

Dokumentation & Projekttagbuch

Innovation Lab 1/2/3
Jahr 2023/24/25

Projekt: Serious Game – Computer Science

Team: 18

1. Allgemeine Informationen

Projektname: Serious Game - Computer Science

Supervisor: Petz Markus, MSc

Innovation Lab 1, Wintersemester 2023/24

Innovation Lab 2, Sommersemester 2024

Innovation Lab 3, Wintersemester 2024/25

Projektteam:

Johann Lang, if22b099@technikum-wien.at, Leader Gamedesign, Programming
Vincent Bernhart-Straberger, if23b188@technikum-wien.at, Leader Artist, Gamedesign
Dominik Neuwirth, if22b022@technikum-wien.at, Leader Programming
Florian Poppinger, if22b009@technikum-wien.at, Programming, Artist
Tobias Kastl, if22b007@technikum-wien.at, Artist, Management

Management-Summary des Projektes

Das Projekt zielt darauf ab, ein interaktives 2D-Rätselspiel für Schüler mit Interesse an Informatik zu entwickeln. Die Schwerpunkte liegen auf der Integration von UML-Editor-Funktionalitäten, der Steuerung von Robotern basierend auf UML-Ablaufdiagrammen, dem Level-Design und der Rätselentwicklung sowie der Sicherstellung einer intuitiven Benutzeroberfläche. Ziel ist es, eine unterhaltsame und lehrreiche Spielerfahrung zu schaffen, die es Schülern ermöglicht, ihre Kenntnisse in UML zu vertiefen und ihre Problemlösungsfähigkeiten zu verbessern.

Rahmenbedingungen und Projektumfeld

Umgebung: Unity, C#

Usability: „Demonstrationsspiel“, für Schüler/innen leicht verständlich

System-Anforderungen: WebBrowser, Keyboard, Maus

Semester-Roadmap

Innolab 1: Ideenfindung (40h), Unity-Basics (30h), Konzeptideen (100h), Arbeitspakete definieren (20h), Basis des Spiels implementieren (170h)

Innolab 2: Arbeitspakete definieren (20h), UML Erstellung (68h), Roboter Steuerung (76h), Progression (36h), Gefahren (40h), Siegbedingungen (32h)

Innolab 3: Arbeitspakete definieren (20h), UML Erstellung (93h), Roboter Steuerung (27h), Progression (32h), Gefahren (15h), Siegbedingungen (15h), UI (6h), Testen (44h),

Tutorial/Guide (26h), Einstellungen (5h), Sound (21h), Post Processing (10h)

Kollaboration & Tooling

Kollaboration: GIT als VCS

Clone: <https://git.technikum-wien.at/innolab/dieter-informatik.git>

Website Link: <https://git.technikum-wien.at/innolab/dieter-informatik/-/tree/InnoLab3>

Tooling: Sprintplanung in FH-Git

Design: Krita, Libre Sprite

Programming: Unity, Visual Studio

Kommunikation: Discord, Zoom, in Person an FH Technikum Wien

2. Projekt-Kurzbeschreibung

Überblick:

Unser Projektziel ist es, ein unterhaltsames und lehrreiches 2D-Rätselspiel zu entwickeln, das speziell auf Schüler mit Interesse an Informatik ausgerichtet ist. Wir möchten eine interaktive Umgebung schaffen, in der Spieler ihre Kenntnisse in UML-Ablaufdiagrammen vertiefen können, während sie gleichzeitig ihre Problemlösungsfähigkeiten durch das Lösen von Herausforderungen stärken.

Ziele und Zielsetzungen:

- Entwicklung eines funktionsfähigen Prototyps eines 2D-Rätselspiels in Unity mit C#, das auf dem Webbrowser läuft.
- Ermöglichen der Erstellung, Bearbeitung und Ausführung von UML-Ablaufdiagrammen innerhalb des Spiels.
- Integration von Robotersteuerung und -bewegung basierend auf den erstellten UML-Ablaufdiagrammen.
- Freischalten neuer Level und UML-Bausteine durch das Lösen von Rätseln.
- Sicherstellen einer intuitiven Benutzeroberfläche für Schüler und Schülerinnen.

Herausforderungen:

- Effektive Integration von UML-Editor-Funktionalitäten in das Spiel, um eine nahtlose Spielerfahrung zu gewährleisten.
- Gestaltung von Rätseln, die sowohl lehrreich als auch unterhaltsam sind und den pädagogischen Zielen entsprechen.
- Sicherstellung einer angemessenen Usability und Verständlichkeit für die Zielgruppe.
- Bewältigung technischer Herausforderungen bei der Implementierung von Robotersteuerung und -bewegung basierend auf UML-Ablaufdiagrammen.

Umfang:

- Entwicklung eines Prototyps des Spiels mit mehreren spielbaren Leveln.
- Implementierung grundlegender UML-Editor-Funktionen.
- Gestaltung von Rätseln und Herausforderungen für die Spieler.
- Sicherstellung einer intuitiven Benutzeroberfläche.
- Nicht-Ziele:
 - Entwicklung fortgeschrittener Grafiken oder komplexer Animationen.
 - Implementierung zusätzlicher Features, die über die grundlegenden Spielmechaniken und UML-Editor-Funktionen hinausgehen.

Implementierungsplan:

- Konzepterstellung und Designphase: Festlegung der Spielmechaniken, Erstellung von Storyboards und Skizzen.
- Entwicklung der UML-Editor-Funktionalitäten: Implementierung von Funktionen zur Erstellung, Bearbeitung und Ausführung von UML-Ablaufdiagrammen.

- Implementierung der Robotersteuerung und -bewegung: Integration von Funktionen zur Steuerung von Robotern basierend auf den erstellten UML-Ablaufdiagrammen.
- Level-Design und Rätsel-Entwicklung: Erstellung von verschiedenen Rätseln und Herausforderungen für die Spieler.
- Usability-Testing und Feedback: Durchführung von Tests mit Schülern und Lehrkräften, um Feedback einzuholen und Anpassungen vorzunehmen.
- Feinschliff und Optimierung: Behebung von Fehlern, Optimierung der Leistung und Verbesserung der Benutzeroberfläche.
- Präsentation des Prototyps: Vorstellung des funktionsfähigen Prototyps des Spiels am Ende des Semesters.

3. Spezifikation der Lösung

3.1. WS 23/24

Sprint 1

- Junior Programmer Pathway
- High Concept erstellen

Sprint 2

- Junior Programmer Pathway vervollständigen
- High Concept vervollständigen
- Projekt aufsetzen

Sprint 3

- Player-Movement-Basics
- Entity-Spawn
- Player Health
- Player Dash
- Level- / Ebenen-Design Prototyp

Sprint 4

- Angriff von Player & Enemies
- Enemy-Spawn
- Gegner designen

Sprint 5

- Angriff
- Reload
- Enemy-Health
- Raumplanung
- Inventar / Items planen
- UI Designs
- Maincharacter

Sprint 6

- Wände, Boden, Löcher
- Healthpacks auf der Map
- Enemy-Movement
- Räume, Inventory, Items planen
- Maincharacter, Wände, Waffen designen

3.2. SS 24

UML Erstellung	
	Als Spieler möchte ich ein UML-Ablaufdiagramm gestalten können, damit ich vorhandene Bausteine verwenden kann.
	Als Spieler möchte ich mit Funktionen im UML-Ablaufdiagramm feststellen können, was sich auf dem nächsten Feld befindet.
	Als Spieler möchte ich verschiedene UML-Ablaufdiagramm definieren können.
	Als Spieler möchte ich in der Lage sein, ein UML-Ablaufdiagramm in einem anderen aufrufen zu können.
	Als Spieler möchte ich in der Lage sein, UML-Ablaufdiagramm zu kopieren.
Roboter Steuerung	
	Als Spieler möchte ich den Roboter anhand von UML-Ablaufdiagrammen steuern können.
	Der Roboter soll sich auf dem Spielfeld bewegen können.
	Der Roboter soll Gegenstände aufsammeln können.
	Als Spieler möchte ich in der Lage sein, mehrere Roboter zu steuern.
	Als Spieler möchte ich den Startpunkt des Roboters ändern können.
Progression	
	Als Spieler möchte ich neue Level freischalten können.
	Als Spieler möchte ich durch Absolvieren von Leveln mehr UML-Bausteine freischalten können.
	Als Spieler möchte ich ein Level beliebig oft wiederholen können.
Gefahren	
	Der Roboter soll durch die Umgebung Schaden erleiden können.
	Als Spieler möchte ich die Leben des Roboters jederzeit einsehen können.
	Der Roboter soll in der Lage sein, Schadensquellen zu neutralisieren.
Siegbedingungen	
	Als Spieler muss ich das Level wiederholen, wenn der Roboter zu viel Schaden erleidet.
	Als Spieler habe ich ein Level erfolgreich abgeschlossen, wenn der Roboter alle Gegenstände gesammelt hat.
	Als Spieler möchte ich meinen Fortschritt jederzeit einsehen können.
	Als Spieler möchte ich meinen Erfolg anhand eines Highscores erkennen können.
	Als Spieler möchte ich den Müll an einem Ort sammeln können.

3.3. WS 24/25

Gelb hinterlegte Tasks wurden in InnoLab2 angefangen aber noch nicht abgeschlossen.

UML Erstellung	
	Als Spieler möchte ich ein UML-Ablaufdiagramm gestalten können, damit ich vorhandene Bausteine verwenden kann.
	Als Spieler möchte ich mit Funktionen im UML-Ablaufdiagramm feststellen können, was sich auf dem nächsten Feld befindet.
	Als Spieler möchte ich verschiedene UML-Ablaufdiagramm definieren können.
	Als Spieler möchte ich in der Lage sein, ein UML-Ablaufdiagramm in einem anderen aufrufen zu können.
	Als Spieler möchte ich in der Lage sein, UML-Ablaufdiagramm zu kopieren.
	Als Spieler möchte ich in der Lage sein, UML-Bäume zu speichern, damit ich meinen Fortschritt sichern kann.
	Als Spieler möchte ich in der Lage sein, in das UML-Diagramm hinein- und herauszuzoomen und eine Scrollfunktion nutzen können, um eine bessere Übersicht und Navigation zu ermöglichen.
Roboter Steuerung	
	Als Spieler möchte ich in der Lage sein, mehrere Roboter zu steuern.
	Als Spieler möchte ich den Startpunkt des Roboters ändern können.
Progression	
	Als Spieler möchte ich neue Level freischalten können.
	Als Spieler möchte ich durch Absolvieren von Leveln mehr UML-Bausteine freischalten können.
Gefahren	
	Der Roboter soll in der Lage sein, Schadensquellen zu neutralisieren.
Siegbedingungen	
	Als Spieler möchte ich meinen Fortschritt jederzeit einsehen können.
	Als Spieler möchte ich den Müll an einem Ort sammeln können.
UI	
	Als Spieler möchte ich einen Titelschirm sehen, wenn ich das Spiel starte.
	Als Spieler möchte ich die Credits einsehen können, um mehr über die Entwickler zu erfahren.
Testing	
	Als Entwickler möchte ich meinen Playtestern einen Fragebogen zum gesamten Spiel und möglichen zukünftigen Änderungen geben, um umfassendes Feedback zu erhalten.

	Als Entwickler möchte ich Playtester suchen und ihr Verhalten dokumentieren, um Aspekte zu erfassen, die den Testern möglicherweise nicht auffallen.
	Als Entwickler möchte ich die Fragebögen auswerten, um das Feedback der Playtester zu verstehen und Verbesserungen am Spiel vorzunehmen.
Tutorial / Guide	
	Als Spieler möchte ich Erklärungen zu den UML-Blöcken erhalten, damit ich weiß, wie man sie verwendet.
	Als Spieler möchte ich eine Erklärung der Benutzeroberfläche bekommen, um mich im Spiel zurechtzufinden.
	Als Spieler möchte ich eine Anleitung zu den Spielmechaniken, um zu verstehen, wie das Spiel funktioniert.
Settings	
	Als Spieler möchte ich die Lautstärke im Spiel anpassen können.
	Als Spieler möchte ich einen Dark Mode aktivieren können.
Sound	
	Als Spieler möchte ich einen Sound hören, wenn ich ein Level gewinne.
	Als Spieler möchte ich einen Sound hören, wenn ich ein Level verliere.
	Als Spieler möchte ich einen Sound hören, wenn der Roboter stirbt.
	Als Spieler möchte ich Sounds im Menü hören, wenn ich herumnavigiere.
	Als Spieler möchte ich einen Sound hören, wenn der Roboter eine Batterie aufnimmt.
	Als Spieler möchte ich spezifische Sounds für die UML-Blöcke hören, wenn ich sie verwende.
	Als Spieler möchte ich einen Sound hören, wenn ein Level geladen wurde, um zu wissen, dass das Spiel bereit ist.
	Als Spieler möchte ich einen Sound hören, wenn ein UML-Ablaufdiagramm gestartet wird.
	Als Spieler möchte ich Titelmusik hören, um den Einstieg in das Spiel angenehmer zu machen.
Post Processing	
	Als Spieler möchte ich Feuer-Effekte sehen, um das Spiel visuell ansprechender zu erleben.
	Als Spieler möchte ich visuelle Effekte sehen, wenn ich ein UML-Ablaufdiagramm erstelle, um die Bedeutung der Aktion zu unterstreichen.
	Als Spieler möchte ich Beleuchtungseffekte im Spiel haben, um die Atmosphäre zu verstärken.

4. Aufwandschätzung

Innolab 1

Ideenfindung (40h)
 Unity-Basics (30h)
 Konzeptideen (100h)
 Arbeitspakete definieren (20h)
 Basis des Spiels implementieren (170h)

= 360h / 5 Personen -> 72h/Person -> 6 Sprints -> 12h pro Person pro Sprint

Innolab 2

Es wurde die Excel-Vorlage aus dem Moodle Class 1 verwendet
 „DAGoPERT (Excel document, usage instructions as document and as screencast)“



DAGoPERT_Group_
 18.xlsx

= 360h / 5 Personen -> 72h/Person -> 6 Sprints -> 12h pro Person pro Sprint
 ~ 252h -> 360h * 0.7 (70% Planen)

Epics / User Stories / Arbeitspakete TopLevel	User Stories / Detail Level / Beschreibung	Optimisti sch (Sp)	Wahrsch einlich (Sp)	Pessimist isch (Sp)	Erwartet nach PERT 1:4:1 (Sp)	Aufwand (T- Shirt: XS-XXL)	Businesswert (T-Shirt: XS-XXL)	Nettobusiness- wert (T-Shirt: XS-XXL)
		172	252	372	258,6666667			
UML Erstellung								
	Als Spieler möchte ich ein UML-Ablaufdiagramm gestalten können, damit ich vorhandene Bausteine verwenden kann.	24	28	36	28,66666667	L	XXL	XXL
	Als Spieler möchte ich mit Funktionen im UML-Ablaufdiagramm feststellen können, was sich auf dem nächsten Feld befindet.	8	12	28	14	L	XL	XL
	Als Spieler möchte ich verschiedene UML-Ablaufdiagramm definieren können.	8	8	12	8,666666667	L	S	XS
	Als Spieler möchte ich in der Lage sein, ein UML-Ablaufdiagramm in einem anderen aufrufen zu können.	8	12	24	13,33333333	L	S	XS
	Als Spieler möchte ich in der Lage sein, UML-Ablaufdiagramm zu kopieren.	4	8	12	8	M	XS	XS

Roboter Steuerung								
	Als Spieler möchte ich den Roboter anhand von UML-Ablaufdiagrammen steuern können.	20	28	36	28	L	XXL	XXL
	Der Roboter soll sich auf dem Spielfeld bewegen können.	8	12	12	11,33333333	L	XXL	XXL
	Der Roboter soll Gegenstände aufsammeln können.	4	12	16	11,33333333	L	XL	XL
	Als Spieler möchte ich in der Lage sein, mehrere Roboter zu steuern.	12	16	32	18	L	XS	XS
	Als Spieler möchte ich den Startpunkt des Roboters ändern können.	4	8	16	8,66666667	L	XS	XS
Progression								
	Als Spieler möchte ich neue Level freischalten können.	12	16	24	16,66666667	L	L	L
	Als Spieler möchte ich durch Absolvieren von Leveln mehr UML-Bausteine freischalten können.	8	16	20	15,33333333	L	L	L
	Als Spieler möchte ich ein Level beliebig oft wiederholen können.	4	4	12	5,33333333	M	M	M
Gefahren								
	Der Roboter soll durch die Umgebung Schaden erleiden können.	8	16	20	15,33333333	L	M	S
	Als Spieler möchte ich die Leben des Roboters jederzeit einsehen können.	4	8	8	7,33333333	M	M	M
	Der Roboter soll in der Lage sein, Schadensquellen zu neutralisieren.	12	16	16	15,33333333	L	S	XS
Siegbedingungen								
	Als Spieler muss ich das Level wiederholen, wenn der Roboter zu viel Schaden erleidet.	4	4	8	4,66666667	M	M	M
	Als Spieler habe ich ein Level erfolgreich abgeschlossen, wenn der Roboter alle Gegenstände gesammelt hat.	4	4	8	4,66666667	M	M	M
	Als Spieler möchte ich meinen Fortschritt jederzeit einsehen können.	4	4	4	4	M	M	M
	Als Spieler möchte ich meinen Erfolg anhand eines Highscores erkennen können.	8	8	12	8,66666667	L	S	XS
	Als Spieler möchte ich den Müll an einem Ort sammeln können.	4	12	16	11,33333333	L	XS	XS

Innolab 3

Es wurde die Excel-Vorlage aus dem Moodle Class 1 verwendet
 „DAGoPERT (Excel document, usage instructions as document and as screencast)“



DAGoPERT_Group_
18.xlsx

Gelb hinterlegte Tasks wurden in InnoLab2 angefangen aber noch nicht abgeschlossen.

= 360h / 5 Personen -> 72h/Person -> 7 Sprints -> 12h pro Person pro Sprint
 ~ 294h -> 420h * 0.7 (70% Planen)

Epics / User Stories / Arbeitspakete TopLevel	User Stories / Detail Level / Beschreibung	Optimist isch (Sp)	Wahrsch einlich (Sp)	Pessimi stisch (Sp)	Erwartet nach PERT 1:4:1 (Sp)	Aufwand (T- Shirt: XS-XXL)	Businesswert (T- Shirt: XS-XXL)	Nettobusiness- wert (T-Shirt: XS-XXL)
		204	282	432	294			
UML Erstellung								
	Als Spieler möchte ich ein UML-Ablaufdiagramm gestalten können, damit ich vorhandene Bausteine verwenden kann.	24	28	36	28,66666667	L	XXL	XXL
	Als Spieler möchte ich mit Funktionen im UML-Ablaufdiagramm feststellen können, was sich auf dem nächsten Feld befindet.	8	12	28	14	L	XL	XL
	Als Spieler möchte ich verschiedene UML-Ablaufdiagramm definieren können.	8	8	12	8,666666667	L	S	XS
	Als Spieler möchte ich in der Lage sein, ein UML-Ablaufdiagramm in einem anderen aufrufen zu können.	8	12	24	13,33333333	L	S	XS
	Als Spieler möchte ich in der Lage sein, UML-Ablaufdiagramm zu kopieren.	4	8	12	8	M	XS	XS
	Als Spieler möchte ich in der Lage sein, UML-Bäume zu speichern, damit ich meinen Fortschritt sichern kann.	8	12	16	12	L	M	S
	Als Spieler möchte ich in der Lage sein, in das UML-Diagramm hinein- und herauszuzoomen und eine Scrollfunktion nutzen können, um eine bessere Übersicht und Navigation zu ermöglichen.	4	8	12	8	M	L	L

Roboter Steuerung								
	Als Spieler möchte ich in der Lage sein, mehrere Roboter zu steuern.	12	16	32	18	L	XS	XS
	Als Spieler möchte ich den Startpunkt des Roboters ändern können.	4	8	16	8,666666667	L	XS	XS
Progression								
	Als Spieler möchte ich neue Level freischalten können.	12	16	24	16,66666667	L	L	L
	Als Spieler möchte ich durch Absolvieren von Leveln mehr UML-Bausteine freischalten können.	8	16	20	15,33333333	L	L	L
Gefahren								
	Der Roboter soll in der Lage sein, Schadensquellen zu neutralisieren.	12	16	16	15,33333333	L	S	XS
Siegbedingungen								
	Als Spieler möchte ich meinen Fortschritt jederzeit einsehen können.	4	4	4	4	M	M	M
	Als Spieler möchte ich den Müll an einem Ort sammeln können.	4	12	16	11,33333333	L	XS	XS
UI								
	Als Spieler möchte ich einen Titelbildschirm sehen, wenn ich das Spiel starte.	2	4	6	4	M	XL	XL
	Als Spieler möchte ich die Credits einsehen können, um mehr über die Entwickler zu erfahren.	2	2	4	2,333333333	M	M	M

Testing								
	Als Entwickler möchte ich meinen Playtestern einen Fragebogen zum gesamten Spiel und möglichen zukünftigen Änderungen geben, um umfassendes Feedback zu erhalten.	4	8	12	8	M	XXL	XXL
	Als Entwickler möchte ich Playtester suchen und ihr Verhalten dokumentieren, um Aspekte zu erfassen, die den Testern möglicherweise nicht auffallen.	20	20	40	23,33333333	L	XXL	XXL
	Als Entwickler möchte ich die Fragebögen auswerten, um das Feedback der Playtester zu verstehen und Verbesserungen am Spiel vorzunehmen.	12	12	16	12,66666667	L	XXL	XXL
Tutorial / Guide								
	Als Spieler möchte ich Erklärungen zu den UML-Blöcken erhalten, damit ich weiß, wie man sie verwendet.	12	18	24	18	L	XXL	XXL
	Als Spieler möchte ich eine Erklärung der Benutzeroberfläche bekommen, um mich im Spiel zurechtzufinden.	2	4	4	3,66666667	M	XXL	XXL
	Als Spieler möchte ich eine Anleitung zu den Spielmechaniken, um zu verstehen, wie das Spiel funktioniert.	2	4	8	4,33333333	M	XXL	XXL

Settings								
	Als Spieler möchte ich die Lautstärke im Spiel anpassen können.	2	2	4	2,333333333	M	M	M
	Als Spieler möchte ich einen Dark Mode aktivieren können.	2	2	4	2,333333333	M	XS	XS
Sound								
	Als Spieler möchte ich einen Sound hören, wenn ich ein Level gewinne.	2	2	2	2	S	M	M
	Als Spieler möchte ich einen Sound hören, wenn ich ein Level verliere.	2	2	2	2	S	M	M
	Als Spieler möchte ich einen Sound hören, wenn der Roboter stirbt.	2	4	4	3,666666667	M	M	M
	Als Spieler möchte ich Sounds im Menü hören, wenn ich herum navigiere.	2	2	2	2	S	S	S
	Als Spieler möchte ich einen Sound hören, wenn der Roboter eine Batterie aufnimmt.	2	2	2	2	S	S	S
	Als Spieler möchte ich spezifische Sounds für die UML-Blöcke hören, wenn ich sie verwende.	2	2	2	2	S	XS	XS
	Als Spieler möchte ich einen Sound hören, wenn ein Level geladen wurde, um zu wissen, dass das Spiel bereit ist.	2	2	2	2	S	XS	XS
	Als Spieler möchte ich einen Sound hören, wenn ein UML-Ablaufdiagramm gestartet wird.	2	2	2	2	S	S	S
	Als Spieler möchte ich Titelmusik hören, um den Einstieg in das Spiel angenehmer zu machen.	2	4	4	3,666666667	M	M	M
Post Processing								
	Als Spieler möchte ich Feuer-Effekte sehen, um das Spiel visuell ansprechender zu erleben.	2	2	4	2,333333333	M	S	XS
	Als Spieler möchte ich visuelle Effekte sehen, wenn ich ein UML-Ablaufdiagramm erstelle, um die Bedeutung der Aktion zu unterstreichen.	2	4	12	5	M	S	XS
	Als Spieler möchte ich Beleuchtungseffekte im Spiel haben, um die Atmosphäre zu verstärken.	2	2	4	2,333333333	M	S	XS

5. Auslieferung

Das Projekt befindet sich sowohl in unserem privaten Innolab-Git-Repository, als auch im Git-Repository der FH Technikum Wien.

Das Spiel sollte in gängigen Webbrowsern wie Google Chrome, Mozilla Firefox, Safari und Microsoft Edge laufen.

Das Spiel erfordert eine zuverlässige Internetverbindung für den Zugriff auf die Spielressourcen und die Interaktion mit anderen Spielern.

Browsergames sind plattformunabhängig und sollten auf verschiedenen Betriebssystemen wie Windows, macOS und Linux laufen.

Da Browsergames in der Regel weniger ressourcenintensiv sind als herkömmliche Spiele, sind die Hardwareanforderungen oft niedrig. Allerdings sollte das Spiel auf verschiedenen Geräten, einschließlich älterer Computer, ohne Leistungsprobleme laufen.

6. Projekt-Tagebuch

Sprint 1

Im ersten Sprint haben wir diverse Meetings und Arbeitsaufgaben durchgeführt, um einen soliden Startpunkt zu schaffen.

Meetings

- Zoom-Meetings: Wir haben insgesamt zwei Zoom-Meetings abgehalten. Im ersten wurden die grundlegenden Informationen zur LV erläutert, im zweiten Meeting wurde die erste Version dieses Dokuments mit Herrn Petz besprochen.
- FH-Meetings: Es gab zwei FH-Meetings, bei denen wir uns kennengelernt und grobe Ideensammlungen durchgeführt haben. Ein weiteres Meeting fand statt, um organisatorische Themen zu besprechen und den Sprint zu evaluieren.
- Discord-Meeting: Ein Meeting auf Discord konzentrierte sich auf organisatorische Angelegenheiten und die Besprechung des Sprints. Ein zweites Meeting wurde wegen der Erstellung des Project Diaries abgehalten.

Tasks

- Junior Programmer Pathway 1/2: Diese Arbeitsaufgabe dauerte insgesamt 16 Stunden und wurde von allen Teammitgliedern außer Johann bearbeitet. Es handelte sich um die Beschäftigung mit dem Junior Programmer Pathway, um die Fähigkeiten aller Teammitglieder zu stärken.
- High Concept: Johann war für die Ausarbeitung des High Concepts verantwortlich und investierte dafür 4 Stunden.

Evaluierung

Der erste Sprint verlief insgesamt reibungslos. Alle geplanten Meetings und Arbeitsaufgaben wurden erfolgreich durchgeführt, und es gab keine Abweichungen vom geschätzten Aufwand.

Sprint 2

Im zweiten Sprint haben wir erneut eine Reihe von Meetings und Arbeitsaufgaben durchgeführt, um die Fortschritte des Projekts sicherzustellen und neue Aspekte zu entwickeln.

Meetings

- Zoom-Meeting: Ein Meeting fand statt, um den Review des ersten Sprints zu besprechen und die nächsten Schritte zu planen.
- FH-Meeting: Ein FH-Meeting wurde abgehalten, um generelle Kunst-Design- und Richtungsaspekte zu klären und zu diskutieren.
- Discord-Meeting: Ein Meeting auf Discord konzentrierte sich auf organisatorische Themen und die Besprechung des zweiten Sprints.

Tasks

- Junior Programmer Pathway 2/2: Das Unity-Tutorial wurde von allen abgeschlossen.
- High Concept: Johann arbeitete weiterhin am High Concept, konnte es jedoch noch nicht abschließen.
- Gamedesign-Skizzen: Johann arbeitete an Gamedesign-Skizzen, bemerkte jedoch, dass die verwendete Software (Krita) nicht seinen Anforderungen entspricht.
- Design des Hauptcharakters: Tobias und Vincent waren mit den bisherigen Ergebnissen nicht zufrieden und setzten sich weiterhin mit dem Design des Hauptcharakters auseinander.
- Projekt aufsetzen: Florian und Dominik beschäftigten sich mit der Einrichtung des Projekts, insbesondere mit der Verwendung von Unity Cloud und Git.

Evaluierung

Der zweite Sprint verlief ebenfalls erfolgreich. Johann konnte das High Concept noch nicht fertigstellen, und es gab einige Unzufriedenheiten bezüglich der verwendeten Software für Gamedesign-Skizzen und dem Charakterdesign.

Es wurde generell weniger Tasks geplant, da in diesen zwei Wochen mehrere Prüfungen stattgefunden haben.

Sprint 3

Im dritten Sprint haben wir verschiedene Meetings und Arbeitsaufgaben durchgeführt, um die Entwicklung des Projekts voranzutreiben.

Meetings

- Zoom-Meeting: Ein Meeting wurde abgehalten, um den Review des zweiten Sprints durchzuführen und Erfahrungen auszutauschen.
- Discord-Meeting (Arbeitspakete): Ein ausführliches Meeting fand statt, um Arbeitspakete zu definieren und den nächsten Entwicklungsschritte zu planen.
- Discord-Meeting (Organisation): Ein Meeting konzentrierte sich auf organisatorische Themen und die Besprechung des dritten Sprints.
- Discord-Meeting (Game-Design/Artstyle): Johann und Vincent diskutierten über das Game-Design und den Artstyle.
- Discord-Meeting (Tilemap-Design): Vincent und Tobias arbeiteten am Tilemap-Design und besprachen die Fortschritte.

Arbeitsaufgaben

- Design: Theme Floor A: Vincent und Tobias arbeiteten am Design für den Boden des ersten Levels.
- Sketch: Ground Floor A: Vincent und Tobias erstellten Skizzen für den Boden des ersten Levels. Es gab eine leichte Abweichung von der geschätzten Arbeitszeit.
- Art: Ground Floor A: Vincent setzte den Stil für den Boden des ersten Levels um, wobei mehr Zeit als erwartet benötigt wurde.
- High Concept: Johann arbeitete am High Concept weiter, die geschätzte Zeit wurde eingehalten.
- Programming: Player Basic Movement: Dominik und Florian arbeiteten an der grundlegenden Bewegung des Spielers.
- Programming: Entity Spawn: Dominik und Florian implementierten die Erzeugung von Entitäten im Spiel.
- Programming: Player Health (Code / UI): Florian arbeitete am Spielerlebenssystem, benötigte dafür etwas weniger Zeit als geplant.
- Programming: Player Dash: Dominik implementierte die Dash-Fähigkeit des Spielers, benötigte dafür etwas weniger Zeit als geplant.

Evaluierung

Der dritte Sprint war erfolgreich und brachte Fortschritte in verschiedenen Aspekten des Projekts. Die Diskussionen und Meetings trugen dazu bei, die nächsten Schritte zu klären. In einigen Fällen gab es leichte Abweichungen von den geschätzten Zeiten.

Der Arbeitsaufwand in diesem Sprint war aufgrund des vorigen Sprints etwas höher geplant.

Sprint 4

Im vierten Sprint haben wir wiederum Meetings und Arbeitsaufgaben durchgeführt, um die kontinuierliche Entwicklung unseres Projekts sicherzustellen.

Meetings

- Zoom-Meeting: Ein Meeting wurde abgehalten, um den Review des dritten Sprints durchzuführen und Erfahrungen zu teilen.
- Discord-Meeting (Organisation): Ein Meeting fokussierte sich auf organisatorische Themen und die Besprechung des vierten Sprints.
- Discord-Meeting (Git-Projekt): Dominik und Florian setzten sich zusammen, um das Git-Projekt zu besprechen und zu organisieren.
- Discord-Meeting (Artstyle-Anpassungen): Vincent und Tobias überarbeiteten den Artstyle, machten ihn simpler und passten die Perspektive an.
- Discord-Meeting (Tilemap-Überarbeitung): Vincent und Tobias überarbeiteten die Tilemap, redeten über die Skalierung der Sprites und passten die Perspektive an.

Arbeitsaufgaben

- Programming: Attack Hitbox Trigger: Dominik und Florian arbeiteten an der Auslösung der Angriffs-Hitbox, benötigten dafür etwas weniger Zeit als geplant.
- Programming: Projectile Behavior: Dominik setzte das Verhalten von Geschossen um und las die Mausposition aus, wofür etwas mehr Zeit benötigt wurde.
- Programming: AOE Behavior: Dominik arbeitete am Verhalten von Flächenangriffen, benötigte dafür etwas weniger Zeit als geplant.
- Programming: Player Dash: Dominik implementierte die Dash-Fähigkeit des Spielers und benötigte dafür die geplante Zeit.
- Programming: Enemy Spawner: Florian implementierte den Spawner für Feinde und benötigte dafür etwas weniger Zeit als geplant.
- Programming: Enemy Collisions: Florian arbeitete an der Kollisionslogik der Feinde, benötigte dafür die geplante Zeit.
- Sketch: Trojaner: Tobias richtete sein Tablet ein und erstellte Skizzen für den Trojaner, benötigte dafür etwas mehr Zeit.
- Sketch: Virus: Vincent erstellte Skizzen für den Virus, benötigte dafür etwas weniger Zeit.
- Design: Trojaner, Spyware, Virus, Ransomware: Johann arbeitete an den Designs für verschiedene Feinde und benötigte dafür die geplante Zeit.

Evaluierung

Der vierte Sprint war überwiegend erfolgreich. Es gab leichte Abweichungen von den geschätzten Zeiten, die jedoch im Rahmen der Flexibilität des agilen Entwicklungsumfelds akzeptiert wurden. Die Anpassungen am Artstyle und die Überarbeitung der Tilemap tragen zur Verbesserung der visuellen Aspekte des Spiels bei.

Sprint 5

Im fünften Sprint setzten wir unsere Meetings und Arbeitsaufgaben fort, um die Entwicklung unseres Projekts voranzutreiben.

Meetings

- Zoom-Meeting: Ein Meeting fand statt, um den Review des vierten Sprints durchzuführen und Erfahrungen zu teilen.
- Discord-Meeting (Organisation): Ein Meeting konzentrierte sich auf organisatorische Themen und die Besprechung des fünften Sprints.
- Discord-Meeting (Gegnerdesign): Vincent und Johann diskutierten über das Design der Gegner.
- Discord-Meeting (Health System/Healthbar): Ein Meeting beinhaltete Diskussionen über das Gesundheitssystem und die Healthbar, in einem weiteren Meeting wurden speziell Designüberlegungen für die Healthbar besprochen.
- Discord-Meeting (Räume Feedback): Vincent und Johann gaben Feedback zum Design der Räume.
- Discord-Meeting (Healthbar in Unity): Dominik und Tobias besprachen die Integration der Healthbar in Unity.

Arbeitsaufgaben

- Programming: Player Weapons Fire: Dominik und Florian arbeiteten an der Implementierung des Feuerns von Waffen des Spielers. Neue Designideen wurden diskutiert, was zu einer leichten Zeitüberschreitung führte.
- Programming: Player Weapons Reload: Dominik implementierte das Nachladen der Waffen des Spielers.
- Programming: Enemy Movement: Florian arbeitete an der Bewegung der Feinde, hatte jedoch mit einem Versionskonflikt in Unity zu kämpfen.
- Programming: Enemy Health (Code / UI): Florian arbeitete an der Gesundheit der Feinde, sowohl am Code als auch an der Benutzeroberfläche.
- Programming: Enemy Attacks: Dominik implementierte die Angriffe der Feinde.
- Design: Räume planen: Johann arbeitete an der Planung der Levelräume, was länger dauerte als erwartet.
- Design: Inventory/Items planen: Johann arbeitete an der Planung des Inventars und der Gegenstände, was ebenfalls mehr Zeit in Anspruch nahm.
- Sketch: Main Character: Vincent und Tobias arbeiteten an Skizzen für den Hauptcharakter, was länger dauerte als erwartet.
- Sketch: Healthbar: Tobias änderte seine Designentscheidung für die Healthbar und erstellte neue Skizzen.
- Sketch: Walls Floor A: Vincent und Tobias arbeiteten an Skizzen für Wände des ersten Levels.
- Sketch: Bit -> Bat: Vincent erstellte eine Skizze für eine Waffe.

Evaluierung

Der fünfte Sprint war anspruchsvoll, da einige Arbeitsaufgaben länger dauerten als erwartet. Die Diskussionen und Designüberlegungen für die Healthbar trugen jedoch dazu bei, das Spiel weiter zu verbessern.

Sprint 6

Im sechsten Sprint haben wir unsere Meetings und Arbeitsaufgaben fortgesetzt, um die Entwicklung unseres Projekts voranzutreiben.

Meetings

- Zoom-Meeting (Review & Präsentation): Ein Meeting wurde abgehalten, um den Review des fünften Sprints durchzuführen und eine Präsentation vorzubereiten.
- Discord-Meeting (Organisation): Ein Meeting konzentrierte sich auf organisatorische Themen und die Besprechung des sechsten Sprints.
- Discord-Meeting (Merge on Dev, Gegnerverhalten): Alle Teammitglieder diskutierten über das Zusammenführen auf dem Entwicklungsbranch und das Verhalten der Gegner.
- Discord-Meeting (Präsentationsvorbereitung): Ein umfangreiches Meeting wurde abgehalten, um die Präsentation vorzubereiten.
- Discord-Meeting (Spielkonzept - Serious Game): Diskussionen über das Spielkonzept als "Serious Game".
- Discord-Meeting (Review + Video): Ein abschließendes Meeting fand statt, um den Review des sechsten Sprints durchzuführen und das Sprintvideo zu erstellen.

Arbeitsaufgaben

- PowerPoint & Video: Johann war für die Erstellung der PowerPoint-Präsentation und des Sprintvideos verantwortlich.
- Programming: Wall, Ground, Hole Tiles: Dominik arbeitete an der Programmierung von Wand-, Boden- und Loch-Tiles, hatte jedoch Probleme mit Texturproblemen.
- Programming: Health Packs: Florian implementierte Health Packs im Spiel.
- Programming: Enemy Movement: Florian setzte die Feindbewegungen um, die aus dem vorherigen Sprint stammen.
- Design: Räume planen: Johann arbeitete weiterhin an der Planung der Levelräume, es gab jedoch keine Fortschritte.
- Design: Inventory/Items planen: Johann arbeitete weiterhin an der Planung des Inventars und der Gegenstände, es gab jedoch keine Fortschritte.
- Sketch: Main Character: Vincent und Tobias arbeiteten an Skizzen für den Hauptcharakter, es gab jedoch nur geringe Fortschritte im Vergleich zum vorherigen Sprint.
- Sketch: Walls Floor A: Vincent und Tobias arbeiteten an Skizzen für Wände auf dem Boden des ersten Levels, jedoch mit geringen Fortschritten.
- Sketch: Bit -> Bat: Vincent erstellte eine Skizze für die Verwandlung von Bits in Fledermäuse.
- Art: Walls Floor A: Dominik und Tobias integrierten Texturen für Wände auf dem Boden des ersten Levels in Unity.

Evaluierung

Es gab bedeutende Fortschritte in der Präsentationsvorbereitung und der Integration von Wandtexturen in Unity.

Sprint 7

Mit dem siebten Sprint begann das zweite Semester von Innolab.

Meetings

- Discord-Meeting: Besprechung der Arbeitsaufteilung.
- Discord-Meeting: Diskussion über das Game Design Document.
- Discord-Meeting: Durchführung des Sprint Reviews.

Tasks

- Demo Graphics: Johann erstellte Demo-Grafiken.
- Main Character Update: Vincent führte ein Update des Hauptcharakters durch.
- Basic Level Setup: Tobias und Florian arbeiteten an der grundlegenden Einrichtung des ersten Levels.
- Character Move on Tiles: Florian implementierte die Bewegung des Charakters auf den Tiles.
- Character Pick Up Trash: Dominik implementierte die Funktion, dass der Charakter Müll aufsammeln kann.
- Drag and Drop UI Element: Vincent arbeitete an der Drag-and-Drop-Funktion für UI-Elemente.
- Copy UI Element/Prefab: Dominik kopierte UI-Elemente und Prefabs vom alten Projekt.
- Call Character Function from UI: Johann implementierte die Funktion, dass Charakteraktionen von der UI aus aufgerufen werden können.
- Sketch of Main View/UI: Tobias und Vincent erstellten Skizzen für die Hauptansicht und UI.

Evaluierung

Der siebte Sprint verlief erfolgreich. Alle geplanten Meetings und Arbeitsaufgaben wurden durchgeführt, und es gab keine Abweichungen vom geschätzten Aufwand.

Sprint 8

Meetings

- Discord-Meeting: Planung des Sprints.
- Discord-Meeting: Neue Definition der Arbeitspakete.
- Discord-Meeting: Durchführung des Sprint Reviews und Diskussion über die Projektidee.

Tasks

- UML-Element zur Auswahl: Florian arbeitete an einem UML-Element, das das Bewegen des Roboters ermöglicht.
- Roboter agiert anhand Diagramms: Dominik und Florian arbeiteten daran, dass der Roboter anhand des erstellten Diagramms agiert.
- UML-Ablaufdiagramm starten: Florian implementierte die Funktion, das UML-Ablaufdiagramm zu starten.
- Level zurücksetzen: Dominik arbeitete daran, dass das Level zurückgesetzt werden kann, um das UML-Ablaufdiagramm anzupassen und erneut zu versuchen.
- Aktuelles UML-Element erkennen: Dominik implementierte die Funktion, um zu erkennen, welches UML-Element gerade ausgeführt wird.
- Roboter hält vor Wand an: Florian sorgte dafür, dass der Roboter automatisch anhält, bevor er gegen eine Wand läuft.
- UML-Elemente mit Pfeilen verbinden: Vincent arbeitete an der Funktion, UML-Elemente mit Pfeilen zu verbinden, um den Ablauf klar zu erkennen.
- UML-Elemente verschieben: Tobias implementierte die Möglichkeit, UML-Elemente mittels Drag and Drop zu verschieben.
- UML-Elemente über PopUp erstellen: Tobias implementierte die Funktion, UML-Elemente über ein PopUp-Fenster zu erstellen.
- UML-Elemente über Vorlage erstellen: Vincent arbeitete daran, UML-Elemente über Drag and Drop einer Vorlage zu erstellen.
- Schadensquellen konfrontieren: Tobias implementierte Schadensquellen, um schwierigere Rätsel zu ermöglichen.
- Wasser als Gefahr: Tobias machte Wasser zu einer Gefahr für die Roboter.
- Begrenzte Leben für Roboter: Johann implementierte die Funktion, dass Roboter eine begrenzte Anzahl an Leben haben.
- Leben eines Roboters einsehen: Johann sorgte dafür, dass die Leben eines Roboters jederzeit eingesehen werden können.
- Anzeige bei Level-Abschluss: Johann implementierte eine Anzeige, die erscheint, wenn ein Level nicht erfolgreich absolviert wurde, sowie eine Anzeige für den erfolgreichen Abschluss.
- Anzahl des Mülls einsehen: Johann arbeitete daran, dass die Anzahl des gesammelten und noch verbleibenden Mülls eingesehen werden kann.

Evaluierung

Der achte Sprint verlief erfolgreich. Alle geplanten Meetings und Arbeitsaufgaben wurden durchgeführt, und es gab keine wesentlichen Abweichungen vom geschätzten Aufwand.

Sprint 9

Meetings

- Discord-Meeting: Durchführung des Sprint Reviews.
- Discord-Meeting: Es traten Fehler und Fragen beim Mergen auf.
- Zoom-Meeting: Review des vergangenen Sprints und Planung des aktuellen Sprints.
- Discord-Meeting: Besprechung eines neuen Git-Ignore und Git-Merges.

Tasks

- Grafiken erstellt: Johann erstellte neue Grafiken.
- Git Ignore und Merge Conflicts: Dominik löste Probleme mit Merges und Git-Ignore.
- UML-Element zur Auswahl: Florian arbeitete an einem UML-Element, das das Bewegen des Roboters ermöglicht.
- Roboter agiert anhand Diagramm: Dominik entschied sich für die Verwendung eines Enums. Florian unterstützte die Implementierung.
- Level zurücksetzen: Dominik arbeitete daran, dass das Level zurückgesetzt werden kann. Dies wurde mit Task 186 abgeschlossen.
- Aktuelles UML-Element erkennen: Dominik implementierte die Funktion zur Erkennung des aktuell ausgeführten UML-Elements. Dies wurde mit Task 186 abgeschlossen.
- UML-Elemente mit Pfeilen verbinden: Vincent löste das Problem des Verbindens.
- UML-Elemente über PopUp erstellen: Tobias legte diese Aufgabe vorerst auf Eis.
- UML-Elemente über Vorlage erstellen: Vincent implementierte die Drag-and-Drop-Funktion für UML-Elemente aus einer Vorlage.
- Anzeige bei Levelabschluss (erfolglos): Johann implementierte die Anzeige für nicht erfolgreich absolvierte Level.
- Anzeige bei Levelabschluss (erfolgreich): Johann implementierte die Anzeige für erfolgreich absolvierte Level.
- Konstante Tick-Rate für UML-Elemente: Dominik setzte die Abarbeitung der UML-Elemente mit einer konstanten Tick-Rate um, basierend auf Coroutinen.
- Statusanzeige für Roboter und UML-Ablauf: Dominik implementierte die Statusanzeige.
- Fehlererkennung im UML-Ablauf: Florian implementierte eine erweiterbare Konsole zur Fehlererkennung.
- Freischaltung neuer UML-Elemente: Vincent begann mit der Implementierung.
- Mindestens 3 unterschiedliche Level: Tobias und Johann planten die Level, die Implementierung in Unity steht noch aus.
- Erkennung des nächsten Feldes durch Roboter: Dominik implementierte diese Funktion.
- Level wechseln: Florian implementierte die Möglichkeit, das Level zu wechseln.
- Startbildschirm zur Levelauswahl: Johann begann mit einem Startbildschirm.
- Leistungsrating: Tobias und Johann implementierten die Darstellung der Leistung.
- Bewegliche Hindernisse in Level: Tobias implementierte bewegliche Hindernisse.
- UML-Element zum Müll aufsammeln: Vincent implementierte ein UML-Element, das den Roboter Müll aufsammeln lässt.

Evaluierung

Einige unvollständige Aufgaben vom letzten Sprint wurden abgeschlossen.

Sprint 10

Meetings

- Discord-Meeting: Durchführung des Sprint Reviews.
- Discord-Meeting: Besprechung des Git-Merges bezüglich Prefabs (Soundeffects).
- Zoom-Meeting: Planung des aktuellen Sprints und Durchführung des Merges.

Tasks

- Neue UML-Elemente freischalten: Vincent implementierte die Funktion, dass im Spiel neue UML-Elemente freigeschaltet werden können.
- Mindestens 3 unterschiedliche Level spielen können: Tobias und Johann arbeiteten an der Implementierung, wobei die Unity-Implementierung noch aussteht.
- Startbildschirm zur Levelauswahl: Johann begann mit der Arbeit an einem Startbildschirm zur Auswahl der Level.
- Passende Hintergrundmusik im Spiel hören: Florian implementierte die Funktion, dass passende Hintergrundmusik im Spiel abgespielt wird.
- Soundeffekte für bessere Spielwahrnehmung: Florian und Tobias integrierten Soundeffekte für eine verbesserte Spielwahrnehmung.
- Roboter mit begrenzter Batterie: Dominik implementierte eine Batterie, die die Anzahl der Aktionen des Roboters beschränkt.
- Batteriestatusanzeige für Roboter: Johann implementierte die Anzeige für den Batteriestatus der Roboter.
- Fortschrittsspeicherung nach Spielende: Johann begann mit der Implementierung der Funktion, dass der Fortschritt auch nach dem Schließen des Spiels gespeichert bleibt.
- Bedingungs-Blöcke mit 2 Blöcken verbinden: Vincent entwickelte die Möglichkeit, Bedingungs-Blöcke mit 2 Blöcken zu verbinden, um sowohl erfüllte als auch nicht erfüllte Bedingungen darzustellen.
- Anpassung an unterschiedliche Bildschirmgrößen: Florian arbeitete an der Anpassung des Spiels auf unterschiedliche Bildschirmgrößen, inklusive der Implementierung von Scrollbars.
- Wiederholung von Teilen des UML-Ablaufdiagramms: Dominik implementierte die Funktion, dass Teile des UML-Ablaufdiagramms eine bestimmte Anzahl wiederholt werden können.
- Felder im Spiel, die Roboter in eine bestimmte Richtung verschieben: Dominik arbeitete an der Dynamisierung der Tickrate für UML-Elemente, um lange Bewegungen zu verhindern.
- Level zur Auswahl, das Wissen über Bedingungen und Schleifen testet: Tobias implementierte ein Level, das das Wissen über Bedingungen und Schleifen auf die Probe stellt.

Evaluierung

Der zehnte Sprint war erfolgreich. Alle geplanten Meetings wurden durchgeführt und die meisten Aufgaben wurden abgeschlossen. Einige Aufgaben wurden begonnen, aber werden im nächsten Sprint fortgesetzt. Es gab keine wesentlichen Abweichungen vom geschätzten Aufwand.

Sprint 11

Meetings

- Discord-Meeting: Durchführung des Sprint Reviews.
- Discord-Meeting: Besprechung und Durchführung von Aufräumarbeiten im Code.
- Discord-Meeting: Planung des aktuellen Sprints und Durchführung des Merges.
- Zoom-Meeting: Besprechung mit Petz.

Tasks

- Fortschrittsspeicherung: Johann implementierte die Funktion, dass der Fortschritt des Spielers nach dem Schließen des Spiels gespeichert bleibt.
- Anpassung an unterschiedliche Bildschirmgrößen: Florian passte das Spiel an, damit es auf verschiedenen Bildschirmgrößen läuft.
- Ersetzen von Sprites und Meshes in Unity: Johann ersetzte Sprites und Meshes in der Unity-Umgebung.
- Implementierung eines Arrow Sprites: Johann integrierte einen neuen Arrow Sprite ins Spiel.
- Freischalten neuer Level: Johann implementierte die Funktion, dass neue Level freigeschaltet werden, wenn ein vorheriges erfolgreich abgeschlossen wurde.
- Visual Effects (VFX) bei Roboterbewegung: Florian fügte Visual Effects hinzu, die sichtbar sind, wenn sich ein Roboter bewegt.
- VFX beim Müllsammeln: Florian implementierte Visual Effects für das Sammeln von Müll durch Roboter.
- VFX bei Roboterbewegung ins Wasser: Florian fügte Visual Effects hinzu, wenn ein Roboter ins Wasser läuft.
- Level mit bewegenden Gegnern: Tobias entwickelte ein Level, das bewegende Gegner enthält.
- Präzise Spielabläufe: Dominik sorgte dafür, dass alle Bewegungen und Aktionen im Spiel einem genauen Ablauf folgen.
- Batterieaufladung für Roboter: Dominik implementierte die Funktion, dass die Batterie eines Roboters wieder aufgeladen werden kann.
- Code + Unity Cleanup:
 - Dominik: Verbesserte das Resetting, trennte die Batterie von UML Actor ab und entfernte unnötige GameManager Variablen und Funktionen.
 - Tobias: Erstellte Level Presets, entfernte unnötige Prefabs und ersetzte White Boxes mit Sprites.
 - Vincent: Behebung der draw upwards arrow Detection und Bereinigung der drawArrow Funktion.
- Änderung der Funktion eines UML-Elements per Dropdown: Vincent implementierte die Möglichkeit, die Funktion eines UML-Elements über ein Dropdown-Menü zu ändern.

Evaluierung

Die meisten Tasks wurden erfolgreich abgeschlossen, Code und Unity Cleanup werden noch weiter fortgesetzt.

Sprint 12

Meetings

- Discord-Meeting: Durchführung des Sprint Reviews.
- Discord-Meeting: Besprechung, was im Video enthalten sein soll.
- Discord-Meeting: Rückblick auf das vergangene Semester und Überlegungen für das kommende Semester.

Tasks

- Code + Unity Cleanup: Florian führte eine umfassende Bereinigung von Code und Unity-Projekt durch, um die Effizienz zu steigern.
- Funktion eines UML-Elements per Dropdown ändern: Vincent implementierte die Funktion, die es ermöglicht, die Funktion eines UML-Elements über ein Dropdown-Menü zu ändern.
- Test 2 Roboters (Identifizierung möglicher Probleme für das nächste Semester): Dominik untersuchte und identifizierte mögliche Probleme bei der Implementierung von zwei Robotern im Spiel, insbesondere hinsichtlich der UI-Anpassungen für unterschiedliche Diagramme.
- Test Custom UML-Blocks (Erstellen eines neuen Blocks durch Zusammenführen anderer Blöcke, Identifizierung möglicher Probleme für das nächste Semester): Vincent führte Tests durch, um benutzerdefinierte UML-Blöcke zu erstellen und identifizierte dabei die Herausforderungen der UI-Anpassungen und des Speicherns von Diagrammen für potenzielle Neustarts.
- Test Exporting Project as exe and BrowserGame: Florian testete das Exportieren des Projekts als ausführbare Datei und Browsergame.
- Cleanup/Update Design Document: Johann und Tobias überarbeiteten das Design-Dokument und passten es an die aktuellen Anforderungen an.
- Cleanup/Update GitHub Documentation: Johann und Tobias aktualisierten die GitHub-Dokumentation.
- Erstellen von PowerPoint und Planung für die Video-Präsentation: Johann und Tobias erstellten eine PowerPoint-Präsentation und planten die Video-Präsentation, inklusive eines Meetings mit allen Teammitgliedern.

Evaluierung

Die offenen Tasks wurden erfolgreich beendet. Zudem wurde bereits ein bisschen Brainstorming betrieben, welche Hauptaufgaben wir noch für das dritte und somit letzte Semester offen haben.

Sprint 13

Mit dem 13. Sprint begann das dritte und somit finale Semester von InnoLab.

Meetings

- Discord-Meeting: Durchführung von InnoLab2 Reviews
- Discord-Meeting: Diskussion und Definition neuer Arbeitspakete, Bearbeitung des Project Diary
- Zoom-Meeting: Besprechung des Project Diary und Planung des Ablaufs für InnoLab3
- Discord-Meeting: Planung des Semesters und Sprints

Tasks

- Code + Unity Cleanup: Vincent behob zwei Bugs: den UML-Scrollbar-Bug und den Bug, dass gesperrte Level auswählbar waren.
- Speichern von UML-Bäumen: Florian implementierte die Möglichkeit, UML-Bäume zu speichern, wobei die Funktion bisher nur teilweise umgesetzt wurde.
- Fortschritt einsehen: Dominik ermöglichte Spielern, ihren Fortschritt jederzeit einzusehen.
- Titelschirm: Johann integrierte einen Titelschirm, der beim Start des Spiels angezeigt wird.
- Fragebogen für Playtester: Tobias erstellte einen Fragebogen für Playtester, wobei bisher nur Fragen zur Person umgesetzt wurden.

Evaluierung

Die Meetings waren erfolgreich und die meisten Tasks konnten abgeschlossen werden. Einige Aufgaben, wie das vollständige Speichern von UML-Bäumen und der Fragebogen für Playtester, sind noch unvollständig.

Sprint 14

Meetings

- Discord-Meeting: Sprintplanung mit Fokus auf die Änderungen für mehrere UML-Bäume sowie die erforderlichen Aufgaben vor dem Testen.
- Discord-Meeting: Besprechung der Anpassungen der UML-Pfeile, um das Laden von Bäumen zu ermöglichen.

Tasks

- Code + Unity Cleanup: Vincent änderte die Funktionalität des UML-Ablaufs, sodass dieser nicht mehr rekursiv ist, sondern direkt im Actor verwaltet wird.
- Code + Unity Cleanup: Tobias fügte die gekauften Assets ein. Aufgrund der Änderungen mussten alle Levels neu erstellt werden, da zuvor Blöcke anstelle von 2D-Sprites verwendet wurden.
- Speichern und Laden von UML-Bäumen: Florian implementierte die Ladefunktion und verschob die relevanten Funktionen in den UML-Manager.
- Fragebogen für Playtester: Tobias erstellte einen erweiterten Fragebogen mit Fokus auf die Spielerfahrung.
- Verwaltung mehrerer UML-Bäume: Dominik entwickelte den UML-Manager zur Verwaltung aller Bäume. Er ermöglichte die Erstellung und Auflistung neuer Bäume, jedoch ist das Löschen von Bäumen noch nicht implementiert.
- Aufruf eines UML-Baums aus einem anderen: Johann integrierte ein UML-Element, das einen anderen Baum aufruft, wobei er auf den UML-Manager wartete.
- Statusanzeige für Roboter und UML-Abläufe: Vincent aktualisierte die Statusanzeige für Roboter und deren UML-Abläufe, da durch die neuen Bäume Anpassungen erforderlich waren.

Evaluierung

Die Sprintplanung und Besprechungen halfen, den Fokus auf die Änderungen für mehrere UML-Bäume zu legen. Viele Tasks konnten erfolgreich abgeschlossen werden, wie die Anpassung der UML-Abläufe und die Integration der neuen Assets. Einige Funktionen, wie das Löschen von UML-Bäumen, sind jedoch noch offen.

Sprint 15

Meetings

- Discord-Meeting: Großes Team-Meeting, um das Gesamtprojekt zu reflektieren und Ziele für das Spiel zu definieren. Die gesamte Code-Basis wurde besprochen, und der Sprint wurde geplant.

Tasks

Code + Unity Cleanup:

- Vincent ersetzte alle Level-Assets durch die neuen, einschließlich Boden und anderer Objekte.
- Florian behob mehrere Fehler:
 - Korrektur fehlerhaft gezeichneter Pfeile bei For-Schleifen.
 - Anpassung der Pfeile, die manchmal falsch positioniert waren.
 - Dynamische Anpassung von Dropdown-Menüs in Blöcken basierend auf hinterlegten Enums.
 - Verhinderung des Löschens von UML-Blöcken und Pfeilen während der Diagrammausführung.
 - Anpassung der maximal verfügbaren Energie in Leveln, um geänderte Energiesparmechaniken durch Schleifen zu berücksichtigen.
- Verwaltung mehrerer UML-Bäume: Dominik arbeitete weiter am UML-Manager, um die Erstellung und Auflistung neuer Bäume zu ermöglichen. Das Löschen von Bäumen bleibt jedoch weiterhin offen.

Tutorials und Anleitungen:

- Tobias, Johann und Dominik erstellten eine PDF-Anleitung sowie Tutorial-Level:
 - Tobias: PDF-Anleitung zur Funktionsweise der UML-Blöcke, zur Benutzeroberfläche und zu den Spielmechaniken.
 - Johann: Ergänzte die Anleitungen um detaillierte Screenshots und Erklärungen.
 - Dominik: Design von Tutorial-Leveln, die grundlegende Funktionen der UML-Blöcke und Spielelemente erklären.
- Die Anleitungen und Tutorials deckten die Benutzeroberfläche, Spielmechaniken sowie die Verwendung der UML-Blöcke ab.
- Lautstärkeregelung: Johann konnte die Implementierung der Lautstärkeregelung aufgrund unerwartet aufwendiger Arbeit an den Anleitungen nicht beginnen.

Evaluierung

In diesem Sprint lag der Schwerpunkt auf der Erstellung von Tutorials und Anleitungen, um neuen Spielern den Einstieg zu erleichtern. Die meisten Tasks in diesem Bereich wurden erfolgreich abgeschlossen, auch wenn einige mehr Aufwand als erwartet erforderten. Die Arbeiten am UML-Manager gingen weiter, jedoch ohne vollständige Funktionalität. Die Lautstärkeregelung blieb unerledigt und wird in einem späteren Sprint adressiert.

Sprint 16

Meetings

- Discord-Meeting: Sprint-Planung und Durchsicht der neu eingefügten Assets.
- Discord-Meeting: Entscheidung über die Testmethodik. Da die Website im Docker nicht stabil lief, wurde entschieden, Tests mit der .exe-Version des Spiels durchzuführen.
- Discord-Meeting: Diskussion der Testergebnisse und Planung weiterer Schritte basierend auf dem erhaltenen Feedback.

Tasks

Code + Unity Cleanup:

- Vincent animierte das Wasser aus dem Asset-Pack und passte einige Assets basierend auf Feedback aus den Meetings an.
- Dominik behob einen Bug, bei dem UML-Tree-Buttons doppelt erstellt wurden.
- Lautstärkeregelung: Johann konnte diesen Task nicht abschließen, da die Tests priorisiert wurden.

Playtesting:

- Vincent: Rekrutierte 4 Tester (Freunde) und dokumentierte ihr Verhalten.
- Tobias: Test mit 6 Personen (Familie, Freunde und FH-Studenten).
- Johann: Test mit 4 Personen (Familie und Freunde).
- Dominik: Test mit 5 Personen (Familie und Mitbewohner), durchgeführt nach Abschluss von Bugfixes und Asset-Anpassungen.
- Florian: Test mit 3 Personen, die keinerlei UML-Erfahrung hatten, um die Lernkurve und Zugänglichkeit zu bewerten.

Feedback-Auswertung:

- Alle Teammitglieder werteten die von ihren Testern ausgefüllten Fragebögen aus.
- Die Ergebnisse wurden analysiert, um Schwächen im Spiel zu identifizieren und Verbesserungspotenziale zu erkennen.

Evaluierung

Dieser Sprint konzentrierte sich stark auf das Testen und Sammeln von Feedback durch Playtester, um die Benutzererfahrung zu verbessern. Die Testphase war erfolgreich, da wertvolle Daten aus einer breiten Testergruppe gesammelt wurden. Die Aufgaben zur Feedback-Auswertung und Bugfixes wurden planmäßig abgeschlossen. Aufgrund der Testpriorisierung blieb die Lautstärkeregelung erneut unerledigt, sollte aber im nächsten Sprint angegangen werden.

Sprint 17

Meetings

- Discord-Meeting: Sprint-Planung und Organisation der anstehenden Tasks.
- Discord-Meeting: Vorbereitung des Meetings mit Petz (finale Abnahme).

Tasks

Code + Unity Cleanup:

- Vincent implementierte ein Grid-System in das Spiel.
- Tobias erstellte Musterlösungen für die Tutorial-Level.
- Lautstärkeregelung: Johann implementierte die Funktion, dass die Lautstärke angepasst werden kann.
- Bedingungsblöcke verbinden: Dominik & Vincent führten farbige Markierungen für die Condition-Pfeile ein, um diese besser erkennbar zu machen. Außerdem wurde ein Offset hinzugefügt, um Überlagerungen bei Loops zu vermeiden.
- Verbinden von UML-Elementen: Dominik & Vincent ergänzten Halbkreise am Rand der UML-Elemente, um das Ziehen von Pfeilen zu erleichtern, und behoben einen Bug, bei dem Pfeile aufgrund eines Mouse-Offsets nicht in bestimmte Richtungen gezeichnet werden konnten.
- Speichern von UML-Bäumen: Florian behob zwei Bugs; einer betraf Dropdown-Werte, die nicht korrekt geladen wurden, der andere falsche Condition-Pfeile und Loops, die nicht ordnungsgemäß angezeigt wurden.
- Tutorial-Pop-ups und Erklärungen: Tobias & Johann erstellten Tutorial-Pop-ups für die UML-Blöcke, die Benutzeroberfläche und die Spielmechaniken. Die Pop-ups wurden mit Inhalten aus einer Tutorial-PDF befüllt und gegebenenfalls angepasst.

Evaluierung

Der Sprint wurde erfolgreich abgeschlossen, wobei der Fokus auf der Verbesserung der Benutzerfreundlichkeit, Fehlerbehebung und der Implementierung wichtiger Features lag. Besonders hervorzuheben sind die Anpassungen an den Tutorial- und Hilfefunktionen sowie die Optimierungen der UML-Tools.

Sprint 18

Meetings

- Discord-Meeting: Planung des Projektvideos
- Discord-Meeting: Diskussion und Finalisierung des Diarys

Tasks

- Vincent behob den Grid-Layer-Bug und passte die Assets für das dritte Level an. Zusätzlich fügte er einen Partikeleffekt für Müll hinzu, um ihn visuell zu verbessern.
- Tobias reparierte die Canvas-Skalierung und fügte ein Play-Icon hinzu.
- Dominik testete die Touch-Steuerung (erfolgreich, jedoch buggy) und benannte den Standardbaum von „New Tree“ in „Level x“ um.
- Johann ergänzte Icons für Einstellungen (Zahnrad), Umbenennen (Stift) und Löschen (Mülleimer oder „X“).
- Dominik behob einen Bug, bei dem neue Trees nicht in das Dropdown-Menü eingefügt wurden.
- Dominik implementierte erfolgreich die Funktion zum Löschen von UML-Bäumen.
- Florian fügte die Möglichkeit hinzu, Bäume umzubenennen.
- Tobias erstellte ein Tutorial mit fehlerhaften Codeblöcken, um typische Anfängerfehler zu demonstrieren.
- Johann fügte ein Hilfe-Menü im Pause-Bereich hinzu, das ein PDF zur Benutzeroberfläche anzeigt, und erstellte ein Tutorial-Icon mit Animationen für einfache Erklärungen.
- Florian fügte einen neuen Sound für den Schaden durch Gegner hinzu.
- Für die Fehleranalyse implementierte Florian Icons und Counter neben der Konsole, die Errors und Warnings darstellen.

Evaluierung

Der Sprint wurde erfolgreich abgeschlossen, mit einem starken Fokus auf die Implementierung der letzten Features und die Optimierung des Benutzererlebnisses. Alle geplanten Aufgaben wurden erfolgreich umgesetzt, und das Projekt ist nun bereit für die finale Präsentation im Abgabevideo.