

Dokumentation zur App "Grüni"

Robert Ackermann

Fachhochschule Erfurt
robert.ackermann@fh-erfurt.de

Dennis Krischal

Fachhochschule Erfurt
dennis.krischal@fh-erfurt.de

Hannes Dröse

Fachhochschule Erfurt
hannes.droese@fh-erfurt.de

Livia Schumm

Fachhochschule Erfurt
livia.schumm@fh-erfurt.de

ABSTRACT

Im Folgenden behandeln wir Anfänge, Entwicklung und Ergebnisse des Semester-Projektes "Grüni" im Rahmen des Medienprojekt-Moduls. Grüni ist die Vision einer Anwendung um Nutzern das Gärtnern sowie das gärtnern lernen zu erleichtern.

KEYWORDS

gärtnern, lernen, AR, Vision, Zukunft, Pflanzen, Technik

1 DIE IDEE

Die Idee einer Gärtnerhilfe-Anwendung entstand bereits Anfang April 2020 während des ersten Brainstormings. Im ersten Treffen des Moduls, am 20.04.2020 haben wir vier - Robert Ackermann, Hannes Dröse, Dennis Krischal und Livia Schumm - uns in einer Projektgruppe zusammengefunden. Am selben Tag hat unser erstes Webex-Meeting stattgefunden, in dem über verschiedene Ideen beratschlagt worden ist. Für die Aufgabenstellung mit dem Motto "Future Interfaces" sollen zwei Teilprojekte abgedeckt werden:

1. Die Entwicklung einer **Vision**, also die Konzeption einer zukunftsweisenden intuitiven Interaktion mit digitalen Inhalten und deren Darstellung in Form eines Visionsvideos
2. Die Entwicklung eines **Prototyps** dieser Vision, sprich die Ausarbeitung einer interaktiven Anwendung

Unsere Projektidee soll also sowohl innovativ als auch in Teilaspekten umsetzbar sein, jedoch sollten wir uns vorerst mehr auf die Vision als auf den Prototypen konzentrieren. Es entstand eine Auswahl folgender Projektthemen:

- Gardening - App zum Gärtnern lernen, unterstützt und lehrt das Gärtnern
- Kontaktloses Bezahlen und papierlose Kassenbons
- Verwaltung - Online Identitätsfeststellungen, Digitale Formulare
- Antlitzdiagnostik - Apps analysieren den Gesundheitszustand, appgestützte Ferndiagnose mit Ärzten

Der gemeinsame Favorit hat sich im Gespräch jedoch schnell abgezeichnet und nach einer kurzen Themenabstimmung mit den betreuenden Dozenten hat die Richtung in den Gardening-Bereich festgestanden.

Recherche

Der nächste Schritt, nachdem eine Projektrichtung festgesteckt worden ist, ist die Recherche in diesem Bereich. Im Fall des Gardenings stellen sich vor allem drei Fragenfelder:

1. Wie sieht Gärtnern aktuell überhaupt aus, woran besteht Bedarf und in welche Richtung gehen die Trends der Zukunft?
2. Welche Apps, Anwendungen und Projekte gibt es bereits und welche sind in der Entwicklung?
3. Wie sehen die technischen Möglichkeiten aus? Welche Sensoren, Messtechniken o.ä. werden in der Landwirtschaft gebraucht oder sind im Garten einsetzbar?

Anmerkung:

Im Folgenden werden nur einige, wesentliche Aspekte dieser Recherche zusammengetragen, weitere gesammelte Informationen und Artikel finden sich in unserem Projekt unter den Recherchen im Management-Tool Asana.¹

Trends und aktuelle Entwicklung des Gärtnerns

Dass die Beliebtheit von Gärten, Pflanzen und Natur steigt, merkt schon alleine, wer sich im eigenen Umfeld umsieht. Zahlreiche Artikel, Blog- und Social Media-Beiträge zeigen die Trends des **Urban Gardening**, des **Urban Jungles**, des **DIY** (Do It Yourself) auf.

Laut dem Zukunftsinstitut² ist die Zahl der Menschen mit Garten oder Balkon in Deutschland zwischen 2007 und 2011 von 50 Millionen auf 55 Millionen gestiegen. Ebenso sinkt

¹Allgemeine Recherche und Trends: <https://app.asana.com/0/1172859492234369/1172897938417709>
Pflanzen Apps: <https://app.asana.com/0/1172859492234369/1172897938417707>

Sensoren und Messtechnik: <https://app.asana.com/0/1172859492234369/1172859492239381>

²Artikel "Die Zukunft ist ein Garten" - <https://www.zukunftsinstitut.de/artikel/wohnen/die-zukunft-ist-ein-garten/>

das Durchschnittsalter von Kleingarten-Pächtern stetig ab.³ Der Trend geht hin zur Begrünung von Gärten, Balkonen, Terrassen, zur Gründung sowie Nutzung von Gemeinschafts- und Nachbarschaftsgärten – vor allem bei der naturhungrigen Stadtbevölkerung. Die Gartenbranche stellt sich langsam auf eine **neue, jüngere Zielgruppe** ein. Die steigende Beliebtheit des Gärtnerns jeglicher Art liegt unter anderem im Ausgleich, den es zum stressigen Stadtalltag bieten kann, aber auch in seiner guten Vereinbarkeit mit dem Nachhaltigkeits-Trend. Verschiedene Faktoren lassen haufenweise individuelle und clevere Ansätze entstehen. Die Nachfrage nach Bio-Produkten und Nutzpflanzen und dafür möglichst einfachen aber autonomen Systemen ist groß.

Der Platzmangel in Städten verlangt z.B. nach vertikalen Lösungen:

- “Pflanzetageren”: Versetzt angeordnete, kleeblattförmige Gefäße, die sich hoch stapeln lassen
- “Minigarden”: vertikales Pflanzsystem, Begrünung von Wänden oder kompletter Balkonbrüstung
- “Skyplanter”: Pflanzen wachsen in einer Art umgedrehtem Blumentopf von der Decke herab nach unten

Doch nicht nur das Interesse in Richtung zurück zum Ursprünglichen – zur Natur – ist groß, sondern auch innovative, konnektive Lösungen sind gefragt, die das Leben in Haus und Garten durch **smarte Technik** vereinfachen. In seinem Vortrag auf dem DIY Garden Summit 2017 in Berlin⁴ erklärt Christian May, Geschäftsführer von Kärcher, wie ein Tag im Garten des Jahres 2030 aussehen könnte:

Während eines gemütlichen Frühstücks in der Sonne auf meiner Terrasse fällt mein Blick auf den Grill: Ihn umgeben unschöne Überbleibsel vom gestrigen Grillfest. Doch dem Hochdruckreiniger fehlt der spezielle Aufsatz für Rußflecken. Schnell ist die passende Vorlage gefunden und kurz darauf druckt der 3D-Drucker den fehlenden Aufsatz aus und die Arbeit kann beginnen. Da ich nun schon einmal im Garten bin, kümmere ich mich auch gleich um meine Blumenbeete. Während ich mich umschaue, zeigen mir einzelne Pflanzen auf meiner Smart-Brille an, dass sie nicht genug Sonne bekommen und umgepflanzt werden müssen. Am Nachmittag ist es Zeit für meinen Videokurs

zum Thema Pflanzen, Pflegen und Ernten seltener Gemüsesorten – meinem Steckenpferd. Meine Expertise auf dem Gebiet teile ich gerne und inzwischen lauschen meinen interaktiven wöchentlichen Livestreams interessierte Hobbygärtner und Landwirte aus Kanada, Südafrika und Japan, die mich wiederum an ihrem Wissen teilhaben lassen. Am Abend bin ich mit Freunden zum Abendessen verabredet. Während der Vorspeise bekomme ich eine Nachricht: Die Sonne brannte heute den ganzen Tag und meine Pflanzen sind durstig. Jetzt, wo es langsam schattiger wird, ist der perfekte Zeitpunkt, meine Rosen zu bewässern. Per Knopfdruck bestätige ich, das automatische Bewässerungssystem legt los und ich kann mich wieder in Ruhe meiner Vorspeise widmen.

Aus diesem Beispiel gehen gleich mehrere mögliche Zukunftsaspekte hervor:

- konnektive Lösung mittels 3D Drucker zum Drucken von Ersatzteilen und Hilfsmitteln
- Smart-Brille erfasst Umgebungs- und Pflanzenzustand (Bewässerung, Sonnenbelichtung etc.)
- Austausch von Wissen über Vernetzungskomponente
- smarte Auswertung verschiedener, zusammenspieler Informationen (Gießzustand der Pflanzen plus Ermittlung des optimalen Zeitpunktes)
- automatisierte Systeme zur Versorgung (in dem Fall Bewässerungsanlage per Knopfdruck)

Technisch relevante Daten der Pflanzenwelt

Aufwendige und smarte Technik wird zurzeit vor allem im Großanbau-Rahmen, also in der Landwirtschaft, eingesetzt. Das **Digital Farming** hat sowohl in der Tierindustrie als auch im Anbau Einzug gehalten – die Landschaft steckt mitten drin in der Digitalisierung, neudeutsch: “**Smart Farming**”.⁵ Für die Landwirte bietet das sowohl Vor- als auch Nachteile, im Hinblick auf ein interaktives Gardening-Projekt sind aber vor allem die technischen Möglichkeiten relevant und was man aus ihrem Einsatz in der Landwirtschaft lernen kann. Allgemein dient Smart Farming der Optimierung von Planung, Effizienz und Ertrag.

Precision Farming: Komplettangebote von IT-Konzernen zur Optimierung der Pflanzen für die Weiterverarbeitung (z.B. Berechnung des Abstands zwischen Pflanzen, damit sie die richtige Größe bekommen). Mithilfe von Sensoren auf dem Feld und in den Böden, vernetzten Maschinen und entsprechenden Analyse-Softwares, die

³ Artikel vom 22.11.2017: “Blick in die Zukunft: So leben wir im Garten 2030” - <https://taspo.de/kategorien/blick-in-die-zukunft-so-leben-wir-im-garten-2030/>

⁴ Artikel vom 22.11.2017: “Blick in die Zukunft: So leben wir im Garten 2030” - <https://taspo.de/kategorien/blick-in-die-zukunft-so-leben-wir-im-garten-2030/>

⁵ vgl. Artikel vom 10.11.2019, Link: https://www.deutschlandfunk.de/digitalisierung-der-landwirtschaft-daten-saen-daten-ernten.740.de.html?dram:article_id=462957

zusätzlich benötigte Daten, wie Geo- und Wetter-Analysen, mit einbezogen wird für eine großflächig abdeckende, informationstechnische Infrastruktur gesorgt.⁶

Der **Einsatz von Drohnen** und **Auswertung von Satellitenbildern** hilft bei der Planung von Aussaat, Pflege und Ernte:

- BayWa-Drohnen ermöglichen beispielsweise eine automatische Schädlingserkennung und -bekämpfung.⁷
- Das Copernicus-Programm beinhaltet zum Beispiel Satellitenfernerkundungssysteme, die verschiedene Daten (Pflanzenstrukturen und Bodenbewegungen, Landbedeckung und Landnutzung) zum optimalen Düngermanagement sammeln.
- Positionsgenaue Regenradar-Apps dienen der optimalen Bewässerung von Feldern.⁸

Farm Management: Eine datengestützte, automatische Dokumentation spart Zeit. In Zukunft soll sogar nur noch geringfügig menschliche Arbeit vonnöten sein, indem Maschinen direkt mit Maschinen kommunizieren.⁹

Herausforderungen für die IT stellen sich hauptsächlich durch die Verwaltung enormer Datenmengen und deren Austausch über das mobile Netz. Gerade in den ländlichen Regionen, in denen Landwirtschaft stattfindet, stellt eine nicht vorhandene Flächendeckung von Netzzugang mit hoher Bandbreite¹⁰ teilweise ein Problem dar.

Zusammenfassend kann man aus dem Beispiel der Landwirtschaft folgende Daten als relevant für jegliche Form von Pflanzenanbau ableiten:

- Bodenbeschaffenheit (Zusammensetzung, Nährstoffe, Durchlässigkeit, Bodenfeuchtigkeit)
- Zustandsanalysen (Größe, Abstände, Schädlinge)
- Wetterdaten (Belichtung, Bewässerung, Unwetter)
- Geo-Daten (Pflanzenstrukturen und Bodenbewegungen, Landbedeckung und Landnutzung)
- Pflanzendaten (Wissen zur Pflanzenart und ihren Bedürfnissen)

Um diese Daten zu gewinnen, bedarf es verschiedener Quellen. Für präzise, lokale Daten bedarf es oft festinstallierter Hardware (Sensoren), die sich wohl nicht bewährt

hat.¹¹ Der Trend entwickelt sich immer mehr in Richtung mobiler Analysen über Roboter und Drohnen. Je größer der Anbau-Rahmen, desto mehr lohnt sich logischerweise großangelegte, technische Systeme. Die Frage ist nun, wie sich solche Systeme flexibel genug an den Privatgebrauch und dessen technische Gegebenheiten anpassen lassen, um auch für diese Zielgruppe lohnenswert zu sein.

Mit dieser Frage beschäftigen sich auch Garten-Dienstleister, Unternehmer und Start-Ups - welches Angebot dafür bereits besteht, behandelt der nächste Abschnitt.

Bereits bestehende Projekte

Interaktive, technische Anwendungen im Bereich des Gardening gibt es auch für die Otto-Normalverbraucher, wenn auch nicht in so ausgefeiltem Stil wie in der Landwirtschaft. Meist greifen sie eine unterschiedliche Auswahl an den im vorigen Abschnitt erläuterten Aspekten auf. Man kann sie anhand der Funktionalität jedoch grob unterteilen in:

- visualisierende Planungs-Tools
- einfache Erinnerungs-Anwendungen
- Pflanzenerkennungs-Apps und
- smarte, teils automatisierte Beete.

Produkte und Projekte gibt es dazu viele. Im Folgenden werden zur Verdeutlichung jeweils einige Beispiele vorgestellt.

Visualisierungs-Software.

- **Garten-Planer:** Beispiel "Home Design 3D Outdoor & Garden"¹² – Eine App, die einen virtuellen Rundgang im eigenen Garten ermöglicht.
- **Beet-Planer:** Beispiel "Alphabeet"¹³ – Wer sich hier anmeldet, kann zunächst einmal seine Beete digital anlegen. Je nach Größe wird vorgeschlagen, welche Pflanzen wo angebaut werden sollten und auch zueinander passen. Ist das Beet fertig geplant, wird automatisch eine Aufgabenliste erzeugt, die per Mail an die täglich nötigen Arbeiten erinnert und online bei Erledigung abgehakt werden können: Tomaten ausgeben, Unkraut entfernen, Stecklinge ziehen, Möhren abdecken, Gießen und vieles mehr.

Hierbei stehen also drei Aspekte im Vordergrund: Planen, Umsetzen und Lernen anhand der Infobibliothek.

⁶vgl. Artikel vom 10.11.2019, Link: https://www.deutschlandfunk.de/digitalisierung-der-landwirtschaft-daten-saen-daten-ernten.740.de.html?dram:article_id=462957

⁷vgl. Artikel vom 23.06.2017, Link: <https://biooekonomie.de/digitale-landwirtschaft-it-fuer-acker-und-stall>

⁸vgl. Artikel vom 23.06.2017, Link: <https://biooekonomie.de/digitale-landwirtschaft-it-fuer-acker-und-stall>

⁹vgl. Artikel vom 21.07.2020, Link: <https://www.computerwoche.de/a/was-sie-ueber-landwirtschaft-4-0-wissen-muessen,3544215>

¹⁰vgl. Artikel vom 23.06.2017, Link: <https://biooekonomie.de/digitale-landwirtschaft-it-fuer-acker-und-stall>

¹¹vgl. Artikel vom 10.11.2019, Link: https://www.deutschlandfunk.de/digitalisierung-der-landwirtschaft-daten-saen-daten-ernten.740.de.html?dram:article_id=462957

¹²Link: <https://en.homedesign3d.net/2018/10/09/update-v4-2-outdoor-garden/>

¹³Link: <https://www.alphabeet.org/>

Gieß- und Pflegeerinnerungs-Apps. Dabei handelt es sich um reine Organisations-Tools, in denen man Aufgaben verwalten und Erinnerungen erstellen kann. Sie ermöglichen außerdem Zugriff auf Datenbanken, die dem User nützliche Informationen zu seinen Pflanzen bieten.

Beispiele für diese Art App sind:

- Gardenia (iPhone // Android)
- myPlants - Manage Tool und Reminder (iPhone // Android)
- PeppyPlant (iPhone)
- Happy Plant (iPhone)
- Plant Watering Reminder (iPhone)
- Plant Diary (Android)

Pflanzenerkennungs-Apps. Die Hauptfunktion solcher Apps ist die Erkennung von Pflanzen anhand von Fotos. Sie verfügen ebenso über eine Pflanzenenzyklopädie, aus der der User weitere Informationen erlangen kann. Außerdem wird ein Austausch über eine Community ermöglicht, über die die App hauptsächlich funktioniert – je größer und aktiver die Community desto größer die Datenbank, desto präziser also auch die Bilderkennung.

Beispiele für diese Art App sind:

- PictureThis¹⁴
- PlantNet¹⁵
- PlantSnap¹⁶
- Garden Flower Identification¹⁷

Smarte, automatisierte Beete. Beispiel: Start-Up **IP Garten**¹⁸

Hierbei handelt es sich um ein Dienstleistungsunternehmen, bei dem man Beete buchen kann. Diese werden live überwacht mittels Kameras und Feuchtigkeitssensoren. Interagiert werden kann von zuhause aus über eine virtuelle Welt auf dem Computer. Eine ferngesteuerte Bewässerung ist für den Kunden möglich, alle weiteren Gärtner-Leistungen werden hinzugebucht und von Mitarbeitern durchgeführt. Der Ertrag des gebuchten Beetes kann nach der Ernte abgeholt oder geliefert werden.

Beispiel: **Urban connected Gardening mit smartem Hochbeet**¹⁹

Dieses Projekt entstand durch eine Kooperation aus "Smartgreen Solutions" und "T-Mobile Austria". Im Vergleich zum ersten Beispiel ist dieses Smartbeet-System autark. Es

wird mit Regenwasser und Solarenergie betrieben. Ausgestattet mit Sensoren, wird eine Vielzahl an Faktoren gemessen: Temperatur, Feuchtigkeit, Sonneneinstrahlung, Stromverbrauch, Wasserdurchfluss etc. Außerdem hat Smartgreen Solutions eigens eine elektronische Steuereinheit entwickelt, um das Beet automatisch reagieren zu lassen. Diese Steuereinheit sendet zudem Daten über T-Mobiles IoT-Box an die Smartgreen-Cloud, wodurch der User sie auf der eigenen App abrufen und kontrollieren kann. Die IoT-Box wiederum versorgt die Steuerung mit Daten, wie zum Beispiel Wetterdaten.

Smart und automatisiert bedeutet also immer, dass eine gewisse Hardware-Ausstattung vonnöten ist und muss lokal installiert sein. Je aufwendiger die Ausstattung, desto mehr Möglichkeiten ergeben sich für das Beet.

Anmerkung:

Eine zusammengetragene Liste weiterer Crowdfunding Projekte zu dem Thema findet sich in unserem Projekt unter den Recherchen im Management-Tool Asana.²⁰

Festlegung des Projektrahmens

Aufgrund der Fülle an Themengebieten und bereits existierenden Projekten, soll nun die Vision und ihr potenzieller Wirkungsraum von den recherchierten Projekten abgegrenzt werden.

Es soll sich um kein landwirtschaftliches Projekt, sondern um eine Anwendung für den Privatgebrauch handeln. In diesem Rahmen soll sie jedoch so universell wie möglich einsetzbar sein. Es stellt sich also die Frage, wie sich möglichst flexible, mobile Systeme mit einer möglichst präzisen, lokalen Datengewinnung gestalten lassen. Sind Sensoren unablässig oder lassen sie sich beispielsweise durch Scan-Geräte ersetzen? Was können Scanner zurzeit und was möglicherweise in der Zukunft? Und welche Geräte sind in welchem Rahmen einsetzbar?

Der erste Schritt, um vor dem Hintergrund dieser Fragestellungen den Projektrahmen zu finden, ist die Erstellung von Zielgruppen Personas. Auf deren Basis können dann die Anforderungen für das Visions-Produkt entstehen.

Erstellen von Zielgruppen Personas

Zielgruppe allgemein. Die allgemeine Zielgruppe umfasst Privatpersonen, die Interesse am Gärtnern haben. Dadurch, dass Gardening ein Trend-Thema ist, handelt es sich also um eine relativ große, breit gefasste Zielgruppe. Im Fokus soll jedoch das Gärtnern Lernen stehen. Dies soll durch Technik unterstützt werden, die beim User bereits vorhanden ist. Der Prozess des Lernens soll dabei durch angeleitete Tätigkeiten am realen Objekt geschehen. Es handelt sich also um kein virtuelles Lerntool.

¹⁴Link: <https://www.picturethisai.com/>

¹⁵Link: <https://plantnet.org/en/>

¹⁶Link: <https://www.plantsnap.com/>

¹⁷Link: <https://apps.apple.com/us/app/garden-flower-identification-plant-identifier-free/id1128290219>

¹⁸Link: <https://ipgarten.de/>

¹⁹Quelle: <https://computerwelt.at/news/iot-macht-aus-hochbeeten-smartbeeten/>

²⁰Link: <https://app.asana.com/0/1172859492234369/1172957097729748>

Beispiel-Personas. Die allgemeine Zielgruppe könnte Personen mit folgenden Steckbriefen enthalten:

Jonas:

- 24 Jahre alt
- Student
- interessiert an Nachhaltigkeit und grünen Themen
- in der Stadt aufgewachsen
- bisher keinen Bezug zum Gärtnern oder zu Pflanzen, aber Sehnsucht nach Grün und Gartenidylle
- hat also kaum Erfahrung damit, möchte es aber Lernen und Ausprobieren
- wenn es ihm gefällt, macht er vielleicht mehr in die Richtung
- ist technisch komplett ausgestattet mit diversen gängigen Geräten (mobil, aber auch stationäre, smarte Assistenten)

Martina:

- 26 Jahre alt
- hat bereits Erfahrung mit Pflanzen
- möchte mehr lernen und Garten-Profi werden

Mike:

- 34 Jahre alt
- begeisterter Gärtner mit sehr großem Garten und vielen Pflanzen
- kann sich gar nicht wirklich um alles kümmern und braucht Hilfe dabei

Der Fokus für das Visionsvideo soll vor allem auf Jonas festgelegt sein.

Features - Vision vs. grobe Richtung Prototyp

Aus der Zielgruppe ergibt sich die Schlussfolgerung ein möglichst Hardware-armes Produkt zu entwickeln, das universell für viele Nutzer einsetzbar ist. Das heißt auf technische Geräte zu setzen, die bereits in jedem Haushalt vorhanden sind. Zurzeit ist dies unbestritten das Smartphone, aber auch erweiternde SmartHome-Geräte halten immer mehr Einzug. Auch wenn sich User Interface und technische Ausstattung stetig wandeln, wird uns auch in Zukunft das Smartphone weiter begleiten. Auf diese Annahme setzt Grüni. Die folgende Grafik zeigt den Stand von definierten Soll-Anforderungen vom 29.04.2020 (vgl. mit späterer *Darstellung im Video* sowie *Umsetzung beim Prototypen*):

Zusätzlich zu der technischen **Ausstattung zur Analyse** gibt es ebenfalls Hardware zur **Versorgung** der Pflanzen. Dies soll mit der Anwendung Grüni zwar möglich sein, aber freiwillig, da weitere Geräteinstallationen dafür nötig sind. Der Lern- und Informations-Aspekt stehen im Vordergrund.

Für den Prototypen war jedoch bereits zu diesem Zeitpunkt klar, dass wir nicht auf externe Sensoren verzichten

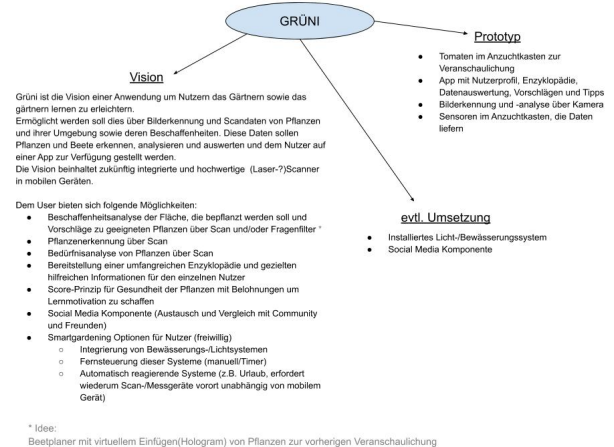


Figure 1: Grafik 'Vision vs. Prototyp'

werden können und auch der Wirkungsraum spezifisch eingegrenzt sein wird. Dazu allerdings erst mehr unter dem Punkt **Prototyp**.

Herausforderungen in der Projektfindungsphase

Eine große Herausforderung hat in der Masse an Recherche-Material bestanden, welches durchgelesen und aussortiert werden musste. Aus dieser Flut hat sich die genaue Definition, was denn nun genau mit der Grüni Anwendung erreicht werden soll, als schwierig herausgestellt. Ebenso ist die klare Abgrenzung zu den vielen anderen Projekten schwer gefallen.

2 DER PROTOTYP

Konzeption

Idee

Aus dem Visionsvideo soll nun ein Prototyp entwickelt werden, der einen ersten Schritt in Richtung Realisierung der Vision darstellen soll.

Selbstverständlich sind Hologramm-Technologien, wie sie im Video gezeigt werden, noch längst nicht umsetzbar. Allerdings gibt es heute schon Möglichkeiten virtuelle Objekte in die reale Welt zu setzen und zwar in Form von Augmented Reality.

Der Aspekt, dass die Anwendung auf so ziemlich jedem Gerät laufen soll, lässt sich realisieren: Der Prototyp wird eine Web-Anwendung. Dadurch geht allerdings auch ein Aspekt verloren: Man kann dem Nutzer keine Erinnerungen/Notifications schicken, wenn er sich nicht auf der Webseite befindet. Diese Einschränkung nimmt das Team für den Prototypen aber in Kauf.

Das Erkennen von Pflanzen und ihrem Zustand anhand von Scans bzw. Bildern ist zwar prinzipiell möglich,

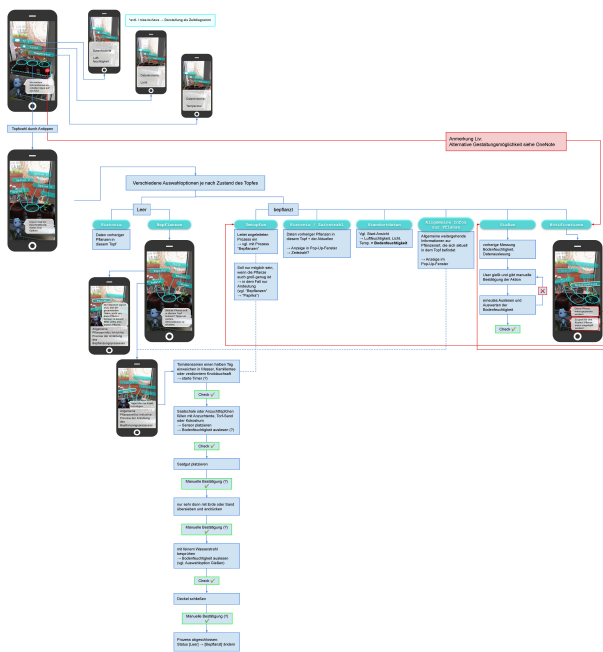


Figure 2: Grafik - Ablauf und Visualisierung

allerdings sind solche Analysesysteme sehr aufwendig und brauchen sehr viele Testdaten, um richtig funktionieren zu können. Das würde den Projektrahmen sprengen. Daher wird der Prototyp nur mit einem Anzuchtkasten umgesetzt, in den zusätzlich Sensoren eingebaut werden.

Zusammengefasst soll also eine AR-Anwendung im Webbrowser realisiert werden, die mit einem Anzuchtkasten funktioniert. In den Anzuchtkasten werden Sensoren eingebaut, die den Zustand im Kasten erfassen. Über mobile Endgeräte erfolgt nun die Überlagerung von virtuellen Objekten in die reale Welt, gestützt durch die Messwerte der Sensoren.

Design

Das daraus abgeleitete Design findet sich in der Grafik wieder. Alternativ befindet es sich ebenfalls im [Anhang](#).

Die Messdaten der Sensoren werden unterschieden in **übergreifende** und **spezifische** Messdaten. Luftfeuchtigkeit, Belichtung und Temperatur gelten für alle Pflanzen im Kasten gleich, darum werden sie auch in der Darstellung an den Kasten angehängt und direkt im Startmenü angezeigt. Lediglich die Bodenfeuchtigkeit gilt spezifisch für jeden Topf einzeln, der User muss diesen Topf also zunächst im Startmenü auswählen, um sich dessen gesamte Daten unter der Menüoption "Standortdaten" anzeigen zu lassen.

Außerdem unterschieden werden muss zwischen einem leeren und einem bepflanzt Topf. Je nachdem wie diese

Figure 3: Grafik der Architektur

Zustandseigenschaft des Topfes belegt ist, ergeben sich bei der Auswahl dessen nämlich auch weitere zu unterscheidende Menü- und Aktivitäts-Optionen. Diese lassen sich der Ablauf-Grafik entnehmen.

Übergreifende Messdaten werden über dem Kasten als türkise Buttons dargestellt. Durch Antippen wird zum jeweiligen Messwert ein Diagramm angezeigt, welches den Verlauf über die letzten Tage visualisiert.

Umfangreichere Inhalte wie Informationen zu einer Pflanze oder auch die Diagramme werden in einem weißen Popup dargestellt, welches sich auf den Bildschirm legt und die AR-Szene überdeckt. Das Popup ist am Bildschirm fest und nicht am Kasten. Er ist scrollbar und kann über ein X geschlossen werden.

Wichtige Notifications, welche in der Grafik rot dargestellte Symbole sind, erscheinen, falls ein nötiger Handlungsbedarf besteht (z.B. wenn die Erde der Pflanze zu trocken ist), am jeweiligen Topf. Durch Antippen sollen diese direkt zu einer entsprechenden Popup-Nachricht führen und zu der erforderlichen Aktion im Menü navigieren.

Popup-Nachrichten sind immer unten zu sehen in Form von Sprechblasen, die von PLANT-E "gesprochen" werden. Diese Popup-Nachrichten sollen auch die Anweisungen z.B. für das Eintopfen darstellen. Dies soll die Kommunikation von PLANT-E mit dem Nutzer visualisieren.

Architektur

Aus diesen Anforderungen ist eine Architektur entstanden, die aus drei Kernkomponenten besteht:

- dem Sensorenhandling über einen Raspberry Pi,
- der Datenverwaltung und Auslieferung der Webseite über einen Backend-Server und
- die eigentliche Webseite, die von den Clients aufgerufen wird.

Zur Übersicht ist ebenfalls ein Diagramm erstellt worden, welches die Komponenten und ihr Zusammenspiel verdeutlicht.

Die Sensordaten werden stetig gemessen und vom Raspberry Pi über UDP-Pakete an das Backend gesendet. UDP ist ein verbindungsloses Protokoll. Es werden lediglich Zieladresse und -port angegeben und schon wird das Datenpaket versendet. Es findet keine Empfangsbestätigung statt. Die Pakete sind immer kleine abgeschlossene Einheiten. Die Idee ist, dass die Sensordaten in einem langen konstanten Datenstrom ausgeliefert werden, egal ob jemand diese Daten entgegennimmt oder nicht. Die Vorstellung eines Radiosenders ist ein geeigneter Vergleich. Der Vorteil dieses

Vorgehens ist, dass stetig und sehr schnell Daten zur Verfügung stehen mit geringstem Overhead.

Der Backend-Server ist das Bindeglied für alle Komponenten. Er nimmt den UDP-Datenstrom des Raspberry Pis entgegen und verarbeitet diese Daten. Zum Einen werden sie für die Erfassung eines historischen Verlaufes gespeichert. Zum Anderen können die Daten vom Frontend nun abgerufen werden. Das Backend liefert immer die aktuellsten erhaltenen Daten aus. Bricht der Datenstrom vom Raspberry Pi ab, sind immer noch die zuletzt empfangenen Daten abrufbar.

Zusätzlich liefert das Backend auch die Frontend-Webseite mit allen zugehörigen Dateien aus. Darüber hinaus stellt das Backend eine REST-API zur Verfügung, die vom Frontend genutzt werden kann, um Daten abzurufen und zu senden. Die Speicherung von zusätzlichen Daten, z.B. ob ein Topf bepflanzt ist und wie lange schon, wird ebenfalls im Backend gespeichert und verwaltet. Das Backend liefert auch einen Pflanzenkatalog mit den unterstützten Pflanzenarten aus.

Das Frontend ist eine Webseite, die in jedem modernen Browser aufgerufen werden kann. Da es sich um eine AR-Anwendung handelt, benötigt der Client zwingend eine Kamera im Gerät. Die Anwendung ist nur in der Lage Informationen rund um den Kasten anzuzeigen. Zur Erkennung des Kastens ist dort ein Marker angebracht dazu – später mehr.

Umsetzung

Raspberry Pi

Der Raspberry Pi soll mittels Sensoren verschiedene Daten zu dem Anzuchtkasten messen und per UDP-Pakete an das Backend senden.

Datenformat. Die gesendeten UDP-Pakete sollen alle Messdaten im JSON-Format enthalten. Hier die Spezifikation der zu sendenden Daten:

```
{
  "timestamp": 1594639596, // unix timestamp
  "airHumidity": 42, // in Prozent, ganzzahlig
  "light": 123, // in lux, ganzzahlig
  "temperature": 24, // in Grad, ganzzahlig
  "pots": [ // die Töpfe von links nach rechts
    {
      "soilHumidity": 73, // in Prozent, ganzzahlig
      // "size": 8, // Höhe der Pflanze in cm
      // ist nicht umgesetzt worden
    }
  ],
}
```

Luftfeuchtigkeit, Helligkeit und Temperatur werden für den gesamten Kasten erfasst. Die Bodenfeuchtigkeit wird für jeden Topf einzeln gemessen und ausgeliefert. Da der

Gruppe nur ein Sensor für die Bodenfeuchtigkeit zur Verfügung gestellt worden ist, kann der Prototyp nur einen Topf bedienen.

Die Analyse der Wuchshöhe ist im Verlaufe des Projektes ausgeklammert worden, da keine geeignete Kamera zur Verfügung gestanden hat. Außerdem ist die Analyse von Kamerabildern ein recht aufwendiges Verfahren, was den Rahmen des Projektes gesprengt hätte.

Verwendete Sensoren. Zum Verarbeiten der Sensordaten wird ein Raspberry Pi 4 benutzt, da dieser der Projekt-Gruppe bereits zur Verfügung gestanden hat. Für die Sensoren ist daher die Auswahl etwas begrenzt gewesen.

- Lichtmessung: *Lichtsensor „BH1750“*
 - wandelt bereits auf der Platine die Analogsignale in Digitalsignale
- Messung von Temperatur und Luftfeuchtigkeit: *Sensor „DHT22“*
 - wandelt ebenso bereits die Signale, sodass diese ohne zusätzliche Bauteile vom Raspberry Pi gelesen werden können
- Messung der Bodenfeuchtigkeit: *„Capacitive Soil Moisture Sensor v2.0“*
 - wurde von Hardwareexperten der FH empfohlen
 - liefert unter den Feuchtigkeitssensoren wohl die präzisesten Daten
- Signalverarbeitung des Feuchtigkeitssensors: *Analog-Digital Wandler „MCP3008“*
 - wurde ebenso gewählt aufgrund der Empfehlung führender Hardwareexperten der FH

Um diese Sensoren auf dem Raspberry Pi nutzen zu können sind externe Bibliotheken nötig gewesen. Hierbei handelt es sich um: Adafruit_ADS1x15²¹, Adafruit_DHT²² und smbus²³.

Anmerkung zur Abgrenzung zwischen dem Gruppenprojekt "Grüni" und dem IT-Praxisprojekt von Dennis Krischal:

Die Projekte überschneiden sich in genau einem Punkt, da es sich angeboten hat für beide die gleiche Hardware zum Auslesen der Sensoren zu benutzen.

Schaltplan.

Skript. Die Anwendung auf dem Raspberry Pi ist als Python-Skript erstellt worden. Es liest die Sensordaten und versendet sie als UDP-Paket an den Server.

Es gibt einen Mock-Modus in dem das Skript keine Sensordaten, sondern Zufallswerte generiert und ausgibt. Das

²¹Link: https://github.com/adafruit/Adafruit_Python_ADS1x15

²²Link: https://github.com/adafruit/Adafruit_Python_DHT

²³Link: <http://wiki.erazor-zone.de/wiki:linux:python:smbus:doc>

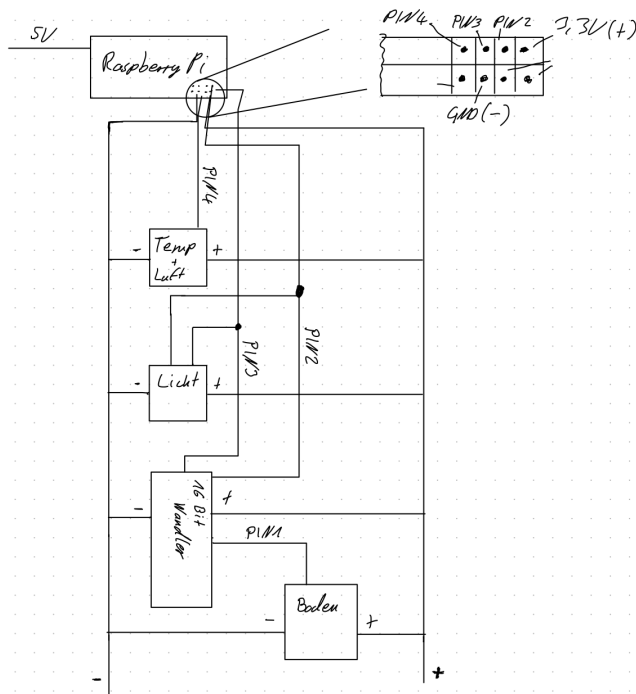


Figure 4: Schaltplan der Sensoren mit dem Raspberry Pi

Skript kann dadurch auch auf einem beliebigen Gerät gestartet werden, liefert dann aber natürlich nur die Zufallswerte.

Backend

Das Backend ist ein Node.js-Server, der in TypeScript implementiert worden ist. Er empfängt zum einen die UDP-Pakete vom Raspberry Pi über einen entsprechenden Socket. Zum Anderen stellt er einen Express.js Server zur Verfügung, der sowohl das Frontend inklusive aller benötigten Dateien ausliefert als auch eine REST-API für das Frontend zur Verfügung stellt.

Es speichert die Daten vom Raspberry Pi in regelmäßigen Abständen in einer Historie und speichert auch Daten, die vom Frontend kommen. Zum Beispiel in welchem Topf sich welche Pflanze befindet und wie lange die dort schon eingetopft ist usw.

Außerdem liegt ein Katalog mit den unterstützten Pflanzenarten im Backend vor. Dieser wird verwendet, um die Pflanzendaten anzureichern.

Die Daten werden in JSON-Dateien gespeichert, die beim Start des Servers geladen werden. Ändern sich die Daten, so werden die Änderungen auch in die JSON-Dateien geschrieben. Die Verwendung von einem Datenbanksystem ist für den Prototypen zu aufwendig gewesen.

Frontend

Das Frontend besteht aus einer Webseite, die von der Funktion her eine SPA (Single-Page-Application) ist. Zum Einsatz kommt das Webframework **Vue.js** sowie die Technologien **AFrame** und **AR.js**.

Vue.js²⁴ ist ein schlankes Frontend-Framework, welches mit recht simplen HTML-Attributen, einem Komponenten- und Vorlagensystem das Erstellen von interaktiven Webseiten vereinfacht. Es kann einfach auf die HTML-Struktur draufgesetzt werden und ist recht leichtgewichtig.

AFrame²⁵ ist ein Wrapper für Three.js, mittels dem 3D-Modelle in eine Szene gesetzt und gerendert werden können. AFrame stellt dabei HTML-Tags zur Verfügung, die 3D-Elemente repräsentieren und entsprechend gerendert werden. Der Mechanismus erfolgt so, dass ein `<a-scene>`-Tag im HTML-Markup gesetzt werden muss. Diese `<a-scene>` wird dann in ein HTML-Canvas-Element umgewandelt und die 3D-Elemente (`<a-entity>`) innerhalb der `<a-scene>` werden in die Szene gesetzt und auf den Canvas gemalt. Mit diesem System lassen sich VR-Anwendungen im Webbrowser darstellen.

AR.js²⁶ ist nun eine Erweiterung für AFrame. Der zugrundeliegende Mechanismus bleibt gleich, allerdings wird der Hintergrund des Canvas mit dem Bild der geräteinternen Kamera gefüllt. Zusätzlich kann ein sog. Marker definiert werden (siehe Bild). Der Marker wird vom Kamerabild erkannt und dient als Referenzpunkt. Die Position des Markers markiert den Ursprungspunkt der Szene (Punkt 0,0,0). Die Breite des Markers gibt die Breite von einer Einheit in AR.js an. Nun können 3D-Elemente relativ zum Marker platziert werden.

AR.js funktioniert so, dass die Elemente nur zu sehen sind, wenn auch der Marker zu sehen ist, zu dem die Elemente ja relativ liegen. Er dient als Referenz- und Ankerpunkt der Szene. Der Einfachheit halber ist der Standard-Marker von AR.js verwendet worden.

Ergebnisse und Herausforderungen

Die Umsetzung des Raspberry Pi, der Sensoren und des Backends hat sehr gut funktioniert. Die Daten werden richtig ausgeliefert, im Backend gespeichert und ausgegeben. Lediglich die Geschwindigkeit der Sensoren ist nicht optimal. Diese brauchen im Schnitt etwas unter einer Sekunde, um die Messdaten zu erheben und zu versenden. Dies ist für sehr interaktive Use-Cases wie zum Beispiel der Gießanzeige zwar unpraktisch, für den Prototypen ist es aber ausreichend. Hier können in Zukunft Optimierungen vorgenommen werden, etwa mit besserer Hardware.

²⁴Link: <https://vuejs.org>

²⁵Link: <https://aframe.io>

²⁶Link: <https://ar-js-org.github.io/AR.js-Docs/>



Figure 5: Bild des Hiro-Markers von AR.js

Bezüglich des Backends fehlen natürlich reale, über einen längeren Zeitraum erhobene Daten. Wären diese vorhanden, könnten sich noch neue Erkenntnisse gewinnen und zum Beispiel eine sehr umfangreiche Historie mit vielen Optionen und Ansichten erzeugen lassen. Für einen Prototypen sind die vorhandenen Daten aber ausreichend.

Sehr ernüchternd ist hingegen das Frontend ausgefallen. Die anfängliche Euphorie über die AR-Technik im Browser ist sehr schnell verfliegen, weil sich sehr bald gravierende Schwierigkeiten ergeben haben. Der Projektfortschritt ist dadurch so stark behindert worden, dass nur ein Bruchteil der geplanten Funktionalität überhaupt umgesetzt werden konnte.

Ein großes Problem, welches immernoch nicht abschließend gelöst werden konnte, ist das Anklicken von Elementen in AR.js. Es hat sehr lange gedauert, bis es überhaupt möglich gewesen ist, auf Klicks zu reagieren. Das hängt damit zusammen, dass Klicks in der Szene über einen Raycaster umgesetzt werden. Dieser Raycaster ist in der Standard-Konfiguration zu langsam. Da im AR-Modus die Szene ständig verschoben wird, aufgrund des wackelnden Markers durch die wackelige Hand des Nutzers, muss der Raycaster sehr schnell immer wieder nach Kollisionen checken, ansonsten werden niemals irgendwelche Klicks erkannt. Diese Erkenntnis zu gewinnen und das Problem zu lösen, hat alleine 2 Wochen in Anspruch genommen.

Damit nicht genug funktionieren Klicks im AR-Modus nur in der Mitte des Bildschirms einigermaßen zuverlässig.

Weiter am Rand funktionieren sie so gut wie gar nicht. Die genaue Ursache ist bis heute nicht geklärt worden. Durch das Wackeln werden zusätzlich manchmal Elemente angeklickt, die man gar nicht klicken wollte. Insgesamt ist das Interagieren mit der AR-Anwendung sehr mühselig und ist alles andere als intuitiv und ausgereift.

Auch die Erkennung des Markers in AR.js funktioniert nur unter guten Lichtverhältnissen und freier Sicht der Kamera auf den Marker. Wird der Marker verloren, so verschwinden einfach alle 3D-Elemente. Dies stellt keine schöne User Experience dar und ist gerade für die Entwicklung sehr anstrengend.

Deswegen ist während der Entwicklung der AR-Modus meistens ausgeschaltet gewesen. Die App wurde hauptsächlich im VR-Modus entwickelt, da hier nichts wackelt und die Entwickler besser testen können, was sie eigentlich tun. Daher funktioniert die Anwendung im VR-Modus recht gut, im AR-Modus eher weniger.

Auch im Design sind die Möglichkeiten von AFrame und AR.js sehr beschränkt. So ist es nicht möglich zum Beispiel die Ecken von einem Button abzurunden. Das komplette Design hat eigentlich auf "Kapseln" basiert. Da sich dies nicht realisieren lassen hat, sind überall langgezogene Kreise und Kugeln zum Einsatz gekommen. Das Erstellen oder Verändern von eigenen Geometrien und Meshes ist schwerlich möglich, da AFrame dafür keinen Editor oder Oberfläche bietet, wie man es z.B. von Unity kennt.

Das gravierendste Problem ist das nicht vorhandene Logging von Fehlern in AFrame und AR.js. Fehler in der Syntax werden nicht angezeigt und auch beim Auftreten von Fehlern während des Aufrufens der Seite, gibt es so gut wie keine Logs oder Hinweise. Die Entwickler tappen die ganze Zeit im Dunkeln und können sich nur durch kleine Änderungen und stetiges Ausprobieren im Browser langsam voranbewegen. Gibt die Dokumentation von AFrame wenigstens noch einen guten Überblick, so ist sie für AR.js viel zu lückenhaft, um damit ordentlich arbeiten zu können. Insgesamt ist die Entwicklung mit diesen Technologien sehr zäh gewesen und hat alles andere als zufriedenstellende Ergebnisse produziert.

Das Projekt ist auch durch die Vielzahl an unterschiedlichen Komponenten sehr komplex gewesen. Dieses Zusammenspiel hat sich aber durch Mocking und Fake-Daten recht elegant lösen lassen. Trotzdem hat das beschwerliche Arbeiten im Frontend dazu geführt, dass viele Funktionalitäten zwar theoretisch über Raspberry Pi und Backend zur Verfügung stehen, diese aber im Frontend nicht umgesetzt werden konnten. Tatsächlich sind die Sensoren und das Backend dem Frontend weit voraus. Dies ist für das Projektteam besonders frustrierend, da ja theoretisch mehr funktioniert, aber das Frontend nicht in der Lage ist, das darzustellen.

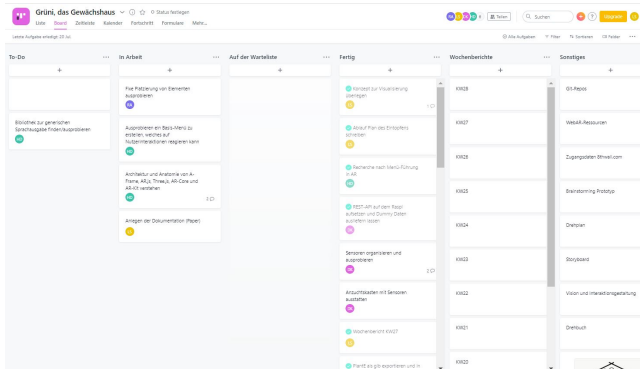


Figure 6: Screenshot von dem Organisationstool Asana

3 FAZIT

Das Projekt Grüni hat das Team auf sehr konstante Art und Weise durch dieses Semester begleitet. Die Projektidee ist schnell gefunden gewesen. Die Begeisterung des gesamten Teams für den Themenbereich eines Gardening-Projekts ist sehr förderlich gewesen, um stetig daran weiter arbeiten zu können.

Auch Organisation, Absprache und Aufgabenverteilung haben sehr gut funktioniert. Dies ist zum großen Teil den wöchentlich stattfindenden Modulveranstaltungen zu verdanken und möglicherweise auch dem Rahmen des (Corona geschuldeten) Online-Semesters. Die durchgeführten Webex-Meetings sind anschließend immer direkt intern weitergeführt worden und sind stets erstaunlich produktiv gewesen. Über Google Drive, Google Documents und ähnliche Tools ist gemeinsames Brainstorming, Festhalten von Ideen, Arbeiten an Dokumenten und Anforderungen konzentriert umsetzbar gewesen.

Als Projektmanagement-Tool ist Asana verwendet worden: Unter dem Link <https://app.asana.com/0/1172859492234369/board> lässt sich die Projektentwicklung nochmals im Detail nachvollziehen. Dort finden sich unter anderem sämtliche, geführten Protokolle unter dem Punkt Wochenberichte wieder, inklusive der Aufgabenverteilung im Team. Erstellte, ausgetauschte Dokumente und Dateien sind in den jeweiligen Aufgaben verlinkt und finden sich gesammelt im Google Drive Projekt-Ordner unter dem Link <https://drive.google.com/drive/folders/10xpkQQMSEUjZaZLw7HL2MjYE71AiZr6m?usp=sharing> wieder.

Der Arbeitsaufwand für ein 8CP-Modul ist selbstverständlich nicht in ein paar Tagen zu bewältigen gewesen. Das Projekt hat sehr viele verschiedene Aufgaben und Disziplin umfasst:

- die Entwicklung einer Produktidee,
- die Beschäftigung mit grafischen Nutzeroberflächen,

- Video-Dreh und -Bearbeitung,
- Animationen in Blender,
- das zusammenfügen von Real- und Animationsfilmteilen,
- Beschäftigung mit neuen, ungewöhnlichen Technologien (VR und AR im Webbrowser),
- das Entwickeln und Programmieren einer Anwendung mit vielen Teilaspekten (hardwarenah, server- und clientseitig)
- und zu guter Letzt LaTeX für das Erstellen der Dokumentation.

Vorteilhaft ist gewesen, dass viele Teile der Aufgaben nicht gänzlich neu gewesen sind, sondern einzeln bereits in vergangenen Modulen behandelt worden ist und das Team so aus bereits gesammelten Erfahrungen geschöpfen hat. Die Kombination aus so vielen Aufgaben stellt allerdings einen erhöhten Organisationsanspruch und auch klassische Zeitfresser, wie z.B. Überlastungen beim Rendern, haben sich nicht gänzlich vermeiden.

Dennoch ist eine schöne Vision für die Zukunft entstanden, die in einem spannenden Film visualisiert worden ist. Wer weiß, vielleicht hat schon bald jeder seinen kleinen PLANT-E zuhause und schöne grüne Beete, Töpfe und Balkone.

Leider ist das Ergebnis im Prototypen für das Team nicht sehr befriedigend ausgefallen, dadurch sind aber auch die Grenzen der aktuellen Technologien deutlich geworden. Gerade für den Blick in die Zukunft ist das Scheitern unabdingbar und die Grundlage für Fortschritt und Entwicklung. Außerdem handelt es sich natürlich auch nur um einen Prototyp, der zeigen sollen, was heutzutage schon möglich bzw. noch nicht möglich ist.

4 ANHANG