

Hochschule Furtwangen - Fakultät Digitale Medien

Interaktionsdesign

Wintersemester 2014/15



Fabian Gärtner, MIM1

Sarah Häfele, MIM1

Alexander Scheurer, MIM1

Linda Schey, MIM2

Johannes Winter, DIM1

Meike Zöckler, DIM1

Prof. Patricia Stolz

Prof. Dr. Matthias Wölfel

4. Februar 2015

Abstract

Die vorliegende Arbeit stellt die Dokumentation zur Konzeption und prototypischen Umsetzung von *BlinkenTiles* dar. BlinkenTiles ist eine vielseitig einsetzbare, interaktive, großflächige Installation, die es Personen erlaubt, durch eine auf den Boden projizierte Sound-Matrix und durch das Tracking einer Microsoft Kinect mit dem eigenen Körper Musik zu machen. Die Installation wurde im Rahmen der Veranstaltung *Interaktionsdesign* in den Masterstudienängen *Design Interaktiver Medien* sowie *Medieninformatik* an der Fakultät Digitale Medien der Hochschule Furtwangen im Wintersemester 2014/2015 unter der Betreuung von Frau Prof. Stolz und Herrn Prof. Dr. Wölfel entwickelt.

Inhaltsverzeichnis

1 Grundidee und Konzept BlinkenTiles	5
1.1 Ideenfindung	5
1.1.1 Idee 1: Interaktiver Werbescreen	5
1.1.2 Idee 2: Der Entschleuniger	6
1.2 Funktion	7
1.2.1 Modus 1: Freestyle	7
1.2.2 Modus 2: Challenge	8
1.2.3 Modus 3: Idle	8
1.3 Zielgruppenanalyse	8
1.4 Design	9
1.5 Werbebotschaft – Einsatz, Sinn und Zweck	11
1.6 Aufgaben-Verteilung	11
2 Umsetzung des Prototypen	13
2.1 Softwaretechnische Entscheidungen	13
2.2 Aufbau	13
2.3 Ansteuerung Scheinwerfer	16
2.4 Multimonitor Support	17
2.5 Die grafische Oberfläche (GUI)	19
2.6 Spielfeld und Felder	21
2.7 Call to Action	22
2.7.1 Der Presentation-Screen	22
2.7.2 Spielfeld-Angaben	23
2.8 Personenerkennung mittels Microsoft Kinect	26
2.8.1 Aufgetretene Probleme	28
2.9 Audiogestaltung und Inspiration	29
3 Ergebnisse	32
3.1 Probleme und Verbesserungen	32
3.2 Nutzergruppen-Analyse	33
3.3 Showreel	34
3.4 Vision	36
4 Impressionen	37
4.1 Impressionen Tag der Medien	37

4.2	Impressionen Erste Tests	41
4.3	Technische Zeichnungen	43

1 Grundidee und Konzept BlinkenTiles

1.1 Ideenfindung

Meike Zöckler

Am Anfang war eine Idee. Oder auch zwei.

1.1.1 Idee 1: Interaktiver Werbescreen

Die erste davon sah ein interaktives Game vor. Unliebsame Wartesituationen sollen endlich sinnvoll genutzt werden: mit Spiel und Spaß im Multiplayer statt gähnender Langeweile. An typischen »Warteorten« sollten Screens aufgestellt werden. Per QR-Code sollte sich der Wartende in das Spiel einloggen. Zur Verfügung stehen sollten bekannte Arcade-Klassiker wie bspw. Pong oder Pacman. Jeder hinzukommende Spieler übernimmt dabei eine Spielfigur. Einer ist Pacman, die nächsten werden die Geistern, die Pacman jagen sollen. Je mehr Spieler dazu kommen, desto größer wird das Labyrinth, desto mehr Geister und Pacmans gibt es. Das eigene Smartphone dient dabei als Controller.

Die **Schwierigkeiten** bei dieser Idee umfassen mehrere Aspekte. Zum einen ist die Hürde zur Teilnahme deutlich höher als bei der nachfolgend beschriebenen Idee. Der Wartende braucht ein Smartphone. Er muss es gut genug bedienen können, um sich via QR-Code einzuloggen und um es dann auch noch als Controller verwenden zu können. Um dem entgegen zu kommen, sollte eine möglichst einfache, übersichtliche und für verschiedene Spiele einsetzbare Steuerung implementiert werden. Ein weiteres Problem hätte die Konnektivität darstellen können. Für ein einwandfreies Spielerleben sind eine stabile und schnelle Verbindung sowie ein unkompliziertes Einloggen vonnöten.

Von technischen Schwierigkeiten abgesehen benötigt der potentielle Spieler genügend (Warte-)Zeit, um sich darauf einzulassen und mit dem nötigen Spaß und Interesse an das Spiel heranzugehen. Er sollte dieses Erlebnis eben mit Spaß verknüpfen und nicht mit Stress und Zeitdruck, was an Wartestationen wie Bushaltestellen oder Bahnhöfen der Fall sein könnte, wo er mitunter auch gewaltsam aus seinem Spielerleben gerissen wird, sollten Bus oder Bahn endlich einfahren. Außerdem hätte der Call-to-Action ein sehr deutlicher und ansprechender sein müssen, um neben den Hardcore-Gamern auch Casual- oder Nongamer anzusprechen. Hier hätte man ggf. einen entsprechenden Anreiz in Form von Gewinnchancen schaffen müssen.

1.1.2 Idee 2: Der Entschleuniger

Die zweite Idee beinhaltete ein Klavier. Dieses Klavier sollte auf einem Platz stehen. Sobald Passanten vorbeilaufen, wird Musik abgespielt. Wobei diese Musik in Tempo des Laufschritts und die Tonhöhe oder das Instrument durch die Statur der Person angepasst sein sollte. Überlegt wurden zwei verschiedene Modi:

Modus 1: Man sollte gemeinsam Musik machen können. Man sollte mit dem eigenen zugewiesenen Instrument und denen anderen Menschen ein Lied »komponieren«. Der Spaß an der Musik und am Experimentieren sollte im Vordergrund stehen, wobei sich hier die Frage aufdrängte, ob denn aus dieser einfachen Spielerei denn auch ein guter Song werden könnte. Die Installation könnte schnell an Attraktivität verlieren, sollten die erzeugten Klänge und Melodien nicht gut klingen.

Modus 2: Dieser Modus verfolgte ein klares Ziel: Entschleunigung. Zu hören ist zunächst ein einfacher Song. Läuft jemand vorbei und das viel zu schnell und zu hektisch, passt sich der Song dem Tempo des Passanten an. Läuft er zu eilig, ist auch das Lied viel zu schnell. Erst wenn er sein Tempo zurücknimmt, kann er das Lied im Originaltempo hören. Damit sollte Aufmerksamkeit auf die Schönheit der einfachen Dinge und eines entschleunigten, vielleicht sogar entspannteren Lebens gelenkt werden. Überlegt wurde, die Installation in dieser Variante auch als Werbemöglichkeit für (Luxus-)Uhren einzusetzen.

Zwischenzeitlich wurde überlegt, zur Verdeutlichung des zu schnellen Laufens die »Temposünder« zu blitzen, was aber verworfen wurde, da die Passanten von selbst auf ihr zu hohes Tempo aufmerksam werden sollten.

Bei dieser Idee waren die **Schwierigkeiten** anderer Natur als zuvor bei der Interaktiven Werbeplattform. Einerseits hätte man zur Präsentation einen großen Platz oder zumindest eine größere Fläche mit hoher Passantenzahl benötigt. Somit eignet sich die Installation nicht für alle Orte. Andererseits hätten sich auch hier die Menschen erst darauf einlassen müssen. Der Reiz hätte stark genug sein müssen, damit sie erkennen, dass die Installation auf sie direkt reagiert. Als Idee zur Lösung wurde vorgeschlagen, dass parallel zu den echten Passanten Comicfiguren über eine bestehende Leinwand laufen und sie so neben dem akustischen auch einen optischen Reiz wahrnehmen, der sie zur Reaktion zwingt.

Da beide Ideen nicht zur vollständigen Zufriedenheit gereichten, ist aus deren Verschmelzung schließlich Idee Nummer drei **BlinkenTiles** entstanden.

1.2 Funktion

Meike Zöckler

BlinkenTiles orientiert sich zunächst an der Funktion einer Soundmatrix. Diese besteht eben aus einem Grid an Feldern, jedem ist ein Ton hinterlegt. Ähnlich einem Stepsequenzer, dessen Funktionsweise später noch beschrieben wird, tastet ein Impuls die Felder ab. Sind die Felder aktiviert, erklingt bei der Abtastung an entsprechender Stelle der zugewiesen Ton.

Wie eine solche Matrix als Applet aussehen kann, ist unter folgender Adresse auszuprobieren: <http://www.buzzedgames.com/sound-matrix-game.html>.

Diese Matrix ist deutlich größer angelegt als unser Prototyp. Eine Aktivierung erfolgt per Klick auf die Felder. Diese werden danach weiß angezeigt. Der Impuls läuft von links nach rechts. Zwar sieht der Nutzer den Impuls nicht, aber denn noch erhält er zusätzlich zum abgespielten Ton ein optisches Feedback der abgespielten Felder. Die Felder bleiben bis zum nächsten Klick darauf aktiviert. Im Falle von BlinkenTiles werden die Felder auf den Boden projiziert. Die Aktivierung eines Feldes erfolgt durch Positionierung einer Person auf einem Feld. Es bleibt solange aktiviert, wie der Mensch auf diesem Feld stehen bleibt.

Wie auch bei den vorherigen Ideen wurden bei BlinkenTiles zwei Modi angedacht:

1.2.1 Modus 1: Freestyle

Zunächst wird ein einfacher Backingtrack abgespielt, in erster Linie bestehend aus Rhythmus und möglicherweise Bass oder einfache Akkorde. Die eigentliche Melodie wird von Menschen geschaffen, die sich auf der Projektion bewegen. Dabei sollte man genügend Raum zum Ausprobieren haben: In der ursprünglichen Idee sollte es möglich sein, mit verschiedenen Mustern, die durch das Ausstrecken von Armen und Beinen entstehen (Linien, Quadrate, Rechtecke oder gleich die ganze Tetrispalette), neue Töne zu erzeugen. Aufgrund der Entscheidung für eine Pentatonik, was im Kapitel zu Audio weiter erläutert wird, haben wir diese Idee verworfen. Die Felder werden so abgespielt, wie sie hinterlegt sind. Entlang der kurzen Seite werden Akkorde erzeugt, entlang der langen sind sie als Notenwerte durch den durchlaufenden Impuls zeitlich versetzt.

1.2.2 Modus 2: Challenge

Der Challenge Mode ist an das Konsolenspiel *GuitarHero* angelehnt. Hier werden den Nutzern die Felder vorgegeben, mit denen die Melodie für die entsprechenden Songs nachgetanzt oder nachgehüpft werden sollen. Je nach Song, wie schnell oder wie komplex die Melodie ist, desto mehr Menschen werden benötigt, um die Melodie fehlerfrei zu hüpfen. Man spielt nicht gegeneinander sondern miteinander und muss teilweise zusammenarbeiten, um die richtigen Felder zu treffen. Durch den Verzicht auf eine Punktevergabe stehen der Spaß und das Miteinander im Vordergrund, nicht der kompetitive Aspekt wie bei *Guitar Hero*.

1.2.3 Modus 3: Idle

Der Idle Mode dient dazu zu zeigen, wie das Spielprinzip im Freestyle funktioniert. Es werden zufällig mehrere Felder pro Impulsdurchlauf aktiviert und mit Fußspuren angezeigt, die klar machen sollen, dass man sich auf die Projektion, auf die Felder stellen muss, damit etwas passiert. Werden diese aktivierte Felder abgetastet, erzeugen sie ebenfalls Töne wie im Freestyle. Somit soll der potentielle Zuschauer intuitiv in Bild und Ton wahrnehmen, wie die Installation funktioniert. Damit kommt sie ohne lange Erklärungen oder zusätzliche Tutorials, Intros oder zusätzliche Screens aus.

Nach 30 Sekunden Inaktivität wird dieser Modus automatisch geschaltet.

1.3 Zielgruppenanalyse

Johannes Winter

Der Vorteil der Installation besteht darin, dass es (zumindest im experimentellen Modus) keine Barrieren für bestimmte Altersgruppen gibt. Menschen jeden Alters ist es möglich mit der Installation zu interagieren, da weder motorisch noch kognitiv anspruchsvolle Usereingaben von Nöten sind. Das Userinterface verlangt außerdem keine Vorkenntnisse, wie es zum Beispiel bei einer App der Fall wäre. Auch ist die Hemmschwelle sowohl technisch als auch psychologisch sehr niedrig, da auf Userseite keine Geräte synchronisiert werden und keine peinlichen Bewegungen vollzogen werden müssen. Auch sind für das effektive Benutzen der Installation keine musikalischen Kenntnisse gefordert. Dennoch

zielt die Installation und die damit verbundene Aussage auf junge, experimentierfreudige und möglicherweise auch musikalisch interessierte Menschen. Besonders der Challenge Modus, der einen deutlich bewegungsorientierteren Ansatz bietet und auch kognitiv anspruchsvoller ist, ist eher für Jugendliche und junge Erwachsene interessant, die Spaß an Bewegung und Herausforderungen in Kombination mit Musik haben.

Schwierigkeiten, Lösungen und schließlich das Endergebnis werden in den nachfolgenden Kapiteln zur Umsetzung beschrieben.

1.4 Design

Johannes Winter

Bedingt durch die Beschränkung durch die Projektionsfläche des Beamers und der menschlichen Körpermaßen wurde die Matrix auf 5x8 Felder beschränkt. Die schmale Seite beschreibt die Tonhöhe – jede Reihe ein Ton der Pentatonik. Die breite Seite sind die Tondauern – ein jedes Feld eine Achtel (oder je nach Song im Challenge Mode andere Notenwerte). Die genaue Funktionsweise ist im Abschnitt 2.9 beschrieben.

Das ursprüngliche Farbschema sah vor, dass entsprechend der Abstufung der Töne auch die Felder in Grautönen abgestuft werden sollten. Von hohen nach tiefen Tönen sollten auch die Felder von Weiß nach Grau eingefärbt werden. Aktivierte Einzelfelder blau, mehrere zusammenhängend aktivierte Felder zur Erzeugung der Mischtöne grün. Der Impuls wurde zunächst violett angedacht und die abgetasteten Felder rot (vgl. Abbildung 1). Da der Boden zu dunkel ist, um die Mischtöne kontrastreich darzustellen, wurde auf diese verzichtet. Statt der Grauabstufungen sind alle Felder weiß. Aktivierte Felder bleiben blau, gerade abgetastete werden grün und erhalten spezielle Effekte, um deutlicher hervorzuheben, dass genau diese Felder gerade vom Impuls abgetastet wurden. Sie shaken und versprühen Partikel. Der Impuls selbst wurde gelb angezeigt.

Im Challenge Mode werden die vorgegebenen Felder blassrot mit blauroter Umrandung angezeigt. Aktivierte Felder sind blau. Ob man richtig steht oder nicht, zeigt sich erst, sobald der Impuls durchläuft. Richtig aktivierte Felder werden grün, falsche rot angezeigt. Im Idle Mode werden in den aktivierte Feldern zusätzlich verschiedene Fußspuren angezeigt. Schuhe, Hunde und Dinosaurier, um das Bild aufzulockern und den spielerischen Charakter von BlinkenTiles zu verdeutlichen.

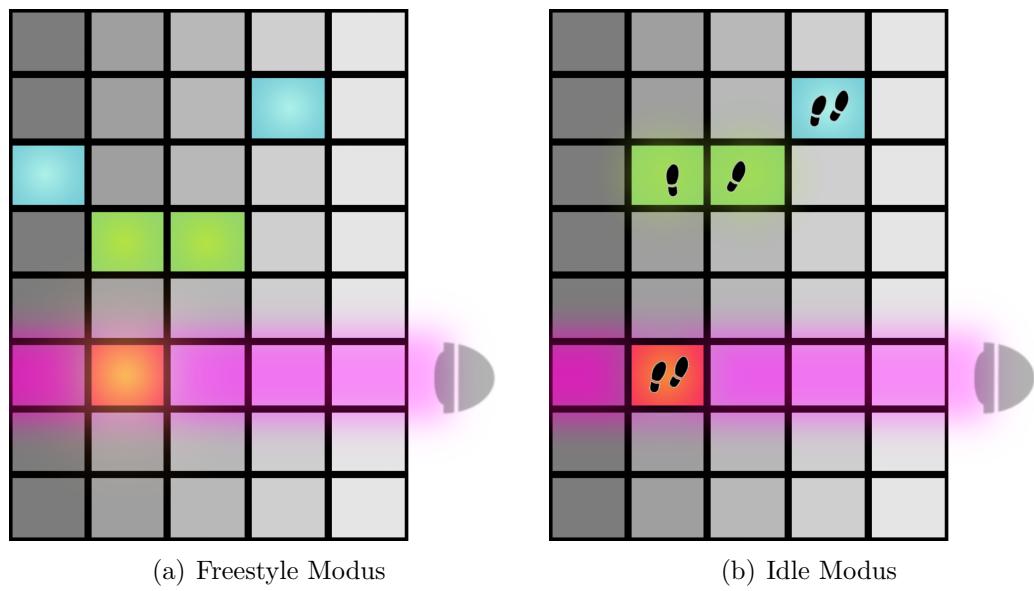


Abbildung 1: Ursprüngliches Farbschemas aus der Konzeptphase

1.5 Werbebotschaft – Einsatz, Sinn und Zweck

Meike Zöckler

Eine durch die Traversen und Stethoskopstative mobile Version der Installation ließe sich für verschiedene Einsatzzwecke verwenden, vorausgesetzt, der Raum, in dem sie aufgebaut werden soll, ist hoch genug.

Im vorliegenden Fall dient sie der Bewerbung der interdisziplinären Kooperation zwischen der Hochschule Furtwangen und der Musikhochschule Trossingen. Viele wissen nichts von dieser Kooperation und nutzen demnach auch nicht das Angebot, dort Wahlpflichtveranstaltungen mit Schwerpunkt Musik zu belegen. Umgekehrt genauso, dass nur wenige Studierende das Angebot nutzen, Veranstaltungen in Furtwangen zu belegen.

Ziel ist es, eine gewisse Awareness für den Partnerstudiengang Musikdesign zu entwickeln und dadurch Kompetenzen im musikalischen Bereich zu erweitern, die künftig möglicherweise durch gemeinsame Projekte vertieft werden könnten.

Die künftige Kooperation könnte verschiedene Studiengänge umfassen:

- **Konzeption und Design:** Medienkonzeption (B.A.), Design interaktiver Medien (M.A.)
- **technische Umsetzung:** Medieninformatik (B.A.), Medieninformatik (M.A.)
- **Musikalischer Input:** Musikdesign (B.Mus.)
- **Onlinepräsentation und Marketing:** Online Medien (B.Sc.)

Zur choreografischen Vorführung und Untermalung könnte die Kooperation auch auf Sing & Move mit Schwerpunkt Music & Movement (B.Mus.) erweitert werden.

BlinkenTiles bietet von allen Studiengängen das Beste: Ansprechendes Design, gute technische Umsetzung, Spiel mit Musik und Bewegung, in dem sich die Nutzer später frei entfalten können, um gemeinsam Musik zu erzeugen.

1.6 Aufgaben-Verteilung

Das Team

Tabelle 1 fasst die verschiedenen Aufgabenbereiche bei der Konzeption und Umsetzung von BlinkenTiles zusammen und listet die Hauptverantwortlichen für die Bearbeitung der einzelnen Aufgaben. Unterstützt wurden die genannten Personen dabei in der Regel durch die anderen Gruppenmitglieder. Genaueres dazu kann dem Repository auf GitHub¹ oder den einzelnen Kapiteln dieser Dokumentation entnommen werden.

Aufgabe	Hauptverantwortlich
Idee und Konzeption	Fabian Gärtner, Sarah Häfele, Alexander Scheurer, Linda Schey, Johannes Winter, Meike Zöckler
Orga Technik	Fabian Gärtner, Alexander Scheurer
Tests und Aufbau	Fabian Gärtner, Sarah Häfele, Alexander Scheurer, Linda Schey
Techn. Zeichnungen	Fabian Gärtner, Sarah Häfele
Logo und Grafiken	Sandra Beuck
Ansteuerung Scheinwerfer	Linda Schey
Bau »Photonenbündler«	Fabian Gärtner, Linda Schey, Meike Zöckler
Multimonitor-Support	Linda Schey
Personenerkennung	Fabian Gärtner
Spielfeld und Tiles	Alexander Scheurer
Spiellogik	Alexander Scheurer
Idle-Animation	Sarah Häfele
Präsentationsbildschirm	Sarah Häfele
Netzwerk/XML	Sarah Häfele, Linda Schey
Musik	Johannes Winter
Kamera & Schnitt Teaser	Fabian Gärtner, Meike Zöckler, Johannes Winter

Tabelle 1: Aufgaben Verteilung - Übersicht

¹<https://github.com/InformatischesQuartett/BlinkenTiles>

2 Umsetzung des Prototypen

2.1 Softwaretechnische Entscheidungen

Das Team

Zur Entwicklung der Anwendung für den Prototypen fiel die Entscheidung auf die 3D-Game-Engine Unity². Gründe hierfür waren u. a., dass der Umgang mit Unity den meisten Projektmitgliedern aus anderen Lehrveranstaltungen geläufig war und die Anbindung externer Softwarebibliotheken (z.B. das *Microsoft Kinect*-SDK oder *OpenCV* zur Personenerkennung, vgl. Seite 26) über C# als Programmiersprache vergleichsweise einfach ist. Zudem wurde entschieden Git als Sourcecode-Verwaltung einzusetzen, wodurch der Programmcode von BlinkenTiles jederzeit online abrufbar ist³. Die wichtigsten Einstellungen in BlinkenTiles können über eine externe Konfigurationsdatei vorgenommen werden. Dadurch können alle Parameter, die in den folgenden Abschnitten näher erläutert werden (vgl. Abschnitte 2.3, 2.6 und 2.8), geändert werden, ohne dass die komplette Anwendung noch einmal kompiliert und exportiert werden muss.

Für die Kommunikation mit den LED-Scheinwerfern wurde *Freestyler*⁴, eine kostenfreie und umfangreiche DMX-Lichtsteuerungs-Software verwendet, da diese für das verwendete DMX512-Interface in der Produktbeschreibung empfohlen wurde und einwandfrei kompatibel ist. Außerdem bietet Freestyler die Möglichkeit per Befehl problemlos aus anderen Programmen (hier die BlinkenTiles-Anwendung) heraus auf Funktionen zuzugreifen, diese zu steuern und so die LED-Scheinwerfer anzusprechen. Mehr dazu im folgenden Abschnitt.

2.2 Aufbau

Sarah Häfele

Die Installation besteht aus mehreren Hardware- und Software-Komponenten und soll hier mit Hilfe einer kurzen, modellhaften Übersicht, die den endgültigen Aufbau des Prototypen zeigt, skizziert werden. Die darauffolgenden Kapitel gehen anschließend näher auf die einzelnen Komponenten ein. Der Prototyp der Installation wurde am 20.01.2015 im Rahmen des *Tag der Medien* der Fakultät

²<http://www.unity3d.com>

³<https://github.com/InformatischesQuartett/BlinkenTiles>

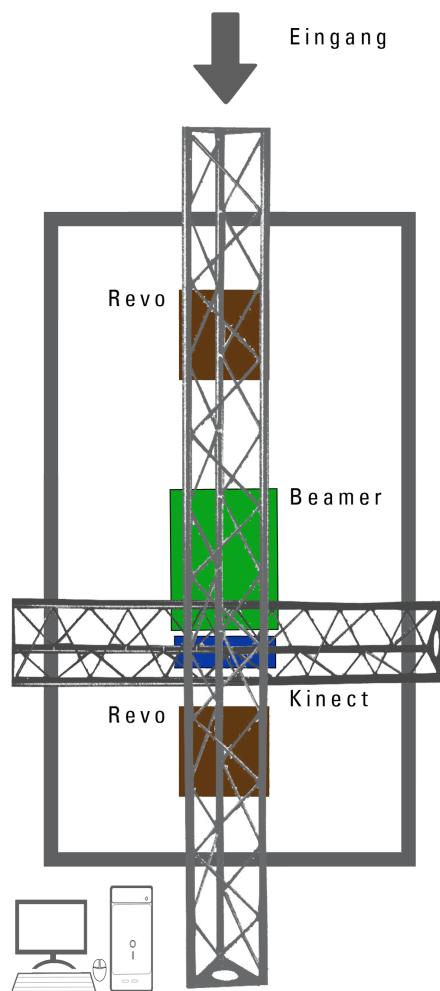
⁴<http://www.freestylerdmx.be/>

Digitale Medien aufgebaut und getestet. Auf zwei Ebenen wurden Gerätschaften installiert, damit die Besucher schon beim Eintreten mit akustischen und visuellen Reizen konfrontiert und zum Mitmachen bewegt wurden. Abb. 2 (a) und 2 (b) zeigen Erdgeschoss und ersten Stock des Gebäudes.

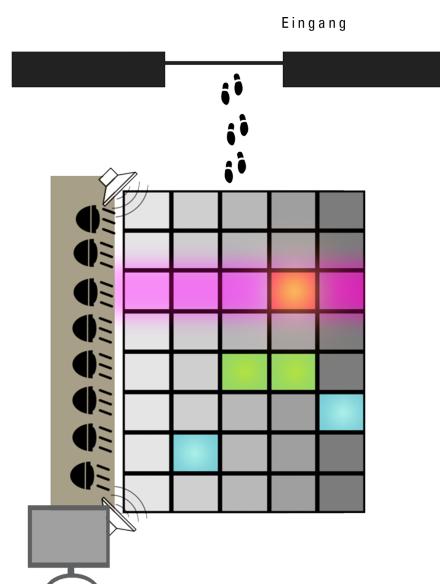
Im ersten Stockwerk hängt über dem Lichthof eine von zwei Stativen (sogenannte Traversenlifte) getragene sieben Meter lange Traverse, in deren Mitte ein *Epson EB-4950WU*-Beamer (in der Abb. 2 (a) grün markiert), der durch eine speziell von der Werkstatt der Hochschule angefertigten Metallhalterung gehalten wird, in das Erdgeschoss des Gebäudes strahlt. Der Beamer, eine Leihgabe des Kreismedienzentrums Schwarzwald-Baar, kann mit seiner Helligkeit von bis zu 4.500 ANSI Lumen und einer Auflösung von 1920 x 1200 Pixeln den dunklen Boden des I-Baus mühelos ausleuchten und ist zudem für längere Betriebszeiten ausgerichtet. Durch die erhöhte Stellung der Traversenlifte hängt der Beamer sechs Meter über dem Erdgeschossboden. Angeschlossen ist der Beamer mittels HDMI an einen Computer, auf dem die Steuerung läuft.

Auf der Brüstung des Lichthofes liegt im 90° Winkel dazu eine zwei Meter lange Traverse, die durch von der Schreinerei der Hochschule angefertigte Holzhalterungen und durch sogenannte *Doughty Clamps* fixiert wird. Sie wird nicht ganz mittig zu der längeren Traverse ausgerichtet, damit die mit Spannfix daran befestigte Microsoft Kinect (in der Abb. 2 (a) blau markiert) nicht das Bild des Beamers stört. Durch die direkte Anbringung auf dem Geländer des Lichthofes, hängt die Kinect auf 4,5 Meter Höhe und kann so optimal die Bewegungen der im Erdgeschoss befindlichen Personen aufnehmen. Zudem sind zwei Scheinwerfer (so genannte *Revos*, in Abb. 2 (a) braun markiert) an der längeren Traverse links und rechts vom Beamer angebracht und versehen den Boden mit Lichteffekten, die automatisiert durch Sound gesteuert werden.

Im Erdgeschoss (siehe Abb. 2 (b)) wird durch den Beamer ein 5 x 8 flächiges Raster projiziert, welches die einzelnen Töne repräsentiert. Acht LED-Scheinwerfer untermalen die Rhythmuslinie, welche bestimmt, wann ein Ton abgespielt wird. Die Scheinwerfer sind durch ein DMX-Interface untereinander und mit dem Computer verbunden und werden durch ein eigenes Programm gesteuert (vgl. Abschnitt 2.3). Ein großer LED-Monitor zeigt den Besuchern Informationen an und dient zusätzlich als Motivation (vgl. auch Abschnitt 2.7). Die Musik wird durch einen mobilen, jedoch leistungsstarken Radiorekorder (*Boombox* genannt) abgespielt. Diese ist ebenfalls über ein XLR-Kabel und mittels Adapter an den Computer angeschlossen.



(a) I-Bau, 1. Stock, Lichthof



(b) I-Bau, Erdgeschoss

Abbildung 2: Aufbau der Installation im I-Bau (Aufsicht)

2.3 Ansteuerung Scheinwerfer

Linda Schey

Für die LED-Scheinwerfer wurden acht *Eurolite LED CLS-18 QCL RGBW* verwendet, die jeweils über einen dreipoligen XLR-Ein- bzw. Ausgang verfügen, so dass sie in Reihe geschaltet über das DMX512-Protokoll mit zwölf Kanälen gesteuert werden können. Ein *Eurolite USB-DMX512-PRO-Interface* dient als Schnittstelle zwischen den Lichtern und dem Computer. Angesteuert werden die Lichter dann mittels der in Abschnitt 2.1 bereits erwähnten Software Freestyler. Abb. 3 zeigt grob, wie die drei Komponenten zusammenhängen. Es wird hier den Pfeilen der Abbildung entsprechend nur in eine Richtung kommuniziert, da das Verhalten der Scheinwerfer ausschließlich durch die Applikation gesteuert wird und keine rückläufige Kommunikation nötig ist.



Abbildung 3: Verknüpfung der Komponenten

Jeder Scheinwerfer besteht aus 18 LEDs die in drei Reihen mit jeweils sechs LEDs angeordnet sind. Zwei Zeilen, also sechs LEDs, werden in der Applikation jeweils als ein Segment verwendet, das mit vier Farbkanälen separat angesteuert werden kann. Insgesamt werden 12 Farbkanäle pro Scheinwerfer eingesetzt. Innerhalb der BlinkenTiles-Anwendung gibt es zwei Klassen, die hauptverantwortlich für die Lichtsteuerung sind: *Spot.cs* und *LightController.cs*. *Spot.cs* repräsentiert in der Anwendung einen LED-Scheinwerfer. Diese Klasse verwaltet die Farbwerte für jeden einzelnen Spot und dessen Segmente. In der Klasse *LightController.cs* werden die Farbwerte für alle acht Scheinwerfer in einem Array gespeichert. Dieses Array wird mit den Farbwerten der einzelnen Scheinwerfer gefüllt. Die Farbwerte der Scheinwerfer, werden regelmäßig pro Frame aktualisiert. Dabei wird überprüft welcher der Scheinwerfer gerade aktiv sein muss und anhand einer Zeitreferenz welches Segment des Scheinwerfers aktiviert sein soll (*UpdateFaderValues()*). Alle anderen Scheinwerfer und deren Segmente werden auf Schwarz gestellt, also nicht leuchtend.

Um die Scheinwerfer aus der Anwendung BlinkenTiles heraus zu steuern wird ein Zwischenschritt über die Software Freesyler gemacht, da diese Software die Ansteuerung der einzelnen Segmente eines jeden Scheinwerfers sehr komfortabel gestaltet. Um aus BlinkenTiles heraus mit Freestyler zu kommunizieren wurde eine DLL (*LetThereBeLight.dll*) erstellt, die durch die beiden C#-Standardfunktionen *FindWindow()* und *SendMessage()* zwischen beiden Anwendungen vermittelt.

Da die Scheinwerfer eingesetzt werden sollten, um die Rhythmuslinie des Spielfeldes zu unterstützen und zu verstärken, musste das Licht der LEDs stärker gebündelt werden, da diese in der Regel, trotz Abgrenzung in einzelne Reihen, eine sehr breite Streuung haben. Als Lösung für dieses Problem wurden acht sogenannte »Photonenbündler« (lediglich ein scherhafter Name) gebastelt. Dabei handelt es sich um schwarzlakierte Kartonkonstruktionen, die so entworfen wurden, dass sie jeweils zwei LED-Reihen stärker fokussieren. Sie können einfach in die LED-Scheinwerfer gesteckt und mittels Klebeband fixiert werden. Der Vorteil zeigt sich nun, wenn die LED-Scheinwerfer auf den Boden gerichtet und ein Lauflicht programmiert wird: Durch die Bündelung ergeben sich einigermaßen schmale Strahlen, die neben der Rhythmuslinie der BlinkenTiles-Anwendung als zusätzliches optisches Merkmal verwendet werden können und so den Nutzern vom Boden aus ebenfalls als Hilfe bei der Interaktion mit der Installation dienen.

2.4 Multimonitor Support

Linda Schey

Schon zu Beginn des Projektes wurde versucht, eine möglichst große Projektionsfläche zu erreichen. In Ermangelung eines leistungsstarken Beamers, wurde auf drei Beamer zurückgegriffen, die dann nebeneinander je ein Drittel der Applikation darstellen sollten. Um dies zu realisieren, musste aber das zu projizierende Bild in drei einzelne Bilder geteilt werden. Da die Beamer aber, um ein größeres Bild zu erzielen, nicht nebeneinander, sondern übereinander aufgebaut werden sollten, mussten die erhaltenen Bilder jeweils um 90° gedreht werden. Dafür wurde in Unity eine neue Kamera erstellt, die die Szene als *RenderTexture* aufnahm. Diese RenderTexture wurde dann in drei einzelne Texturen getrennt, jede Textur um 90° gedreht und anschließend nebeneinander auf der GUI dargestellt. Dies war bereits fertig in BlinkenTiles implementiert.

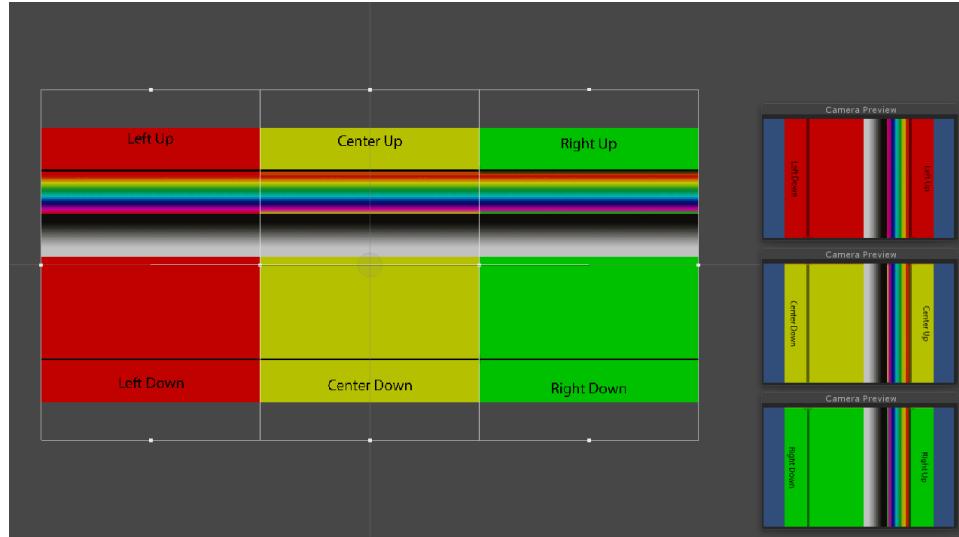


Abbildung 4: Aufbau der Szene in Unity mit drei RenderTexture-Ausschnitten am Beispiel eines Testobjektes

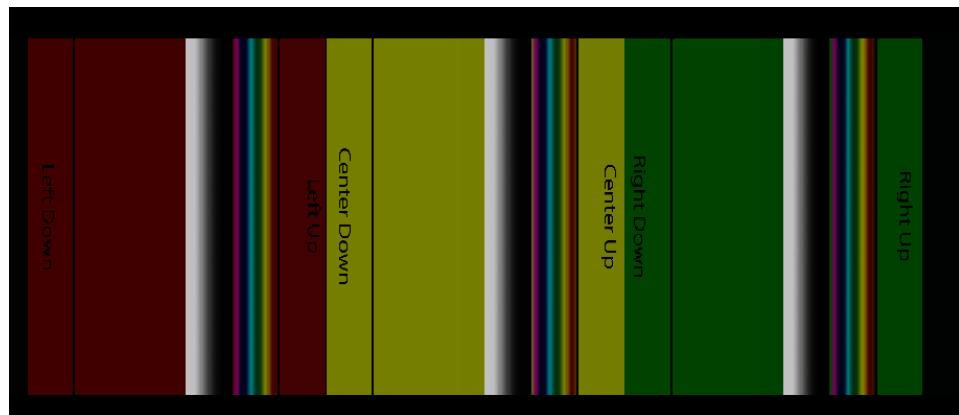


Abbildung 5: RenderTextures nebeneinander auf die GUI gerendert

Ein anderer Ansatz war drei um 90° gedrehte Kameras in die Szene zu integrieren die jeweils ein Drittel des Spielfeldes filmten. Hier wurde zunächst mit einem Beispielobjekt gearbeitet, das ein Material zugewiesen bekam. Dies vereinfachte die Entwicklung, da so erkannt werden konnte, ob die Kameras richtig ausgerichtet waren und dann auch entsprechend richtig auf die GUI gerichtet wurden. Die drei Kameras mussten jedoch von Hand auf die Szene eingestellt werden, so dass es auch hier wieder zu Ungenauigkeiten hätte kommen können. Andererseits wäre man hier flexibler beim Ausrichten der Szene auf die drei Beamer gewesen, indem man zum Beispiel die Übergänge zwischen den drei Texturen jeweils so hätte platzieren können, dass sie auf einem Spalt zwischen den Feldern des Spielfeldes gelegen wären. Auch dieser Ansatz wurde umgesetzt und hat funktioniert allerdings auch mit kleinen Ungenauigkeiten.

Beide Ansätze waren jedoch nicht zu 100 Prozent exakt, was das Aufteilen der Szene in drei Teil-Texturen betraf. Außerdem standen keine drei identischen Beamer zur Verfügung, sodass es unter den einzelnen Beamern Unterschiede bei der Lichtintensität gab. Da anstatt der Lösung mit den drei Beamern letztlich doch noch ein geeignetes Gerät gefunden wurde, das alleine die gewünschte Projektionsfläche erbrachte, wurde die Idee mit den drei Beamern sowie die bisherige Implementierung dafür verworfen und ist im finalen Stand des Projektes nicht mehr enthalten.

2.5 Die grafische Oberfläche (GUI)

Teil der grafischen Oberfläche ist das Spielfeld (Abb. 6), mit dem die Nutzer wie im Konzept beschrieben, interagieren kann. Es stellt nicht nur die einzelnen Töne als quadratische Flächen dar, sondern dient auch als Informationsfeld (siehe auch Abschnitt 2.7). Die Tiles können ihre Farbe wechseln und haben verschiedene Animationen. Näheres hierzu in Abschnitt 2.6.

Neben dem Spielfeld wurde für die Personenerkennung der BlinkenTiles-Anwendung eine weitere Bedienoberfläche entwickelt (siehe Abb. 7). Sie liegt auf dem linken Teil der gesamten GUI und kann so nur auf dem PC-Bildschirm eingesehen werden, während der rechte Bildschirm das Spielfeld darstellt und vom Beamer auf den Boden projiziert wird. Die Einstellungen ermöglichen u. a. die Kalibrierung der Kinect und die Abstimmung mit der tatsächlichen Spielfeldgröße. Näheres hierzu findet sich in Abschnitt 2.8.

Eine dritte grafische Oberfläche wurde für den Presentation-Screen entwickelt, siehe hierzu Abschnitt 2.7.

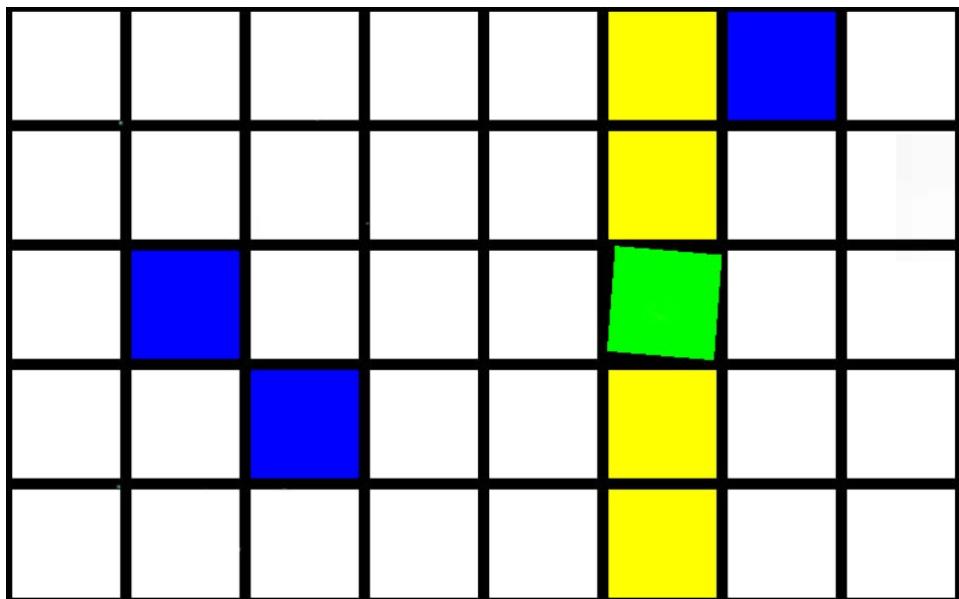


Abbildung 6: Das Spielfeld

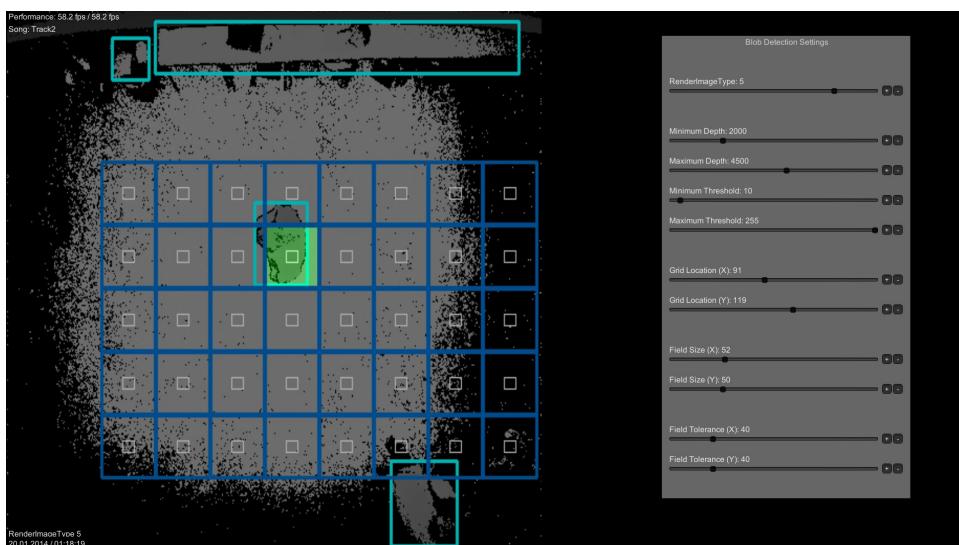


Abbildung 7: GUI für die Bildverarbeitung

2.6 Spielfeld und Felder

Alexander Scheurer

Das Spielfeld und die Felder werden von zwei Klassen bestimmt, die *TileController*- und *TileBehaviour*-Klasse. Erstere ist die zentrale Klasse. Sie beinhaltet die Spiellogik, baut das Spielfeld auf und verarbeitet Eingaben (von Personenerkennung, Maus, Tastatur, Konfiguration und Idle Mode).

Die Spiellogik umfasst überwiegend das Timing der Abläufe und in Abhängigkeit davon das Zuweisen von Zuständen an die einzelnen Felder. Die Zustände, die zugewiesen werden, beschreiben z.B. ob auf diesem eine Person steht, ob sich der Zeitindex derzeit in der Reihe dieses Feldes befindet oder ob sogar beides genannte zutrifft. Dies hätte zur Folge, dass auch der entsprechende Ton zu dem Feld abgespielt werden muss. Die *TileController*-Klasse weist den Feldern den entsprechenden Zustand sequenziell zu:

1. Alle Felder werden zurückgesetzt (*Clear pass*)
2. Der Zustand für Zeitimpuls wird bei den entsprechenden Feldern gesetzt (*Timer pass*)
3. Der Zustand für Feldern auf denen Personen erkannt wurden, werden gesetzt (*People pass*)
4. Der vereinigte Zustand von 2. und 3. wird gesetzt (*Shake'n'Play pass*)

Verarbeitet wird dieser Zustand von jedem Feld selbst mit Hilfe der *TileBehaviour*-Klasse. In dieser passiert die Umsetzung von Zustand zu Aktion, also welcher Shader (Farbe) für das Feld gesetzt wird oder ob es wackeln soll, weil es den entsprechenden Zustand vom 4. Schritt zugewiesen bekommen hat.

Zum Start des Spiels hat der *TileController* die Aufgabe das Spielfeld aufzubauen, dies macht er anhand von JSON-Konfigurationsdateien einmal für die Grundkonfiguration als auch Konfigurationen für jeden Song. Die Grundkonfiguration definiert unter anderem welche Ausmaße das Spielfeld hat, oder auch welche Farbe einem bestimmten Feldzustand zugewiesen ist. Die Song-Konfiguration beinhaltet zum Beispiel die Angabe über die Beats pro Minute und die Pfade zu den Audiodateien.

Benutzereingaben nimmt der *TileController* entgegen und veranlasst die entsprechende Aktion. Zum Beispiel einen Songwechsel oder das verarbeiten eines Feldzustandes von der Personenerkennung oder dem Idle-Node.

2.7 Call to Action

Sarah Häfele

Die Vermutung, dass bei einer Installation mit Musik und der Notwendigkeit, sich zu bewegen, die Hemmschwelle diese zu nutzen höher liegt, liegt nahe. Neben Lichteffekten durch die Scheinwerfer und dem offiziellen Spielfeld dienen deswegen weitere Komponenten zur *Call to Action*, also zur Motivation der Besucher. Diese sollen nicht nur zum Mitmachen bewegen, sondern auch gleichzeitig anleitend wirken.

2.7.1 Der Presentation-Screen

Der Presentation-Screen ist als eine Art Präsentationsfläche gedacht, die den Nutzern weitere Informationen zur Installation liefern soll. In der Spieleengine Unity mit C# entwickelt, kann sie als eigenständige Anwendung auf einem großen Bildschirm neben dem Spielfeld angezeigt werden. Die Anwendung besteht bislang aus vier Modi, angefangen mit dem Startmenü, welches das Logo der Installation und weitere Menüpunkte anzeigt. Der *How-To*-Modus zeigt ein Video welches kurz erklärt, wie das Spielprinzip funktioniert, erstellt von Fabian Gärtner in Adobe Premiere mit Grafiken von Sandra Beuck. Im *Behind-the-Scenes*-Modus wird ein aktuelles Bild der Kinect sowie das Beamerbild gezeigt um zu erklären, wie die Technik dahinter funktioniert. Die Anwendung kommuniziert dabei über das Internet mit der BlinkenTiles-Anwendung. Zuletzt zeigt der *Time-Lapse*-Modus ein Video des Aufbaus der Installation, aufgenommen mit einer GoPro Kamera und geschnitten von Fabian Gärtner.

Die Modi können in einem Loop abgespielt werden, wobei nach einiger Zeit oder wenn das Video zu Ende ist das Programm in den nächsten Modus springt. Wenn die Hauptanwendung von BlinkenTiles jedoch in den Challenge-Modus wechselt, verändert sich auch der Presentation-Screen: er springt zum *How-To*-Video und zeigt zudem Information zum abgespielten Lied an. Dies beinhaltet den Musiktitel und einen Countdown (siehe Abbildung Abbildung 8).

Die BlinkenTiles-Anwendung kommuniziert mit einem XML-File über das Internet mit dem Presentation-Programm. Die Hauptanwendung erstellt das File und legt es in einem Netzwerkordner ab. Dieser Teil wurde von Linda Schey implementiert. Die Presentation-Anwendung greift auf das File zu, parst die Textinformationen und speichert sie in entsprechenden Variablen. Abbildung Abbildung 9 zeigt ein minimales Beispiel, welche Informationen so an den



Abbildung 8: Presentation-Screen im Challenge Mode

Presentation-Screen übermittelt werden können.

```
<?xml version="1.0"?>
<NetworkSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ChallengeMode>true</ChallengeMode>
  <DemoTime>16</DemoTime>
  <Song>Track1</Song>
  <Length>42</Length>
</NetworkSet>
```

Abbildung 9: XML-File mit aktuellen Informationen für den Presentation-Screen

Die Variablen werden dann an die einzelnen Codeelemente weitergegeben und für die Anzeigen verwendet. Der Countdown wird durch die übermittelte Länge des Liedes ermittelt und dann selbstständig dekrementiert. Endet der Challenge Mode, wird der Loop normal fortgesetzt.

2.7.2 Spielfeld-Angaben

Das Spielfeld, welches durch den Beamer auf den Boden projiziert wird, bietet ebenfalls eine gute Möglichkeit, direkt mit den Akteuren zu kommunizieren. Wenn der Challenge Mode startet, wird das Feld deswegen komplett weiß, eine Textinformation zeigt den Modus an und ein Countdown zählt die Beats zum

Start herunter.

Eine Boolean-Variable checkt außerdem noch, ob sich gerade Personen auf dem Feld befinden. Wenn dies nicht der Fall ist, wird nach einer gewissen, konfigurierbaren Zeit der so genannte *Idle Mode* gestartet. Um die Besucher zu motivieren, die Spielfläche zu betreten und um anzuzeigen, wie die Installation funktioniert, werden dabei Fußstapfen auf zufälligen Feldern verteilt. Nach einer gewissen Zeit ändern sie ihre Position. Die Fußstapfen sind Texturänderungen des Materials der Felder per Code. Ein mit einem Fußstapfen-Material belegtes Feld gilt als aktiv und wird so behandelt, als würde eine Person darauf stehen. Eine *Update*-Methode überprüft regelmäßig, ob sich jemand auf das Spielfeld begibt. Ist dies der Fall, endet der Idle Mode sofort und die Fußstapfen verschwinden. Als kleine Auflockerung wurden nicht nur menschliche Fußstapfen verwendet, sondern auch die von Dinosauriern und Hunden (siehe Abbildung Abbildung 10).

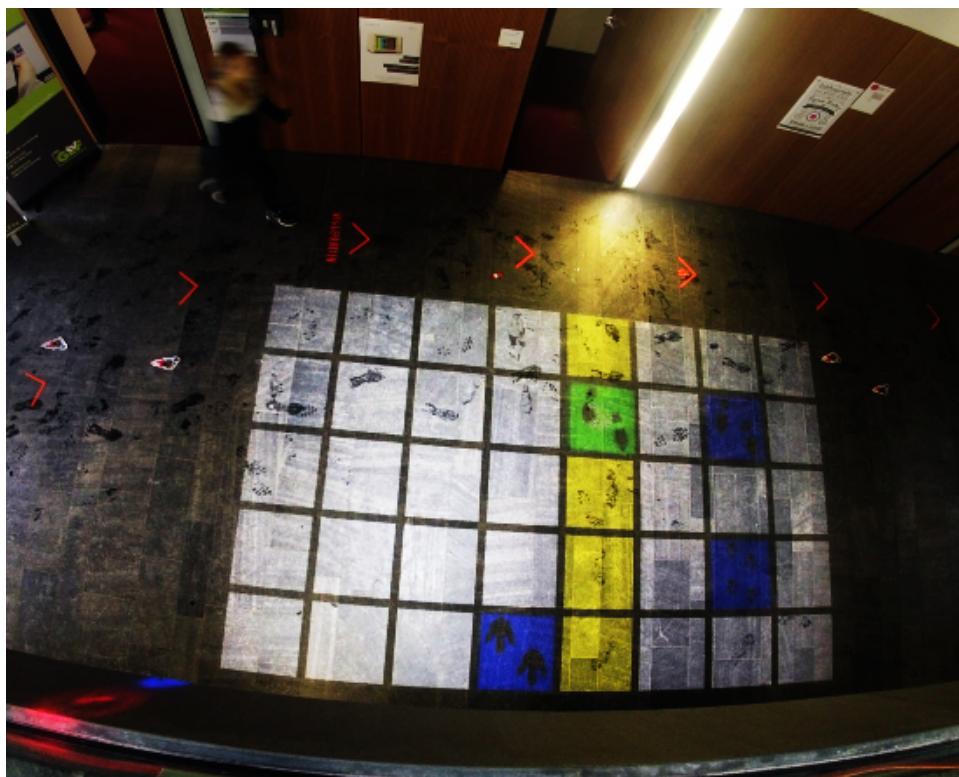


Abbildung 10: Idle Modus mit Fußstapfen

2.8 Personenerkennung mittels Microsoft Kinect

Fabian Gärtner

Zur Erkennung von Personen auf dem Spielfeld mittels Microsoft Kinect konnten keine Standardfunktionen verwendet werden, da das Kinect-SDK den Einsatz der Kinect aus der Vogelperspektive nicht vorsieht. Stattdessen muss das von der Kinect aufgenommene Tiefenbild mittels Bildverarbeitungstechniken auf Personen bzw. verallgemeinert gesprochen auf Objektkonturen hin untersucht werden. In BlinkenTiles sind dafür die drei Klassen *KinectManager*, *BlobDetection* und *BlobDetectionThread* zuständig. Erstere fragt unter Anwendung des Kinect-SDKs von Microsoft das Tiefenbild (30 Bilder pro Sekunde mit einer Auflösung von 512 x 424 Pixel) und – bei Bedarf – das Farbbild (15 Bilder pro Sekunde mit einer Auflösung von 1920 x 1080 Pixel) der Kinect ab. Das Tiefenbild ist dabei ein zweidimensionales Array, das mit Werten zwischen 0 und 8000 die pro Pixel gemessene Tiefe in Millimeter angibt. Da sich die Farbkamera von der Tiefenkamera in Auflösung und Bildwinkel unterscheidet, müssen die Werte des Farbbildes, um beide Bilder übereinander legen zu können, mittels einer vom Kinect-SDK bereitgestellten Funktion umgerechnet werden. Auch diese Funktionalität und die Möglichkeit, Tiefenbilder in Form von Dateien abzuspeichern und für den Fall, dass die Kinect nicht angeschlossen ist, als Testdaten zu verwenden, stellt die Klasse bereit.

Die Weiterverarbeitung des Tiefenbildes und damit die eigentliche Bildverarbeitung findet in der Klasse *BlobDetectionThread* statt und wird zur Laufzeit in einem eigenen Thread ausgeführt. Dies war zu Beginn der Entwicklung noch notwendig, da die anfangs prozessorlastige Objekterkennung zu einer niedrigen Bildrate in der Hauptanwendung führte und damit auch Auswirkungen auf den Spielablauf und das Abspielen der Musik hatte. Dieser Prozess konnte im Verlauf der Entwicklung aber stetig optimiert werden, sodass die Bildverarbeitung auf eine Dauer von lediglich 12 Millisekunden pro Tiefenbild reduziert werden konnte. Mit etwas mehr als 80 Bildern pro Sekunde liegt die Objekterkennung damit bereits weit über der Bildrate der Kinect mit rund 30 Bildern pro Sekunde und kann ohne Verzögerung und in Echtzeit stattfinden. Die Auslagerung in einen eigenen Thread wurde aber beibehalten, sodass die Arbeitslast auch weiterhin gleichmäßig auf mehrere CPU-Kerne verteilt wird und mögliche Verzögerungen bei der Abfrage der Tiefenbilder keine Auswirkungen auf die Hauptanwendung und den Spielablauf haben.

Hauptsächlich für die Bildverarbeitung verwendet wird die kostenfreie Open-Source-Bibliothek OpenCV, die über den C#-Wrapper EmguCV angesteuert wird. OpenCV erlaubt die Anwendung von Algorithmen und grundlegenden mathematischen Operationen auf Matrizen und Bilder. Konkret wird bei BlinkenTiles zunächst das als Array vorliegende Bild gefiltert und normalisiert. Die Filterung arbeitet anhand vom Nutzer eingestellter Werte für minimale und maximale Tiefe. Alle Werte über und alle Werte unter diesen Vorgaben werden auf 0 gesetzt, sodass diese die Erkennung nicht weiter stören. In der Praxis kann damit ein stark reflektierender Boden herausgefiltert werden. Die anschließende Normalisierung hilft, den Kontrast des Tiefenbildes zu erhöhen und so die dann folgende Binarisierung des Bildes zu optimieren. Auf das binarisierte Bild wird letztlich die Konturerkennung von OpenCV angewendet, die Objekte ab einer bestimmten Größe erkennt und ein entsprechendes Konturrechteck zurückliefert. Dieses Rechteck kann anschließend mit den Feldern der Matrix abgeglichen werden und bei Überschneidungen zwischen Konturrechteck und einem Feld das Feld aktiviert werden. Jeder dieser Schritte ist vom Nutzer über eine im Folgenden näher erläuterte GUI separat darstellbar und konfigurierbar, um eine optimale Objekterkennung zu ermöglichen. Des Weiteren werden regelmäßig die einzelnen Schritte als Bilder herausgespeichert, sodass sie den Spielern über den Präsentationsbildschirm angezeigt werden können.

Die Klasse *BlobDetection* schließlich ist hauptsächlich für die Verwaltung der grafischen Benutzeroberfläche (GUI) der Bildverarbeitung und die Verwaltung der eben beschriebenen Klassen zuständig. Dieser Teil der GUI, wie er in Abbildung X auf Seite X dargestellt ist, nimmt die linke Hälfte der gesamten GUI von BlinkenTiles ein und erstreckt sich damit über einen Monitor. Die GUI stellt das Tiefenbild in seiner ursprünglichen Form oder nach einem der zuvor beschriebenen Verarbeitungsschritte, bspw. in binarierter Form oder mit farbigen Konturrechtecken um erkannte Objekte, dar. Des Weiteren erlaubt die GUI über eine Reihe von Slidern die Einstellung von Filterwerten, Spielfeld- und Feldergröße sowie Spielfeldposition. Ebenso kann ein Raster über dem Tiefenbild dargestellt werden, bei dem alle aktivieren Felder, also alle Felder auf denen Personen erkannt wurden, farbig markiert sind. Um dieses Raster dem vom Beamer erzeugten Spielfeld anzupassen, also um Beamer und Kinect zu synchronisieren, besteht die Möglichkeit eine Überlagerung von Farb- und Tiefenbild anzuzeigen, die dann die Anpassung der Größe und Ausrichtung des Rasters auf dem Tiefenbild mittels der Slider erlaubt. Alle Einstellungen können auch zuvor über die externe Konfigurationsdatei geladen werden.

2.8.1 Aufgetretene Probleme

Am Tag der Medien zeigte sich, dass die Tische, auf denen u. a. die LED-Scheinwerfer standen, aufgrund ihrer hellen Oberfläche nicht aus dem Tiefenbild herausgefiltert werden konnten und daher ebenfalls als Objekte erkannt wurden. Dies führte dazu, dass Personen, die auf einem der Felder in der Nähe der Tische standen, zusammen mit den Tischen als ein großes Objekt erkannt und dadurch gelegentlich zu viele Felder gleichzeitig aktiviert wurden. Beheben ließe sich das Problem, indem das Tiefenbild vor der weiteren Verarbeitung auf den relevanten Bereich, also auf Spielfeldgröße zugeschnitten wird. Auch hierfür könnten entsprechende Regler in die GUI integriert werden, sodass der Zuschnitt manuell und je nach Bedarf eingestellt werden kann.

Ein weiteres Problem ist die Perspektive, vor allem dann, wenn die Kinect nicht mittig über dem Spielfeld hängt. So funktioniert die Erkennung von Personen, die sich direkt unter der Kinect befinden zwar tadellos, je weiter eine Person aber von der Kinect entfernt ist, desto mehr Felder deckt sie aufgrund der schrägen Aufsicht gleichzeitig ab. Eine kurzfristige Lösung am Tag der Medien war es, die Felder in der Anwendung größer als in der Realität einzustellen. Des Weiteren wird bereits nicht das ganze Feld auf Überschneidungen mit Konturen überprüft, sondern lediglich ein kleinerer (ebenfalls einstellbarer) Bereich in der Mitte der Felder. Um das Problem aber besser zu lösen, wäre entweder eine zweite Kinect oder eine perspektivisch korrekte Berechnung und ein unsymmetrisches Raster notwendig. So könnten bspw. die Felder des Rasters in den Randbereichen vergrößert oder die Tiefenfilterung zu den Rändern hin verstärkt werden, sodass in diesen Bereichen nur noch die Beine von Personen für die Überschneidung relevant sind, nicht aber der Kopf, der möglicherweise ein anderes Felder überdeckt. Welche Methode sich hier am besten eignet, müsste durch weitere Tests herausgefunden werden.

Störend war auch, dass die Einstellungen zwar wie beschrieben aus einer Konfigurationsdatei geladen werden können, die Werte aber momentan noch per Hand eingetragen werden müssen. Interessant neben einem Button zum Speichern der Einstellungen könnte für eine solche Installation daher auch eine automatische Kalibrierung sein, um bei einem Neustart die manuelle Neukonfiguration zu vermeiden. Technisch ließe sich das über ein vom Beamer dargestelltes Testmuster lösen, das von der Bildverarbeitung erkannt wird und das Raster entsprechend dimensionieren und positionieren lässt. Auch die optimalen Werte für die Filterung könnten so automatisiert ermittelt werden.

2.9 Audiogestaltung und Inspiration

Johannes Winter

Das Konzept der Installation beruht auf einem klassischen Step-Sequenzer. Solche Sequenzer steuern die Klangerzeugung eines Synthesizers dahingehend, dass sowohl Rhythmus als auch Tonhöhe programmiert werden können. Der Name bezieht sich dabei auf die einzelnen »Steps« die mit Tönen belegt werden können. Klassische analoge Step-Sequenzer wie der Roland TB 303 bieten 16 Schritte an, womit also pro Durchlauf 16 Töne gespielt werden können. Somit entspricht jeder Schritt einer 16tel Note eines Taktes. Es ist hervorzuheben, dass bei einem Step-Sequenzer die Noteneingabe nicht unmittelbar zu einer Klangerzeugung führt. Stattdessen tastet ein Impuls nacheinander alle „Steps“ ab und übermittelt die Daten an den Klangerzeuger. Nachdem der Impuls einmal durchgelaufen ist, beginnt der er wieder bei dem ersten Step. Dadurch entstehen repetitive Tonfolgen, sogenannte Loops. Step-Sequenzer haben gerade aufgrund dieser Beschränkung zahlreiche Genre wie Acid House, Electronic Body Music und Drum and Bass entscheidend geprägt.

Das Konzept der Installation weicht in mehreren Punkten vom klassischen Step-Sequenzer ab. Zum Einen gibt es statt 16 Schritten nur 8. Außerdem ist die Tonauswahl auf 5 Tonhöhen pro Schritt begrenzt. Des Weiteren wird ein Ton nur so lange aktiviert, wie auch eine Person auf dem entsprechenden Feld steht. Das musikalische Konzept berücksichtigt diese Einschränkungen. Da nur 5 Töne gespielt werden können und auch nur 8 Steps zur Verfügung stehen, wird ein Backing Track benötigt, der eine harmonische Grundlage für das Spiel des Instrumentes bietet.

Anspruch der Installation war es aber weniger ein komplexes Instrument zu bieten, viel mehr eine musikalische Spielwiese. Aus diesem Grund liegt die Entscheidung nahe, sich die technischen Beschränkungen zu Nutze zu machen. Es wurde also auf eine Skala zurückgegriffen, die einerseits nur 5 Töne hat und andererseits keine Halbtorschritte aufweist: die Anhemitonische Pentatonik. Der Vorteil liegt darin, dass keine kleinen Sekunden und Tritoni gespielt werden können, die für unsere Hörgewohnheiten »unrein« klingen. Durch diese Maßnahme wurde also sichergestellt, dass unabhängig von Menge und Position der Benutzer ein recht harmonisches Gesamtbild entsteht, auch wenn dadurch auf die Leittonwirkung einer Diatonik oder Hemitonischen Pentatonik verzichtet werden muss. Außerdem war der Tonumfang natürlich auf eine Oktave be-

schränkt, mehr als 5 Tonhöhen hätten das musikalische Ergebnis interessanter gestaltet aber auch die Nutzerfreundlichkeit eingeschränkt. Da die Installation in erster Linie intuitiv bedienbar sein sollte, wurde der Tonumfang nicht weiter ausgebaut.

Für den experimentellen Modus der Installation wurden drei Backing Tracks mit jeweils 5 zugehörigen Tonhöhen vorproduziert. Der durchlaufende Impuls wurde auf die Geschwindigkeit des entsprechenden Backing Tracks angepasst und die zugehörigen Töne auf die Felder gemappt. Da der Klangerzeuger kein Synthesizer sondern ein Sampler war, mussten die Einzeltöne im Voraus synthetisiert und gerendert werden. Grundsätzlich wurden eher sphärische Klänge mit langem Nachhall (entweder Reverb oder Hüllkurvengenerator) eingesetzt, da die Benutzer der Installation keinen Einfluss auf die Länge des Tons hatten. In Hüllkurvenparametern gesprochen musste also der Attack deutlich hörbar sein um ein auditives Feedback zu geben, die Sustainlautstärke musste relativ schnell erreicht werden und wesentlich leiser als der Attack sein, damit sich bei mehreren Personen die Sounds nicht zu stark überdeckten. Dadurch wurden relativ perkussive Klänge erzeugt, die durch den Nachhall eine gewisse Stetigkeit erreichten. Die Backing Tracks bildeten das rhythmische und harmonische Grundgerüst und wurden ebenfalls im Voraus produziert und gerendert. Dabei wurde Wert darauf gelegt, zwar eine harmonische Orientierung zu bieten, der Backing Track sollte jedoch keine komplexe harmonische Struktur aufweisen. In den Backing Tracks wurde also ebenfalls weitgehend auf die oben erwähnte Pentatonik zurückgegriffen, um aber ein wenig musikalische Spannung zu kreieren, wurden an ausgewählten Stellen auch Töne aus diatonischen Skalen verwendet.

Für den Challenge Modus wurden ebenfalls drei Backing Tracks und passende Töne vorproduziert, da jedoch die technischen und konzeptionellen Bedingungen andere waren, soll darauf noch näher eingegangen werden. Das Ziel des Challenge Modus ist es, mit der Installation bekannte Lieder nachzuspielen, ähnlich wie bei Guitar Hero. Problematisch ist jedoch, dass die Reaktionsgeschwindigkeit der Benutzer nicht so schnell ist wie es eigentlich nötig wäre. Das größte Hindernis ist jedoch, dass die Nutzer die Installation verlassen müssen, wenn sie keinen Ton spielen wollen. Sobald sie auf irgendeinem Feld stehen und der Impuls dieses erreicht wird der Ton getriggert, auch wenn er falsch ist. Anders gesagt: Die Zustände der Nutzer auf dem Spielfeld sind analog, benötigt wären aber digitale Zustände, an oder aus. Aus diesem Grund mussten Songs gefunden werden, die einerseits recht bekannt sind, ein minimales Tonspektrum

aufweisen und langsam genug sind um mit der Installation spielbar zu sein. Die Entscheidung fiel auf »Smoke on the water« von Deep Purple, »Paint it black« von den Rolling Stones und »One« von Swedish House Mafia.

Die Ausschnitte der Lieder die benutzt werden sollten, wurden zunächst nachgespielt und aufgenommen. Dabei war es wichtig, hervorstechende Klanggeigenschaften der Originale zu berücksichtigen um den Wiedererkennungswert nicht zu verlieren. Bei »Smoke on the water« wurde beispielsweise ein bekannter britischer Röhrenverstärker mit vorgeschaltetem Overdrive verwendet. Nachdem alle Einzelspuren aufgenommen waren musste bestimmt werden, welche Spuren den Backing Track bildeten und welche Spur von den Nutzern gespielt werden sollte. Der Backing Track wurde dann ohne die entsprechende Spur gerendert. Die Spur die der Nutzer spielen sollte, musste nun in kleine Samples geschnitten werden um auf die verschiedenen Felder aufgeteilt werden zu können.

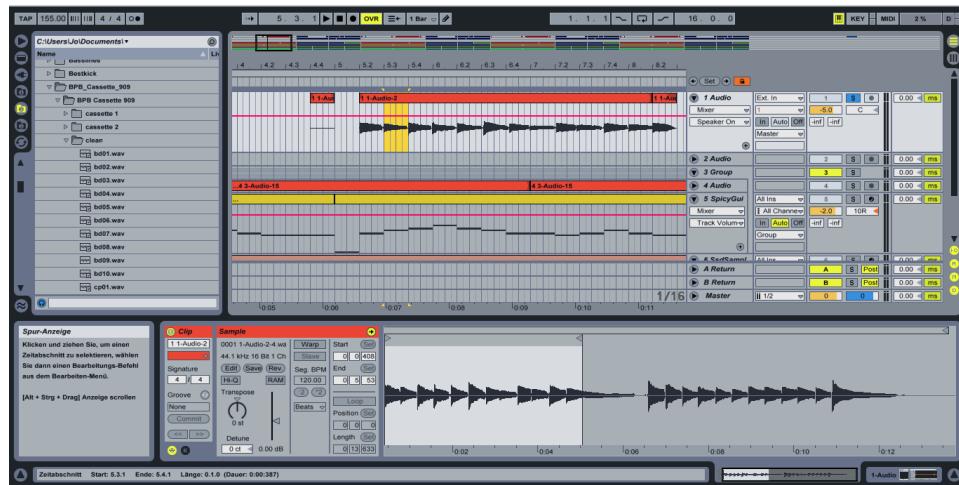


Abbildung 11: Schneiden der Spuren in Ableton Live

Dabei ergaben sich in einigen Fällen Probleme bezüglich der Tonlänge. Da die Schritte des Step-Sequenzers immer auf einen bestimmten Rhythmuswert quantisiert waren, zum Beispiel 8tel oder 16tel-Noten, mussten hinsichtlich Synkopen oft Kompromisse eingegangen werden. Bei »One« führte dies dazu, dass der Impuls so schnell war, dass das Lied damit praktisch unspielbar wurde. Für den Challenge Modus erwies sich damit das Konzept der Soundmatrix insgesamt als relativ ungeeignet, während im experimentellen Modus musikalisch interessante Ergebnisse erreicht wurden.

3 Ergebnisse

3.1 Probleme und Verbesserungen

Sarah Häfele

Nach mehrmaligem Testen wurde der Prototyp am Tag der Medien zum ersten Mal am Publikum getestet. Die Rückmeldung und der Anklang waren positiv und die Installation wurde gut frequentiert. Im ersten richtigen Betrieb fielen jedoch natürlich einige Mängel auf, die verbesserungswürdig sind. Neben den schon von Fabian Gärtner in Abschnitt 2.8.1 aufgezählten Problemen sollen hier weitere Problemfaktoren und Verbesserungsvorschläge skizziert werden.

Der Idle Modus, der startet, wenn keine Besucher auf dem Spielfeld sind, ist nicht eindeutig genug und sollte eventuell durch eine andere Farbe deutlicher gemacht werden. Auch der Challenge Modus hat Farbgebungsprobleme und zeigt so nicht prominent genug, wenn ein neues Lied beginnt. Hier könnte man sich auch dem Presentation-Screen bedienen und sogar akustische Signale verwenden. Ein allgemeines akustisches Feedback würde viele Signale einfacher erkennbar und verständlicher machen.

Die Impulslinie ist von oben gut erkennbar, steht man aber auf dem Spielfeld, ist sie schon schwerer zu sehen und man erkennt vor allem im Challenge Modus nicht richtig, welches Feld als nächstes belegt werden muss. Der Impuls muss besser dargestellt werden, zum Beispiel wie bei einem Ladebalken, der großflächig die Felder abdeckt, die nicht mehr aktuell sind. Dieses Symbol sollte allgemein bei der Zielgruppe bekannt sein und sollte den Impuls prominenter machen. Weitere Verbesserungsmöglichkeiten wären Noten- oder Fußsymbole, die die nächst zu besetzenden Felder markieren.

Durch Hardwareprobleme kam es zu Treiberproblemen, weswegen die Kommunikation über das Internet nicht mehr richtig funktionierte, der Presentation-Screen deshalb schnell umgeschrieben werden musste und die DMX-Steuerung nicht mehr reagierte. Daraufhin wurden die Scheinwerfer vom System abgetrennt, um weitere Systemabstürze zu verhindern.

Zusätzlich zu den aufgetretenen Problemen wurden auch einige andere Verbesserungsvorschläge beim Beobachten der Probanden klar: Der Wunsch nach einem noch kreativeren Weg, Musik zu machen, kam mehrmals auf. Hier wäre ein Modus ohne Impuls und mit freiem Rhythmus sicher spannend, würde aber natürlich ein wenig von dem eigentlichen Konzept abweichen.

Auch bietet die Kinect gute Möglichkeiten für noch mehr Interaktion seitens der Nutzer. So könnte der Impuls mit den Bewegungen der Personen schneller oder die Lautstärke der einzelnen Töne durch ein Handsignal geregelt werden. Interessant sind auch Soundmatritzen, die verschiedene Pulse verwenden, wie z.B. sich zyklisch ausbreitende oder zufallsgesteuerte Impulse.

Bei all den Problemen war die Installation trotzdem ein großer Erfolg. Natürlich war dies auch der guten Position geschuldet, die die Installation gleich beim Betreten des I-Baus in den richtigen Fokus rückte. Zum anderen weckten Musik, Farben, Bewegung und die Herausforderung die Kreativität des doch meist medienaffinen Publikums. Da sich einige für die Installation interessierten, lockten diese auch weitere Personen an. Details wie die Animation der Felder und der Challenge Modus kamen gut an. Der Freestyle Modus, der keine großen Vorgaben hat, war jedoch der beliebteste Modus und wurde bis zum Abbau der Installation begeistert genutzt.

3.2 Nutzergruppen-Analyse

Meike Zöckler

Am Tag der Medien konnten unterschiedliche Nutzertypen bzw. Verhaltensweisen im Umgang mit der Installation beobachtet werden. Diese werden folgendermaßen kategorisiert:

- **Kategorie 1:** Laufen über die Projektion ohne sie eines Blickes zu würdigen.
- **Kategorie 2:** Laufen skeptisch an der Projektion vorbei; wollen nicht im Mittelpunkt stehen oder trauen der Sache nicht.
- **Kategorie 3:** Steht am Rand und diskutiert mit anderen Leuten, was es ist, wie es funktioniert und ob sie das nicht doch mal ausprobieren sollten...?
- **Kategorie 4:** Leute, die es zögerlich ausprobieren und sich dann doch etwas damit beschäftigen, bevor sie weiterhin skeptisch weitergehen
- **Kategorie 5:** Leute, die sich direkt darauf einlassen und dann einige Zeit darauf verbringen und alle Möglichkeiten austesten

- **Kategorie 6:** Leute, die damit spielen; sowohl Challenge als auch Spiel im Freestyle, indem sie mit der Erkennung spielen und Muster laufen oder die versuchen, mit dem Impuls mitzulaufen, die Himmel und Hölle hüpfen, die sich auf den Boden werfen, um möglichst viele Felder gleichzeitig abzudecken, die balancieren und den ganzen Körper verbiegen um neue Muster auszuprobieren, die Ausdruckstanz wagen, die es als Meditation wahrnehmen (besonders mit der sphärischen Standardversion)

Neben der direkten Interaktion mit BlinkenTiles wurden außerdem Fotos aus sämtlichen Perspektiven geschossen, es wurde gefilmt, diese Fotos und Videos geteilt, untereinander und teilweise über Facebook. Für den Challenge Mode wurden aktiv alle umstehenden Personen angesprochen und zum Mitmachen animiert. Dabei spielte es keine Rolle, ob sie sich kannten oder nicht; die Gruppen setzten sich zusammen aus Nutzern jeden Alters, darunter Studierende, Professoren, Mitarbeiter, Externe wie Besucher. Und sehr viele haben sich für die Technik dahinter interessiert und sich diese ausführlich erklären lassen.

3.3 Showreel

Johannes Winter

Eine der Anforderungen der Veranstaltung an die Studenten bestand darin, einen abschließenden Teaser zu erstellen, welcher einen kurzen Überblick über die Installation geben, Impressionen vermitteln und Lust auf die Interaktion mit der Installation machen soll. Das Showreel (siehe Abb. 12 (a)), geschnitten von Fabian Gärtner, Meike Zöckler und Johannes Winter, kann über YouTube⁵ und auf Vimeo⁶ angeschaut werden. Der QR-Code in Abb. 12 (b) führt direkt zum Video auf YouTube.

Das Showreel soll für die Installation begeistern und Lust darauf machen diese selbst auszuprobieren. Dabei steht klar der spielerische, musikalische und gemeinschaftliche Aspekt im Vordergrund. Da die Installation musikalisch geprägt ist, wurde auf der Audioebene bewusst auf Sprachmitschnitte, Umgebungsgeräusche und Sprechertexte verzichtet. Es wurde besonderer Wert auf die Synchronisation der Musik und des Bildes gelegt, um eine bestmögliche Synthese der Sinneseindrücke zu erlangen.

⁵<https://www.youtube.com/watch?v=oIt5Z8fHog0>

⁶<http://vimeo.com/118690388>



(a) Ausschnitt aus dem Showreel



(b) QR-Code zum Teaser

Abbildung 12: Der Teaser zur Installation

Auch das dramaturgische Konzept des Showreels lehnt sich an dem musikalischen Aufbau des verwendeten Stückes an. So sind bei Build-Ups vorwiegend Timelapse Aufnahmen zu sehen, die dann an Punkten des musikalischen Klimax von emotionalen und von Reaktionen geprägten Aufnahmen gebrochen werden. Bei solchen Aufnahmen wird ein breites Spektrum an Userreaktionen und Interaktionsmöglichkeiten gezeigt. Dabei wurden verschiedene Umgangsmöglichkeiten mit der Installation in den Fokus gerückt um die vielfältigen Möglichkeiten aufzuzeigen. Somit war es wichtig zu zeigen, welch unterschiedliche Formen der Joy-of-Use bei der Installation annimmt. Für einige Nutzer lag der Wettstreit mit anderen im Vordergrund, während andere mit der Erfassung verschiedener Gliedmaßen experimentierten. Wieder andere gingen sehr zögerlich und vorsichtig mit der Installation um und waren überrascht über die musikalischen Folgen ihrer Interaktion. Andere Nutzer interagierten eher unbewusst mit der Installation indem sie einfach darüber liefen und on-the-fly interessante Tonfolgen kreierten, die weitere Zuschauer anlockten.

3.4 Vision

Meike Zöckler

Ein großer Mehrwert würde durch den Verzicht auf Beamer und die Traversen entstehen. So könnte man die Installation an beliebigen Orten, zum Beispiel im hellen Sonnenlicht oder in niedrigen Räumen, einsetzen. Optimal wäre für diese Art der Umsetzung eine Art Tanzmatte oder leuchtende Bodenplatten im Stile der 80er Jahre. So könnten außerdem technische Erfassungsprobleme der Kinect und die Schattenbildung durch den Beamer ausgeschlossen werden. Zusätzlich könnte in dieser Ausführung nicht nur auditives und visuelles, sondern auch haptisches Feedback durch Vibration der Platten eingeführt werden.

Der Challenge Mode ließe sich zudem weiter ausbauen. Je nach Anzahl der Spieler und nach Schwierigkeitsgrad könnte es vier verschiedene Level oder Niveaus geben: Easy, Medium, Expert und Boss. Für jede dieser Stufen sollte ein ausreichend großer Pool an Songs zur Verfügung stehen, die per Zufall ausgewählt werden, um genügend Abwechslung zu garantieren. Die Frage ist, wie die Auswahl und der Input gestaltet werden. Wird es eine Konsole geben, erkennt der Boden selbstständig, wie viele Menschen sich darauf befinden und wählt selbstständig ein Level aus?

Alternativ wäre auch eine Challenge im Sinne diverser Tanzspiele wie *Dance Dance Revolution* aus dem Hause *Konami*. Man teilt die Felder in Bereiche oder Zonen auf und die Spieler müssen gegeneinander antreten und auf die vorgegebenen Felder springen. Je nach dem, wie richtig oder falsch, wie schnell man die Felder erreicht hat, bekommt man Punkte. Dazu wäre außerdem eine Anzeigetafel oder ein Screen vonnöten, um den Punktestand darzustellen.

4 Impressionen

Sarah Häfele

Zum Abschluss noch einige Fotos vom Aufbau und vom Tag der Medien. Im Anhang dazu finden sich die technischen Skizzen, die, teilweise veraltet, die Testaufbauten dokumentieren und der Projektgruppe als Planungshilfe dienten.

4.1 Impressionen Tag der Medien

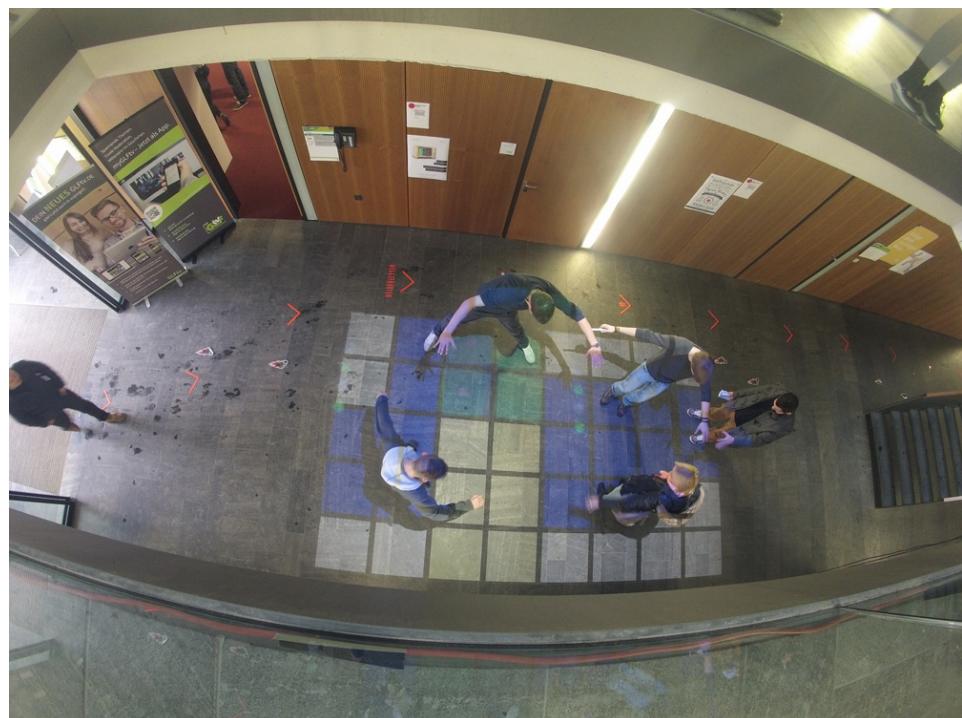


Abbildung 13: Installation von oben

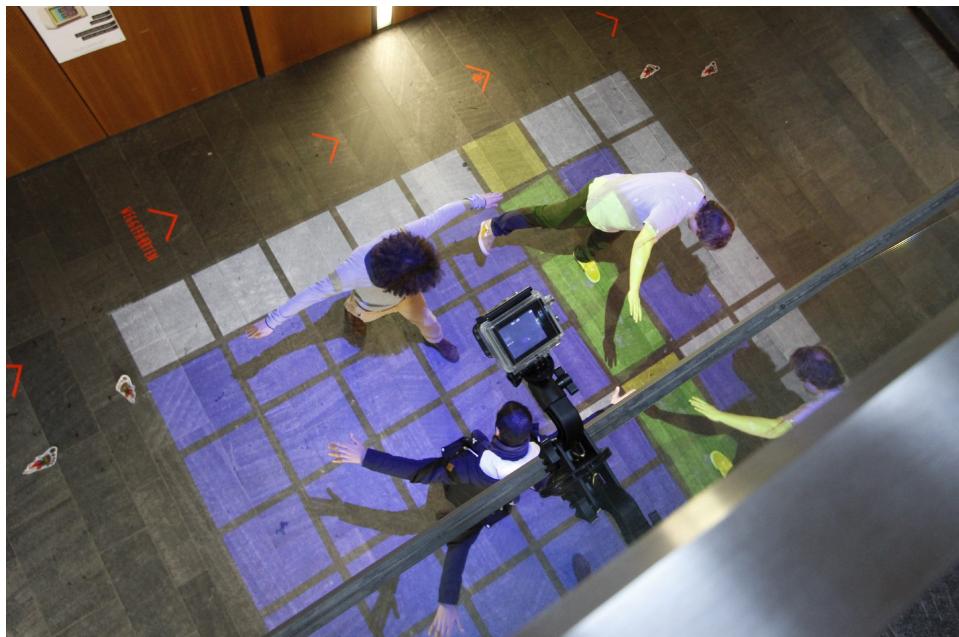


Abbildung 14: Installation im Einsatz (*Foto: GLF TV*)



Abbildung 15: Erdgeschoss (*Foto: Ramazan Gündogdu*)

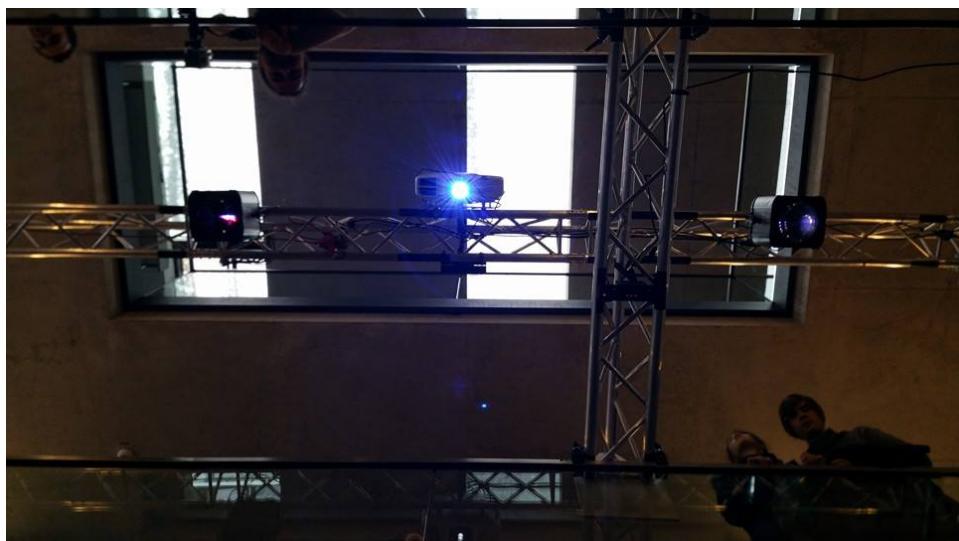


Abbildung 16: Traversenkonstruktion (*Foto: Ramazan Gündogdu*)



Abbildung 17: Erster Stock, Kontrollprogramm (Foto: Ramazan Gündogdu)

4.2 Impressionen Erste Tests

Die Installation wurde schon Wochen vor dem Tag der Medien in verschiedenen Varianten aufgebaut und auch an anderen Orten, wie in der Aula, getestet.



Abbildung 18: Erste Traversentests im I-Bau

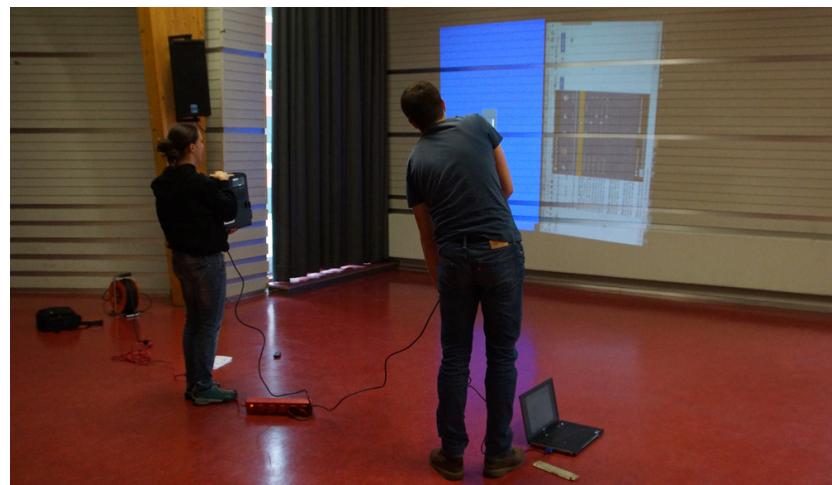


Abbildung 19: Tests mit zwei Beamern in der Aula



Abbildung 20: Tests mit der Kinect in der Aula

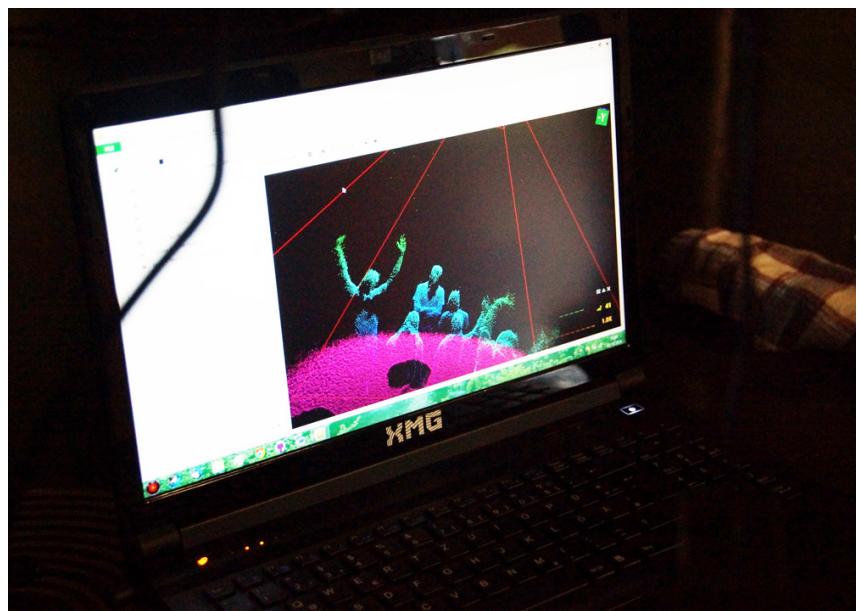


Abbildung 21: Tests mit der Kinect in der Aula

4.3 Technische Zeichnungen

Die folgenden Seiten enthalten unter anderem die handgezeichneten technischen Zeichnungen und Bemaßungen, die während den Tests gemacht wurden.

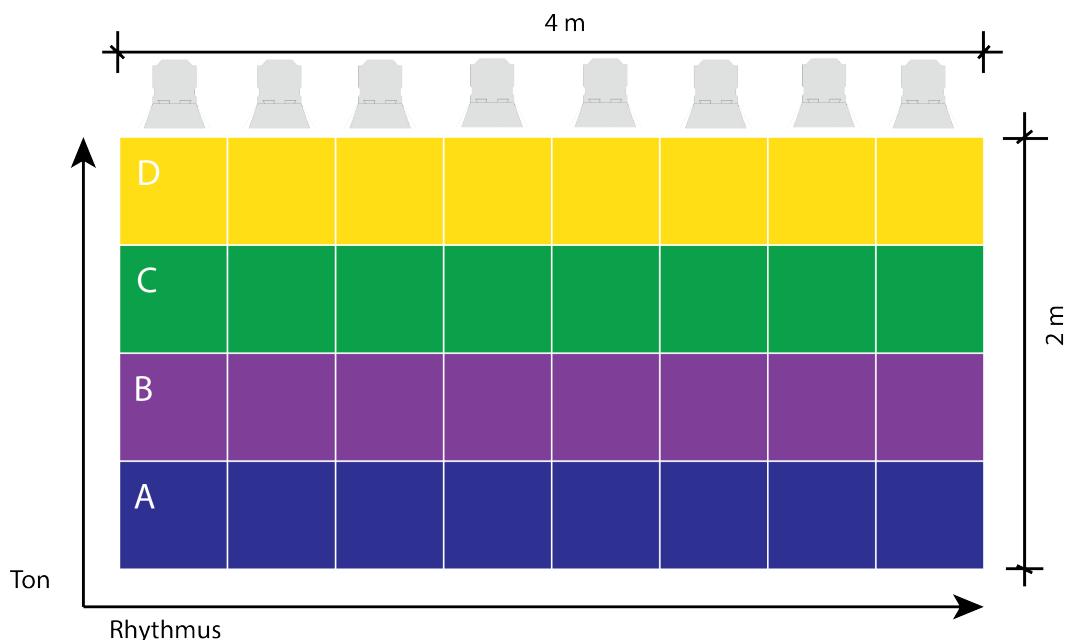


Abbildung 22: Erste Feldbemaßung

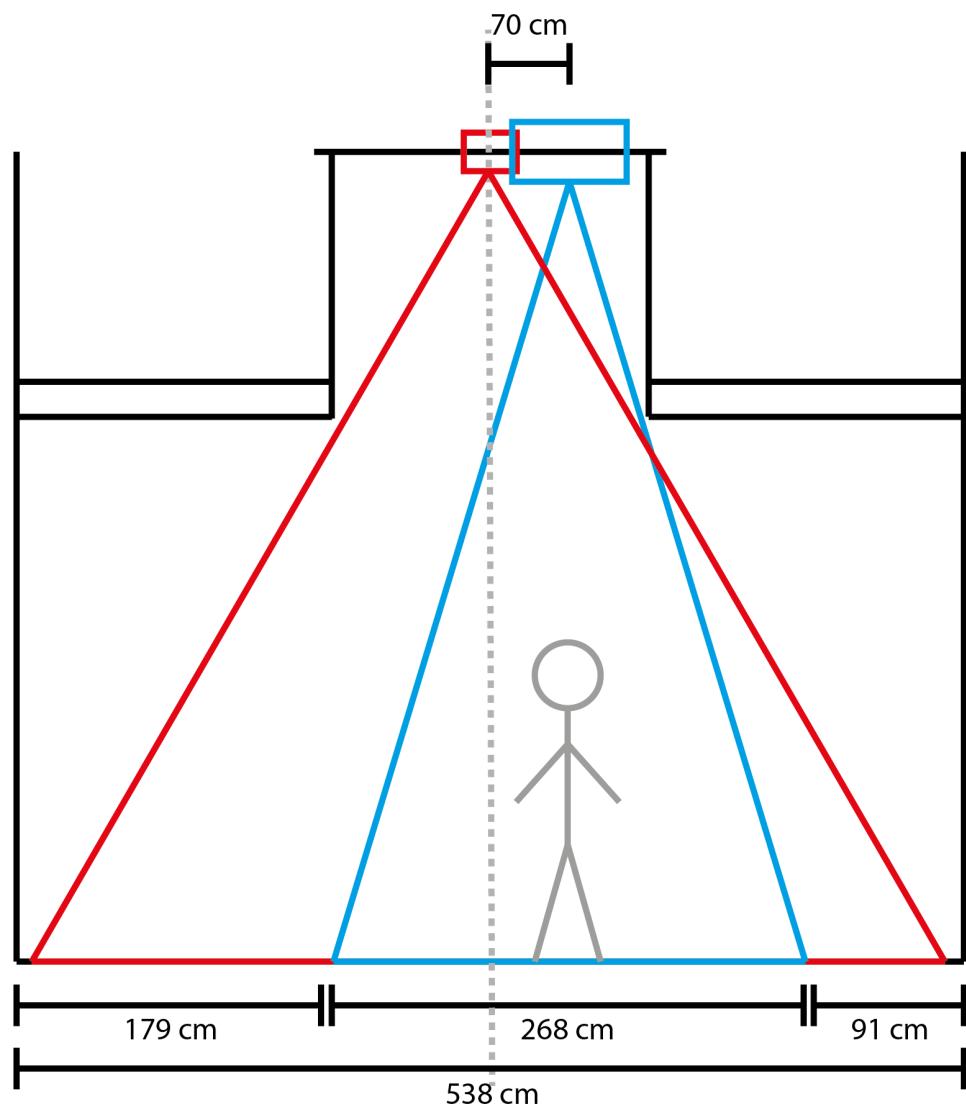
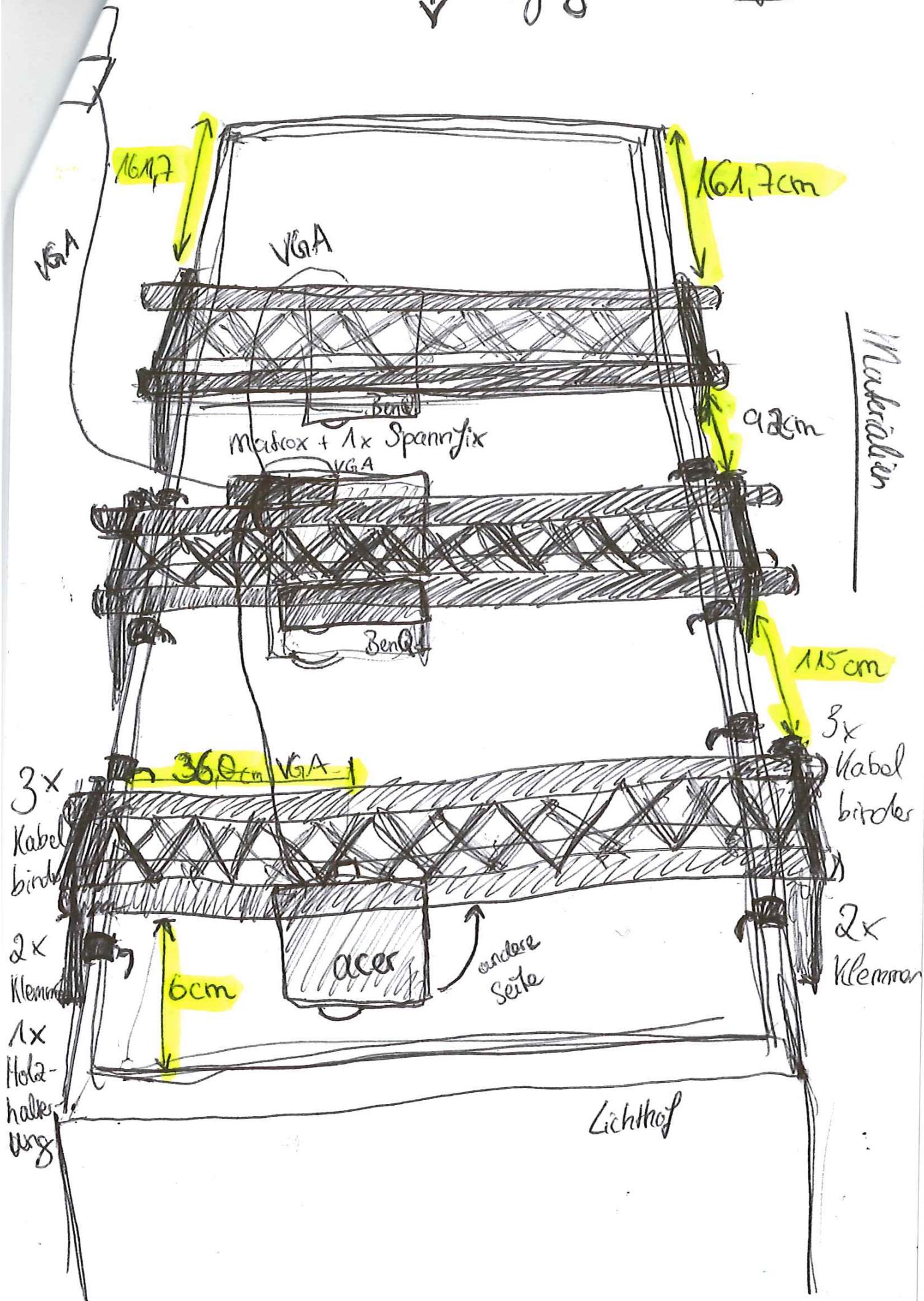


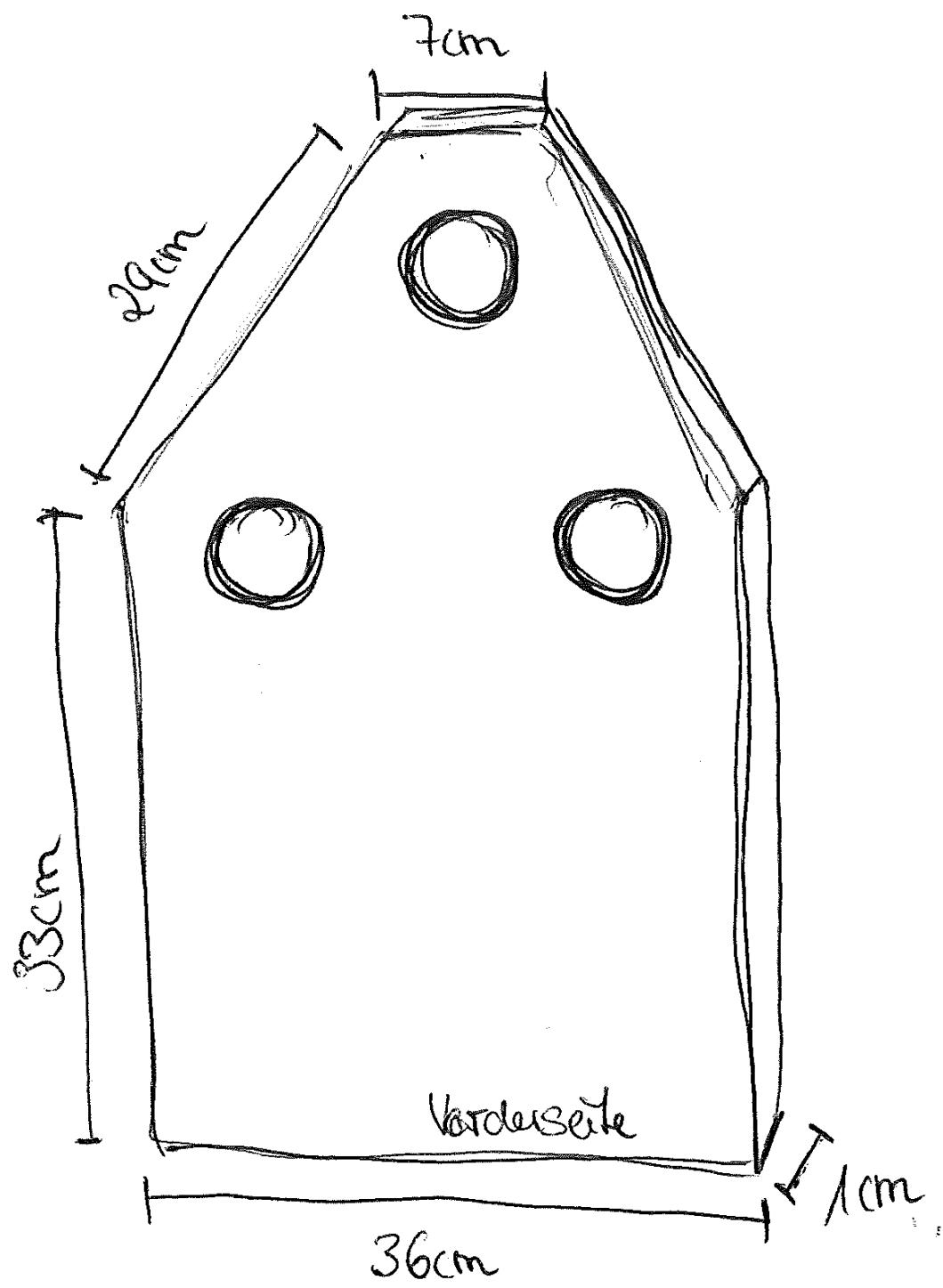
Abbildung 23: Ältere Beamer und Kinect-Bemaßung

↑ Eingang

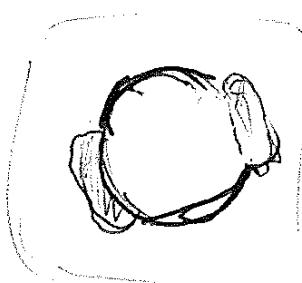
Aufbau 2



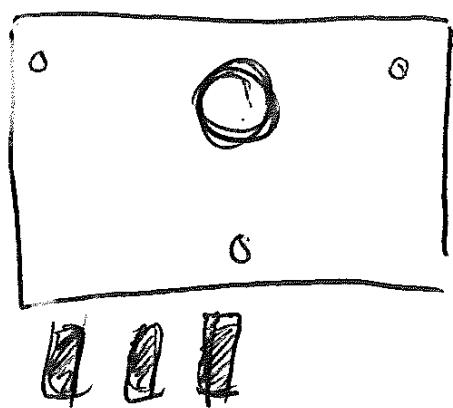
Holzhalterung für Traversen



Rückseite
löcher mit
Einbuchtungen



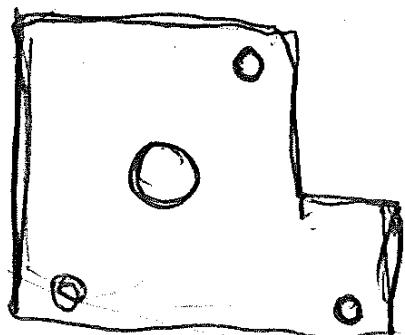
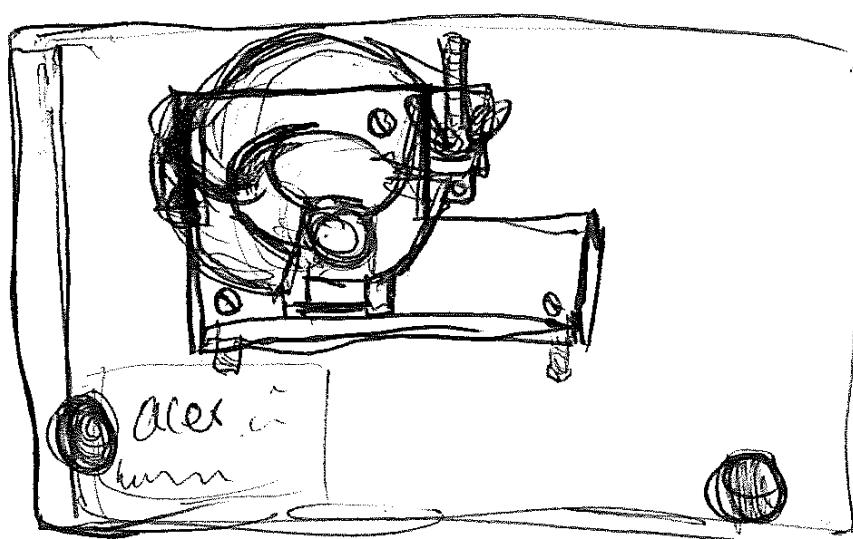
BenQ



• 1x Stromkabel

• 1x langes VGA-Kabel

• 1x Platine

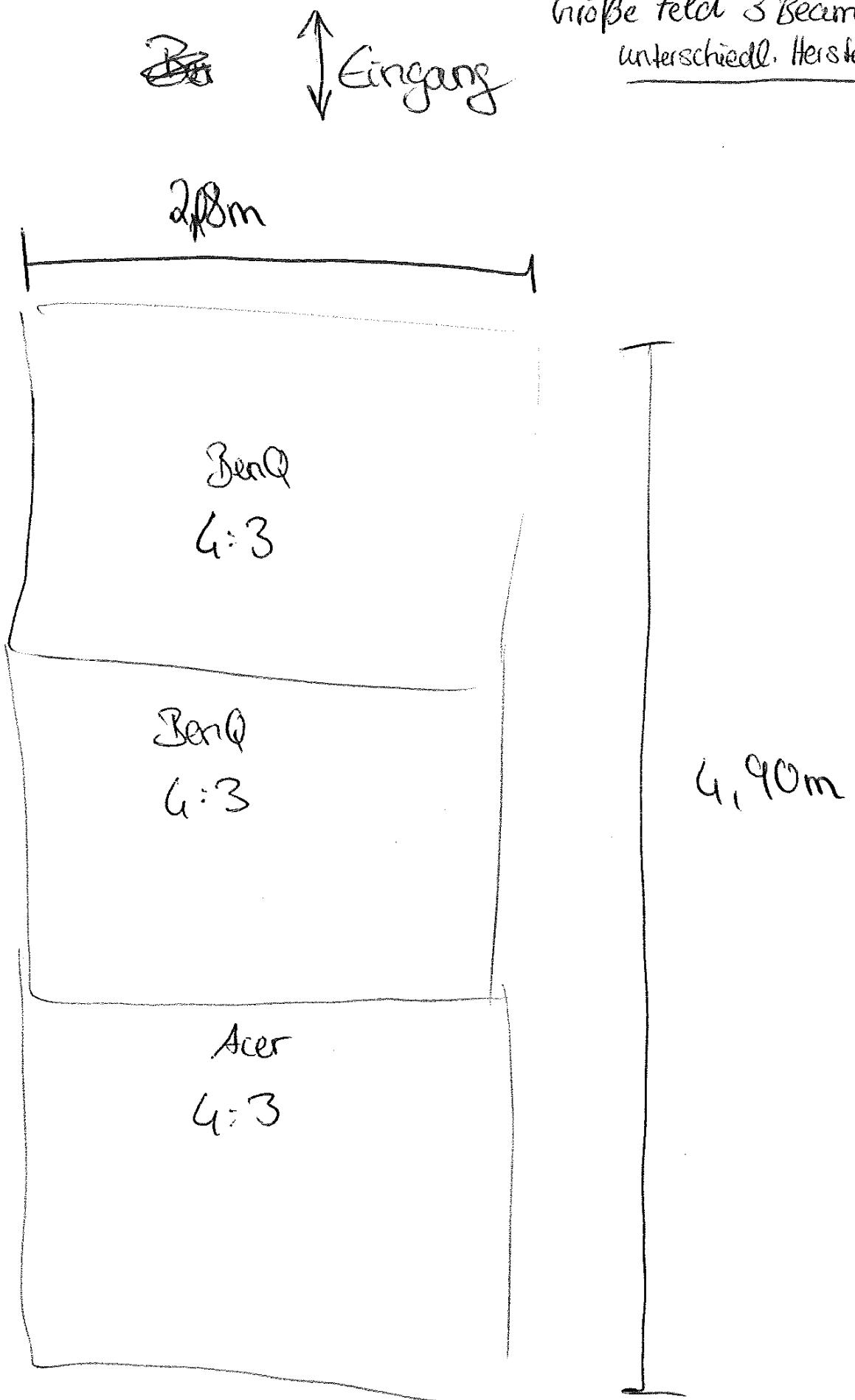


3x Abstandshalter

3dorne Schrauben

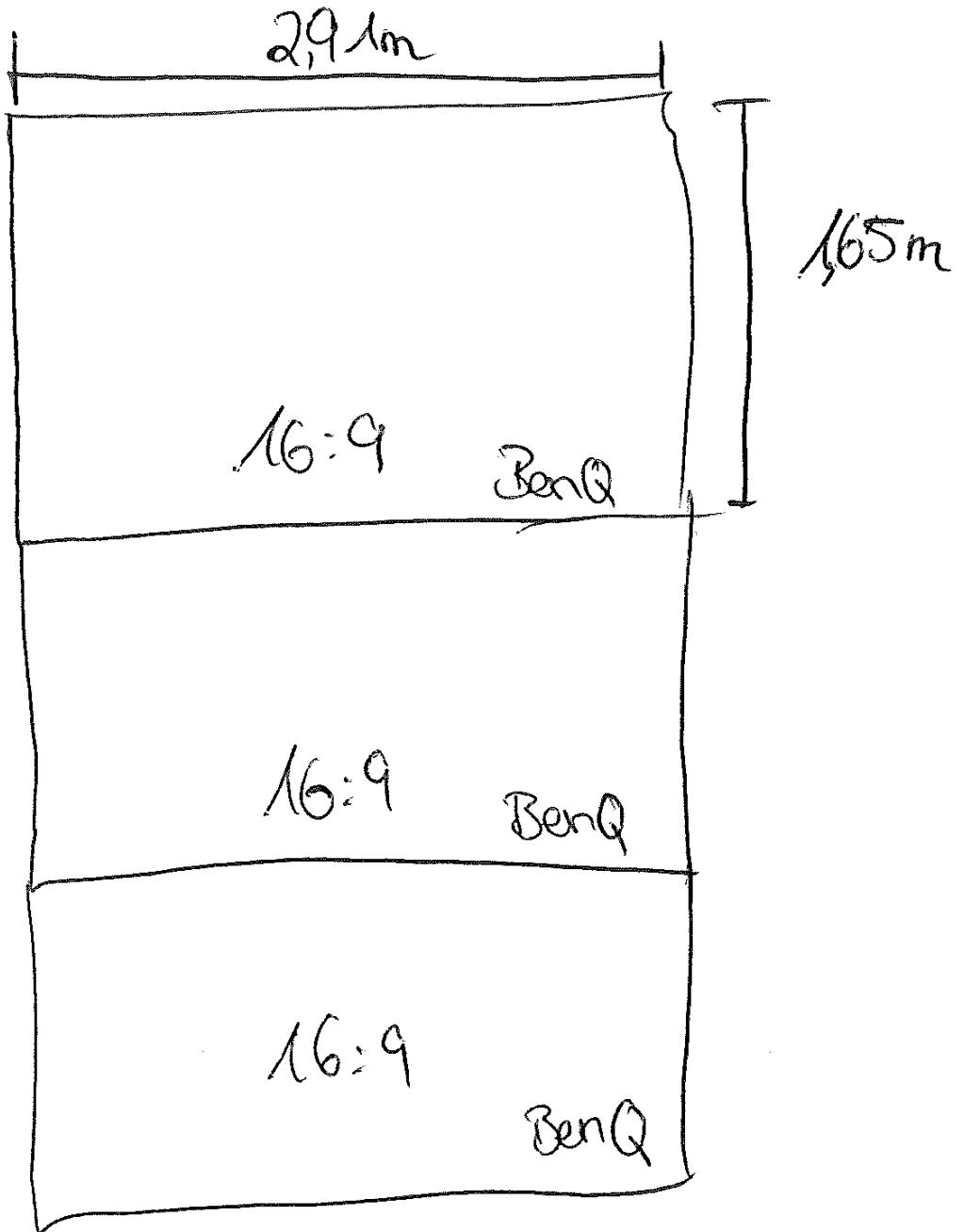
1x Daughtyc Lam

Große Feld 3 Bäume
unterschiedl. Hersteller



Eingang

Große Feld 3 Beamer
gleicher Hersteller



60x60 Felder

Eingang
↓
Aufbau 3

