LOOMOX

Medienprojekt 6. Semester

Studiengang Interaktive Medien (BA)

SS 2014

GLIEDERUNG

1. Einleitung

2. Team

3. Idee

3.1. Motivation

3.2. Recherche

3.3. Konzept

4. Umsetzung

4.1. Design

4.1.1. Entstehung des LOOMOX-Wesens

4.1.1.1. Der Schwarm

4.1.1.2. Grundidee "Wobbl"

4.1.1.3. Finaler LOOMOX als Spielfigur

4.1.1.4 LOOMOX Stilfindung

4.1.1.5 LOOMOX Emotions

4.1.1.6 LOOMOX-Personalisierung

4.1.2. Game- und Interaktionsdesign

4.1.2.1. Erstkontakt

4.1.2.2. Spielauswahl

4.1.2.3. Spielkonzepte

4.1.2.3.1. Frisbee

4.1.2.3.2. Bubbles

4.1.2.3.3. Monsters

4.1.3. Corporate Identity

4.2. Technik

4.2.1. Screen Anwendung

4.2.1.1. Tracking

4.2.1.2. Anwendungslogik

4.2.1.3. KI's

4.2.1.4. DB Anbindung

4.2.2. WebApp

4.2.2.1. LOOMOX via Kinetic JS

4.2.2.2. MySQL DB

4.2.2.3. GPS Location für Rückkehr

4.3 Finale Anwendung

5. Präsentation

5.1 Vortrag

5.2 Stand

6. Fazit

1. EINLEITUNG

Ziel unseres Projekts war es, eine interaktive Anwendung an der Fassade der Hochschule Augsburg zu installieren, die diese in der Öffentlichkeit repräsentiert. Obwohl sich der ursprünglich hierfür angedachte Ort als ungeeignet erwies, verfolgten wir dieses Ziel weiter und entwickelten das Konzept für die interaktive Installation LOOMOX. Die Beschreibung unserer Methodik und der konkreten Umsetzung ist Inhalt der folgenden Dokumentation.

2. TEAM

Technische Umsetzung

Sebastian Antosch

Sebastian Harter

Patrick Schröter

Susanne Rauchbart

Stephan Reichinger

Christian Baur

Design Umsetzung

Lukas Fornaro

Christian Reichart

Jonathan Irschl

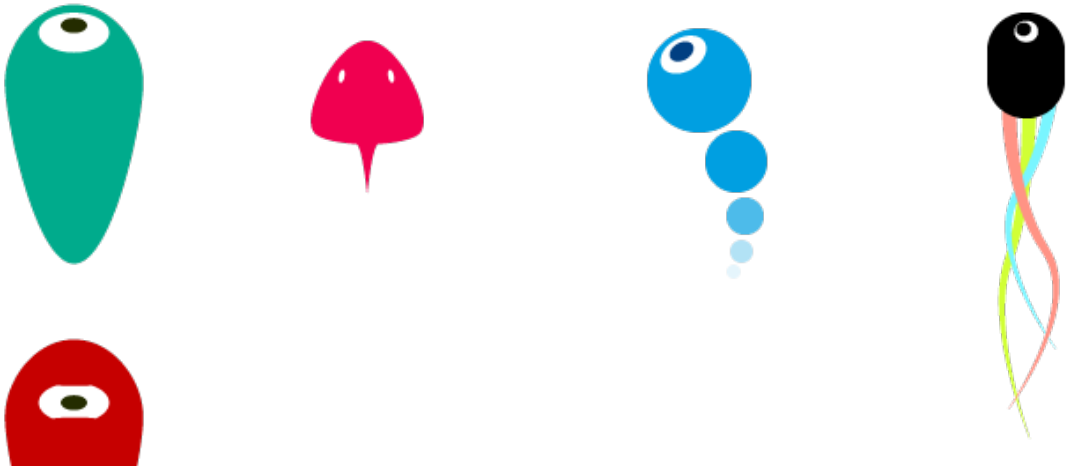
3. IDEE

3.1 Motivation

Da zu Beginn des Projekts der Zielort der Installation eine Scheibe zur Bushaltestelle vor der Fachhochschule war, war es unsere Motivation, die Menschen mit einer optisch ansprechenden Installation zu begeistern. Sie sollten Spaß daran haben unsere Anwendung zu benutzen und es sollte ihnen die Wartezeit zur nächsten Straßenbahn verkürzen. Zudem war es uns wichtig, den Nutzern eine Interaktion ohne weitere Hilfsmittel zu ermöglichen. Großen Wert legten wir außerdem auf eine zukünftige Erweiterbarkeit des Projekts.

3.2 Recherche

Anregungen und Inspiration erhielten wir aus vielen verschiedenen Quellen. Dazu gehörte beispielsweise das Schwarmverhalten nach Reynolds, aber auch das textbasierte Onlinespiel "Big Red Button". Aus diesen Recherchen entstanden dann erste Charakterdesigns.



Wichtig war es uns auch, den Einblick in die Hochschule und das gemeinsame Lernen, das hier jeden Tag praktiziert wird, aufzugreifen und widerzuspiegeln.

3.3 Konzept

Wir entschieden uns aufgrund technischer Probleme dafür eine Installation an der Fassade der Bibliothek an der Hochschule Augsburg. Mittels einer Rückprojektionsfolie und einem Beamer wird das Bild direkt in auf eine Scheibe projiziert. Es befindet sich immer ein Loomox, ein fremdartiges Wesen, auf der Leinwand sodass ein Nutzer jederzeit mit der Anwendung interagieren kann. Er kann mit dem Loomox gegen ein von einer KI gesteuerten Wesen spielen. Hat er das Spiel beendet wird ihm angeboten den Loomox mittels eines Codes, den er in einer Web App eingibt, zu adoptieren.

3.3.1 Gemeinsam

Der Loomox soll zum gemeinsamen Spielen animieren und dadurch eine Gemeinschaft zwischen den Usern entwickeln. Zu Beginn hat beherrscht jeder Loomox nur ein Spiel, besiegt er jedoch einen anderen Loomox, in einem ihm unbekannten Spiel, erlernt er dieses.

3.3.2 Ortsgebunden

Der gewählte Ort, an der Bibliothek der Hochschule Augsburg spiegelt den Gedanken des Voneinanderlernens wieder und bietet durch die Mensa genug Publikumsverkehr.

3.3.3 Personalisierbar

Hat man einen Loomox adoptiert kann dieser durch in Spielen gewonnen Items in seiner Form, der Farbe seiner Iris und der Größe seines Auges angepasst werden. Dadruch erhält man seinen eigenen persönlichen Lieblings-LOOMOX, zu dem man mehr Bindung aufbaut als zu einem zufällig generierten. Diesen LOOMOX kann man nun anderen Personen auf der Leinwand oder auf seinem Smartphone präsentieren.

4. UMSETZUNG

4.1. Design

Das Design erstreckte sich von der visuellen Ästhetik der Installation über einen stimmigen Marketingauftritt bis hin zu konkretem Game- und Interaktionsdesign.

4.1.1. Entstehung des LOOMOX-Wesens

4.1.1.1. Der Schwarm

Wie bereits erwähnt, wollten wir zunächst ein visuell abstraktes Wesen schaffen, welches sich in der Fensterscheibe aufhalten sollte. Der erste Gedanke hierbei war ein digitaler Schwarm, der aus mehreren einzelnen Teilen besteht, jedoch mit seinen Aktionen und seinem Verhalten wie ein einzelner Organismus funktioniert.
Hier experimentierten wir etwas und wollten den einzelnen Teilen des Schwarms ein niedliches, verspieltes Aussehen geben, sodass

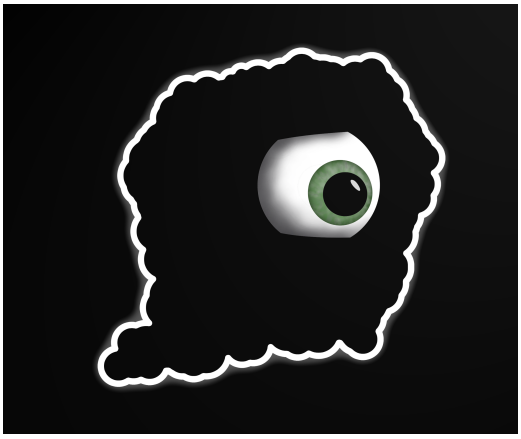
sich die Menschen besser mit diesem abstrakten Wesen identifizieren können.

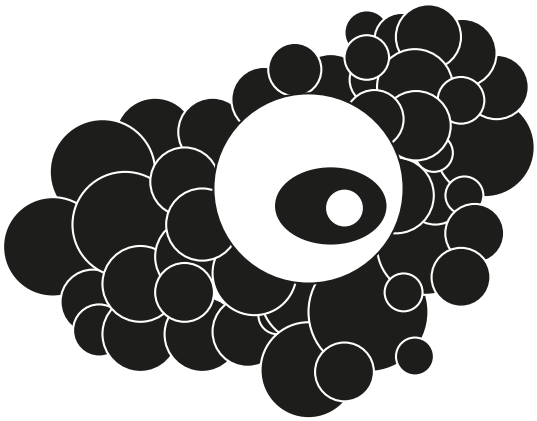
Um den Schwarm-Charakter des Wesens zu erhalten, mussten die einzelnen Teilchen recht klein werden, was jedoch zur Folge hatte, dass der angedachte "Niedlichkeitsfaktor" nicht mehr erkennbar war. Auch Emotionen ließen sich so nur sehr schwer darstellen.

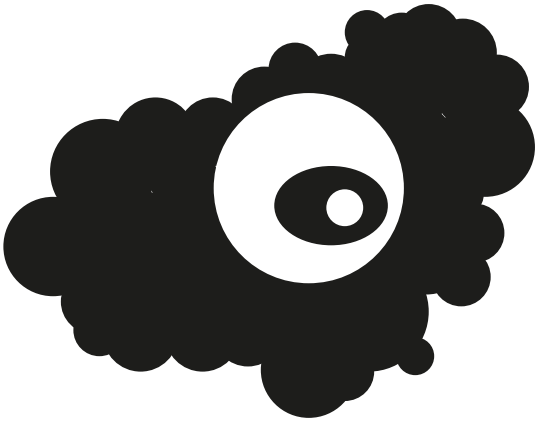
4.1.1.2. Grundidee "Wobbl"

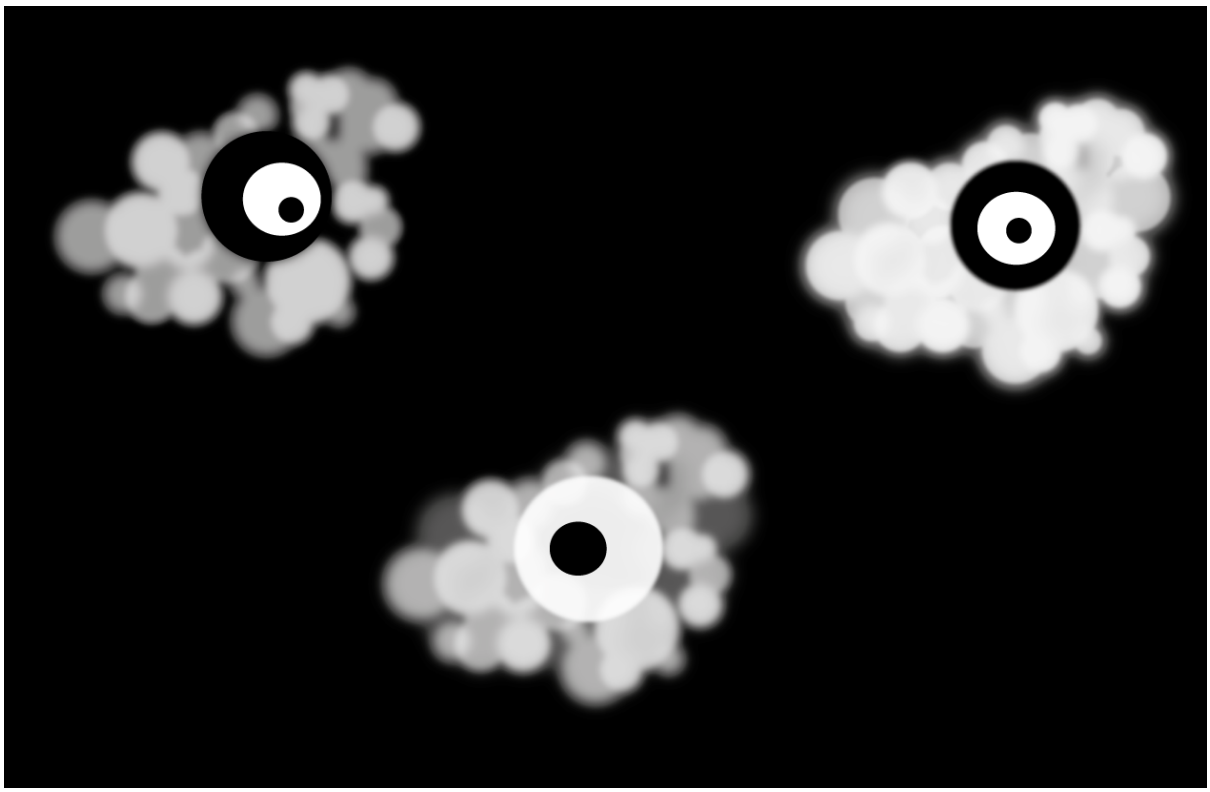
Das Problem lösten wir dadurch, dass wir die Schwarmteilchen als einfache Kreise darstellten und nah aneinander rückten. So wirkte unser Wesen wie ein einziger amorpher Körper. Anschließend erhielt das Wesen ein großes Auge, über das es von nun an kommunizieren sollte. Auch die Form des gesamten Körpers sollte zu Kommunikationszwecken in bestimmte Symbolformen zu bringen sein.

Dieser Schwarm, den wir zunächst als "Wobbl" umschrieben war nun also der neue Ausgangspunkt für unser Design. Auffallend hierbei war, dass unser "Wobbl" nur aus einfarbigen Kreisen bestand. Monochrome Flächen und kreisförmige Objekte definierten wir deshalb zu unseren Kernpunkten des visuellen Erscheinungsbildes.





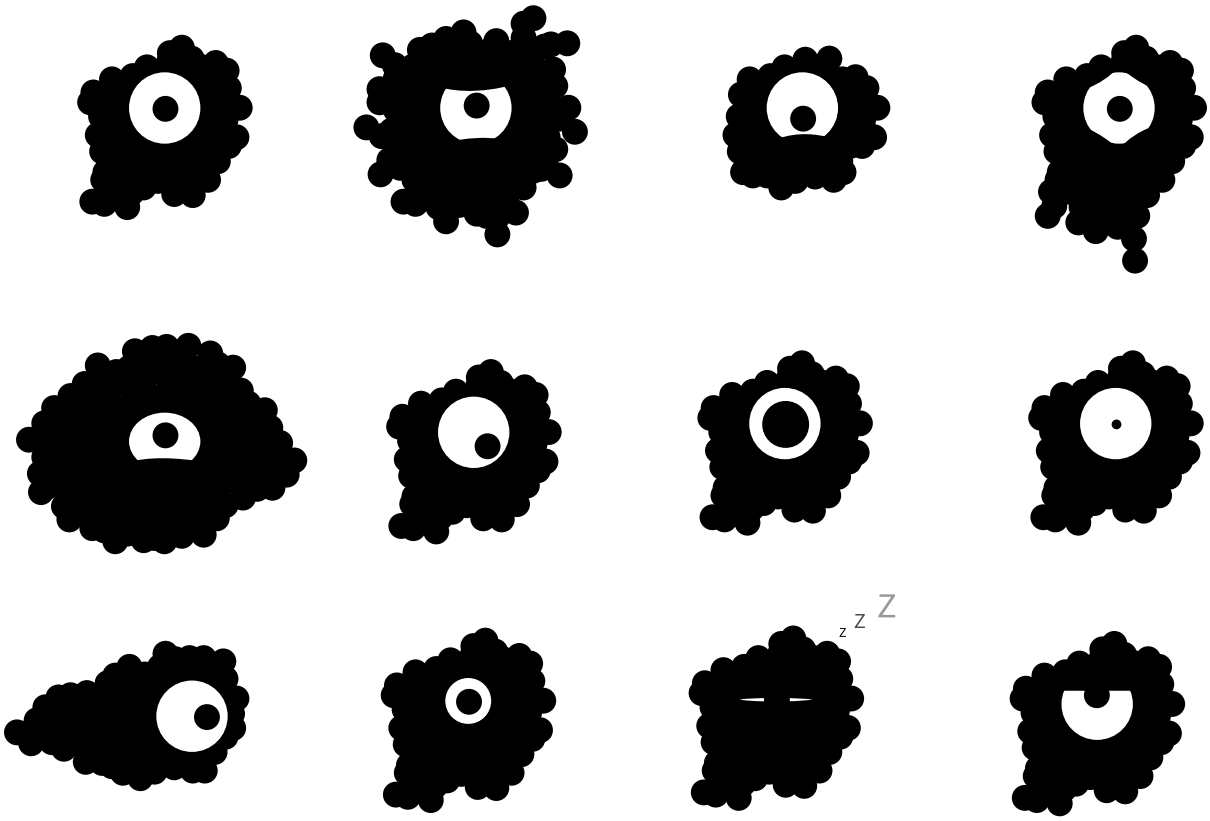


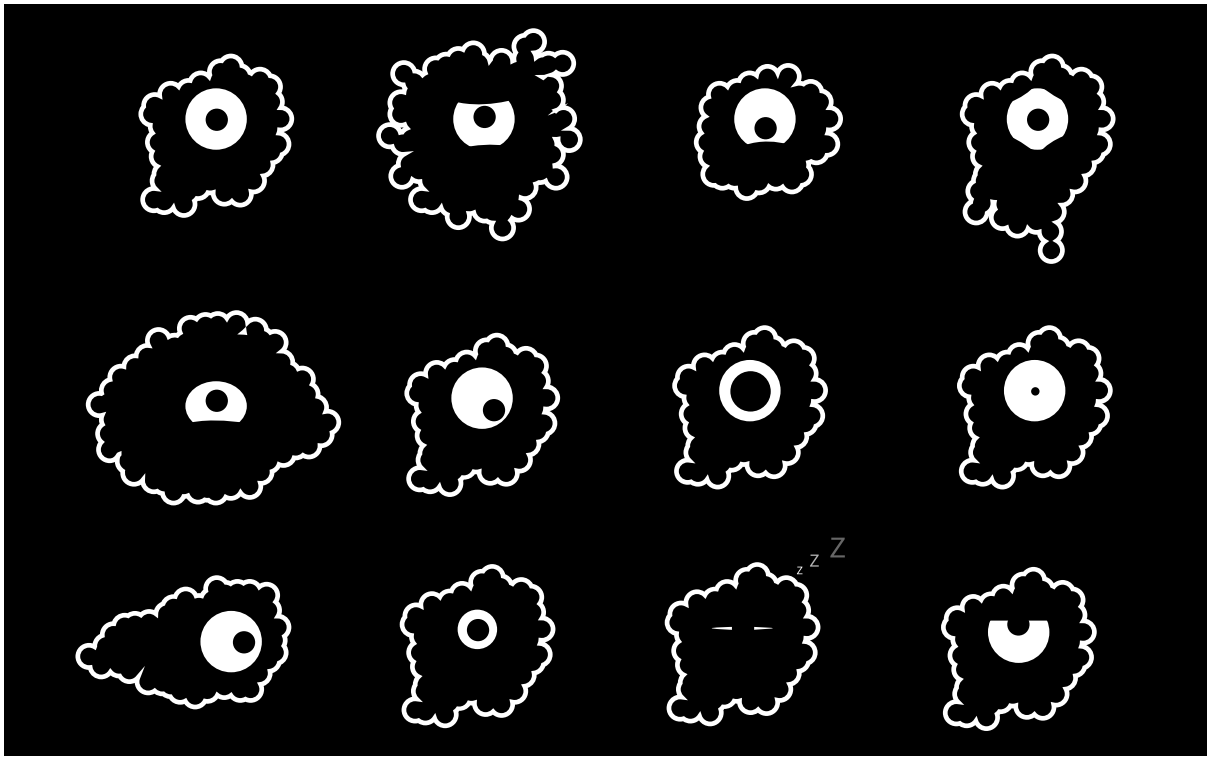


Diese Festlegung hielt uns natürlich nicht davon ab weiterhin mit verschiedenen „Wobbeln“ in allen möglichen Aggregatszuständen und Stilen herumzuexperementieren.

4.1.1.3. Finaler LOOMOX als Spielfigur

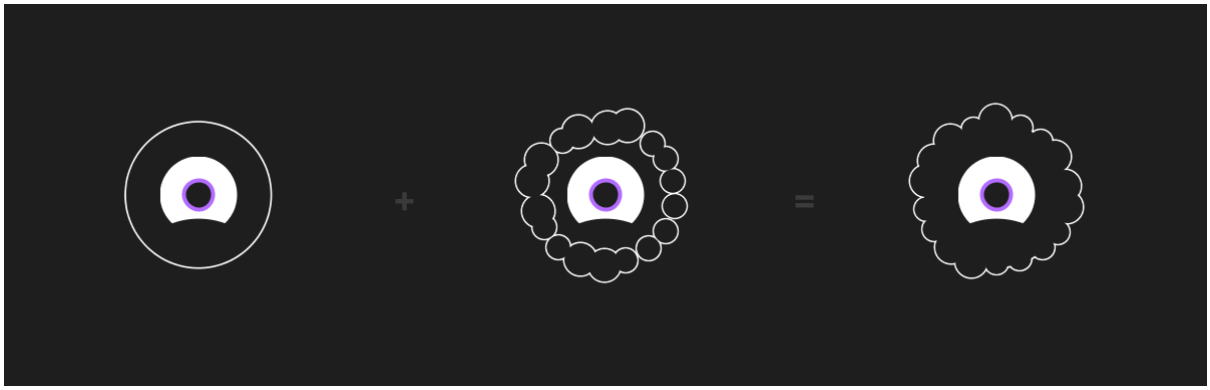
Für den Hintergrund der Fensterinstallation war zunächst eine weiße Fläche angedacht. Diese sollte durch ihre Leuchtkraft die Menschen auf die Installation aufmerksam machen und gleichzeitig einen starken Kontrast zum schwarzen “Wobbl” herstellen, sodass dieser auch bei schlechten Lichtverhältnissen gut erkennbar bleibt. Schnell jedoch stellten wir fest, dass diese fast schon “sterile” Umgebung nicht wirklich zum “Wobbl” passt und änderten die Farbe schließlich in ein sehr dunkles Grau. Damit die Spielfigur trotzdem sichtbar bleibt, gaben wir ihm eine weiß leuchtende Kontur. So wirkte das Wesen nicht mehr wie ein schwarzer “Kleks”, sondern bestand nur noch aus einem Kreis für das Auge und einer sich bewegenden, weißen Lichtkontur.





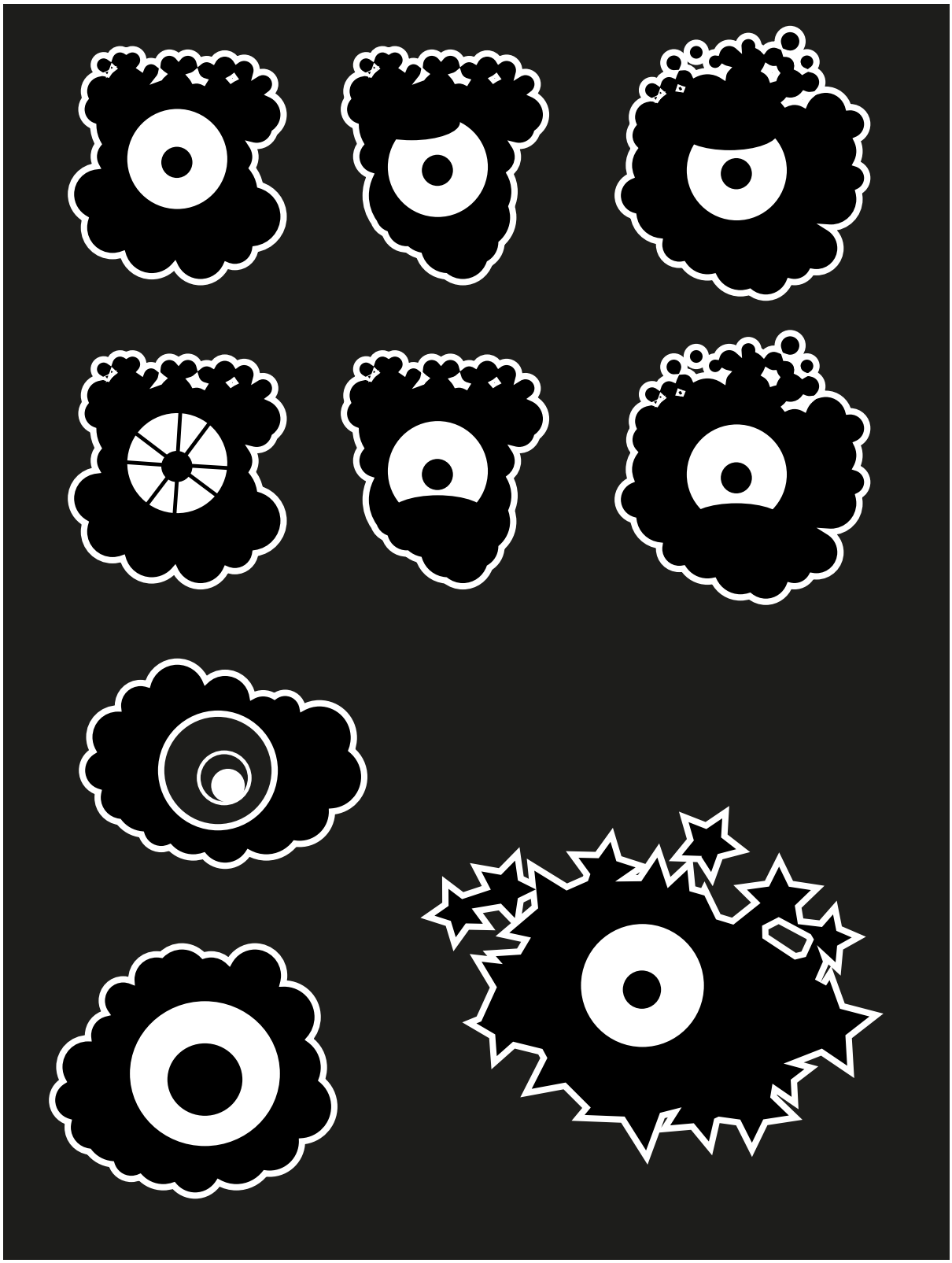
Diese Lichtkontur machte den Umriss des Körpers deutlich. Hierbei entstand auch der finale Name LOOMOX (von “lumen” bzw. “lux”, lat. für Licht).

Da der LOOMOX als Spielfigur eingesetzt werden soll, bekamen wir Probleme mit der unkonstanten Form des Wesens. Die wilden Formänderungen machten eine Kollisionsabfage auf der Technikseite fast unmöglich, bzw. unperformant.
Die Lösung war auch hier schnell klar. Die nahezu runde Körperform, die der LOOMOX im Ruhezustand einnimmt, sollte komplett beibehalten werden, ohne dabei den “wabbligen” Charakter der sich bewegenden Kontur aufzugeben. So definierten wir einen großen Kreis als Grundform des Wesens, auf dessen

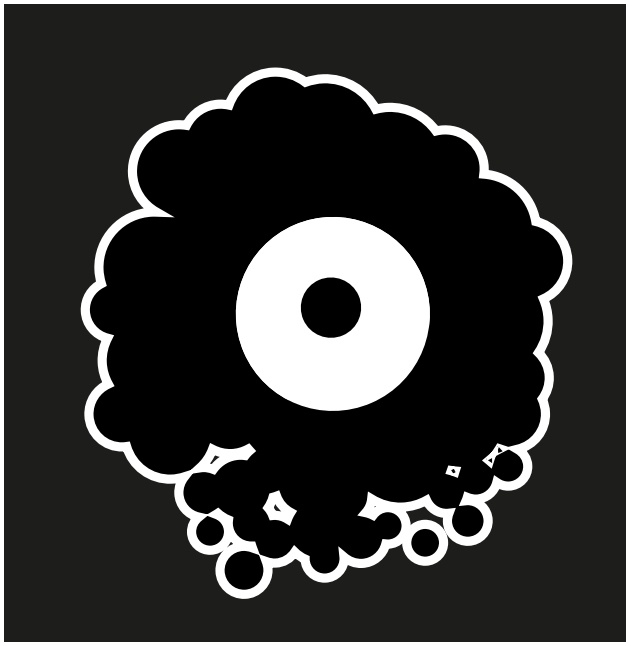


Umrissbahn sich immer noch die kleineren Kreise aus der ursprünglichen Version bewegen.

4.1.1.4 LOOMOX Stilfindung



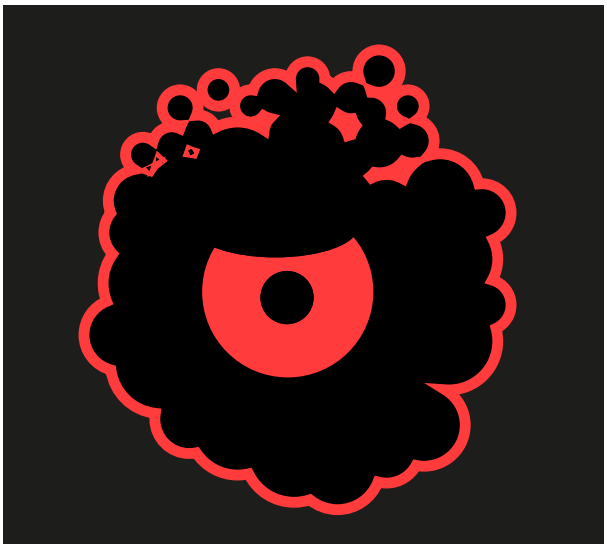
Bei der "Identitätsfindung" des Loomox entstanden viele verschiedene Körperformen. Wir kamen immer wieder zu dem Ergebnis, dass ein nahezu runder Körper dem Loomox am Besten stand. So entstand die erste Version des „Default-Loomox", der noch aus vielen kleinen Partikeln bestand, die eine Art Antrieb andeuten sollten.



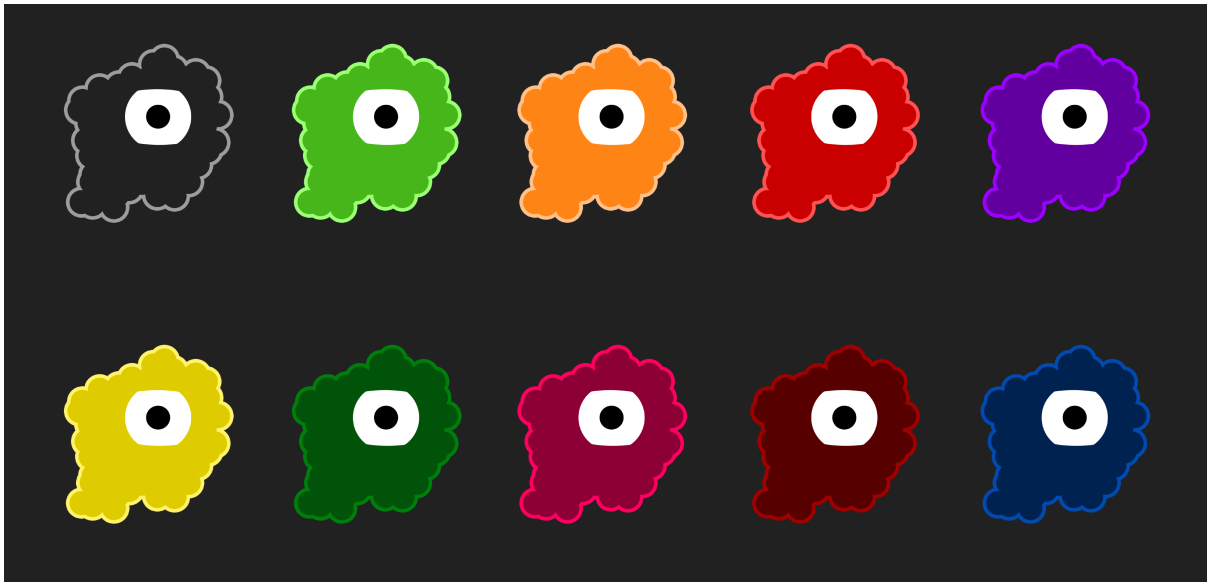
Um ein Gefühl dafür zu bekommen, in welcher Umwelt sich der Loomox befinden sollte, testeten wir den Loomox in unterschiedlichen Hintergründen. Durch diese Tests wurde uns aber immer wieder deutlich, dass wir unseren Loomox in eine minimalistische Umgebung aussetzen wollten, da detaillierte, vor allem farbenfrohe Hintergründe dem

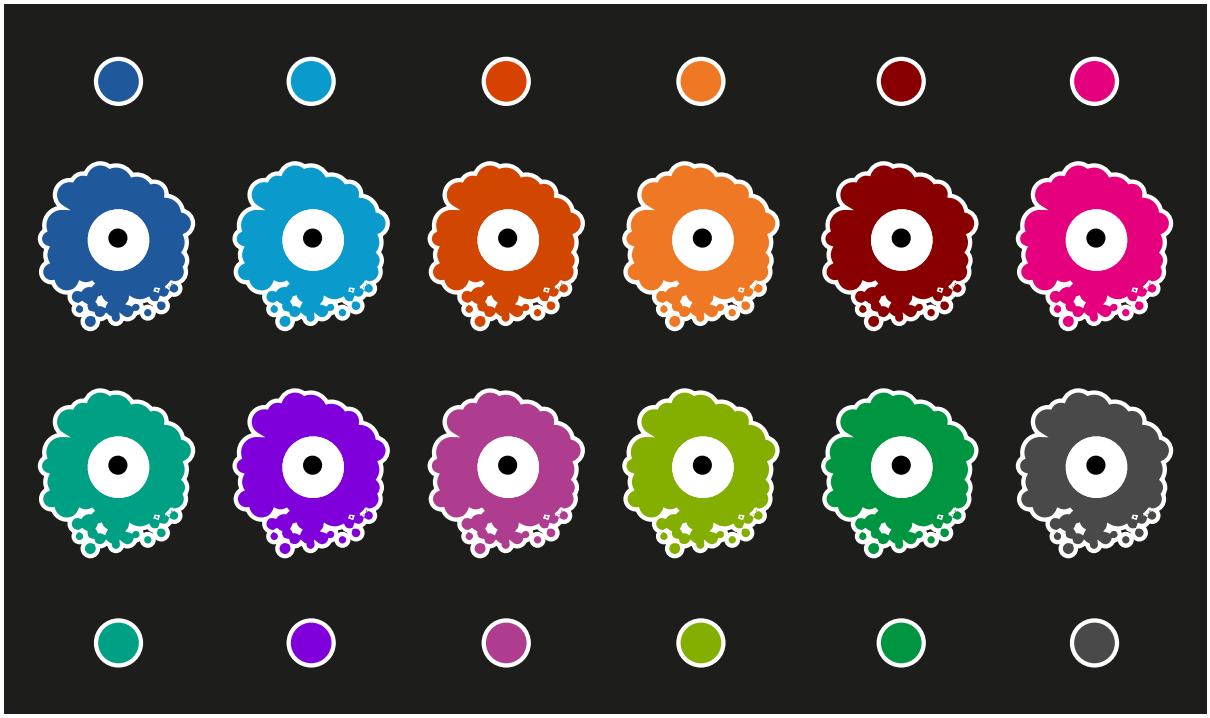
doch eher schlichten Loomox die benötigte Aufmerksamkeit stahlen.

Natürlich spielten wir auch mit Farben herum. Gerade für den Personalisierungsgedanken war die Möglichkeit die Farbe zu ändern ja quasi unerlässlich. So probierten wir farbige Konturen für den Loomox aus...



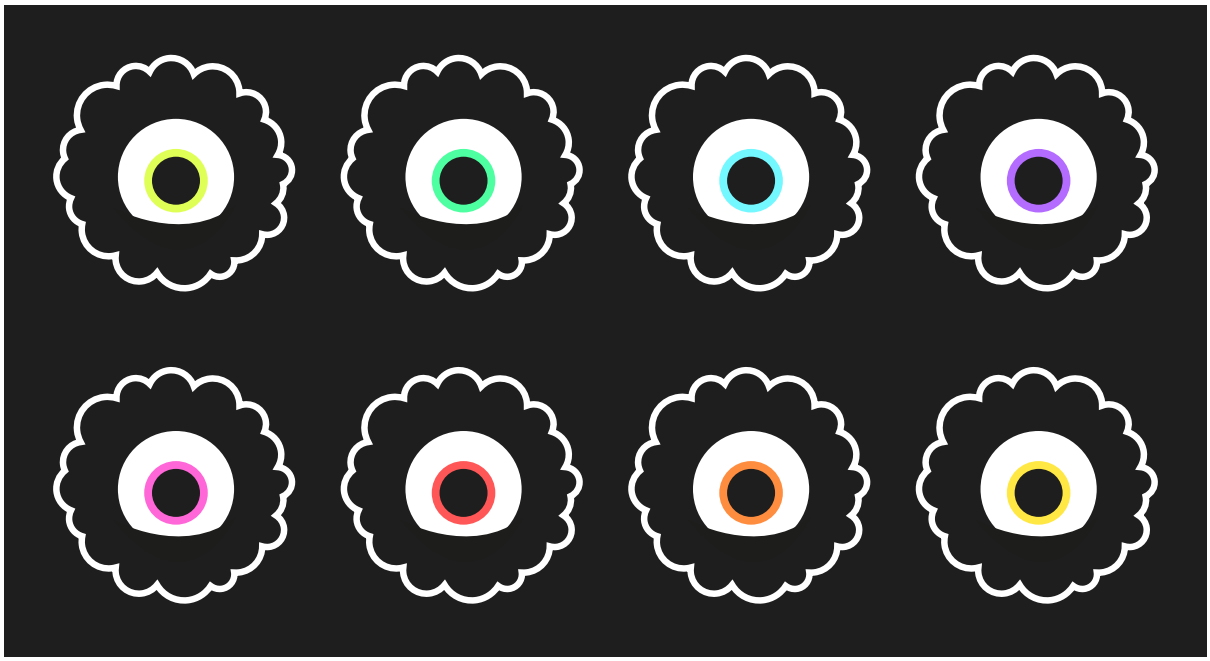
...aber auch verschiedenfarbige Körper.





Wir waren zwar nie richtig zufrieden mit den Ergebnissen, kamen aber anfangs noch zu keiner besseren Lösung, als den Körper des Loomox einzufärben und die weiße Kontur beizubehalten.

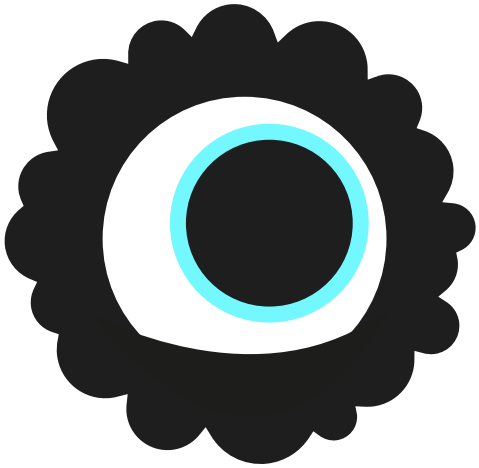
Doch dann kamen wir auf die Idee, der Pupille des Loomox eine Iris zu verpassen, die nun in unterschiedlichen Farben dargestellt werden konnte. Trotz dezentem Einsatz von Farbe ließ der Loomox so unterschiedliche Charakterzüge zu.



Wir stellten eine Farbpalette zusammen, die aus acht hellen, gut unterscheidbaren Farben bestand, um einerseits den Kontrast zum dunklen Hintergrund möglichst hoch zu halten, und andererseits, weil diese leuchtende Farben eine anziehende Präsenz ausstrahlen.

4.1.1.5 LOOMOX Emotions

Wir wollten das Experiment wagen, den LOOMOX trotz fehlender Gesichtspartien, abgesehen natürlich von dem einen Auge, mit einer Fülle von Emotionen auszustatten. Es war interessant zu sehen, wie viele Emotionen man mit einem einzelnen Auge und dessen Lidern vermitteln kann. Dabei spielten neben den Lidern die Faktoren Größe des Auges, Größe der Pupille sowie Position und Bewegung der Pupille eine unmittelbare Rolle. Ein großes Auge mit großer Pupille wirkte zum Beispiel immer gleich wie ein Baby-Loomox. Dadurch hatten wir schonmal einen Ansatz, um den Loomox niedlicher zu gestalten.

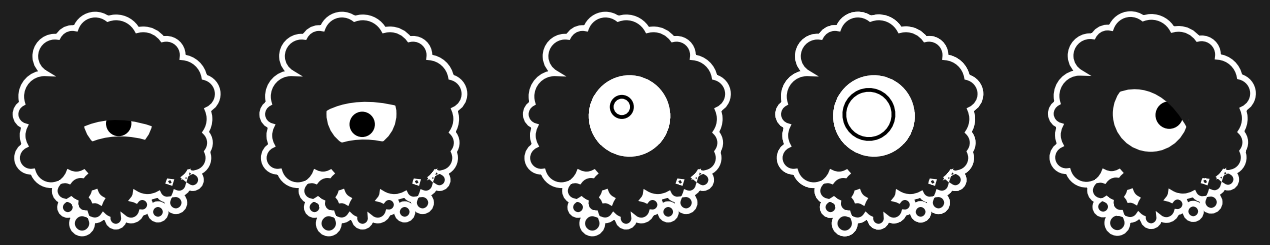


Baby-Loomox

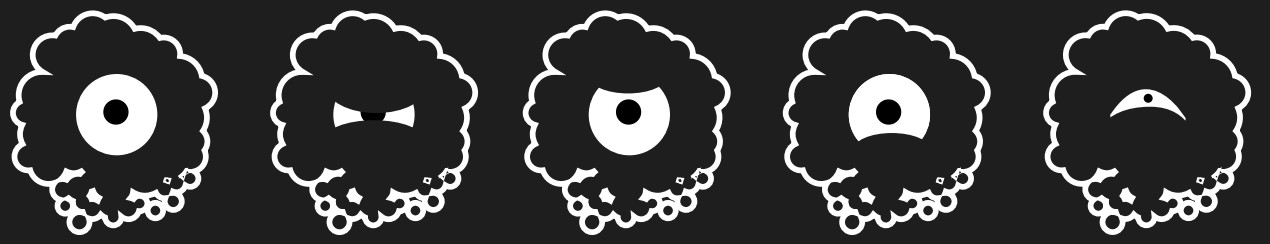
Manche Emotionen sind zwar zweideutig bzw. werden von unterschiedlichen Personen unterschiedlich gedeutet, fallen aber meist in eine ähnliche Kategorie. Was für den einen wie ein müdes Auge aussieht, interpretiert ein anderer als einen traurigen Loomox. Auf jeden Fall lässt sich behaupten, dass jede einzelne Loomox Emotion für jeden Menschen in jedem Fall emotional ansprechend ist und das ist unser Hauptanliegen gewesen.

VERWUNDERT | HASST DICH | TRÜBSINNIG / RESIGNIERT | ÄNGSTLICH | VERTRÄUMT

WÜTEND 1 | WÜTEND 2 | VORFREUDE | GENERVT 1 | GENERVT 2



MÜDE 1 | MÜDE 2 | GESCHOCKT 1 | GESCHOCKT 2 | ANGRIFFSLUSTIG



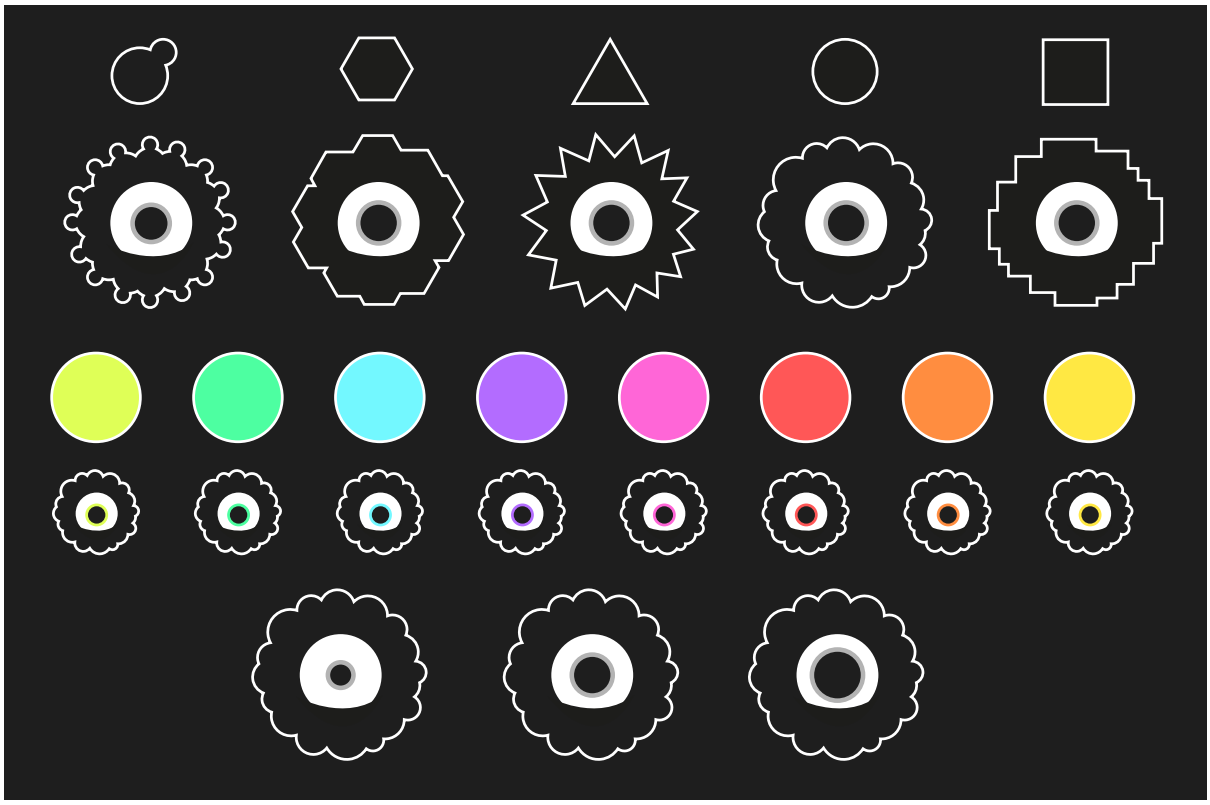
NORMAL | VERACHTET DICH | SAUER | SCHEU | ÄNGSTLICH

ROLLT MIT DEM AUGE | TRAURIG | VERSTOHLEN | MISSMUTIG/ MISSTRAUISCH | HINTERHÄLTIG

HINTERHÄLTIG | GIERIG | HOCHNÄSIG 1 | HOCHNÄSIG 2 | ERFREUT

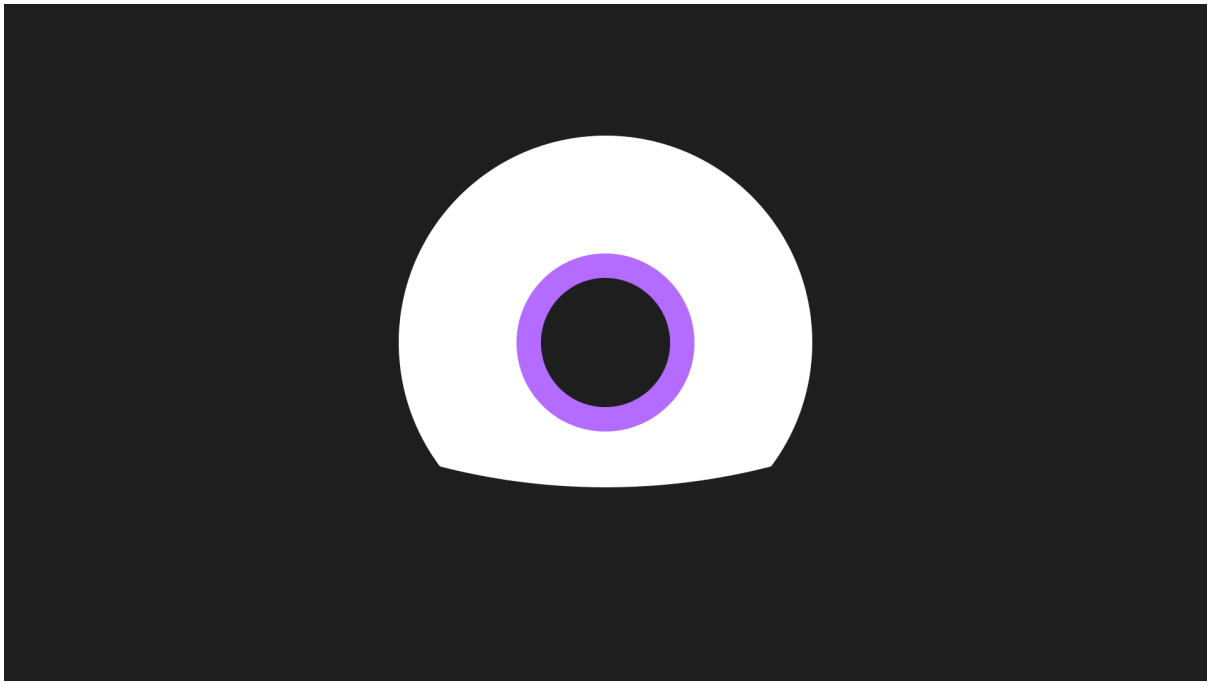
4.1.1.6 LOOMOX-Personalisierung

Unter dem Aspekt, dass man sich auch außerhalb der Installation mit dem LOOMOX beschäftigen können soll und um eine größere Abwechslung in die visuelle Ästhetik der Anwendung zu bringen, wurde die Personalisierung des LOOMOX eingeführt.
Hierbei legten wir uns auf drei Bereiche des Wesens fest, die man individuell anpassen kann:
Form der einzelnen Partikel des LOOMOX, Farbe der Iris und Pupillengröße.



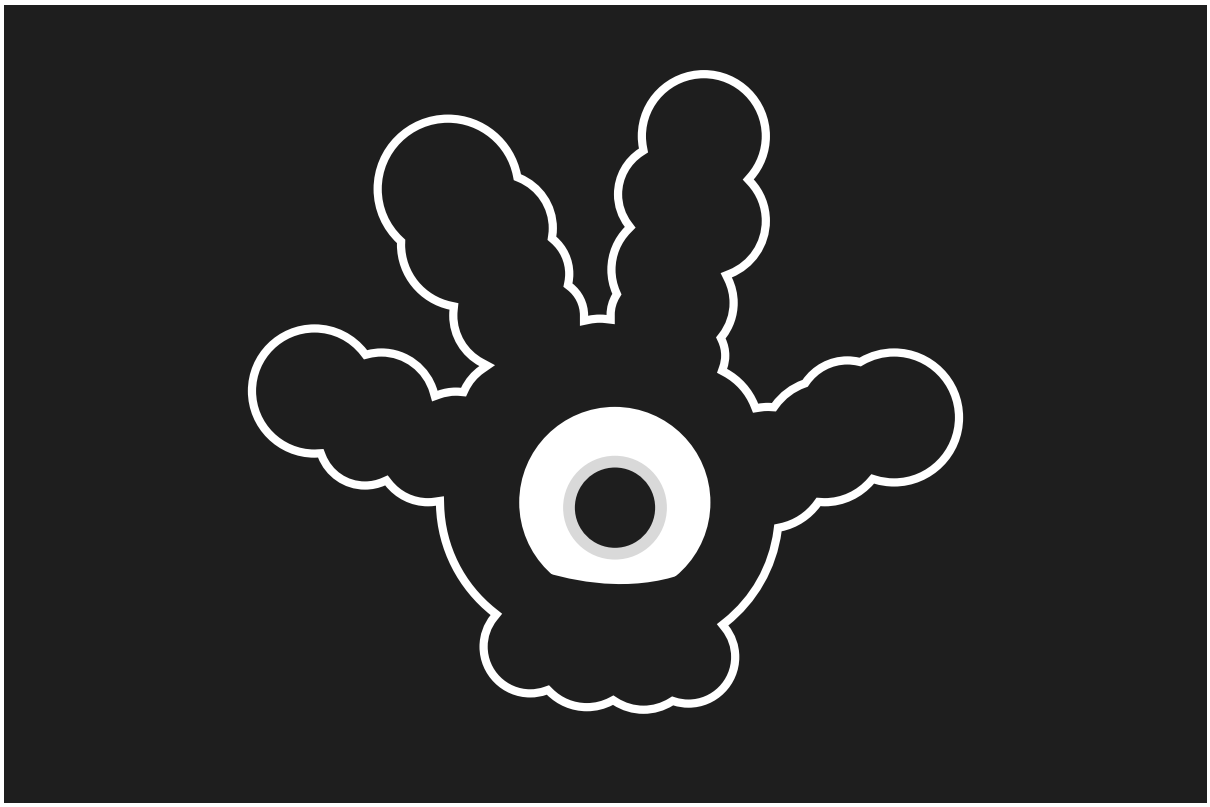
4.1.2 Game- und Interaktionsdesign

Um die Anwendung möglichst intuitiv bedienbar und leicht verständlich zu machen, wurde viel Zeit in die Ausarbeitung der Interaktionskonzepte gesteckt.



4.1.2.1 Erstkontakt

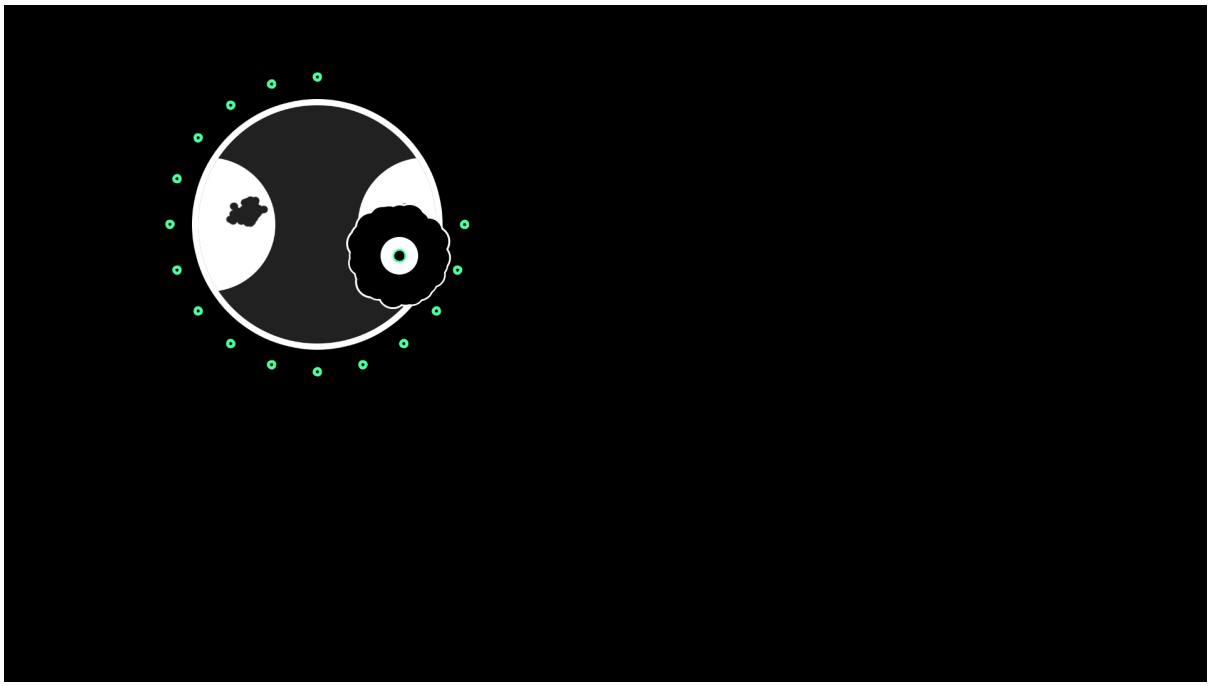
Solange sich kein Nutzer vor der Installation befindet, zeigt diese lediglich das große Auge des LOOMOX an, dass sich umschaut und ab und zu blinzelt.



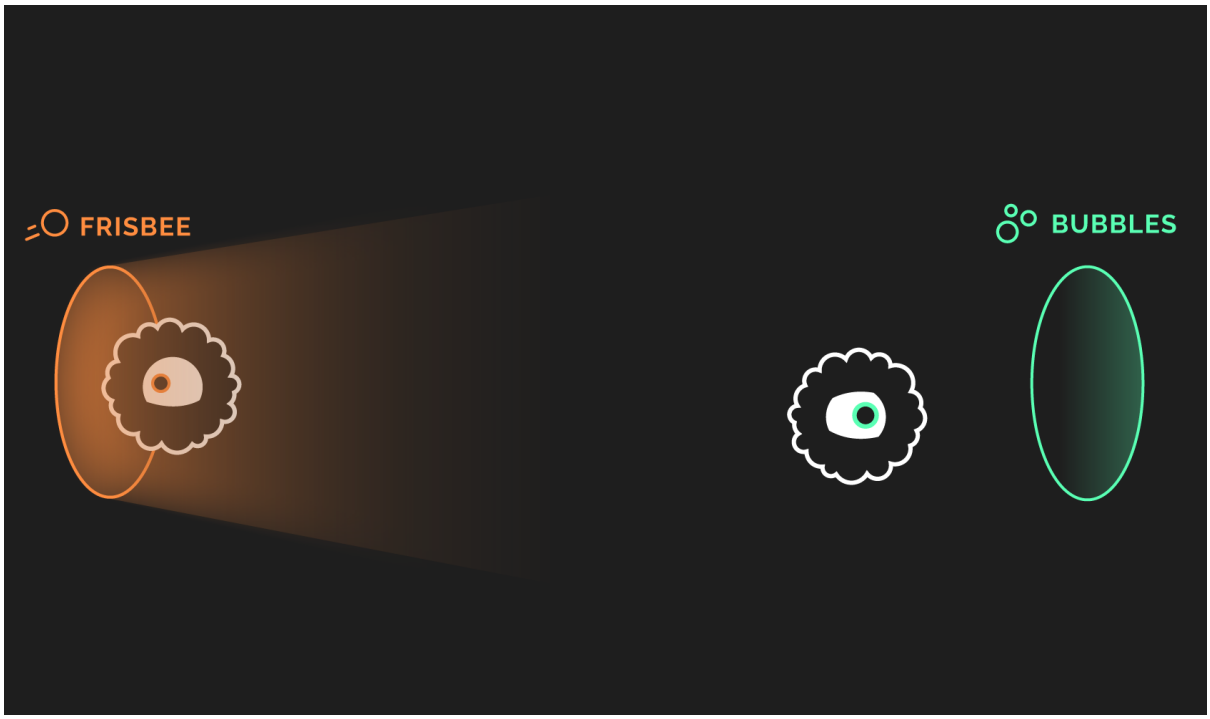
Sobald sich ein oder zwei Nutzer vor dem Fenster befinden, verkleinert sich das Auge entsprechend des Abstandes der Personen zum Screen. Sind die Personen nahe genug an die Anwendung herangetreten, wird die nächste Interaktionsphase eingeleitet und aus dem Auge entsteht ein kompletter LOOMOX samt Kontur. Diese Kontur verwandelt sich nun in eine handähnliche Form und das Wesen beginnt, dem Nutzer zuzuwinken. Sobald die Person die Hand zum Winken hebt, wird diese von der Microsoft Kinect erkannt und als Steuerungshand für den LOOMOX interpretiert. Er folgt nun dieser Hand und kann so über Armbewegungen gesteuert werden.

4.1.2.2 Spielauswahl

Folgt der LOOMOX der Spielerhand, befindet man sich nach kurzer Zeit in der Spieleauswahl. Um das visuelle Kernelement, den Kreis, wieder aufzugreifen, war zunächst angedacht, die Auswahl über kreisförmige Spielsymbole zu gestalten. Auf diesen musste man zur Bestätigung seiner Entscheidung lange genug mit der Spielfigur verweilen und startete so das jeweilige Spiel.
Ohne Erklärung versuchten die Nutzer jedoch ständig, durch diverse Aktionen oder Gesten ihre Auswahl zu bestätigen. Das



Konzept des Verweilens auf einer Fläche war also nicht intuitiv genug.



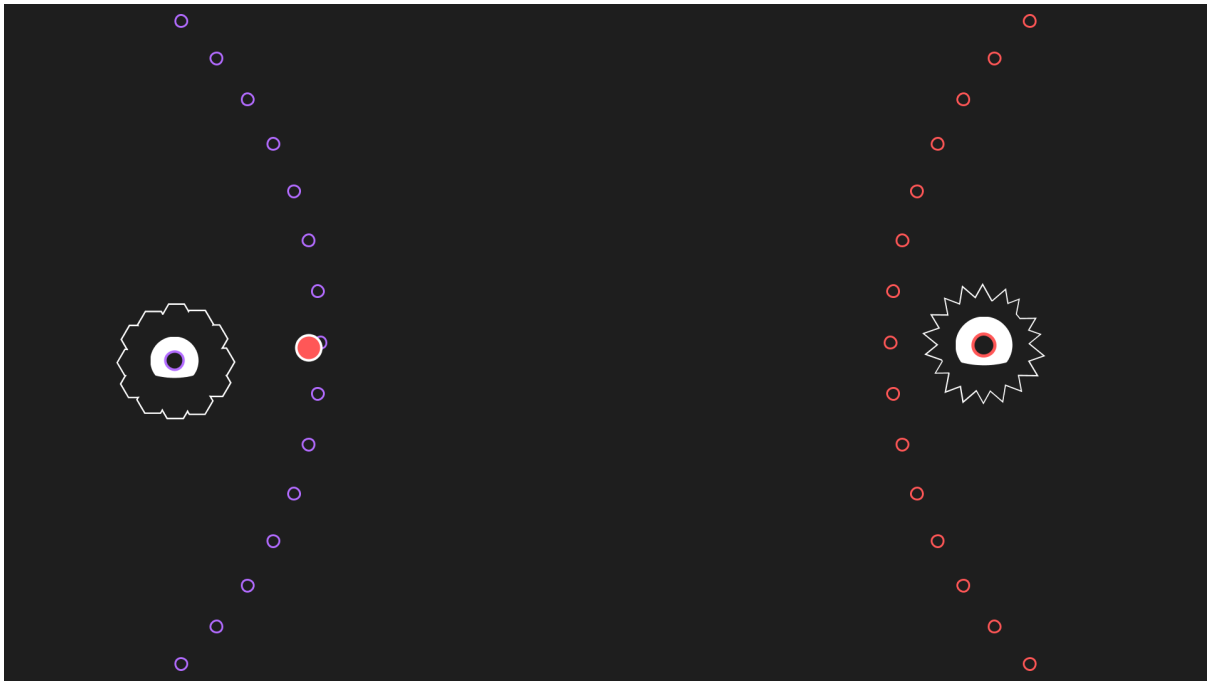
Die Kreise wurden schließlich durch Portale ersetzt, durch die man einfach mit seinem LOOMOX hindurchgehen kann. Sind zwei Spieler gleichzeitig auf dem Auswahlscreen muss der erste im Portal warten, bis beide hindurch können.

4.1.2.3 Spielkonzepte

Bei den Spielen wurde darauf geachtet, sie möglichst einfach und kurz zu gestalten. Sie sollten für einen Zwischenstopp an der Hochschule leicht verständlich sein, sodass jeder mit der Installation zurecht kommt.

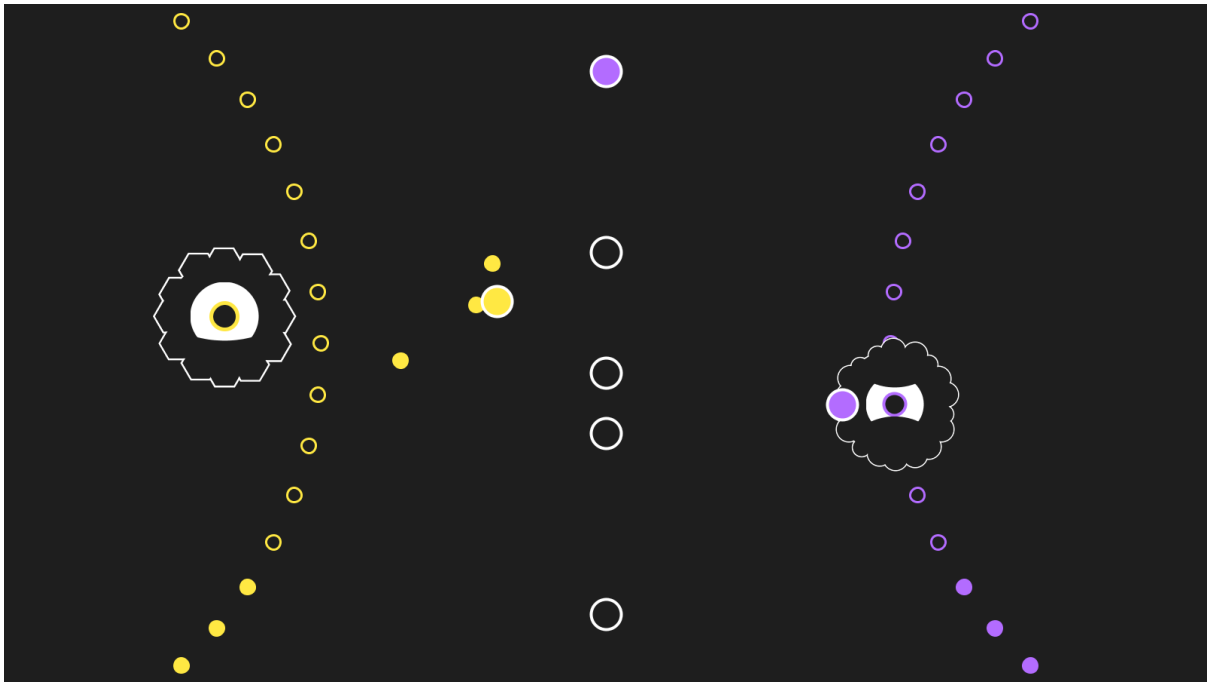
4.1.2.3.1 Frisbee

Frisbee ist ein an Pong angelehntes Spiel. Die Spieler links und rechts versuchen beim jeweils anderen ein Tor zu schießen. Es gibt nur eine Frisbee. Wer zuerst 5 Tore geschossen hat, gewinnt.



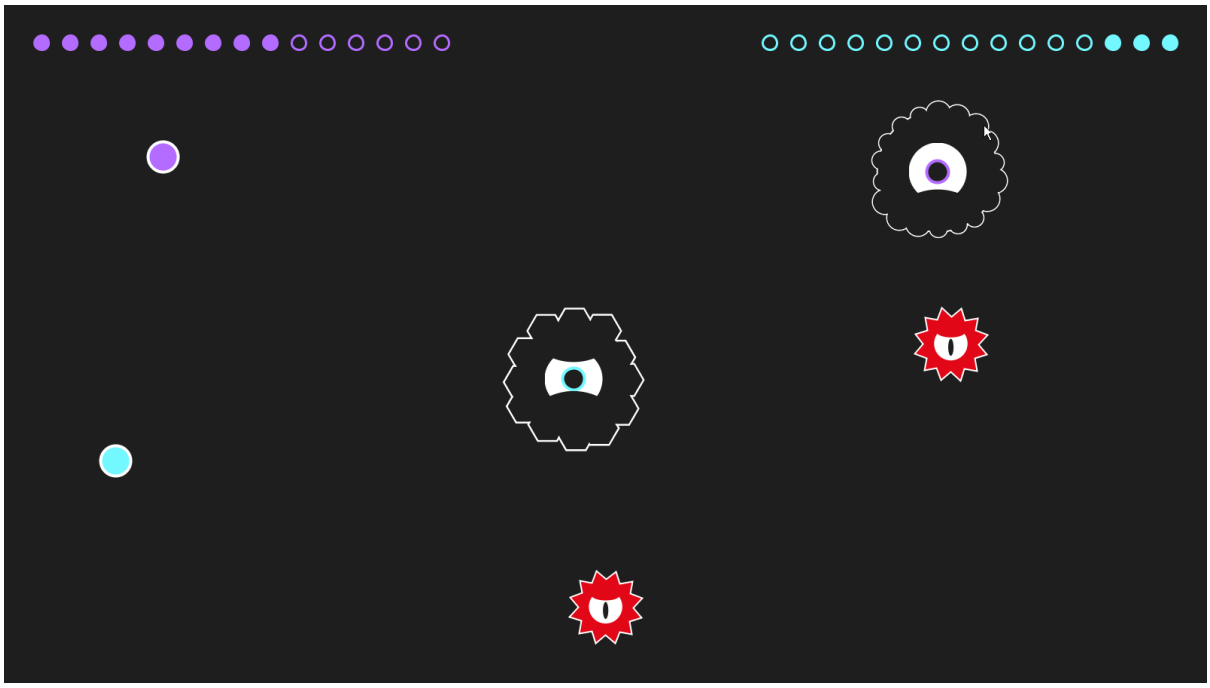
4.1.2.3.2 Bubbles

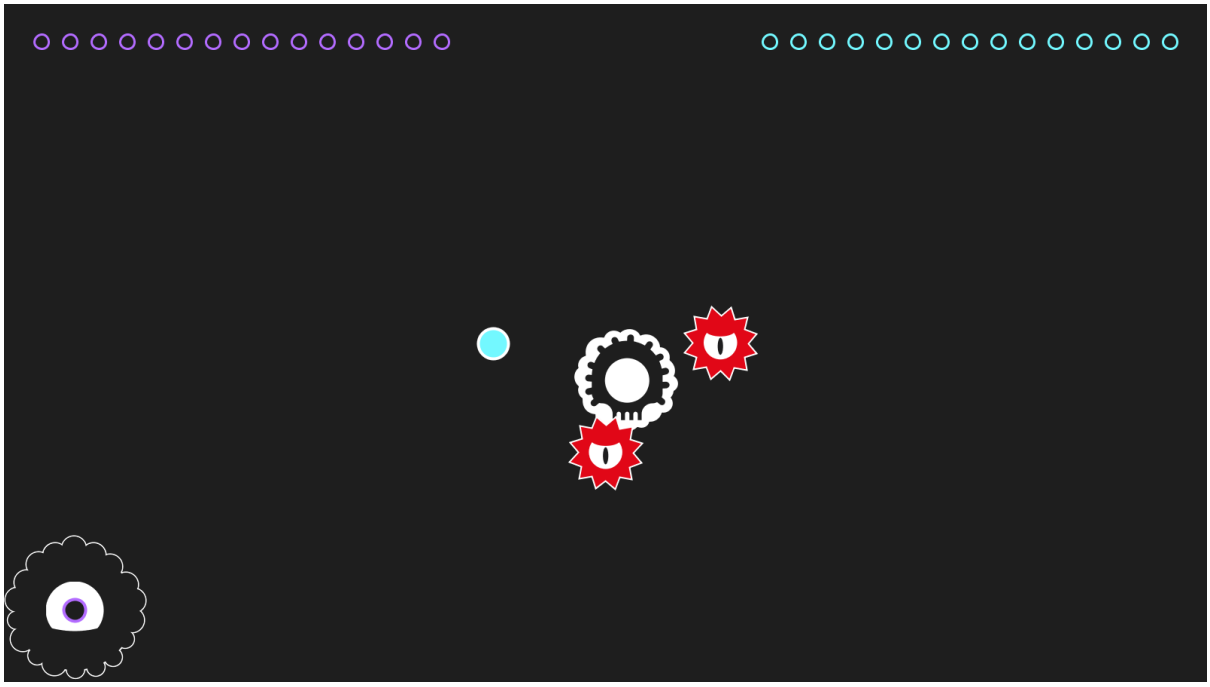
Bei diesem Spiel stehen beiden Spielern unendlich viele Frisbees zur Verfügung. Diese werden nach einem Schuss eine kurze Zeit lang nachgeladen. Wer die meisten "Bubbles" mit seinen Frisbees "einfärbt", bevor sie am oberen Rand ankommen, gewinnt. Dabei ist zu beachten, dass man eine bereits eingefärbte Bubble des Gegners auch „klauen" kann (also neu einfärben).



4.1.2.3.3 Monsters

Hierbei handelt es sich um ein Spiel ähnlich “Snake”. Man muss versuchen, möglichst
viele Frisbees der eigenen Farbe einzusammeln, bevor der Gegner die Maximalpunktzahl
erreicht. Dabei werden die Spieler von zwei Gegnern gestört, die dem LOOMOX bei Berührung einen Stromschlag versetzen.





4.1.3 Corporate Identity

4.1.3.1 Exposé Design

LOOMOX

Medienprojekt
Studiengang Interaktive Medien
Hochschule Augsburg

Projektbetreuer:

Prof. Dr. Michael Kipp
michael.kipp@hs-augsburg.de

Prof. Dr. Kipp forscht an Themen der Mensch-Computer Interaktion und multimodaler Kommunikation. Er ist (Ko-)Autor von über 70 internationalen Publikationen und ist als Gutachter u.a. für die EU, den Schweizerischen Nationalfonds, mehrere Fachzeitschriften und internationale Konferenzen tätig.

Prof. Daniel Rothaug
daniel.rothaug@hs-augsburg.de

Prof. Rothaug ist Spezialist für Interfacegestaltung und Mitgründer des vielfach ausgezeichneten Würzburger Designbüros „Zum Kuckuck". Zu seinen Kunden zählen namhafte Unternehmen wie die Telekom, Toshiba, Volkswagen, uvm

Eine interaktive Fassaden-Installation

Als Projektgruppe der Hochschule Augsburg werden wir in den kommenden acht Wochen eine interaktive Installation an der Fassade der Hochschul-Bibliothek auf die Beine stellen, die das Konzept des Voneinanderlernens auf spielerische Art vermittelt und dabei eine Brücke zwischen realer Umgebung und digitaler Welt schafft

Die Installation

Mittels eines Beamers und einer Rückprojektionsfolie wird ein einäugiges Wesen, ein sog. Loomox, an die Außenfassade der Hochschulbibliothek projiziert, dass Passanten begrüßt und zu einem interaktiven Spiel auffordert.

Die Interaktion

Geht man auf das Wesen an der Fassade ein, hat man die Möglichkeit dieses per Gestensteuerung durch eines von zehn kurzweiligen Spielen zu führen

Die Mitnahme

Wer sich auf das Wesen einlässt, erhält die Chance, dieses durch Eingabe eines Codes auf der Projektwebsite mit nach Hause zu nehmen, für sich zu beanspruchen und zu personalisieren

Die Rückkehr

Wann immer das Wesen und sein Besitzer zur Hochschule zurückkehren, können sie an der Fassade der Bibliothek im Duell gegen den Computer oder einen realen Gegner ein neues Spiel erlernen

Möchten Sie bei diesem Projekt mitwirken?

Die Realisierung unserer Projektidee erfordert einen hohen technischen und finanziellen Einsatz. Benötigt wird u.a. ein leistungsstarker Beamer, eine großflächige Rückprojektionsfolie, ein Rechner, eine Webcam und vieles mehr. Bisher konnten wir mit Prototypen und konzeptionellen Überlegungen sehr viel erarbeiten. Für die nächsten Schritte freuen wir uns über jede Art der Unterstützung - egal, ob Geld- oder Sachwerte!

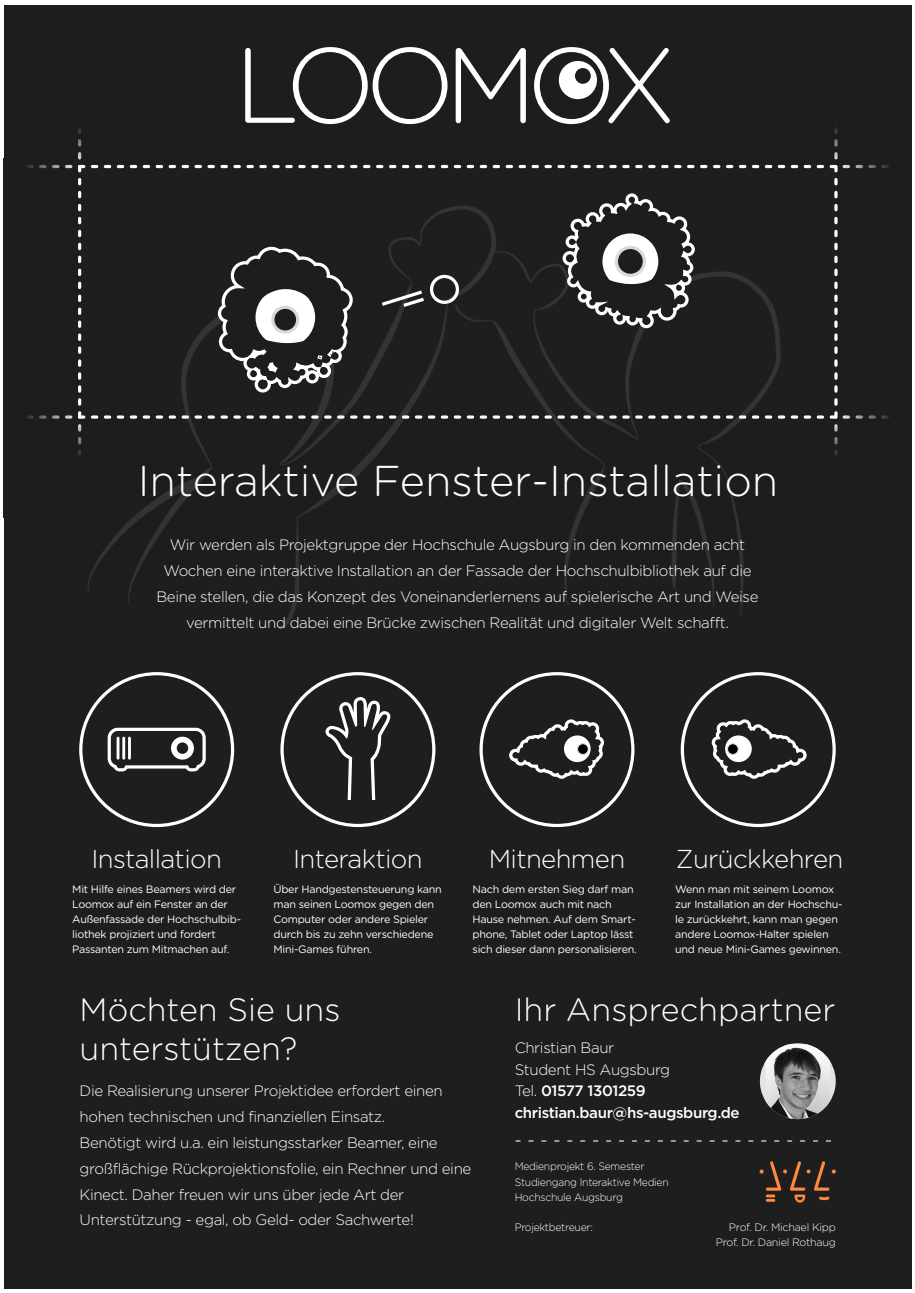
Ihr Ansprechpartner:

Christian Baur
Student HS Augsburg
Tel.: 01577 130 1259
E-Mail: christian.baur@hs-augsburg.de

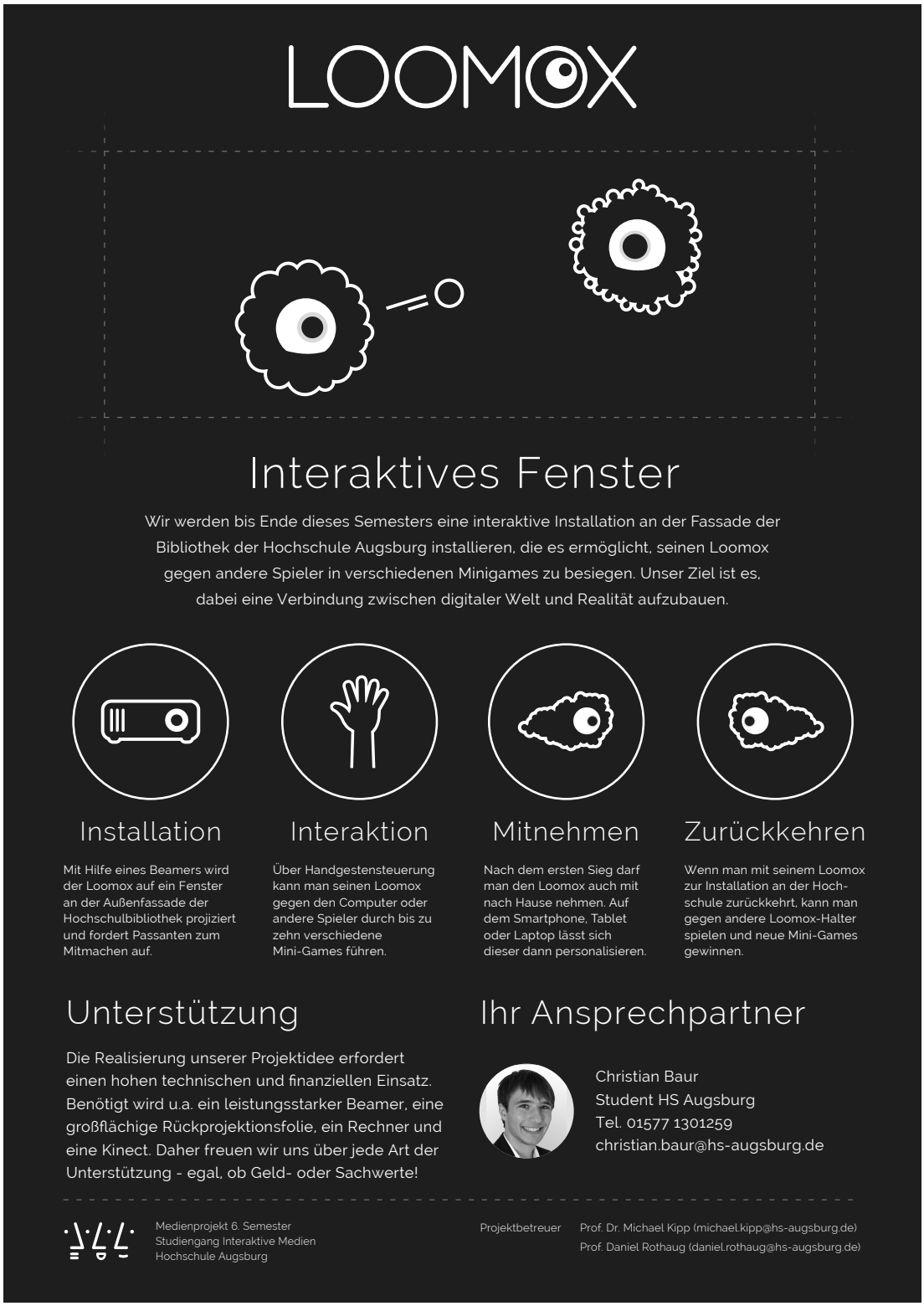
Personen-Silhouetten von *vecteezy.com*

Bereits in einem frühen Stadium des Projekts erarbeiteten wir ein Exposé, mit welchem wir primär Sponsoren finden wollten und das uns als Grundlage zur Entwicklung einer Corporate Identity diente. Die Illustration am Anfang des Dokuments sollte unser Fenster an

der Hochschule zeigen, gleichzeitig wollten wir durch die Figuren im Hintergrund aber auch darstellen, dass die Spielfiguren durch Menschen gesteuert werden. Wir wussten, dass wir eine klare, dünne Schriftart verwenden wollten, die sich aber von der neutralen, strengen Helvetica unterscheiden würde und etwas “lockerer” wirkte. So entschieden wir uns anfangs für die Gotham Rounded in der Stärke “Light”, welche durch ihre leichte Abrundung nicht zu seriös wirkte. Aus lizenzrechtlichen Gründen konnten wir diese Schriftart allerdings nicht öffentlich verwenden und mussten somit weitersuchen.



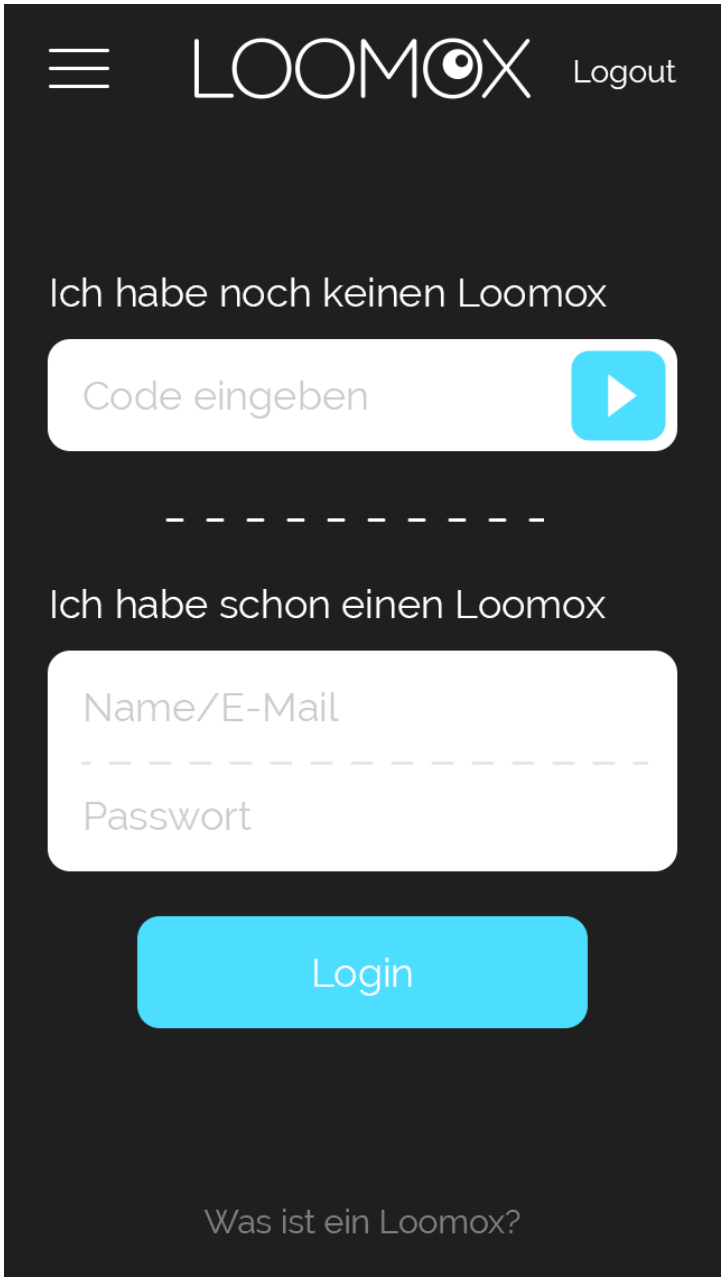
Im nächsten Entwurf verwendeten wir die kostenlose Schriftart Museo, die wir auf Grund seines verspielten, dynamischen und originellen Charakters wählten. Wir versuchten die Museo mit einer anderen Schriftart für die Überschriften zu kombinieren, waren aber nie vollständig zufrieden mit dem Ergebnis.



Nachdem wir den kompletten Fontpool der Hochschule Augsburg ausgeschöpft hatten, entschieden wir uns letztendlich für die Schriftart Raleway, eine kostenlose Googlefont, die auch in Spielen wie "Dots" und "Two Dots" verwendet wird. Diese Schriftart fanden wir sehr passend, da sie unserer ersten Wahl, der Gotham, wieder sehr nahe kam, aber noch einen Tick verspielter wirkte. Mit etwas mehr Laufweite und den Schnitten Light für Überschriften und Regular für Fließtext waren wir schließlich sehr zufrieden.

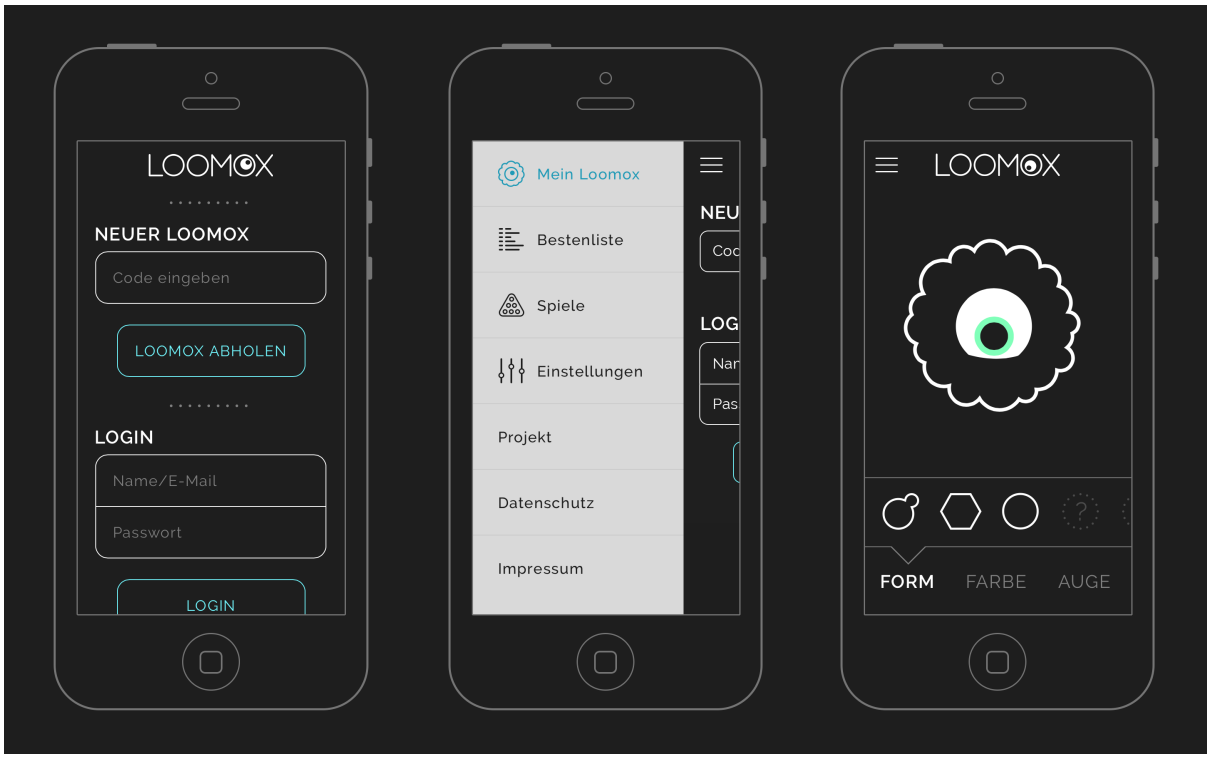
4.1.3.2 Web App Design

Beim ersten Entwurf für die Web App dominierten noch Flächen das Erscheinungsbild, z.B. weiße Flächen für die Textfelder oder blaue Flächen als Buttons.

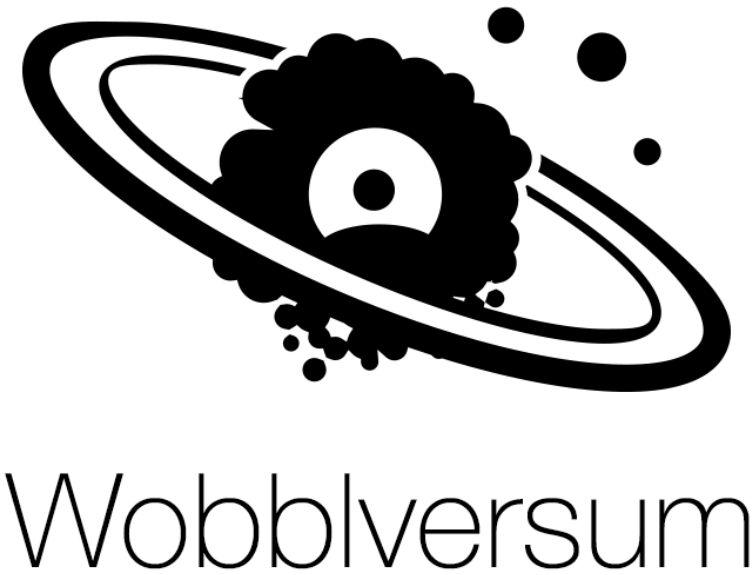


Als sich das Erscheinungsbild der Installation immer mehr festigte, kamen wir aber zu dem Schluss, dass die Flächen nicht mehr ins Konzept passten, da wir sonst in allen Entwürfen fast ausschließlich weiße Konturen verwendeten. Dies beruhte darauf, dass der LOOMOX seine Präsenz am Besten entfalten konnte, wenn er an sich keine Fläche besaß und die

dunkelgraue Hintergrundfarbe annahm. Dieses Konzept zieht sich durch jedes Dokument, jeden einzelnen Screen und jedes Icon.



4.1.3.3 Logo Design



4.1.3.3.1 Wobblversum

Unsere ersten Entwürfe kreisten um das Thema Universum. Da dieser Ansatz verworfen wurde, wurden auch diese Logos verworfen. Zu diesem Zeitpunkt fand auch das ausgedehnte Namensbrainstorming noch nicht statt, weshalb hier auch noch der Arbeitsbegriff “Wobbl” zu finden ist.

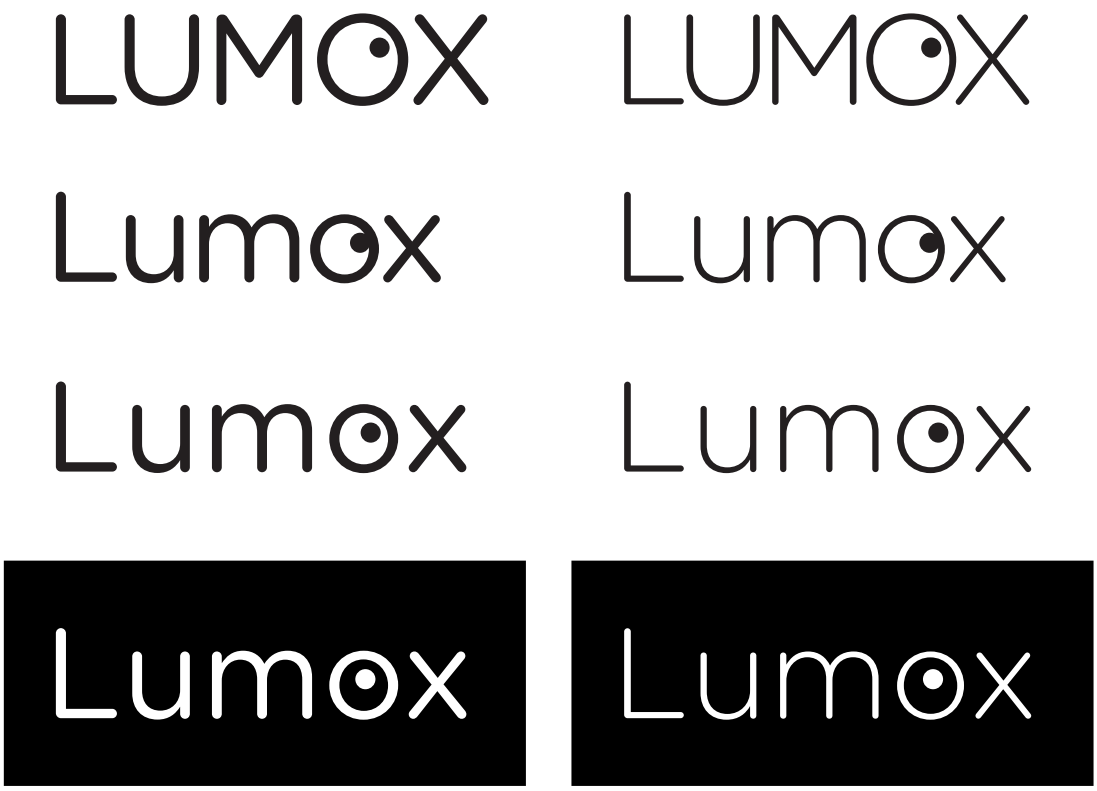
4.1.3.3.2 Speculum

SPECULUM

Der zweite Ansatz als Name für unsere Installation war “Speculum”. Dieser Begriff leitet sich aus dem lateinischen Wort “speculum” ab, was in diesem Zusammenhang für Spiegel/Fenster stehen würde. Gleichzeitig sollte der Begriff die Konnotation mit “Spektakel” hervorrufen.

4.1.3.3.3 Lumox

Eine Alternative zu unserem letztendlich gewählten Namen war “Lumox”, der den Begriff “Lumen” (lat. für Licht) noch stärker herausstellte. Allerdings fiel uns eine starke Ähnlichkeit zu der beliebten “LUMIX” Serie der Panasonic Fotokameras auf. Um den Begriff mehr zu verenglischen und einen Hauch mehr “Coolness” zu verleihen, entschieden wir uns schließlich für “LOOMOX”.



4.1.3.3.4 LOOMOX

LOOMOX sollte zunächst nur unsere Spielfigur heißen. Nachdem der Name allerdings bei allen gut ankam, entschieden wir uns diesen Begriff auf unser gesamtes Projekt auszudehnen. Das finale

Logo basiert auf dem ersten Entwurf, in welchem wir die Gotham Rounded einsetzten, wurde auf dieser Grundlage aber in Illustrator komplett neugebaut und für den Einsatz als Logo optimiert.

4.1.3.4 Storyboard

Um den genauen konzeptionellen und gestalterischen Ablauf vor der Installation planen zu können, benötigten wir ein Storyboard, das alle Schritte der Interaktion, vom Erstkontakt über die Spieleauswahl und das Mitnehmen bis zur Rückkehr enthält. Dieses wurde im Laufe des Semester ständig an die aktuellen technischen und konzeptionellen Entwicklungen angepasst.

4.1.3.5 Items

Die Items, die am Anfang als Früchte angedacht waren, mit denen der LOOMOX gefüttert wird, wurden im Laufe der Zeit zu Items umfunktioniert, die das Aussehen des LOOMOX unmittelbar verändern. So macht ein Dreieck den LOOMOX zu einem dreiecksförmigen LOOMOX, ein grünes Auge gibt ihm ein Auge mit einer grünen Iris, ein großes Auge eine große Pupille.

4.1.3.6 Das LOOMOX Universum

Nachdem das Aussehen und die Bewegung des LOOMOX grob festgelegt worden war, musste noch ein Design für die Welt gefunden werden, in der sich der LOOMOX befindet. Unser Gedanke, dass der LOOMOX aus einem (Parallel-)Universum stammt, das uns Menschen nicht zugänglich ist, brachte uns auf die Idee, den LOOMOX in ein Setting zu platzieren, das unserem Orbit ähnelt. Da dies jedoch die Aufmerksamkeit des Betrachters zu sehr vom LOOMOX abgelenkt hätte, verwarfen wir die Idee wieder und konzentrierten uns weiterhin auf unser minimalistisches und dynamisches Design.

4.2 Technik

4.2.1 Screen Anwendung

Die LOOMOX Screen Anwendung wurde mithilfe von Java und der Processing Library realisiert, da diese schon viele Möglichkeiten zur grafischen Gestaltung zur Verfügung stellt. Um die Programmstruktur aufrecht zu erhalten und alle Java Klassen einsetzen zu können, kamen verschiedene IDEs zum Einsatz, unter anderem Netbeans und IntelliJ.
Zur Codeverwaltung verwendeten wir das SVN Repository der Hochschule Augsburg.

Als Steuerungskomponente der Anwendung wurde die Microsoft Kinect gewählt, um eine möglichst direkte Steuerung ohne die Verwendung von Eingabegeräten umzusetzen.

4.2.1.1 Tracking

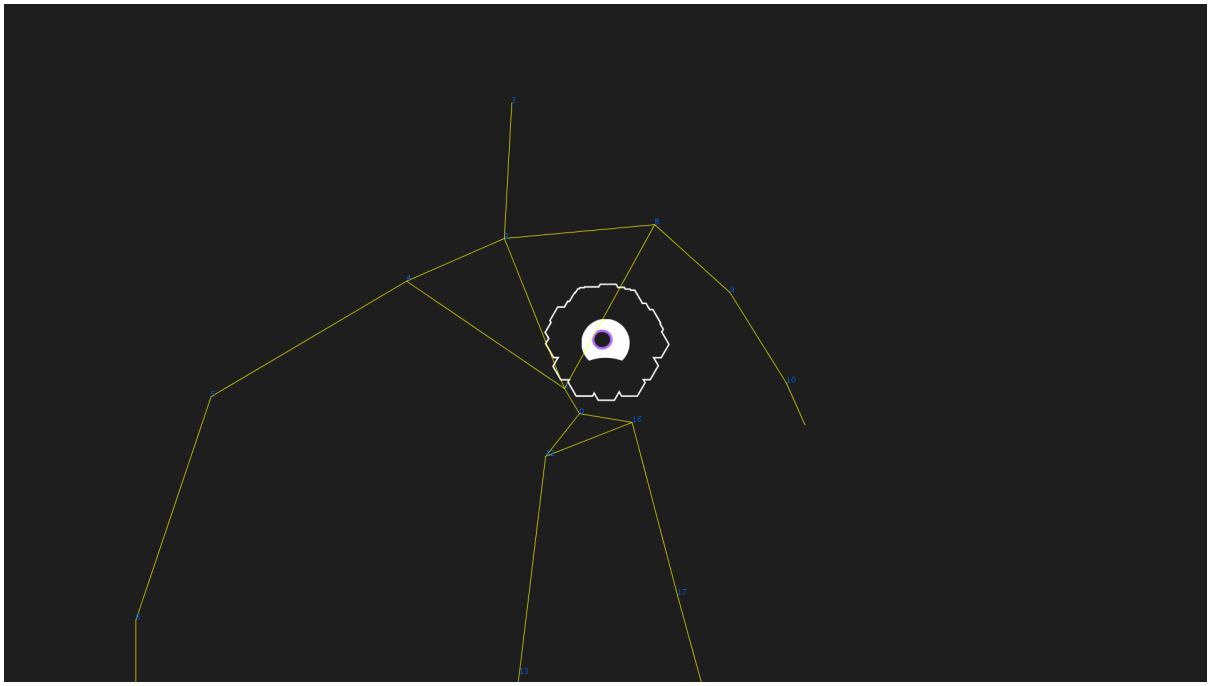
Um die Kinect in Processing verwenden zu können, benötigt man neben dem offiziellen Windows Kinect Treiber eine zusätzliche Java Bibliothek. Hier entschied man sich für die PKinect Library von Brian Chung.
Zum Tracking wurden Skeletons verwendet, welche automatisch beim Erkennen einer Person erstellt werden. Diese besitzen eine eindeutige ID sowie ein Bonesystem, welches die Positionen aller Gelenke angibt. Hiermit konnte eine eindeutige Zuordnung von Spieler zu LOOMOX und die Steuerung über einen der TrackingBones gewährleistet werden.

Jedoch bringt die Verwendung der Skeletons auch einige Nachteile mit sich. Zum einen verfügtt die Kinect nur über eine geringe Auflösung von 640x480 Pixel. Dies führt bei der Übertragung auf eine größere Auflösung zu Ungenauigkeiten, kann jedoch nicht behoben werden.

Zum anderen verfügt die Kinect am PC nur über eine Framerate von 10 Bildern pro Sekunde. Da die Anwendung allerdings mit mindestens 30 Frames pro Sekunde laufen muss, hatte dies zunächst eine sprunghafte Bewegung des LOOMOX zufolge. Um dem entgegen zu wirken, wurden jeweils die letzten fünf Positionen der getrackten Skeletons zwischengespeichert und daraus ein Mittelwert berechnet um die Bewegung flüssiger erscheinen zu lassen.

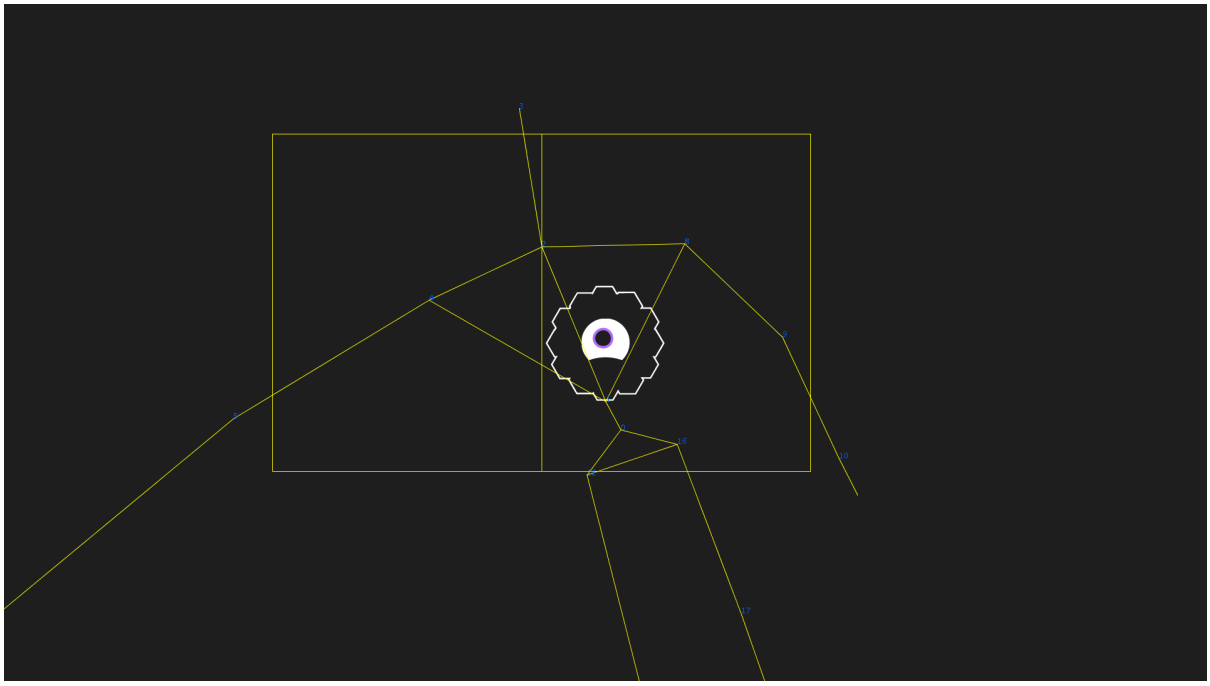
Um den LOOMOX zu steuern wurde zunächst eine direkte Projektion des Kinectbildes mit den Skeletons der Spieler auf die Anwendung verwendet. Befand sich also die Hand des Spielers in der rechten oberen Ecke des von der Kinect aufgenommen Bildes befindet sich der LOOMOX in der selben Ecke des Bildschirms. Hierdurch ergaben sich mehrere Probleme. Ist der Spieler weit von der Kinect entfernt, so ist er nur klein auf dem aufgenommen Bild und muss somit hin und her laufen um die seitlichen Ränder zu

erreichen und kann den LOOMOX nicht bis zum oberen Rand der Anwendung steuern, was gerade für kleine Menschen und Kinder eine große Hürde darstellt. Des Weiteren mussten mehrere Nutzer, die gemeinsam spielten, aneinander vorbeilaufen um die Seiten zu erreichen, was das Tracking der Kinect beeinträchtigen kann.



Um diese Probleme zu lösen wurde die direkte Projektion durch eine lokale Steuerung im Verhältnis zu Körpergröße und Armlänge des Spielers ersetzt. Dabei wird die Entfernung zwischen den Trackingbones des Genicks und der Hüfte berechnet. In Abhängigkeit von dieser Länge wird jeweils ein Bereich um die rechte und linke Körperseite des Spielers berechnet. Die Position der rechten bzw. linken Hand in diesem Bereich gibt nun die Position des LOOMOX auf dem Screen an.

Um das Konzept der Annäherungphase zu realisieren, bei welcher das Auge des LOOMOX zunächst groß auf dem Screen zu sehen ist und mit abnehmendem Abstand des Spielers zum Screen kleiner wird, bis schließlich der LOOMOX erscheint, wurde mittels der Tiefeninformation des Genickbones die Größe des Auges berechnet wobei die Änderung der Größe mit abnehmendem



Abstand immer schneller voranschreitet. Um die Blickrichtung des LOOMOX auf den Spieler zu setzen, wurde mittels der Tiefeninformation sowie der Position des Genicks des Spieler auf dem Screen berechnet, wohin der LOOMOX schauen muss.

4.2.1.2 Anwendungslogik

Um schnell Ergebnisse erzielen zu können, wurde zu Beginn des Projektes in der von Processing mitgelieferten Entwicklungsumgebung gearbeitet, da hier ohne großen Aufwand die ersten grafischen Prototypen bereit stehen und kleine Designfragen schon während der Diskussion beantwortet werden konnten. Processing formt Java so um, dass es als Skriptsprache, also ohne Klassen, funktioniert, was jedoch dazu führt, dass gerade komplexere Vorgänge oder einfach nur langer Code schwer zu verwalten ist. Aus diesem Grund wird man früher oder später gezwungen auf eine “richtige” IDE umzusteigen. Durch die Einbindung der Processing Bibliothek kann man auch weiterhin auf dessen Funktionen zugreifen.

Immer wiederkehrende technische Elemente der Anwendung sind Timer und Animationen, also Zustände welche sich nach oder über

einen Zeitraum hinweg verändern. Dafür wurden extra Helper Klassen implementiert, welche die Handhabung solcher Events vereinfachen sollten. Diese speichern alle nötigen Variablen und berechnen auch den absoluten, relativen und prozentualen Fortschritt.

Um unterschiedliche Zustände und Anwendungsfälle zu unterscheiden und gleichzeitig gemeinsames Verhalten zu koordinieren ohne dabei sehr lange und verschachtelte Abfragen zu verwenden, wurde das State Pattern eingeführt. Dieses wird eingesetzt, um unterschiedliche, zustandsabhängige Verhaltensweisen von Objekten zu kapseln.

Die einzelnen Zustände der gesamten Anwendung lassen sich in Passiv-, Approach-, Wave-, Follow-, Menu-, Game- und Goodbye-State unterteilen. Je Zustand können zwei, ein oder kein Benutzer mit der Anwendung interagieren. Dies führt zu einer komplexen Abfolge von Vor- und Folgebedingungen für den Wechsel der Zustände. Zudem müssen stets Abbruchbedingungen überprüft werden, d.h. beim Verlust des Trackings wird ein Timer gesetzt, der es dem Spieler ermöglicht erneut getrackt oder von einem anderen Spieler ersetzt zu werden. Realisiert wurde diese Logik mitttels einer abstrakten Klasse "State" welche das Standard-Verhalten definiert, ebenso wie die grundsätzlichen Zeichen- und Logik-Methoden. Des Weiteren wird bei jedem Wechsel des States eine Initialisierungsfunktion aufgerufen. Die konkreten Unterklassen überschreiben in manchen Fällen das Standard-Verhalten und implementieren die konkreten Methoden draw() und logic().

Um Redundanzen zu vermeiden wurde die Implementierung mithilfe von abstrakten Klassen und Interfaces auch in anderen Bereichen der Anwendung verwendet. Wiederzufinden ist derart gekapselte und vererbungsorientierte Softwarearchitektur in den Klassen Player, Game, KI und Ball. Diese Struktur ermöglichte

zudem die Bereitstellung und Verwendung wiederverwendbarer Methoden.

4.2.1.3 KI's

Die beiden Spiele Frisbee und Bubbles sind auch für den Einzelspieler konzipiert, daher wurde dafür ein Gegenspieler mit automatisiertem, situationsbedingtem und annähernd intelligentem Verhalten implementiert, welcher somit eine schwache Form von Künstlicher Intelligenz darstellt.

Umgesetzt wurde dies mit einem simplen Algorithmus der das Abschussverhalten für beide Spiele mit der selben Funktion ermöglicht. Zudem wurde der FrisbeeKI die Aufgabe zugeteilt dem Ball des Gegenspielers unter bestimmten Bedingungen zu folgen.

Die KIs wirken noch sehr starr und teilweise willkürlich. Der Schwierigkeitsgrad sollte noch abstufbar implementiert werden, hierfür gibt es verschiedene Möglichkeiten der Umsetzung, zum Beispiel Veränderung der Trägheit oder Schussgeschwindigkeit der KI.

4.2.1.4 DB Anbindung

Über JDBC (Java Database Connectivity) verbindet sich die Anwendung mit der MySQL Datenbank auf unserem Webserver. Somit kann mittels der ID des LOOMOX abgefragt werden, wie dieser von seinem Besitzer in der WebApp personalisiert wurde, sodass der LOOMOX genau so auf dem Screen angezeigt werden kann.
Zusätzlich wird jedes Spiel das der Benutzer an der Installation beginnt, gewinnt oder verliert in der Datenbank gespeichert, um über Spielstatistiken zu verfügen.

4.2.2 WebApp

Ein wichtiges Element, welches zur Einzigartigkeit des Projekts LOOMOX beiträgt, ist die zugehörige WebApp. Hierüber kann der Nutzer seinen persönlichen LOOMOX mitführen und ihn nach seinen Wünschen gestalten.

4.2.2.1 Aufbau & Funktionsweise

Über die WebApp können die LOOMOXe „besucht", also aufgerufen werden. Wenn der User am Screen einen Code erhalten hat, kann er diesen hier eingeben und sich den LOOMOX ansehen und ihn benutzen. Alternativ kann er ihn auch direkt „adoptieren", sodass der LOOMOX an ihn gebunden ist.
Sowohl freie als auch adoptierte LOOMOXe verfügen über den gleichen Funktionsumfang, sodass der Nutzer nicht gezwungen wird, sich zu registrieren, da ein freier LOOMOX ohne Login zugänglich ist. Die Tatsache, dass der LOOMOX dann jedoch keinen schönen Namen erhält, sondern nur über den ursprünglichen Code aufgerufen wird und darüber hinaus auch von anderen Spielern benutzt oder sogar adoptiert – und damit für einen selbst unzugänglich gemacht – werden kann, soll den User jedoch dazu bewegen, im Rahmen einer Registrierung den LOOMOX zu adoptieren, was die persönliche Bindung stärkt.
Wenn ein Nutzer nun seinen LOOMOX aufruft, wird er von diesem begrüßt und erhält einen Überblick über die zur Verfügung stehenden Items, sowie einen Button, der es bei korrektem Aufenthaltsort ermöglicht, den LOOMOX zurück an den Screen zu schicken.
Alle bei Spielen mit dem LOOMOX gewonnenen Items können einfach auf den LOOMOX gezogen werden und werden sofort angewendet. Es ist möglich die Augenfarbe sowie die Form des LOOMOX zu ändern. Derzeit stehen somit 40 verschiedene Kombinationsmöglichkeiten zur Verfügung.

Der Button zum Zurückschicken des LOOMOX auf den Screen ist zunächst deaktiviert und informiert den User darüber, dass er sich nicht am Screen befindet. Wenn erfolgreich sichergestellt wurde, dass sich der User an der Bibliothek befindet, wird der Status der Installation überprüft. Falls diese nicht besetzt ist wird der Freilassen-Button aktiviert.
Wenn der User diesen nun drückt, bewegt sich sein LOOMOX vom Handybildschirm und eine Meldung informiert den Nutzer, dass er sich nun vor den Screen begeben und beide Hände zur Erkennung heben soll.
Neben der Möglichkeit mit dem eigenen LOOMOX zu interagieren, bietet die Webapp noch die Möglichkeit, sich einen Überblick über verfügbare Spiele zu verschaffen, sich über das Projekt zu informieren und außerdem über den Menüpunkt „Einstellungen" sein Passwort und seine E-Mail-Adresse zu ändern, oder auch für das aktuelle Gerät die Bewegung/Animation des LOOMOX auszuschalten, da diese auf älteren Geräten partiell zu Performanceproblemen führt.

4.2.2.2 Struktur

Die Struktur der Webapp teilt sich in drei große Bereiche:

- **MySQL Datenbank**
- **PHP Backend**
- **Javascript/HTML5 Frontend**

Die Website bzw. WebApp wurde in HTML5, CSS3 und mithilfe des Bootstrap Framework umgesetzt. Datenverarbeitung und Datenbankanbindung wurden in PHP realisiert, Funktionalitäten der Homepage mit Javascript unter Verwendung von jQuery und Standortüberwachung und LOOMOX-Animation mit objektorientiertem Javascript unter Verwendung der HTML5 Canvas Javascriptbibliothek KineticJS.

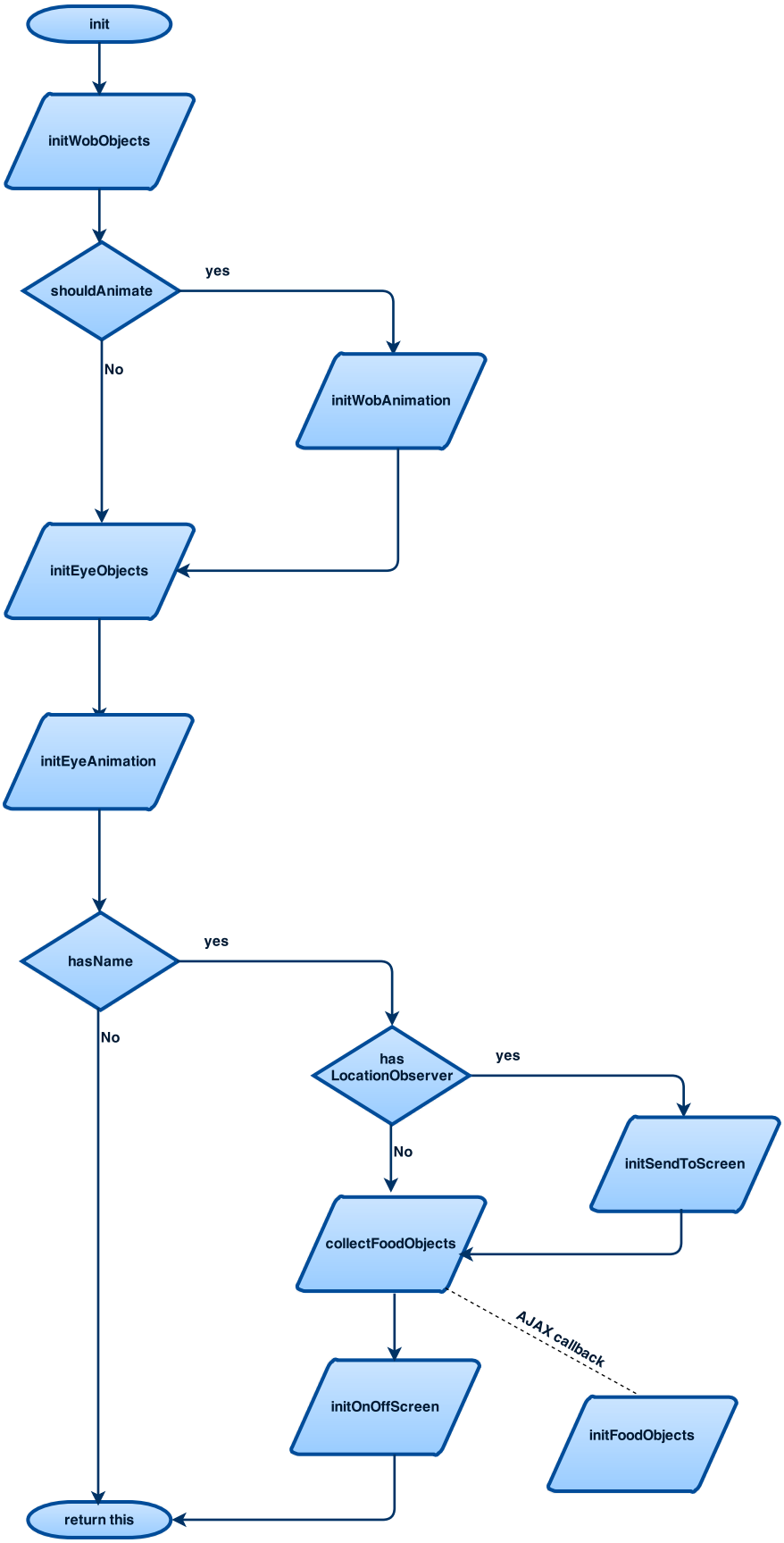
4.2.2.3 Der „Webwobbl“

Der animierte „Webwobbl“ – also der LOOMOX der Webapp – stellte in mehreren Punkten eine Herausforderung dar, in erster Linie bezüglich Performance, aber auch bei exakter Nachbildung vorgegebener Designs und Erhaltung der Objektorientiertheit.

4.2.2.3.1 Programmierung

Es wurden drei instanziierbare Javascript „Klassen“ entwickelt:
Wobbl
Food
LocationObserver
Dies wurde gegenüber einem für Javascript klassischen, imperativen Ansatz gewählt, um eine hohe Erweiterbarkeit sowie Wiederverwendbarkeit der einzelnen Bestandteile zu gewährleisten. Daraus resultiert, dass mit theoretisch nur einer Zeile Javascript Code an jeder beliebigen Stelle der WebApp ein oder auch mehrere LOOMOXe erstellt werden können, angepasst über eine Reihe überwiegend optional anzugebender Parameter. Dies wird zum Beispiel auf den Fehlerseiten genutzt, um diese ansprechender zu gestalten (z.B. http://loomox.de/project/asdfadfafdadf -> 404).
Die beiden erstgenannten Klassen sind abhängig von der Library kineticJS.

Der LOOMOX selbst verfolgt beim Erstellen folgende **Ablauflogik**:


init
initWobObjects
shouldAnimate
yes
No
initWobAnimation
initEyeObjects
initEyeAnimation
hasName
yes
No
has
LocationObserver
yes
No
initSendToScreen
collectFoodObjects
AJAX callback
initOnOffScreen
initFoodObjects
return this

4.2.2.3.2 Animation

Die Animation des LOOMOX wird mit der Javascript-Library KineticJS realisiert und innerhalb der Klasse „Wobbl" gekapselt. Die einzelnen Boids des LOOMOX bewegen sich alle auf Kreisbahnen, deren Mittelpunkt wiederum auf einer Kreisbahn liegt, sodass sie die Kante des zentralen Körpers überdecken und ein organischer Eindruck entsteht. Um die Animation nicht zu gleichförmig erscheinen zu lassen, werden Geschwindigkeit, Drehrichtung und Größe beim Erstellen per Zufall ermittelt und abgespeichert.

Um die Animation möglichst performant zu halten, wurde die Anzahl der Boids auf das Niedrigmöglichste reduziert. Je nach Form variiert diese Zahl, bei "Bobbln" werden zum Beispiel weniger benötigt als bei Dreiecken.

Neben der Animation der Boids kuckt das Auge des LOOMOX umher. Dies geschieht über eine rekursive Funktion die in zufälligen Zeitabständen die Position der Pupille innerhalb des Auges ändert. Die Zwinker-Animation wird äquivalent erstellt.

Neben den automatisch ablaufenden Animationen kann auch der User Einfluss nehmen. Wenn der LOOMOX im Auge berührt wird, zwinkert er und wenn sein Körper berührt wird zuckt er zusammen. Tut man dies zu oft ärgert dies den LOOMOX und er kuckt den User für kurze Zeit grimmig an.

4.2.2.3.3 Items

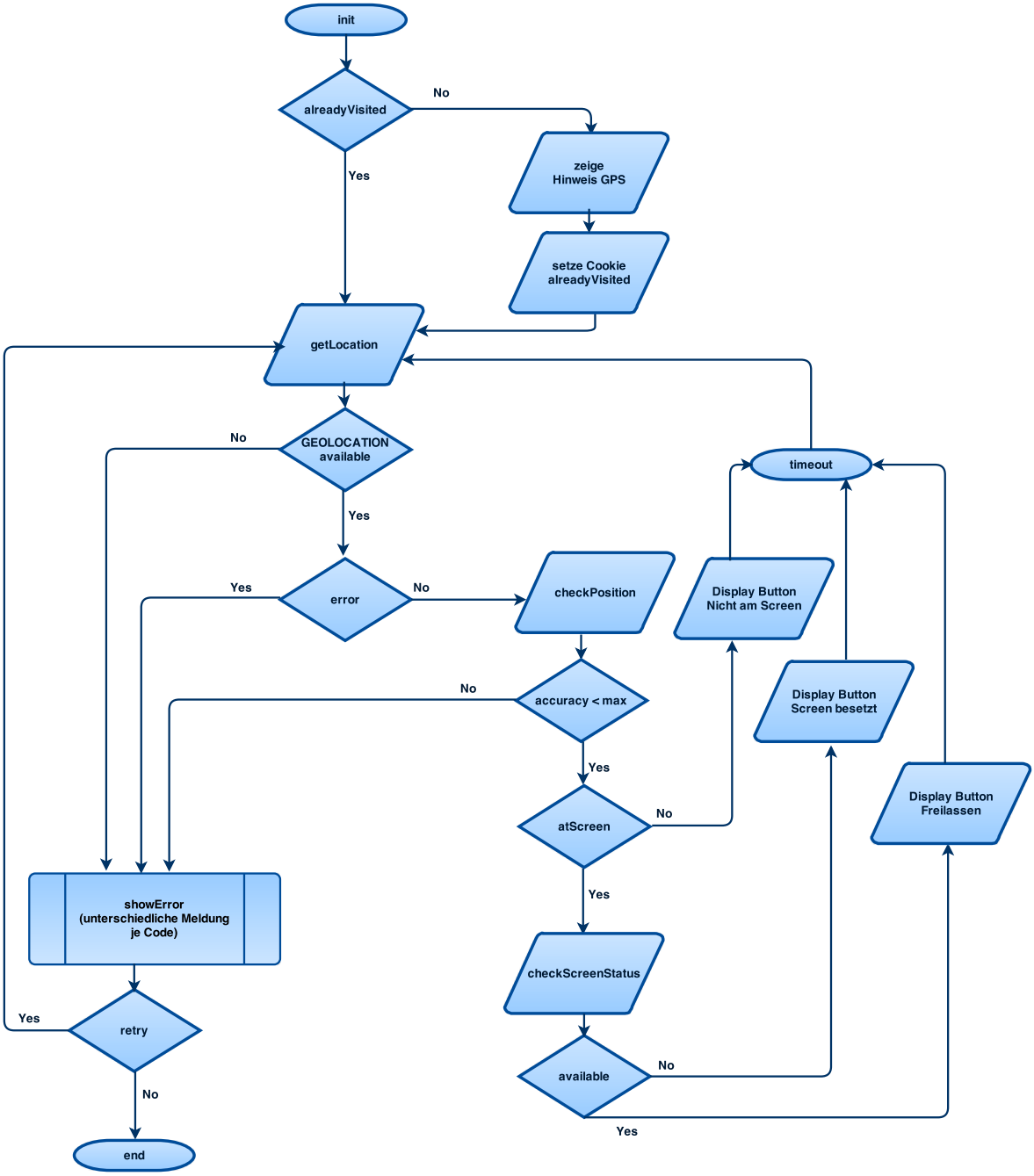
Die dem LOOMOX zur Verfügung stehenden Items werden per AJAX (bzw. eigentlich AJAJ, da als Datenformat JSON verwendet wird) bei Bedarf vom Server nachgeladen. Sie befinden sich dann in einer Reihe unter dem LOOMOX. Diese Reihe kann per Maus Drag and Drop oder per Touch hin und her bewegt werden – einzelne Items können darüber hinaus in Richtung LOOMOX gezogen werden. Wenn ein Item angewendet wird, wird dies im

Hintergrund wiederum direkt per AJAX (eigentlich AJAP, da es sich um einen simplen POST-Request ohne Rückgabewert handelt) an den Server übertragen – es ist also kein explizites „Speichern" oder Neuladen der Seite notwendig.

4.2.2.3.4 Standortverarbeitung

Die Standortverarbeitung läuft komplett clientseitig und bedient sich der HTML5 Geolocation API. Die Genauigkeit und Funktionsweise dieser API wurde zuvor mit einer Testhomepage ermittelt und für ausreichend gut befunden. Diese ist unter http://loomox.de/geolocation-test.html einsehbar.
Die Javascript-Klasse LocationObserver wird mit Koordinaten der Hochschulbibliothek, einer gewünschten Maximal-Genauigkeit und dem gewünschten Maximalumkreis sowie den IDs der einzelnen Buttons die ein und ausgeblendet werden sollen und den IDs der Fehlermeldungs-Fenster instanziiert.

Der verfolgte **Ablauf** sieht folgendermaßen aus:



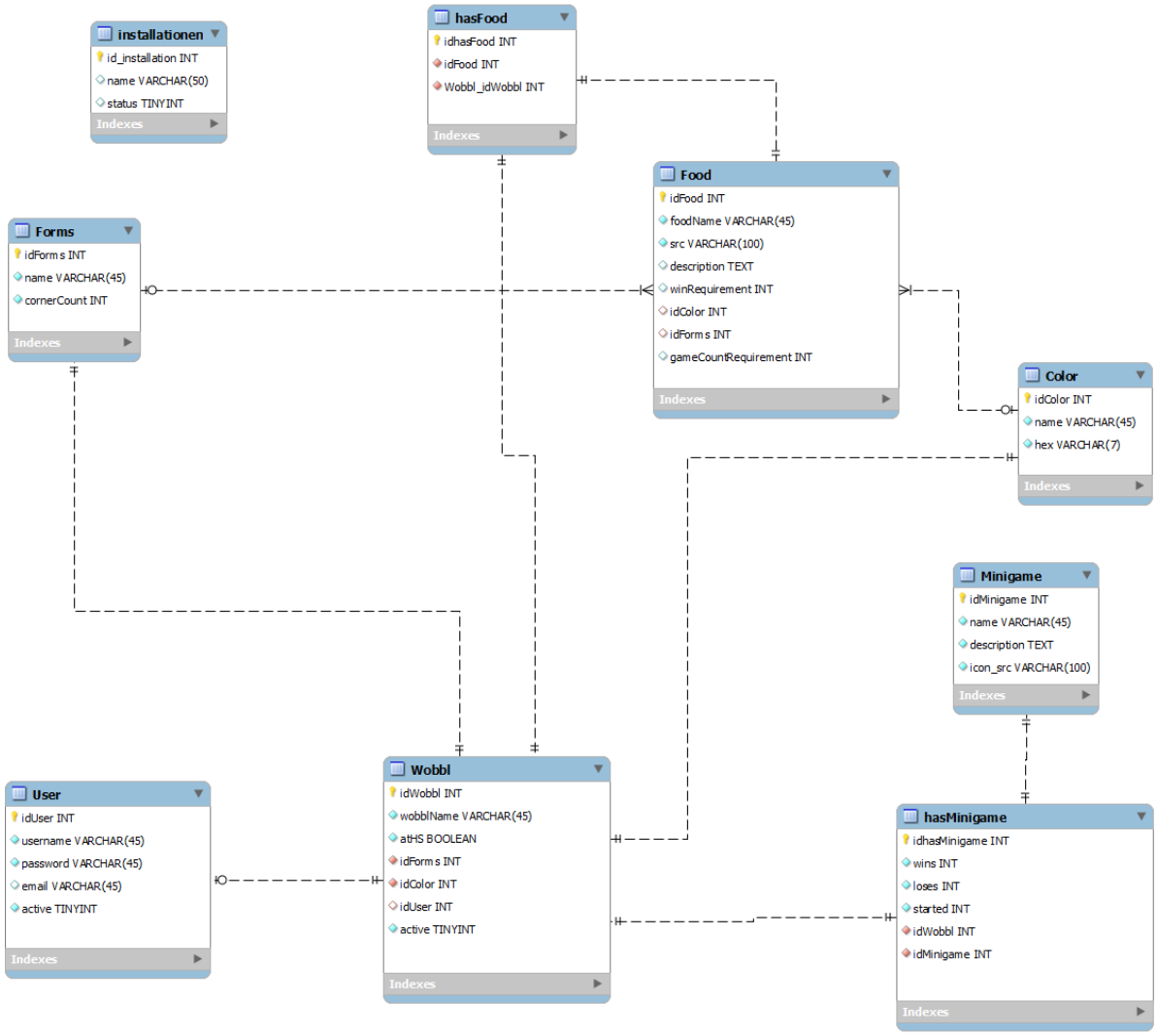
4.2.2.3.5 API

Um den LOOMOX bei Bedarf verändern zu können, wurden diverse Funktionen zur Verfügung gestellt, die die entsprechend nötigen Befehle kapseln, unter anderem solche zum Auslösen eines Zwinkern oder Zuckens, zum Ändern von Farbe, Form, Geschwindigkeit oder dem Hinzufügen weiterer Items.
Zur Verarbeitung der diversen AJAX-Anfragen, die clientseitig gesendet werden, wurden als erweiterte Datenbankanbindung

bzw. API eine Reihe von PHP-Skripten entwickelt, die diese Anfragen bearbeiten, unter anderem zum Abrufen der zur Verfügung stehenden Items, des Status des Screens, dem Anwenden eines Items oder dem Freilassen des LOOMOX. Die Kommunikation erfolgt hier je nach Eignung via GET, POST oder JSON.

4.2.2.4 Datenbank

Das Datenbankmodell wurde im Hinblick auf optimale Erweiterbarkeit erstellt. Noch vorgesehen sind zum Beispiel Statistiken und Requirements für einige Items, die bereits hinterlegt werden können, wenn auch noch nicht verarbeitet werden.



Die Webapp kommuniziert mit dieser über PDO (PHP Data Objects) und ist somit bereits gegen SQL-Injections abgesichert.

4.2.2.5 PHP Programmierung

Für die PHP Programmierung wurde kein Framework verwendet, da uns eine schnelle Reaktionszeit des Servers sehr wichtig war und Overhead somit vermieden werden sollte. Aufgrund des schlanken Codes ist es nun möglich, dass die Anfrage nach dem Status des Screens in unter 130ms bearbeitet wird, die komplette Liste an verfügbaren Items in unter 140ms und die Seiten der Webapp in äquivalenten Zeiträumen. (Hierbei werden nachzuladende Libraries, Bilder etc. nicht mit eingerechnet.)
Zum Vergleich: Homepages, die gängige Frameworks benutzen und auf dem gleichen Server liegen benötigen deutlich länger zum Ausliefern ihrer Startseiten:
Joomla 3: ca. 450ms
TYPO3 6.1: ca. 300ms
Selbst programmiert wurden neben der Ausgabe des LOOMOX ein Usersystem mit Registrierungsfunktion, bei der ein LOOMOX adoptiert und benannt werden kann und ein dazugehöriger Login.

4.2.2.6 Sicherheit

In die Absicherung der Anwendung und der Nutzerdaten wurde viel Arbeit gesteckt.

Im Registrierungsformular ist ein verstecktes INPUT-Feld als Spamschutz vorhanden. Da es für User nicht sichtbar ist, wird es von diesen leer gelassen, wohingegen Spambots in der Regel einfach alle Felder ausfüllen. So können diese abgewiesen werden, da eine Anfrage, bei der das versteckte Feld befüllt ist, nicht bearbeitet wird.

Nach erfolgreicher Registrierung bekommt der Nutzer einen Aktivierungslink per Email zugesendet. Dieser beinhaltet als Parameter den Usernamen und einen MD5 Hashwert des Usernamen kombiniert mit einem statischen Secret. Nur wenn die Werte zueinanderpassen wird der User beim Klick auf den Link aktiv geschalten und kann sich anschließend einloggen.

Passwörter werden stark verschlüsselt in der Datenbank abgelegt. Anstelle einer oft vorgeschlagenen, aber als unsicher geltenden md5 Verschlüsselung (bzw. Hashing) wird der sha256 Hash-Algorithmus verwendet, der 256-Bit Schlüssel erstellt, welche durch die größere Länge deutlich sicherer sind, da die Kollisionswahrscheinlichkeit mit anderen Passwörtern deutlich geringer ist.

Darüber hinaus wird die Methode des Passwort-Salting angewendet. Das „Salzen" eines Passworts sorgt dafür, dass eventuell aus einer Datenbank gestohlene Daten weniger einfach mit Rainbow-Tables geknackt werden können und verhindert, dass Nutzer mit demselben Passwort denselben Hash-String erhalten und dieser Umstand – also, dass beide dasselbe Passwort verwenden - somit ersichtlich wäre. Der exakte Algorithmus, mit dem die LOOMOX-Passwörter gesalzen werden, wird aus Sicherheitsgründen hier nicht verraten, lediglich, dass neben einem nutzerspezifischen Salt zusätzlich ein statischer Salt und ein zusätzlich gehashter Salt kombiniert werden, um ein Maximum an Verschleierung zu erreichen.

Die Domain loomox.de verfügt leider nicht über ein SSL-Zertifikat, da ein solches zusätzliche Kosten in Höhe von mindestens 100€ pro Monat verursachen würde. Jegliche Bemühungen um serverseitige Passwortsicherheit wären folglich sinnlos, wenn die Passwörter bei Login und Registrierung unverschlüsselt übertragen werden. Zum Glück ist im verwendeten Hostingpaket aber ein so genanntes „Shared SSL Zertifikat" enthalten – auf den Server kann somit anstelle von http://loomox.de auch SSL-

verschlüsselt über https://secure63.tophost.ch/~sebasti1/webspace/wobbl/ zugegriffen werden. Diesen Umstand machten wir uns zunutze, indem wir jede Formularübertragung die sensible Logindaten beinhaltet, über diese SSL-verschlüsselte Verbindung an den Server leiten. Die entsprechenden Skripte leiten nach erfolgreicher Datenverarbeitung direkt wieder auf loomox.de weiter, sodass der User die Domain nicht zu Gesicht bekommt.

Diese Technik führte jedoch zu dem Problem, dass der Login zuerst nicht mehr funktionierte, da die verwendeten Sessioncookies nun für die Domain secure63.tophost.ch gültig waren und auf loomox.de nicht zur Verfügung standen. Aus diesem Grund wurde eine weitere Weiterleitung eingebaut. Zuerst verarbeitet ein über die sichere URL aufgerufenes Skript die Logindaten, startet eine Session und setzt die Sessionvariablen wie gewünscht. Anschließend extrahiert das Skript die SessionID und leitet den User an ein über die Domain loomox.de aufgerufenes Skript weiter und übergibt dabei die SessionID als GET-Parameter. Dieses Skript eröffnet nun erneut eine Session, verwendet hierfür aber die übergebene SessionID. Das Sessioncookie dieser Session ist nun für die Domain loomox.de gültig und da die Zuordnung von ID zu Variablen serverseitig erfolgt, stehen die im ersten Skript unter anderer Adresse gesetzten Variablen jetzt auch über die Session, die loomox.de zugeordnet ist, zur Verfügung und der Nutzer ist erfolgreich eingeloggt. Zuletzt wird der Nutzer sofort wieder auf eine Seite von loomox.de weitergeleitet, sodass er auch die als Parameter übergebene SessionID nicht zu Gesicht bekommt.

LOOMOX | Login

POST login.php	302 Moved Temporarily	secure63.tophost.ch	0 B	194.150.249.140:443
GET sessionredirect.php?id=a...61b56e8ceb3cebb588	302 Moved Temporarily	loomox.de	4 B	194.150.248.140:80
GET /mein-loomox/	302 Moved Temporarily	loomox.de	0 B	194.150.248.140:80
GET testmeiner6	200 OK	loomox.de	9,3 KB	194.150.248.140:80
GET get-foods.php?name=testmeiner6	200 OK	loomox.de	12 B	194.150.248.140:80
5 Anfragen			9,4 KB	

Die Übertragung der Passwörter ist somit gegen ein Auslesen von Dritten abgesichert. Der Nachteil dieser Methode gegenüber der

SSL-Verschlüsselung der kompletten Seite besteht darin, dass die Sessioncookies leider über eine ungesicherte Verbindung verschickt und somit ausgelesen und von Dritten verwendet werden können (sogenanntes Session Hijacking). Da mit einem LOOMOX-Account jedoch keine sensiblen Aktionen durchgeführt werden können und „Account Lockout" – also die Änderung des Passworts durch einen Fremdnutzer – durch erneutes Abfragen des Passworts im Passwort-Ändern Dialog verhindert wird, ist dieses Sicherheitsrisiko vertretbar, da kein Schaden angerichtet werden kann.

5. PRÄSENTATION

Um die entstandene Anwendung möglichst realitätsnah und attraktiv zu präsentieren, wurden Konzepte zum Vortrag, dem Stand des Teams, sowie einer durchgehenden Präsenz des LOOMOX im Theater entwickelt. Zusätzlich wurde im Vorfeld über Facebook Aufmerksamkeit für unsere Installation generiert.

5.1 Vortrag

Als Vortragseinstieg entwickelten wir eine kleine Showeinlage, bei der ein renommierter Wissenschaftler die Entdeckung einer neuen Spezies (unseres LOOMOX) vorstellt. Durch einen Zwischenfall, bei dem der Wissenschaftler seine Unterlagen durcheinander bringt, wird dann die eigentliche Präsentation eingeleitet, in der wir auf unsere Ziele und die verwendete Technik eingingen und die Zuschauer anschließend, unterstützt durch ein eigens für diesen Zweck produziertes Trailervideo, Schritt für Schritt durch unsere Anwendung führten.

5.2 Stand

Bei der Umsetzung des Standes wurde besonderer Wert darauf gelegt, das Design unseres Konzeptes aufzugreifen und somit eine stimmige Atmosphäre zu schaffen. Es wurden hauptsächlich schwarze und weiße Elemente verwendet, um im Einklang mit der Installation zu stehen. Beim Dresscode des Teams - es wurde vereinbart, weiße Hemden zu tragen - wurde auf die Wirkung in einem abgedunkelten Raum geachtet.
Für die Präsentation unserer Installation stand ein Foyer zur Verfügung, welches sich am Aufgang zur Ausstellung befand. Unsere Präsenz teilte sich in zwei Bereiche, einerseits die eigentliche Installation, an welcher Besucher unsere Anwendung testen konnten, und andererseits einen Stand, der die Möglichkeit zur Diskussion mit Interessenten bot.
Der Aufbau für unsere Installation bestand im Wesentlichen aus einem Beamer, einer Rückprojektionsleinwand, sowie einer Verkleidung des erstgenannten. Bei der Konstruktion der

Verkleidung wurde Wert auf Realitätsnähe zur Hochschulfassade gelegt. So wurde ein Gerüst aus Latten gebaut, welches die Fenster der Hochschule nachbildete und bis auf die eigentliche Projektionsfläche von innen mit schwarzer Pappe verkleidet war. Höhe und Breite der Projektionsfläche hatten das Originalverhältnis des Fensters an der Bibliothek von 1 zu 1,7.

Um mit Interessierten in Kontakt zu treten, bauten wir einen matt schwarzen Stand, der so platziert wurde, dass er für jeden Besucher beim Betreten der Veranstaltung sichtbar war. Um hierbei direkt aufzufallen wurde an der sichtbaren Seite des Standes eine mit LEDs beleuchtete LOOMOX-Lampe angebracht. Des Weiteren wurde der Stand durch das Anbringen unseres Logos und Aufstellen eines Bildschirms, welcher Videomaterial unser Anwendung zeigte, aufgewertet.

Um im gesamten Theater nachhaltiges Interesse für LOOMOX zu generieren, wurden neun unterschiedliche LOOMOX Poster an verschiedenen Stellen aufgehängt. Ebenso wurden Flyer mit

LOOMOX-Codes verteilt, um den Besuchern die Möglichkeit zur Registrierung auf unserer WebApp zu geben. So können auch leicht Interessenten benachrichtigt werden, wenn die Installation fertiggestellt wurde.

6. FAZIT

Trotz zahlreicher Hindernisse ist im vergangenen Semester eine Projektarbeit entstanden, die uns, aber auch allen, die die Gelegenheit hatten, diese zu testen, viel Freude bereitete. Das Team LOOMOX bedankt sich bei den Betreuern, Herrn Prof. Dr. Kipp und Herrn Prof. Rothaug, bei allen Testern der Anwendung im Rahmen unseres Usability-Tests, sowie allen weiteren Helfern, die das Projekt vorangetrieben haben.