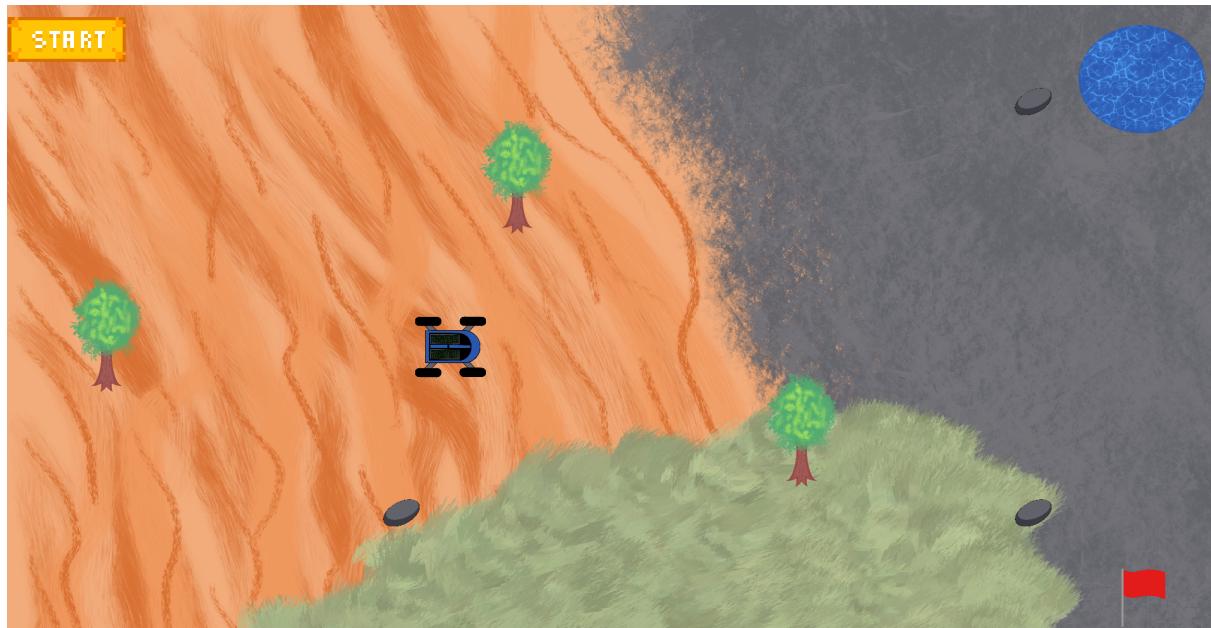


Projektbericht Space Rover

*Projektarbeit im Rahmen des 2. Semester für den Studiengang Informatiker HF
an der Teko Luzern.*



Celin Villars und Sarah Birrer

L-TIN-22-Di-a

11.09.2022

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

Management Summary

Als Vorbereitung für das neue Rover Game wurden mehrere Prototypen mit den Grundmechaniken erstellt. Darin enthalten sind Fortbewegung auf einer Karte und Kollision mit Hindernissen. Die Idee des Prototyps war, eine Basis zu schaffen, auf der der Rest des Spiels aufgebaut werden kann. Der Prototyp konnte wie geplant umgesetzt werden. Dabei wurde etwas zu viel Zeit darauf verwendet zu bestimmen, welches GUI Modul verwendet werden soll. Die Planung nach dem HERMES Modell lieferte eine wichtige Grundlage, jedoch wurde nicht exakt nach dem Modell gearbeitet, was von den Mitarbeitern als praktischer empfunden wurde. Nach dem aktuellen Stand sind die erwähnten Grundmechaniken erfolgreich umgesetzt worden. Damit ist die Basis für das Spiel gelegt und kann in einem nächsten Schritt erweitert werden.

Bei dem Projekt sind wir zu den Erkenntnissen gekommen, dass die Planung gezielter gemacht werden muss. Weniger Zeit für Entscheidungen und Evaluationen und mehr für Entwicklung einsetzen. Dokumentation laufend führen, nach Initialisierungsphase direkt Dokument erstellen. Technische Dokumentation einfach als PDF exportierbar machen.

Die App kann noch weiter ausgebaut werden, z.B. können weitere Levels mit anderen Karten und Hindernissen erstellt werden. Gegner, welche den Rover beschädigen können. Optionen im Menü, mit diversen Einstellungen wie Steuerung des Rovers, Schwierigkeit. Feld für Benutzernamen im Menü, um eine Rangliste am Ende des Spiels anzuzeigen mit Zeitmessung oder Punktestand. Weiter müssen noch Unit Tests für die Erhöhung der Testabdeckung erstellt werden.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

Inhaltsverzeichnis

Management Summary	1
1. Einleitung	4
2. Anforderungen	5
2.1. Ziele	5
2.2. Technische Dokumentation	5
2.3. Projektmanagement Dokumentation	5
2.4. Präsentation	5
2.5. Evaluation der GUI Library	6
2.6. Statische Hindernisse auf der Karte	6
2.7. Steuerung des Rovers	6
2.8. Architektur	6
3. Risikoanalyse	7
3.1. Bewertung	7
3.2. Massnahmen	8
4. Verwendete Technologien / Tools	9
4.1. Programmiersprache	9
4.2. Klassendiagramme UML (Unified Modeling Language)	9
4.2.1. Pygame	9
4.3. VSC (Version Control System)	9
4.3.1. Github	10
4.4. IDE (Integrated Development Environment)	10
4.5. Planung	10
4.6. Dokumentation	11
4.6.1. Projektmanagement Dokumentation	11
4.6.2. Technische Dokumentation	11
4.7. Kreativitäts Tool	11
4.8. Kommunikation	12
5. Vorgehen	13
5.1. Initialisierung	13
5.2. Planung	13
5.2.1. Anforderungen (MindMap)	13
5.2.2. Terminplan	15
5.3. Konzept	16
5.3.1. Nutzwertanalyse GUI Komponente	19
5.4. Realisierung	20
5.4.1. Technische Dokumentation	24
5.5. Einführung	25

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

5.5.1. Testbericht	25
1. Testfall: Spiel kann geladen werden	25
Ergebnis	25
2. Testfall: Rover lässt sich bewegen	25
Ergebnis	25
3. Testfall: Rover kann in keine Hindernisse fahren	25
Ergebnis	25
4. Testfall: Rover erreicht das Ziel	26
Ergebnis	26
6. Abschlussbericht	27
6.1. Zielerreichung	27
6.2. Verbesserungsvorschläge	27
6.2.1. Projektmanagement Verbesserungen	27
6.2.2. Software Verbesserungen	28
6.3. Persönliche Reflexion	28
6.3.1. Celin Villars	28
Lessons Learned	28
6.3.2. Sarah Birrer	29
Lessons Learned	29
7. Quellenverzeichnis	30
7.1. Abbildungen	31
8. Anhang	32
8.1. GIT-Repository	32
8.2. Terminplan / Gantt	32
8.3. Nutzwertanalyse GUI Komponente	32
8.4. Einladung Projekt Meeting	32
8.5. Projekt Meeting Protokoll	32
8.6. Technische Dokumentation	32

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

1. Einleitung

Für unsere Projektarbeit an der Teko haben wir eine Codebasis inkl. eines laufenden Prototyps für ein Space Rover Spiel erstellt. Die Idee war, selbstständig Basiselemente eines Spiels zu programmieren, dadurch Python besser kennenzulernen und verschiedene Objekte miteinander interagieren zu lassen. Im Rahmen der Projektarbeit wurde eine passende GUI Library ausgewählt, damit für zukünftige Erweiterungen eine saubere Basis vorhanden ist. Uns war es wichtig, die Grundsteine des Spiels abzukapseln und in Module zu unterteilen, damit der Code übersichtlich und leserlich ist.

Ein wichtiges Kriterium für uns war es, dass wir alles Open Source verwenden und auch an die Open Source Community weitergeben. Deshalb wurde unser Projekt komplett in Github (9) erstellt und veröffentlicht mit einer technischen Dokumentation.

Die Schritte und Methoden, die wir angewendet haben, sind in den nachfolgenden Themen erläutert.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

2. Anforderungen

2.1. Ziele

- Ein benutzerfreundlicher Prototyp mit dem Thema Planetenerkundung wird erstellt, für den keine Anleitung benötigt wird.
- Der Rover kann sich auf der Karte bewegen.
- Es soll eine modulare Codebasis erstellt werden, welche mit weiteren Modulen ergänzt werden kann.

2.2. Technische Dokumentation

- Die technische Dokumentation soll im Repository des Projektes abgelegt werden, damit weitere Entwickler direkt alle nötigen Informationen zur Implementierung im Zugriff haben.
- Sie enthält detaillierte Informationen zur Architektur, Funktionalität und Implementierung der Software.
- Die technische Dokumentation soll ausschließlich in Englisch geschrieben werden, da das Projekt auf Github (9) veröffentlicht wird und somit Open Source ist.

2.3. Projektmanagement Dokumentation

- Es wird eine eigene Dokumentation für das Projekt selber erstellt, in welcher Stolpersteine, Methoden etc. beschrieben werden.
- Diese Dokumentation soll losgelöst von der technischen Dokumentation sein.
- Die Dokumentation soll als PDF gespeichert werden können.

2.4. Präsentation

- Als Abschluss der Projektarbeit soll anhand der Vorgaben der Teko eine Präsentation mit Demo gehalten werden.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

2.5. Evaluation der GUI Library

- Es wird eine NWA von mind. zwei Open Source GUI Libraries erstellt.
- Kriterien für die NWA sind Funktionalität, Anwendung und Kompatibilität sein.

2.6. Statische Hindernisse auf der Karte

- Beim Starten des Spiels sollte eine statische Karte mit Hindernissen geladen werden.
- Die Hindernisse dürfen vom Rover nicht befahren werden .

2.7. Steuerung des Rovers

- Der Rover soll mit gewohnter Steuerung WASD bewegt werden können.
- Der Rover soll nicht vorwärts, rückwärts und nach links und rechts bewegen können auf der Karte.

2.8. Architektur

- Es soll eine möglichst saubere Codebasis geschaffen werden, größere externe Libraries sollen mittels Wrapper abgekapselt werden. Dies soll einen einfachen Austausch ermöglichen.
- Der Wrapper ermöglicht eine klare Übersicht der Nutzung der Library

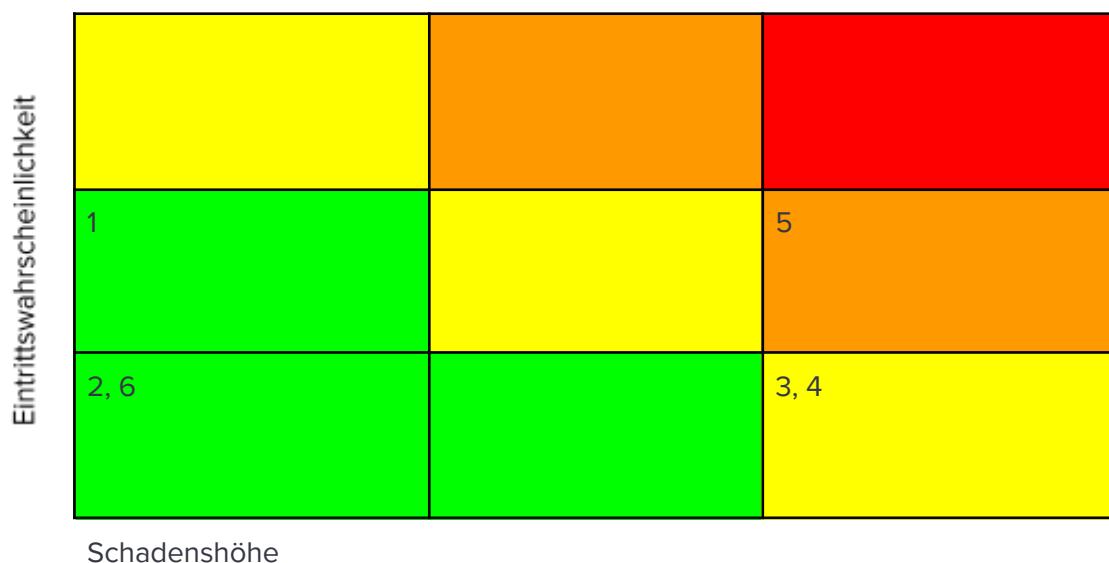
Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

3. Risikoanalyse

1. Projektleiter fällt aus.
2. Persönliche Geräte werden defekt oder unbrauchbar.
3. Codebasis wird gelöscht oder geht verloren.
4. Durch schlechtes Zeitmanagement kann das Projekt nicht in der angegebenen Zeit umgesetzt werden.
5. Durch zu wenig Python Know-How kann das Projekt nicht in der angegebenen Zeit umgesetzt werden.
6. Github (9) ist down.

3.1. Bewertung



Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

3.2. Massnahmen

1. Projektleiter Stv. übernimmt.
2. Tägliches Backup des Codes auf Github (9) wie auch sonstige Daten im Cloud Storage.
3. Tägliches Backup des Codes.
4. Gute Zeitplanung, genug Reserven einbauen.
5. Wöchentlicher Austausch der Probleme und Transparenz bzgl. des Fortschrittes.
6. Code ist auch auf den Geräten lokal gespeichert.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

4. Verwendete Technologien / Tools

4.1. Programmiersprache

Als Programmiersprache haben wir Python verwendet. Python ist eine Open Source Programmiersprache, die wir für die Realisierung des Projektes verwendet haben. Die Einarbeitung in die Sprache ist einfach.



4.2. Klassendiagramme UML (Unified Modeling Language)

Mit Hilfe von Pylance konnten wir direkt Klassendiagramme von unserem Source Code generieren lassen. Die Diagramme konnten wir in ein PlantUML (7) Diagram File exportieren und erweitern.



4.2.1. Pygame

Im Kapitel [Nutzwertanalyse GUI Komponente](#) wird das genaue Verfahren für die Evaluation einer GUI Komponente beschrieben. Anhand der Nutzwertanalyse wurde entschieden, dass diese für unser Projekt am geeignetsten ist.



4.3. VSC (Version Control System)

Für die Sourcecode Verwaltung haben wir GIT (10) verwendet, da dieses Tool einen großen Bekanntheitsgrad hat.



Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

4.3.1. Github

Github (9) ist ein führender Anbieter für Open Source Code Projekte. Github (9) bietet viele Möglichkeiten, die Zusammenarbeit zwischen mehreren Entwicklern zu vereinfachen. Wir haben uns bewusst dafür entschieden, dass wir auch einen Build für die Software einrichten können und somit eine Absicherung für einen immer lauffähigen Main Branch haben. Zusätzlich kann auch mittels anderen Tools Tests und Test Coverage gemessen werden.



Github Logo

4.4. IDE (Integrated Development Environment)

Als IDE haben wir Visual Studio Code (11) standardisiert, da es Open Source ist und es sehr einfach in der Handhabung ist. Zusätzlich hat unser Team bereits Erfahrungen damit gesammelt.



VS Code Logo

4.5. Planung

Für unsere Planung haben wir Monday (3) verwendet. Dadurch könnten wir mit wenig Aufwand unsere Tasks erfassen und ein Gantt generieren lassen. monday (3) bietet eine kostenlose Version für maximal zwei Personen mit eingeschränkten Funktionen. Es hat uns die Zusammenarbeit wie auch die Planung erleichtert.

A screenshot of the Monday work management software interface. The top navigation bar shows 'monday work management' and a search bar. On the left, there's a sidebar with 'Startseite', 'Meine Arbeit', and a search field containing 'space rover'. The main area displays two Gantt charts. The first chart, titled 'Initialisierung', has tasks like 'Projektantrag' and 'Projektkonzept + Protokoll' both marked as 'Done'. The second chart, titled 'Planung', has tasks like 'Anforderungen definieren', 'Planung der Zusammenarbeit', and 'Gantt einrichten', also marked as 'Done'. Both charts show a timeline from April 25 to June 6. A legend indicates that green means 'Done' and red means 'In Progress'. The interface is dark-themed.

Terminplan in monday

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

4.6. Dokumentation

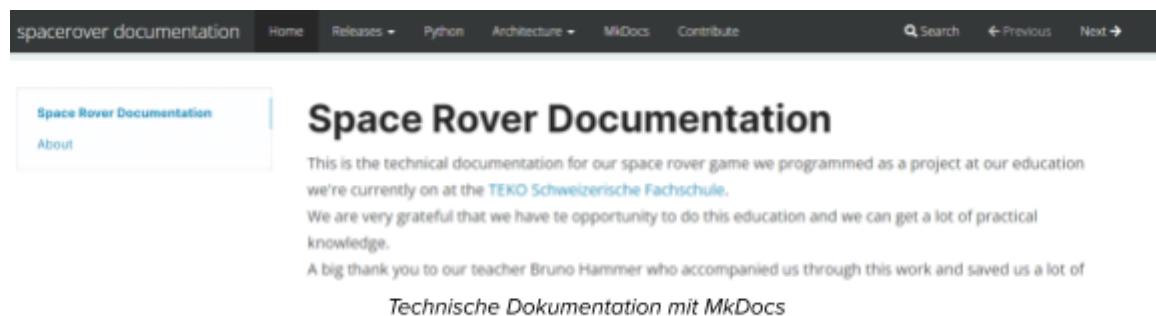
4.6.1. Projektmanagement Dokumentation

Für den Projektbericht haben wir Google Docs (5) verwendet. Da auch dieses Tool open source ist und einfach zu handhaben ist. Der Hauptgrund war aber, dass wir gleichzeitig ein Dokument bearbeiten können.



4.6.2. Technische Dokumentation

Für die technische Dokumentation haben wir in unser Repository MkDocs eingebunden. Dadurch können wir sehr einfach unsere Dokumentation erweitern und auf Github (9) Pages publizieren, was es ermöglicht, sie als Website darzustellen.



A screenshot of the "Space Rover Documentation" website. The header includes navigation links for "Home", "Releases", "Python", "Architecture", "MkDocs", "Contribute", "Search", and "Previous/Next". The main content area features a large title "Space Rover Documentation" and a paragraph of text explaining the purpose of the documentation, mentioning TEKO Schweizerische Fachschule and Bruno Hammer. Below this is a link to "Technische Dokumentation mit MkDocs".

Als Dokumentation Tool für die Projektmanagement Dokumentation haben wir Google Docs (5) verwendet. Google Docs (5) ist die Zusammenarbeit einfach und es ist ebenfalls Open Source.

4.7. Kreativitäts Tool

In der Startphase brauchten wir ein Tool, um ein Mindmap zu erstellen, damit wir unserer Kreativität freien Lauf lassen konnten. Dafür haben wir Miro (4) verwendet.



Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

4.8. Kommunikation

Als Kommunikationsmittel haben wir Discord, Whatsapp und Microsoft Teams verwendet.
Microsoft Teams haben wir mehrheitlich während den Unterrichtszeiten verwendet.



Discord Logo



Whatsapp Logo



MS Teams Logo

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

5. Vorgehen

5.1. Initialisierung

In der Initialisierungsphase wurden der Projektantrag und das Sitzungsprotokoll erstellt. In dieser Phase haben wir ebenfalls den ersten Termin für die Zusammenarbeit gesetzt.

Nach dieser Phase haben wir unseren ersten Meilenstein erreicht.

5.2. Planung

In der Planungsphase wurde die Anforderung an das Projekt definiert. Es wurden die Termine gesetzt und wir haben Tasks erstellt mit den Features, die wir im Projekt haben möchten. Die Definition der Zusammenarbeit haben wir ebenfalls in dieser Phase erledigt.

In dieser Phase haben wir diese Punkte definiert:

- **Lauffähigkeit:** Unser Main Branch muss immer lauffähig sein.
- **Testing:** Testing muss immer während der Entwicklung geschehen, automatisiert oder manuell
- **Stabilität:** Es wird kein Code gepusht, welcher Fehler wirft
- **Modular:** Features werden in eigenen Klassen implementiert

Nach dieser Phase haben wir unseren zweiten Meilenstein erreicht.

5.2.1. Anforderungen (MindMap)

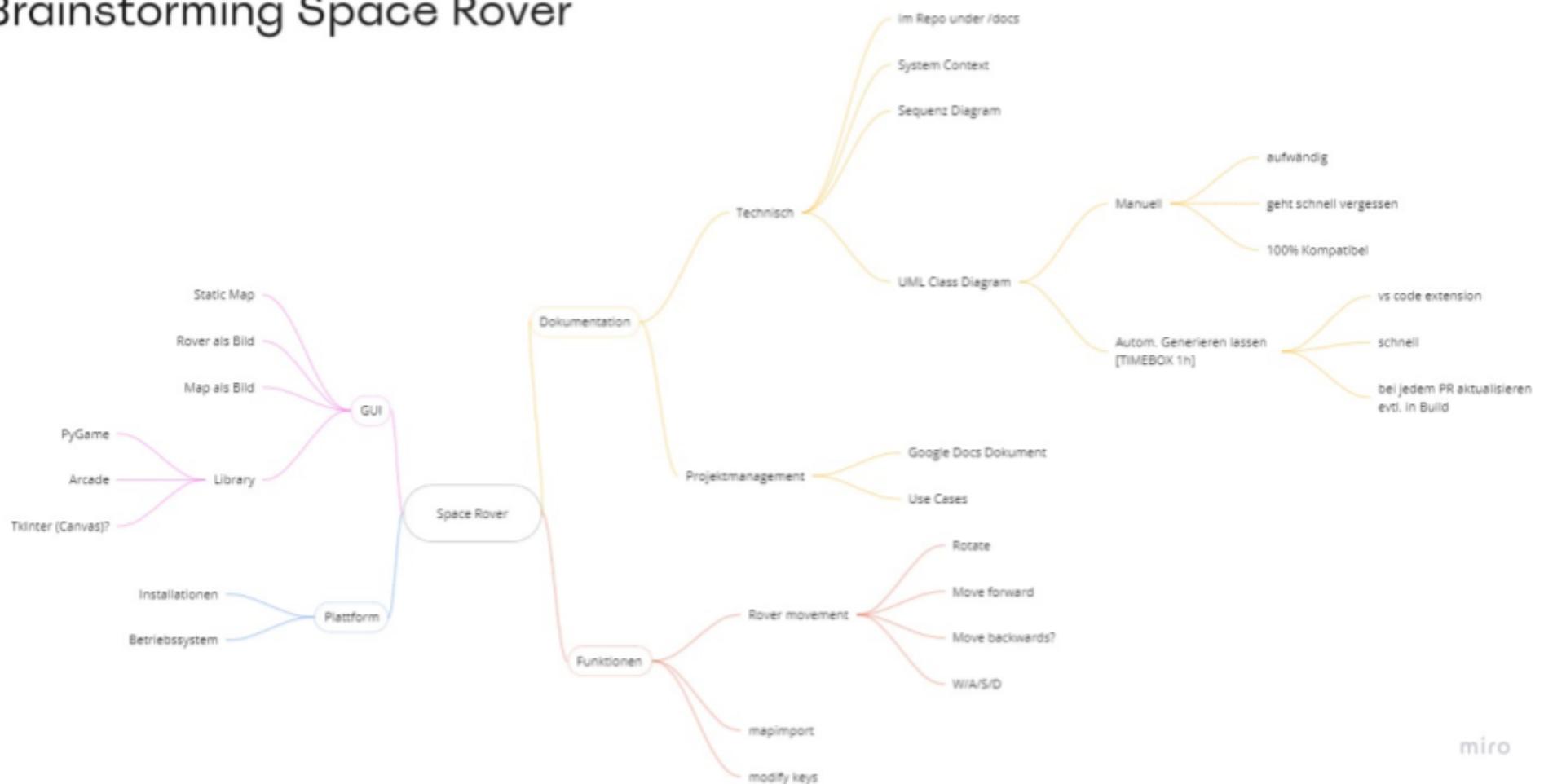
Anfangs hatten wir einige Ideen, die wir umsetzen wollten. Leider haben wir nach kurzer Zeit festgestellt, dass wir nur einen Bruchteil davon wirklich umsetzen können. Wir haben uns auf das Ziel, eine saubere Codebasis bereitzustellen, fokussiert. Uns war es wichtig, dass wir Funktionen implementieren, die beim Testen erfolgreich sind. Einen immer lauffähigen Main Branch, welcher möglichst wenige Bugs beinhaltet. Um unsere Kreativität zu fördern, haben wir ein MindMap erstellt, um herauszufinden, welche Funktionen uns wichtig sind und wie wir diese unterteilen.

Anschließend konnten wir priorisieren, welche Features wir sicher im ersten Prototyp umgesetzt haben möchten.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

Brainstorming Space Rover



miro

Mindmap für Anforderungsdefinition

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

5.2.2. Terminplan

Die Terminplanung war einer der wichtigsten Schritte des Projektes. In diesem Schritt haben wir die Meilensteine, Features und die Abhängigkeiten definiert. Zuerst haben wir die Punkte vom Mindmap priorisiert und dafür definiert, was wir alles brauchen, um die definierten Punkte zu erfüllen.

Die HERMES Projektmanagement Methode hat uns dafür strukturierte Grundlagen gegeben. Jede Phase Initialisierung, Planung, Konzept, Initialisierung und Einführung war für uns ein Meilenstein. Innerhalb dieser Phasen haben wir die Tasks nach Priorität geordnet, um den Überblick zu behalten.

In diesem Schritt ist es wichtig, fixe Termine zu setzen, bis wann wir welche Arbeiten erledigen müssen. Deshalb haben wir direkt die Termine gesetzt und jeweils ein Meeting eingeplant, bei welchem wir die erledigten Arbeiten besprechen und das weitere Vorgehen besprechen.

Der komplette Terminplan wie auch das Gantt ist im Anhang des Berichts zu finden.

Initialisierung							
Aufgabe	Person	Status	SOLL-Zeit (h)	Datum	Abhängigkeit	Zeitraum	+
> Projektantrag 1	CB	Done	2	Mai 2	-	Apr. 25 - Mai 2	
> Projektsitzung + Protokoll 1	CB	Done	2	Juni 6	Projektantrag	Juni 6	
+ Aufgabe hinzufügen							

Planung							
Aufgabe	Person	Status	SOLL-Zeit (h)	Datum	Abhängigkeit	Zeitraum	+
> Anforderungen definieren 1	CB	Done	2	Juli 9	Projektantrag	Juli 1 - 9	
> Planung der Zusammenarbeit 1	CB	Done	1	Juli 16	Projektantrag	Juli 10 - 16	
> Gantt einrichten 1	CB	Done	5	Juli 17	Planung der Zus...	Juli 10 - 16	
> Gantt befüllen 1	CB	Done	5	Juli 18	Anforderungen ...	Juli 10 - 16	
+ Aufgabe hinzufügen							

Konzept							
Aufgabe	Person	Status	SOLL-Zeit (h)	Datum	Abhängigkeit	Zeitraum	+
> 1. Prototyp Movement 1	CB	Done	4	Juli 17	Projektantrag	Juli 1 - Juli 30	
> Repository Struktur 1	CB	Done	5	Juli 17	Projektantrag	Juli 12 - Juli 16	
> Evaluation GUI Komponente 1	CB	Done	25	Juli 23	Gantt befüllen	Juli 17 - 23	
> Technische Dokumentation vorber.. 1	CB	Done	4	Juli 30	Gantt befüllen	Juli 24 - 30	
+ Aufgabe hinzufügen							

Terminplan in monday

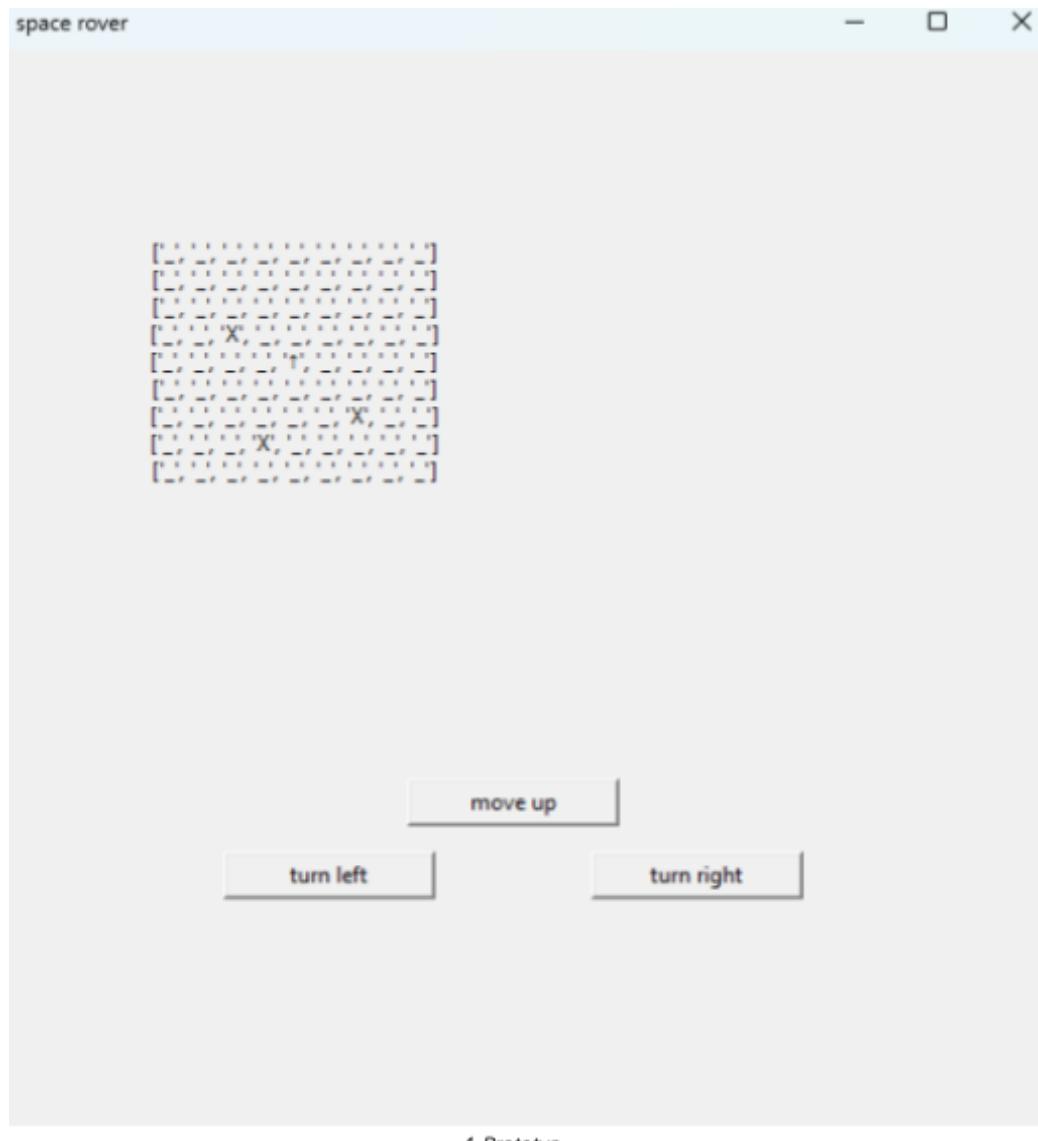
Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

5.3. Konzept

In der Konzept Phase haben wir das Repository und Dokumentation erstellt. Der erste Prototyp mit einem minimalistischen GUI wurde erstellt, welcher sich drehen und bewegen konnte. Der Rover wurde als Pfeil \uparrow , \rightarrow , \downarrow , \leftarrow dargestellt und die Hindernisse als X.

Dieser Prototyp hat es uns ermöglicht, bereits die ersten Erfahrungen mit Python zu sammeln und das Handling der Sprache kennenzulernen. Das hat uns die erste Hürde, den Code zu erstellen, sehr minimiert, wir hatten bereits Berührungspunkte in einem klein gestrickten Rahmen und konnten dadurch besser abschätzen, was im Rahmen der Möglichkeiten liegt.



1. Prototyp

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

Wir haben in dieser Phase parallel den 1. Prototyp mit der uns zugewiesenen GUI Komponente erweitert.

Ein weiterer Bestandteil dieser Phase ist auch die Entwicklungsumgebung. Wir haben unseren Main Branch geschützt und mit Prüfungen versehen, damit wir eine Sicherheit haben, dass unser Code weniger Fehler beinhaltet. Wir konnten somit Pull Requests erstellen, beim erstellen wurden Github (9) Actions ausgeführt, welche unseren Code nach Syntaxfehlern durchsuchten. Grundsätzlich war die Idee, dass jeweils der andere vom Team den Pull Request reviewed und approved.

Ebenfalls haben wir Deployment Actions erstellt, welche unsere technische Dokumentation automatisch bei einem Merge im Main Branch veröffentlicht.

scaled finish text #15

Open atomic-light wants to merge 1 commit into **main** from **feature/atomic-light/FixErrors**

Conversation 0 Commits 1 Checks 0 Files changed 1

atomic-light commented now

No description provided.

scaled finish text 2289783

Add more commits by pushing to the **feature/atomic-light/FixErrors** branch on **teko-L-TIN-22-Di-a/rover**.

This branch has not been deployed

No deployments

Review required

At least 1 approving review is required by reviewers with write access. Learn more about pull request reviews.

Some checks haven't completed yet

2 queued and 1 in progress checks

CI / build (macos-latest) (pull_request) Queued — Waiting to run this check... Details

CI / build (ubuntu-latest) (pull_request) In progress — This check has started... Details

CI / build (windows-latest) (pull_request) Queued — Waiting to run this check... Details

Pull Request mit Checks auf Github

Celin Villars
 L-TIN-22-Di-a
 Im Grund 2
 8910 Affoltern am Albis
 079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
 L-TIN-22-Di-a
 Brünigstrasse 34
 6055 Alpnach Dorf
 079 607 67 11
sarah.birrer@edu.teko.ch

Nach dieser Phase haben wir unseren dritten Meilenstein erreicht und können ohne grosse Aufwände darauf los entwickeln. Die Grundsteine für unser Projekt sind nun gesetzt.

Nachfolgendes Klassendiagramm veranschaulicht die Code-Organisation des 1. Prototyps.



5.3.1. Nutzwertanalyse GUI Komponente

Die Vielfalt an Libraries, die wir für das Projekt verwenden können, ist gross. Nach einer gründlichen Recherche haben wir unsere drei Favoriten fürs Rennen vorbereitet.

Aus diesen Gründen haben wir uns für die Komponente PyGame (8) entschieden:

- Bietet mehr Funktionalität
- Einfache Anwendung und Einstieg durch Tutorials
- Vielfältige Dokumentation
- Regelmässige Updates

Das Modul Tkinter (2) bietet sehr wenig Funktionalität und hat uns einen grossen Mehraufwand generiert.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

Da das Modul Arcade (1) seit Dezember 2022 nicht mehr aktualisiert wurde, haben wir keine weiteren Aufwände fürs Prototyping betrieben.

Die vollständige Nutzwertanalyse ist im Anhang unter dem Kapitel [Nutzwertanalyse GUI Komponente](#) zu finden.

5.4. Realisierung

In der Realisierungsphase haben wir die ganze Logik für das Rover Movement implementiert. Hindernisse wurden eingefügt und es wurde ein komplettes Refactoring des Codes gemacht. Der 1. Der Prototyp wurde somit komplett überarbeitet und wir haben uns für unsere Naming Conventions entschieden und unser Repository stetig danach angepasst. Wir haben ein Menü implementiert mit einem Start- und Quit Button.

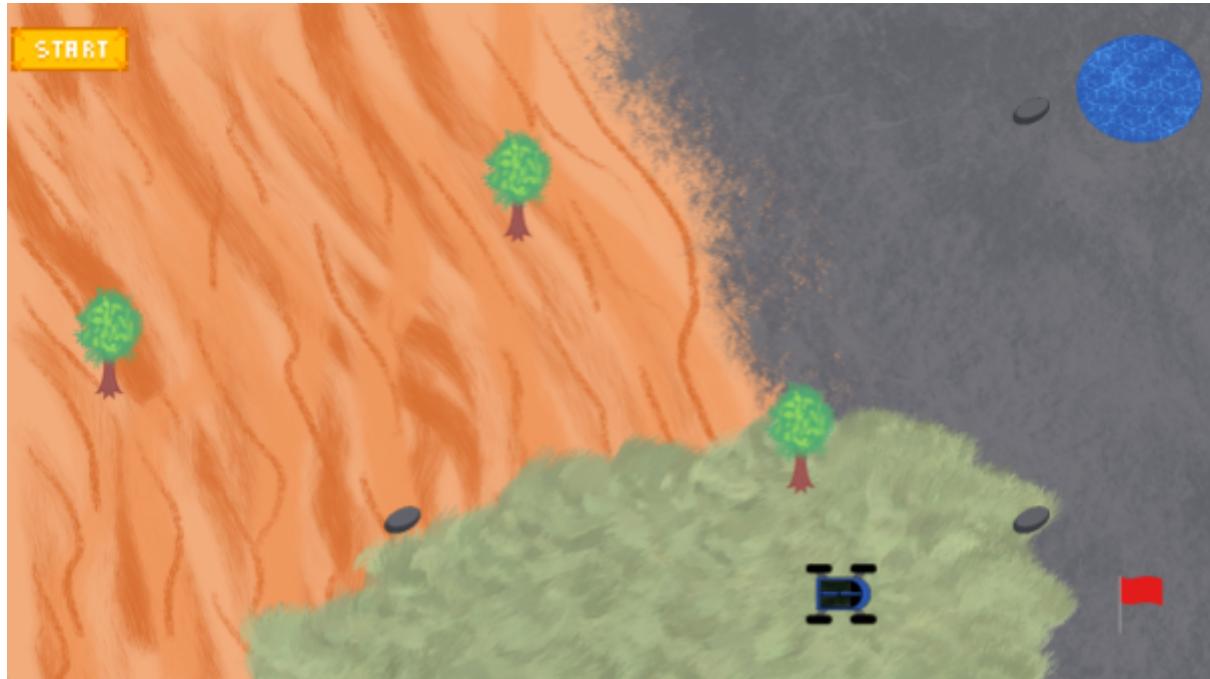


Startmenu

Nach sämtlichen Entwicklungen und Refactoring ist somit der 2. Prototyp erstellt, welcher das erste Level implementiert hat. Die Fehler aus dem vorherigen Prototyp wurden behoben. Der Code wurde komplett überarbeitet und weiter abstrahiert.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch



Level 1

Die Hindernisse werden in einer Liste gespeichert, vorgängig geladen und beim Starten des 1. Levels auf dem Screen angezeigt.

Beim Erreichen der Flagge erscheint ein Finish Label auf dem Spiel.

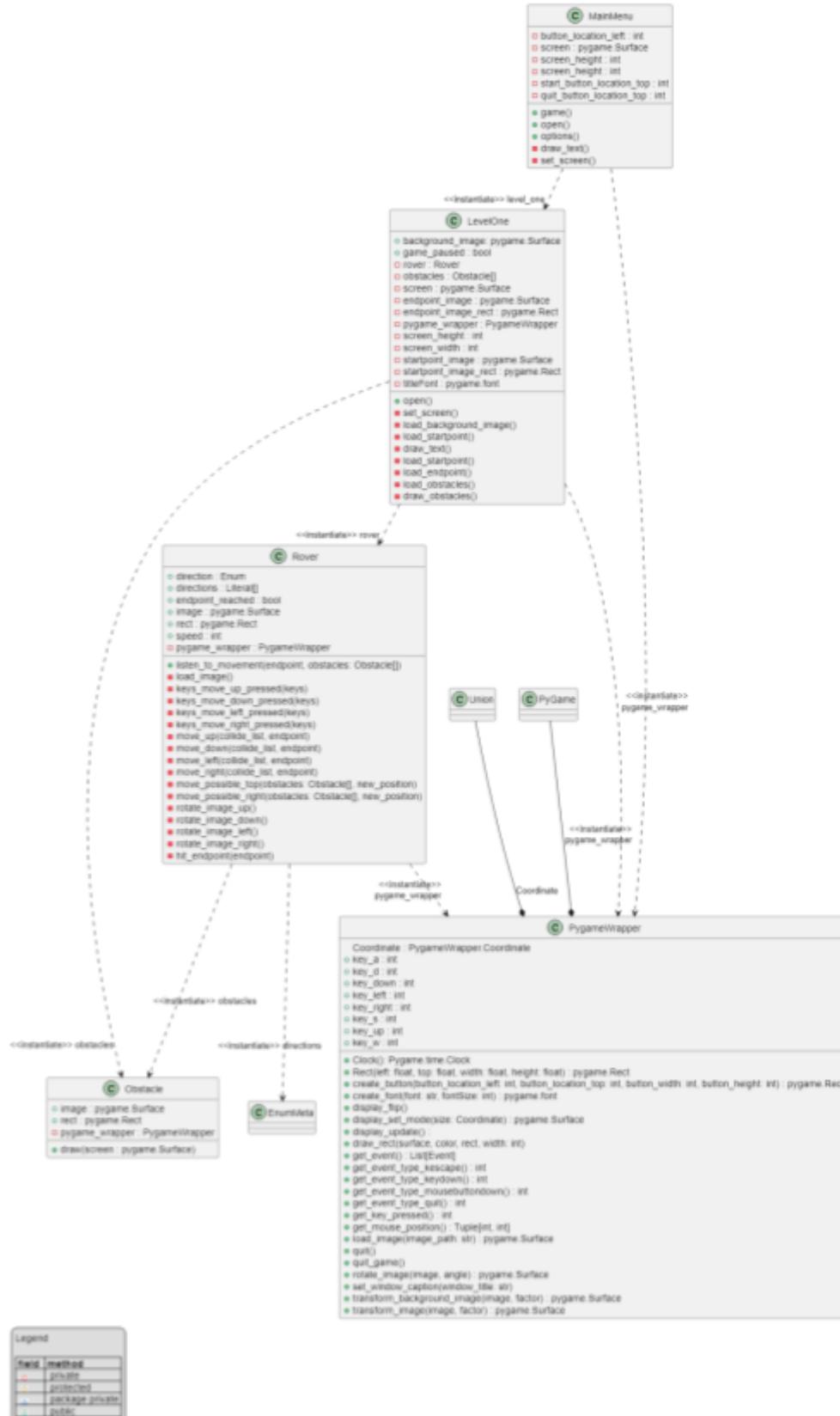


Level 1 finished

Im Klassendiagramm sind die Abhängigkeiten der einzelnen Klassen ersichtlich.

Celin Villars
 L-TIN-22-Di-a
 Im Grund 2
 8910 Affoltern am Albis
 079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
 L-TIN-22-Di-a
 Brünigstrasse 34
 6055 Alpnach Dorf
 079 607 67 11
sarah.birrer@edu.teko.ch



Klassendiagramm

Celin Villars
 L-TIN-22-Di-a
 Im Grund 2
 8910 Affoltern am Albis
 079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
 L-TIN-22-Di-a
 Brünigstrasse 34
 6055 Alpnach Dorf
 079 607 67 11
sarah.birrer@edu.teko.ch

Ein großer Teil der Software ist das Rover Movement. Der Rover muss sich auf der Karte frei bewegen, aber nicht in Hindernisse fahren können.

Dieses Sequenzdiagramm veranschaulicht, Implementation der Methode `listen to movement()`, welche sich in der Klasse `Rover` befindet.



Sequenzdiagramm rover movement

Die technische Dokumentation wurde während des ganzen Realisierungsprozesses ebenfalls überarbeitet und erweitert.

Nach dieser Phase haben wir unseren vierten Meilenstein erreicht und alle Features wurden fertig implementiert.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

5.4.1. Technische Dokumentation

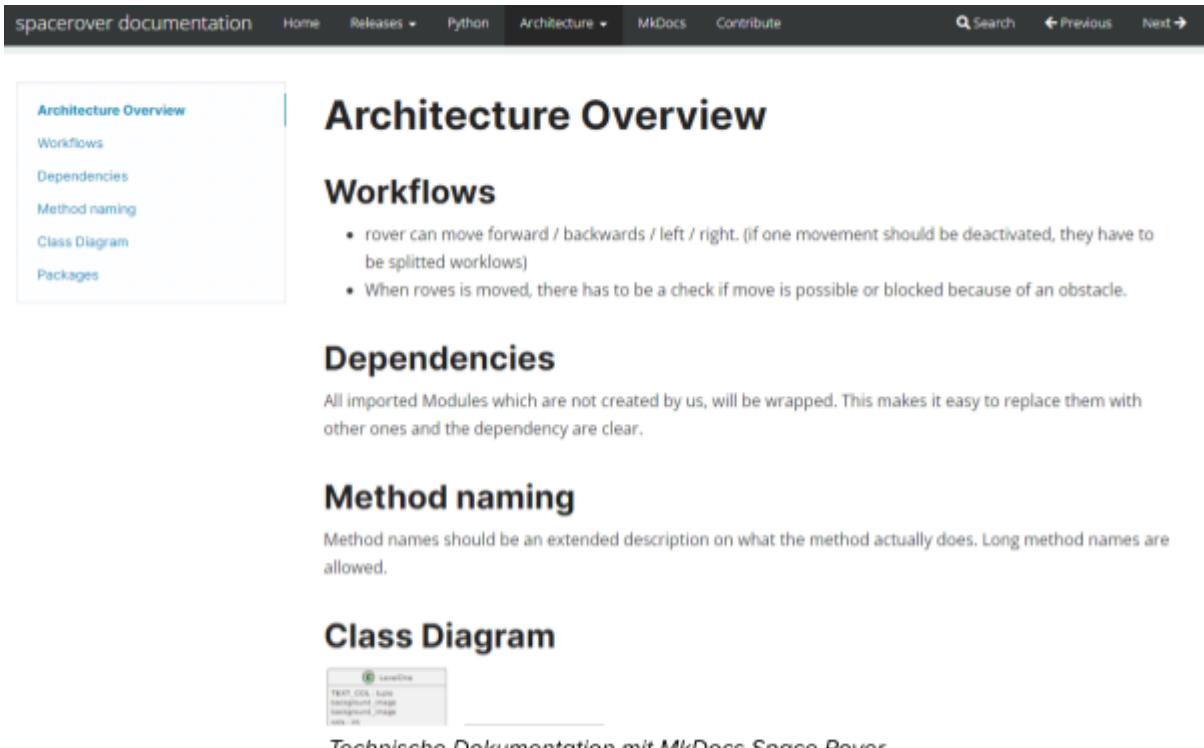
Die technische Dokumentation wurde in Englisch verfasst, da diese über Github (9) Pages veröffentlicht wurde und die ganze App als Open Source gilt.

In der technischen Dokumentation sind alle relevanten Informationen für die Weiterentwicklung des Spiels festgehalten, wie auch die Softwarearchitektur und die Beschreibung wichtiger Funktionen.

Die Dokumentation wurde bewusst getrennt, damit bei der Weiterentwicklung alle wichtigen Informationen für die Software direkt in der Software enthalten sind. Somit sind bei Anpassungen die relevanten Einträge in der Dokumentation zu diesem Zeitpunkt direkt zu finden. Gerade weil wir uns im Open Source Bereich bewegen, macht es Sinn, dass diese Informationen im Code erfasst sind.

Für die Weiterentwicklung des Spiels ist nur die technische Dokumentation notwendig.

Der Link zur technischen Dokumentation ist im Anhang unter [Technische Dokumentation](#) zu finden.



The screenshot shows a MkDocs generated website for a project named "Space Rover". The top navigation bar includes links for "Home", "Releases", "Python", "Architecture", "MkDocs", and "Contribute", along with a search bar and navigation icons for "Previous" and "Next". On the left, there's a sidebar titled "Architecture Overview" with links to "Workflows", "Dependencies", "Method naming", "Class Diagram", and "Packages". The main content area features a large heading "Architecture Overview" and several sections: "Workflows" (with a bulleted list about movement and obstacles), "Dependencies" (noting wrapped imported modules for easy replacement), "Method naming" (describing extended descriptions for methods), and "Class Diagram" (showing a small UML diagram of a class with attributes like "TEXT_COL", "TEXT", "background_color", "foreground_color", and "font_size"). A footer at the bottom reads "Technische Dokumentation mit MkDocs Space Rover".

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

5.5. Einführung

In dieser Phase haben wir uns um den Feinschliff unserer Dokumentation wie auch für das Projekt gekümmert. Wir haben die letzten Verschönerungen vorgenommen.

5.5.1. Testbericht

Die Tests wurden manuell durch Ausführen des Codes getätigt. Da die meisten Funktionen mit dem GUI zusammenhängen, wäre der Aufwand, alle Kopf- und Tastendrücke zu simulieren und einen kompletten Test zu schreiben, größer gewesen, als der Ertrag gerechtfertigt hätte. Die manuelle Methode war zweckdienlich und zeigte, falls nötig mit einigen gezielten Ausgaben in der Konsole, alle nötigen Daten zur Korrektur des Codes an. So konnten auftretende Probleme direkt behoben werden.

1. Testfall: Spiel kann geladen werden

Das Spiel kann geladen werden, es erscheint das Startmenü mit den beiden Knöpfen Start und Quit. Die Knöpfe entsprechen Ihrer Funktion.

Ergebnis

Beim Laden des Spiels wird das Menü geladen und die Knöpfe funktionieren. Nach Betätigen des Startknopfes wird das erste Level geladen und der Rover erscheint beim Startpunkt. Bei Betätigung des Quit Knopfes wird das Spiel beendet.

2. Testfall: Rover lässt sich bewegen

Der Rover kann sich bewegen, ohne dass das Spiel abstürzt.

Ergebnis

Der Rover kann sich auf der Karte bewegen.

3. Testfall: Rover kann in keine Hindernisse fahren

Ergebnis

Falls der Rover auf ein Hindernis trifft, kann er sich nicht weiter in das Hindernis hineinbewegen und bleibt stehen.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

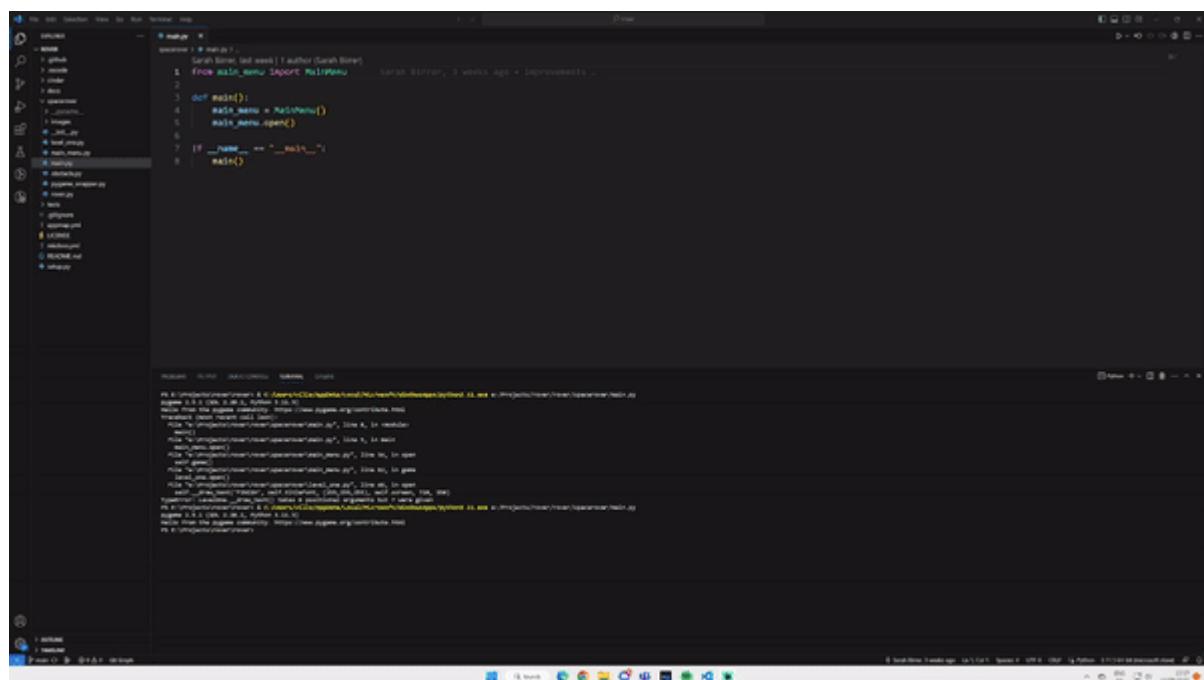
4. Testfall: Rover erreicht das Ziel

Der Rover berührt die Zielfahne.

Ergebnis

Das Spiel wird beendet.

Abschluss test als GIF:



The screenshot shows a Windows desktop environment. In the center, there is a terminal window with the following text:

```
PS C:\Users\birrer\OneDrive\Documents\GitHub\TIN-22-Di-a\TIN-22-Di-a> python main.py
Rover has reached the goal!
```

To the left of the terminal is a code editor window in VS Code. The file being edited is named "main.py". The code is as follows:

```
#!/usr/bin/env python3
# encoding: utf-8
# author: Sarah Birrer (Sarah Birrer)
# date: 2023-04-10
# version: 1.0

from main_menu import MainMenu
import time

def main():
    main_menu = MainMenu()
    main_menu.open()

    if __name__ == "__main__":
        main()

if __name__ == "__main__":
    main()
```

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

6. Abschlussbericht

6.1. Zielerreichung

Ziel: Ein benutzerfreundlicher Prototyp mit dem Thema Planetenerkundung wird erstellt, für den keine Anleitung benötigt wird.

Beurteilung: Ziel wurde erreicht, das Spiel verwendet die bekannte Steuerung WASD und Pfeiltasten. Zudem enthält es im Menu einen Start- und Quit Button.

Ziel: Der Rover kann sich auf der Karte bewegen.

Beurteilung: Ziel wurde erreicht, der Rover kann sich problemlos auf der Karte bewegen und kann auch nicht über Hindernisse fahren.

Ziel: Es soll eine modulare Codebasis erstellt werden, welche mit weiteren Modulen ergänzt werden kann.

Beurteilung: Ziel wurde erreicht, die Codebasis kann mit weiteren Modulen ergänzt werden. Es gibt ebenfalls eine Beschreibung für die Mitarbeit fürs Spiel in der technischen Dokumentation.

6.2. Verbesserungsvorschläge

6.2.1. Projektmanagement Verbesserungen

- Planung gezielter machen, Termine sollten fix eingeplant werden und bei jedem Termin einen Folgetermin einplanen oder prüfen ob dieser noch Sinn macht.
- Weniger Zeit für Entscheidungen und Evaluationen und mehr für Entwicklung einsetzen.
- Dokumentation laufend führen, nach Initialisierungsphase direkt Dokument erstellen.
- Technische Dokumentation einfach als PDF exportierbar machen.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

6.2.2. Software Verbesserungen

- Weitere Levels mit anderen Karten und Hindernissen.
- Gegner, welche den Rover beschädigen können
- Optionen im Menü, mit diversen Einstellungen wie Steuerung des Rovers, Schwierigkeit.
- Feld für Benutzername im Menü, um eine Rangliste am Ende des Spiels anzuzeigen mit Zeitmessung oder Punktestand.
- Architektur überdenken, damit PyGame (8) einfach zu mocken ist.
- Unit Tests erstellen für Testabdeckung.

6.3. Persönliche Reflexion

6.3.1. Celin Villars

Für mich war die Projektarbeit sehr lehrreich und hat mir sehr viele Anwendungen in Python näher gebracht. Da ich noch sehr neu in diesem Bereich und froh bin, Erfahrung sammeln zu können, ist das Erreichen der Ziele und das Erarbeiten neuer Kenntnisse definitiv als Erfolg zu werten. Es lief natürlich nicht alles perfekt, doch wir konnten die Probleme erfolgreich lösen.

Der Start lief sehr gut, wir hatten schnell eine gute Basis und konnten gut auf dieser aufbauen. Vielleicht sind wir in diesem Bereich sogar etwas zu schnell vorangekommen, denn als es um die Darstellung ging und wir mehrere Wege begutachteten, hatten wir bereits vieles programmiert, was danach nicht mehr nötig war oder umgeschrieben werden musste. Die Dokumentation wurde in diesem Zeitraum auch etwas vernachlässigt, was uns gegen Ende des Projekts dazu zwang, uns mehr auf die Dokumentation als um Erweiterungen des Codes zu kümmern.

Lessons Learned

- Früher über die (Darstellungs-) Methoden entscheiden, um später nicht zu viele "Umbauten" vornehmen zu müssen.
- Laufend dokumentieren

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

6.3.2. Sarah Birrer

Ich persönlich finde die Projektarbeit war ein Erfolg, da wir alle unsere Ziele erreichten. Meine persönlichen Erwartungen wurden leider nicht alle erfüllt. Ich habe erwartet, dass wir mehr entwickeln können in dieser Zeit. Leider hat das ganze Projektmanagement mehr Zeit in Anspruch genommen als erwartet. Anstatt Code Verbesserungen vorzunehmen, mussten wir leider unseren Fokus auf die Dokumentation und Präsentation legen. Im Rahmen der Projektarbeit hätte ich gerne versucht, Clean Architecture praktisch anzuwenden, damit die Codebasis qualitativ besser ist und einfacher zu erweitern ist.

Die Zusammenarbeit im Team habe ich als sehr gut empfunden, wir haben uns mehrmals abgestimmt und jeder hat seinen Teil gefunden für die Realisierung. Durch die Aufteilung für die GUI Komponente konnten wir gut parallel arbeiten.

Als wir 80% des Projektes erledigt haben, waren wir nicht mehr so ambitioniert wie am Start des Projektes. Wir haben aber trotzdem die für uns wichtigen Punkte noch umsetzen können.

Lessons Learned

- Komplexe Methoden aufzeichnen in einem passenden Diagramm, dadurch kann das Problem veranschaulicht werden und der Code ist einfacher zu schreiben.
- Dokumentation regelmäßiger führen
- Fixe Termine im Kalender können sehr hilfreich sein, um den Zeitplan einzuhalten.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

7. Quellenverzeichnis

Quellen der technischen Dokumentation sind direkt in den Markdown Files hinterlegt.

1. "The Python Arcade Library." *The Python Arcade Library*, 2022,
<https://api.arcade.academy/en/latest/>. Accessed 26 July 2022.
2. "TkInter." *Python interface to Tcl/Tk*, 8 9 2023,
<https://docs.python.org/3/library/tkinter.html#module-tkinter>. Accessed 11 June 2023.
3. "monday." *monday work management*, 2023, <https://monday.com/>. Accessed 6 June 2023.
4. *Die visuelle Kollaborationsplattform für jedes Team*. Miro, 2023. *Miro*, <https://miro.com/>.
Accessed 4 July 2023.
5. *Google Docs*. Google Docs, 2023, https://www.google.com/intl/de_ch/docs/about/.
Accessed 2 May 2023.
6. *PlantUml Editor*. PlantUml, 2023. *PlantUml - Ein kurzer Überblick*, <https://plantuml.com/de/>.
Accessed 29 Juli 2023.
7. *Legende PlantUml*. stack overflow, 2019. *PlantUML Class Diagram - field/method visibility in legend*,
<https://stackoverflow.com/questions/54882822/plantuml-class-diagram-field-method-v>
isibility-in-legend. Accessed 10 September 2023.
8. *Pygame*. PyGame, 2023. *PyGame Getting Started*,
<https://www.pygame.org/wiki/GettingStarted>. Accessed 10 July 2023.
9. *Github*. Github, 2023, <https://github.com/>. Accessed 25 April 2023.
10. *Git*. Git, 2023. *--distributed-even-if-your-workflow-isnt*, <https://git-scm.com/>. Accessed 25 April 2023.
11. *Visual Studio Code*. Microsoft, 2023. *Code editing. Redefined.*,
<https://code.visualstudio.com/>. Accessed 25 April 2023.

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

7.1. Abbildungen

Bild	Quelle
Python Logo	https://en.wikipedia.org/wiki/Python_%28programming_language%29
Pygame Logo	https://www.pygame.org/docs/logos.html
Git Logo	https://www.stickpng.com/img/icons-logos-emojis/tech-companies/git-fu_ll-logo
Github Logo	https://wwwvecteezy.com/vector-art/16833880-github-logo-git-hub-icon-with-text-on-white-background
VS Code Icon	https://de.m.wikipedia.org/wiki/Datei:Visual_Studio_Code_1.35_icon.svg
Discord Icon	https://discord.com/branding
Whatsapp Icon	https://de.m.wikipedia.org/wiki/Datei:WhatsApp.svg
Microsoft Teams Icon	https://de.m.wikipedia.org/wiki/Datei:Microsoft_Office_Teams_%282018%E2%80%93present%29.svg
Google Docs Icon	https://www.flaticon.com/free-icon/google-docs_5968517
Miro Logo	https://worldvectorlogo.com/logo/miro-2
PlantUml Logo	https://de.m.wikipedia.org/wiki/Datei:Plantuml_Logo.svg

Celin Villars
L-TIN-22-Di-a
Im Grund 2
8910 Affoltern am Albis
079 505 91 60
celin.villars@edu.teko.ch

Sarah Birrer
L-TIN-22-Di-a
Brünigstrasse 34
6055 Alpnach Dorf
079 607 67 11
sarah.birrer@edu.teko.ch

8. Anhang

8.1. GIT-Repository

Das komplette Projekt ist als GIT-Repository verfügbar.

Teko: <https://elad.ch/gitblit/summary/~birrers!spacerover.git>

Github: <https://github.com/teko-L-TIN-22-Di-a/rover>

8.2. Terminplan / Gantt

Terminplan wie auch das Gantt ist im GIT-Repository unter /docs/projectmanagement/ abgelegt.

8.3. Nutzwertanalyse GUI Komponente

Die Nutzwertanalyse ist im GIT-Repository unter /docs/projectmanagement/ abgelegt.

8.4. Einladung Projekt Meeting

Die Einladung für die Projektsitzung ist im GIT-Repository unter /docs/projectmanagement/ abgelegt.

8.5. Projekt Meeting Protokoll

Das Meeting Protokoll ist im GIT-Repository unter /docs/projectmanagement/ abgelegt.

8.6. Technische Dokumentation

Die technische Dokumentation ist unter folgendem URL erreichbar.

Github: <https://teko-l-tin-22-di-a.github.io/rover/>