ambient-Light kann für zusätzliche atmosphärische Unterstützung sorgen. Beim Auftreten des Endgegners könnte zum Beispiel eine dunklere-rötlichere Farbe für eine bedrohliche Situation verwendet werden.

2.5. Klassifikation von Spielen für eine Medienfassade

Aufbauend auf den vorigen Abschnitten kann geschlussfolgert werden, dass einige Arten von Spielen für Medienfassaden besser geeignet sind, als andere Arten. Im Folgenden sollen dabei einige Merkmale hervorgehoben werden, die ein Spiel gerade für die Nutzung an einer Medienfassade qualifizieren:

Spieldauer: Häufig steht der Nutzer vor der Fassade und verweilt auch nicht lange vor ihr. Dementsprechend sind Spiele mit kurzer Spieldauer empfehlenswert. Dadurch kann eine höhere Anzahl von Spielern pro Zeiteinheit erreicht werden und verhindert werden, dass wenige Personen die Fassade lange blockieren.

Spielkomplexität: Das Spiel muss schnell erlernbar sein und eine möglichst geringe Komplexität aufweisen. Dabei sollten die Spielregeln einfach und leicht verständlich sein. Der Spieler ist vermutlich vor einer Medienfassade nicht bereit sich erst

2. Spiele für Medienfassaden

länger in ein Spiel einzuarbeiten. Außerdem würde dies nur unnötig die Spieldauer verlängern, die möglichst gering sein sollte.

Zielgruppe: Sobald ein Spiel auf einer Medienfassade im öffentlichen Raum spielbar ist, sollte auch eine möglichst große Menge an Leuten von ihr profitieren. Die Zielgruppe sollte daher so konzipiert sein, dass sie viele verschiedenen Menschen, die potenziell an dieser Medienfassade vorbeigehen, auch anspricht. Dabei ist natürlich auch die Lage der Medienfassade zu berücksichtigen.

Steuerung: Die Steuerung des Spiels sollte vorzugsweise leicht zu erlernen und dementsprechend intuitiv sein. Der Nutzer sollte ohne große Einweisung mit der Steuerung schnell zurechtkommen.

Spieleranzahl: Eine Medienfassade soll eine möglichst große Anzahl von Menschen erreichen. Daher ist es hilfreich, wenn das Spiel einen Multiplayermodus hat. Gerade vor einer Medienfassade, wo auch eine größere Anzahl von Zuschauern präsent ist, ist es spannend, wenn das Spiel einen konkurrierenden oder kooperativen Spielanteil enthält.

Flexibilität: Es ist von Vorteil, wenn das Spiel einfache Erweiterungsmöglichkeiten enthält, wie zum Beispiel neue Level oder Spielmodi. Besonders ansprechend wäre, wenn dies durch die Spieler erstellt werden kann, indem zum Beispiel ein Leveleditor zur Verfügung gestellt wird. Damit erhält die Medienfassade einen partizipativen Anteil. Dies kann dafür sorgen, dass auch über einen längeren Zeitraum die Begeisterung für das Spiel erhalten bleibt.

Jugendschutz: Eine Medienfassade ist normalerweise frei zugänglich und auch aus größerer Entfernung gut einsehbar. Eine Art von Alterskontrolle der Nutzer und Zuschauer ist schwer zu realisieren. Folglich dürfen nur Spiele ohne Altersbeschränkung auf Medienfassaden dargestellt werden.

Bei der Berücksichtigung dieser Merkmale ergibt sich, dass besonders kleinere *Actionspiele*, vereinfachte *Sportspiele* und *Puzzlespiele* gut für eine Medienfassade geeignet sind. Sogenannte *Casual Games*, welche sich ebenfalls durch viele der erarbeiteten Merkmale auszeichnen, sind ebenso gut geeignet.

3. Technische Umsetzung

Für die Demonstration eines Spiels an einer Medienfassade wurde prototypisch ein Computerspiel entwickelt. Im Folgenden werden typische Probleme bei der Umsetzung für eine Medienfassade und mögliche Lösungswege erörtert.

3.1. Auswahl des Spiels

Aufbauend auf der Klassifikation in Unterabschnitt 2.5 wurde festgelegt, dass eine Klon des Spiels *Bombberman* umgesetzt werden soll. Bombberman erfüllt sehr gut die Anforderungen an ein Spiel für eine Medienfassade:

- Es hat einen Mehrspielermodus für vier Spieler, wobei es auch Varianten mit bis zu acht Spielern gibt.
- Das Spiel ist sehr leicht zu erlernen und hat eine geringe Komplexität.
- Es hat eine simple Steuerung.
- Es ist auch für jüngere Nutzer geeignet.
- Das Spiel hat einen hohen Bekanntheitsgrad, wodurch die Regeln für viele potenzielle Spieler bereits ohne Einweisung klar sind.
- Es ist leicht durch neue Level, Powerups oder Spielmodi erweiterbar.
- Die Spielrunden sind kurz und zeitlich begrenzt.

3.2. Anforderungen

Für die prototypische Umsetzung des Spiels sollen die nachstehenden Anforderungen auf jeden Fall erfüllt werden:

- Bis zu vier Spieler.
- Begrenzung der Spielzeit pro Runde und Anzeige der Restzeit.
- Steuerung der Spielfigur über ein Smartphone.
- Basisfunktionalität von Bombberman: Spielfigur bewegen, Bewegung durch Hindernisse begrenzen, Bombe platzieren, Steine zerstören, Spieler ausschalten, Kettenreaktion bei Explosionen, Explosion durch Hindernisse begrenzen.
- Spielfeld dynamisch aus Datei laden.
- Korrekte Kollisionserkennung.
- Client-Server-Architektur für die Umsetzung des Mehrspielermodus.
- Konfigurierbarkeit der Spielparameter.

3. Technische Umsetzung

Damit nach Abschluss des Projektes der Quelltext veröffentlicht werden kann, ist es notwendig quelloffene Frameworks zu nutzen. Durch die Veröffentlichung kann eine Erweiterung des Spiels und eine Nutzung als Vorlage für weitere Projekte an der Medienfassade ermöglicht werden. Dementsprechend sollten ebenso kostenlose Bibliotheken bevorzugt werden. Da die endgültige Infrastruktur der Medienfassade noch nicht feststeht, wäre es vorteilhaft, wenn das Spiel auf möglichst vielen gängigen Plattformen laufen würde.

3.3. Benötigte Infrastruktur

Für die Umsetzung des Projekts wird eine gewisse Infrastruktur benötigt. Diese wird auf der voraussichtlich vorhandenen Infrastruktur der Medienfassade aufbauen. In der Abbildung 2 ist die für die Anwendung relevante Infrastruktur visualisiert. Als Clients werden Smartphones verwendet, die mit dem Internet verbunden sind. Im internen Netzwerk, welches in der Abbildung gestrichelt umrandet ist, befindet sich ein Server. Dieser kommuniziert über DVI mit dem Controller der LED-Wand, welcher sein Signal über DMX an die einzelnen LED-Kacheln weiterleitet.

Da das Programm in der ersten Version nur die LED-Fläche verwendet, bleibt die Ansteuerung der Komponente überschaubar. Bei einer Nutzung der weiteren Technologien muss die Infrastruktur dementsprechend erweitert werden.

3.3.1. Client: Eingabe

Für die Interaktion auf Client-Seite wird ein Smartphone verwendet, da dieses sehr gut die Umsetzung der Anforderungen ermöglicht und inzwischen zunehmend verbreitet ist. Im 1. Quartal 2012 war laut einer Studie der Bitkom das am weitesten verbreitete Betriebssystem für Smartphones *Android OS* [Pak12]. Da eine möglichst große Menge an potenziellen Nutzern erreicht werden soll, wird der Client auch für Android-OS umgesetzt.

Interessant ist hierbei noch der Aspekt, dass es inzwischen mehrere Lösungen für Cross-Plattform-Entwicklung für Smartphones gibt. Der Markt an Betriebssystemen für Smartphones ist zum Beispiel mit iOS, Android OS, Symbian OS, Blackberry OS, Windows Mobile, webOS und weiteren inzwischen ziemlich groß. Es ist sehr reizvoll, nur eine Anwendung zu entwickeln, die auf allen Systemen läuft. Es gibt zurzeit vier verschiedene Ansätze, wie eine Cross-Plattform Anwendung zu realisieren ist [Beh10]:

Mobile Web Apps: Web Apps werden nicht für eine Plattform, sondern für den Browser entwickelt. HTML5, CSS und Javascript ermöglichen ein großes Spektrum für

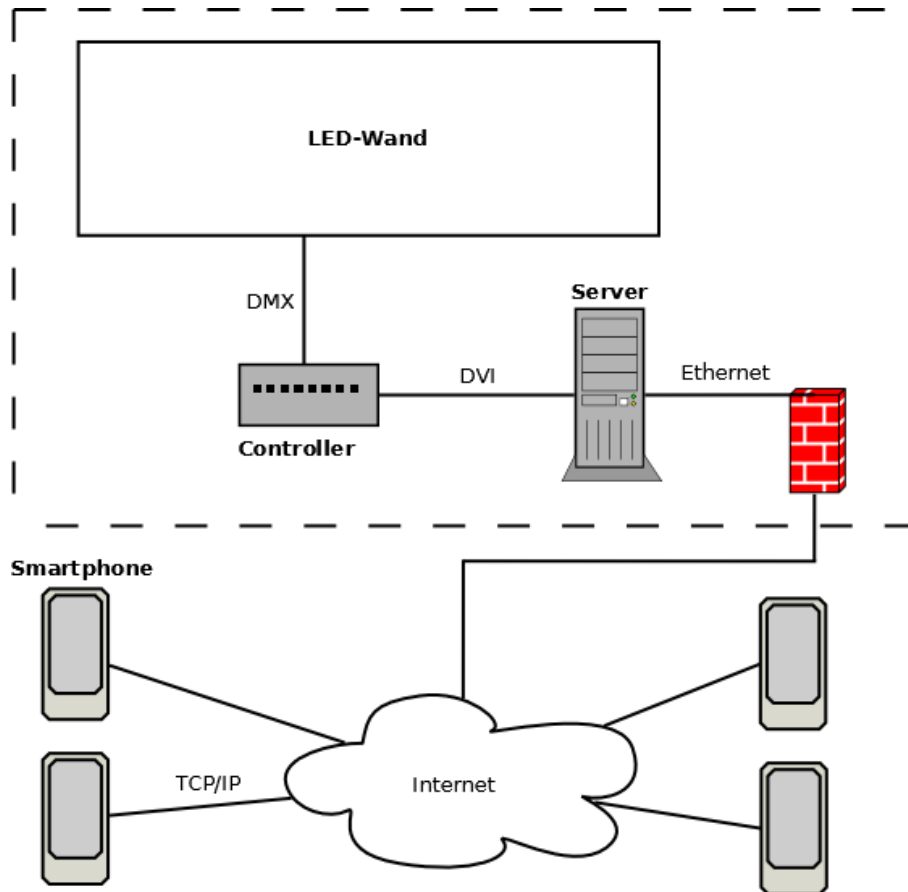


Abbildung 2: Benötigte Infrastruktur für ein Spiel mit Client-Server Architektur an der Medienfasade

die Entwicklung der Anwendung. Web Apps laufen dadurch auf verschiedenen Plattformen, haben jedoch den großen Nachteil, dass sie bei der Ansteuerung der Hardware des Geräts limitiert sind und auch nicht das Gefühl einer echten nativen App aufkommt. Des Weiteren sind sie in der Geschwindigkeit nicht mit nativen Apps vergleichbar.

Hybride Apps: Bei hybriden Apps wird ein kleiner Teil des Programms auf dem Smartphone installiert. Dieser übernimmt die Rolle des Browsers, wodurch eine bessere Unterstützung für die Ansteuerung der Hardware möglich ist. Eventuelle browser-spezifische Probleme können damit auch umgangen werden. Es bleibt von Nachteil, dass weiterhin nicht das Gefühl einer nativen App gegeben ist und der Entwickler an die Paradigmen und die Weiterentwicklung des Herstellers des Frameworks gebunden ist. Ein Beispiel für die Entwicklung von hybriden Apps ist das Framework *PhoneGap*.

Interpretierte Apps: Interpretierte Apps können die plattformspezifischen Elemente der GUI nutzen, wodurch das Gefühl von nativen Apps entsteht. Dabei wird eine gemeinsame Codebasis verwendet, die dann in eine interpretierbare Zwischensprache übersetzt wird. Auch bei dieser Art von Apps hat der Entwickler den Nachteil, dass er an die Weiterentwicklung und Paradigmen des Frameworks gebunden ist. Dies kann gerade im mobilen Bereich ein großer Nachteil sein, da sich die Technologien in diesem rasant weiterentwickeln. Bekannte Beispiele für diese Frameworks sind *Rhodes* (Ruby), *Monotouch* (C# und .NET) oder *Appcelerator* (Javascript).

Generierte Apps: Die letzte Möglichkeit sind generierte Apps. Hierbei werden native Apps aus einer gemeinsamen Codebasis generiert. Dies erfordert jedoch ein komplexes Framework, welches für jede Plattform einzeln aufwändig implementiert werden muss. Der aktuelle Stand ist, dass solche Frameworks erst noch in der Entwicklung und noch nicht für den produktiven Einsatz geeignet sind. Ein Beispiel hierfür ist *XMLVM*, das zurzeit iOS und Android OS unterstützt. Bei XMLVM wird nicht der Quelltext cross compiled, sondern der Bytecode, wobei Bytecode von Java, .NET und Ruby unterstützt wird.

Mobile Apps sind aufgrund der eingeschränkten Möglichkeiten des Zugriffs auf die Hardware für die Umsetzung von Spielen oder Spieleclients eher ungeeignet. Eine gute Lösung könnte es sein, ein Framework zu verwenden, welches interpretierte Apps zur Verfügung stellt. Eine ausführliche Evaluation der verschiedenen Frameworks soll im Rahmen dieser Arbeit allerdings nicht erfolgen, weshalb für den Prototyp auf eine native App gesetzt wurde. Für weitere Entwicklungen sollte die Nutzung dieser jedoch in Betracht gezogen werden, da damit eine größere Anzahl von Nutzer erreicht werden kann.

3.3.2. Server: Ausgabe

Der Server wird entweder ein Apple Mac Mini oder ein Desktop-PC mit einer Linux-Distribution sein. Beides ist aufgrund der Cross-Plattform-Entwicklung mit SDL (siehe Unterunterabschnitt 3.5.2) möglich, wobei theoretisch auch ein Windows-Betriebssystem ohne Probleme genutzt werden könnte. Für die Entwicklung des Prototyps wurde die Server-Komponente auf einem Mac Mini der 5. Generation mit Mac OS X 10.7, sowie auf einem Ubuntu 12.04 getestet.

Auf dem Server läuft der Hauptteil der grafischen Benutzeroberfläche, welche dann auf der LED-Wand ausgegeben wird. Es wird zum Beispiel das Spielfeld, die Spielzeit oder der aktuelle Status des Spiels dargestellt. Eine detaillierte Beschreibung der Oberfläche

3. Technische Umsetzung

des Servers ist im Unterunterabschnitt 3.5.4 zu finden.

Für den Prototyp wird nur die LED-Wand verwendet, aber auch die anderen Technologien der geplanten Medienfassade des FKIs könnten für das Spiel genutzt werden. Vorstellbar wäre zum Beispiel, dass das Ambient-Light verwendet wird, um die Restzeit der aktuellen Spielrunde zu verdeutlichen. Läuft die Zeit ab, könnte mit Hilfe des Ambient-Lights dies farblich durch den langsamen Wechsel in einen anderen Farbton hervorgehoben werden. Die Rückprojektion kann zwar nur bei Nacht zum Einsatz kommen, aber dennoch wäre es möglich sie mit einzubeziehen. Zum Beispiel könnte dort die aktuelle Highscore-Liste dargestellt werden oder zusätzliche Hilfestellungen für einen leichteren Einstieg in das Spiel angezeigt werden.

3.4. Aufbau der Medienfassade des FKI

Die aktuelle Planung der Medienfassade sieht eine Aufteilung in drei Technologien vor: LEDs, Rückprojektion und Ambient-Light. Wie in der Abbildung 3 rot dargestellt, wird es einen zentralen Bereich mit LED-Kacheln geben. Dieser wird aus 12 Reihen mit jeweils 36 Modulen bestehen. Hierbei kommen Grid 25 Module der Firma Freitag Technologies zum Einsatz, welche jeweils eine Auflösung von $8 * 8$ Pixeln haben [Tec]. Eine Reihe hat dementsprechend $36 * 12 = 288$ Pixel. In der Höhe sind es $8 * 12 = 96$ Pixel. Daraus ergibt sich die Gesamtauflösung von **288 * 96** Pixel, bzw. ein Bildformat von **3 : 1**. Vorteilhaft an den LEDs ist, dass sie über ein großes Farbspektrum verfügen und sehr hell sind.

Des Weiteren werden die Fenster des Gebäudes voraussichtlich mit Rückprojektion über Beamer bespielt werden. Hierbei werden vier HD-Beamer pro Etage zum Einsatz kommen, die die komplette Fensterfläche beleuchten. Zusätzlich wird die Rückprojektion noch auf die ersten Fenster um die Ecke mit jeweils einem weiteren Beamer erweitert werden. Dies bietet die Möglichkeit Spielelemente auf getrennten Häuserseiten darzustellen und somit für verschiedene Spieler nicht einsehbar zu machen. Pro Gebäudeseite könnte ein Spieler sein Spielfeld haben, ohne dass der andere Spieler sehen kann, was der jeweils andere Spieler macht. Das ergibt neue Spielmöglichkeiten, die nicht möglich wären, wenn die Fassade nur auf einer Seite des Gebäudes wäre. Ein Beispielszenario wäre *Schiffe versenken*, bei dem es essentiell ist, dass die beiden Spieler nicht das Spielfeld des anderen einsehen können. Die Rückprojektion wird jedoch nur bei Nacht zum Einsatz kommen, weil tagsüber die Räumlichkeiten als Büroräume genutzt werden.

Als dritte Komponenten kommt Ambient-Light zum Einsatz. Die komplette Fassade wird mit Hilfe des Ambient-Light in verschiedenen Farben oder Farbkombination be-

3. Technische Umsetzung

leuchtbar sein. Damit lässt sich in Spielen beispielsweise eine bestimmte Atmosphäre verstärken.

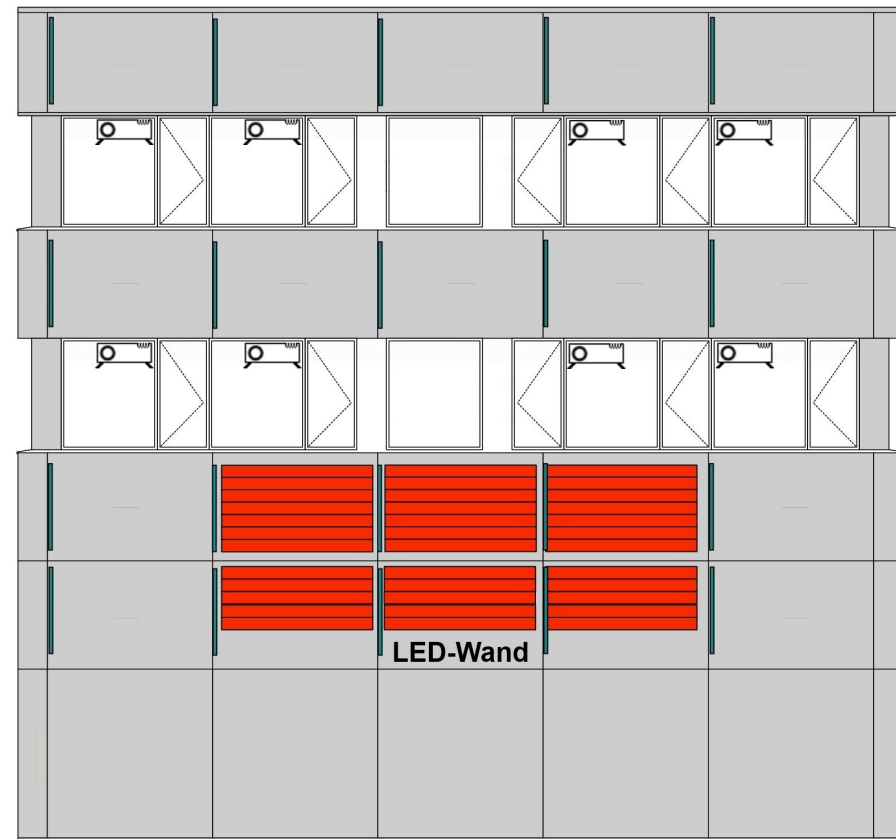


Abbildung 3: Aufbau der Medienfassade des FKI-Gebäudes

3.4.1. Testlabor für die Medienfassade

Zum Entwickeln gibt es ein Testlabor in dem eine Rückprojektion aufgebaut ist. Zusätzlich stehen zwei Reihen mit jeweils 9 LED-Kacheln inklusive Controller zur Verfügung, die eine Auflösung von **72 * 16** Pixel ermöglichen. Der Aufbau ist in Abbildung 4 dargestellt. Mit Hilfe des Testlabors kann ohne die eigentliche Medienfassade eine Anwendung für diese entwickelt werden.

Hierbei hat das Testlabor besonders bei der Gestaltung der Oberfläche eine wichtige Rolle gespielt. Mit Hilfe des Labors konnte der Eindruck des Spiels auf den LED-Kacheln getestet werden. Die Maximale Entfernung zu den LED-Kacheln lag allerdings räumlich bedingt bei nur $4,5\text{ m}$. Der visuelle Eindruck war daher noch nicht unbedingt so, wie er an der fertigen Fassade sein wird. Bei dieser wird Betrachter voraussichtlich weiter

3. Technische Umsetzung

entfernt stehen. Aufgrund des hohen Pixelabstands kommen bei 4,5 m auch viele Details optisch noch nicht wie auf einem Monitor zum Vorschein, auf dem die Pixeldichte sehr viel höher als auf den LED-Kacheln ist. Als mögliche Lösung für dieses Problem war es hilfreich, von den LED-Kacheln im Labor Fotos zu machen. Diese konnten dann auf dem Bildschirm skaliert werden, wodurch die Pixel-Abstände zwischen den LEDs wieder kleiner wirkten, genauso als ob der Betrachter weiter von der Fassade entfernt stehen würde. Hierdurch war es möglich, zumindest einen ungefähren Eindruck zu bekommen, wie die Oberfläche auf der Medienfassade wirken wird.



Abbildung 4: Aufbau des Testlabors mit Projektionsleinwand und LED-Kacheln

3.5. Entwurf des Prototyps

Im Folgenden werden einige der wichtigsten Phasen im Entwurf des Spiels genauer beschrieben. Dabei wird das Spiel näher beschrieben, die genutzten Technologien erläutert und auf mögliche Probleme sowie Lösungsansätze dazu verwiesen.

3.5.1. Authentifizierung

Bei einem netzwerkbasierendem Spiel ist es häufig von Vorteil, wenn der Benutzer sich in irgendeiner Art *authentifizieren* kann. Einerseits kann damit eine Highscore erstellt oder benutzerspezifische Einstellungen geladen werden, andererseits können aber auch unerwünschte Spieler blockiert werden, die zum Beispiel durch versuchtes Schummeln negativ

3. Technische Umsetzung

aufgefallen sind. Dabei könnte jeder Spieler einen eindeutigen Spielernamen wählen, welches auch die Identifizierung von Informationen für diesen Spieler auf dem Bildschirm erleichtert. Anstelle von „Spieler 1 gewinnt“ ist ein „Max Mustermann gewinnt“ für die Beteiligten leichter zuzuordnen. Des Weiteren würde eine Farbwahl für die Spielfigur es dem Spieler leichter machen, seine Figur auf dem Spielfeld zu erkennen.

Von Nachteil ist dabei, dass der Benutzer sich erst registrieren muss, was eine zusätzliche Hürde für das Spielen an der Fassade darstellt. Eine Lösungsmöglichkeit wäre die Nutzung einer Gast-Option, wodurch Spieler auch ohne Registrierung spielen können, dann aber nicht in die Highscore aufgenommen werden oder ihre Spielfigur individuell gestalten können.

Für den Prototyp wird auf eine Authentifizierung des Benutzers des Umfangs wegen verzichtet. In späteren Versionen des Programms wäre es jedoch sinnvoll ein solches Feature noch zu realisieren.

3.5.2. Simple Directmedia Layer

Simple Directmedia Layer (SDL) ist eine plattformunabhängige Multimedia-Bibliothek. Sie wurde dafür entwickelt, einen einfachen Zugriff auf das Soundsystem, die Tastatur, die Maus, den Joystick, die 3D Hardware via OpenGL und den 2D Video-Framebuffer zu ermöglichen. Im Gegensatz zu DirectX werden verschiedene Plattformen unterstützt, wie zum Beispiel Windows, Linux, Mac OS X oder auch FreeBSD. Dabei ist es unter der GNU LGPL Version 2 lizenziert, wodurch eine kostenlose Nutzung der Bibliothek, sogar in kommerziellen Produkten, ermöglicht wird. [SDL]

Für die Entwicklung des Spiels wurde die SDL in der Version 1.2 verwendet. Gerade die Plattformunabhängigkeit, sowie die gute Unterstützung bei der Erstellung von 2D Spielen, spricht für die SDL. Die Anwendung kann somit sowohl auf Mac OS X, Linux, als auch Windows laufen, je nachdem, wie das System für die Medienfassade final aufgebaut sein wird.

Für die Umsetzung des Spiels wurden zwei weitere Bibliotheken als Erweiterung für die SDL genutzt:

SDL_net V. 1.2.8: SDL_net ist eine kleine plattformunabhängige Netzwerkbibliothek [SLM]. Sie abstrahiert die Netzwerkfunktionalitäten verschiedener Betriebssysteme und vereinfacht den Umgang mit dem Aufbau von Verbindungen, sowie dem Datentransfer. Zusätzlich gibt es eine einfache Lösung für die Verwaltung mehrerer Verbindungen gleichzeitig mit Hilfe der sogenannten *SocketSets*.

SDL_ttf V. 2.0.11: SDL_ttf ist eine auf freetype2 basierende Bibliothek, mit der auf einfache Art und Weise TrueType-Schriften gerendert werden können. [Lan]

3.5.3. Farbgebung

Ein wichtiger Punkt beim Entwurf des Spiels war die Farbgestaltung. Da die LED-Kacheln einen großen Pixelabstand im Gegensatz zu einem herkömmlichen Display aufweisen, ist die Erkennung zusammenhängender Elemente unterstützend über Farben besonders wichtig. Ebenso muss beachtet werden, dass dunkle Farben schwieriger zu sehen sind, da bei einem Schwarz die LEDs nicht aktiv sind. Helle Farben hingegen sind besonders gut sichtbar. Die farbliche Gestaltung des Spiels musste daher regelmäßig im Testlabor überprüft werden.

Es hat sich dabei gezeigt, dass es wichtig ist, Kontraste zu verwenden und besonders wichtige Elemente sollten in einer sehr hellen Farbe dargestellt werden. Dies bedeutet für den Bomberman-Klon, dass die Bombe weiß ist und die Explosion in einem hellen gelb-orange dargestellt wird. Diese beiden Komponenten sind besonders wichtig für den Spieler im Auge zu behalten. Da es ebenso wichtig ist seine Spielfigur zu erkennen, hat diese weiße Augen mit denen sie farblich hervorsteht.

Leider waren die Kontraste auf dem Monitor häufig anders, als die Kontraste mit den LED-Kacheln. Somit musste die Farbgestaltung häufiger nachkorrigiert werden, bis sich die einzelnen Elemente auf den LED-Kacheln gut sichtbar unterschieden.

3.5.4. Benutzeroberfläche Server

Da die Benutzeroberfläche des Servers auf den LED-Kacheln läuft, ist die Auflösung wie in Unterabschnitt 3.4 detailliert beschrieben, auf $288 * 96$ Pixel beschränkt. Dementsprechend ist die Bildfläche bei einem Seitenverhältnis von $3 : 1$ sehr breit. Bei dem für Bomberman typischen eher quadratischem Spielfeld ergibt sich eine noch recht große ungenutzte Fläche. Im Prototyp soll diese, wie in Abbildung 5 dargestellt, für die Anzeige von Statusinformationen genutzt werden. Ein deutlich breiteres Spielfeld macht bei vier Spielern keinen Sinn, weil dann das Ungleichgewicht der Entfernung zwischen den Spielern zu groß wäre. Der Spieler oben links wäre dem Spieler unten links sehr viel näher, als dem Spieler oben rechts. Dadurch entsteht eine große Fläche neben dem Spielfeld, welche sich nur ungenügend mit Statusinformationen füllen lässt. Hier wird zum Beispiel die aktuell noch übrige Spielzeit der Runde dargestellt. In einer Weiterentwicklung des Programms könnte die Spielerzahl auf sechs erhöht werden. Die weiteren Spieler würden oben in der Mitte und unten in der Mitte ihre Startposition erhalten.

3. Technische Umsetzung

Dann könnte das Spielfeld auch noch in der Breite erweitert werden, welches insgesamt eine bessere Nutzung der vorhandenen Fläche ergeben würde.

Das eigentliche Spielfeld wird dynamisch aus einer Datei geladen. In einer Textdatei ist für jedes Feld über eine Zahl definiert, von welcher Art dieses Feld ist. Eine **0** steht dabei für ein begehbares Feld, eine **1** für einen nicht zerstörbaren Stein und eine **2** für einen zerstörbaren Stein. Durch diese Art der Speicherung des Spielfelds, ist es sehr einfach, dieses anzupassen oder neue Spielfelder zu entwickeln. Ebenso wäre es unkompliziert hierfür einen Karteneditor zu entwickeln.

In der Abbildung 6 ist eine Darstellung eines Teils der Oberfläche des Servers im Testlabor zu sehen.

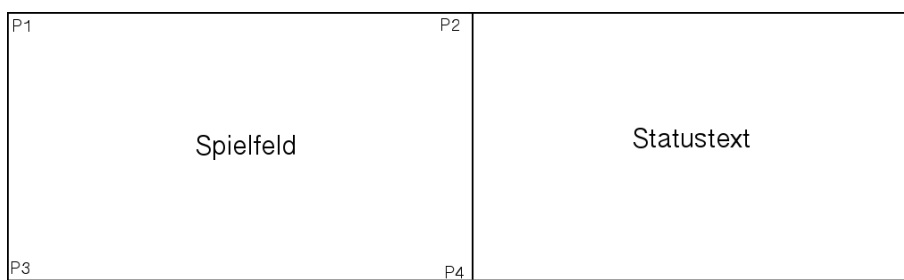


Abbildung 5: Entwurf der Benutzeroberfläche des Servers

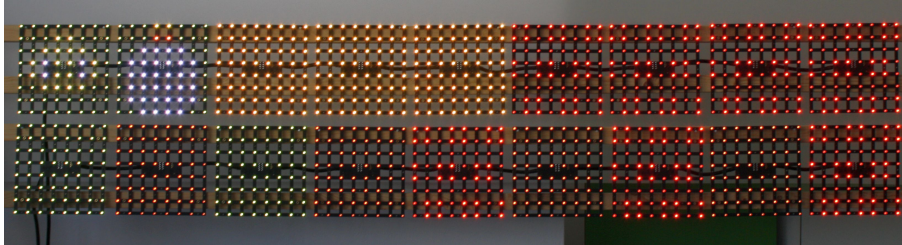


Abbildung 6: Benutzeroberfläche des Servers auf den LED-Kacheln im Testlabor

3.5.5. Benutzeroberfläche Client

Die Benutzeroberfläche des Clients ist einfach gehalten. Da der Client primär nur die Funktion der Steuerung übernimmt, sind aufwändigere Oberflächen nicht notwendig. Für die bessere Handhabung sind alle Dialoge im Querformat, wobei es insgesamt drei Dialoge gibt:

Startbildschirm: Der Startbildschirm in Abbildung 7 visualisiert, gibt dem Benutzer nur ein knappes Menü zur Auswahl. Er kann entweder das Spiel starten, die Ein-

3. Technische Umsetzung

stellungen öffnen oder die Anwendung wieder beenden. Wenn der Benutzer das Spiel starten möchte, erscheint der für Android typische Ladebildschirm, bis die Verbindung hergestellt werden konnte.

Einstellungen: In den Einstellungen kann der Nutzer den Port und die IP-Adresse des Servers konfigurieren.

Spielbildschirm: Dieser Dialog ist der wichtigste, da er die eigentliche Steuerung des Spiels enthält. Es gibt im Prototyp wie in Abbildung 8 dargestellt, vier Knöpfe für die Steuerung und einen Knopf für das Platzieren einer Bombe. Beim Verlassen des Bildschirms wird automatisch die Verbindung zum Server wieder getrennt.

3.5.6. Kommunikation zwischen Client und Server

Eine besondere Herausforderung bei der Realisierung des Spiels ist die Kommunikation zwischen dem Client auf dem Smartphone und dem Server im Gebäude mit der Medienfassade. Generell wird die Datenübertragung mit dem Smartphone zwar immer schneller, soll aber eine möglichst große Zielgruppe erreicht werden, muss bei der Entwicklung der Software von geringeren Geschwindigkeiten ausgegangen werden. Die Verbindung zwischen dem Client und dem Server ist eine TCP-Verbindung. Ein Befehl entspricht dabei einem Kommando für die Spielfigur sich in eine Richtung zu bewegen oder eine Bombe zu platzieren.

Je nach Verfügbarkeit hat der Nutzer eine Verbindung über GSM bis hin zu LTE. Im schlechtesten Fall hat er daher nur GSM zur Verfügung und damit bis zu $14,4 \frac{kBit}{s}$. Dementsprechend ist die Kommunikation so konzipiert, dass ein sehr geringes Datenaufkommen vorhanden ist. Ein Befehl vom Client zum Server hat stets nur 1 *Byte* Dateninhalt zuzüglich des TCP-Overheads, wobei der TCP-Head eines TCP-Pakets ungefähr 20 *Byte* benötigt.

Der Android-Simulator ermöglicht sowohl die Netzwerkgeschwindigkeit als auch die Netzwerklatenz für verschiedene Standards zu simulieren. So konnte der Datentransfer auch mit geringen Übertragungsgeschwindigkeiten und hohen Latenzen getestet werden. Der Client sendet ungefähr 10 Befehle pro Sekunde, wenn ein Knopf gedrückt gehalten wird. Dementsprechend ist das Datenaufkommen auch vergleichsweise gering, womit es auch bei langsamen Übertragungsgeschwindigkeiten keinerlei Probleme geben sollte.

Der genaue Ablauf der Kommunikation zwischen dem Client und dem Server ist in Abbildung 9 visualisiert. Der Client sendet seine Befehle nach dem Drücken des jeweiligen



Abbildung 7: Startbildschirm auf dem Android-Client

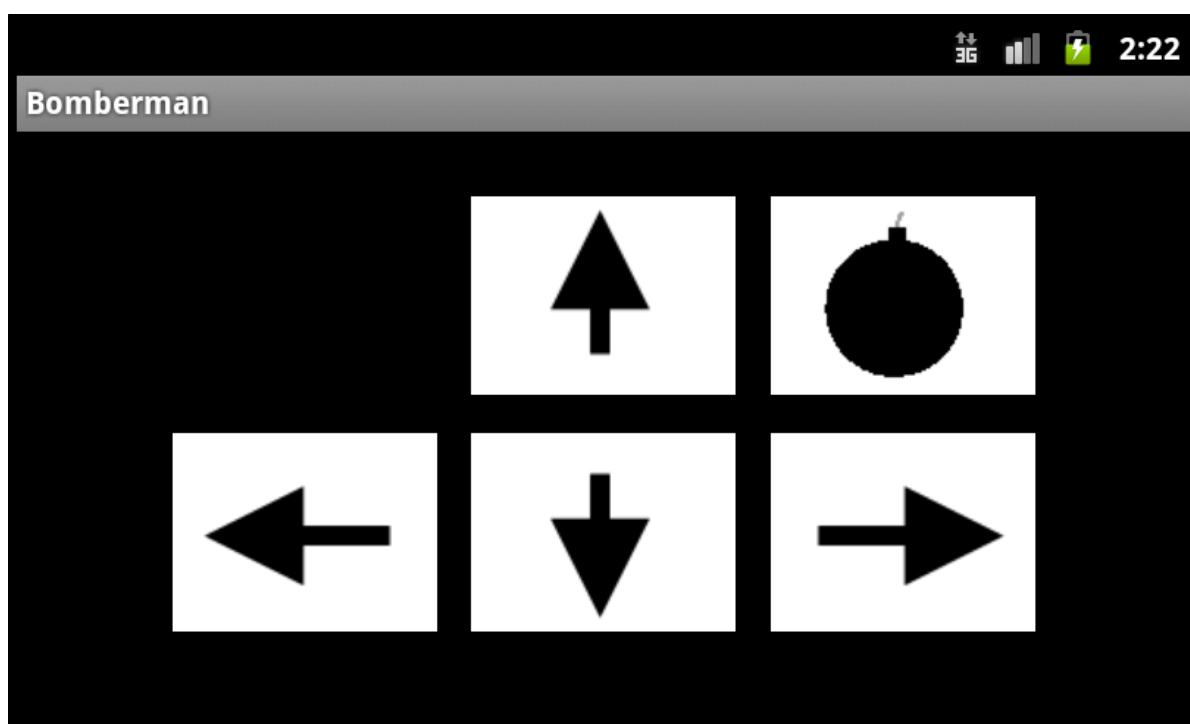


Abbildung 8: Spielbildschirm auf dem Android-Client

3. Technische Umsetzung

Buttons. Dies erfolgt in einem speziellen Kommunikationsthread. Dabei ist die Menge der gesendeten Befehle bei gedrücktem Knopf immer gleich, damit ein Spieler mit einer schnelleren Datentransferrate keinen Vorteil erhält. Beim Server werden die Befehle in eine Warteschlange eingereiht, die nach dem Prinzip „First In - First Out“ abgearbeitet wird. Jeder Spieler hat dabei auf dem Server seine eigene Warteschlange. Pro Frame wird ein Befehl aus der Warteschlange auf Serverseite pro Spieler abgearbeitet. Da der Server normalerweise mit deutlich mehr als 10 Frames pro Sekunden läuft, bleibt die Größe der Warteschlange immer gering.

Die Bewegung der Spielfigur ist auf Serverseite abhängig von der Zeit, die zwischen zwei Frames vergeht. Je mehr Zeit vergeht, desto größer ist die Bewegung. Damit wird sichergestellt, dass die Spielfigur auf unterschiedlichen Rechnern immer gleich schnell bewegt wird. Zusätzlich ist eine maximale Bewegung pro Frame programmiert, um durch eventuelle Verzögerungen sprunghafte Bewegungen zu verhindern.

Es war einiges Feintuning erforderlich bis die Spielfigur sich wirklich flüssig auf dem Server bewegt hat. Für eine spätere Version wäre es trotzdem empfehlenswert, die Kommunikation auf eine Kombination von TCP und UDP zu ändern. Per TCP könnten dann die wichtigen Befehle, wie zum Beispiel ein Disconnect gesendet werden, und per UDP die Bewegungsbefehle, bei denen es nicht problematisch ist, wenn davon welche verlorengehen. Dies könnte die Performanz vermutlich noch weiter verbessern.

Des Weiteren war es notwendig, dass für den Server ein Session Handling implementiert wird. Der Server muss darauf reagieren können, wenn ein Client die Spielsitzung verlässt oder ein neuer hinzukommt. Ebenso ist es notwendig zu überwachen, wann der Server keine neuen Clients mehr aufnehmen kann und das Spiel starten bzw. enden soll. Für das Testen der Kommunikation mit mehreren Clients wurde ein Testclient entwickelt.

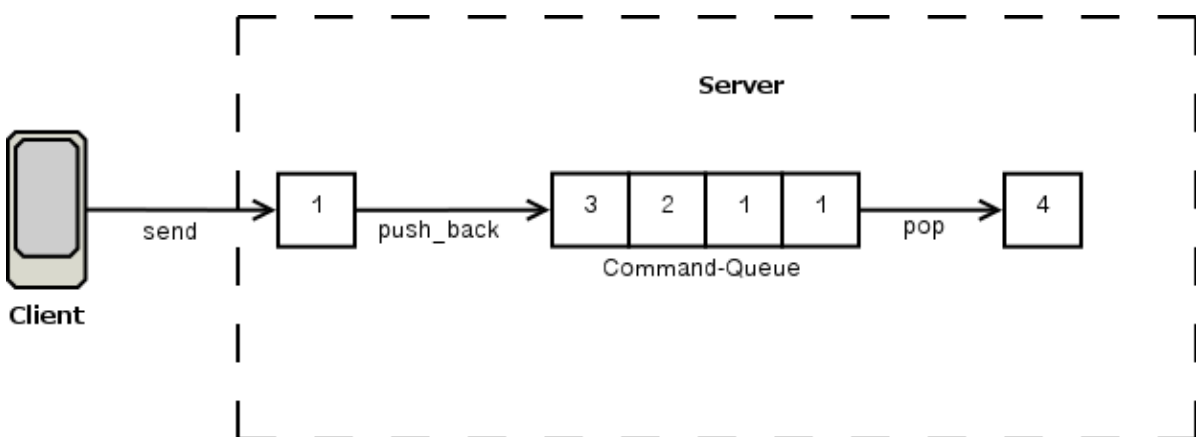


Abbildung 9: Kommunikation zwischen Client und Server je Spielfigur

3.5.7. Programmstruktur

Die grundlegende Programmstruktur orientiert sich an der Vorlage von Heiko Kalista, wie er sie in dem Buch „C++ für Spieleprogrammierer“ [Kal09] beschreibt. In dem Kapitel 12 „Ein Spiel mit der SDL“ erläutert Kalista den grundlegenden Aufbau für die Erstellung eines Spiels mit Hilfe der SDL. Die Implementierung einiger grundlegender Klassen, wie einer Timer-, Framework- oder auch Sprite-Klasse, wurde für den Entwurf des Spiels für die Medienfassade übernommen.

Für das Testen der Anwendung wurde noch die Möglichkeiten des lokalen Spiels hinzugefügt. Dies war besonders beim Debugging des Spielkerns äußerst hilfreich. Insgesamt verfügt der Prototyp über 12 Klassen, wobei im Folgenden nur die wichtigsten Klassen kurz beschrieben werden sollen:

Game: Die *Game*-Klasse regelt den eigentlichen Spielablauf und ist Ausgangspunkt für alle weiteren Programmabläufe. In der Klasse wird weiterhin der aktuelle Status des Spiels verwaltet und mögliche Statusinformationen rechts neben dem Spielfeld ausgegeben.

Server: Mit Hilfe der *Server*-Klasse werden die gesamte Datenkommunikation und die Client-Sitzungen gehandhabt. In einem Thread wird regelmäßig auf neue Clients geprüft und mit einem Polling werden die vorhandenen Clients auf neue Daten abgefragt. So ist sichergestellt, dass es nicht zu einer Blockade kommt, wenn auf Daten eines Clients gewartet wird.

Map: Die *Map*-Klasse lädt das Spielfeld aus einer Datei und rendert das Spielfeld mit allen Feldern. Dafür hält die Map-Klasse einen Vektor mit Feldern.

Tile: Jedes 8*8 Pixel großes Feld ist eine Instanz der Klasse *Tile*. Die Klasse verwaltet Informationen wie die Art des Feldes (zerstörbarer Stein, nicht zerstörbarer Stein, Untergrund) oder ob sich auf dem Feld eine Bombe befindet.

Player: Jede Spielfigur ist die Instanz der *Player*-Klasse. In der Klasse wird die Bewegung, der Status und die Kollisionserkennung der Spielfigur geregelt.

Options: In der *Options*-Klasse sind die verschiedenen Spielparameter konfigurierbar. Hierzu zählen zum Beispiel die Laufzeit einer Runde oder die Wahrscheinlichkeit für ein PowerUp.

Timer: Die *Timer*-Klasse kann zur Messung von Zeitabständen verwendet werden. In der Realisierung eines Spiels sind Zeitmessungen häufig sinnvoll.

4. Ergebnis

Immer mehr Medienfassaden verfügen über Spiele. Im Abschnitt 2 wurde zunächst der theoretische Hintergrund der Thematik beleuchtet. Dabei hat sich gezeigt, dass einige Spielgenres besonders gut für die Umsetzung an Medienfassaden geeignet sind. Hier sind insbesondere kleinere *Actionspiele*, vereinfachte *Sportspiele* und *Puzzlespiele* hervorzuheben. Diese Genres erfüllen am besten die in Unterabschnitt 2.5 erarbeiteten Merkmale. Des Weiteren wurde ein Überblick über typische Probleme bei der Mehrspielerinteraktion geschaffen, sowie Lösungsansätze erarbeitet. Dabei wären insbesondere die Darstellung der Informationen für mehrere Spieler, die Steuerung und die Kommunikation zwischen verschiedenen Geräten hervorzuheben.

Im Abschnitt 3 wurde auf diesen Ergebnissen aufbauend prototypisch ein Spiel für die Medienfassade entwickelt. Dabei wurde anhand eines Bomberman-Klons gezeigt, wo die Unterschiede zu einem herkömmlichen Spiel für einen Monitor oder Fernseher liegen, welche Entwicklungsprobleme es dabei gibt und wie mit diesen umgegangen werden kann. Es hat sich als besonders wichtig herausgestellt, dass stets die Besonderheiten einer Medienfassade und deren Technologien berücksichtigt werden. Das Ergebnis ist ein funktionaler Prototyp, der als Grundlage für weitere Projekte an einer Medienfassade dienen kann.

4.1. Ausblick

Das entwickelte Spiel kann gut als Grundlage für weitere Spiele für die Medienfassade des FKI-Gebäudes genutzt werden. Sollte ein Einsatz an der Fassade angestrebt werden, dann wären ein paar der folgenden weiteren Features noch sinnvoll:

- Ein Karteneditor für das Erstellen neuer Spielfelder.
- Die Erweiterung auf sechs Spieler.
- Umstellung auf die Nutzung von TCP und UDP.
- Ein Timeout für Clients.
- Das Konfigurieren der Spielparameter über eine Datei.
- Erweiterung des Spielclients um zusätzliche Informationen wie zum Beispiel die Anzeige der Farbe der eigenen Spielfigur.
- Eine Funktion für das zufällige Erstellen von Spielfeldern.
- Weitere PowerUps wie zum Beispiel ein PowerUp, mit dem die Geschwindigkeit des Spielers erhöht werden kann.

4. *Ergebnis*

- Eine Benutzerauthentifizierung und damit verbunden eine Highscoreliste, sowie eine Individualisierung der Spielfigur.

A. Kommunikations-Protokoll

Befehl	Zahl	Bedeutung
NA	0	Nicht definierter Befehl
LEFT	1	Spielfigur nach links bewegen
RIGHT	2	Spielfigur nach rechts bewegen
UP	3	Spielfigur nach oben bewegen
DOWN	4	Spielfigur nach unten bewegen
BOMB	5	Bombe platzieren
DISCON	10	Verbindung trennen

Literatur

- [Beh10] BEHRENS, HEIKO: *Cross-Platform App Development for iPhone, Android & Co. - A Comparison*. Webseite: <http://heikobehrens.net/2010/10/11/cross-platform-app-development-for-iphone-android-co-%E2%80%94-a-comparison-i-presented-at-mobiletechcon-2010/>; Zuletzt besucht am 05.09.2012, Oktober 2010.
- [BGW⁺] BORING, SEBASTIAN, SVEN GEHRING, ALEXANDER WIETHOFF, MAGDALENA BLÖCKNER, JOHANNES SCHÖNING und ANDREAS BUTZ: *Multi-User Interaction on Media Facades through Live Video on Mobile Devices*.
- [Bli] BLINKENLIGHTS: *Blinkenlights*. Webseite: <http://blinkenlights.net/blinkenlights>; Zuletzt besucht am 16.08.2012.
- [BS12] BERGEMANN, STEPHAN und ROBIN SCHLEGEL: *Medienfassade - Möglichkeiten von Visualisierung und Interaktion*, 2012.
- [Don12] DONATH, ANDREAS: *Elektronischer Würfel für Tabletspiele*. Webseite: <http://www.golem.de/news/dice-elektronischer-wuerfel-fuer-tabletspiele-1206-92424.html>; Zuletzt besucht am 25.08.2012, September 2012.
- [Kal09] KALISTA, HEIKO: *C++ für Spieleprogrammierer*. Hanser, 2009.
- [Lan] LANTINGA, SAM: *SDL_ttf 2.0*. Webseite: http://www.libsdl.org/projects/SDL_ttf; Zuletzt besucht am 16.08.2012.

Literatur

- [Pak12] PAKALSKI, INGO: *Android vor Symbian und iOS*. Webseite: <http://www.golem.de/news/smartphones-in-deutschland-android-vor-symbian-und-ios-1205-92120.html>; Zuletzt besucht am 05.09.2012, Mai 2012.
- [RA03] ROLLINGS, ANDREW und ERNEST ADAMS: *Game Design*. New Riders, 2003.
- [Sau04] SAUTER, JOACHIM: *Das vierte Format: Die Fassade als mediale Haut der Architektur*, 2004.
- [SDL] SDL: *SDL*. Webseite: <http://www.libsdl.org/>; Zuletzt besucht am 16.08.2012.
- [SLM] SAM LANTINGA, ROAY WOOD und MASAHIRO MINAMI: *SDL_net 1.2*. Webseite: http://www.libsdl.org/projects/SDL_net; Zuletzt besucht am 16.08.2012.
- [SZ04] SALEN, KATIE und ERIC ZIMMERMAN: *Rules of Play - Game Design Fundamentals*. MIT Press, 2004.
- [Tec] TECHNOLOGIES, FREITAG: *GRID25*. Webseite: <http://www.ledmedia.de/>; Zuletzt besucht am 31.08.2012.