

# Mediengestaltung 3 - Projekt

# LPCS

*Laserpointer-Pumping-Control-System*

**Hochschule für Angewandte Wissenschaften Hamburg**

**Fakultät Design, Medien und Information**

**Department Medientechnik**

**Studiengang Media-Systems**

**Wintersemester 2013/14**

**Prof. Gunther Rehfeld, Prof. Ralf Hebecker**

**Johannes Bagge**

**Sebastian Bohn**

**Martin Trapp**

## Inhaltsverzeichnis

1 Ziel des Projekts	4
2 Spielprinzip	4
3 Vorlagen / Inspirationen	4
3.1 Rätsel- und Suchspiele	4
3.2 Geschicklichkeit	6
4 Entwicklung des Konzepts	8
5 Zielgruppenanalyse	9
6 Schwerpunktverteilung der Aufgaben	10
7 Blender	10
8 Photoshop	11
9 Typographie	12
10 Sounds	12
11 Controller	14
11.1 Vorüberlegungen	14
11.2 Umsetzung	16
11.3 Tracking	18
11.4 Komponenten	21
11.5 Grenzen und Ausbaumöglichkeiten	22
12 Das Game	23
12.1 Menu	23
12.2 Spielansicht	23
12.3 Spielende	24
13 Kosten	25
14 Zeitplan	25

MG3-Projekt „LPCS“

15 Quellenverzeichnis 26

16 Abbildungsverzeichnis 26

# 1 Ziel des Projekts

In unserem MG 3 Projekt fanden wir uns zusammen, um gemeinsam ein Game zu entwickeln. Das Ziel war es ein Feuerwehr-Spiel zu programmieren, das mit Hilfe eines eigens entwickelten Controllers funktioniert.

Der Controller besteht aus einer originalen Kübelspritze und einem in der Schlauchspitze integrierten Laserpointer.

## 2 Spielprinzip

Gespielt wird aus der Ego-Perspektive. Vor sich ist ein Haus zu sehen, in dessen Fenster nach und nach Feuer auftauchen. Mit dem Laserpointer wird ein Wasserstrahl gesteuert, um das Feuer zu löschen. Durch Betätigung der Pumpe wird der Wasserstrahl verstärkt.

Das Spiel ist beendet wenn alle aktiven Feuer gelöscht sind. Daraufhin wird die benötigte Zeit eingeblendet.

## 3 Vorlagen / Inspirationen

Bevor wir zu dem eigentlichen Spiel kamen, waren einige Überlegungen anzustellen, welche Spielidee umgesetzt werden soll. Fest stand, das es ein Spiel mit einer alternativen Steuerung wird. Wir einigten uns Aufgrund der Vorerfahrung von Martin, im Bereich Tracking, auf eine Controller-Variante mit einem Laserpointer. Dazu wurden verschiedene Vorschläge gesammelt, die mit dieser Peripherie möglich wären.

### 3.1 Rätsel- und Suchspiele

Möglich wäre ein Suchspiel zu konstruieren, wo man Fehler aufdecken oder aber auch einfach nur

gesuchte Objekte entdecken muss.



Abbildung 1: Wimmelbild-Spiel (PC-Wimmelbox)

Ein Spiel in dem man mit dem Laserpointer bestimmte Formen zeichnen muss, um Aktionen auszulösen wäre denkbar. Ähnlich wie bei „Okami“, aber weniger komplex.



Abbildung 2: Okami

## 3.2 Geschicklichkeit

Spiele die eine kurzweiliges Prinzip verfolgen und Highscore basiert sind.

„Fruit Ninja“ ist ein gutes Beispiel und wäre mit geplanter Steuerung gut umzusetzen.

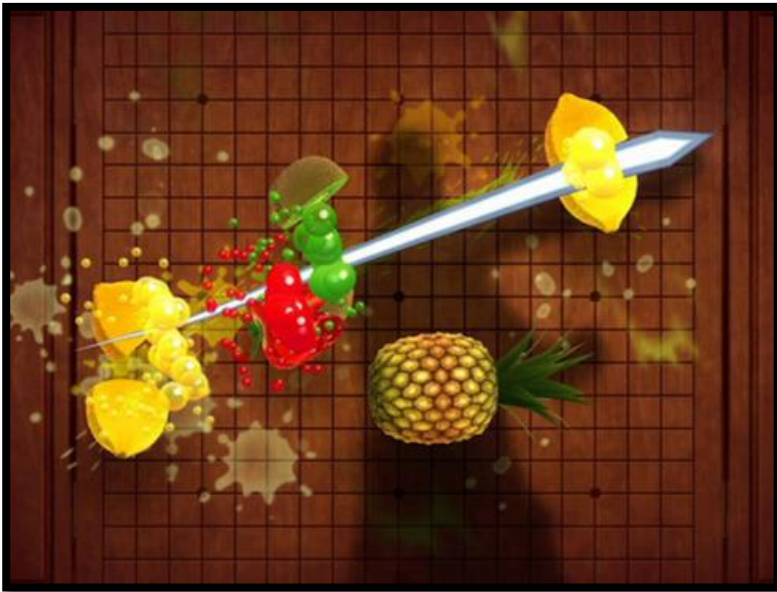


Abbildung 3: Fruit Ninja

Ein einfacher Shooter wie „Duck Hunt“, war auch im Gespräch.

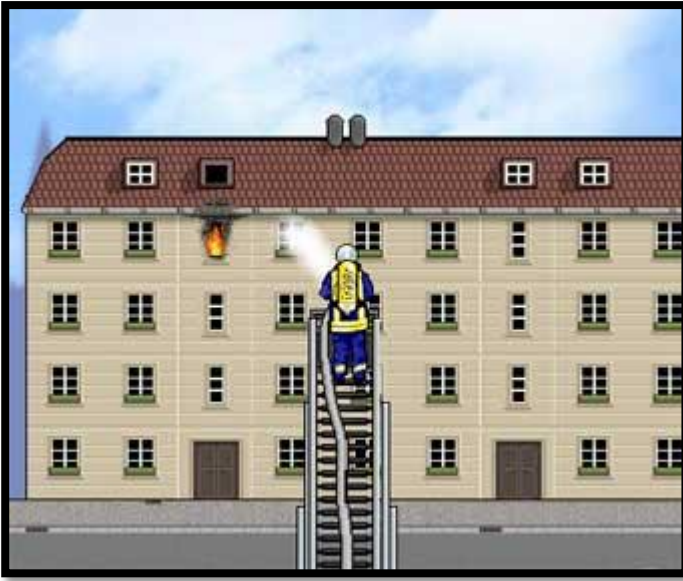


Abbildung 4: Duck Hunt

Obwohl es viele gute Vorlagen und Konzepte gibt, wollten wir aber etwas einzigartiges entwickeln. Wir wollten eine Kombination die noch nicht allzu verbraucht ist. Es kam uns die Idee ein Feuerwehspiel zu entwerfen. Das jetzige Resultat unserer Arbeit, war zu diesem Zeitpunkt noch wilde Phantasie. Das man eine Pumpe baut, die den Wasserdruck reguliert, war eine Wunschfiktion. Nach einem Brainstorming, das Abwägung von Können, Möglichkeiten und Motivation der Gruppe analysierte, wurde beschlossen Fiktion zur Realität zu machen.

Auch wenn es schon eine ungefähre Vorstellung gab, wie Aussehen und Ablauf sein sollen, wurde recherchiert, welche Feuerwehspiele auf dem Markt sind.

Das Spiel was unserer Idee relativ nahe kommt, ist ein Browser-Game mit Namen „Fire-Fighter by Dräger safety“. Hierbei handelt es sich um ein Werbespiel genannter Firma. Man steuert einen Feuerwehrmann auf einer Leiter und löscht sich ausbreitende Brände auf einem Haus.



*Abbildung 5: Fire Fighter by Dräger safety*

## 4 Entwicklung des Konzepts

Was sich bei unserem Spiel unterscheiden sollte, ist die starre Ego-Perspektive. Zu steuern sollte nur der Wasserstrahl sein und kein sichtbarer Avatar.

Die Interaktion und Steuerung sollte an unsere Idee angepasst, sowie das Balancing verbessert werden. Hierzu erstellten wir einen ersten Konzeptentwurf.



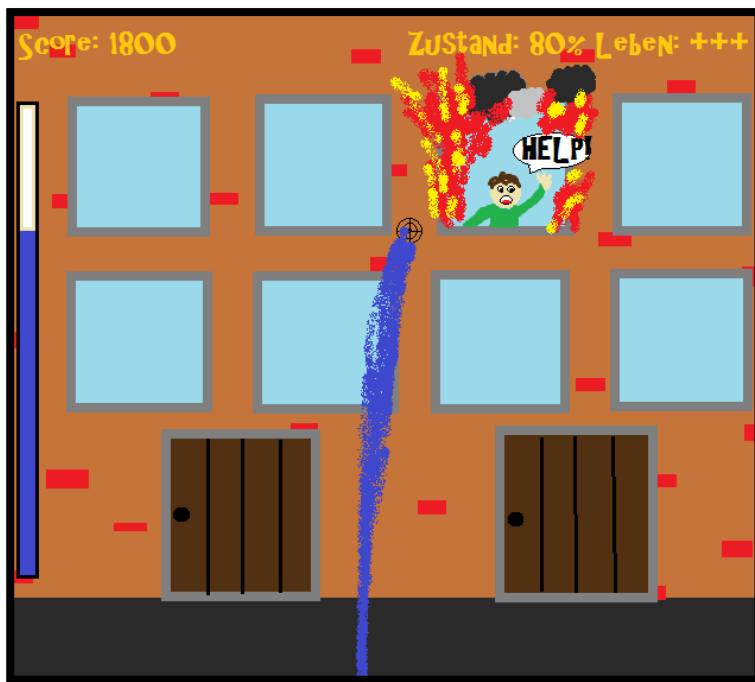


Abbildung 6: Konzeptentwurf

Es sollte einen kurzweiligen Spielverlauf geben. Hierzu wurden zwei verschiedene Abläufe erarbeitet.

Das erste Prinzip basiert darauf einen Highscore zu erspielen. Begrenzt wäre die Spielzeit durch eine Gesundheitsanzeige des Gebäudes, die durch die Brände sinkt und das Spiel bei 0% beendet.

Das zweite Prinzip basiert auf einen Timer, der die benötigte Zeit misst, die man braucht um alle Feuer zu löschen.

Wir entschieden uns für die zweite Variante. Die Vorteile die wir sahen, sind die Einfachheit und bessere Verständlichkeit des Spielziels.

## 5 Zielgruppenanalyse

Gedacht ist das Spiel für Leute die Spaß an Wettkampf und Bestzeiten haben, die sich für neue, experimentelle Konzepte interessieren und sich nicht vor der Anstrengung der

Controllerbedienung scheuen.

## 6 Schwerpunktverteilung der Aufgaben

Für die Verwirklichung gab es viele Punkte zu erarbeiten.

Martin kümmerte sich um den Bau der Hardware und die Programmierung dieser. Dazu gehörte der Controller mit Pumpe und Laserpointer, sowie das Tracking mit der Kamera.

Johannes war für die Gamelogik und Skripte in Unity zuständig.

Sebastian war für das Design, den Bau der Elemente in Blender und die Grafiken in Photoshop verantwortlich.

## 7 Blender

Für die Konstruktion der Modelle wurde das Open-Source 3D Animationsprogramm Blender v2.67 genutzt. Durch die große und ständig wachsende Community um diese Software, kann diese sich problemlos mit der kostenpflichtigen Konkurrenz messen. Für die Arbeit mit Blender ist eine Einarbeitungszeit und Recherche notwendig, da der Umgang gewöhnungsbedürftig ist und die Funktionen sehr umfangreich sind. Vorteilhaft ist, dass Blenderdateien fast problemlos in Unity benutzt werden können. Was jedoch beachtet werden muss, ist das die Skalierung in beiden Programmen unterschiedlich ist.

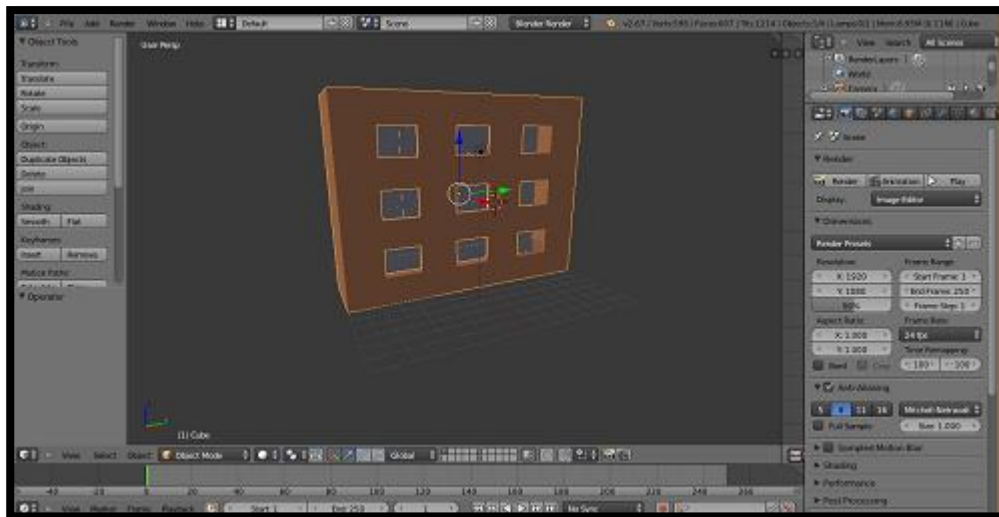


Abbildung 7: Blender v2.67

## 8 Photoshop

Das Design der Grafiken, wie Texturen, Feuer und Wasser wurden mit einer Testversion von Adobe Photoshop erstellt. Diese ist eine der umfangreichsten und mächtigsten Bildbearbeitungsprogramme auf dem Markt und verhilft Dank der vielen Tutorials im Internet zu guten Ergebnissen. Eine sehr gute Open-Source Alternative findet man mit dem Programm Gimp.



Abbildung 8: Photoshop

## 9 Typographie

Die Textelemente in dem Spiel sind recht gering gehalten und auch weniger der Schwerpunkt im Design. Google bietet auf einer seiner Webseiten eine große, nutzbare Auswahl an Fonts. Um den Gamecharakter zu unterstützen, wählten wir einen Font im 8-Bit Stil (8-Bit Wonder.ttf).

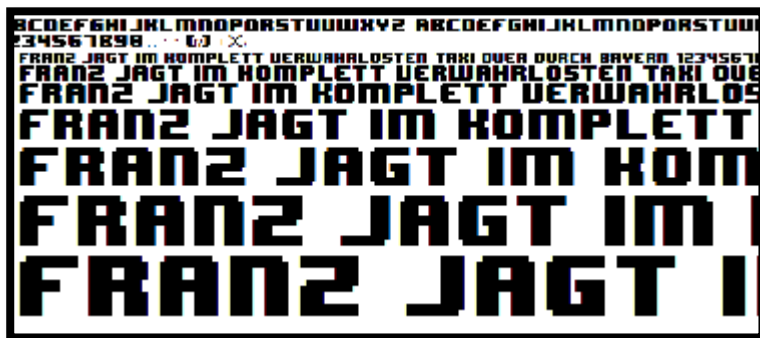


Abbildung 9: Font Beispiel

## 10 Sounds

In der aktuellen Version des Spiels sind nur zwei Audiodateien implementiert. Zum einen das leise Knistern des Feuers und zum anderen das Rauschen des Wasserstrahls. Die Lautstärke des Wasserstrahls wurde in Abhängigkeit der Strahlstärke programmiert.

MG3-Projekt „LPCS“

Die Sounds wurden von der kostenfreien Webseite [freesound.org](https://freesound.org) gewählt und mit dem Freeware Audiobearbeitungsprogramm „Audacity“ angepasst.

# 11 Controller

## 11.1 Vorüberlegungen

Bereits früh in der Entscheidungsfindung freundeten wir uns mit dem Gedanken an einen eigenen Controller herzustellen. Dieser Controller sollte alle Anforderungen des Spiels abdecken, aber nicht mehr. Wenn möglich, sollte dabei auf klassische Eingabekonzepte wie Buttons oder Sticks verzichtet werden. Das Resultat war ein Feuerwehspiel in Kombination mit einer Kübelspritze.



Abbildung 10: Kübelspritze

Um mit dem Spiel interagieren zu können, sollten zwei Aspekte erfüllt werden.

1. Analog zur realen Funktion einer Kübelspritze, muss auch im Spiel Wasserdruck erzeugt werden. Dies soll mithilfe der Pumpe geschehen, wobei der Mechanismus weder zu hart noch zu leichtgängig sein sollte. Perfekt wäre es, wenn es so schwergängig wie möglich und so leichtgängig wie nötig ausfällt, um den Spieler nach einer gewissen Zeit ins Schwitzen zu bringen und es gleichzeitig noch möglich ist, den Pumpenmechanismus mit einer Hand zu bedienen.
2. Der Schlauch der Kübelspritze soll als Maus fungieren, man soll also mit dem Schlauch zielen können.

Ob sich eine Kübelspritze aber tatsächlich als Controller eignet, konnte erst nach dem Kauf festgestellt werden. Schließlich war Niemandem von uns klar, inwieweit man das Innere zu unseren Gunsten zerlegen kann.

So war die erste Überlegung, den Wasserdurchfluss beim Pumpen durch einen Durchflusssensor zu messen, also tatsächlich bis zu einem gewissen Maß Wasser in die Kübelspritze zu füllen.

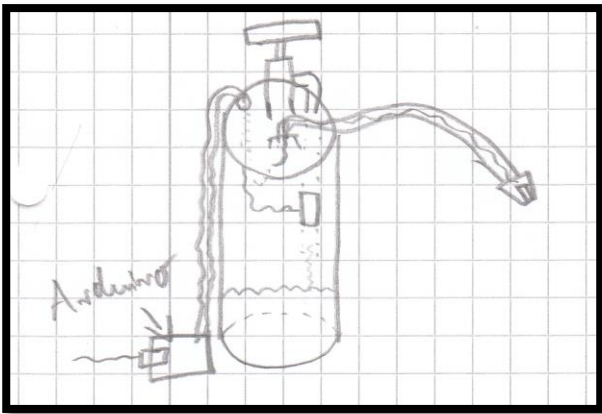


Abbildung 11: Erste Skizze des Mechanismus

Dieses System wurde dann aber zugunsten der heute verbauten Logik aufgegeben, da sich die Kübelspritze als wartungsfreudiger erwies als im Vorfeld angenommen wurde.

## 11.2 Umsetzung

Nachdem es nicht mehr nötig erschien einen Durchflussmesser zwecks Messung der Wassermenge zu verbauen, kamen andere Systeme in Betracht.

Feder + Drucksensor : Eine Passende Feder müsste einerseits sehr dehnbar, andererseits aber auch so stark komprimierbar sein, um einen möglichst großen Pumpenhub zu gewährleisten. Der Wertebereich des Drucksensors würde wahrscheinlich überschritten werden und zu einem ungenauen Ergebnis führen.

Distanzsensor auf Infrarotbasis : Von den Ausmaßen zu Groß um das Innere des Rohres zu erfassen, außerdem erst ab einer zu großen Distanz akkurat.

Ultraschallsensor : Würde zwar von der Größe her passen, allerdings werden erst ab einer zu großen Distanz brauchbare Werte geliefert.



Magnetisches System auf Basis von Reed Sensoren : Erfolg ungewiss, da Anbringung von außen am Rohr. Die Röhre schirmt möglicherweise zu stark ab um brauchbare Ergebnisse zu erhalten. Des Weiteren wäre die Wartbarkeit des Systems stark eingeschränkt.

Da keines der Systeme in Betracht kam, wurde entschieden ein eigenes System bestehend aus einer LED sowie eines Lichtsensors zu verbauen. Die LED emittiert Licht in die Röhre, welches dann von dem Spiegel, welcher an der Unterseite des Rohres angebracht ist, reflektiert wird. Je weiter unten sich der Spiegel in der Röhre befindet, desto mehr Licht wird zurückgeworfen.

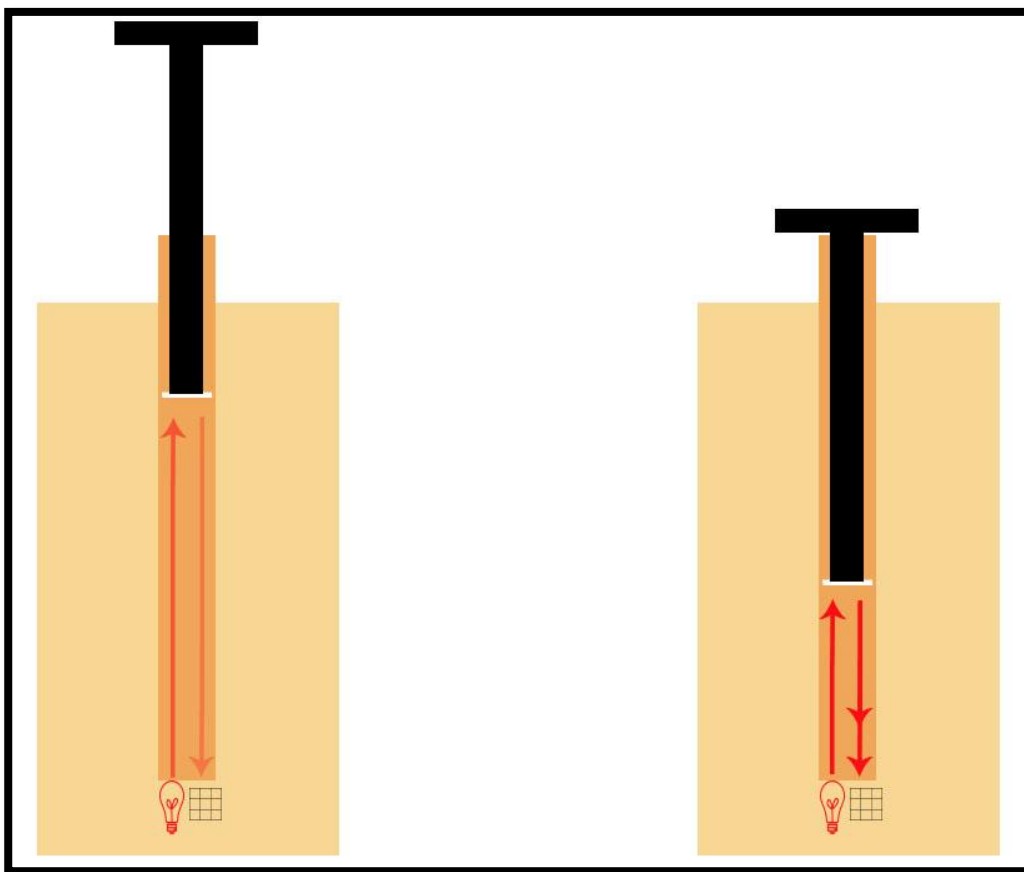


Abbildung 12: Änderung der Helligkeit beim Pumpenhub

Da die erste Version des Sensors keine zufriedenstellenden Daten lieferte, wurde eine zweite Version gebaut, mit welcher bereits präzisere Informationen abrufbar sind.

Der obere Bereich der Skala ist aber noch zu stark verrauscht.



Abbildung 13: Lichtsensor

Für einen Einsatzzweck außerhalb des Projekts, wäre es daher sinnvoll eine dritte Version zu fertigen, welche die gewünschte Präzision erreichen sollte.

## 11.3 Tracking

Um den Schlauch zur Eingabe zu benutzen, musste abgewogen werden welche Systeme dafür in Betracht kämen.

Lightgun : Aktuelle Lightgun-Systeme funktionieren ähnlich wie ein Wii Controller. Ältere Systeme sind nur mit Röhrenmonitoren kompatibel.

Wii Controller: Man würde einen Wii Controller samt Sensorbar, welche unter der Leinwand positioniert wird, benötigen. Der Controller wäre vom Formfaktor her problematisch, da er sich nicht in den Schlauch / in die Spitze integrieren lässt. Ein Selbstbauversuch wäre mit zu vielen unbekannten Faktoren verbunden gewesen und hätte das Projekt womöglich zu sehr

verzögert.



*Abbildung 14: Schlauchspitze mit Laserdiode*

Schließlich wurde in der Schlauchspitze ein Laserpointer integriert welcher über eine externe Kamera erfasst wird.

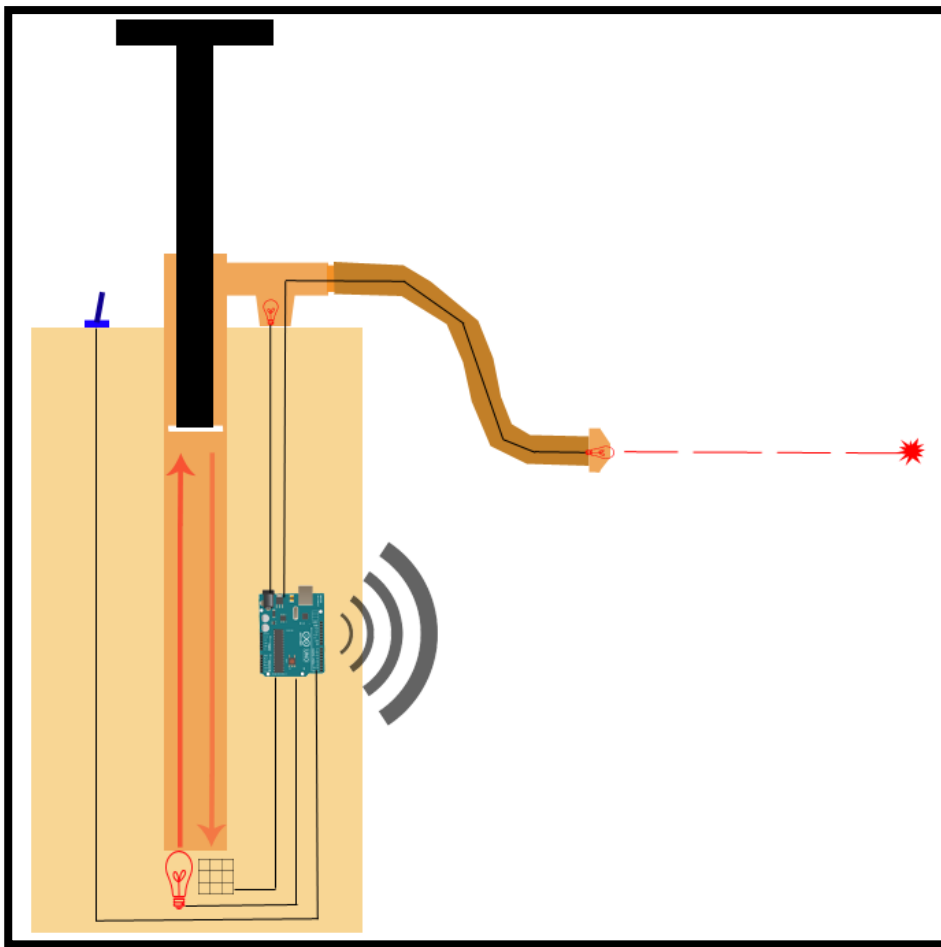


Abbildung 15: Controller schematisch

Die benötigte Software zum tracken des Laserpointers wurde in einem vorhergehenden Projekt mit Processing selbst entwickelt und für dieses Projekt angepasst und verbessert.

Sie erfüllt folgende Funktionen:

- Einlesen des Kamerabildes.
- Finden des hellsten/rotesten Pixels.
- Umrechnen der gefundenen Koordinaten von Start zu Zielauflösung (Webcam zu Bildschirmauflösung)
- Interpolation der Koordinaten um eine möglichst flüssige Bewegung zu Gewährleisten.
- Senden der Koordinaten ans Zielsystem (Spieleengine).

Hierfür gab es mehrere Möglichkeiten der Realisierung. Ein Versenden der Koordinaten mithilfe des OSC Protokolls wäre die stabilste Lösung, erfordert allerdings auch den höchsten Programmieraufwand, da sowohl im Sendesystem (Trackingsoftware) als auch im Zielsystem (Spieleengine) entsprechende Lösungen erarbeitet werden müssten. Als Alternative würde es sich anbieten den Mauscursor in der Trackingsoftware zu manipulieren. Dies bedeutet zwar, dass man am Computer auf den Mauscursor nicht mehr direkt Einfluss nehmen kann, ist allerdings auch einfacher und schneller umzusetzen, da für eine Realisierung in der Spieleengine bereits alle Tools zum Auslesen der Mausposition vorhanden sind. Außerdem können so die beiden Applikationen getrennt voneinander entwickelt werden, ohne dass es zu zeitlichen Verzögerungen kommt. Im Notfall, oder zur privaten Begutachtung der Professoren, kann so das Spiel auch komplett mit Maus und Tastatur gespielt werden.

Diese Umsetzung bietet außerdem den Vorteil, einfacher mit Spielen Dritter zu interagieren, ohne dass eine API notwendig wäre.

## 11.4 Komponenten

Laser Presenter, besteht aus den Komponenten:

- Rote Laserdiode: Die Farbe Rot schien für den Anwendungszweck am angemessensten, da sie sich natürlich ins Konzept eines Feuerwehrspiels einfügt. Andere Wellenlängen kamen nicht in Betracht, da sie entweder farblich nicht passend zum Konzept oder zu Dunkel oder Gefährlich für das Augenlicht sind. Die Leistung der Diode beträgt <1mw.
- Mikrocontroller : Regelt die Stromversorgung für die Laserdiode und baut eine Funkverbindung mit dem Computer auf. Wird vom Arduino-Mikrocontroller angesteuert.
- USB-Stick : Empfängt Funkdaten von Controller des Laser Presenters.

Arduino-Mikrocontroller: Dieser liest die Sensordaten ein und steuert den Controller des Laser Presenters an, um dem Computer über Funk mitzuteilen dass ein Pumpenhub erreicht wurde.

Kamera : Die Playstation Eye liefert 60 Bilder pro Sekunde bei einer Auflösung von 640 x 480 Pixeln.

## 11.5 Grenzen und Ausbaumöglichkeiten

Die aktuelle Version des Controllers ist zwar funktionsfähig, es bedarf aber noch mindestens einer weiteren Version um ein stabileres Auslesen des Pumpenhubs und ein akkurateres Trackingergebnis zu erreichen.

So ist es momentan nur möglich, dem Computer mitzuteilen, dass ein Pumpenhub erreicht wurde. Für erweiterte Gamemechaniken wäre es allerdings nützlich, die jeweils aktuelle Position der Pumpe zu übertragen. Hierfür ist eine Neukonzeption des Funkübertragungssystems notwendig. Da das Trackingergebnis stark von äußeren Einflüssen abhängig ist (Helligkeit des Raumes, Beschaffenheit der Leinwand), ist es gerade bei Einsätzen im Freien nötig, das System zu verbessern indem man einen stärkeren Laserpointer verbaut und eine bessere Kamera einsetzt.

LED-Leinwände wie sie oft bei großen Events eingesetzt werden, wären trotz der Verbesserungen am System ungeeignet. Hierfür müsste ein komplett neues Trackingsystem entwickelt werden.

Um dem User das Spielgefühl noch näher zu bringen, wäre es nützlich, bei zunehmendem Druck die Handhabung des Controllers zu erschweren.

In der aktuellen Version spürt der User keine Veränderung des Pumpgefühls wenn Druck aufgebaut wird. Ziel müsste es sein, dieses Pumpgefühl proportional zum Druck zu gestalten.

Durch ein steuerbares Ventil könnte dies erreicht werden.

Der Schlauch bleibt, auch bei maximalen Druck, regungslos.

Durch unkontrolliertes Ausschlagen des Schlauchs würde eine weitere anspruchsvolle Dimension ins Gameplay einfließen. Dies könnte durch einen Unwuchtmotor realisiert werden.

## 12 Das Game

Das Spiel enthält drei Grundlegende Ansichten, die im Folgenden Erläutert werden.

### 12.1 Menu

Es gibt ein einfaches Startmenü, in dem man starten oder beenden kann.



Abbildung 16: Startmenü

Durch zielen auf eines der Optionen, färbt sich der angepeilte Text. Um die Wahl zu bestätigen, ist es erforderlich die Pumpe zu betätigen.

### 12.2 Spielansicht

Die eigentliche Spielsitzung ist simpel und übersichtlich gestaltet. Auch für Erstspieler ist das Ziel und die Funktionen schnell verständlich.

Ziel ist es, mit dem Wasserstrahl die nach und nach auflodernden Brände zu löschen. Um einen Wasserstrahl zu erzeugen, ist es nötig die Pumpe zu betätigen. Zusätzlich lässt sich die Löschgeschwindigkeit durch schnelleres pumpen beeinflussen. Je schneller man pumpt, desto stärker der Strahl. Oben rechts im Bild sieht man einen Timer, der einem die verstrichene Zeit

anzeigt. Wenn man es schafft, alle Feuer im Bild zu löschen, ist das Spiel zu ende. Feuer die in einem Fenster gelöscht wurden, erscheinen kein zweites mal.

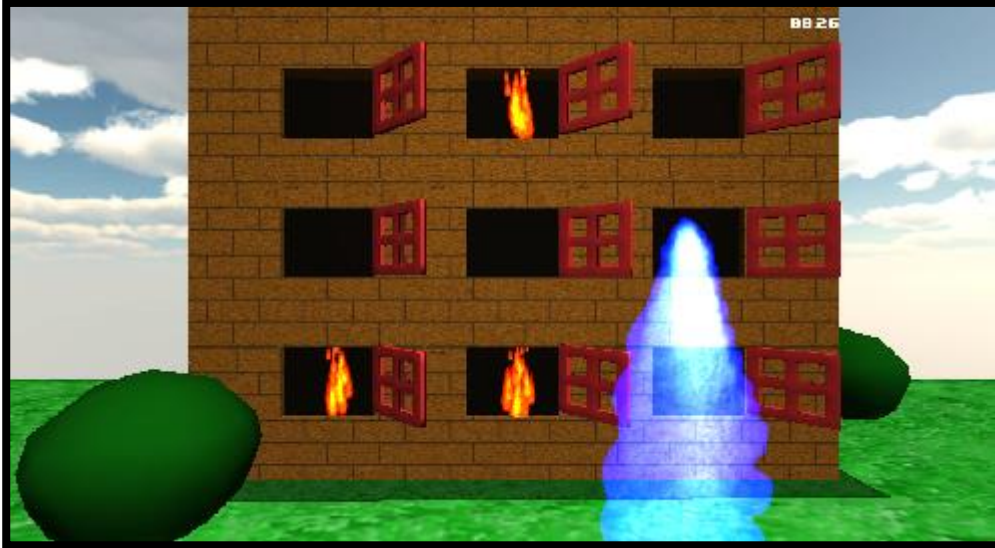


Abbildung 17: Spielfeld

## ***12.3 Spielende***

Das Spielende wird mit der benötigten Zeit bestätigt. Eine Bestenliste gibt es in der aktuellen Version noch nicht.





Abbildung 18: Spielende

## 13 Kosten

Die Kosten für den Controller belaufen sich auf 132€. Die verwendete Kamera befand sich bereits in unserem Besitz.

Artikel	Preis
Kübelspritze	45,89 €
Div. Einbaumaterialien	20,17 €
Laser Presenter	10,95 €
Arduino Mikrocontroller	18,90 €
Div. El. Bauelemente	25,80 €
Batterien	10,95 €
Summe	132,66 €

## 14 Zeitplan

	Bis KW 40	KW 41	KW 42	KW 43	KW 44	KW 45
<b>Martin</b>	Konzeption	Präsentation	Einarbeitung Arduino	Controller	Controller	Controller

<b>Johannes</b>	Konzeption	Präsentation				
<b>Sebastian</b>	Konzeption	Präsentation	Einarbeitung Blender	Konstruktion der Game- elemente	Konstruktion der Game- elemente	Einarbeitung Unity
	KW 46	KW 47	KW 48	KW 49	Bis KW 3	
<b>Martin</b>	Tracking	Controller	Aushilfe Unity	Präsentation	Dokumentation	
<b>Johannes</b>				Präsentation	Dokumentation	
<b>Sebastian</b>	Unity Logik und Design	Tests und Anpassungen	Komponenten zusammensetzen und Anpassun- gen	Präsentation	Dokumentation	

## 15 Quellenverzeichnis

[www.blender.org](http://www.blender.org)

[www.freesound.org](http://www.freesound.org)

[www.google.com/fonts](http://www.google.com/fonts)

[www.adobe.com/de/products/photoshop.html](http://www.adobe.com/de/products/photoshop.html)

[www.4players.de](http://www.4players.de)

[www.unity3d.com](http://www.unity3d.com)

## 16 Abbildungsverzeichnis

Abbildung 1: Wimmelbild-Spiel (PC-Wimmelbox) .....	4
Abbildung 2: Okami.....	4
Abbildung 3: Fruit Ninja .....	5
Abbildung 4: Duck Hunt .....	5
Abbildung 5: Fire Fighter by Dräger safety.....	6
Abbildung 6: Konzeptentwurf .....	7
Abbildung 7: Blender v2.67 .....	9
Abbildung 8: Photoshop.....	9

Abbildung 9: Font Beispiel .....	10
Abbildung 10: Kübelspritze .....	11
Abbildung 11: Erste Skizze des Mechanismus .....	12
Abbildung 12: Änderung der Helligkeit beim Pumpenhub .....	14
Abbildung 13: Lichtsensor .....	15
Abbildung 14: Schlauchspitze mit Laserdiode .....	16
Abbildung 15: Controller schematisch .....	17
Abbildung 16: Startmenü .....	20
Abbildung 17: Spielfeld .....	21
Abbildung 18: Spielende .....	21
Abbildung 19: Kosten .....	22
Abbildung 20: Zeitplan .....	22