



<BILD: Silhouetris StandbyScreen>

Abstrakt

Die menschliche Silhouette trifft auf ein bekanntes Retro-Videospiel. „Silhouetris“ vereint das Spiegelbild mit dem Kultspiel „Tetris“.

<VIDEO>

Inhaltsverzeichnis

1. Konzept & Idee
 1. Hinleitung „Transforming Mirrors“
 2. Konzeption
 3. Tag der Medien
 4. Aufbau Skizze
 5. Erwartungen
2. Technische Umsetzung
 1. MAX/MSP/JITTER Patch & Java Code
 2. Silhouetris-Patch
 3. Hardware
3. User Erlebnis
 1. Interaktion
 1. Geplante Interaktion
 2. Ungeplante Interaktion
 2. Kritische Punkte
4. Verbesserungen
 1. Aus konzeptioneller Sicht
 2. Aus technischer Sicht
5. Ausblick
 1. Weitere Ziele & Ideen
6. Danksagungen

Konzept & Idee

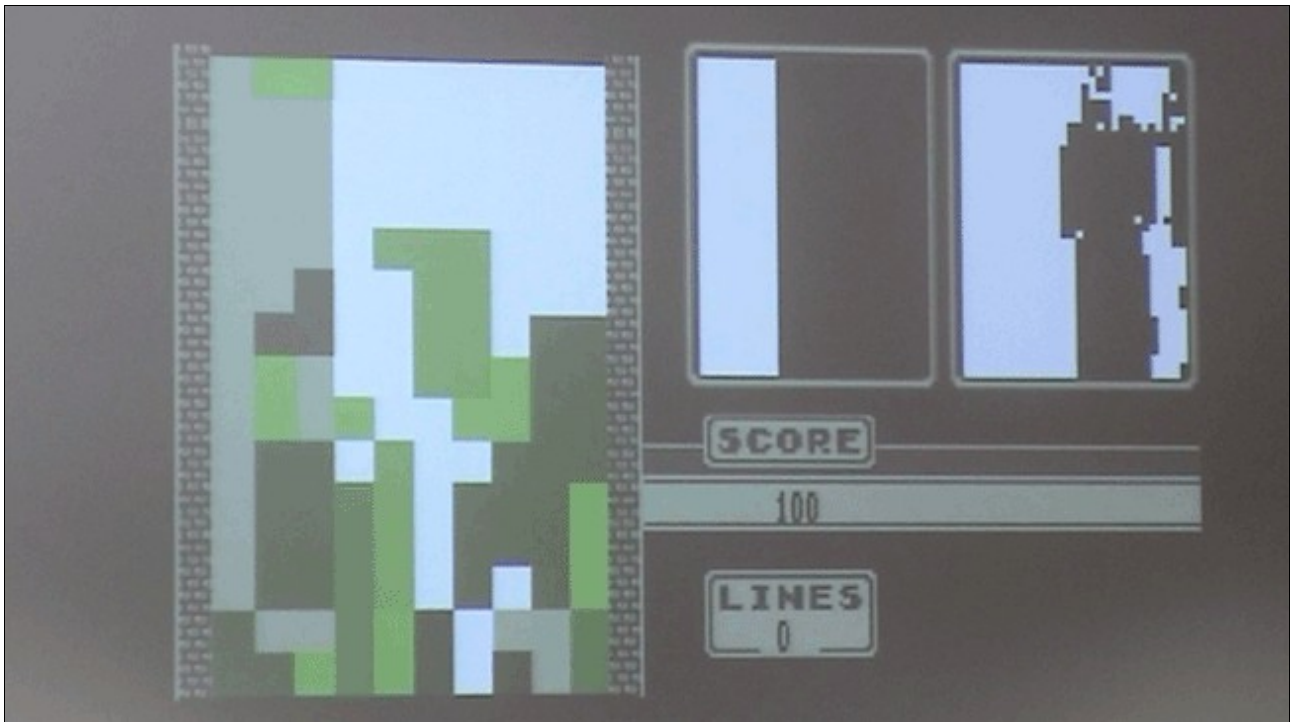
Hinleitung „Transforming Mirrors“

Diese Medieninstallation entstand im Rahmen des Wahlpflichtfaches „Interaktive Medieninstallation“ an der Hochschule Furtwangen University.

Die Idee zum vorgegebenen Thema „Transforming Mirrors“ war, das eigene Spiegelbild mit einem interaktiven Spiel zu verknüpfen. Dabei sollten spielerische und soziale Aspekte im Vordergrund stehen.

Konzeption

„Silhouetris“ nutzt die Silhouette des Users als Schnittstelle zum Spiel.



<BILD: Silhouetris Screen>

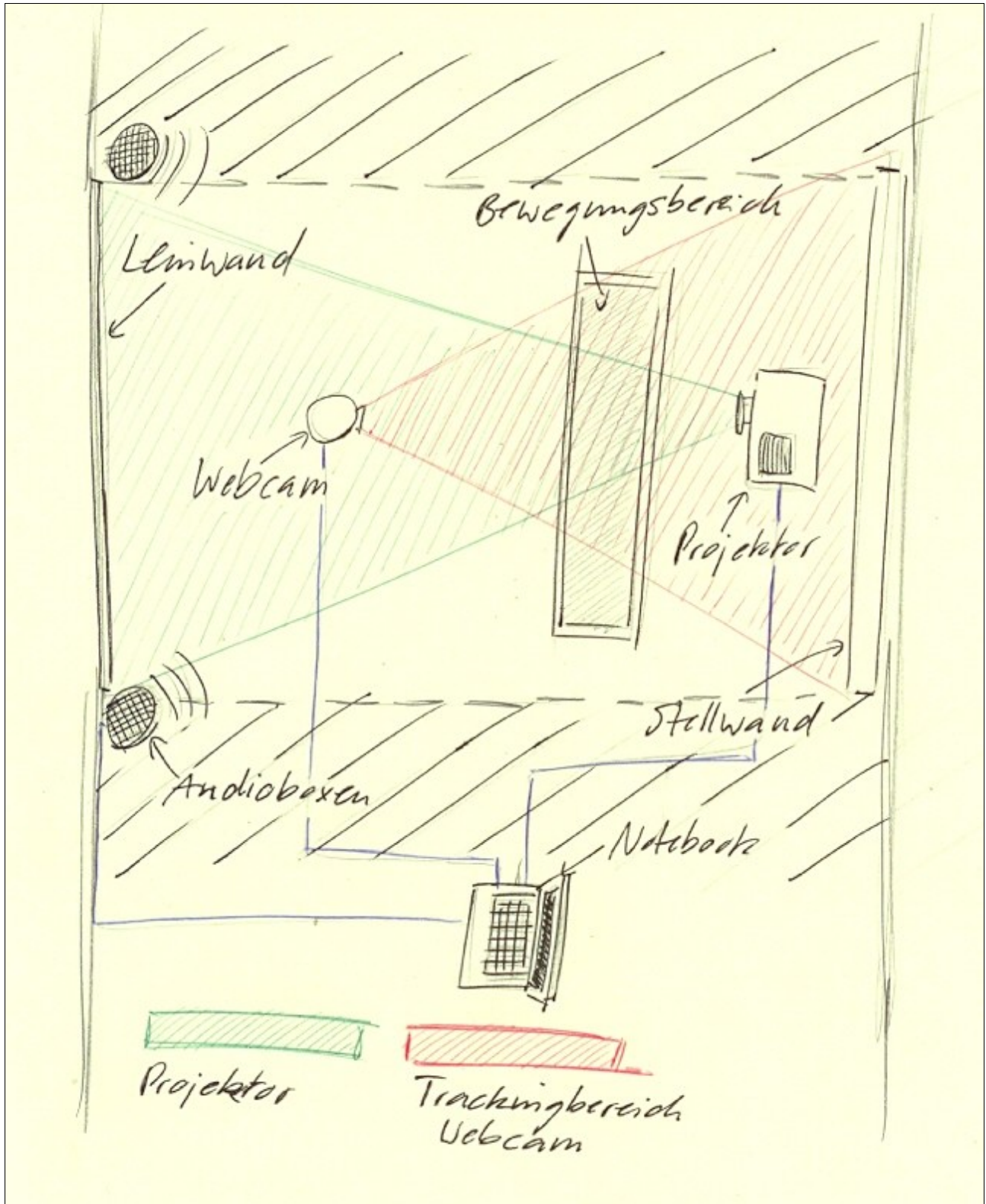
Das Ziel des Spiels ist identisch mit dem des „Tetris“-Spiels. Durch die nach unten wandernden Blöcke müssen so viele Reihen wie möglich vervollständigt werden, um das Spielfeld frei und den Punktestand hoch zu halten. Jedoch ist die Kontrolle zum original „Tetris“ umgedreht. Die Spalte (und evtl. die Drehung) sind vom Spiel vorgegeben und man muss die Form der Blöcke verändern. Dies geschieht mittels des eigenen Spiegelbildes.

Es wird die gespiegelte Silhouette genutzt. Sie bestimmt die Form des jeweils nächsten Blocks und wird dafür größtmöglich vereinfacht. Das Spiegelbild wird auf ein 3 zu 4 Raster reduziert und bietet den Rahmen zur Erstellung der Blöcke. Die menschliche Silhouette mit den Armen und Beinen, einem unterschiedlich verlagerten Schwerpunkt und die versetzten Positionen sind die entscheidenden Merkmale für die Form der Blöcke.

Tag der Medien

Am 30. Juni 2009 wurde die Installation für den Tag der Medien an der Hochschule Furtwangen University aufgebaut und zum ersten Mal einem Publikum präsentiert. Dieses Datum war ein Stichtag für die Fertigstellung der Installation. Einige Punkte konnte bis dahin nicht fertiggestellt oder ausreichend getestet werden. Als Ort diente ein Grafikpool mit Projektor und Leinwand.

Aufbau Skizze



<BILD: Aufbau>

Der Raum der ganzen Installation wird vorne von der Leinwand des Projektors und hinten von einer einfarbigen Stellwand eingegrenzt. Unterhalb der Leinwand steht eine Webcam, die den Spieler frontal vor der Stellwand aufnimmt. Der Spieler befindet sich in einem definierten Bewegungsraum, der eine seitliche Bewegung erlaubt und gleichzeitig die Bewegung auf die Kamera zu und weg von der Kamera einschränkt. Der Blick ist auf die Leinwand gerichtet auf der seine Silhouette die Form der Blöcke bestimmt. Seitlich

ist Platz für Zuschauer und wartende Leute. Die Beleuchtung ist eine normale Tageslichtsituation, wobei auch fast jede andere Beleuchtung umsetzbar wäre, solange der Akteur keinen Schatten auf die Stellwand wirft. In einem optimal aufgebauten Raum könnte man die Stellwand durch eine Raumwand ersetzen oder würde sogar ganz wegfallen solange die Lichtsituation und der Hintergrund unverändert bleiben.

Demnach ist die Installation, mit wenigen räumlichen Voraussetzungen, an vielen Orten denkbar und erlaubt eine hohe Flexibilität.

Erwartungen

Es entsteht ein interaktives Erlebnis durch die körperlich aktive Steuerung und die intuitive Blockformung. Die meisten Leute sind mit dem Spiel „Tetris“ und dessen Spielregeln vertraut und werden schnell erkennen wie „Silhouetris“ zu spielen und zu steuern ist. Die Gestaltung ist absichtlich simpel und einfach gehalten. Sie kommt ohne große Effekte aus und erinnert an ein altes Arcade Game. Daher wird keine große Hemmschwelle vorhanden sein die Installation auszuprobieren.

Technische Umsetzung

MAX/MSP/JITTER Patch & Java Code

Die komplette Umsetzung fand in MAX/MSP/JITTER und Java statt. Bereiche wie die Logik des Spiels wurden in Java geschrieben und mit einem speziellen Java-Patch in MAX/MSP/JITTER eingebunden. Die Videoaufnahme des Akteurs, die Weiterverarbeitung der daraus gewonnenen Jitter-Matrixen, die Einspielung von Sounds und Musik, sowie die Zusammensetzung des letztendlich gezeigten Video-Outputs wurde in einem MAX-Patch realisiert.

Silhouetris-Patch

Der Patch ist in sieben logische Phasen aufgeteilt. Jede Phase ist ein eigener Subpatch mit oftmals weiteren Subpatches. Zusätzlich wurden zur Erklärung der Spielregeln zwischen rein kurze Animationen eingeblendet.

Beim starten des Patches werden mit einem *loadbang* alle benötigten Bild- und Sounddateien, sowie die Einstellungen für die Kamera, geladen. Danach folgt es den einzelnen Phasen des Patches, durch die die Installation gleichzeitig strukturiert ist.

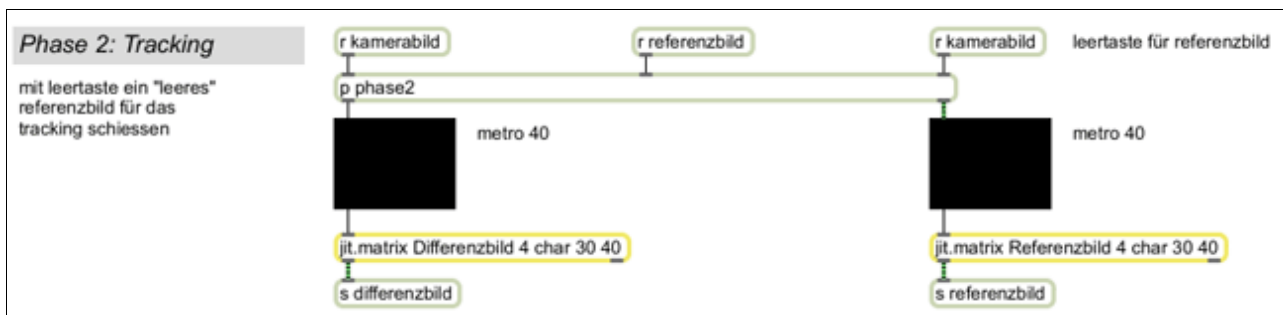
Phase 1: Kamera



<BILD: phase1.png>

Die Einstellungen für die Webcam werden gesetzt. Das Videobild wird sofort gespiegelt, um bei der Ausgabe ein „echtes“ Spiegelbild wiederzugeben. Das Kamerabild wird in einer Auflösung von 60 zu 80 aufgenommen und an die nächste Phase weitergegeben.

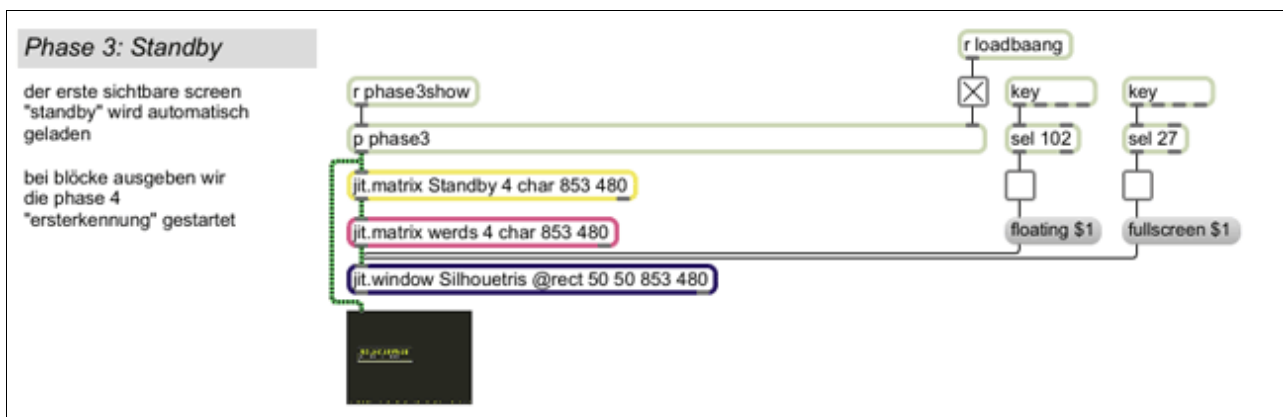
Phase 2: Tracking



<BILD: phase2.png>

Das Kamerabild wird auf Schwarz und Weiß gefiltert. Mit einem Referenzbild der leeren Stellwand wird dann ein Differenzbild generiert, das die reine Silhouette des sich vor der Stellwand befindenden Spielers freistellt. Dieses Differenzbild, welches veränderte Bildpunkte mit schwarz kennzeichnet, wird an Phase 4 weitergegeben.

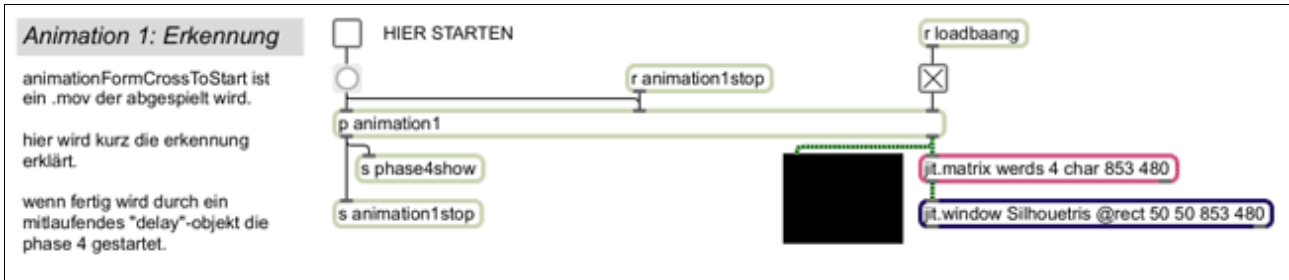
Phase 3: Standby



<BILD: phase3.png>

Wenn der Patch das erste Mal gestartet wurde oder wenn gerade nichts passiert, wird das Standby-Bild angezeigt. Dies ist das erste Bild, das dem Akteur gezeigt wird.

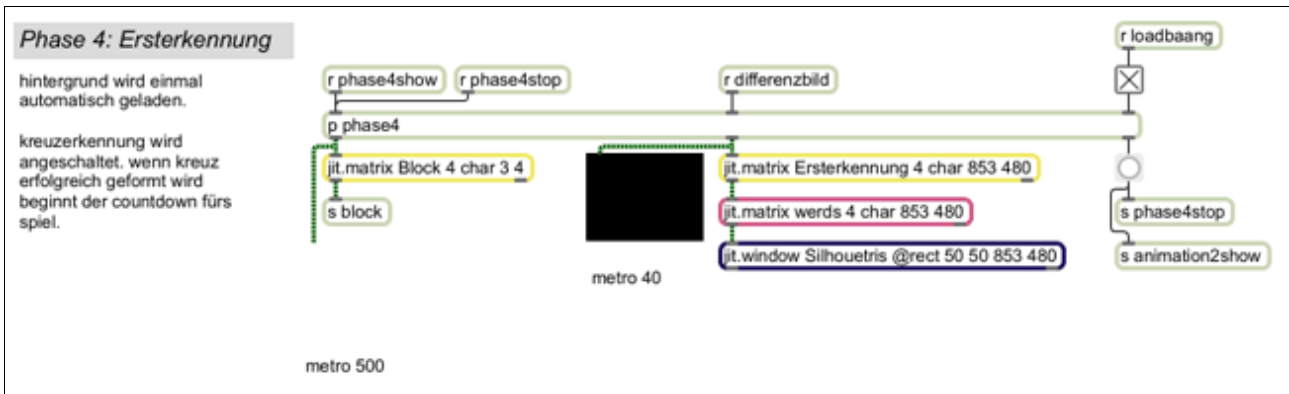
Animation 1: Erkennung



<BILD: animation1.png>

Die erste Animation zeigt dem Akteur wie er das Spiel starten kann.

Phase 4: Ersterkennung



<BILD: phase4.png>

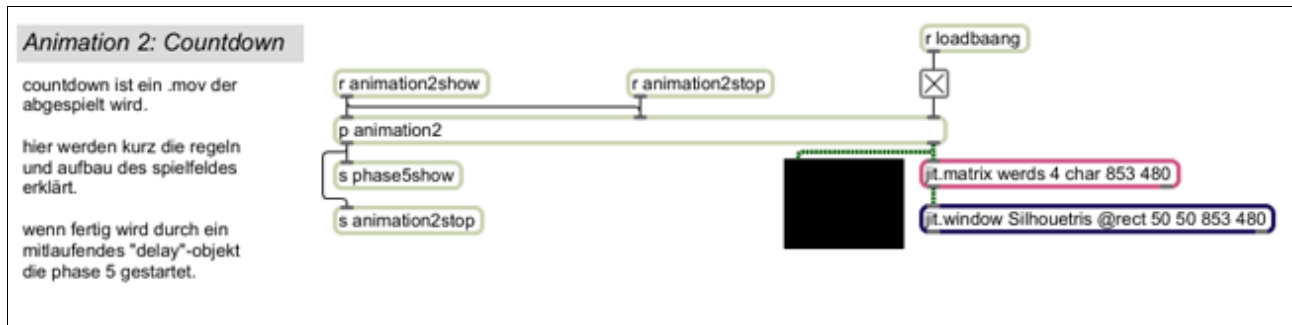
In dieser Phase werden die beiden selbstgeschriebenen Klassen, *getCross* und *ToBlock*, genutzt. (Wie man *eclipse* und *MAX/MSP/JITTER* für das Einbinden eines solchen Patches konfiguriert, ist auf [„http://compusition.com/web/articles/maxmsp-eclipse“](http://compusition.com/web/articles/maxmsp-eclipse) gut erklärt.) *ToBlock* teilt das Differenzbild aus Phase 3 in ein 3x4 Raster auf. Die Auflösung eines Rasterteilchen beträgt demnach 20x20 Pixel. Anschließend wird für jeden der 12 Rasterteilchen überprüft, ob die Anzahl der schwarzen Bildpunkte, größer als ein Schwellwert ist. Schwarze Bildpunkte stehen für Veränderungen im Bild. Diese Veränderungen entstehen durch die Silhouette der Akteure, die sich im vorgegeben Bewegungsraum befinden. Ein 20-prozentiger Anteil in jedem Rasterteilchen hat sich in der Testphase als ein relativ guter Schwellwert herausgestellt. Eine Optimierungsmöglichkeit, welche aber bis dato noch nicht implementiert wurde, wäre ein Puffer, der anstatt eines festen Schwellwertes einen Übergang erzeugt. Dies könnte konkret so aussehen, dass ab 22 Prozent ein Block des Rasters auf schwarz gesetzt wird, er aber nur bei 18 Prozent wieder entfernt würde. Dies sollte dazu führen, dass die Erzeugung der Blöcke nicht so präzise ist, dafür aber einen weicheren Übergang zwischen dem Setzen und Entfernen von Blöcken schafft.

Um das Spiel zu starten muss der Akteur ein vorgegebenes Kreuz mit seinem Körper formen. Die Erkennung dieser Form wird von *getCross* überprüft und gegebenenfalls das Signal zum Starten des Spiels gesendet. Dieser Kalibrierungsmodus war anfangs nur als Hilfe für den Spieler gedacht, damit er den richtigen Standpunkt in Bezug zur Kamera findet. Es hat sich herausgestellt, dass dies ein optimaler Zeitpunkt ist, ohne Zeitdruck, z.B. durch einen nach unten wandernden Block, verschiedene Formen ausprobieren.

Insgesamt bekommt der Spielende hierdurch ein besseres Gefühl, wie die

Blockformung funktioniert und es bringt sogar noch einen gewissen Spaß-Effekt mit sich.

Animation 2: Countdown



<BILD: animation2.png>

Bei der Countdown Animation wird in vier kurzen Sätzen die Spielregeln erklärt. Da sie sehr einfach sind, werden sie nach mehrmaligem Spielen nicht mehr benötigt und deshalb ist die Animation auch zeitlich kurz gehalten.

Phase 5: Display



<BILD: phase5.png>

In diesem Patch wird die komplette Spiellogik samt der daraus resultierenden Anzeigeänderung abgewickelt. Die Sounds werden abgespielt und sämtliche Jittermatrixen, wie die Matrix der Silhouette, des Blocks, der Score-Anzeige, der Lines-Anzeige und des Spielfeldes werden zusammengefügt. Der Großteil der Logik dieser Vorgänge läuft in dem Java Patch *SilhouetrisLogic* ab und wird dann im MAX/MSP/JITTER Patch zusammengesetzt.

Die Eingangssignale des *SilhouetrisLogic* Patches sind:

- Die 3x4 Matrix aus Phase 4 aus der die Steine geformt werden.
- Ein *metro*-Signal, das als Bildaktualisierungstakt des Spielfeldes dient.
- Ein *movedown*-Signal, das die Fallgeschwindigkeit der Steine festlegt.
- Die Signale *reset*, *pause* und *start* zur Steuerung des Spielablaufs.

Die Auswertung dieser Signale erfolgt in einer Vielzahl von Funktionen, wobei die Kollisionserkennung die wohl höchste Komplexität besitzt.

Die Ausgabe der Berechnungen erfolgt dann über 5 Ausgänge:

- Das Spielfeld als Matrix in der Größe von 10x19 Pixel.
- Ein Signal bei einer Kollision, was das Abspielen eines Tons bewirkt.
- Ein Signal, wenn das Spiel verloren ist.
- Die aktuelle Punktzahl als Nummernwert.
- Alle bisher erzielten vollen Linien als Nummernwert.

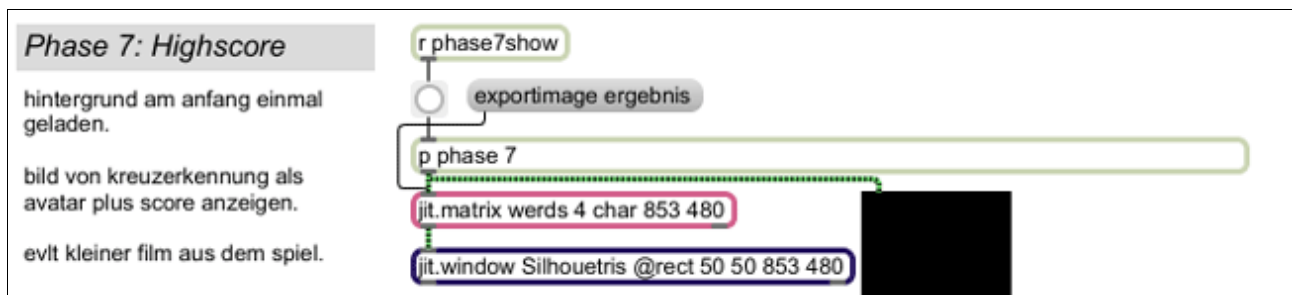
Phase 6: Gameover



<BILD: phase6.png>

Der Gameover Screen wird für 4 Sekunden angezeigt.

Phase 7: Highscore



<BILD: phase7.png>

Hier war die Highscore-Anzeige gedacht, jedoch wurde diese Phase nicht fertiggestellt und stattdessen wird das Spielfeld mit dem Zustand zum Zeitpunkt des Spielendes auf der ganzen Leinwand angezeigt.

Hardware

Die wichtigsten technischen Komponenten sind ein Notebook auf dem der MAX/MSP/JITTER Patch läuft, eine Videoanzeige (z.B. einen Projektor) und eine USB-Webcam, da kein hochauflösendes Bild zum Tracking benötigt wird. Für die Musik sollte das Notebook mit Lautsprechern verbunden sein.

Für den Raum sind eine einfarbige Stellwand bzw. eine einfarbige Raumwand von Nöten, die das Tracking erleichtern.

User Erlebnis

Interaktion

Die Interaktion in dieser Medieninstallation war in erster Linie zwischen dem Spiel und dem Akteur geplant. Es entsteht ein interaktives Erlebnis durch die körperlich aktive Steuerung und die intuitive Blockformung, die einem die größtmögliche Einfachheit seiner eigenen Silhouette darstellt und einem die Möglichkeit gibt, diese gekonnt einzusetzen, um in diesem Spiel viele Punkte zu erreichen. Es war auch ein Zwei-Spieler-Modus mit zwei unabhängigen Displays geplant, bei dem die Spieler gegeneinander antreten können.

Geplante Interaktion

Die Interaktion mit dem Spiel selber war interessant zu beobachten, da die Akteure auf unterschiedlichste Art die Blockformung genutzt haben. Manche stellten sich ganz Aufrecht hin und bewegten sich nur bei einem Reihenwechsel zu einer neuen Position, an der sie wiederum die gleiche Form generierten. So entstanden viele vertikale Balken und die Interaktion war sehr eintönig. Dann gab es viele, die sofort schwierige Formen ausprobierten und erstmal feststellen mussten, dass die Form der eigenen Silhouette nicht immer mit der geplanten Form aus der Vorstellung übereinstimmte. Dadurch entstand nach und nach ein sehr anspruchsvolles Spielfeld, in das es galt weitere komplexere Formen einzufügen. Diese Interaktion benötigte oft viel Körpereinsatz und wurde vom Publikum interessiert angeschaut.

Ungeplante Interaktion

Je nachdem, wie gespielt wurde, hat das Publikum angefeuert oder hat Tipps zugerufen. Die Hintergrundmusik hat zum Mitwippen angeregt und somit die Bereitschaft beim Akteur mitzufiebern erhöht. Grundsätzlich erfolgte die Interaktion also wie geplant, jedoch wurde sie noch um eine Ebene erweitert als die Akteure anfangen zu zweit zu spielen. Dies hat schnell Gefallen gefunden und letztendlich waren fast nur noch Zweierteams dabei, andere Zweierteams im Highscore zu schlagen. Es entstand ein regelrechter Wettkampfcharakter, wobei nie der Spaß verloren ging, den es machte, mit einem Partner zusammen, auf unterschiedlichster Weise, die besten Formen zu erzeugen. Betrachtete man nur die Spieler, so sah es teilweise aus wie ein Tanz dieser beiden. Ganz Engagierte haben versucht bestimmte Funktionen im Spiel bewusst auszunutzen. Der nach unten wandernde Block bleibt stehen, wenn sich kein Spieler im Trackingbereich befindet. So kam es durchaus vor, dass sich Spieler auf den Boden legten, um den Trackingbereich zu verlassen und das Spiel somit kurz anzuhalten und die nächste Form zu planen. Dabei gab es rege Absprache mit dem Partner, wenn es, wie so oft, im Zweispielermodus gespielt wurde.

Kritische Punkte

Die kurzen Anleitungen wurden teilweise nicht beachtet teilweise nicht erkannt. Dies führte dann des öfteren dazu, dass dem Akteur nicht bewusst war, dass er die Reihe, in der der Block nach unten wandert, nicht beeinflussen kann, wie man es aus dem Original Tetris kennt. Dies konnte anfangs für Verwirrung sorgen bis man es erklärt bekommen hat.

Die Spielzeit schwankt von Spieler zu Spieler, was an der unterschiedlichen Spielkompetenz liegt. Spielt es jemand zum ersten Mal ist das Spiel oft

schnell vorbei, wobei es wiederum bei einem erfahrenerem Spieler lange dauern kann und die Schwierigkeit nicht zunimmt.

Die Lichtverhältnisse stellte sich als äußerst problematisch heraus. Schon kleine Veränderungen im Hintergrund haben manchmal zu ungewollter Blockformung geführt und haben das Spielen negativ beeinflusst. Man musste ein neues Differenzbild schießen und konnte dann das Spiel weiterführen. Anscheinend hatte dies auch mit der Kleidung und dem Abstand zur Kamera zu tun, da dadurch, obwohl der Autofokus ausgeschaltet war, das Differenzbild anders ausgewertet wurde.

Verbesserungen

Aus konzeptioneller Sicht

Bei dem Installationsaufbau im Grafikpool der Hochschule ging es oftmals zu hektisch zu, als dass man sich die Zeit nehmen konnte die Anleitungen zu lesen. Diese müsste entweder zusätzlich von einem Sprecher während der Animation gesprochen werden oder im Vorhinein klar sein. Für erfahrene Spieler sollte es verschiedene Schwierigkeitslevel geben.

Die verschiedenen Matrizen, wie das Original Aufnahmebild, das gepixelte Bild, die Blöcke und das Spielfeld werden temporär in separaten Matrizen gespeichert, um weiterverwendet zu werden. Ein Schnelldurchlauf des gerade gespielten Spiels kann z.B. am Ende gezeigt werden.

Aus technischer Sicht

Am Tracking kann noch vieles verbessert werden. Der ausgeschaltete Autofokus der Webcam ist nicht genug, um die Differenzbildänderungen bei unterschiedlichen Lichtsituationen und unterschiedlich weit entfernten Akteuren zur Webcam, so niedrig zu halten, dass es die Blockformung nicht beeinflusst. Man muss eine gewisse Farbänderungstoleranz beim Referenzbild miteinberechnen, sodass erst ein Differenzbild entsteht, wenn sich wirklich ein Akteur vor der Webcam befindet und nicht die Sonneneinstrahlung im Hintergrund. Man muss bemerken, dass die Stellwand beim Aufbau im Grafikpool der Hochschule im oberen Bereich nicht alles abgedeckt hat und somit der sich dahinter befindende Raum bei den oberen Rasterteilchen mitberücksichtigt wurde. Die Kleidung der Akteure hat bei dieser Technik leider auch negative Auswirkungen gehabt, deshalb wäre der Ansatz, der Hintergrund komplett mit hellem Licht auszuleuchten, in dem Fall geschickter gewesen.

Ausblick

Weitere Ziele & Ideen

Ein Ziel ist das gesammelte Videomaterial während eines Spiels am Ende nochmals zu verwerten. Eine Idee ist die Bewegungen der Akteure im Zeitraffer nochmals abzuspielen, um diese dem Akteur nochmals ins Bewusstsein zu rufen, was er eigentlich die ganze Zeit geleistet hat. Auch andere Auswertungen sind möglich, wie die Benutzung des zum Spielende „gebauten“ Spielfeldes als Bild, das der Spieler unbewusst generiert hat.

Eine weitere Idee wäre ein Spiel über mehrere Tage bzw. Wochen aufzuziehen. Da sich Blöcke immer nur bewegen, wenn sich jemand im Trackingbereich befindet, setzt man den Trackingbereich an eine belebte Stelle und das Spielfeld wird sich so nach und nach fühlen. Dieses Spielfeld müsste dann eine große Dimension haben und am Ende hätte man ein, von den vorbeigehenden und teilweise mitspielenden Leuten, generiertes Kunstwerk.

Danksagungen

Abschließend ein Dankeschön an die beiden Professoren Prof. Dr. Wolfgang Taube und Prof. Daniel Fetzner für diese motivierende Veranstaltung.

Zusätzlich möchten wir uns auch ganz herzlich für die Unterstützung bei unseren Kommilitonen Maurizio Camagna, Georg Graf, Matthias Kutschner, Simon Scherzinger und vor allem Stefan Valouch bedanken.