

Hochschule Furtwangen - Fakultät Digitale Medien

## Interaktionsdesign

Wintersemester 2014/15



Fabian Gärtner, MIM1

Sarah Häfele, MIM1

Alexander Scheurer, MIM1

Linda Schey, MIM2

Johannes Winter, DIM1

Maike Zöckler, DIM1

Prof. Patricia Stolz

Prof. Dr. Matthias Wölfel

3. Februar 2015

## **Abstract**

# Inhaltsverzeichnis

<b>1</b>	<b>Grundidee</b>	<b>4</b>
1.1	Ideenfindung . . . . .	4
1.2	Zielsetzung . . . . .	4
1.3	Werbebotschaft . . . . .	4
1.4	Strategische Planung . . . . .	4
<b>2</b>	<b>Planung und Skizzierung</b>	<b>4</b>
2.1	Zielgruppenanalyse . . . . .	4
2.2	Tests . . . . .	4
2.3	Technischer Aufbau . . . . .	4
<b>3</b>	<b>Umsetzung des Prototypen</b>	<b>4</b>
3.1	Entwicklungs-Entscheidungen . . . . .	4
3.2	Model Aufbau . . . . .	4
3.3	DMX-Scheinwerfer . . . . .	4
3.4	Multimonitor Support . . . . .	6
3.5	Spielfeld und Tiles . . . . .	8
3.6	Call to Action . . . . .	9
3.7	Kinect-Erkennung . . . . .	9
3.8	Audiogestaltung und Inspiration . . . . .	9
<b>4</b>	<b>Tag der Medien</b>	<b>12</b>
4.1	kleiner Bericht mit Fotos . . . . .	12
4.2	Praxiserfahrungen . . . . .	12
<b>5</b>	<b>Showreel</b>	<b>12</b>
<b>6</b>	<b>Fazit, mögl. Weiterentwicklungsmöglichkeiten</b>	<b>12</b>

# **1 Grundidee**

## **1.1 Ideenfindung**

## **1.2 Zielsetzung**

## **1.3 Werbebotschaft**

## **1.4 Strategische Planung**

## **2 Planung und Skizzierung**

### **2.1 Zielgruppenanalyse**

### **2.2 Tests**

### **2.3 Technischer Aufbau**

## 3 Umsetzung des Prototypen

### 3.1 Entwicklungs-Entscheidungen

Zur Entwicklung des Prototypen haben wir uns für die Gameengine Unity<sup>1</sup> entschieden. Gründe hierfür waren, dass der Umgang mit Unity den meisten Projektmitgliedern aus anderen Lehrveranstaltungen geläufig war und die Anbindung externer Module (z.B Personenerkennung) sehr einfach ist.

### 3.2 Model Aufbau

### 3.3 DMX-Scheinwerfer

*Linda Schey*

Für die LED-Scheinwerfer wurden acht Eurolite LED CLS-18 QCL RGBW 18x8W 12° 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. Des weiteren können sie Ein Eurolite USB-DMX512-PRO Interface B-Stock dient als Schnittstelle zwischen den Lichtern und dem PC. Angesteuert werden die Lichter mittels Freestyler<sup>2</sup>, einer kostenfreien DMX-Lichtsteuerungs-Software. Abbildung 1 zeigt eine Skizze 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. Jeder Scheinwerfer besteht aus 18



Abbildung 1: Verknüpfung der Komponenten

LEDs die in drei Reihen à sechs LEDs angeordnet sind. Zwei Zeilen, also sechs

---

<sup>1</sup><http://www.unity3d.com>

<sup>2</sup><http://www.freestylerdmx.be/>

LEDs, werden in der Applikation jeweils als ein Segment verwendet und kann mit vier Farbkanälen separat angesteuert werden. Insgesamt werden also 12 Farbkanäle pro Scheinwerfer eingesetzt. Innerhalb der „Blinken Tiles“ 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. Die Klasse *Spot.cs* 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 befüllt. 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. Alle anderen Scheinwerfer und deren Segmente werden aus Schwarz gestellt, also nicht leuchtend. Jeder Scheinwerfer besteht aus 18 LEDs die in drei Reihen à sechs LEDs angeordnet sind. Zwei Zeilen, also sechs LEDs, werden in der Applikation jeweils als ein Segment verwendet und kann mit vier Farbkanälen separat angesteuert werden. Insgesamt werden also 12 Farbkanäle pro Scheinwerfer eingesetzt. Innerhalb der „Blinken Tiles“ 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. Die Klasse *Spot.cs* 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 befüllt. 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. Alle anderen Scheinwerfer und deren Segmente werden aus Schwarz gestellt, also nicht leuchtend. Um die Lichter aus der Anwendung „Blinken Tiles“ heraus zu steuern wird ein Zwischenschritt über die Software Freestyler gemacht, da diese Software die Ansteuerung der einzelnen Segmente eines jeden Scheinwerfers sehr komfortabel gestaltet. Um aus „BlinkenTiles“ mit Freestyler zu kommunizieren wurde eine DLL (*LetThereBeLight.dll*) erstellt die beim Start der Applikation eine Verbindung mit Freestyler herstellt (*FindWindow((string) null, "FS");*) und, die per *SendMessage()* die gewünschten Befehle an Freestyler weiter gibt und von dort an die Scheinwerfer bzw. das betreffende Scheinwerfer-Segment. Die Funktion *UpdateFaderValues(int curCol, float timerCol)*

### 3.4 Multimonitor Support

*Linda Schey*

Zu Beginn des Projektes wurde versucht eine möglichst große Projektionsfläche zu erhalten. In Ermangelung eines Beamer, der die gewünschte Größe projizieren kann wurde auf drei Beamer zurückgegriffen, die dann nebeneinander ein jeder ein Drittel der Applikation darstellen sollte. Um dies zu realisieren musste aber das komplette Bild das projiziert werden sollte in drei einzelne Bilde geteilt und jeweils um 90° gedreht werden, da die Beamer nicht nebeneinander, sondern über einander aufgebaut werden sollten, damit das komplette Bild hoch und breit genug ist. Dafür wurde in Unity eine extra Kamera in die Szene eingefügt, die die Szene als RenderTexture aufnimmt. Diese RenderTexture wurde dann in drei einzelne Texturen getrennt. Jede Textur um 90° gedreht und dann nebeneinander als Gui-Elemente dargestellt. Soweit war diese Funktionalität auch schon implementiert, bis auf kleine Justierungen beim Zuschneiden der einzelnen Teil Texturen.

Ein anderer Ansatz war drei um 90° gedrehte Kameras in die Szene zu integrieren die jeweils ein Drittel des „Spielfeldes“ filmten. Daraus erhielt man drei

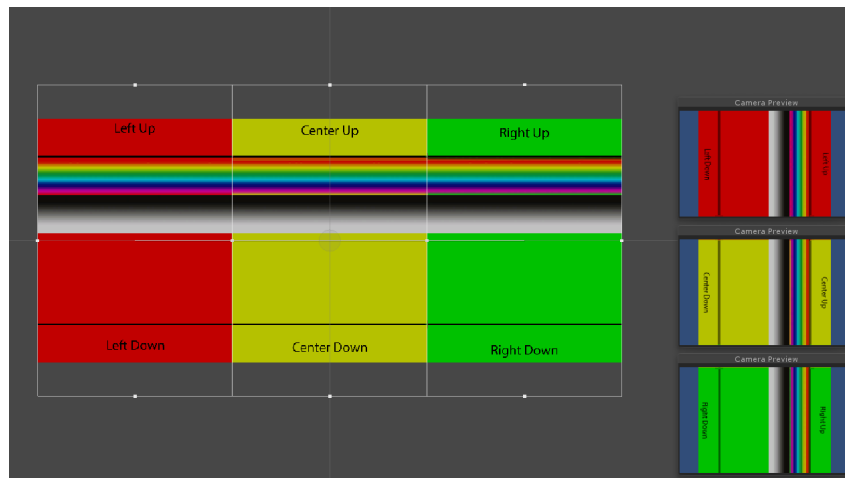


Abbildung 2: Aufbau der Szene in Unity mit den drein RenderTexturre Ausschnitten am Beispiel eines Dummy Objektes

RenderTextures und dann nebeneinander als Gui rendern konnte. Dafür wurde



zunächst mit einem Beispiel Objekt gearbeitet, das ein Material zugewiesen bekam, das es bei der Entwicklung vereinfachte zu erkennen, ob die Kameras richtig ausgerichtet waren und dann auch entsprechend richtig auf die Gui gerendert wurden.

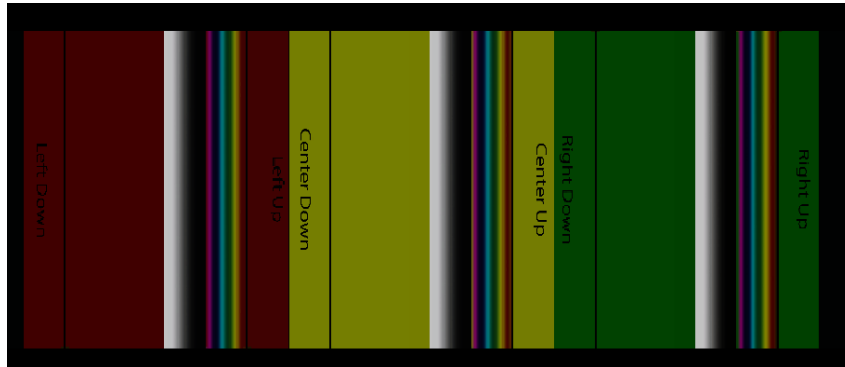


Abbildung 3: RenderTextures nebeneinander auf die Gui gerendert

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 gewesen die Szene auf die drei Beamer ausrichten könne, in dem man zum Beispiel den Übergänge zwischen den drei Texturen jeweils so hätte Platzieren könne, dass sie einem Spalt auf dem Spielfeld entsprächen. Beide Ansätze waren jedoch nicht zu 100 Prozent exakt, was das Aufteilen der Szene in drei Teil Texturen betraf. Außerdem hatten wir keine drei identischen Beamern zur Verfügung, sodass es unter den einzelnen Beamern Unterschiede bei der Lichtintensität gab. Da anstatt der Lösung mit den drei Beamern doch noch ein 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.

### 3.5 Spielfeld und Tiles

*Alexander Scheurer*

Das Spielfeld ist so konzipiert dass es für jeden Bedarf dynamisch angepasst

werden kann. Der Aufbau des Spielfelds (Maße, Anzahl der Tiles, etc.) wird über eine Konfigurationsdatei gesteuert und können auch zur Laufzeit geändert werden.

Als Datenspeicher zur Laufzeit wurden eigene Datentypen definiert die 2-Dimensional verschachtelt in Listen gehalten werden. Über Methoden kommunizieren hiermit alle anderen Klassen die Einfluss auf das Verhalten der Tiles, zum Beispiel Farbwechsel, haben.

Das Spielfeld verwaltet auch als zentrales Element die Spielelogik und aktualisiert bei Bedarf die Konfiguration und auch alle anderen Elemente. Er gibt auch die Taktung des Spiels vor was in vier verschiedene Gameloops aufgeteilt werden kann:

- Update (Unity eigen)
- LateUpdate (Unity eigen)
- FixedUpdate (Unity eigen)
- BPM Update

In Update und LateUpdate wird die Spiellogik, das Zeichnen des Bildes und alle nicht zeitkritischen Operationen durchgeführt. FixedUpdate verwaltet aufgrund seines zeitlich definierten Ablaufes die Taktung des Spiels und bildet in Abhängigkeit des aktuellen Liedes das "BPM Update", das Beispielsweise den Takt für die DMX-Lichtersteuerung Vorgibt.

Die Zuweisung des aktuellen Status der Tiles passiert der Übersichtlichkeit wegen sequenziell.

1. *Clear pass* - Reset
2. *Preview pass* - Preview Status bei Challenge Mode
3. *Timer pass* - Setzen des Zeitimpulses
4. *People pass* - Setzen der aktiven Tiles anhand der Personenpositionen
5. *Shake'n'Play pass* - Abspielen der Audio-Clips anhand des Zeitimpulses und der und der Personenposition

Die Tiles verwalten sich größtenteils selbst. Von außen wird der derzeitige Status zugewiesen und es wird benachrichtigt bei Songwechsel. Laden der Shader, laden des entsprechenden Audio-Clips und rendern des Tiles übernimmt es selbst.

### **3.6 Call to Action**

### **3.7 Kinect-Erkennung**

### **3.8 Audiogestaltung und Inspiration**

*Johannes Winter*

Das Konzept der Installation beruht auf einem klassischen Step-Sequencer. Solche Sequencer 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-Sequencer 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-Sequencer 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-Sequencer 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-Sequencer ab. Zum Einen gibt es statt 16 Schritten nur 8. Außerdem ist die Tonauswahl auf 5 Tonhöhen pro Schritt begrenzt. Des Weiteren wird ist 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 Halbtonschritte 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 beschränkt, 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 Klangeigenschaften 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.

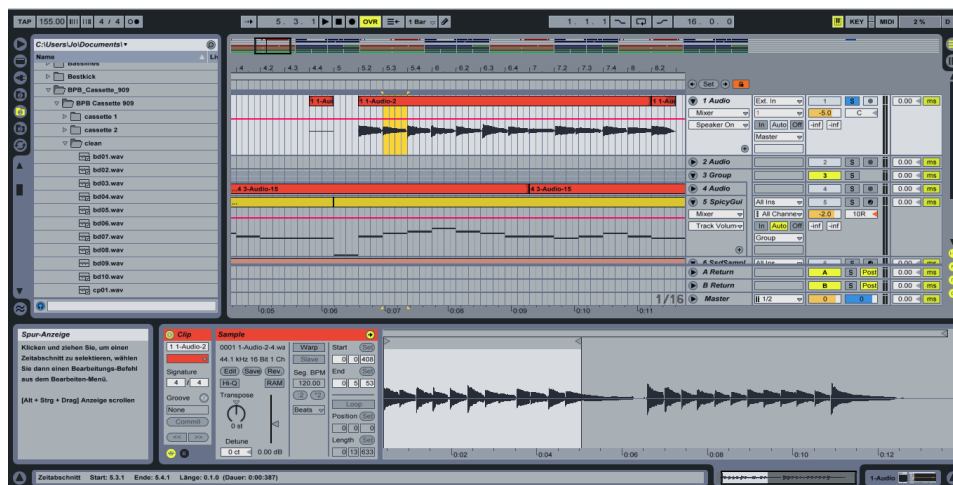


Abbildung 4: Schneiden der Spuren in Ableton Live

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.

## **4 Tag der Medien**

### **4.1 kleiner Bericht mit Fotos**

### **4.2 Praxiserfahrungen**

## **5 Showreel**



## **6 Fazit, mögl. Weiterentwicklungsmöglichkeiten**