

Individuelle Abschlussarbeit BLJ

Migros Tycoon

Finn Domeisen

27. Juni 2025

Der Migros-Genossenschafts-Bund

Finn



Inhaltsverzeichnis

Inhaltsverzeichnis	2
1 Änderungstabelle	4
2 Einleitung	7
2.1 Aufgabenstellung und Ziel des Projekts	7
2.2 Projektbeschreibung.....	7
2.3 Bekannte Risiken zu Projektbeginn.....	7
3 Planung	8
3.1 Geplante Projektstruktur	8
3.2 Tatsächlicher Projektverlauf	9
3.3 Planung	10
3.4 Terminplan (Gantt-Diagramm)	10
3.5 Welche Wege gehe ich? Welche Möglichkeiten habe ich?	10
3.6 Entscheidungsmatrix	11
3.7 Fazit Planung.....	11
4 Hauptteil	12
4.1 Vorgehen	12
4.1.1 Projektbewilligung und Zielsetzung	12
4.1.2 Tutorials & Einführung in Roblox.....	13
4.1.3 Roblox Account erstellen und vorbereiten	13
4.1.4 Plattform & Map vorbereiten	14
4.2 Was ist ein Tycoon?	18
4.3 Was ist Roblox?	19
4.4 Milestones	19
4.5 Milestone 1	20
4.5.1 Milestone 2	20
4.6 Milestone 3	21
4.7 Ergebnis der Arbeit	21
4.8 Funktionsbeschreibung	21
4.8.1 Dropper und Förderband:	21

4.8.2	Collector:.....	21
4.8.3	Kaufen-Buttons:	22
4.8.4	Upgrader:.....	22
4.9	Arbeitsjournal	22
4.10	Testplan	23
	• Kauf Funktion	23
	• Dropper Logik.....	23
	• Soundauslösung	23
	• GUI-Anzeige	23
	• Tutorial	23
4.11	Fazit	23
5	Arbeitsjournal	25
5.1	Tag 1 – 4. Juni 2025.....	25
5.2	Tag 2 - 5. Juni 2025	25
5.3	Tag 3 - 6. Juni 2025	25
5.4	Tag 4 - 11. Juni 2025	25
5.5	Tag 5 - 12. Juni 2025	26
5.6	Tag 6 - 13. Juni 2025	26
5.7	Tag 7 - 18. Juni 2025	26
5.8	Tag 8 - 19. Juni 2025	26
5.9	Tag 9 - 20. Juni 2025	27
5.10	Tag 10 - 25. Juni 2025	27
5.11	Tag 11 - 26. Juni 2025	27
5.12	Tag 12 - 27. Juni 2025	27
6	Anhang	29
6.1	Quellenverzeichnis / Literaturverzeichnis	29
6.2	Glossar	29
6.3	Programm-Code und Scripts	30
6.4	Foto-Dokumentation	30

1 Änderungstabelle

Version	Datum	Änderung	Beschreibung
0.1	4.6.2025	Repository erstellt	Milestones und Issues auf GitHub erstellt.
0.2	5.6.2025	Dokumentation erstellt und Theorie erarbeitet	Word-Datei für die Dokumentation erstellt und Issues eingefügt. Schwerpunkt lag auf Theorie: Videos geschaut, Artikel zu Avatar, UI, User Input, Objekten und Scripting gelesen. Wichtige Punkte notiert.
0.3	6.6.2025	Weltgestaltung abgeschlossen und Spawnbereich aufgebaut	Die Spielfläche wurde erstellt: Bodenplatte mit Gras, Umrandung aus Hochhäusern und Bergen, Himmelmmodell aus der Toolbox. In der Mitte ist der Start-Spawn mit Migros-Logo. Von dort führen vier Wege zu den Tycoon-Plätzen. Unsichtbare Wand hinzugefügt und dekorative Bäume platziert.
0.4	11.6.2025	Erster Dropper, Förderband und Geldanzeige umgesetzt	Im Homeoffice wurden Floor, Wände, Decke und Eingang des Tycoons mit Parts modelliert. Anschliessend entstand der erste Dropper, der mithilfe von Tutorials und ChatGPT pro Sekunde einen Block erzeugt. Ein erstes Förderband mit Script wurde eingebaut, ebenso eine funktionierende Geldanzeige via ScreenGUI und LocalScript. Der Collector zur Umwandlung der Blöcke in Geld wurde begonnen, funktioniert aber noch nicht wie gewünscht.
1.0	12.6.2025	Gameplay und Kauflogik verbessert, neues Design umgesetzt	Der Fokus lag auf Gameplay und Lua-Scripting. Die Kauflogik wurde verbessert: Geld wird korrekt abgezogen, Käufe sind nur bei ausreichendem Betrag möglich. Das Haus erhielt ein neues Design

			mit realistischen Materialien. Die Sliding Door wurde optimiert. Zudem wurde begonnen, die Kaufreihenfolge zu begrenzen. Ein realistisches 3D-Modell eines Migros Ice Tea wurde mithilfe einer KI erstellt. Die Dokumentation wurde heute kaum erweitert.
1.3	13.6.2025	Weitere Dropper, Produkte-Modelle und GitHub-Issues ergänzt	Zwei Dropper und ein Upgrader wurden gebaut. Mehrere Migros-Produkte (Glace, Energy, Wackelpudding) wurden als 3D-Modelle via AI generiert und ins Regal eingefügt. Ein Titelbild wurde ebenfalls mit AI erstellt. GitHub-Issues wurden aktualisiert. Die Dokumentation blieb unbearbeitet.
1.7	18.6.2025	Dokumentation: Einleitung, Planung & Tabelle ergänzt	Heute den ganzen Tag an der Dokumentation gearbeitet. Einleitung geschrieben, neue Änderungstabelle erstellt, Gantt-Diagramm begonnen und Entscheidungsmatrix eingefügt.
1.9	19.6.2025	Dokumentation ergänzt, Sounds & Modelle überarbeitet	Gantt-Diagramm und Planung in der Dokumentation abgeschlossen. Mehrere Sounds recherchiert, importiert und zentral im ReplicatedStorage eingebunden. Modelle repariert.
2.0	20.6.2025	Ladebildschirm & Tutorial-GUI erstellt, Dokumentation ergänzt	Animierten Ladebildschirm gestaltet und vollständig umgesetzt. Tutorial-GUI beim Spielstart eingebaut. Erste Arbeiten am Hauptteil der Dokumentation begonnen.
2.1	25.6.2025	Hauptteil der Dokumentation verfasst	Heute nur an der Dokumentation gearbeitet. Hauptteil fast fertiggestellt. Bilder und Präsentation fehlen noch, Spiel ist abgeschlossen.
2.2	26.6.2025	Dokumentation weitergeführt	Ganzen Tag an der Dokumentation gearbeitet. Spiel ist fertig. Fotos fehlen noch, Anhang und Tagesjournale werden als Nächstes ergänzt.

2.2	27.6.2025	Dokumentation abgeschlossen & Präsentation erstellt	Heute die gesamte Dokumentation fertiggestellt. Danach mit der Präsentation begonnen und direkt erste Inhalte eingefügt und bearbeitet.
-----	-----------	---	---

2 Einleitung

2.1 Aufgabenstellung und Ziel des Projekts

Im Rahmen meines Abschlussprojekts habe ich mich entschieden, ein eigenes Spiel in Roblox zu entwickeln. Die Plattform Roblox Studio bietet die Möglichkeit, Spiele mit einer Kombination aus 3D-Modellierung, Benutzeroberflächen und der Programmiersprache Lua zu erstellen. Mein Ziel war es, ein Tycoon Spiel zu realisieren, bei dem man durch verdientes Geld eine Migros Filiale Schritt für Schritt aufbauen kann. Dabei beginnt der Spieler mit einem leeren Grundstück und verdient Geld durch Button Klicks oder automatische Dropper. Dieses Geld kann genutzt werden, um verschiedene Bauelemente wie Wände, Dächer, Beleuchtung und Migros Produkte zu kaufen.

2.2 Projektbeschreibung

Die Aufgabenstellung bestand darin, ein funktionierendes und visuell ansprechendes Spiel zu gestalten, das eine klare Logik und ein motivierendes Belohnungssystem hat. Zusätzlich sollte das Spiel die Möglichkeit bieten, den Spielfortschritt nicht zu speichern, damit Spieler das Spiel mehrmals spielen kann. Neben dem technischen Aspekt war es mir auch wichtig, die Migros als Marke im Spiel wiederzuerkennen. Deshalb integrierte ich typische Migros Produkte, das Logo und das bekannte orange Farbdesign.

2.3 Bekannte Risiken zu Projektbeginn

Zu Beginn des Projekts waren mir wenige Risiken bewusst. Einer der grössten Unsicherheitsfaktoren war meine fehlende Erfahrung mit der Programmiersprache Lua. Ich wusste, dass es einige Zeit brauchen würde, um die Grundlagen zu verstehen und sie korrekt anzuwenden. Auch der komplexe Aufbau eines Tycoon Spiels mit mehreren Droppern, einem Förderband, einem Sammler, einem Geldsystem, Benutzeroberflächen und automatisierten Abläufen stellte eine grosse Herausforderung dar. Zudem war unklar, ob alle Modelle und Grafikelemente technisch sauber funktionieren und wie viel Zeit das Modellieren und das Testing beanspruchen würde.

Trotz dieser bekannten Risiken habe ich mich bewusst für ein Projekt entschieden, das mich sowohl gestalterisch als auch technisch fordert. Mein Ziel war es, nicht nur ein Spiel zu bauen, sondern dabei auch viel über die Zusammenhänge zwischen Design, Logik und Code zu lernen.

3 Planung

3.1 Geplante Projektstruktur

	4.6	5.6	6.6	11.6	12.6	13.6	18.6	19.6	20.6	25.6	26.6	27.6
Milestone 1												
Projektbewilligung												
Tutorial Videos												
Tutorial 1. lesen												
Tutorial 2. lesen												
Tutorial 3. lesen												
Roblox Account												
Tutorial Einführung												
Plattform erstellen												
Spawnpunkt erstellen												
Spawnpunkt dekorieren												
Tycoon Plätze												
Tycoon Auswahl												
Geld UI												
Map Wände												
1. Dropper												
Blender Recherche												
Blender installieren												
Milestone 2												
Alles kaufbar machen												
Migros Modelle												
Sounds												
Ladebildschirm												
Milestone 3												
Kontrollieren												
Testen												
Optional: Veröffentlichen												

3.2 Tatsächlicher Projektverlauf

	4.6	5.6	6.6	11.6	12.6	13.6	18.6	19.6	20.6	25.6	26.6	27.6
Milestone 1												
Projektbewilligung												
Tutorial Videos												
Tutorial 1. lesen												
Tutorial 2. lesen												
Tutorial 3. lesen												
Roblox Account												
Tutorial Einführung												
Plattform erstellen												
Spawnpunkt erstellen												
Spawnpunkt dekorieren												
Tycoon Plätze												
Tycoon Auswahl												
Geld UI												
Map Wände												
1. Dropper												
Blender Recherche												
Blender installieren												
Milestone 2												
Alles kaufbar machen												
Migros Modelle												
Sounds												
Ladebildschirm												
Milestone 3												
Kontrollieren												
Testen												
Optional: Veröffentlichen												

3.3 Planung

Für ein erfolgreiches Projekt war eine klare und realistische Planung wichtig. Deshalb habe ich mir bereits zu Beginn überlegt, wie ich meine Zeit aufteilen möchte, welche Ziele ich erreichen will und welche verschiedenen Wege zur Auswahl stehen. Mein Ziel war nicht nur ein funktionierendes, spannendes Roblox Spiel, sondern auch ein Projekt, bei dem ich Neues lernen und mich weiterentwickeln kann.

3.4 Terminplan (Gantt-Diagramm)

Das Projekt dauerte insgesamt zwölf Tage. Um die Zeit sinnvoll zu nutzen, habe ich mein Projekt in mehrere Meilensteine in GitHub unterteilt. Diese Meilensteine halfen mir, die Arbeitsetappen logisch zu strukturieren.

In einem Gantt-Diagramm habe ich diese Schritte zeitlich visualisiert. Dabei habe ich auch Pufferzeiten eingeplant, zum Beispiel für Fehler oder längeres Debugging. In diesen freien Zeiten habe ich ausserdem an der Dokumentation gearbeitet oder mein GitHub-Projekt beziehungsweise das Repository aktualisiert, um den Fortschritt sauber festzuhalten und die Übersicht zu behalten.

3.5 Welche Wege gehe ich? Welche Möglichkeiten habe ich?

Bereits zu Beginn habe ich verschiedene Projektideen gesammelt und ihre Vor- und Nachteile miteinander verglichen. Ich wollte ein Projekt umsetzen, das sowohl kreativ als auch technisch anspruchsvoll ist. Es sollte mir die Möglichkeit geben, eine Programmiersprache zu lernen, visuell zu gestalten und gleichzeitig ein funktionierendes Produkt abzugeben.

3.6 Entscheidungsmatrix

Möglichkeit	Vorteile	Nachteile	Entscheidung
Roblox Tycoon	Kreativ, frei gestaltbar, Lua lernen, Community-Support	Kein Vorwissen mit Programmiersprache, Einarbeitung nötig	Gewählt
ABAP-Tool	Bezug zur zukünftigen Arbeit (Migros SAP), realistisch	Kein Spiel, weniger kreative Gestaltung, schwierige Fehlerbehebung	Abgelehnt
Unity Mini Game	Moderne Engine, viele Möglichkeiten, gute Grafik	Hohe Komplexität, schon mal gemacht	Abgelehnt

Die Wahl fiel nicht direkt klar auf das Roblox-Tycoon-Spiel. Jedoch kombiniert es Spiellogik, Gestaltung, Animation, Programmierung und Benutzerinteraktion. Das sind genau die Bereiche, die mich interessieren. Ausserdem konnte ich mit Lua eine neue Sprache lernen und hatte dank der Roblox Community Zugriff auf viele Hilfestellungen, Vorlagen und Hilfeforen.

Das ABAP-Projekt hätte zwar gut zu meiner künftigen Abteilung gepasst, wäre aber deutlich technischer und weniger kreativ gewesen. Die Umsetzung eines Spiels mit Unity habe ich mich dagegen entschieden, weil ich schon ein Escape Room als Sportferien Projekt gemacht habe.

3.7 Fazit Planung

Insgesamt war die Planung für mich ein sehr wichtiger Teil des Projekts. Sie hat mir geholfen, realistisch zu bleiben, klare Prioritäten zu setzen und meine Zeit sinnvoll einzuteilen. Auch wenn ich nicht jeden einzelnen Punkt exakt nach Plan erledigen konnte, war die grobe Struktur für meinen Fortschritt sehr hilfreich.

4 Hauptteil

4.1 Vorgehen

4.1.1 Projektbewilligung und Zielsetzung

Zuerst musste ich ein selbst gewähltes Projekt planen und bewilligen lassen. Dabei hatte ich drei Ideen, die mir am besten gefielen und über welche ich mir viel Gedanken gemacht habe. Mit Hilfe von dem vorbereiteten Chat GPT prompt von Frank, konnte mir viele Ideen geben, die ich möglicherweise als Projekt machen könnte. Leider war ich mit keiner richtig glücklich und habe überlegt, was mich in der Freizeit interessiert, oder ob ich ein Problem habe, für das ein Programm das Problem lösen würde oder etwas mit Migros zu tun hat.

Zur Auswahl waren: Ein Spiel mit Unity, ein Tool mit ABAP und ein Roblox Spiel. Da ich in den Sportferien schon ein Spiel mit Unity erstellt habe, fand ich es wäre langweilig, wenn ich nochmal ein Unity Spiel machen würde. Jedoch hat mir das Spiel erstellen sehr spass gemacht und ich fand es sehr spannend.



Abbildung 1 Bild von Unity Logo mit Beschriftung

Das zweite; ein ABAP-Tool war zum einen grossartig, da es mit Migros, beziehungsweise meiner zukünftigen Abteilung zu tun hat. Jedoch finde ich ein Tool zu machen nicht sehr ansprechend, wenn man jetzt andere Möglichkeiten hätte.

Selbst spiele ich ab und zu Roblox und weiss deshalb, dass sie eine Spiele Engine gratis anbieten. Dann dachte ich mir ich könnte doch ein Roblox Spiel erstellen. Die Idee das ich etwas mit Migros verbinde habe ich nicht vergessen, deswegen habe ich ein Tycoon-Spiel in Roblox erstellt, bei dem man eine Migros-Filiale aufbauen kann.



Abbildung 2 Roblox

Nachdem ich meine Idee Reto vorgetragen, musste ich ihm noch erklären was Roblox überhaupt ist. Als Reto mein Projekt bewilligt hat, fing ich direkt an.

Als erstes habe ich das GitHub Repository erstellt und das ReadMe geschrieben. Dazu habe ich noch das GitHub Project erstellt und mit dem Repository verknüpft.

Im Project habe ich die 3 Meilensteine erstellt und habe die ersten Issues geschrieben. Dies war am Anfang schwer, weil man nicht weiss, was noch dazwischen kommt oder ob etwas fehlt.

4.1.2 Tutorials & Einführung in Roblox

Ich startete mit mehreren Roblox Video-Tutorials und learning Artikel gelesen die allgemein um Roblox gehen, aber auch um spezifische Themen wie Avatar, Animation, UI, Userinput, Objekte, Software, Scripting.

Es gibt von dem offiziellen Roblox Account sehr viele hilf-, erklär und Tutorial-Videos. Diese habe ich auf YouTube geschaut oder habe auf <https://create.roblox.com/docs/> nachgelesen

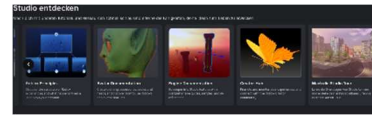


Abbildung 3 Tutorials von Roblox

Ich startete mit mehreren Roblox- und Lua-Tutorials, um die Grundlagen zu verstehen:

- Benutzeroberfläche von Roblox Studio
- Wie Lua funktioniert (Variablen, Bedingungen, Events, Funktionen)
- Wie man GUIs, Buttons und Trigger erstellt
- Was ein Tycoon-Modul ist und wie man es anpasst
- Was Avatare sind in Roblox
- Die die Animation funktioniert, aussieht und wie man sie anpassen oder selbst ändern könnte.

Ich las mehrere Tutorials und habe ab und zu das wichtigste notiert, das ich dachte, dass ich später oder für den Tycoon noch brauchen werde.

4.1.3 Roblox Account erstellen und vorbereiten

Ich erstellte einen Roblox Account. Bei der Auswahl des Namens habe ich mir viel Zeit genommen und habe ChatGPT nach Ideen gefragt und wie er meine originale Idee: «MigrosTycoon» fand. Ebenfalls fragte ich mein Banknachbar Benni, ob er den Namen passend fand oder ob er noch eine bessere Idee hätte.



Abbildung 5 Account erstellen

Nach dem Erstellen meines Accounts richtete ich die Roblox Studio Umgebung ein. Ich habe geschaut, wie es aussieht und habe geschaut, was man alles erstellen kann. Daraufhin habe ich viele Vorlagen gefunden, die man direkt importieren kann und dann einfach selbst bearbeiten und anpassen kann.

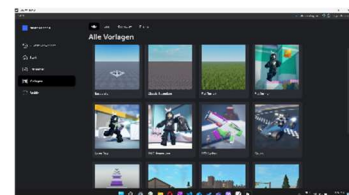


Abbildung 4 Vorlagen für Spiele

Nach dem begann ich mit einem leeren Template, in das ich mein eigenes Tycoon-System einbauen wollte. Es war einfach eine leere graue Platte, die sehr gross war, aber an einem bestimmten Ort ins nichts endet.

4.1.4 Plattform & Map vorbereiten

Ich fing an meine eigene Map zu bauen:

- Auf die bereits erstellte Plattform habe ich eine grüne Grasfläche gesetzt, damit es schöner aussieht.
- Da die Map nicht unendlich ist, habe ich sie mit unsichtbaren Barrieren umrandet, damit man nicht aus Versehen ins nichts fällt und dann nichts passieren wird.
- Auf der einen Seite der Welt habe ich viele Hochhäuser hingestellt und verschiedenen Grössen und Richtungen angepasst. Diese bereits vorgefertigten Modelle habe ich von der Toolbox in Roblox importieren können.
- Als Himmel konnte ich ebenfalls einen sehr schönen von der Toolbox importieren.
- Dann fing ich mit dem Spawnpunkt an. Das ist der Punkt, auf dem jeder Spieler am Anfang erscheint, wenn er ins Spiel geht. Das Modell, also die Textur von diesem Block, auf dem man spawnnt habe, ich mit dem Migros Logo ersetzt.
 - Das habe ich mit dem Programm «Blender» gemacht.
- Von diesem Punkt in der Mitte aus habe ich vier Steinwege erstellt, die zu den jeweiligen Tycoons führen.
- Für die Dekoration habe ich ein paar Bäume eingefügt.

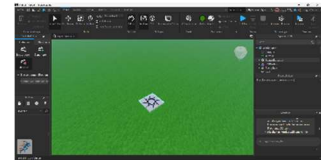


Abbildung 6 Grasfläche

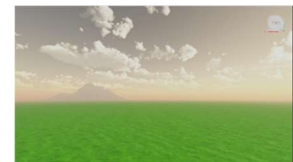


Abbildung 7 Schöner Himmel



Abbildung 8
Spawnpunkt



Abbildung 9 Bäume
erstellt

4.1.4.1 Tycoon-System und erstes Gameplay

Dann begann ich mit der Umsetzung des eigentlichen Spiels:

- Erstmal fing ich mit einem Tycoon an, den ich dann drei weitere Male kopieren könnte.
- Ich erstellte mal eine graue Grundplatte, auf der ich dann anfangen konnte.
- Anschliessend habe ich die Wände und das Dach gebaut und dabei eine passende Textur ausgewählt.

- Dann habe ich mit dem ersten Dropper angefangen. Ich habe kurze Tutorials gelesen und geschaut und dann angefangen das Gerüst zu bauen aus zwei Blöcken. Der Dropper, also der Quader wo die Drops rauskommen, habe ich angefangen zu programmieren. Mit Hilfe von Onlinehilfen und ChatGPT konnte ich es programmieren, dass jede Sekunde ein Block aus diesem Quader kommt.
- Ein essenzieller Teil eines Tycoons ist das Förderband, welche die Blöcke transportiert. Im Internet habe ich ein bisschen recherchiert, bis ich einen Lua Script gefunden habe, welcher genau das macht. Es hat funktioniert, aber noch nicht sehr geschmeidig.
- Nach dem Transportieren der Blöcke kommt eine Platte, welche die Blöcke auffängt, löscht und anschliessend in Geld umwandelt. Diesen habe ich versucht zu erstellen und programmieren und habe es erfolgreich geschafft. Dieser heisst «Collector».
- Um nachzusehen wieviel Geld man hat, habe ich die Geldanzeige gemacht. Mit einem ScreenGUI, TextLabel und einem LocalScript konnte ich es umsetzen.
- Das Geld wird nicht direkt zu deinem Geldstand hinzugefügt, sondern zu einem «giver». Wenn man ihn berührt, bekommt man das Geld ausbezahlt.
- Damit nur die ausgewählte Person Gegenstände für den Tycoon kaufen kann, richtete ich die Ownership-Systeme ein (Touch-to-Claim).
- Das Geld funktioniert mit einer Variable, welche mit 0 definiert ist und immer hochgezählt wird, wenn ein Block den Collector berührt.

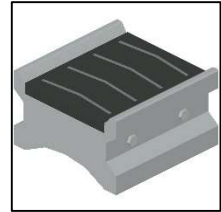


Abbildung 10
Förderband

4.1.4.2 Dropper, Förderband & Upgrader

Ich entwickelte die ersten Dropper, die regelmässig grauen Blöcke erzeugen. Diese bewegen sich über ein Förderband zum Collector.

Später fügte ich einen Upgrader hinzu, welche die Blöcke umwandelt, damit sie mehr Geld wert sind.

4.1.4.3 Kaufen-Logik und Buttons

Ich programmierte die Kauf-Logik:

- Spieler sehen rote Flächen mit einem Preis und der Beschriftung, was der Gegenstand ist.
- Nur wenn sie genug Geld haben, können sie kaufen. Das wird mit einem Event geprüft, ob die Variable genug hoch ist.

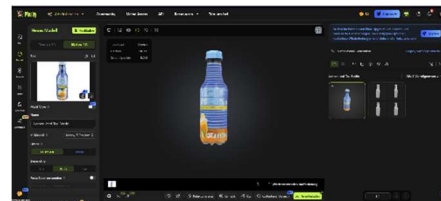
- Nach dem Kauf wird das Objekt sichtbar. Das kann zum Beispiel ein Dropper sein, welche dann anfängt die Blöcke zu spawnen.
- Die Buttons verschwinden danach und geben den nächsten frei. Das funktioniert mit einem Abhängigkeitsscript. Im Code habe ich angegeben, dass man erst ein Gegenstand kaufen kann, wenn der vorherige gekauft wurde. Das kann man in den Properties von den einzelnen Items anpassen.

4.1.4.4 Migros-Modelle mit AI erstellen

Ich wollte Migros-Produkte in mein Spiel implementieren, zum Beispiel in einem Regal oder das M auf dem Dach. Da ich nicht alles selbst modellieren konnte, habe ich im Internet nach AIs gesucht, welche das können. Zum Glück habe ich eine sehr gute AI gefunden namens <https://www.meshy.ai>.

So nutzte ich diese AI, die aus Bildern 3D-Modelle generierte. So entstanden:

- Migros Ice Tea
- Energy Drink
- Glace
- Wackelpudding



Diese stellte ich im Spiel in einem Regal dar.

Abbildung 11 AI modellierter Ice Tea

4.1.4.5 Sounds hinzufügen

Ich baute passende, unterschiedliche Sounds ein, wie:

- Beim Kauf eines Objekts
- Beim Einsammeln von Geld
- Beim Beanspruchen eines Tycoon-Platzes

Diese Sounds liegen im ReplicatedStorage und werden zur Laufzeit abgespielt.

4.1.4.6 Ladebildschirm

Um den Spielstart optisch ansprechend zu gestalten und den Ladevorgang zu verstecken, programmierte ich ein eigenes GUI, das automatisch beim Betreten des Spiels angezeigt wird. Dieses GUI besteht aus mehreren grafischen und funktionalen Elementen:

- Eine grosse, zentrierte Schrift mit dem Titel „Migros Tycoon“, die mit einem animierten Leuchteffekt versehen ist. Dabei füllt sich der Schriftzug nach und nach in der Migros-typischen orangen Farbe. Für die Schriftart habe ich mich von meinen Banknachbarn beraten lassen und habe eine sehr passende, moderne gefunden.
- Direkt darunter befindet sich ein horizontaler Ladebalken, der sich von links nach rechts mit oranger Farbe auffüllt. Dieser Visualisiert den Ladeprozess.
- In der Mitte unter dem Ladebalken ist eine Prozentanzeige eingebaut, die synchron mit dem Ladebalken den Fortschritt von 0 % bis 100 % anzeigt.

Die gesamte Animation wird durch ein LocalScript gesteuert, dass die UI-Elemente mit Tweening-Effekten modifizieren. Dadurch wirken die Bewegungen weich und flüssig. Sobald der Ladebalken 100 Prozent erreicht hat, blendet sich das GUI automatisch aus und das eigentliche Spiel erscheint.

Dieses Feature verbessert den ersten Eindruck vom Spiel und macht es insgesamt professioneller. Durch die Farben, die Animationen und den klaren Aufbau wirkt der Start ins Spiel spannender und gut gemacht.

4.1.4.7 Slide-Tutorial erstellen

Ich wollte neuen Spielern den Einstieg ins Spiel erleichtern, damit sie die wichtigsten Funktionen schnell verstehen. Deshalb habe ich ein eigenes Tutorial erstellt, das direkt nach dem Laden des Spiels erscheint. Es handelt sich um eine Slideshow, die Schritt für Schritt die Spielmechanik erklärt.

Das Tutorial besteht aus einem einfachen GUI mit einem „Next“ Button, mit dem man sich durch die einzelnen Seiten klicken kann. Auf jeder Seite wird ein kurzer Text angezeigt, z. B. wie man sich bewegt, dass man mit Shift schneller rennen kann oder dass man auf rote Flächen stehen muss, um etwas zu kaufen. Zusätzlich habe ich passende Bilder eingefügt, die zeigen, was gemeint ist – zum Beispiel ein Dropper, das Förderband oder der Collector.

Am Ende der Slideshow wollte ich noch etwas Lustiges einbauen, darum habe ich als letztes Slide ein animiertes GIF angezeigt, um das Tutorial mit einem kleinen Highlight

abzuschliessen. Die ganze Einführung hilft dabei, dass man das Spiel besser versteht und motivierter loslegt. Ich finde, es macht das Spiel zugänglicher und sieht auch cool aus.

4.1.4.8 Testphase

Ich testete das Spiel ausführlich:

- Funktionieren alle Buttons?
- Werden Objekte korrekt freigeschaltet?
- Spielt der Sound zur richtigen Zeit?
- Läuft das Tutorial sauber durch?
- Funktionieren Dropper und Upgrader zuverlässig?

Fehler dokumentierte ich in meinem GitHub-Issue-Board und bearbeitete sie dort.

4.2 Was ist ein Tycoon?

Ein Tycoon ist ein Spielgenre, bei dem die Spielerinnen und Spieler ein Unternehmen, eine Organisation oder ein bestimmtes System Schritt für Schritt aufbauen und verwalten. Ziel ist es meistens, durch geschickte Entscheidungen und strategisches Handeln immer mehr Geld zu verdienen und damit neue Inhalte freizuschalten. Im Gegensatz zu Actionspielen steht beim Tycoon der wirtschaftliche Aufbau im Vordergrund.

Besonders in Roblox sind Tycoon-Spiele sehr beliebt. Dort beginnt man oft mit einer leeren Fläche und baut durch das Sammeln von Geld und das Kaufen von sogenannten "Upgrades" nach und nach eine bestimmte Einrichtung auf – zum Beispiel eine Fabrik, einen Flughafen oder wie in meinem Projekt eine Migros-Filiale. Geld erhält man in der Regel automatisch über sogenannte "Dropper", die regelmässig kleine Geldbeträge erzeugen. Diese Beträge sammelt man und verwendet sie, um neue Funktionen oder Gebäudeteile freizuschalten.

Das Tycoon-Prinzip ist einfach zu verstehen, bietet aber durch die Kombination aus Belohnung und Fortschritt einen grossen Spielreiz. Man sieht ständig, wie sich die eigene Spielfläche verändert und besser wird. Das motiviert viele Spielerinnen und Spieler, weiterzuspielen und ihre Einrichtung vollständig fertigzustellen.

4.3 Was ist Roblox?

Roblox ist eine Online-Plattform, auf der Nutzerinnen und Nutzer eigene Spiele entwickeln und veröffentlichen können. Gleichzeitig ist es auch eine Spieleplattform, auf der man die Werke anderer spielen kann. Das Besondere an Roblox ist, dass die meisten Spiele nicht von professionellen Firmen, sondern von privaten Personen oder kleinen Teams erstellt wurden. Dadurch gibt es eine grosse Vielfalt an Spielideen, von einfachen Minispielen bis hin zu komplexen Simulationen.

Die Spiele in Roblox werden mit der Programmiersprache Lua erstellt. Roblox stellt dafür eine Entwicklungsumgebung namens Roblox Studio zur Verfügung. Damit kann man 3D-Modelle platzieren, Oberflächen gestalten, Animationen einbauen und die Spiellogik programmieren. Auch Sounds, Benutzeroberflächen und Effekte lassen sich einbauen.

Roblox richtet sich besonders an junge Entwicklerinnen und Entwickler, weil die Plattform kostenlos ist und der Einstieg relativ einfach gelingt. Durch das Baukastensystem mit vielen vorgefertigten Objekten können auch Anfänger schnell erste Ergebnisse sehen. Fortgeschrittene Nutzer können mit eigenen Skripten und 3D-Modellen sehr individuelle Spiele erstellen.

Insgesamt bietet Roblox also nicht nur Spielspass, sondern auch eine gute Möglichkeit, erste Erfahrungen im Bereich Game Design und Programmierung zu sammeln. Genau aus diesem Grund eignet sich Roblox besonders gut für ein Schulprojekt wie mein Migros-Tycoon-Spiel.

4.4 Milestones

Mein Projekt „Migros Tycoon“ startete mit einer strengen Einarbeitungsphase in die Grundlagen von Roblox Studio und der Programmiersprache Lua. In den ersten Tagen arbeitete ich mit Online-Tutorials, Dokumentationen und Videos, um die Grundlogik von Tycoon Spielen zu verstehen. In dieser Phase war mein Hauptziel, die Komplexität des Vorhabens realistisch einzuschätzen und gleichzeitig die Projektstruktur zu planen.

Ich habe mein Projekt in drei zentrale Meilensteine eingeteilt, die den Ablauf strukturieren und den Fortschritt sichtbar machen:

- Milestone 1: Vorbereitung & Grundlagen
- Milestone 2: Spielmechanik und Inhalte
- Milestone 3: Testing & Kontrolle (Dokumentation)

Jeder Meilenstein war mit konkreten Aufgaben verknüpft, zum Beispiel „Roblox Account erstellen“, „Spawnpunkt dekorieren“, „Erster Dropper“, „Sounds“ oder „Ladebildschirm“. Diese Planung hat mir sehr geholfen, den Überblick zu behalten und mich jeden Tag auf einzelne Schritte zu konzentrieren.

4.5 Milestone 1

beinhaltete die theoretische Einführung. Ich lernte viel über die Funktionsweise von GUIs, UserInput, Animationen und Objekte in Roblox.

Ausserdem bereitete ich die Spielwelt vor: Ich gestaltete die Hauptplattform, fügte eine Grasplatte darüber hinzu, modellierte Gebäude und Berge und legte den zentralen Spawnpunkt fest. Dieser wurde mit einem Migros Logo dekoriert. Anschliessend baute ich vier Wege, die zu den jeweiligen Tycoon Plätzen führen.

Im Verlauf dieses Abschnitts entwickelte ich auch erste einfache Tycoon Mechaniken: Ich programmierte einen Dropper, der regelmässig Blöcke an einem bestimmten Ort erzeugt, die sich über ein programmiertes Förderband zur Collector Platte bewegen, welche die Blöcke löscht und den Geldstand hochzählt. Hier kam Lua intensiv zum Einsatz. Ich lernte, wie man „Touched“-Events nutzt, GUI-Werte aktualisiert und Variablen auswertet. Dies war eine grosse Herausforderung, da ich vorher keine Lua Erfahrung hatte. Ich habe manchmal die Unterstützung von ChatGPT genutzt und dessen Vorschläge in mein Projekt eingebaut.

4.5.1 Milestone 2

fokussierte sich auf die spielerische Interaktivität.

Ich machte die bereits erstellten Gegenstände kaufbar, indem ich ein Button System programmierte. Spieler können Objekte nur dann kaufen, wenn sie genug Geld haben. Ausserdem arbeitete ich an einem System, das eine Reihenfolge beim Freischalten von Objekten erzwingt. Also dass man erst das vorherige Objekt kaufen muss, damit man das nächste überhaupt sieht.

Parallel dazu erstellte ich Migros Produkte mithilfe von AI-generierten 3D-Modellen (Ice Tea, Glace, Wackelpudding etc.), die in einem Regal ausgestellt werden. Das Erlebnis wurde durch passende Sounds ergänzt, die beim Kaufen oder Einsammeln von Geld abgespielt werden. Auch der Ladebildschirm habe ich in dieser Phase programmiert. Dieser zeigt animiert „Migros Tycoon“ sowie eine Prozentanzeige und einen Ladebalken.

4.6 Milestone 3

konzentrierte sich auf Kontrolle und Test.

Ich überprüfte die Funktionalität aller Scripts, testete die Logik der Dropper, überprüfte UI-Anzeigen, kaufbare Objekte und die Animationen. Um neuen Spielern die wichtigsten Funktionen verständlich zu machen, habe ich eine Slide-Show erstellt, die zentrale Spielmechaniken wie das Rennen mit Shift oder das Kaufen über rote Flächen erklärt. Ich plante keine Speicherung des Spielfortschritts, da der Tycoon nicht sehr lange dauert und man ihn deshalb mehrmals machen können sollte.

4.7 Ergebnis der Arbeit

Das Ergebnis ist ein vollständiges, fehlerfreies Tycoon-Spiel, das es Spielern ermöglicht, ihre eigene Migros Filiale aufzubauen. Es beginnt mit der Auswahl eines Platzes und führt über den Aufbau von Droppern, Upgrader und Deko-Objekten zur Simulation eines funktionierenden Geschäfts. Alle Objekte werden visuell durch Produkte und Logos dargestellt. Die Benutzeroberfläche ist benutzerfreundlich und wurde komplett mit GUIs, Sounds und Ladebildschirm ausgestattet. Das Spiel bietet einsteigerfreundliches Gameplay und eine positive, markentreue Atmosphäre.

4.8 Funktionsbeschreibung

Die Hauptfunktionen lassen sich in vier Gruppen einteilen:

4.8.1 Dropper und Förderband:

Der Dropper erzeugt automatisch Blöcke, die im Spiel Migros-Produkte darstellen. Sobald der Spieler den Dropper gekauft hat, wird er sichtbar und beginnt durch ein Lua-Skript alle paar Sekunden einen Block zu erzeugen.

Diese Blöcke fallen auf das Förderband, das sie mit konstanter Geschwindigkeit in Richtung Collector bewegt. Das geschieht über ein Skript, das die Velocity der Blöcke beeinflusst. Beide Elemente Dropper und Förderband müssen zuerst gekauft werden, bevor sie aktiv sind.

Gemeinsam sorgen sie dafür, dass der Spieler automatisch Geld verdienen kann.

4.8.2 Collector:

Der Collector ist der Teil des Spiels, der die vom Förderband transportierten Blöcke einsammelt und in Geld umwandelt. Sobald ein Block den Collector berührt, wird er

durch ein Skript erkannt, zerstört und der entsprechende Geldwert dem Spieler gutgeschrieben.

Das funktioniert über ein Touched-Event, das prüft, ob der Block ein Attribut mit Geldwert besitzt. Der Collector ist ebenfalls ein kaufbares Objekt und erst aktiv, nachdem er freigeschaltet wurde.

Er bildet den letzten Schritt im Produktionsablauf und ist notwendig, um aus den produzierten Blöcken tatsächlich Cash zu generieren.

4.8.3 Kaufen-Buttons:

Der Kauf-Button ist ein rotes Feld im Spiel, das es dem Spieler ermöglicht, neue Objekte wie Dropper, Wände oder Förderbänder freizuschalten. Wenn der Spieler auf den Button tritt und genügend Geld hat, wird das zugehörige Objekt sichtbar und aktiviert.

Im Hintergrund prüft ein Skript, ob der Spieler der Besitzer des Tycoons ist und genug Geld besitzt. Ist das der Fall, wird der Betrag abgezogen und das Objekt wird dem Spielbereich hinzugefügt. Nach dem Kauf verschwindet der Button oder wird ausgeblendet.

Zusätzlich kann über eine Abhängigkeitsfunktion („Dependency“) gesteuert werden, dass bestimmte Buttons erst erscheinen, wenn vorherige Objekte bereits gekauft wurden. So entsteht eine feste Kaufreihenfolge.

4.8.4 Upgrader:

Der Upgrader verbessert die vom Dropper erzeugten Blöcke, indem er ihnen beim Durchlaufen einen höheren Geldwert verleiht. Sobald ein Block den Upgrader passiert, wird sein Wert erhöht – das bedeutet, der Spieler erhält beim Einsammeln mehr Cash.

Technisch wird beim Berühren des Upgraders ein Attribut im Block angepasst, das den Geldwert speichert. Der Upgrader funktioniert nur, wenn er zuvor über einen Kauf-Button freigeschaltet wurde.

Er ergänzt das Produktionssystem und motiviert den Spieler, seine Anlage weiter auszubauen, um effizienter Geld zu verdienen.

4.9 Arbeitsjournal

Ich habe täglich ein Journal geführt, dass meine Fortschritte, Herausforderungen und Lösungswege dokumentiert. Diese Einträge zeigen den realen Verlauf im Vergleich zur

geplanten Gantt-Übersicht. Besonders deutlich wird dabei, dass ich zwischendurch mehr Zeit mit Coding verbrachte als ursprünglich angenommen.

4.10 Testplan

Um mein Spiel auf Funktionalität zu testen, habe ich folgende Tests durchgeführt:

- **Kauf Funktion:** Spieler können nur mit genug Geld kaufen, Reihenfolge wird eingehalten.
- **Dropper Logik:** Blöcke werden korrekt erzeugt, transportiert und eingesammelt. Es läuft flüssig auf dem Förderband und wird direkt eingesammelt ohne Verzögerung.
- **Soundauslösung:** Alle geplanten Sounds (Kauf, Einsammeln, Ownership) spielen zur richtigen Zeit.
- **GUI-Anzeige:** Ladebalken, Texte und Buttons funktionieren korrekt. Geldstand funktioniert und wird direkt hochgezählt.
- **Tutorial:** Slide Wechsel funktioniert, Bilder und Texte passen und sind logisch. Bilder sind nicht zu fest verzogen und sind scharf, dass man alles erkennt.

Fehler oder Bugs habe ich in dokumentiert und die Lösungen aufgeschrieben.

4.11 Fazit

Das Abschlussprojekt war für mich eine sehr lehrreiche und spannende Erfahrung. Ich konnte mein eigenes Roblox-Spiel von Grund auf planen, gestalten und programmieren – etwas, das ich vorher noch nie gemacht habe. Dabei habe ich nicht nur meine technischen Fähigkeiten verbessert, sondern auch gelernt, wie wichtig saubere Planung, Geduld und Ausdauer bei einem Projekt sind.

Besonders stolz bin ich auf die Umsetzung der Spielmechanik mit Droppern, Förderbändern, dem Collector, den Upgrade-Systemen und der Geldlogik. Auch das Design von Benutzeroberflächen, Soundeffekten und der Ladebildschirm haben dazu beigetragen, dass mein Spiel einen professionellen Eindruck macht. Durch die Tutorial-Slideshow wird neuen Spielern der Einstieg erleichtert, was mir persönlich wichtig war.

Natürlich gab es auch Herausforderungen, wie Bugs in der Logik oder unklare Lua-Fehler, aber mit Recherche, Tests und Hilfe von ChatGPT konnte ich alle Probleme lösen.

Ich habe gelernt, wie man mit GitHub strukturiert arbeitet, ein Kanban-Board nutzt und wie man ein Spiel mit einer klaren Struktur und Reihenfolge aufbaut.

Insgesamt bin ich sehr zufrieden mit meinem Ergebnis. Ich konnte ein kreatives und spielbares Projekt realisieren, das auch optisch überzeugt. Für die Zukunft habe ich nun eine gute Grundlage, um weiter mit Lua, Roblox Studio und Game Design zu arbeiten.

5 Arbeitsjournal

5.1 Tag 1 – 4. Juni 2025

Ich habe mein Abschlussprojekt offiziell gestartet. Anfangs war ich noch unsicher über das Thema. Mit Hilfe eines KI-Prompts suchte ich nach Ideen, aber keine hat mich wirklich überzeugt. Ich dachte über sinnvolle Tools oder Spiele nach und kam schliesslich auf die Idee, ein eigenes Spiel zu machen. Da ich privat manchmal Roblox spiele, entschied ich mich für ein **Tycoon-Spiel, bei dem man eine Migros-Filiale aufbaut**. Das passte auch zu meinem Lehrbetrieb. Ich richtete mein GitHub Repository ein, erstellte das Kanban-Board und definierte erste Issues.

5.2 Tag 2 - 5. Juni 2025

Heute war ein reiner Theorie-Tag. Ich war vom vielen Lesen und Aufnehmen ziemlich erschöpft. Ich habe mir verschiedene Lerninhalte angeschaut: Tutorials, Artikel, Dokumentationen zu Roblox Studio und Lua. Themen waren unter anderem: Avatare, Animationen, UI, Userinput, Objekte, Scripting. Ich machte mir Notizen zu den wichtigsten Punkten und füllte meine Dokumentationsstruktur weiter.

5.3 Tag 3 - 6. Juni 2025

Ich habe heute die Spielwelt gebaut. Ich begann mit einer Grundplatte, darüber eine Grasfläche. Der Rand bestand auf einer Seite aus Hochhäusern, auf der anderen aus Bergen. Den Himmel habe ich aus der Toolbox importiert. Ich erstellte vier Wege von einem zentralen Punkt aus, wo auch der Spawnpunkt liegt – dekoriert mit einem Migros-Logo. Um die Welt realistischer zu machen, baute ich noch Bäume und eine unsichtbare Wand, damit Spieler nicht aus der Welt herauslaufen können.

5.4 Tag 4 - 11. Juni 2025

Ich hatte Homeoffice und arbeitete intensiv am Spiel. Ich erstellte Floor, Walls, Decke und Eingang eines Tycoon-Gebäudes. Dann begann ich mit dem ersten Dropper: Zwei Parts und ein Skript erzeugen regelmässig einen Block. Mit Tutorials und Hilfe von ChatGPT funktionierte das. Danach baute ich ein Förderband, das noch nicht ganz flüssig läuft. Ich gestaltete auch die Geldanzeige mit ScreenGUI, TextLabel und LocalScript. Am Schluss arbeitete ich am Collector, welcher Geld geben soll, aber noch nicht korrekt funktioniert.

5.5 Tag 5 - 12. Juni 2025

Ich machte grosse Fortschritte im Bereich Gameplay und Logik. Die Hausstruktur überarbeitete ich mit besseren Materialien, es sieht jetzt realistischer aus. Ich optimierte die Sliding Door und programmierte die Kauf-Logik: Man kann nur kaufen, wenn man genug Geld hat. Auch die Reihenfolge von Upgrades wird nun geprüft. Für das Förderband suchte ich nach besseren Lösungen. Mit AI erstellte ich ein 3D-Modell vom Migros Ice Tea, das hervorragend gelang.

5.6 Tag 6 - 13. Juni 2025

Ich arbeitete weiter an der Spielmechanik. Ich generierte ein grossartiges Titelbild mit AI, das die Stimmung des Spiels gut einfängt. Ich pflegte mein GitHub-Projekt mit neuen Issues und schloss erledigte Aufgaben ab. Ich erstellte weitere Migros-Modelle: Energy Drink, Glace, Wackelpudding. Im Regal sehen sie sehr schön aus. Ich baute zwei neue Dropper und begann mit dem Upgrader, der graue Blöcke in wertvollere Migros-Produkte verwandelt.

5.7 Tag 7 - 18. Juni 2025

Ich arbeitete heute ausschliesslich an der Dokumentation. Ich erstellte eine vollständige Änderungstabelle und schrieb die Einleitung. Anschliessend begann ich mit der Planung: Ich baute ein Gantt-Diagramm auf, was durch viele Aufgaben recht aufwändig war. Zusätzlich erstellte ich eine Entscheidungsmatrix zur Projektwahl und schrieb den zugehörigen Textabschnitt.

5.8 Tag 8 - 19. Juni 2025

Heute hatte ich Homeoffice. Am Vormittag schrieb ich an der Dokumentation, schloss die Planung und das Gantt-Diagramm ab. Am Nachmittag arbeitete ich im Spiel: Ich suchte passende Sounds, konvertierte sie und importierte sie ins Roblox Creator Hub. Statt jeden Sound mehrfach einzubauen, lagerte ich sie im ReplicatedStorage. Ich reparierte auch fehlerhafte Modelle, die in der Luft schwebten. Als nächstes plante ich den Ladebildschirm.

5.9 Tag 9 - 20. Juni 2025

Ich konnte heute den Ladebildschirm fertigstellen – animiert, farbig, mit Ladebalken und Prozentanzeige. Zuerst hatte ich ein Problem mit der Darstellung über den ganzen Bildschirm, dass ich durch eine Änderung im LocalScript lösen konnte. Danach begann ich mit dem Tutorial-GUI, das beim Spielstart erscheint. Es erklärt in mehreren Slides die Grundfunktionen des Spiels. Ich räumte den Code noch etwas auf. Zum Schluss schrieb ich weiter an der Dokumentation.

5.10 Tag 10 - 25. Juni 2025

Heute habe ich mich vollständig auf die Dokumentation konzentriert. Ich arbeitete hauptsächlich am Hauptteil und konnte dort grosse Fortschritte machen. Der grösste Teil ist nun geschrieben, eventuell werde ich ihn noch etwas erweitern oder einzelne Punkte ergänzen, falls mir später noch etwas einfällt.

Mit dem Spiel selbst bin ich so gut wie fertig, weshalb ich mich nun voll auf die schriftliche Arbeit fokussieren kann. Bilder habe ich bisher noch keine eingefügt und auch mit der Präsentation habe ich noch nicht begonnen. Mein Ziel ist es nun, die Dokumentation sauber abzuschliessen und anschliessend mit der Vorbereitung der Präsentation zu starten.

5.11 Tag 11 - 26. Juni 2025

Heute habe ich den ganzen Tag ausschliesslich an der Dokumentation gearbeitet. Mein Spiel ist vollständig fertig, sodass ich mich nun ganz auf den schriftlichen Teil konzentrieren kann.

Fotos habe ich noch keine eingefügt, das plane ich morgen zu erledigen. Was noch fehlt, ist der Anhang mit Quellen, Glossar und Codes sowie das Einfügen der aktuellen Tagesjournale. Ich bin mit dem Fortschritt zufrieden und bin zuversichtlich, dass ich die Dokumentation bald abschliessen kann.

5.12 Tag 12 - 27. Juni 2025

Heute habe ich die schriftliche Dokumentation zu meinem Migros-Tycoon-Spiel in Roblox fertiggestellt. Ich habe alle wichtigen Themen ergänzt, unter anderem eine Erklärung, was ein Tycoon ist, und eine Beschreibung der Plattform Roblox. Dabei habe ich darauf geachtet, dass der Text sachlich und gut verständlich bleibt.

Nachdem die Dokumentation abgeschlossen war, habe ich mit der Präsentation begonnen. Ich habe direkt ein neues Präsentationslayout erstellt und mit den ersten

Inhalten gefüllt. Dabei habe ich wichtige Punkte aus meiner Arbeit übernommen, zum Beispiel den Aufbau des Spiels, meine Meilensteine und meine Erfahrungen mit Lua und Roblox Studio. Ich habe ausserdem gleich mit der Bearbeitung begonnen, um ein übersichtliches und passendes Design zu finden. Das Ziel ist es, dass die Präsentation sowohl optisch ansprechend als auch inhaltlich klar aufgebaut ist.

Insgesamt war es ein produktiver Tag, da ich zwei wichtige Teile des Projekts gut voranbringen konnte.

6 Anhang

6.1 Quellenverzeichnis / Literaturverzeichnis

- Roblox Developer Hub – <https://create.roblox.com/docs>
- Offizielle Lua Website – <https://www.lua.org/docs.html>
- YouTube Tutorials von [BinzuDev](#), [Roblox Learn](#)
- ChatGPT (OpenAI) – Unterstützung bei Code, Texten und Planung
- GitHub Repository:
https://github.com/finndomeisen/Abschlussprojekt_2025_API-2_findom_Board_Migros-Tycoon
- <https://www.meshy.ai> Generierung von Migros-Produkten als 3D-Modelle
- freesound.org – für frei verwendbare Soundeffekte

6.2 Glossar

Begriff	Erklärung
Tycoon	Spielgenre, bei dem man ein Geschäft oder eine Produktion aufbaut
Dropper	Objekt, das automatisch Spielteile generiert
Conveyor (Förderband)	Bewegt Blöcke automatisch Richtung Collector
Collector	Wandelt Blöcke in Spielgeld um
GUI (Graphical User Interface)	Benutzeroberfläche zur Anzeige von Infos und Interaktionen
Lua	Skriptsprache, die in Roblox zur Spielprogrammierung verwendet wird
LocalScript	Ein Roblox-Skript, das auf der Client-Seite ausgeführt wird
Dependency	Abhängigkeit zwischen kaufbaren Objekten

6.3 Programm-Code und Scripts

Alle wichtigen Objekte wurden im GitHub Repository dokumentiert:

https://github.com/finndomeisen/Abschlussprojekt_2025_API-2_findom_Board_Migros-Tycoon

6.4 Foto-Dokumentation

Screenshots aus dem Spiel und vom Entwicklungsprozess folgen auf den nächsten Seiten:

- Spielübersicht mit Umgebung
- Kaufbare Objekte & Dropper
- Benutzeroberfläche im Spiel
- Ladebildschirm & Tutorial-GUI
- GitHub-Projektstruktur

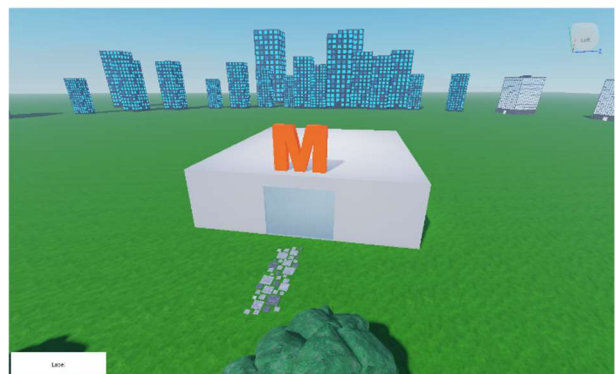


Abbildung 12 von aussen

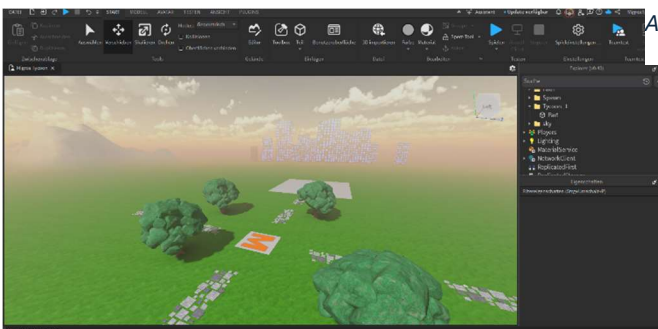


Abbildung 15 Spawnpunkt

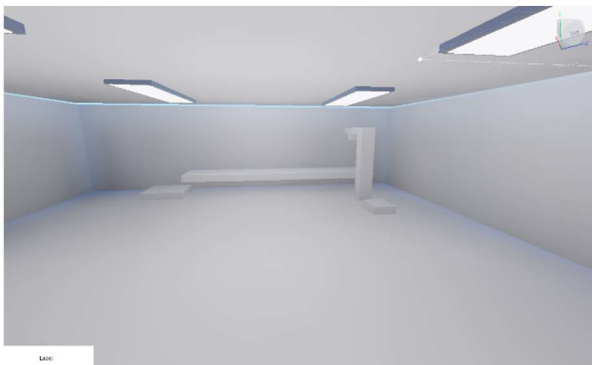


Abbildung 13 Von innen

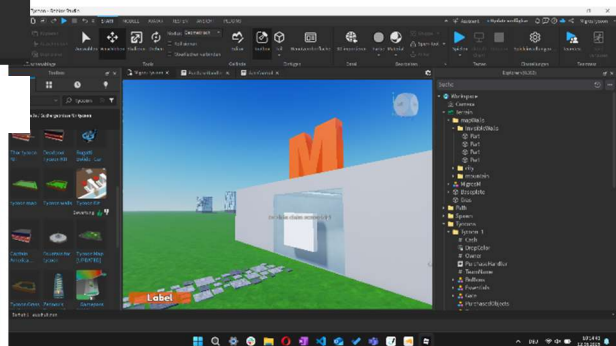


Abbildung 14 Touch to claim door