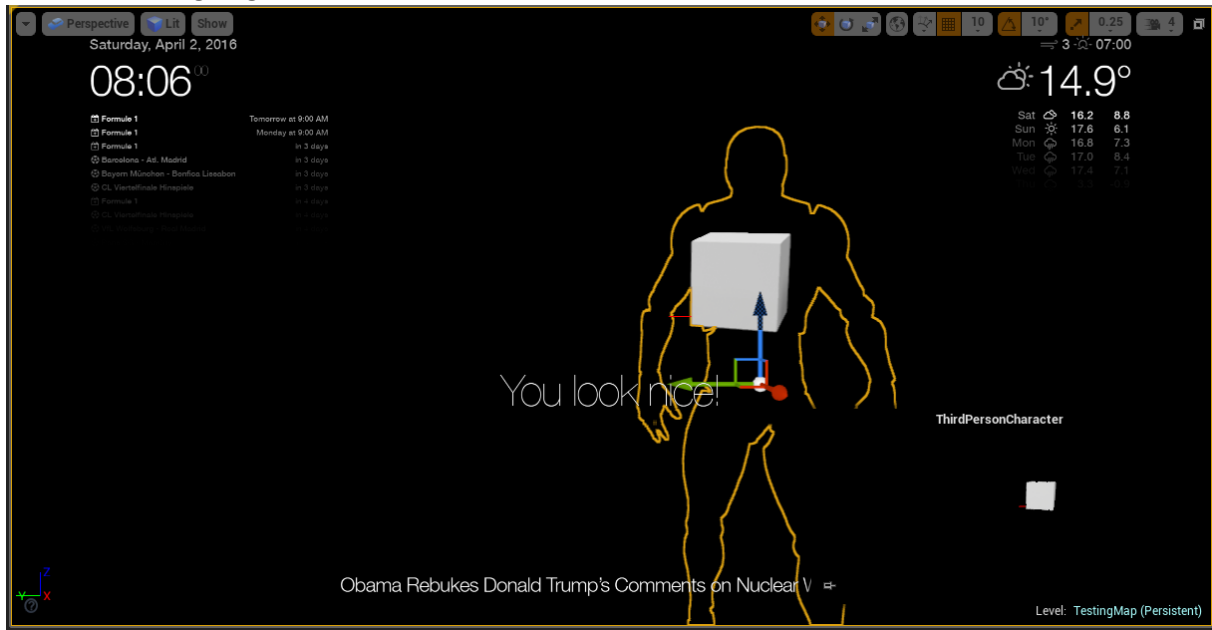


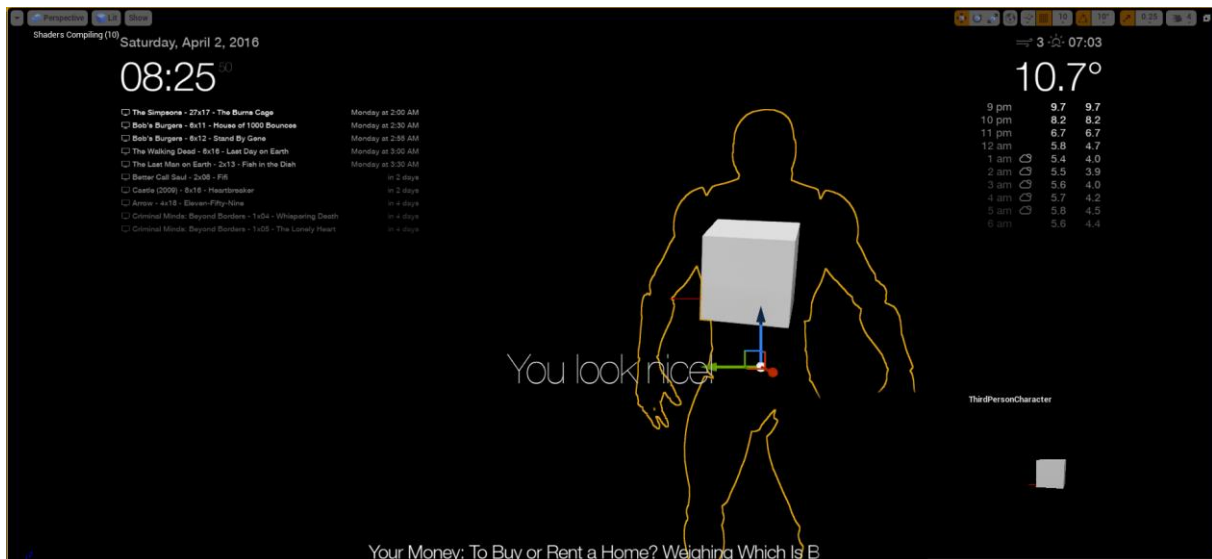
Protokoll Maturaarbeit: Virtual Wardrobe

2.4.2016:

- 3 Stunden: Testen von MagicMirror Github Projekt (<https://github.com/MichMich/MagicMirror>) in Unreal Engine 4. Testen ob durchsichtige Charakter Objekte tragen können. Weiter wurde die Wetterdaten API umgeändert. Zudem wurden ein Stündliche Wettervorhersage sowie aufkommende TV Shows hinzugefügt.



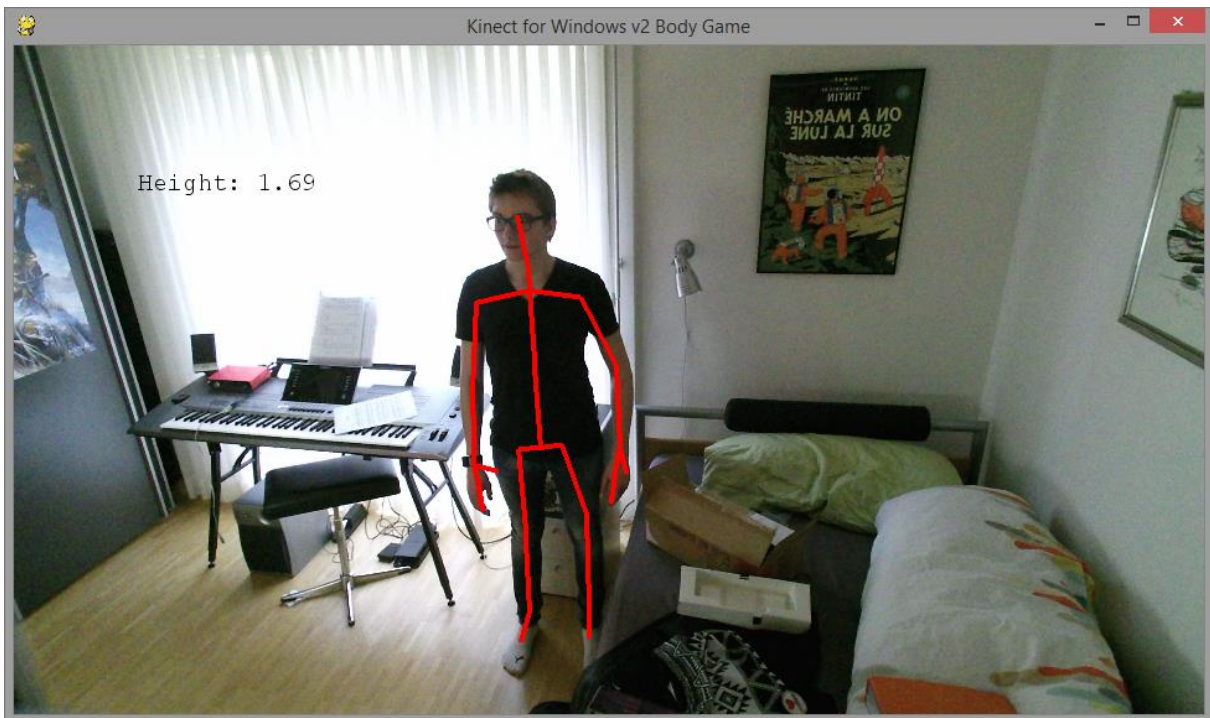
Vorher



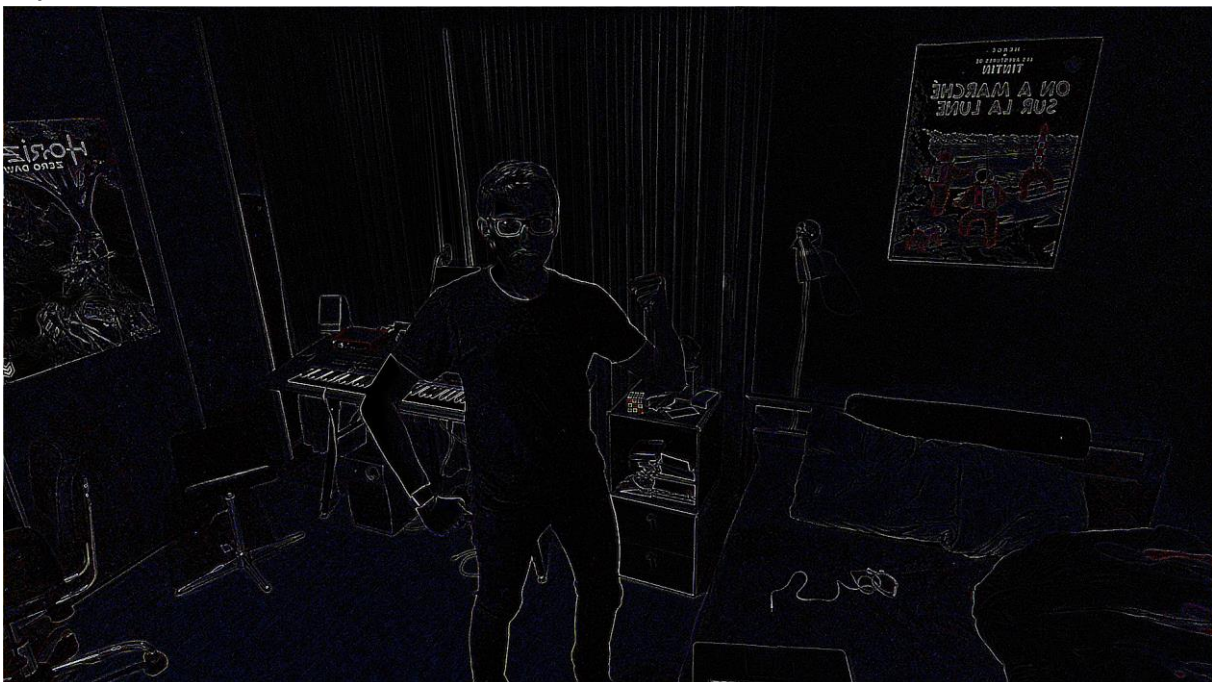
Nachher mit neuer API.

8.4.2016:

- 2 Stunden: Erstes testen mit Kinect. Einfaches Programm geschrieben für Erkennung der Grösse von Personen.

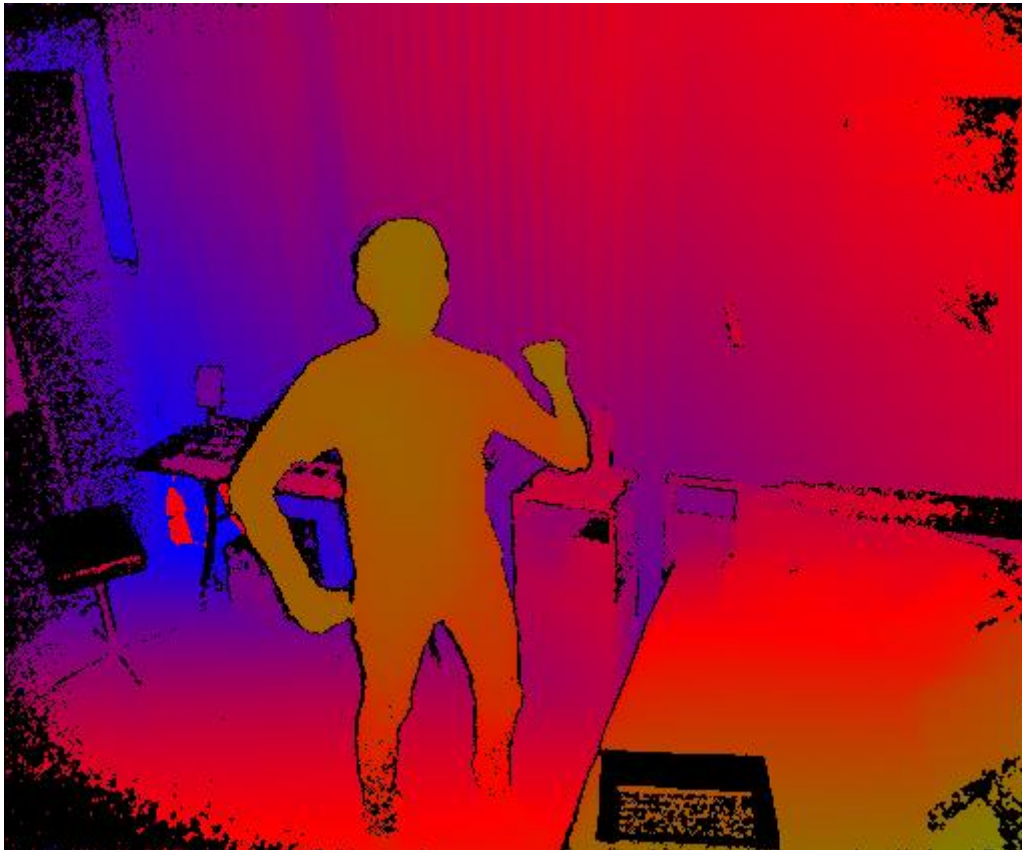
**9.4.2016:**

- 3 Stunden: Umschreiben der Daten, die von der Kinect kommen, auf Python verständliche Bilder. Damit konnten verschieden Filter ausprobiert werden, um zum Beispiel die Kanten von Personen und Objekten zu erkennen.

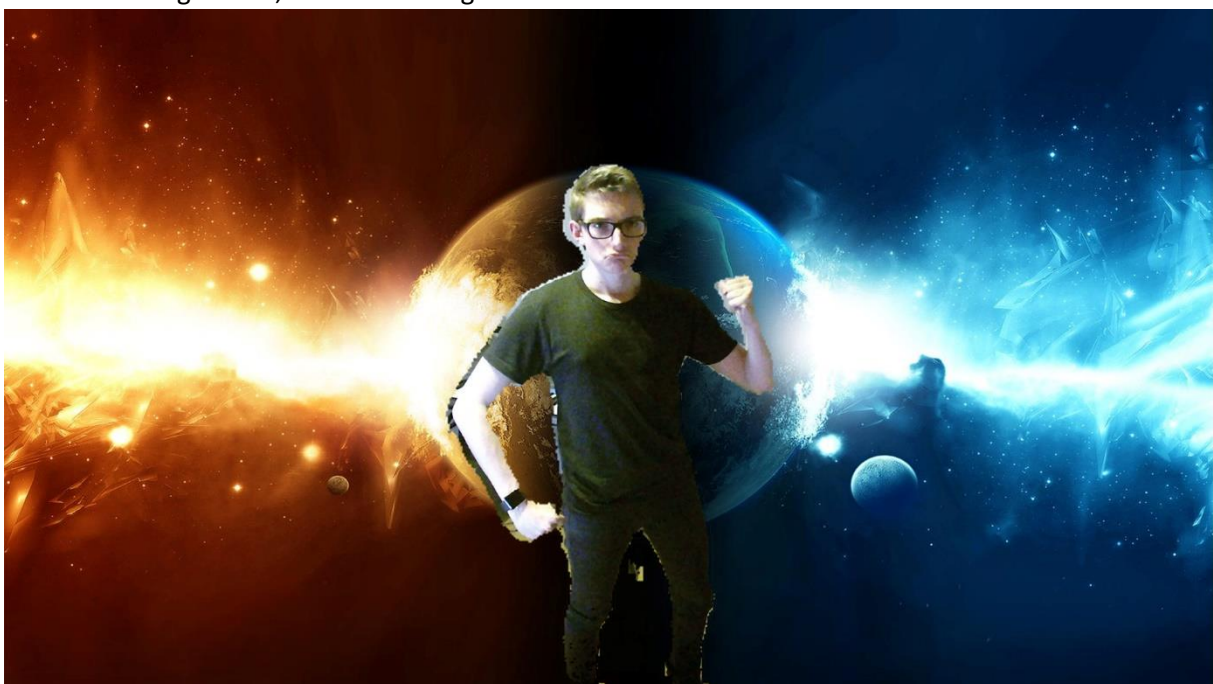


10.4.2016:

- 3 Stunden: Neu Programmieren der Verarbeitung von der Tiefenwahrnehmung der Kinect. Nun wird ein Objekt das nahe der Kinect liegt Grün. Wenn sich ein Objekt weiter von der Kinect entfernt, wird es zuerst Rot, dann Violett, Blau und dann Schwarz.



Weiter wurde getestet, wie der Hintergrund entfernt werden kann.



13.4.2016:

- 3 Stunden: Testen von konvertieren der Angaben der Tiefe in Koordinaten der richtigen Welt. Damit konnte ziemlich genau und konsistent die Höhe von Hüfte zu Genick bestimmt werden (Gemessene Höhe: 56cm):

```
hip to neck: 53.832893042186214 cm or using joints: 59.22208478402984  
Body distance: 1039.0
```

15.4.2016:

- 1.5 Stunden: Erstes testen vom T-Shirt einscannen. Dieser Algorithmus probiert Kanten eines T-Shirts zu erkennen und markiert diese Rot:



Obwohl der Algorithmus vorerst nur für einfarbige Flächen konzipiert wurde, funktioniert er schon bei Mustern gut. Als nächstes folgt eine Interpolation der roten Punkte damit der Rand besser bestimmt werden kann.

25.4.2016:

- 3 Stunden: Verbesserung des Algorithmus. Die Seiten werden nun auch erkannt. Weiter wird ein erstes interpolieren unternommen am unteren Rand des T-Shirts.

Vor Interpolation:





Nach Interpolation (Grüner rand). Somit sieht es schon sehr gut aus.

03.5.2016:

- 2.5 Stunden: Weiteres verbessern des Algorithmus. Neu werden auch die Seiten extrapoliert. Somit muss nur noch die obere Seite und die beiden Ärmel erkannt werden.

**11.5.2016:**

- 3 Stunden: Anwenden des Algorithmus auf Bilder direkt von der Kinect. Leider funktioniert er noch nicht so gut.

18.6.2016:

- 2 Stunden: Erstes Implementieren der Datenbank in Python und Anlegen der Datenbankstruktur.

19.6.2016:

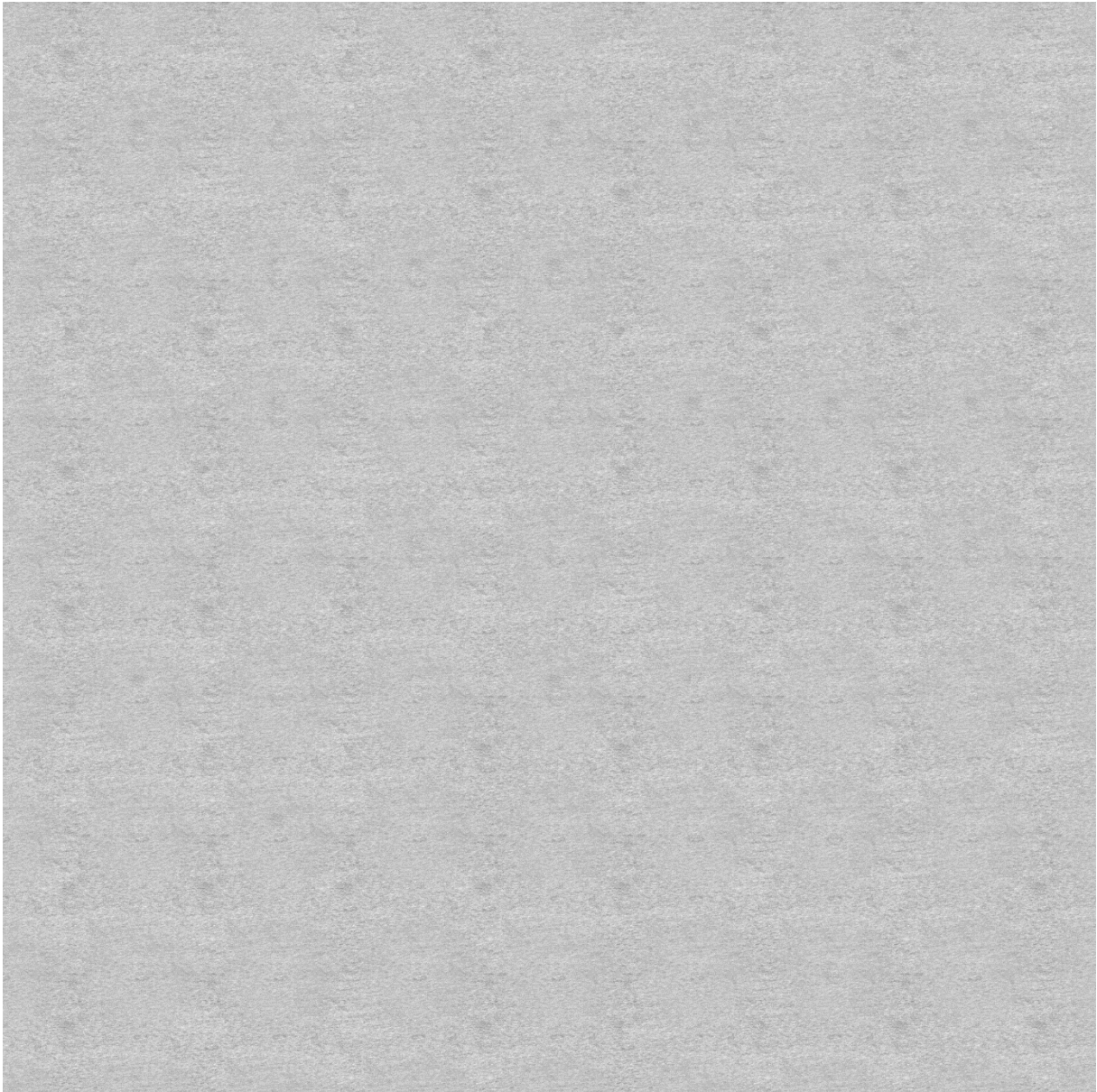
- 1.5 Stunden: Weiteres Implementieren der Datenbank in Python und Verbesserung der Datenbankstruktur.

22.6.2016:

- 4 Stunden: Implementieren verschiedener Algorithmen zur Textur Erstellung. Diese funktionieren neu dadurch, dass ein kleiner Teil des Kleidungsstück eingescannt wird und dieser wird dann vergrößert, so dass er auf ein 3D Model passt.

26.6.2016:

- 2.5 Stunden: Verbessern des Noise Algorithmus und Zusammenfügen mit anderen Algorithmen. Erstes Resultat einer vergrößerten Textur, welche nachher auf ein 3D Model gepflanzt wird:

**8.7.2016:**

- 1.5 Stunden: Austesten von Texturerkennung mit Hilfe von OpenCV2.

10.7.2016:

- 2 Stunden: Verbessern des Algorithmus um Texturen zu generieren. Weiter wurde ein erstes C++ Projekt mit Kinect SDK hinzugefügt.

12.7.2016:

- 4 Stunden: Erstes Scannen von T-Shirts in C++. Mit Hilfe von ImageMagick wird ein Canny Edge Detection durchgeführt. Im Moment funktioniert es aber noch nicht so gut. Weiter wurde am Texturen Algorithmus gefeilt.

18.7.2016:

- 6 Stunden: Erstes implementieren von Brücke zwischen C++ und Python. Weiter ist das T-Shirt Scannen nun voll funktionsfähig. Es wird erkannt wann eine einfarbige Farbfläche (T-Shirt) im Scanbereich vorliegt, und dann zwei verschiedene Textur Samples genommen. Ausserdem wurde ein neuer Algorithmus zur Textur Generation erstellt, der viel besser funktioniert als vorher. Auch wurden erste Schritte von Python in C++ zu integrieren unternommen.

19.7.2016:

- 2 Stunden: Verbessern der C++ zu Python Brücke und des T-Shirts Scannen.

9.8.2016:

- 1 Stunde: Refractoring und hinzufügen des KinectV2 Plugins zum UE4 Projekt.

10.8.2016:

- 6 Stunden: Hinzufügen des Wardrobe Managers zum UE4 Projekt. Weiter wurde versucht das Kinect SDK in diesen einzubetten. Leider hatte das nicht so gut funktioniert, da das Polling von den Bildern nicht synchron mit der Kinect war.

11.8.2016:

- 7 Stunden: Umschreiben des Plugins, damit C++ Funktionen die vollen Bilddaten bekommen. Weiter wurde auch hier die Python Brücke implementiert und dies versucht auf einem anderen Thread zu machen, damit es nicht das UI blockiert.

12.8.2016:

- 7 Stunden: Die Kommunikation zwischen C++ und Python wurde im Grunde fertiggestellt. Weiter wurden mehrere Memory Leaks in dieser Brücke gestopft.

13.8.2016:

- 4 Stunden: UI hinzugefügt und Python Brücke verbessert.

14.8.2016:

- 3.5 Stunden: Modus hinzugefügt bei dem man den Charakter ankleiden kann.

15.8.2016:

- 6.5 Stunden: UI wurde drastisch verbessert. Es wurde ausserdem ein Modus hinzugefügt bei dem man Aufdrucke einscannen kann. Im Moment funktioniert hier die Python Brücke noch nicht.

16.8.2016:

- 4 Stunden: Aufdruck einscannen funktioniert nun komplett, mit Python Brücke. Ausserdem ist das Aufdruck einscannen nun nicht mehr abhängig von der Helligkeit eines T-Shirts.

17.8.2016:

- 6 Stunden: Mehrere Kleider können nun ausgewählt werden. Auch mehrere Schichten, daher: Hoodie und T-Shirt. Ausserdem werden die 3D Modelle der Kleidungsstücke dynamisch geladen. Weiter können jetzt auch andere Kleidungsstücke eingescannt werden. Im Moment ist aber nur der Hoodie auch wirklich integriert. Zudem wurde der Hintergrund noch angepasst.

18.8.2016:

- 2 Stunden: Nun kann man auch Jeans und Shorts einscannen.

19.8.2016:

- 5 Stunden: Das Projekt wurde nun für andere Plattformen (OSX, iOS) optimiert und sollte darauf Kompiliert werden können. Weiter wurde ein Crash verhindert der entsteht wenn die Datenbank nicht vorhanden ist.

25.8.2016:

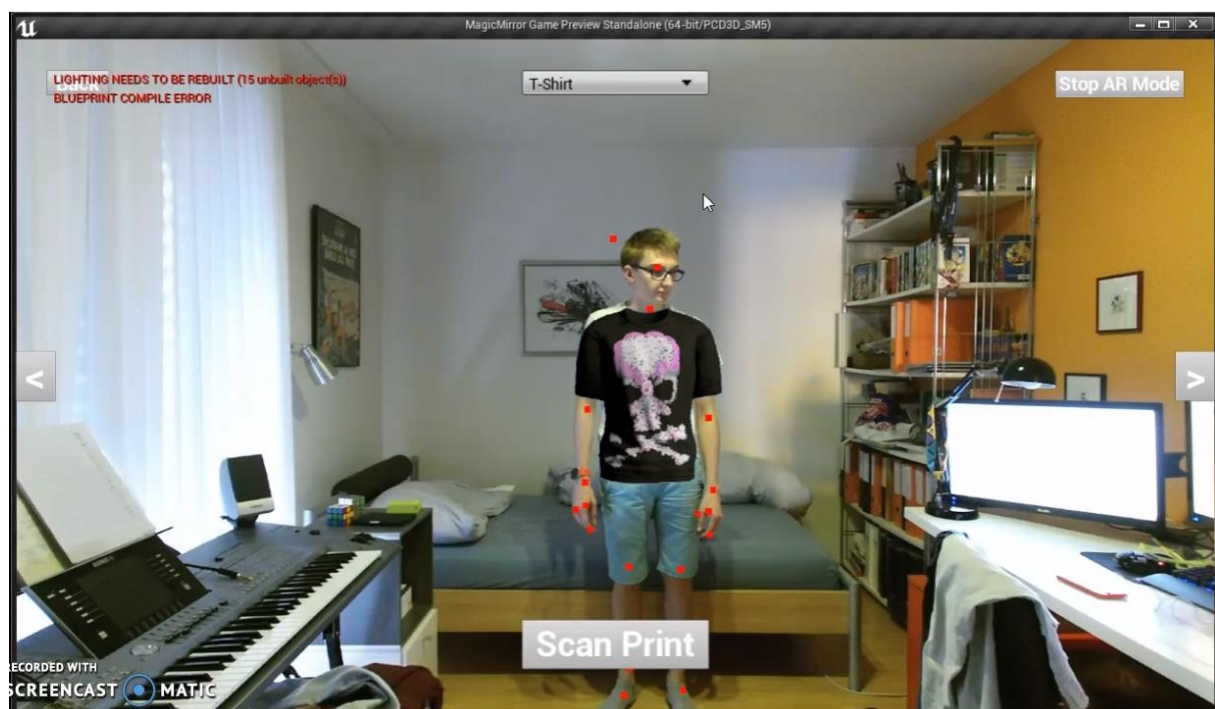
- 4.5 Stunden: Es wurden nun sqlite Bibliotheken für iOS hinzugefügt und weitere kleine Probleme behoben. Weiter wurde auch eine neuer Level hinzugefügt der wie ein Zimmer aussieht. Er beinhaltet ein Spiegel und einen Schrank.

28.8.2016:

- 3 Stunden: Erste Schritte in Richtung von einer AR Funktion wurden unternommen.

31.8.2016:

- 7.5 Stunden: Die AR Funktion ist nun schon sehr weit fortgeschritten und funktioniert schon recht gut:



10.10.2016 & 11.10.2016:

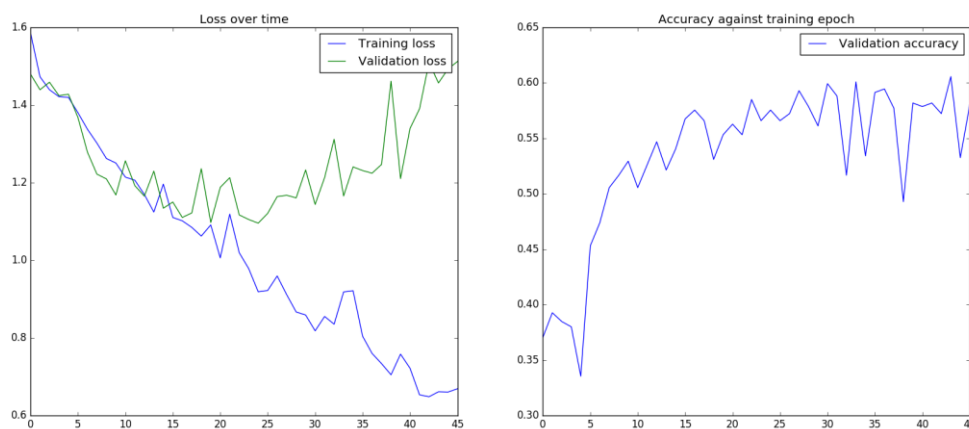
- 8 Stunden: Lesen des <http://neuralnetworksanddeeplearning.com/> Buches um eine Idee zu bekommen wie ein Deep Learning System funktioniert.

12.10.2016:

- 6 Stunden: Erstes testen von Deep Learning zur Erkennung von Handgeschriebenen Zahlen.

18.10.2016:

- 3.5 Stunden: Erster Versuch ein «Convolutional Neural Network» aufgrund vom Image-net.org Datensatz zu trainieren. Leider blieb die «Validation Accuracy» bei ca. 60% stehen und es entstand ein grosses «Overfitting»:

**19.10.2016-22.10.2016:**

- 8 Stunden: Schreiben des 2. Kapitels: Technical Introduction.

12.11.2016:

- 2 Stunden: Versuch mit Hilfe von Google Images die Anzahl von Trainings Daten zu erhöhen und somit bessere Resultate zu bekommen. Leider waren die Bilder von Google nicht besonders gut und es blieb die «Validation Accuracy» bei ca. 60%.

03.12.2016:

- 6 Stunden: Integrieren des CNNs in die finale Applikation. Die C++ Python Brücke ist nun auch viel zu verlässlicher, da Thread Locks integriert wurden.

04.12.2016 & 05.12.2016:

- 7 Stunden: Schreiben des Kapitels 3.1: Creating Textures from Clothes, sowie Anfänge des Kapitels 3.2: Wardrobe Manager.

10.12.2016:

- 4 Stunden: Fertigstellung von Kapitel 3.2, sowie Schreiben des Kapitels 3.3: Data Storage und 3.4: Combining the C++ and Python Parts.

11.12.2016 & 12.12.2016:

- 6 Stunden: Anfang des Kapitles 4.1: Deep Learning Explained geschrieben.

15.12.2016 & 17.12.2016:

- 8 Stunden: Kapitel 4.1 und Kapitel 4.2: Creating a CNN to Recognize Clothes wurden fertiggestellt.

17.12.2016:

- 1.5 Stunden: Integrieren von ImageNet.org Datensatz und somit extremes Verbessern der Kleider Erkennung.

18.12.2016:

- 4 Stunden: Kapitel 5: Outlooks wurde fertig geschrieben.

19.12.2016:

- 6 Stunden: Kapitel 6: Conclusion und Kapitel 7: Evaluation wurden verfasst. Weiter wurde die ganze Arbeit überarbeitet und das Kapitel 3.5: Final Application hinzugefügt.

21.12.2016:

- 4 Stunden: Generelles Überarbeiten des Textes und hinzufügen des Abstracts, sowie des Kapitel 1: Introduction.

22.12.2016:

- 3.5 Stunden: Fertigstellung der gesamten Arbeit und letzter Feinschliff an der Applikation.