

Bearbeitungsbeginn: [01.09.2021]

Vorgelegt am: [27.02.2022]

# **Thesis**

Zur Erlangung des Grades

**Bachelor of Science**

Im Studiengang Medieninformatik

An der Fakultät Digitale Medien

**Christian Micka**

Matrikelnummer: 259753

**Entwicklung eines Serious Game-Prototyps mit Fokus auf die  
Verbindung von Storytelling und Lernerfahrung am Beispiel  
"Grundlagen des Programmierens"**

Erstbetreuer: Prof. Jirka Dell'Oro-Friedl

Zweitbetreuer: Prof. Dr. Dirk Eisenbiegler



# Inhaltsverzeichnis

Vorwort.....	5
Disclaimer.....	5
Abstract.....	7
1. Einleitung .....	9
1.1. Konzept .....	9
1.2. Idee .....	10
1.3. Aufbau der Arbeit.....	12
2. Konkurrenzanalyse.....	13
2.1. Minecraft – Education Edition .....	13
2.2. Duskers .....	14
2.3. CodeCombat.....	14
3. Konzeption .....	15
3.1 Wahl des Settings.....	15
3.2. Modularität.....	15
4. Umsetzung .....	17
4.1. Verwendete Technologien .....	17
4.1.1. FUDGE.....	17
4.1.2. TypeScript.....	17
4.1.3. GitHub .....	17
4.1.4. Live Server .....	18
4.1.5. Affinity Designer, Affinity Photo .....	18
4.1.6. Blender.....	18
4.2. Aufbau des Spiels.....	19
4.2.1. Der strategische Teil.....	19
Wie wird die Welt generiert? .....	20
Wie orientieren sich Drohnen? .....	22
Drohnen bauen und Funktionen freischalten.....	23
Wie kämpfen Drohnen? .....	25
4.2.2. Die Lerninhalte .....	26
Die Geschichte .....	26
Die Quests.....	26
Die Rätsel.....	27
Das Archiv .....	29
7.3. Grafischer Aufbau .....	29
7.4. Erzählung .....	30

7.4.1. Warum eine Geschichtserzählung in Serious Games? .....	30
7.4.2. Geschichte .....	32
7.5. Probleme bei der Umsetzung .....	33
8. Testphase mit gemischter Nutzergruppe.....	35
8.1. Wussten Sie, wie Sie sich im Spiel zurechtfinden können? .....	35
8.2. War das bereitgestellte Lernmaterial verständlich formuliert? .....	36
8.3. Hatten Sie Probleme beim Lösen der Rätsel? .....	36
8.4. Verspüren Sie Interesse, die Geschichte trotz der Rätsel weiter zu verfolgen? ....	37
8.5. Wären Sie bereit, weitere Lernthemen zu verinnerlichen, um die Geschichte weiter zu erleben? .....	38
8.6. Fazit .....	39
9. Geplante Features .....	41
9.1. Erweiterung der Lernthemen und Rätsel .....	41
9.2. Verbesserte Visualisierung für Kämpfe.....	42
9.3. Bewegliche Feinde .....	42
9.4. Verbesserung der Orientierung im Spiel.....	43
9.5. Zusätzliche Funktionen für Drohnen .....	43
9.6. Online-Funktionalität.....	43
9.7. Speicherfunktion.....	44
10. Fazit.....	45
11. Quellenverzeichnis.....	47
12. Eidesstattliche Erklärung.....	49

# Vorwort

Das Konzept von Serious Games beschäftigt und interessiert mich nun schon seit geraumer Zeit. Dies gilt ebenso für die Konzeption von Spielen, weshalb ich bereits die Module "Prototyping interaktiver Medien-Apps und Games", sowie den "Gamedesign-Workshop" besuchte. Da ich selbst weiß, dass Lernthemen teilweise sehr trocken gestaltet sein können und die Lernmotivation deshalb oft sinkt, eine gute Geschichte jedoch sehr fesselnd sein kann, kam mir die Idee für diese Arbeit. Da ich bereits zuvor die Engine FUDGE in Verbindung mit Prototyping genutzt hatte, entschied ich mich dazu, diese erneut zu verwenden.

Aus den oben genannten Gründen ergab sich meine Themenwahl für diese Arbeit, den Prototyp eines Serious Games zu entwickeln, welcher darauf setzt, die Lernmotivation anhand der Narration hoch zu halten und die Themen trotzdem auch spielerisch zu vermitteln.

An diesem Punkt möchte ich mich für die umfassende Hilfe meines Erstbetreuers, Prof. Jirka Dell'Oro-Friedl und meines Zweitbetreuers, Prof. Dr. Dirk Eisenbiegler bedanken, die mir stets mit gutem Rat und Ideen zur Seite standen und mit kritischen Fragen dafür sorgten, dass dieses Projekt sich in eine gute Richtung entwickelt hat. Ebenso gilt mein Dank all denen, die sich bereit erklärt haben, an meiner Testphase teilzunehmen und mir wertvolle Informationen über meinen Prototyp und seine Handhabung zu geben.

## Disclaimer

Aus Gründen der besseren Lesbarkeit wird in dieser Arbeit das generische Maskulinum verwendet. Die Inhalte dieser Arbeit treffen ebenso auf weibliche Personen und Personen mit anderweitiger Geschlechteridentität zu, soweit es für die Aussage erforderlich ist. [1]



# Abstract

Bei einem Serious Game handelt es sich um ein Spiel, bei welchem nicht ausschließlich die Unterhaltung im Fokus steht, sondern in denen der Unterhaltungsaspekt mit dem Lernaspekt aufgewogen wird. Ziel dieser Bachelorarbeit ist es, ein „Serious Game“ zu entwickeln, welches die Lernmotivation mithilfe der Einbindung in die Geschichte des Spiels steigern und erhalten soll und mithilfe dieses Spiels zu prüfen, ob erhöhter Fokus auf Geschichtserzählung die Lernmotivation steigern kann. Dabei wird darauf Wert gelegt, die Lernthemen austauschbar zu gestalten und somit das Anwendungsspektrum zu steigern, indem problemlos neue Themengebiete und Geschichten implementiert werden können. Es wurden ähnliche Spiele betrachtet, die ihre Lernthemen mit Storytelling kombinieren und deren Probleme wurden analysiert, um die Lernerfahrung in dieser Arbeit zu verbessern. Um dieses Ziel zu erreichen, wurde ein Prototyp konzipiert und umgesetzt, welcher die Bereiche Unterhaltung, Wissen erlangen und Wissen abfragen miteinander kombiniert und aufeinander aufbauen lässt. Der Nutzer kommt dadurch mit allen Bereichen des Spiels in Kontakt und erfährt die Lernthemen, in diesem Fall das Themenfeld der Informatik, aus verschiedenen Blickwinkeln.





# 1. Einleitung

Mit diesem Projekt wird die These geprüft, ob die Erzählung einer Geschichte und die Implementierung der Spielinhalte in diese Geschichte die Lernmotivation des Nutzers fördern kann.

Ein Serious Game, welches den Fokus auf Storytelling und die Implementierung der Lerninhalte in besagte Geschichte legt, ist in der heutigen Medienlandschaft noch kaum zu finden. Um diesen Teilbereich der Serious Games näher zu erforschen, befasst sich diese Arbeit mit der Konzeption und Erstellung eines Prototyps, welcher anschließend durch eine Nutzerumfrage getestet wird. Anhand der gesammelten Umfrageergebnisse wird anschließend festgestellt, ob das Konzept einen Mehrwert für die zukünftige Konzeption von Serious Games bietet. Ebenso werden die Ergebnisse genutzt, um den Prototyp dahingehend zu erweitern und zu konzipieren, dass festgestellte Mängel oder Probleme in späteren Ausführungen behoben oder verbessert werden.

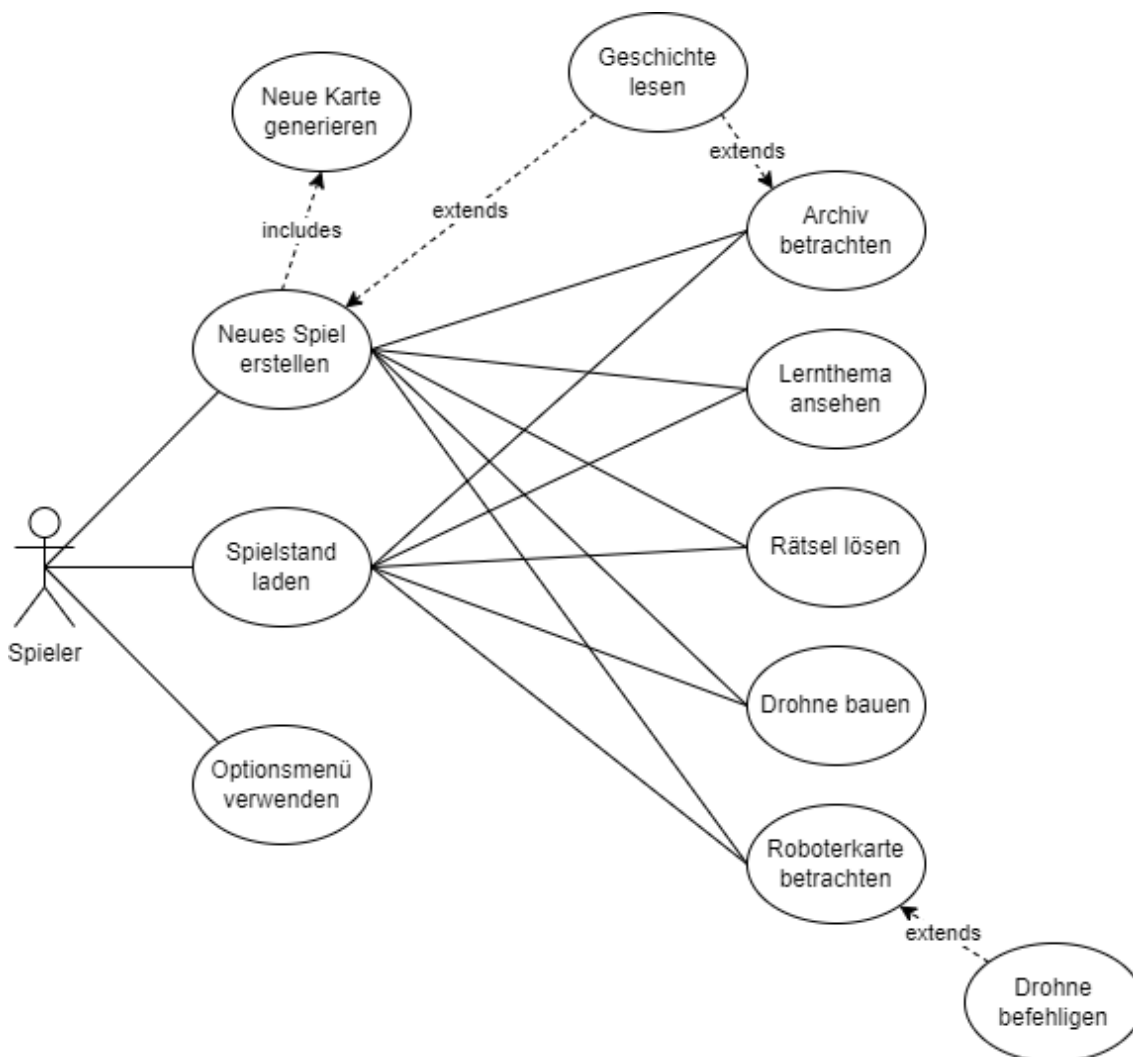
## 1.1. Konzept

Die Verbindung von Serious Games und Narration ist bisher noch relativ selten verwendet worden. Der Großteil der Serious Games verwendet spielerische Aufgaben, um das jeweilige Lernthema zu vermitteln, bindet dabei allerdings kaum eine Geschichte mit ein. Wenn eine Geschichte Teil des Spiels ist, dann wird diese meist nebenbei erzählt und ist nicht relevant für das Spielerlebnis.

Eine gute Geschichte hat jedoch den Vorteil, dass sie das Interesse des Lesers oder in diesem Fall das Interesse des Spielers weckt und mit spannenden Wendepunkten den Drang nach mehr Informationen weckt. Der Spieler möchte wissen, wie sich die Geschichte der Welt und seines Charakters weiterentwickelt und ist deshalb eher gewillt, Strapazen auf sich zu nehmen, um dies zu erreichen. Durch die Verbindung von Lernthemen und einer auf diese Themen zugeschnittene Geschichte wird ein hoher Grad an Immersion erzeugt, die den Spieler ermutigen soll, die Lernthemen zu verinnerlichen, um dadurch weitere Kapitel der Geschichte freizuschalten und nebenbei die Lerninhalte selbst anzuwenden.

## 1.2. Idee

Das Grundkonzept für das Projekt entstand aus der Idee, Drohnen zu programmieren, die selbstständig agieren und nach vorgegebenen Handlungsmustern auf Ereignisse reagieren können. Daraus folgte die Verbindung, das Spiel als Serious Game mit Lerninhalten zum Thema Programmieren zu gestalten. Im Laufe der Arbeit wurde das Konzept dahingehend erweitert, die Spielinhalte so zu strukturieren, dass einzelne Themen und Komponenten austauschbar gestaltet werden. Für das Thema der Geschichte wurde die Erforschung eines fremden Planeten mithilfe von Drohnen gewählt, welche die toxischen Außengebiete der Spielwelt erkunden und bewirtschaften können, während sich der Spielercharakter selbst auf die Lerninhalte und die ferngesteuerten Befehle der Drohnen konzentriert. [siehe Abbildung 1, Abbildung 2]



[Abbildung 1, Use-Case-Diagramm]



## 1.3. Aufbau der Arbeit

Diese Arbeit ist unterteilt in die Bereiche Konkurrenzanalyse, Verwendete Technologien, Umsetzung, Testphase und Zusammenfassung.

Das Kapitel "Konkurrenzanalyse" verschafft einen Überblick über einige Serious Games, welche Ähnlichkeiten zu diesem Projekt aufweisen. Analysiert wird, inwiefern die für dieses Projekt relevanten Aspekte in besagten Konkurrenzprodukten verwirklicht werden und welche Aspekte fehlen, um das gewünschte Ziel dieser Arbeit ebenfalls zu erreichen.

Anschließend werden unter "Verwendete Technologien" die Programme und Hilfsmittel näher erläutert, die zur Anfertigung dieses Prototyps genutzt wurden. Es werden ebenso Gründe geliefert, weshalb die Wahl auf diese Technologien fiel.

Im Kapitel "Umsetzung" wird die Herangehensweise zur Erstellung des Prototyps genauer beschrieben. Es wurden Aspekte herausgearbeitet, auf die sich dieses Projekt beziehen und stützen soll, neben der Erläuterung dieser Aspekte wird die Wahl des Settings und des grafischen Aufbaus beleuchtet.

Da dieser Prototyp einigen Nutzern für eine Testphase zur Verfügung gestellt wurde, werden die Ergebnisse dieser im Kapitel "Testphase" erläutert und ausgewertet. Beschrieben wird der Ablauf der Testphase, sowie die Ergebnisse und die daraus entstandenen Änderungen.

Das Kapitel „Geplante Features“ befasst sich mit Erweiterungen des Spiels, welche im Rahmen dieses Projektes nicht umgesetzt werden konnten oder für die Klärung der These nicht ausschlaggebend waren.

Abschließend finden sich im Kapitel "Zusammenfassung" Lösungsansätze für Probleme, die durch die Testphase aufgekommen sind und ein abschließendes Fazit.

## 2. Konkurrenzanalyse

Im nachfolgenden Kapitel wird ein Überblick über vergleichbare Spiele gegeben, die überschneidende Aspekte mit diesem Projekt haben.

### 2.1. Minecraft – Education Edition

Die Minecraft - Education Edition ist eine von Mojang und Microsoft entwickelte Version des Spiels Minecraft. Innerhalb der Education Edition ist es Lehrern möglich, virtuelle Lernatmosphären zu erschaffen und verschiedenste Themen zu lehren. Hierzu wurden verschiedene Werkzeuge entwickelt, welche die Funktionen der normalen Minecraft-Version erweitern. So können beispielsweise verschiedene virtuelle Tafeln aufgestellt werden, um beispielsweise den Ablauf von Arbeitsschritten zu erläutern oder Aufgaben zu vergeben.

Ein besonderes Feature der Education-Edition ist der Code-Editor, mit welchem die Schüler in vereinfachter Art und Weise Code zusammenfügen können. Dieser Editor erinnert stark an die Programmieroberfläche Scratch und nutzt ebenfalls Bausteine, um funktionalen Code zu generieren. Diese Bausteine können mit verschiedenen Funktionen definiert und mit Parametern ausgestattet werden, um die Funktionsweise an die gegebenen Anforderungen anzupassen.

Durch ihre Modularität ist diese Version von Minecraft eines der beliebtesten Serious Games, das eine nutzerfreundliche Oberfläche für verschiedenste Themen liefert.

Ein großes Problem ist allerdings, dass für die Nutzung der Minecraft – Education Edition ein gültiger Microsoft 365 Account notwendig ist (Stand 13.02.2022).

Die Minecraft - Education Edition bietet zwar auch Ansätze, welche dem Ziel dieses Projektes ähneln, jedoch ist die Implementierung der Geschichte nebensächlich und nicht relevant für den Abschluss der Lektionen. Des Weiteren leidet diese Version darunter, dass Minecraft ein gerade bei jüngeren Jugendlichen weit verbreitetes Spiel ist, welches durch Freiheit und Vielfalt überzeugt. Diese Freiheit wird durch die stromlinienartige Lernumgebung eingeschränkt, was bei vielen Nutzern zu Unmut führen und die Lernmotivation senken kann. [2]

## 2.2. Duskers

Duskers ist ein Taktikspiel mit Rogue-lite Elementen, bei dem der Spieler eine Gruppe von Drohnen durch verlassene Raumschiff-Wracks navigiert, um diese zu erforschen und auszuschlachten. Hierbei hat der Spieler neben der direkten Steuerung die Möglichkeit, die Drohnen mithilfe einer sehr vereinfachten Programmiersprache über Codeeingabe zu steuern und mehrere Befehle aneinander zu reihen. Das Konzept von Duskers zeigt einige Parallelen zur grundlegenden Idee des strategischen Teils dieses Projektes. Mithilfe der Codeeingabe lassen sich grundlegende Abläufe und Automatisierung trainieren. Während Duskers nur ein geringes Maß an Geschichte vorweist, so ist es für dieses Spiel ein Vorbild, sowie eine Inspiration für diverse Features in diesem Projekt. Die Lernerfahrung beschränkt sich zwar auf das Schreiben von grundlegendem Code, aber die Übung darin ist für das Erlernen von Programmiersprachen und deren Programmabläufe definitiv hilfreich. [3]

## 2.3. CodeCombat

Auch CodeCombat befasst sich primär damit, dem Nutzer Programmierkenntnisse beizubringen. Mithilfe von Codeeingaben steuert der Spieler Helden durch verschiedene, mit Hindernissen ausgestattete Levels und gibt ihnen Anweisungen, um diese zu bestehen. Es besteht die Möglichkeit, aus verschiedenen Programmiersprachen wie JavaScript, Python und C++ zu wählen, um die Level zu absolvieren. Die verschiedenen Zonen der Spielwelt befassen sich dabei mit unterschiedlich komplexen Programmieraufgaben aus diversen Themenfeldern. Ebenso ist es dem Spieler möglich, zu wählen, ob er sich eher in Richtung Web-Entwicklung oder Spieleentwicklung fortbilden möchte. Während der Lernaspekt in CodeCombat stark hervorgehoben wird, so bleibt die Geschichte des Spiels oberflächlich und nicht zwingend relevant für das Spielerlebnis oder den Abschluss. Auch CodeCombat wirkte inspirativ auf dieses Projekt, indem es die ansteigende Schwierigkeit der Rätsel und Aufgaben mit unterschiedlichen Themengebieten gut umsetzt und dem Spieler ermöglicht, sich Stück für Stück durch ein bestimmtes Thema durchzuarbeiten. [4]

## 3. Konzeption

### 3.1 Wahl des Settings

Zu Beginn lag der Fokus auf einer packenden Narration, da sich das Spiel stark auf die erzählte Geschichte stützen und die Lerninhalte darin einbinden sollte. Die futuristischen, Science-Fiction-angehauchten Aspekte des Spiels bieten eine gute Grundlage, um aufregende Handlungen zu erzählen und das Lernthema Programmierung passend einzubinden.

Der strategische Anteil des Spiels wurde dabei als erstes geplant und sollte dem Spieler die Möglichkeit nehmen, die Drohnen direkt zu steuern, sondern ihn dazu aufzufordern, sich passende Handlungsmuster zu überlegen, mit denen die Drohnen selbst ihre Aufgaben erkennen und umsetzen. So hat der Spieler die Möglichkeit, zu wählen, welche Rohstoffe eine Drohne abbauen soll oder wie er sich im Kontakt mit Feinden verhält. Im Laufe des Spiels werden weitere Funktionen durch den Abschluss der entsprechenden Kapitel freigestaltet, um komplexere Abläufe festzulegen und die Drohnen noch autonomer arbeiten zu lassen.

Anschließend wurde die Geschichte in Lernkapitel aufgeteilt, die sich jeweils mit einem bestimmten Thema befassen. Die Rätsel, die der Spieler während des jeweiligen Kapitels lösen muss, befassen sich mit den bereits gelernten Inhalten und setzen sich zum einen aus bereits gelernten Themen, aber auch aus neuen Themen zusammen. Somit soll das gelernte Wissen immer wieder abgefragt und gefestigt werden. Die Rätsel sind auf verschiedene Schwierigkeitsgrade angepasst und nehmen entsprechend dem Fortschritt im Spiel an Komplexität zu. Während zu Beginn primär textbasierte Aufgaben gestellt werden, so wandeln sich diese im späteren Spielverlauf beispielsweise zu Rätseln, bei denen einer Drohne ein mehrstufiger Handlungsablauf einprogrammiert werden muss, um ein Labyrinth oder Ähnliches zu bewältigen.

### 3.2. Modularität

Ein wichtiger Teil des Projektes war es weiterhin, ein hohes Maß an Modularität zu erhalten, sodass mit entsprechendem Schreibaufwand einfach neue Lernthemen implementiert werden können. Auch die Geschichte ist so gestaltet, dass sie durch neue Handlungsstränge erweitert oder ausgetauscht werden kann. Jede

Komponente des Spiels kann somit angepasst werden, während die anderen Komponenten erhalten bleiben. Sollte beispielsweise ein neues Setting gewünscht sein, welches ein anderes Lernfeld bedient, so können die Inhalte anhand von JSON-Dateien umgeschrieben und eingefügt werden. Der strategische Anteil des Spiels, also die Fernsteuerung der Drohnen und Sammlung von Ressourcen bedarf zwar eines gewissen Programmieraufwandes, aber auch diese kann durch ein passenderes Spielkonzept ersetzt werden, wenn dies gewünscht ist.



## 4. Umsetzung

### 4.1. Verwendete Technologien

#### 4.1.1. FUDGE

Bei FUDGE handelt es sich um eine auf TypeScript basierende open-source Game-Engine für Studenten. Entwickelt wurde sie von Prof. Jirka Dell'Oro-Friedl und mehreren Studenten der Hochschule Furtwangen. Die Wahl von FUDGE basiert auf der Nutzung dieser Software im Modul Prototyping interaktiver Medien-Apps und Games, in welchem die Anwendung zur Erstellung eines Software-Prototyps gelehrt wird. Aufgrund der Vertrautheit erwies sich FUDGE als naheliegendste Wahl für die Durchführung dieses Projektes. Ebenso kam die Expertise von Prof. Dell'Oro-Friedl zugute, was die Arbeit an dem Prototyp in vielen Bereichen vereinfachte. Alternativ wäre Unity eine mögliche Engine gewesen, die Entscheidung fiel letzten Endes nach Abwägung der Vor- und Nachteile jedoch auf FUDGE. [5]

#### 4.1.2. TypeScript

Bei TypeScript handelt es sich um eine von Microsoft entwickelte Programmiersprache auf Basis von JavaScript. TypeScript besticht durch die starke Typisierung. Dadurch ist das Programm weniger anfällig für Fehler bei der Verarbeitung von Variablen mit unterschiedlichen Datentypen, da derartige Fehler von Compilern erkannt werden und nicht erst zur Laufzeit auftreten. Ebenso erweitert TypeScript die grundlegenden Funktionen von JavaScript, kann gleichzeitig aber auch reinen JavaScript-Code verarbeiten, wodurch bestehende in JavaScript definierte Funktionen genutzt werden können. Die Tatsache, dass FUDGE in TypeScript geschrieben ist, bestärkt dessen Nutzung zusätzlich. [6]

#### 4.1.3. GitHub

Bei GitHub handelt es sich um einen Dienst zur Versionsverwaltung. Mit GitHub lässt sich programmierter Code einfach in ein Online-Repository hochladen, um vor Datenverlust zu schützen und mehrere Versionen eines Codes zu haben, auf die man im Falle eines Problems zurückgreifen kann. In GitHub kann nicht nur Code, sondern auch die Dokumentation und weitere Dateiformate gespeichert werden. Mithilfe von Branches können auf sichere Weise weitere Features implementiert

werden, ohne die grundlegende Version des Codes zu verändern oder zu gefährden. Mithilfe von GitHub Pages wurde der Prototyp der Nutzergruppe der Testphase zur Verfügung gestellt, ohne eine separate Website hosten zu müssen. Gewählt wurde GitHub aufgrund der einfachen Nutzung mithilfe der Software GitHub Desktop, welche lokalen Code direkt in das dafür angelegte Online-Repository pushen kann. Zeitweise wurde auch die Nutzung von GitBash für die Versionsverwaltung erwägt, jedoch war die Struktur dessen nicht intuitiv genug und das User Interface konnte nicht überzeugen, da es eher einer PowerShell gleicht, anstatt eine voll ausgearbeitete, grafische Nutzeroberfläche zu bieten.

#### 4.1.4. Live Server

Live Server ist eine Erweiterung für den Code Editor Visual Studio Code, welche es auf einfache Weise ermöglicht, den aktuellen Stand des Codes im Browser zu testen, ohne eine Internetverbindung zu erfordern.

#### 4.1.5. Affinity Designer, Affinity Photo

Die Affinity Programme Designer und Photo stellen eine kostengünstige und hochwertige Alternative zu den Programmen Photoshop und Illustrator von Adobe dar. Das Funktionsspektrum ist nahezu identisch, während die Kosten sich auf eine einmalige Anschaffung beschränken, anstatt ein Abonnement für die gesamte Adobe Cloud zu erfordern.

Affinity Photo ist ein pixelbasiertes Zeichenprogramm und wurde von für die Gestaltung der Pixelgrafiken der Roboter und Drohnen, sowie die Kacheln der Weltkarte genutzt. Es bietet eine einfache Handhabung und hilfreiche Tutorials, welche die komplexeren Funktionen optimal erklären. Durch die Unterstützung von Grafik Tablets und Druckintensitäten der zugehörigen Stifte können außerdem Freihand-Zeichnungen mit schöner Optik erstellt werden. Dies wurde unter anderem für die Illustration der Geschichte genutzt.

Affinity Designer ist ein vektorbasiertes Zeichenprogramm, mit dem skalierbare Bilddateien erstellt werden können.

#### 4.1.6. Blender

Mit der 3D-Grafiksuite lassen sich 3D-Modelle modellieren, rendern und animieren. Die Handhabung des Programmes ist sehr nutzerfreundlich und leicht zu erlernen, weshalb sich Blender gegenüber Konkurrenten wie 3DsMax und Maya hervorgetan

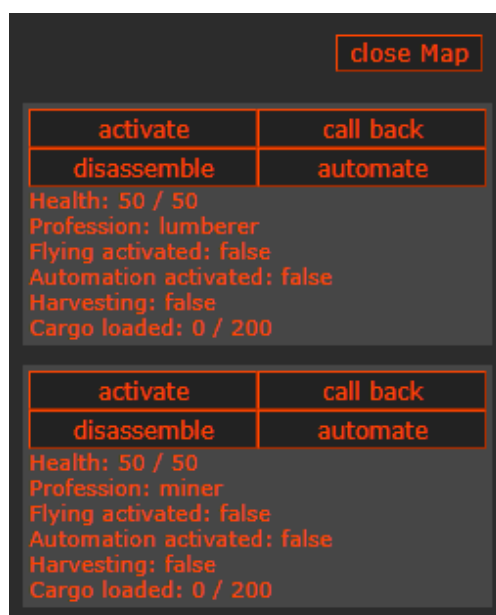
hat. Mithilfe von Blender wurden die Grafiken für das Hauptmenü erstellt und gerendert. Auch die Texturierung ist innerhalb von Blender möglich, wurde für den Prototyp jedoch übergangen, da der Fokus auf der technischen Umsetzung und nicht auf der Gestaltung der Oberflächen lag.

## 4.2. Aufbau des Spiels

Das Spiel selbst ist in zwei grundlegende Spielelemente unterteilt. Diese sind der strategische Teil, sowie das Lösen von Übungsaufgaben. Unterstützt und verbunden werden diese durch die Spieloberfläche, welche Zugang zu allen Funktionen des Spiels bietet.

### 4.2.1. Der strategische Teil

Der strategische Teil des Spiels befasst sich damit, die Erkundungsdrohnen in die Außenwelt zu senden und zu überwachen. Hier können den Drohnen Befehle gegeben werden [siehe Abbildung 3], um diese beispielsweise zur Basis zurückkehren oder automatisiert arbeiten zu lassen. Des Weiteren bietet die strategische Oberfläche einen Überblick über alle aktiven Drohnen und deren Zustand. Die Weltkarte, auf welcher sich die Drohnen bewegen, wird zu Beginn des Spiels berechnet und ändert sich nur minimal. So werden Rohstofffelder durch Ebenen ersetzt, sobald die Rohstoffe des Feldes aufgebraucht sind.



[Abbildung 3, Seitenleiste der Roboterkarte mit Kontrolloptionen]

## Wie wird die Welt generiert?

Bei der Generierung der Welt wird eine Noisemap berechnet, die sich an der gewünschten Größe der Karte bemisst. Diese Noisemap besteht aus Werten zwischen -1 und 1. Entsprechend der berechneten Werte werden anschließend Felder zugeordnet und mit der entsprechenden Textur versehen. Die möglichen Arten von Feldern sind der nachfolgenden Tabelle [Tabelle 1] zu entnehmen.

Feldbezeichnung	Symbol	Beschreibung
PLAINS - Ebene		Das Ebenenfeld ist ein leeres Feld, welches keine Rohstoffe oder besonderen Interaktionen beinhaltet. Rohstofffelder, welche komplett abgeerntet wurden, werden anschließend durch Ebenenfelder ersetzt. Der Anteil an Ebenenfelder in der Weltkarte beträgt etwa 21%.
METAL - Erzvorkommen		Erzvorkommen können von Bergbaudrohnen abgeerntet werden, um Metall zu erhalten und sind für den Bau von neuen Drohnen essentiell. Sie stellen 2% der vorhandenen Ebenenfelder.
OIL - Ölvorkommen		Das Ölvorkommen ist das für den Spieler am schwersten zu erreichende Feld. Nicht nur, da die erste Ölbohr-Drohne erst nach Abschluss des Tutorials gebaut werden kann, sondern auch, da das Ölfeld als Wasserfeld zählt und deshalb nur betreten werden kann, wenn die jeweilige Drohne ein Schwebemodul ausgerüstet hat. Im Ölfeld kann eine Ölbohr-Drohne Öl fördern, welches für die komplexeren Funktionen der Drohne benötigt wird. Ölfelder stellen etwa 15% der Wasserfelder und somit knapp 4% der Weltkarte dar.

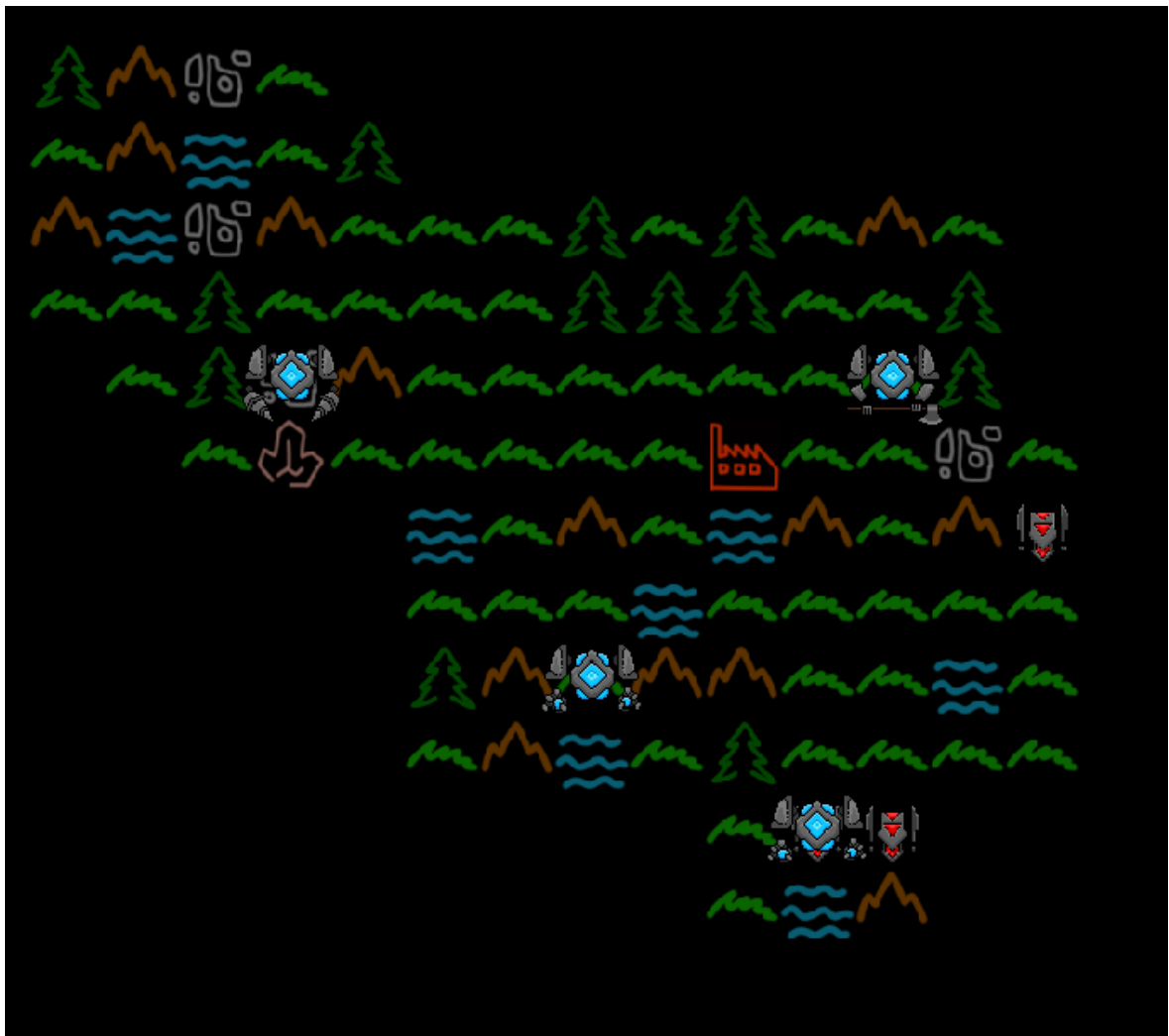
FOREST - Biomasse- Vorkommen		Das Wald-Feld stellt nach den Terrainfeldern Ebene, Wasser und Gebirge den größten Anteil der Karte dar und ist der reichhaltigste Rohstoff, welcher gefördert werden kann. Im Waldfeld lässt sich Biomasse fördern, welche die Grundlage für nahezu alle Funktionen der Drohnen ist, da sie als Treibstoff verbrannt werden kann. Wald-Felder bedecken etwa 17% der Weltkarte.
WRECKAGE - Schrottüberreste		Schrottüberreste sind eine Art von Überresten, die die verlorene Zivilisation von Sol-3 zurückgelassen haben. Diese Teile von alten Robotern können wiederverwendet werden, um neue Drohnen zu erschaffen und sind dementsprechend einer der wichtigsten Rohstoffe. Diese Felder decken etwa 2% der Weltkarte ab.
WATER - Wasser		Das Wasserfeld ist eines der Hindernisfelder, welche die Drohnen des Spielers umgehen müssen. Um dieses Hindernis zu überwinden, muss der Spieler seine Drohnen mit Schwebemodulen ausrüsten. Im späteren Verlauf können Wasserfelder genutzt werden, um permanent Biomasse zu fördern (hierzu mehr unter „9. Geplante Features“).
MOUNTAIN - Gebirge		Das Gebirgsfeld ist das zweite Hindernisfeld. Ebenso wie das Wasserfeld, kann auch das Gebirgsfeld nur mithilfe des Schwebemoduls überwunden werden. Auch das Gebirgsfeld kann später durch eine freigeschaltete Funktion genutzt werden. Hier lässt sich Erz fördern. (hierzu mehr unter „9. Geplante Features“).

FACTORY - Fabrik		Die Fabrik zeigt den Mittelpunkt der Spielkarte an. An diesem Ort startet jede neue Drohne und hier müssen die Arbeiterdrohnen ihre Rohstoffe abladen, bevor sie erneut in die Außenwelt vorstoßen.
ENEMY - Feind		Das Feind-Feld impliziert eine Begegnung mit einem feindlich gesinnten Roboter. Sollte eine der Erkundungs-/Abbaudrohnen auf ein Feindfeld laufen, so muss sie abhängig von ihrer Kampfeinstellung gegen diesen Roboter kämpfen oder fliehen. Sollte der Roboter besiegt werden, so wird das Feld in ein normales Ebenenfeld umgewandelt und der Spieler erhält eine Menge an Rohstoffen, die dem zufällig generierten Level des Feindes entspricht
WORLDBORDER - Rand der Karte		Dieses Feld zeigt den Rand der Karte an. Seine einzige Funktion ist, dass eine Drohne dieses Feld nicht betreten kann und stattdessen nach einem anderen freien Feld sucht.

[Tabelle 1, Arten von Kartenfeldern]

#### Wie orientieren sich Drohnen?



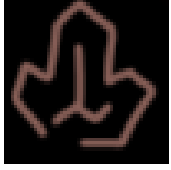

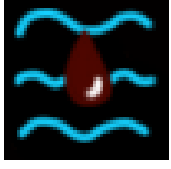

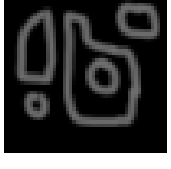

Die Drohnen des Spielers orientieren sich auf der Karte [siehe Abbildung 4], indem sie zufällig eine neue Laufrichtung zugeordnet bekommen, oder mithilfe des später freigeschalteten „improved wayfinding“ prüfen, ob eines der gescannten Felder den Rohstoff trägt, welchen die Drohne abbauen möchte. Nachdem die Laufrichtung berechnet wurde, bewegt sich die Drohne auf das neue Feld und startet eine Interaktion, bei der sie sich die Information des Feldes aus einem Hilfs-Array zieht und auswertet. Je nach Feld wird somit beispielsweise ein Kampf gegen einen Roboter ausgelöst oder die Drohne startet den Abbau der ihr zugeordneten Ressource. Sollte keine Interaktion ausgelöst werden, wird im nächsten Schritt ein neues Feld gesucht, auf welches sich die Drohne bewegt.



[Abbildung 4, Weltkarte mit aufgedeckten Feldern und Drohnen]

### Drohnen bauen und Funktionen freischalten

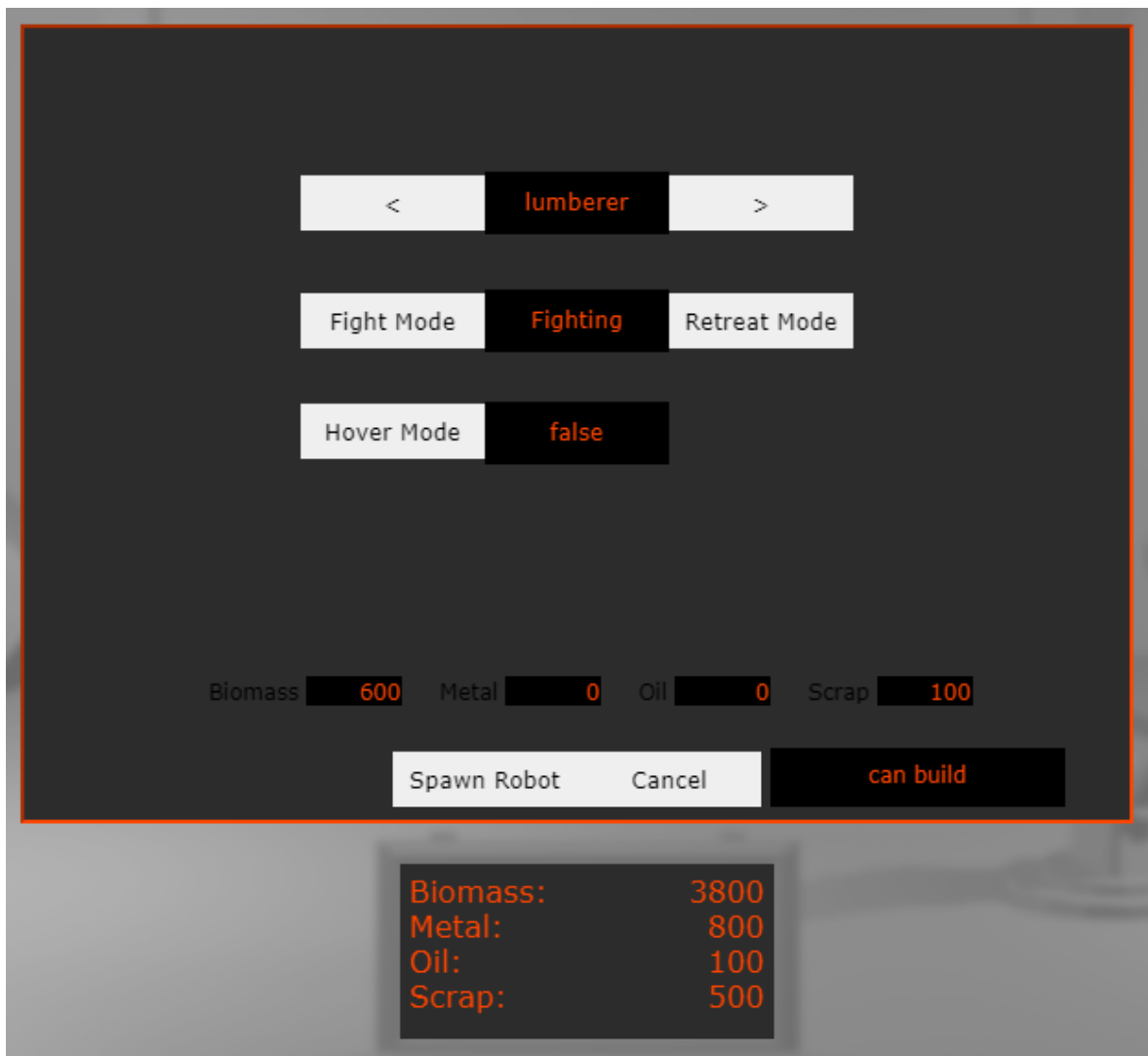
Der Spieler hat die Möglichkeit, neue Drohnen zu fertigen und mit diesen die Außenwelt zu erforschen. Hierzu benötigt er eine Kombination von Rohstoffen, welche durch die Drohnen gesammelt werden können. Die Rohstoffe, um die es sich hierbei handelt, sind folgende [Tabelle 2]:

Name	Symbol	Abbaudrohne	Beschreibung
biomass - Biomasse			Biomasse wird auf Waldfeldern gesammelt. Sie dient als Brennmaterial und Treibstoff für die Drohnen des Spielers
metal - Metall			Metall ist der Grundrohstoff, aus welchem alle Drohnen gebaut werden können. Je nach verwendetem Abbauwerkzeug sind größere oder geringere Mengen notwendig
oil - Öl			Öl dient bei den gebauten Drohnen als Schmiermittel. Es ist für eine reibungslose Bedienbarkeit der Drohnen notwendig und dient zu Beginn als Hürde, die es für den Spieler zu überwinden gilt, da Ölbohr-Drohnen teuer in der Produktion sind.
scrap - Schrott			Schrottreste sind ein weiterer Rohstoff, welcher für neue Drohnen verwendet werden kann. Es handelt sich hierbei um Überreste von alten Robotern, die wiederverwendet werden können. Außerdem können Schrottreste von zerstörten Feinden geborgen werden.

[Tabelle 2, Arten von Arbeiterdrohnen]

Bei der Erstellung werden verschiedene Werte wie beispielsweise das gewünschte Erntemodul gewählt [siehe Abbildung 5]. Dieses entscheidet darüber, welche Rohstoffe die Drohne ernten kann und wieviel Schaden sie feindlichen Robotern zufügt. Ebenso wird hier der Kampfmodus, sowie die Art der möglichen Bewegung eingestellt. Das dazugehörige Menü zeigt auch an, wie kostenintensiv diese Drohne wird. Verschiedene Funktionen werden erst im Laufe des Spiels möglich sein, da zu Beginn des Spiels entweder die entsprechenden Rohstoffe noch nicht erhältlich sind oder die Funktion erst in einem späteren Kapitel freigeschaltet wird.





[Abbildung 5, Menü zur Erstellung neuer Drohnen]

### Wie kämpfen Drohnen?

Sollte eine der Drohnen ein Feld betreten, auf welchem sich ein Feind befindet, so wird ein Kampf ausgelöst. Das Level des Gegners wird hierbei abhängig von seiner Entfernung zum Mittelpunkt der Karte und Startpunkt der Drohnen berechnet. Anschließend prüft die Drohne, ob die eigenen Kampfeinstellungen auf Rückzug gestellt sind. Sollte dies zutreffen, so erhält sie nur einmalig Schaden von dem jeweiligen Feind und verlässt anschließend das Feld. Sollte der Kampfmodus nicht auf Rückzug stehen, so sendet die Drohne eine Meldung an den Spieler, dass sie im Begriff ist, gegen einen feindlichen Roboter zu kämpfen. Der Spieler hat nun ein kurzes Zeitfenster, in dem er der Drohne befehlen kann, zurück zur Fabrik zu kommen. Sollte dieser Befehl ausbleiben, so startet der Kampf. In einer WHILE-Schleife fügen sich die Drohne und der feindliche Roboter abwechselnd ihren jeweiligen Waffenschaden zu [siehe Drohnen bauen & Funktionen freischalten, S.23]. Sobald die Lebenspunkte eines der beiden Kampfteilnehmer auf 0 sinken,

endet der Kampf. War die eigene Drohne unterlegen, so wird diese zerstört und aus der Liste der verfügbaren Arbeiter entfernt. War der feindliche Roboter unterlegen, so erhält der Spieler einige Rohstoffe als Belohnung und das Feld wird auf der Weltkarte durch eine Ebene ersetzt.

#### 4.2.2. Die Lerninhalte

Neben dem strategischen Teil des Spiels bilden die Lerninhalte die zweite Hälfte. Diese umfassen die Geschichtserzählung, sowie die Lernkapitel und Lernaufgaben. Hier hat der Spieler die Möglichkeit, die Geschichte des Spiels zu verfolgen und wieder aufzurufen, neue Lernthemen zu verinnerlichen und anschließend sein Wissen zu testen. Die Lerninhalte lassen sich in die folgenden Bereiche unterteilen: Geschichtserzählung, Rätsel, Quests und das Archiv.

##### Die Geschichte

Zu Beginn des Spiels erfährt der Spieler in einem Prolog das Setting der Geschichte. In leicht illustrierten Fenstern wird er über die vorangegangene und aktuelle Situation informiert und kann sich so ein Bild der Geschehnisse machen, die zu dem Punkt geführt haben, an dem er sich gerade befindet. Nach dem Prolog beginnt jedes Kapitel [siehe „Die Quests“] damit, die Geschichte weiter zu erzählen und den Spieler immer tiefer in die Geschehnisse eintauchen zu lassen. Abschließend wird die Geschichte mit einem Epilog beendet, der beschreibt, wie es dem Spieler gelingt, sich selbst zu retten oder auch auf dem Planeten zu verbleiben, je nachdem, wie er sich entscheidet. Die Inhalte der Geschichte werden von der Klasse StoryHandler.ts gesteuert, welche neue Inhalte berechnet und in das Archiv einpflegt, sobald diese freigeschaltet wurden. Sie ist auch dafür zuständig, die verschiedenen Geschichtsteile aus der zugehörigen JSON-Datei zu laden und umzuwandeln.

##### Die Quests

Die Quests sind Kapitel im Spiel, die sich jeweils einem neuen Lernthema zuwenden. Zu Beginn eines neuen Kapitels kann der Spieler hier einsehen, was er tun muss, um die Quest abzuschließen und in ein neues Thema voranzuschreiten. Die Anforderungen können sich von der Sammlung von Rohstoffen über den Bau von Drohnen bis zum Abschluss einer bestimmten Anzahl Rätsel erstrecken. Auch eine Kombination dieser Faktoren ist möglich. In der Quest selbst kann der Spieler

ebenfalls das aktuell bereitgestellte Lernmaterial [siehe Abbildung 6] einsehen, welches es in diesem Kapitel zu bearbeiten gilt. Sollten alle Anforderungen erfüllt sein, so kann über eine Schaltfläche die aktuelle Quest beendet werden. Die Inhalte der Quest werden anschließend archiviert. Für die Quest wurde eine Klasse namens QuestHandler.ts erstellt, welche alle Funktionen steuert. Diese Klasse generiert anhand einer zugehörigen JSON-Datei nach Abschluss eines Kapitels das darauffolgende und ordnet sämtliche Inhalte zu. Ebenso steuert diese Klasse das Voranschreiten der Geschichte und schaltet neue Funktionen frei.

close image

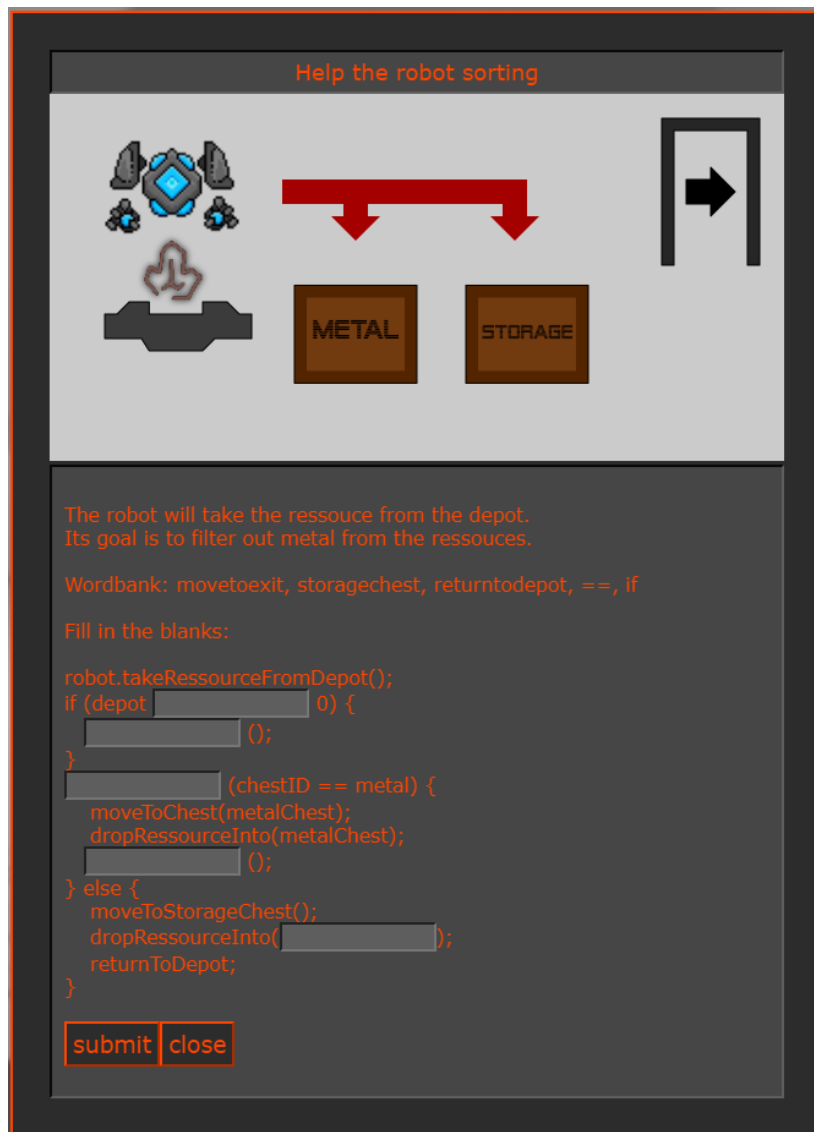
Description	Example	Output												
<p><b>If-Condition:</b> If statements check, if a condition is true or false. If the statement is true, the surrounded code will be executed.</p>	<pre>number = 4; if (number &gt; 2) {     print "Number is bigger than 2"; }</pre>	<p>Output of the condition in Terminal will be "Number is bigger than 2" The condition checks, if the given variable in this case is bigger than the number 2.</p>												
<p>An If-Else condition checks, if a condition is true or false, then afterwards, if the result didn't match the condition, goes on and executes the else-condition. This way, if the given variable doesn't match, the code can go on with another output.</p>	<pre>Robot.color = "red"; if (Robot.color == "white") {     print Robot.color; } else {     print "Robot is not white"; }</pre>	<p>Output of the condition in Terminal will be " Robot is not white". The condition checks the color of the robot, realising it is red and therefore resolves the if statement as false. Afterwards it checks for an else-condition. In this case, the program resolves the code in the else-statement and prints the text.</p>												
<p>After an else, there can be another if-condition.</p>	<pre>RobotOne.color = "red"; RobotTwo.color = "blue";  Are there green Robots?  if (RobotOne.color == "green") {     print "yes"; } else if (RobotTwo.color == "green") {     print "yes"; } else {     print "no"; }</pre>													
<p><b>Logic symbols:</b></p> <table><tr><td>==</td><td>equals</td></tr><tr><td>&lt; or &lt;=</td><td>lesser than / lesser than or equal to</td></tr><tr><td>&gt; or &gt;=</td><td>greater than / greater than or equal to</td></tr><tr><td>&amp;&amp;</td><td>and</td></tr><tr><td>  </td><td>or</td></tr><tr><td>!=</td><td>not</td></tr></table>	==	equals	< or <=	lesser than / lesser than or equal to	> or >=	greater than / greater than or equal to	&&	and		or	!=	not		
==	equals													
< or <=	lesser than / lesser than or equal to													
> or >=	greater than / greater than or equal to													
&&	and													
	or													
!=	not													

[Abbildung 6, Lernmaterial zum Thema If-Bedingungen]

## Die Rätsel

Um das gelernte Wissen abzufragen, kann der Spieler verschiedene Lernaufgaben und Rätsel bearbeiten, die auf unterschiedliche Art und Weise zu lösen sind. Während die Komplexität der Rätsel in den leichteren Themen noch sehr einsteigerfreundlich gestaltet ist, so soll deren Schwierigkeit entsprechend dem Fortschritt im Spiel ansteigen und verschiedene Lernthemen miteinander verbinden. Der Aufbau dieser Rätsel [siehe Abbildung 7] ist zu Beginn hauptsächlich durch Lückentexte und Quizaufgaben geprägt und soll durch Aufgaben ergänzt werden, bei denen beispielsweise Codeteile richtig aneinandergereiht werden müssen. Diese Rätsel orientieren sich thematisch sowohl an dem jeweiligen Lernkapitel als auch an dem Themenfeld, welches in dem Spiel generell bearbeitet wird. Der

Aufbau der Rätsel ist zu einem gewissen Grad modular gestaltet. Die Anzahl der zu füllenden Felder orientiert sich anhand der vom Entwickler gegebenen Antwortmöglichkeiten und erweitert anhand dessen die Aufgaben. Erstellt werden diese Rätsel in Form einer JSON-Datei, welche definiert, wie viele Aufgaben sich innerhalb eines Kapitels befinden und ob es sich um die finale Aufgabe eines Kapitels handelt. Ebenso besteht anhand dieser Datei die Möglichkeit, ein Bild anzufügen, welches das Rätsel beschreibt oder untermalt.



[Abbildung 7, Rätsel zum Thema If-Bedingungen]

## Das Archiv

Im Archiv werden alle bereits erhaltenen Geschichtsinformationen und die bearbeiteten Lernthemen gespeichert. Der Spieler kann hier nachschlagen, wenn er etwas bezüglich der Geschichte oder eines vorangegangenen Themas vergessen hat und wiederholen möchte. Die Inhalte des Archivs werden mithilfe der Klassen QuestHandler.ts und StoryHandler.ts gepflegt.

Während jeder dieser beiden Hauptteile des Spiels austauschbar ist, so sollten die Hälften thematisch aufeinander abgestimmt sein, um ein immersives und harmonisches Gesamtbild zu erzeugen. Ebenso sollten nicht beide Hälften das gleiche Ziel verfolgen und beispielsweise Lerninhalte vermitteln, um die spielerische Vielfalt zu erhalten.

## 7.3. Grafischer Aufbau

Als Optik wurde für dieses Projekt eine Mischung aus 3D-Grafik und Piktogrammen genutzt. Die Menüs sind in verschiedenen Grautönen mit orangeroten Highlights gestaltet und nutzen eine gut lesbare Schrift. Das Spielmenü stellt den Innenraum der Fabrik dar [siehe Abbildung 8], in welcher sich die Spielfigur aufhält. Es ist aus Sicht ebendieser gestaltet und stellt den Blick auf die verschiedenen Gerätschaften dar. Es wurde mit der 3D-Software Blender gestaltet und gerendert. Die Buttons, um auf die unterschiedlichen Untermenüs des Spiels zuzugreifen, stellen einen optischen Bezug auf ihre Verwendung dar. Sie sind zusätzlich mit einem Leuchteffekt ausgestattet, falls man mit der Maus über diese fährt, um besser zu indizieren, welcher Button gedrückt wird. Zusätzlich wird jeder dieser Buttons später mit einem optisch ansprechend gestalteten Text versehen, um die Lesbarkeit und Verständlichkeit weiter zu erhöhen.

Die Weltkarte, auf welcher der Spieler die Drohnen verfolgen kann, ist in einer vereinfachten Grafik gehalten und nutzt Piktogramme, die symbolisch darstellen, welche Felder sich in der Nähe der Drohnen befinden. Diese Optik wurde gewählt, um zu untermalen, dass es sich hierbei nicht um eine direkte Darstellung der Umwelt handelt. Da der Spieler nicht direkt mit dieser interagieren soll, war die Darstellung in vereinfachter Form immersiver. Es wurden zwischenzeitlich Kacheln im Pixel-Art-Stil konzipiert und umgesetzt, allerdings war diese Optik zu nah an einem direkten

Blick in die Außenwelt und wurden deshalb wieder durch die Piktogramme, welche ursprünglich als Platzhalter gedacht waren, ersetzt.

Die Rätsel und Menüs sind wie bereits erwähnt in schlichten Farben mit gut sichtbaren Details gehalten, um die Lesbarkeit zu garantieren und die Verwendung einfach und übersichtlich zu halten. Die Buttons nutzen Soundeffekte, um ein Feedback zu geben, wenn diese betätigt werden. Ebenso sind die meisten dieser Buttons mit einem Leuchteffekt versehen, wenn man darüberfährt. Das Design des Spiels soll einen futuristischen Eindruck vermitteln, der dem Setting des Spiels, der Geschichte und des Lernthemas angepasst ist.



[Abbildung 8, Spielmenü, Sicht des Spielers]

## 7.4. Erzählung

### 7.4.1. Warum eine Geschichtserzählung in Serious Games?

Die Frage des Nutzens einer Geschichte für ein Serious Game ist eine der zentralen Fragen, die in dieser Arbeit beantwortet werden sollen. Im Anschluss wird erläutert, welche Wirkung die Geschichtserzählung auf den Spieler hat und weshalb sie für das Lernen von neuen Themen hilfreich sein kann [7].

Für den Menschen ist es deutlich einfacher, sich Informationen einzuprägen, wenn diese mit einem Impuls emotionaler Natur verknüpft sind. Indem der Körper Dopamin ausschüttet, werden mehr Synapsen aktiviert, wodurch die gelernten Informationen schneller und langanhaltender abgespeichert werden können [8].

Ebenso werden durch eine Geschichte die Aufmerksamkeit und Konzentration des Spielers gefördert, indem die Lernmotivation durch den Aufbau von Spannung erhöht wird. Somit hilft die Geschichtserzählung bei der Generierung von Erinnerungen.

Durch die Immersion innerhalb der Spielwelt soll der intrinsische Spieltrieb des Menschen dazu beitragen, sich in die Geschehnisse des Spiels zu versetzen und die Hindernisse zu bewältigen.

Mit verschiedenen Narrationstechniken kann der Spieler somit durch das Spiel geleitet werden und bekommt die jeweiligen Lerninhalte so vermittelt, dass sie sich als Teil der Geschichte anfühlen. Für dieses Projekt bewältigt der Spieler eine Heldenreise [9], welche ihn über mehrere Akte vom Prolog bis hin zu der Auflösung der Geschichte leitet. Die Erzählung folgt dabei der gleichnamigen Narrationstechnik „Heldenreise“, welche von Joseph Campbell erstmals erkannt und von Christopher Vogler verfeinert wurde. Bei der Heldenreise absolviert der Spielercharakter 12 Stationen, die ihm eine Identität geben und ihn prüfen sollen. Innerhalb dieser Heldenreise wird die Spielwelt dem Spieler nähergebracht und er wird damit konfrontiert, dass sich diese Welt ändert, indem der Spieler aus seiner Komfortzone gerissen wird. Innerhalb der Heldenreise gibt es Schwellen, welche überwunden werden müssen und welche eine dauerhafte Auswirkung auf das Spiel haben werden. In diesem Projekt wird dies innerhalb der Geschichte durch das Betreten des Landungsschiffes und die unmögliche Rückkehr zur eigenen Flotte symbolisiert. Der Spielercharakter muss sich Bewährungsproben stellen und auf einen finalen Showdown hinarbeiten, in dem sich entscheidet, ob er erfolgreich ist, oder scheitert. Zum Ende der Geschichte hin erhält er seine Belohnung, in diesem Fall die Rettung oder Abreise von dem Planeten, auf welchem er gestrandet war und beginnt die Rückkehr in sein ehemaliges Umfeld. Der Spieler erlebt, wie der Spielercharakter wieder in den Alltag zurückkehrt und mit den Erfahrungen seiner eigenen und der neu erforschten Welt umgeht. Die Belohnung und Rückkehr in den Alltag wird in diesem Spiel durch den Epilog erzählt, in welchem der Spielercharakter wichtige Informationen über die Abstammung seines Volkes sammeln konnte und nun hilft, die Archive seines Planeten neu aufzubereiten. Die Erzählung dieser Geschichte und die Vernetzung mit den Lerninhalten soll dem Spieler stets die Möglichkeit geben, positive Emotionen zu erfahren, die ihn beim Lernen unterstützen, um in der Geschichte des Spiels voranzuschreiten.

### 7.4.2. Geschichte

Die Geschichte des Spiels dreht sich um das Mitglied einer Forschungsexpedition, welche von einer Kolonie auf dem Planeten Kepler aus losgesendet wird, um die umliegenden Sonnensysteme zu erkunden. Zu diesem Zeitpunkt ist den Bewohnern von Kepler nichts über ihre Abstammung bekannt. Zu einem späteren Zeitpunkt in der Geschichte wird der Spieler herausfinden, dass die Bewohner von Kepler Flüchtlinge des Planeten Erde, in diesem Spiel als Sol-3 bekannt, sind, welche einer Katastrophe entflohen sind, die dazu geführt hat, dass die Zivilisation der Erde zusammenbrach und die Atmosphäre für Menschen toxisch wurde. Während der Expedition kommt es zu einem Unglück, bei welchem das Raumschiff der Expedition zerstört wird und im Zuge dessen auch das Landungsschiff der Spielfigur beschädigt, wodurch diese auf der Erde strandet und in einer alten Fabrik eingeschlossen ist. Eine der Erkundungsdrohnen der Expedition ist unbeschädigt und dient als einziges Mittel, die Außenwelt zu begehen. Bei der Untersuchung der Gerätschaften in der Fabrik stellt der Spieler fest, dass die Roboter des Planeten eine Programmiersprache nutzen, welche der Spieler dechiffrieren kann. Im weiteren Verlauf der Geschichte werden so weitere Funktionen und Hinweise auf die Bevölkerung des Planeten und die Roboter, welche immer noch tätig sind, freigeschaltet. Der Spieler erfährt, was der Auslöser für die Katastrophe des Planeten und die damit verbundene Vergiftung der Atmosphäre war und wie er eine Möglichkeit finden kann, seine Flotte zu kontaktieren und dem Planeten zu entfliehen.

Diese Geschichte wurde gewählt, da sie sich mit dem Lernthema Programmieren sehr gut vereinbaren lässt. Sie bietet Kapazität, um die Programmerrätsel und das Lernen der Sprache zu erklären. Die Roboter und Drohnen lassen sich gut in die verschiedenen Rätsel implementieren und mit diesen können beispielsweise Funktionen wie das Sortieren von Rohstoffen erklärt werden. Auch das Mysterium um die Geschichte der Zivilisation von Kepler und der Erde steigert den Erkundungsdrang und die Motivation.



## 7.5. Probleme bei der Umsetzung

Während der Umsetzung des Projektes wurden verschiedene Inhalte mehrfach überarbeitet, da sie den Spielfluss gestört haben oder als nicht zielführend bewertet wurden.

Zu Beginn hatte der Spieler einen eigenen Spielercharakter, welcher sich innerhalb der Fabrik, die als primäre Oberfläche des Spiels dient, bewegen konnte. Die Spielfigur hatte dabei eine Lauffunktion inklusive Kollisionsabfrage mit den verschiedenen Elementen der Umgebung und sollte mit den Konsolen und Maschinen interagieren können, über welche die weiteren Inhalte des Spiels gesteuert werden. Die Implementierung dieser Funktion hätte allerdings keinen spielerischen Mehrwert zur Folge gehabt, da die Interaktionsmöglichkeiten durch die Geschichte des Spiels beschränkt waren und es für die Spielfigur keine Hindernisse zu bewältigen gegeben hätte, die eine Bewegungssteuerung erfordern. Bei der Gestaltung des strategischen Anteils des Spiels musste ein passendes System der Weltgenerierung und Speicherung der Spielwelt gefunden werden. Da der Rechenaufwand für eine komplett geladene Welt sehr hoch war, wurde ein Sichtbereich für die Drohnen des Spielers eingeführt, der die Spielkarte Stück für Stück aufdeckt und nur die Kacheln auffrischt, mit welchen Interaktionen stattgefunden haben. Somit ließ sich die Rechenleistung massiv verringern und durch die verdeckten Gebiete gibt es nun auch noch unvorhersehbare Ereignisse für den Spieler.

Eine weitere Funktion, welche für den Prototyp gestrichen wurde, war die Möglichkeit, den bisherigen Fortschritt zu speichern. Während dies für das Endprodukt eine durchaus sinnvolle Ergänzung darstellt, ist dies für den Prototypen aufgrund seiner Dauer nicht von Nöten.

Abschließend gestaltete sich die Erstellung der Menüs und nötigen Buttons dahingehend problematisch, da die meisten Fenster durch Code mithilfe von TypeScript erstellt werden und variable Ausmaße haben. Durch die außergewöhnliche Form der Knöpfe im Spielmenü war es nicht möglich, einen Mouse-Hover-Effekt anhand von CSS zu implementieren, da dieser Effekt nur die Ränder der Bilddatei, nicht aber den durchsichtigen Hintergrund der .png-Datei berücksichtigt. Deshalb wurde der Effekt mithilfe von Affinity Photo erstellt, was demnach erforderte, diese Buttons, welche durch den Hover-Effekt neue Außenmaße hatten, pixelgenau auszurichten, damit sie nicht flackern oder sich

verschieben, wenn man die Maus darüber zieht. Auch die generierten Fenster mussten so ausgerichtet werden, dass sie trotz variablem Inhalt im sichtbaren Bereich des Spiels verbleiben konnten. Für dieses Problem wäre eventuell die Nutzung von CSS-columns sinnvoll gewesen und könnte in einem zukünftigen Prototyp angewandt werden.

## 8. Testphase mit gemischter Nutzergruppe

Um den entwickelten Prototyp zu testen, wurde eine Gruppe von Nutzern zusammengestellt, welche über unterschiedliche Stufen von Programmiererfahrung verfügten. Die Nutzer lassen sich in eine Altersgruppe von 20-35 einordnen, welche für diese Arbeit als relevant für das Erlernen von Programmierkenntnissen angesehen wird. Zahlenmäßig umfasste die Testgruppe 15 Personen. Durchgeführt wurde die Testphase, indem den Nutzern der Prototyp mit einer Anleitung zum Start des Spiels zur Verfügung gestellt wurde. Daraufhin sollten die Nutzer selbstständig das Spiel spielen und die gestellten Aufgaben abschließen, ohne externe Hilfe in Anspruch zu nehmen. Im Anschluss an den Test galt es, eine Reihe von Fragen zu beantworten, die Aufschluss über die Effektivität des Spiels geben sollen und feststellen, ob die Lernmotivation durch die Einbindung des Lernthemas in eine Geschichte tatsächlich gesteigert werden kann. Im Anschluss werden die Ergebnisse dieser Befragung ausgewertet und abschließend ein Fazit inklusive Lösungsansätzen für festgestellte Probleme gegeben. Am Ende jeder Frage wird eine kurze Zusammenfassung der Ergebnisse gegeben, die diese Frage für die Verbesserung des Prototyps erbringen konnte.

### 8.1. Wussten Sie, wie Sie sich im Spiel zurechtfinden können?

Den Nutzern war es grundsätzlich gut möglich, sich im Spiel zu orientieren. Durch die Erklärungen im Spiel und die Beschriftung der Buttons war die Navigation einfach und die gesuchten Funktionen ließen sich leicht finden. Einige Nutzer bemängelten ein schlechtes Feedback der gedrückten Buttons. Dies wurde anschließend umgesetzt, indem die Buttons sowohl einen Hover-Effekt, als auch ein Klick-Geräusch beim Betätigen erhalten haben. Probleme hatten einige Nutzer bei der Erstellung der ersten Weltkarte, da ihnen der Unterschied zwischen Weltgenerierung und Erstellung unklar war. Dies könnte in Zukunft besser benannt werden, um Verwirrung zu vermeiden. Zu einem gewissen Grad war es möglicherweise auch ein Problem innerhalb der Anleitung, in welcher die Weiterstellung erklärt wurde. Die Anleitung hätte also in das Spiel integriert werden sollen, anstatt sie mitsamt dem Fragebogen und dem Link vor dem Start des Spiels

zu übergeben. Auch war es einem Nutzer nicht gleich ersichtlich, das Lernmaterial des ersten Kapitels aufzuspüren, da der Button hierfür missverständlich betitelt wurde. Auch dieses Problem wurde während der Testphase bereits durch eine genauere Benennung behoben, wodurch nun kein Missverständnis mehr entstehen sollte. Ein weiteres Problem einiger Nutzer war die undeutliche Erklärung der Drohnenfunktionen auf der Roboterkarte. Dies lässt sich in Zukunft mit einer kurzen Erklärung oder Tooltips beheben und ist unter dem Kapitel „Geplante Features“ genauer aufgeführt. Ein Nutzer hatte größere Probleme mit der Navigation, da er nicht mit der Nutzung von Videospielen bewandert ist und überfordert davon war, was seine Aufgaben sind und wie er diese bewältigt. Die gegebenen Antworten der ersten Frage konnten also auf die Überarbeitung des Button-Feedbacks und eine bessere Erklärung der Funktionen des Spiels hinweisen.

## 8.2. War das bereitgestellte Lernmaterial verständlich formuliert?

Für die meisten Nutzer konnten die bereitgestellten Informationen deutlich beim Abschluss der Rätsel Hilfestellung geben. Auch für unerfahrene Spieler waren die Inhalte so strukturiert, dass sie die Funktionsweise einer If-Bedingung anhand des gegebenen Lernmaterials nachvollziehen und anwenden konnten. Ein Nutzer wünschte sich dennoch ein besser strukturiertes Anwendungsbeispiel mit ausführlicherer Erklärung. Als Lösungsansatz wäre ein ausführlicheres Anwendungsbeispiel mit einer Erklärung aller Zwischenschritte angebracht, welches auch die Verwendung von Operatoren ausführlicher beleuchtet. Grundsätzlich konnten die Testpersonen jedoch mit den gegebenen Informationen arbeiten und die Rätsel lösen, ohne abubrechen und erneut nachzusehen, welche Lösungen in den Rätseln korrekt wären.

## 8.3. Hatten Sie Probleme beim Lösen der Rätsel?

Aufgrund der anfänglichen Probleme mit der Benennung des zugehörigen Buttons fiel es einem Teil der Tester schwer, das Lernmaterial zu finden und die Rätsel zu bearbeiten. Nachdem dieses Problem behoben wurde, konnten die Rätsel, wie in 8.2. bereits erläutert, von nahezu allen Testern gelöst werden. Vor allem bei Testern, die bereits Kontakt mit Programmiersprachen hatten, traten keine Probleme auf, da die Rätsel einfach aufgebaut waren und die Inhalte

Einsteigerwissen abdecken. Die Komplexität wurde für den Beginn des Spiels und das bis zu diesem Punkt erlangte Wissen als angemessen empfunden.

## 8.4. Verspüren Sie Interesse, die Geschichte trotz der Rätsel weiter zu verfolgen?

Bei der Testgruppe bestand mit Ausnahme eines Testers großes Interesse daran, den Verlauf der Geschichte weiter zu verfolgen. Der Prolog wurde als interessanter Beginn der Geschichte wahrgenommen und unterstützte das fiktionale Setting des Spiels. Ein Nutzer hob hervor, dass er einen noch größeren Einfluss der Geschichte auf die Rätsel und den strategischen Anteil des Spiels möchte. So könnten verschiedene Funktionalitäten des Spiels durch das Fortschreiten in der Geschichte freigeschaltet werden, anstatt diese mit dem Abschluss von Rätseln zu erhalten. Auch wurde der Vorschlag unterbreitet, Teile der Geschichte und eventuelle Nebenstränge der Geschichte mithilfe der Erkundungsdrohnen zu erspielen, wenn diese beispielsweise wichtige Orte auf der Karte aufdecken. Dies würde die Möglichkeit bieten, das Erkunden der Karte noch aufregender und belohnender zu gestalten. Die Implementierung von Nebengeschichten stellt mit Sicherheit ein weiteres spannendes Element der Erzählung dar und könnte in einer späteren Version des Spiels definitiv implementiert werden. Indem auf der Weltkarte zufällig Strukturen erschaffen werden, die mit der Geschichte verknüpft sind, könnten diese Handlungsstränge sich mit der Erforschung der ehemaligen Zivilisation des Planeten oder der Erkundung von Ruinen befassen, welche auch neue Möglichkeiten bieten, verschiedene Lernthemen zu lehren, die sich nicht direkt in die Hauptgeschichte einbauen lassen. Auch das Erlangen von passiven Verbesserungen für die Drohnen des Spielers waren ein Punkt, welcher von mehreren Nutzern gewünscht wurde. Diese Boni könnten durch die oben genannte Erkundung von Strukturen der Zivilisation erspielt werden. Dies bietet auch eine interessante Grundlage, andere Rätselkonzepte einzubinden, die sich mit dieser Zivilisation befassen und nicht zwangsläufig direkt mit dem Thema Programmierung zusammenhängen. Somit würde der Lernaspekt in diesem Bereich zwar dezent in den Hintergrund rücken, wodurch sich das Interesse an dem Spiel durch solche Rätsel erhöht, da sich der Spieler nicht durchgehend mit den Lernthemen befassen muss, sondern Minispiele bearbeiten kann, bei denen es um andere Themen geht, welche keine Vorbereitung erfordern.

Ein Nutzer empfand die Geschichtserzählung als uninteressant, da ihm die Geschichte eines Spiels unwichtig ist. Dieses Problem wird bei manchen Nutzern auftreten und ist nur schwer zu beheben, da es nicht für jeden Spieler wichtig ist, sich mit der Spielwelt auseinanderzusetzen, sondern nur das Gameplay an sich zu erleben. In Anbetracht des positiven Feedbacks gegenüber der Geschichte wird die These bekräftigt, dass sich die Geschichtserzählung positiv auf das Interesse der Spieler auswirkt.

## 8.5. Wären Sie bereit, weitere Lernthemen zu verinnerlichen, um die Geschichte weiter zu erleben?

Mit Ausnahme eines Einzelfalles war jeder Nutzer bereit, weitere Lerninhalte zu bearbeiten, um in der Geschichte des Spiels voranzuschreiten, sofern der Spielfortschritt nicht ausschließlich mithilfe von Rätseln errungen werden kann. Gerade die Möglichkeit, leichtere Übungsrätsel durch Rohstoffe freizuschalten wurde als potentielle Hilfestellung genannt, um das Erlernen der Themen zu erleichtern und langsameren Lernern eine zusätzliche Hilfe zu bieten. Besonders der Wunsch nach noch mehr Geschichte wurde genannt. Dies lässt sich in zukünftigen Versionen durch eine bessere Verteilung der Geschichtsinhalte auf verschiedene Bereiche des Spiels verwirklichen und sollte ein festes Ziel für zukünftige Versionen sein. Bei erfahreneren Testern wurde der Wunsch nach komplexeren Themen laut, da die ersten Kapitel für diese Spieler sehr schnell zu bewältigen waren. Auch diese Nutzer bewerteten die gegebenen Lerninhalte jedoch als hilfreich und zeigten Interesse, weitere Themen zu bearbeiten, wenn sie dafür den Fortschritt der Geschichte verfolgen können. Auch mehrere Schwierigkeitsgrade wären denkbar, um erfahrenere Spieler trotzdem auf Dauer zu motivieren. Die Testgruppe bestätigte bei dieser Frage erneut, dass das Interesse an den Lerninhalten des Spiels durch die Geschichtserzählung verstärkt wird, beziehungsweise erhalten bleibt.

## 8.6. Fazit

Während der Durchführung der Testphase konnte ein durchaus positives Feedback erzielt werden. Den meisten Nutzern war es möglich, den Prototyp in vollem Umfang abzuschließen und die erlernten Kenntnisse anzuwenden. Das Interesse an der Geschichte konnte laut Aussage der Spieler merklich zu der Lernmotivation beitragen und war förderlich. Es konnten jedoch auch Problematiken aufgezeigt werden, welche das Thema Geschichte in Serious Games beeinflussen. Ein Teil der Nutzer hatte aufgrund durchwachsender Englischkenntnisse leichte Probleme, der Geschichte zu folgen und die Aufgabenstellungen zu verstehen. Der Wunsch einer deutschen Übersetzung wurde mehrfach genannt. Dies lässt sich für die Zukunft definitiv bewerkstelligen, hatte für die Fertigung des Prototyps jedoch eine niedrige Priorität, da Englisch als Sprache gewählt wurde und ein Großteil der Nutzer ausreichende Englischkenntnisse vorweisen konnte, um problemlos das Spiel abzuschließen.

Ein Nutzer hatte sehr große Probleme beim Verständnis der Aufgaben und dem Abschluss des Prototyps. In einem Gespräch ließ sich herausarbeiten, dass dies auf eine Dyskalkulie zurückzuführen ist. Bei einer Dyskalkulie handelt es sich um eine Beeinträchtigung der Rechenfähigkeiten, welche nicht aufgrund einer Intelligenzminderung oder unangemessenen Beschulung entsteht. Der Nutzer hat dadurch große Probleme, logische Schlussfolgerungen zu ziehen und anzuwenden [10]. Es handelt sich um eine von der Weltgesundheitsorganisation WHO anerkannte Benachteiligung. Sie zeichnet sich beispielsweise durch Probleme mit grundlegenden Rechenarten aus. Dies demotivierte den Nutzer bei der Bearbeitung der Aufgaben ungemein, da die Anwendung von logischen Operatoren und das Verstehen von Funktionen nicht möglich war. Aus dieser Erkenntnis lässt sich ableiten, dass nicht jeder Nutzer mit der Handhabung aufgrund der Umsetzung des Spieles, sondern auch aufgrund externer Faktoren auf Probleme stößt. Für diese Nutzergruppe ist das Themenfeld Programmierung unverständlich und deshalb nicht relevant.

Ein weiteres Problem zeigte sich durch Nutzer, die nicht motiviert waren, sich mit dem Spiel zu befassen und die Inhalte zu verinnerlichen. Ein geringes Maß an Interesse ist für ein Lernspiel sehr hinderlich, welches darauf setzt, dass der Nutzer Informationen erlernt und anwendet. Diese Nutzer zu motivieren, stellt sich als große Hürde dar, da sie von Beginn an kein Interesse zeigen, das Spiel zu

verstehen und zu lösen. Nicht jeder Spieler befürwortet das Konzept eines Lernspiels. Dies kann verschiedene Gründe haben. Manche Spieler sehen Videospiele nur als Freizeitaktivität und wollen keine neuen Lernthemen bearbeiten, sondern sich eher mit entspannenden Themen befassen. Ebenso ist nicht jeder daran interessiert, das Themenfeld der Programmierung zu erkunden. Auch gibt es Nutzer, welchen Videospiele im Allgemeinen nicht geläufig sind. Der Umgang mit einem Spiel ist in diesem Fall aufgrund fehlender Vorerfahrung erschwert, da sich der Nutzer erst in das Konzept des Videospieles an sich einarbeiten muss und der Lerninhalt, beziehungsweise die erforderliche Lernmotivation den Nutzer hindert, dieses Konzept zu verinnerlichen.

Auch zeigt nicht jeder Mensch Interesse daran, die Geschichte eines Spiels zu verfolgen. Da dieses Projekt massiv darauf aufgebaut ist, die Geschichtserzählung als treibende Motivation zu nutzen, ist es für oben genannte Nutzer nicht geeignet. In Betrachtung dieser Problematiken in Abwägung mit dem positiven Feedback der Nutzer, welche den Prototyp abschließen konnten, lässt sich dennoch feststellen, dass das Konzept der Einbindung der Lerninhalte in eine Geschichte funktioniert und die gewünschte Wirkung der gesteigerten Lernmotivation erfüllt werden kann, sofern der Nutzer das nötige Interesse mitbringt, sich mit den gestellten Aufgaben auseinander zu setzen und etwas Neues lernen zu wollen. Die Implementierung der Lerninhalte in die Narration des Spiels wirkt sich förderlich auf die Lernmotivation aus und kann den Nutzer dazu bringen, Lernthemen, welche er unter anderen Umständen umgehen würde, zu bearbeiten, um die Geschichte weiter verfolgen zu können.



## 9. Geplante Features

Während der Arbeit an dem Projekt und der Durchführung der dazugehörigen Testphase ergaben sich mehrere Features, welche das Spiel und die Lernerfahrung verbessern und erweitern würden. Dieses Kapitel umfasst sowohl eine geplante Weiterentwicklung, als auch optionale Erweiterungen, die als sinnvoll erachtet werden und das Spielerlebnis fördern.

### 9.1. Erweiterung der Lernthemen und Rätsel

Als oberste Priorität ist die Erweiterung der Lernthemen des Hauptspiels zu sehen. Es sollen Kapitel und Rätsel erstellt werden, welche weitere Bereiche des Oberthemas Programmierung abdecken und lehren. So soll entsprechend des Fortschritts im Spiel die Komplexität der Themen erhöht werden und Bereiche wie zum Beispiel Schleifen, Klassen und sonstige essentielle Grundlagen der Programmierung abdecken. Es ist hierbei erforderlich, diese Lerninhalte zu strukturieren, um den Spielfluss interessant zu halten und nicht zu früh zu schwere Aufgaben zu erteilen.

Dies sollte in Verbindung mit neuen Rätselkonzepten und im Einklang mit der Geschichte des Spiels geschehen. Um einige dieser Rätselkonzepte zu erläutern, werden im Anschluss die Rätsel aufgeführt, welche während der Entwicklung konzipiert und aufgeschoben wurden.

Eine Möglichkeit wäre, einen Roboter mit einer Reihe von Befehlen auszustatten und diesen durch ein Labyrinth oder einen Gebäudekomplex manövrieren zu lassen, ohne währenddessen einzugreifen. Aus einer Liste von Bausteinen kann diese Abfolge von Befehlen zusammengestellt werden, bevor der Roboter entsendet wird.

Eine weitere Option wäre ein Code-Editor basierend auf einer vereinfachten Programmiersprache, in welchem Funktionen direkt selbst gestaltet werden können. Die Nutzung einer bestehenden Programmiersprache stellt hier die Gefahr dar, dass im Thema Programmierung bewanderte Spieler durch bestimmte Kombinationen von Zeichen auf den Quellcode des Spiels zugreifen und diesen verändern könnten. Die Nutzung einer eigenen Programmiersprache und eines entsprechenden Compilers senkt dieses Risiko und gibt dem Entwickler eine bessere Möglichkeit, diese Sprache in die Geschichte zu implementieren.

Ein Code-Editor mit Bausteinen im Stil von Scratch bietet sich dafür an, komplexere Strukturen, beispielsweise für die Automatisierung von Drohnen, zu gestalten und das bisherige, relativ einfache System der Roboter-Karte zu ersetzen, über welches die Drohnen entsendet, zerlegt, zurückgerufen und automatisch erneut losgeschickt werden können. Dieses System könnte auch während dem Spiel durch den Abschluss eines Kapitels implementiert werden, um dem Spieler einen sichtbaren Fortschritt in der Bedienung der Drohnen zu gewähren. In einem separaten Editor könnten Kombinationen von Befehlen getestet und gespeichert werden, bevor sie während der Erstellung neuer Drohnen implementiert werden. Diese Funktionen könnten der Übersicht halber über das Archiv eingesehen und eventuell wieder entfernt werden.

## 9.2. Verbesserte Visualisierung für Kämpfe

Das Kampfsystem der Drohnen und feindlichen Roboter sollte auch überarbeitet und besser visualisiert werden. Im Moment geschieht dies für den Spieler unsichtbar im Hintergrund und berechnet sich aus dem Schaden der ausgerüsteten Abbau-Module der Drohne und der Entfernung der Feinde zum Mittelpunkt der Karte beziehungsweise der Basis des Spielercharakters. Eine Möglichkeit wäre, den Kampf rundenbasiert in einem separaten Fenster zu gestalten und durch verschiedene Kommandos in den Kampf einzugreifen, um der eigenen Drohne einen Vorteil bei der Vernichtung des feindlichen Roboters zu gewähren. Hierzu würde ein Kampffenster generiert werden, welches beide Kampfteilnehmer darstellt und Animationen der Angriffe und Fähigkeiten zeigt, um dem Spieler einen höheren Grad an Immersion zu bieten.

## 9.3. Bewegliche Feinde

Die feindlichen Roboter selbst sind im aktuellen Stand des Spiels örtlich an ihre Felder gebunden. Im Lauf der Geschichte ließe sich argumentieren, dass die feindlichen Roboter die Anwesenheit der eigenen Drohnen wahrnehmen und sich gezielt auf diese zubewegen oder patrouillieren könnten. Dies würde erfordern, dass die eigenen Drohnen besser realisieren, wo sich Feinde befinden und diese Felder gezielt umgehen, um eine Konfrontation zu vermeiden.

## 9.4. Verbesserung der Orientierung im Spiel

Die visuelle Darstellung des Spiels sollte um Tooltips erweitert werden, welche erklären, was es mit bestimmten Buttons auf sich hat. Durch das Bewegen der Maus über diese Buttons sollte ein Fenster generiert werden, aus dem der Spieler erfahren kann, was bei der Betätigung des Buttons geschieht. Dies wurde während der Testphase von mehreren Probanden gewünscht. Das Spielmenü selbst könnte mit Animationen versehen werden, die beispielsweise eintreffende Drohnen beim Abladen von Rohstoffen zeigt, bevor diese wieder losziehen. Auch eine Übersicht über die inaktiven Drohnen wäre so bildlich umzusetzen.

## 9.5. Zusätzliche Funktionen für Drohnen

Eine Funktion, welche die Interaktion des Spielers mit der Außenwelt verbessern könnte, ist die Möglichkeit, Strukturen in der Welt zu errichten, mit welchen der Spieler passiv Rohstoffe generieren kann. Hierzu bietet sich die Errichtung von Minen auf Bergfeldern oder eine automatische Algenfarm auf Wassergebieten an, welche von Drohnen errichtet und bearbeitet wird. Ebenso könnten Drohnen mit Kameramodulen ausgestattet werden, welche eine dreidimensional generierte Darstellung der Umgebung ermöglichen, welche sich um die Drohne selbst befindet. Beide definierten Funktionen könnten durch den Abschluss von Kapiteln freigespielt werden und bieten dem Spieler eine Belohnung, mit welcher sich seine Spielerfahrung auf Dauer verbessern lässt.

## 9.6. Online-Funktionalität

Früh in der Entwicklung entstand die Idee, einen Online-Modus für Lerngruppen zu implementieren. Mithilfe einer Chat-Funktion und der Möglichkeit, den eigenen Bildschirm zu teilen, können sich Nutzer gegenseitig bei Problemen aushelfen und gemeinsam im Spiel voranschreiten. Verschiedene Spieler könnten sich hierbei als Mentoren markieren und auf Anfrage von Spielern, die eventuell Probleme mit einem Thema oder Rätsel zeigen, deren Spiel einsehen und Hinweise geben, die für den Abschluss und das weitere Vorankommen hilfreich sind. Eine solche Funktion ließe sich mit einer Servertechnologie oder einer direkten Verbindung umsetzen. Für die spätere Umsetzung ist der Nutzen dieser Funktion allerdings fraglich, da sich Lerngruppen auch über eine Software wie Discord, AlfaView, Zoom oder Skype austauschen und den gleichen Effekt erzeugen können. Eine

Implementierung würde zwar den Vorteil bringen, dass alle Funktionen zentralisiert innerhalb des Spiels zu finden sind, jedoch bemerkten die befragten Nutzer, dass sie eher bestehende Technologien nutzen würden, als auf ein solches Feature im Spiel zurückzugreifen.

## 9.7. Speicherfunktion

Abschließend ist eine Speicherfunktion zu implementieren, die es dem Spieler ermöglicht, seinen aktuellen Fortschritt im Spiel abzuspeichern und später wieder laden zu können. Dies ließe sich einerseits mithilfe einer JSON-Datei oder durch Zuhilfenahme des „local storage“ realisieren. Veränderte Variablen und Objekte müssten hierfür zwischengespeichert und umgewandelt werden, so dass sie zu einem späteren Zeitpunkt wieder aufgerufen werden können. Der Erkundungsgrad der Karte müsste hierbei ebenfalls berücksichtigt werden, da nur die Kacheln der Welt aktiviert werden, welche bereits von einer Drohne besucht wurden.

## 10. Fazit

Abschließend konnte ich feststellen, dass die Verbindung von Serious Games und der Erzählung einer Geschichte bei vielen Nutzern ein positives Feedback ergeben hat. Dieses Konzept wird aktuell noch kaum genutzt, bietet gerade unerfahrenen Nutzern aber die Möglichkeit, ihr Interesse, neue Themen zu lernen, zu steigern, weil sie den weiteren Verlauf der Geschichte erfahren wollen und deshalb bereit sind, sich mit den Lernthemen auseinander zu setzen.

Auch wenn sich während der Arbeit diverse Probleme aufgetan haben, so sehe ich dieses Projekt als erfolgreich an. Durch die aufgetretenen Probleme konnte ich neue Blickwinkel finden, mit welchen ich diese betrachten und lösen konnte. Auch wenn die Ergebnisse des Nutzertests nicht durchgehend positiv ausfielen, beleuchten sie die Aspekte des Projektes, welche zusätzliche Arbeit erfordern und dienen der Weiterentwicklung des Spiels im Ganzen. Weiterhin war es mir durch die Umsetzung dieses Projektes möglich, einen tieferen Einblick in die Programmiersprache TypeScript zu bekommen und neue Funktionen und Möglichkeiten zu finden, welche mir auch weiterhin von großem Nutzen sein werden. Der entwickelte Prototyp und das Konzept des Spiels konnten nicht jeden Nutzer überzeugen, was mit Sicherheit auf die unterschiedlichen Präferenzen eines jeden Menschen zurückzuführen ist. Nicht jeder zeigt Interesse an allen Genres der Spieleindustrie und nicht jeder ist interessiert daran, die Geschichte eines Spiels zu verfolgen, aber sofern man Willens ist, sich mit dem Thema des Spiels und der Geschichte auseinanderzusetzen, so ist es diesen Nutzern doch möglich, neues zu lernen und zu erfahren.

Mithilfe der geplanten Erweiterungen lässt sich auch dieser Prototyp so erweitern, dass ein weiteres Nutzerfeld angesprochen werden kann.

Ich nehme die gesammelten Erkenntnisse dieses Projektes als sehr positiv für den weiteren Verlauf der Entwicklung auf und bin durchaus zufrieden mit den Ergebnissen, die ich vorzuweisen habe. Somit sehe ich meine These, dass eine ausführliche Geschichte sich positiv auf die Lernmotivation von Nutzern auswirkt, als bestätigt an.



# 11. Quellenverzeichnis

Alle aufgeführten Quellen entsprechen zum Zeitpunkt 26.02.2022 den verwendeten Informationen, welche für diese Arbeit genutzt wurden.

[1] Disclaimer bezüglich gendergerechter Sprache

<https://www.scribbr.de/wissenschaftliches-schreiben/gendern-in-wissenschaftlichen-arbeiten/>

[2] Minecraft Education-Edition für Konkurrenzanalyse

<https://education.minecraft.net/en-us>

[3] Duskers-Steamseite für Konkurrenzanalyse

<https://store.steampowered.com/app/254320/Duskers/>

[4] CodeCombat-Startseite für Konkurrenzanalyse

<https://codecombat.com/play>

[5] FUDGE-Wiki, geführt von Prof. Jirka Dell'Oro-Friedl auf GitHub

<https://github.com/JirkaDellOro/FUDGE/wiki>

[6] TypeScript-Dokumentation

<https://www.typescriptlang.org/docs/handbook/intro.html>

[7] Storytelling und Serious Games

<https://www.mauth.cc/de/storytelling-im-e-learning/>

[8] Warum Menschen durch emotionale Impulse besser lernen

<https://www.erfolgreich-lernen24.de/hirnforschung/besser-lernen-durch-emotionen/ist-lernen-spuerbar-oder-sind-emotionen-notwendig-um-besser-zu-lernen.html>

[9] Narrationstechnik Heldenreise

<https://vertus.co/heldenreise-storytelling/>

[10] Information über Dyskalkulie (Relevanz aufgrund von Problemen in der Testphase)

<https://www.bvl-legasthenie.de/dyskalkulie.html>





## 12. Eidesstattliche Erklärung

Hiermit erkläre ich eidesstattlich, dass die vorliegende Bachelorarbeit von mir ohne unzulässige Fremdhilfe erstellt wurde. Alle verwendeten Quellen und Hilfsmittel sind in dieser aufgeführt.

---

Bad Dürkheim, den