

# Arbeitsjournal & Zeitplan

## INHALT

Zeitplan .....	Error! Bookmark not defined.
Grob .....	Error! Bookmark not defined.
Detailliert Sommer (insgesamt 7-8 Wochen) (geplant) .....	Error! Bookmark not defined.
Detailliert Sommer (insgesamt 7-8 Wochen) (realisiert) .....	Error! Bookmark not defined.
Arbeitsjournal.....	Error! Bookmark not defined.

## VORWORT

Diese Datei wurde mithilfe von ChatGPT auf sprachliche Fehler überprüft.

## ZEITPLAN

### Grob

Zeitpunkt	Ziel
<b>Bis Intensivwoche</b>	Game engine; Tool zum Schreiben (Overleaf oder word) und Algorithmen wählen, Hintergrund Info studieren, die Tools nutzen zu lernen.
<b>Bis Ende Sommer</b>	Haupt Arbeitsphase. Das Spiel strukturieren und dann implementieren.
<b>Bis Ende Matur Abgabe</b>	Das Spiel polieren, schriftliche Arbeit zum Spiel schreiben.
<b>Bis Mündliche Matur.</b>	Sich vorbereiten.

### Detailliert Sommer (insgesamt 7-8 Wochen) (geplant)

Zeitpunkt	Ziel
<b>1,2 Woche (23.06 – 6.07)</b>	Ein normaler Top down shooter roguelike schaffen.
<b>3,4 Woche (7.07 – 20.07)</b>	Die KI-Mechaniken mehr studieren und in dem Spiel implementieren und mit denen experimentieren.
<b>5,6 Woche (21.07 – 3.08)</b>	Assets finden (eventuell kaufen), sie im Spiel gut nutzen, möglicherweise die Geschichte zum Spiel addieren. Allgemein das Spiel polieren und beenden.
<b>7,8 Woche (4.08 – 17.08)</b>	Die Schriftliche Arbeit schreiben.
<b>Bis Abgabe</b>	Alles Mögliche verbessern.

## Detailliert Sommer (insgesamt 7-8 Wochen) (realisiert)

Zeitpunkt	Realisation des Ziels
<b>1,2 Wochen</b>	Ein normales Top down shooter roguelike fast geschafft.
<b>3 Woche</b>	Fast nichts gemacht, war busy.
<b>4 Woche</b>	Weiter basis-Spiel Arbeit, Sockets-Geschichte. KI-Design. Teil Assets gefunden.
<b>5, 6 Wochen</b>	Mehr Arbeit an Spiel, Weitere Graphische und akustische Assets gefunden.
<b>7,8 Woche (4.08 – 17.08)</b>	Die Schriftliche Arbeit geschrieben.
<b>Bis Abgabe</b>	Alles Mögliche verbessert.

Also: weniger Arbeit mit KI und Assets, mehr mit Spiel und Sockets.

## ARBEITSJOURNAL

Unter Kategorie ist vermerkt, um welche Art von Arbeit es sich handelt:

Organisation (Planung, Besprechung o.ä.), Informationsbeschaffung (Material, Lektüre, Experiment o.ä.), Auswertung (Schreiben, Korrigieren Herstellen o.ä.). Die Kategorisierung hilft, den Zeitaufwand einzelner Arbeitselemente insgesamt zu überblicken.

Den Zeitaufwand ist in Stunden, abgekürzt Std. notiert,

Um eine detailliertere Änderungshistorie des Projekts zu sehen, können Sie die GitHub Commits des Projekts besuchen. (Wenn Sie Zugang haben).

Datum (2025)	Arbeit	Kategorie	Zeit
<b>5 - 6.03</b>	Ideen gesammelt; das Projekt konkretisiert; Godot ein wenig ausprobiert.	O	3 Std.
<b>15.03</b>	Repository auf GitHub erstellt; dieses Dokument erstellt; auf Overleaf nach Vorlagen für schriftliches Teil der Arbeit gesucht; Projektplanung im Allgemeinen.	O	2 Std.
<b>19.03</b>	Weitere Planung, Struktur des Berichts in Overleaf gemacht. Recherche über die Q-Learning und Genetische Algorithmen.	O, I	2.5 Std.
<b>22.03</b>	Godot Engine als Projekt Game-Engine gewählt. Es mit einem Guide ausprobiert.	I	3 Std.
<b>2.04</b>	Installationsanleitungen und Ziele der Arbeit zu Repo addiert.	O	0.5 Std.
<b>Intensivwoche (1. Woche Sommer)</b>			
<b>23.06</b>	Die Sommer Arbeit ausgeplant. Mithilfe an Produkt gearbeitet: Player, Schießen- und dashing- Mechaniken gemacht. Die Ordnerstruktur weiterentwickelt.	I, A, O	6 Std.

<b>24.06</b>	Weiter an dem Spiel gearbeitet: Gegner, Arena, Managers zu der besseren Kontrolle (Game-Manager, Projektil-Manager, Entities-Manager ...). Spawning Mechanik der Gegner – zur Zeit rechteckige Spawners.	A, I	6 Std.
<b>25.06</b>	Das Leben Mechanik des Spielers und HUD. Beginn an GUI arbeiten. Das Code verbessert, um es mehr nachhaltig zu machen.	A, I	5,5 Std.
<b>26.06</b>	GUI gemacht: Haupt Menu, Todesmenü, Pausenmenü. Die Logik dahinter hinzugefügt. 1 Stunde an einem dummen Fehler verschwendet.	A, I	5 Std.
<b>27.06</b>	Das Welle Mechanik hinzugefügt: Nun spawnen die Gegner in bestimmten Mengen bei Wellen. Die Gegner Menge ist variiert je nach Welle, und steigt jede Welle. (Nun ist es +5 Gegner pro Welle mit Anfangswert von 10) HUD angepasst, sodass es zeigt, wie viel Gegner bleiben und welche Welle ist es	A, I	5 Std.
<b>28.06</b>	Spieler Progression durch Boni/Items addiert: Nach jeder Welle wird drei Optionen gegeben, die aus anderen Pool zufällig genommen werden und den Spieler stärker machen (z. B. Steigerung der Schaden von Kugeln oder der Bewegungsgeschwindigkeit des Spieler)	A, I	4 Std.
<b>2. Woche</b>			
<b>30.06</b>	Gegner, die schießen können gemacht. Dafür Code mehr modular gemacht: das Spieler und dieses Gegner Typ können dasselbe Waffenklassen tragen	A, I	3.5 Std.
<b>1.07</b>	Ressourcen gelernt und im Spiel implementiert. Das hilft um mehr Gegenstände (Upgrades), Waffen, Gegener zu machen.	A, I	4.5 Std.
<b>3.07</b>	Code refactoring: 3 früher-gemachten Managers (globalen singleton autoloads) abgeschafft und stattdessen Referenzen zu wichtigen Nodes (z. B. Spielerinstanz oder HUD) mehr zentriert in Level Node weitergegeben (was natürlich es schwieriger, aber sauberer macht, da das Prinzip von «Encapsulation» besser gehalten wird). Als Resultat ist nur 2 Autoloads geblieben: Game-Manager – notwendig für Szenen Wechsel (z. B. Main menu -> Spiel) und EntitiesManager (eher Wave manager eigentlich), den ich vielleicht später abschaffen werde. Der Grund dafür war die Realisation, dass die Managers auch während des Main menu geladen sind und Gedanke es sauberer versuchen zu machen. Später wurde es aber klar, dass es nicht zu nötig war: mein Spiel wird nicht so gross, dass den Prinzipien des OOPs insbesondere Encapsulation wirklich wichtig sind. Ich arbeite zudem in diesem Projekt allein.  Waffen gemacht: Handgun, Shotgun, Machine gun, Sniper rifle – alle mit eigenen Stats.	A, I, O	7 Std.
<b>5.07</b>	Hud verbessert: nun zeigt es deine Gegenstände und Waffe. Bugs gefixt	A, I	3 Std.

6.07	Ton und Musik addiert.	A, I	4 Std.
<b>3 Woche:</b>			
<b>Nicht viel Arbeit gab es an der Matura, da meine Mutter hatte viele Arzttermine zur Vorbereitung zur Operation und ich musste übersetzen</b>			
9.07	Begann an der Q Tabellen, um KI der Gegner zu verbessern.	I,A,O	5 Std.
<b>4 Woche:</b>			
14.07	<p>Planung wegen verlorener Woche und Reflexion geändert: Noch ein bisschen Arbeit an normales Gameplay schien nötig zu sein, also noch in dem Gebiet Arbeit geplant, bis es gut genug fühlt (aber Frist bis Ende Woche).</p> <p>Entschieden, dass Assets nicht gekauft werden werden (kein Budget, und professionelle kosten zu viel), sondern freie aus Internet herunterladen oder wenn es Zeit gibt selbst machen (unwahrscheinlich).</p> <p>Waffen zur Gameplay addiert, wobei sie zur Wahl des Spielers jede 5 Welle gegeben werden. Wahl der Items verbessert: jedes Item hat «rarity», die bestimmt wie häufig es nach der Welle in dem Upgrade Menu erscheinen kann. Die Rarity bestimmt zudem die Farbe von Border der Upgrade Knopf</p>	I, A, O	6 Std.
15.07	<p>KI-Logik zu Python geschoben. Dafür die Python Scripts für Server, Q_Learner class, und Config gemacht. Die Verbindung zwischen Server (Python Scripts) und Client (Godot Gegnern) durch TPC Sockets gemacht, die Kommunikation ist durch Austausch von JSON-Dateien passiert.</p> <p>Durch Delegation der Code zum Python, ist viel Flexibilität gewonnen: Das ermöglicht viel Libraries zum KI, wie z, B, Pytorch und zur Schaffung der Grafiken für schriftliche Arbeit einfach mit matplotlib oder seaborn. Ich kenne zudem Python viel besser als GdScript oder Godot.</p> <p>Begann an Logging: server Seite sagt nun alle nötige Info, um effektiv debuggen zu können.</p>	I, A, O	7.5 Std.
16.07	Sockets, mit Server und Client hat es ziemlich mehr kompliziert gemacht. Die Kommunikation zwischen Server und Klient funktionierte nicht wirklich. An diesen Tag habe ich die Kommunikation ermöglicht, jedoch die Sachen funktionierten nicht ganz, wenn es mehr als ein Gegner gibt. An der Logging von Client (Godot, also Hauptspiel), gearbeitet.	I, A	5 Std.
17.07	<p>Mehr Socket Arbeit, ein bisschen KI Rewards und Aktionen verbessert, aber es gibt immer noch Problem mit Godot – Python – Godot. Es soll folgende Reihe der Sachen passieren (auf Englisch, da ich auch ein prompt zur ChatGPT gemacht habe, um Hilfe zu erhalten):</p> <p>server receives new state --&gt; server chooses some action for that state for that enemy --&gt; client gets the action, does it,</p>	I, A, O	6 Std.

	<p>and sends reward for that action --&gt; server gets it and updates the reward for that state.</p> <p>Passierte aber eher zufällig. Ich habe dann es mit dem Buffer teilweise gelöst, aber es gibt immer noch Problem, wenn es 2+ Gegnern gibt.</p> <p>Viel Planung und Design-Gedanken:</p> <ol style="list-style-type: none"> <li>1. Ob Socket Kommunikation durch mehrere Threads und Sockets gemacht werden soll. – Für jetzt entschieden, ein Socket zu haben, wobei verschiedenen Gegner durch ihre ID ihre Daten bekommen.</li> <li>2. Wie sollen Reward und Lern Parametern nicht arbiträr sein. Entschieden später ein Algorithmus wählen und Spiel mit vielen Configs mit zufälligen (oder grid-basierten) Parametern zu simulieren, wobei der Spieler auch simuliert wird.</li> <li>3. Setting, mit dummer Geschichte und Ziel des Spiels, und zwar sehr einfache, die durch zufällige Assets, die ich schon von Internet habe, in meinen Kopf gefallen ist: Spieler ist ein Ritter, der einfach mit seiner Familie chillt, aber dann wird teleportiert zu Goblin Camp. Das Ziel wird die Toren zu brechen und einfach fliehen. Um es aber zu machen, wird ein DPS check gemacht. Also es wird nötig Boni zu Schaden nehmen, bis es gelangt es in einer bestimmten Zeit zu brechen.</li> <li>4. Um 2. Effektiv und nie mehr später machen, braucht es schon, dass das Spiel sich nicht viel ändert. Also möchte ich 3. zuerst realisieren.</li> </ol> <p>Auch gab es Realisation, dass es nicht viel Zeit gibt, also soll ich mehr Arbeiten.</p>		
<b>18.07</b>	Mehr Socket Arbeit: alle Gegner q tables und Rewards requests werden zentral verarbeitet und dann gleichzeitig durch ein JSON zu Server geschickt. So wird es viel einfacher das Prozess zu kontrollieren (früher hatten die Gegner selbst immer eigene JSON, nur mit eigenen q table geschickt). Funktionierte aber immer nicht ganz.	I, A	5 Std.
<b>21.07</b>	Endlich Sockets gefixt. Options und Credits Menus hinzugefügt. Neue Assets aus dem itch.io heruntergeladen, sie werde ich nun zum fertigen Produkt nutzen. Mit denen die Welt neu gemacht.	I, A	6 Std.
<b>5 Woche</b>			
<b>22.07</b>	Schatten, Gewinn Bedingung, Toren, Gewinn Menu.	I, A	5 Std.
<b>23.07</b>	Nahkampf Waffen Klasse, einige Bugs gefixt	I, A	5 Std.
<b>24.07</b>	GUI schöner gemacht, eine Cutscene vor dem Spiel addiert.	I, A	5 Std.

<b>25.07 – 30.07</b>	Vergessen hier report zu schreiben, jedenfalls so etwas gemacht: Tutorial, Audiooptions, neue Nahkampf Waffen, die neue Logik für Waffen Selektion der Gegnern, bessere Reward Logik.	I, A	18 Std.
<b>6 Woche</b>			
<b>31.07 – 3.08</b>	Angefangen mit der Optimierung der Rewards, mehr Items gemacht, bugs gefixt. Spiel kompiliert und launcher gemacht (der aktiviert den Server und das Spiel), sodass man es einfach an itch.io herunterladen kann.	I, A	13 Std.
<b>4.08 – 10.08</b>	7 Woche Schriftliches Teil der Arbeit: Erste Draft an Einleitung und Theoretischer Hintergrund	A	18 Std.
<b>11.08 - 17.08</b>	8 Woche Methodik und Implementierung Sektion	A	18 Std.
<b>18.08 – 24.08</b>	9 Woche Einige Bug-Fixes / Zielsetzung	A, I, O	6 Std.
<b>25.08-7.09</b>	10-11 Woche Spiel auf Itch.io veröffentlicht, dafür nötige Dateien gemacht und gefunden, Kleine Spiel-Verbesserungen (Schönere Hintergründe an GUI, Dekorative Objekten in der Arena, Sterbeanimation der Gegner, Slash Effekt an Nahkampf Waffen), Ton während dieses Effektes. Balancing (Weniger Gegner, Gegner werden stärker jede 4 Welle, Ammo für Fernkampf Waffen) + Leaderboard das mit Firebase funktioniert.  Ich habe vergessen, hier Report zu schreiben daher 10-11 Wochen, falls das genaue Date jeder diesem Betrage wichtig ist, könnte man an GitHub Commit Geschichte es detaillierter anschauen.	A, I, O	20 Std.
<b>8.09 – 14.09</b>	12 Woche Methodik und Implementierung Sektion verbessert: Addition der Bewertung, einige Bilder und andere.	A, I	4 Std.
<b>Totale Arbeitszeit in Stunden:</b>			

