



FRONTIER

Projektskizze



8. MÄRZ 2025

Kevin Sonnek, Thomas Hilberink, Sercan Ay, Rafael Gonçalves, Oliver Hearn, Lars Volkheimer,
Manuel Streng

Inhaltsverzeichnis

1 Ausgangslage & Idee	3
Definition Tower Defense.....	3
Inspiration	3
Frontier	4
Zielgruppe.....	5
Verkaufsargument / Kicker.....	5
2 Stand der Technik / Konkurrenzanalyse.....	6
3 Kontextszenario (Hauptablauf)	7
Einleitung.....	7
Spielstart & Vorbereitung.....	7
Hauptspielablauf	7
Tageszyklus & Bauphase.....	7
Kampfphase (Gegnerwelle)	8
Zeitreise-Mechanik.....	8
Tagesabschluss & Progression.....	8
Sieg- und Verlustbedingungen	9
Weitere Überlegungen	9
4 Weitere Anforderungen & Weiterentwicklung.....	9
Erweiterung der Gameplay-Mechaniken	9
Technische Weiterentwicklungen	10
Balance & Schwierigkeitsanpassung	10
Langfristige Perspektiven	10
5 Ressourcen und Zeitplan	10
Projektteam.....	10
Erfahrungen & Teamkompetenzen	11
Aktuelle und zukünftige Voraussetzungen Projekt	11
Technische Einarbeitung	11
Arbeitsweise & Zusammenarbeit	11
Zukünftige Anforderungen	11
Zeitplan / Roadmap.....	12
6 Risiken.....	13
7 Ausblick.....	16
Weiterentwicklungsmöglichkeiten.....	16
Spielbare, aber nicht vollendete Version	16
NoTech-Produkt & Dokumentation	16

Fazit	16
8 Anhang.....	17
Game-Concept.....	17
Audio	17
Artkonzepte	17
Lore.....	17
Grob-Ablauf	18
Spielablauf	18
UI-Overview	20
Allgemeine Spielmechaniken	23
Vorhandene Ressourcen	23
Gebäude	23
Zeit/Gegnerwelle.....	24
Gegner	24
Gameplayloop	26
Gewinnbedingung	26

1 Ausgangslage & Idee

Das Projekt zielt darauf ab, ein Tower-Defense-Spiel im Retrostil zu entwickeln, das klassische Elemente des Genres mit modernen Management- und Strategiaspekten verbindet. Inspiriert durch erfolgreiche Titel wie *They Are Billions*, soll das Spiel nicht nur herausfordernde Verteidigungsmechaniken bieten, sondern auch dem Spieler ermöglichen Ressourcen sowie Gebäude zu managen, während es gilt sein Hauptquartier von Wellen von Gegnern zu verteidigen. Das zentrale Konzept greift die Idee einer Zeitreise auf – das Hauptgebäude (HQ) besitzt eine einzigartige Fähigkeit, den Spieltag zu wiederholen, was neue strategische Tiefen eröffnet.

Definition Tower Defense

Tower Defense (TD) bezeichnet ein Genre von Strategiespielen, bei dem der Spieler Verteidigungsanlagen – sogenannte Türme – entlang von Pfaden oder in ausgewiesenen Bereichen platziert, um anrückende Gegnerwellen abzuwehren. Im klassischen Ansatz geht es darum, mit begrenzten Ressourcen optimale Verteidigungsstrategien zu entwickeln, wobei Türme oft auf bestimmte Gegnerarten spezialisiert sind oder mit Upgrades verbessert werden können. In unserem Konzept wird diese Grundidee erweitert, indem zusätzlich Elemente des Ressourcenmanagements und die Möglichkeiten Gebäudefähigkeiten zu Benutzen.

Inspiration

Das Spielkonzept wird massgeblich inspiriert von *They Are Billions*, einem Spiel, das Tower Defense mit strategischem Management verknüpft. Neben dem Hauptfeature, Verteidigungsanlagen zu bauen, steht insbesondere die Möglichkeit im Fokus, Einheiten dynamisch zu bewegen und gezielt einzusetzen – ein Feature, das sowohl taktische Flexibilität als auch zusätzliche Herausforderungen in den Spielablauf bringt. Im Rahmen des Ressourcenmanagements muss der Spieler den stetigen Fluss an Rohstoffen wie Gold, Nahrung, Öl und weiteren Ressourcen effizient verwalten. Dies bedeutet, dass der gezielte Abbau, die strategische Verteilung und der effiziente Einsatz der Ressourcen entscheidend sind, um Verteidigungsanlagen zu errichten, Upgrades durchzuführen und Reparaturen vorzunehmen. Diese Mischung aus klassischen TD-Mechaniken, innovativen Steuerungselementen und anspruchsvollem Ressourcenmanagement bildet als Hauptinspiration für unser Spiel.



Abbildung 1 «They are Billions» – rot = Gegnerwelle, blau = Türme und Mauern, grün = Einheiten

Frontier

Das Spiel wird als 2D-Titel im Pixelart-Stil umgesetzt, um den nostalgischen Retro-Charme zu bewahren und gleichzeitig die Entwicklung effizient zu halten. Diese visuelle Entscheidung ermöglicht es dem Team, sich auf das Gameplay und die Mechaniken zu konzentrieren, ohne durch aufwendige Grafikerstellung ausgebremst zu werden.

Das Spiel bietet neben dem charmanten 2D-Pixelart-Look auch eine Reihe effektiver Features, die den Spielspass und die strategische Tiefe erhöhen:

- **Zeitreise-HQ-Fähigkeit:** Das Hauptquartier kann nach einer bestimmten Anzahl von Tagen «aufgeladen» werden, so dass der Spieler einen erfolglosen Tag einmalig wiederholen kann. So können Fehler korrigiert und neue Strategien ausprobiert werden, um den Spielverlauf zu optimieren.
- **Vielschichtiges Ressourcenmanagement:** Der Spieler muss unterschiedliche Ressourcen wie Gold, Holz, Stein und Eisen effizient abbauen, verwalten und einsetzen, um Gebäude zu errichten, Upgrades durchzuführen und Reparaturen vorzunehmen.
- **Modulare Gebäudestrukturen:** Verschiedene Gebäudetypen, etwa für den Ressourcenabbau, verfügen über individuelle Eigenschaften und Lebenspunkte, die strategisch verbessert werden können. In der ersten Demo konzentriert sich das Spiel auf die Implementierung der HQ-Fähigkeit, während weitere Gebäude-Features in späteren Versionen folgen.
- **Dynamische Gegnerwellen und Balancing:** Das Spiel setzt auf einen gestaffelten Schwierigkeitsgrad, bei dem leichtere Gegnerwellen von anspruchsvolleren abgelöst werden. Mathematisch abgestimmte Funktionen sorgen dabei für einen stetigen, aber fairen Anstieg der Herausforderung.
- **Flexible Karten- und Spielmodi:** Zum Spielstart wird mindestens eine vorgefertigte Karte bereitgestellt. Langfristig ist als optionales Ziel die Implementierung zufällig generierter Karten geplant, um jede Runde einzigartig zu gestalten. Zudem könnte ein optionaler Map-Builder integriert werden, der es Spielern ermöglicht, eigene Karten individuell anzupassen.
- **Benutzerfreundliche UI und Audio-Atmosphäre:** Ein übersichtlicher Lade- und Titelschirm sowie eine intuitive Benutzerführung sorgen für einen reibungslosen Einstieg. Hintergrundmusik, Soundeffekte und Ambient-Music tragen zur stimmungsvollen Atmosphäre bei.

Diese Features schaffen eine abwechslungsreiche und tiefgründige Spielerfahrung, die sowohl Liebhaber klassischer Tower-Defense-Spiele als auch Fans moderner Strategiespiele anspricht – und das alles in einem ressourcenschonenden, retro-inspirierten Paket.

Für weiteren Einsicht in das Spielkonzept sind im Anhang weitere erste Spielkonzepte zusammengetragen.

Zielgruppe

Die Kernzielgruppe, sind Fans des Tower-Defense-Genres sowie Liebhaber von Strategie- und 2D-Spielen. Besonders angesprochen werden Spieler, die den Retro-Stil schätzen und eine Vorliebe für Pixelart haben. Gleichzeitig richtet sich das Spiel an Nutzer, die Spass an taktischen Herausforderungen und dem Management von Ressourcen haben.

Verkaufsargument / Kicker

Das zentrale Verkaufsargument des Spiels liegt in seinem Retrostil: Der 2D-Look sorgt für einen geringen Arbeitsspeicherverbrauch und hohe Portabilität über verschiedene Systeme hinweg. Als besonderes Feature hebt sich das Spiel durch das «Tag wiederholen» – bzw. das Zeitreise-Feature des Hauptgebäudes (HQ) – ab. Diese Mechanik ermöglicht es, vergangene Spielphasen zu wiederholen und damit strategisch von Fehlern zu lernen oder alternative Ansätze auszuprobieren. Zudem soll das Ressourcenmanagement eine weitere strategische Tiefe bieten.

2 Stand der Technik / Konkurrenzanalyse

In dieser Übersichtstabelle werden die wesentlichen Merkmale mehrerer Tower-Defense- und Strategiespiele miteinander verglichen. Sie zeigt auf, wie die einzelnen Titel unter anderem Verteidigung, Ressourcenmanagement, Gebäude-Upgrades und Grafikstil umsetzen. Die Gegenüberstellung hilft dabei, Gemeinsamkeiten und Unterschiede zu erkennen und liefert wertvolle Inspiration für die Entwicklung unseres eigenen Spiels.

Tabelle 1 Konkurrenzanalyse

Feature/Funktionalität	Unser Game	They are Billions	The Last Spell	Bloons TD 6
Tower als Hauptverteidigung	Ja	Ja aber teilt sich gleiche Wichtigkeiten mit Einheiten, die helfen zu verteidigen	Ja aber hauptsächlich über Einheiten	Nein, Weg muss verteidigt werden
Konzept «Verteidigen gegen Wellen von Gegnern»	Ja	Ja	Ja	Ja
Ressourcen sammeln	Ja – erhältlich durch Felder auf einer Karte und sammeln durch Gebäude	Ja – erhältlich durch Felder auf einer Karte und sammeln durch Gebäude	Ja – erhältlich durch Generieren von Gebäuden und bekämpfen von Gegnern	Ja – erhältlich durch bekämpfen von Gegner und überstehen von Runden
Ressourcen managen	Ja – benutzbar um Gebäude zu bauen und upzugraden	Ja – benutzbar um Gebäude zu bauen, upzugraden und Einheiten zu kaufen / zu verbessern	Ja – benutzbar um Gebäude zu bauen, upzugraden und Einheiten zu verbessern / zu kaufen	Ja – benutzbar um Gebäude zu bauen, upzugraden
Gebäude upgraden	Ja	Ja	Ja	Ja
Gebäude haben Fähigkeiten, die helfen zu verteidigen	Ja – aktive und passive Fähigkeiten	Passive Fähigkeiten	Passive Fähigkeiten	Ja, aber nur wenige Gebäude
Einheiten als zusätzliche Verteidigungsmechanik	Nein	Ja	Ja - Hauptverteidigung	Nein
Einheiten upgradebar	Nein	Ja - indirekt	Ja - direkt	Nein
Grafik	2D – von oben	2D - isometrisch	2D - isometrisch	2D – von oben
Art-Style / Thematik	Pixelart / SCFI, Mittelalter oder Zombies (noch nicht fix)	Comic-Stil mit Cellshading-Look / Post-Apocalyptic mit Zombies	Pixelart / Dark Fantasy	Einfache Comic-Grafik / Cartoon-Krieg
Karte	Raster mit «Tiles»	Raster mit „Tiles“ nur sichtbar bei bauen von Gebäuden	Raster mit «Tiles» immer sichtbar	Freies Feld

3 Kontextszenario (Hauptablauf)

Einleitung

Das Ziel unseres Projektes ist es, eine spielbare erste Version zu implementieren. Es soll ein spielbarer Prototyp sein. Spielbarer Prototyp bedeutet, dass unser Spiel über die Mindestanforderungen verfügt, um als Tower-Defense-Spiel eine funktionale Gameplay-Loop zu bieten, bei der der Spieler Gebäude errichten, Ressourcen verwalten und Gegnerwellen abwehren kann.

Spielstart & Vorbereitung

Das Spiel startet mit einem Ladebildschirm, gefolgt von einem Titelschirm mit grundlegenden Menüoptionen. Hier kann der Spieler ein neues Spiel beginnen oder einen gespeicherten Spielstand laden.

Ein Einstellungsmenü wird bereitgestellt, jedoch nur mit den notwendigsten Optionen. Das Spiel wird eine automatische Schwierigkeitsskala haben, bei der die Gegnerwellen mit jeder Runde anspruchsvoller werden. Eine explizite Schwierigkeitsstufenauswahl ist als Stretch Goal vorgesehen.

Hauptspielablauf

Das Spiel wird mit einer von den Entwicklern erstellten Karte gestartet. Ein Map-Builder ist vorgesehen, der später ins Spiel integriert werden kann. Als Stretch Goal ist zudem eine zufällige Kartengenerierung geplant.

Der Spieler beginnt mit einem Hauptquartier (HQ) auf der Karte und erhält Startressourcen. Anschliessend kann er sofort beginnen, Gebäude zu errichten und sich auf die erste Gegnerwelle vorzubereiten.

Zu den Hauptmechaniken gehören das Platzieren von Verteidigungsgebäuden, der Bau von Ressourcensammelgebäuden und die kontinuierliche Optimierung der Basis. Nach jeder überlebten Runde erhält der Spieler zusätzliche Ressourcen, um seine Verteidigung weiter auszubauen.

Tageszyklus & Bauphase

Jeder Tag im Spiel beginnt mit einem festgelegten Ressourcenbestand. Der Spieler kann verschiedene Gebäude errichten, um Ressourcen zu sammeln oder Verteidigungsanlagen zu platzieren. Dabei gibt es unterschiedliche Gebäudearten:

- Ressourcengebäude: Forstwirtschaft, Steinbruch, Eisenmine zur langfristigen Versorgung mit Baumaterialien.
- Verteidigungsanlagen: Türme und Mauern zum Schutz vor Gegnerwellen.
- Upgrades: Strukturen, die mit Ressourcen verbessert werden können, um effizienter zu arbeiten oder zusätzliche Funktionen freizuschalten.

Sobald der Spieler bereit ist, startet er die nächste Gegnerwelle.

Kampfphase (Gegnerwelle)

Jede Gegnerwelle startet entweder automatisch nach einer bestimmten Zeit oder sobald der Spieler den «Ready»-Knopf drückt. Die Gegner erscheinen in einem bestimmten Bereich der Karte und bewegen sich in Richtung des HQs.

- Verhalten der Gegner: Sie können entweder direkt auf das HQ zulaufen oder versuchen, Lücken in der Verteidigung auszunutzen.
- Kampfmechanik: Türme greifen automatisch Gegner an.
- Folgen eines Angriffs: Falls das HQ oder wichtige Gebäude zerstört werden, kann dies entweder zu einem Game Over führen oder erfordert einen strategischen Neuaufbau mittels Zeitreise-Mechanik.

Zeitreise-Mechanik

Das HQ besitzt eine besondere Fähigkeit: Nach einer bestimmten Anzahl von Tagen kann es aufgeladen werden, um einen vergangenen Tag zu wiederholen. Dies ermöglicht es dem Spieler, aus Fehlern zu lernen und alternative Verteidigungsstrategien auszuprobieren.

- Bedingungen für die Zeitreise: Der Spieler kann nur die letzte Phase wiederholen, aber nicht weiter zurückgehen.
- Strategische Vorteile: Fehlentscheidungen können korrigiert werden, ohne das gesamte Spiel von vorne beginnen zu müssen.
- Integration in die Spielwelt: Die Mechanik ist in die Spiel-Lore eingebunden, indem das HQ als «Zeitreisestein» fungiert. Die allgemeine Spiel-Lore wird im Game-Booklet noch weiter und tiefer beschrieben.

Tagesabschluss & Progression

Nach jeder überstandenen Gegnerwelle beginnt die Vorbereitungs-Phase erneut. Der Spieler erhält:

- Ressourcen aus bestehenden Gebäuden (Holz, Stein, Eisen, Währung vom HQ).
- Möglichkeiten zur Expansion: Neue Gebäude können errichtet, Verteidigungsanlagen verbessert oder beschädigte Strukturen repariert werden.
- Gegnerprogression: Gegnerwellen werden mit jeder Runde stärker, entweder durch grössere Gruppen oder neue Gegnertypen.

Falls Gebäude im vorherigen Kampf zerstört wurden, bleiben sie zerstört, bis der Spieler sie erneut errichtet.

Sieg- und Verlustbedingungen

Das Spiel kann auf verschiedene Weise enden:

Mögliche Siegbedingungen:

- Überleben einer festen Anzahl von Tagen.
- Freischalten eines Endlosmodus nach einer bestimmten Zeit.
- Erfolgreiche Verteidigung eines festgelegten strategischen Punktes über eine definierte Dauer.

Mögliche Niederlagebedingungen:

- Das HQ wird vollständig zerstört.
- Alle Ressourcen sind aufgebraucht, sodass keine weiteren Gebäude oder Verteidigungen errichtet werden können.
- Die Gegner dominieren das Spielfeld, sodass keine sinnvolle Gegenwehr mehr möglich ist.

Weitere Überlegungen

Zusätzlich zu den Kernmechaniken gibt es weitere Aspekte, die das Spielerlebnis verbessern und das Gameplay abrunden können.

- Benutzerfreundliche UI & Steuerung: Eine intuitive Benutzeroberfläche soll sicherstellen, dass alle wichtigen Informationen leicht zugänglich sind. Dazu gehören Hinweise zu Ressourcen, Gebäudeplatzierung und Verteidigungsstatus.
- Visuelles und akustisches Feedback: Warnungen bei Angriffen auf das HQ oder bei vollen Ressourcenbeständen können helfen, das Spielgeschehen transparenter zu machen.
- Erweiterbare Mechaniken: Stretch Goals wie zufällig generierte Karten könnten langfristig das Gameplay erweitern.
- Balancing und Schwierigkeitsgrad: Feintuning der Gegnerwellen und Ressourcenverteilung soll sicherstellen, dass das Spiel herausfordernd, aber fair bleibt.

4 Weitere Anforderungen & Weiterentwicklung

Die aktuelle Version des Spiels legt den Grundstein für ein Tower-Defense-Spiel mit strategischem Ressourcenmanagement und der Zeitreise-Mechanik. Nach der initialen Umsetzung in PM4 gibt es mehrere mögliche Entwicklungsrichtungen, um das Spiel weiter zu verbessern und den Spielumfang zu erweitern.

Erweiterung der Gameplay-Mechaniken

- Mehr Gegnertypen & KI-Verbesserungen: Unterschiedliche Gegnerarten mit individuellen Verhaltensmustern könnten für mehr Herausforderung sorgen.
- Zusätzliche Verteidigungsanlagen: Neue Turmartentypen und erweiterbare Verteidigungsstrukturen könnten strategische Vielfalt bieten.
- Mehr Ressourcentypen & Wirtschaftssystem: Zusätzliche Rohstoffe könnten für ein komplexeres Ressourcenmanagement genutzt werden.

Technische Weiterentwicklungen

- Map-Builder: Bereits vorgesehen als Feature, um eigene Karten zu erstellen. Langfristig könnte dieser auch ins Hauptspiel integriert werden.
- Zufällige Kartengenerierung (Stretch Goal): Erhöht den Wiederspielwert und sorgt für mehr Abwechslung.
- Optimierung der Benutzeroberfläche & Bedienbarkeit: Verbesserung der Steuerung und Interface-Elemente für eine intuitivere Nutzererfahrung.

Balance & Schwierigkeitsanpassung

- Dynamische Schwierigkeitsprogression: Aktuell steigen Gegnerwellen rundenweise an. Eine smartere KI oder adaptive Anpassungen an den Spielfortschritt könnten für mehr Herausforderung sorgen.
- Early-Access oder Beta-Testphase: Spielerfeedback könnte helfen, das Balancing zu optimieren und Bugs zu beheben.

Langfristige Perspektiven

- Modding-Support: Falls realisierbar, könnte eine Schnittstelle zur Erstellung eigener Inhalte bereitgestellt werden.
- Veröffentlichung auf Plattformen wie Steam (Theorie): Sollte das Spiel vollumfänglich entwickelt werden, wäre eine Veröffentlichung auf Steam oder ähnlichen Plattformen eine Möglichkeit. Dies würde jedoch erhebliche zusätzliche Entwicklungsarbeit, Marketing und Support erfordern und ist derzeit nicht vorgesehen, da es den Rahmen des Projekts bei weitem übersteigen würde.

Diese Entwicklungsrichtungen würden das Projekt nach PM4 auf eine neue Stufe heben und das Spielerlebnis weiter vertiefen.

5 Ressourcen und Zeitplan

Projektteam

Projektleiter, Kevin Sonnek: Verantwortlich für Planung und Projektmanagement

Stellvertretender Projektleiter, Sercan Ay: Projektmanagement

Technischer Leiter, Thomas Hilberink: CTO

Stellvertretender technischer Leiter: Manuel Strenge: Deputy CTO

Stellvertretender technischer Leiter: Lars Volkheimer: Deputy CTO

Development-Team: Oliver Hearn, Rafael Goncalvez: Projektteam

Entwicklung, ganzes Team: Verantwortlich für die Umsetzung der Applikation des Frontend bis zum Backend

Erfahrungen & Teamkompetenzen

Das Projekt wird mit Java und der libGDX-Engine entwickelt, weshalb unser Team gut vorbereitet ist, um die Umsetzung effizient anzugehen. Das Team besteht aus sieben Personen, die über Programmiererfahrung verfügen. Vier Teammitglieder sind hauptberufliche Entwickler, während die restlichen drei auf das Wissen aus ihrem Studium zurückgreifen, insbesondere in Java.

- Zwei der Programmierer haben bereits Erfahrung mit der libGDX-Engine und konnten ein eigenes Spiel umsetzen.
- Ein Teammitglied hat bereits ein rudimentäres Tower-Defense-Spiel entwickelt, wenn auch mit einem anderen Framework.

Durch dieses Wissen im Bereich Spieleentwicklung und Framework-Nutzung ist das Team gut aufgestellt, um innerhalb der kommenden Wochen eine spielbare Version des Projekts zu realisieren.

Aktuelle und zukünftige Voraussetzungen Projekt

Um eine effiziente Zusammenarbeit im Team zu gewährleisten und eine erfolgreiche Umsetzung des Projekts sicherzustellen, gibt es einige grundlegende technische und organisatorische Voraussetzungen, die alle Teammitglieder erfüllen sollten.

Technische Einarbeitung

- Alle Teammitglieder setzen sich mit libGDX auseinander, um eine einheitliche Entwicklungsbasis zu schaffen.
- Vertrautheit mit den vorhandenen GitHub-Tools und Repository-Strukturen ist essenziell, um Code-Kollaboration, Versionierung und Fehlerbehebung effizient zu gestalten.

Arbeitsweise & Zusammenarbeit

- Regelmässige Code-Reviews und wöchentliche Meetings zur Abstimmung der aktuellen Entwicklungen.
- Klare Aufgabenverteilung und Verantwortlichkeiten innerhalb des Teams zur Vermeidung von Redundanzen und Missverständnissen.

Zukünftige Anforderungen

- Falls erforderlich, könnten weitere Tools oder Frameworks in Betracht gezogen werden, um spezifische Herausforderungen besser zu bewältigen.
- Eine kontinuierliche Verbesserung der internen Entwicklungsprozesse anhand von Lessons Learned während des Projekts.

Diese Massnahmen stellen sicher, dass das gesamte Team effizient zusammenarbeitet und das Projekt zielgerichtet vorangetrieben wird.

Zeitplan / Roadmap

Dies ist ein grober Plan und sollte verstanden werden als Richtlinie. Die Sprints werden zweiwöchentlich reflektiert, um den nächsten Sprint zu planen.

Tabelle 2 Projekt Zeitplan

Meilenstein ende Sprint	Iteration	Start [SW] – Ende [SW]	Ziele
Grundaufbau des Projektes und Planung	Sprint 1	SW2 – SW4	<ul style="list-style-type: none"> - Ende Aufbau Infrastruktur - Aufbau von Automatischen Tests - Lauffähiger Prototyp - LibGDX vertraut machen - Technische Dokumentation starten - Spielkonzepte ausarbeiten
Kernmechaniken des Spiels	Sprint 2	SW4 – SW7	<ul style="list-style-type: none"> - Technische Dokumentation erweitern - Die Kernmechaniken implementieren
	Sprint 3	SW7 – SW12	Ziele des Sprints werden laufend angepasst und das Projekt wird mit sinnvollen Mechaniken erweitert. <ul style="list-style-type: none"> - Gegner-KI - Ressourcenmechanik - Tower-Defense-Mechanik - Gameplay-Balancing - Testing - UI - Gameplay-Loop
	Sprint 4		
	Sprint 5		
Projekt fertig	Sprint 6	SW12 – SW14	<ul style="list-style-type: none"> - Lauffähiges Projekt - Debugging - Ausführliches Playtesting - Fertiges Game-Booklet - Abgabe Individuellen Leistungen

6 Risiken

Im Rahmen des Projekts haben wir eine Risikoanalyse durchgeführt, um sicherzustellen, dass wir alle potenziellen Herausforderungen proaktiv erkennen und angehen können. Diese Analyse hebt Risiken hervor, die über das übliche Mass hinausgehen, einschliesslich des Einsatzes neuer oder unbekannter Technologien, besonders kritischer Anforderungen und Bereiche, in denen unser Wissen möglicherweise begrenzt ist. Unsere Priorisierung der Risiken richtet sich nach der Kombination aus Eintrittswahrscheinlichkeit und Schadenspotential, wobei wir Massnahmen festgelegt haben, um diese Risiken zu steuern. Diese proaktive Herangehensweise soll die Wahrscheinlichkeit von unerwarteten Hindernissen minimieren und einen reibungslosen Projektfortschritt gewährleisten. Wahrscheinlichkeit und Schadenspotential werden, wie in der Tabelle 3 beschrieben, eingestuft.

Tabelle 3 3 Einstufungs Legende

Einstufung	Wahrscheinlichkeiten	Schadenspotential
Niedrig	Das Risiko ist unwahrscheinlich und tritt nur unter aussergewöhnlichen Umständen ein.	Geringe Auswirkungen auf das Endprodukt; kleinere Anpassungen oder Korrekturen reichen aus.
Mittel	Das Risiko könnte auftreten, ist jedoch nicht zwingend zu erwarten.	Das Problem verursacht moderate Einschränkungen, kann jedoch mit angemessenen Massnahmen kompensiert werden.
Hoch	Das Risiko ist wahrscheinlich und sollte in der Planung berücksichtigt werden.	Es sind alternative Lösungen oder umfassende Anpassungen erforderlich, um das Problem zu bewältigen.
Sehr Hoch	Das Risiko wird mit hoher Sicherheit eintreten, weshalb präventive Massnahmen erforderlich sind.	Das Risiko gefährdet die gesamte Projektstruktur und erfordert eine grundlegende Neuausrichtung oder Umplanung.

Tabelle 44 Risikyanalyse

#	Name	Beschreibung	Wahrscheinlichkeit	Schadenpotential	Massnahme
Strategische Risiken					
1	Technologieentscheidungen	Falls sich herausstellt, dass libGDX nicht optimal für das Projekt geeignet ist, könnte eine Umstellung erforderlich werden.	Niedrig	Sehr Hoch	Frühe Tests und Analysen zur Bestätigung der Tool-Auswahl
Operationelle Risiken					
1	Personalausfall	Eine Person fällt krankheitsbedingt oder durch Abwesenheit aus (z. B. Militärdienst, Krankheit, Kündigung)	Niedrig	Mittel	Vertretungsregelungen definieren, Wissenstransfer sicherstellen, Dokumentation verbessern
2	Fehlende Erfahrung mit libGDX	Einige Teammitglieder haben keine oder nur geringe Erfahrung mit libGDX, was die Entwicklung verlangsamen könnte.	Mittel	Mittel	Gemeinsame Einführung in libGDX, Austausch von Ressourcen, Pair-Programming-Sessions, Erstellen eines Wikis in GitHub mit best-practices
3	Performance und Lesitungsprobleme	Falls das Spiel nicht performant läuft, könnte dies das Spielerlebnis beeinträchtigen.	Mittel	Hoch	Performance-Optimierungen einplanen, Frühzeitige performance tests.
4	Schlechte usability	Die Anwendung ist für Nutzer schwer verständlich oder umständlich bedienbar	Niedrig	Mittel	Frühzeitige Nutzer-Tests

#	Name	Beschreibung	Wahrscheinlichkeit	Schadenpotential	Massnahme
	Sicherheitsrisiken				
1	Lizenzfragen bei Assets & Musik	Falls lizenzierte Inhalte verwendet werden, könnte dies zu rechtlichen Problemen führen.	Niedrig	Niedrig	Sicherstellen, dass alle Assets entweder selbst erstellt oder mit passenden Lizenzen genutzt werden
	Projektmanagement				
1	Zeitliche Einschränkungen	Der straffe Zeitplan von PM4 könnte dazu führen, dass einige Features nicht wie geplant umgesetzt werden.	Mittel	Mittel	Realistische Sprint-Planung, Fokus auf Kernmechaniken, spätere Erweiterungen als Stretch Goals definieren
2	Abstimmung & Kommunikation	Fehlende Kommunikation im Team könnte zu Missverständnissen und ineffizienter Arbeit führen.	Mittel	Mittel	Regelmässige Meetings, klare Aufgabenverteilung -> Nutzung von Kanban-board
3	Balancing & Gameplay-Probleme	Falls das Balancing nicht gut abgestimmt ist, könnte das Spiel zu frustrierend oder zu einfach sein.	Sehr Hoch	Niedrig	Playtesting-Sessions

7 Ausblick

Nach Abschluss von PM4 soll ein funktionaler Prototyp des Spiels vorliegen, die über eine vollständige Gameplay-Loop verfügt. Die zentrale Mechanik des Spiels – das strategische Tower-Defense-Prinzip mit Zeitreise-Funktion – soll implementiert und spielbar sein.

Weiterentwicklungsmöglichkeiten

- Das Projekt dient als Einstieg in die Spieleentwicklung und ermöglicht es dem Team, erste Erfahrungen mit der Umsetzung eines vollständigen Spiels zu sammeln.
- Nach PM4 kann das Spiel als eigene Testumgebung und Spielplatz für neue Ideen weiterverwendet werden.
- Stretch Goals wie zufällige Kartengenerierung, Modding-Support oder zusätzliche Gegnertypen könnten in späteren Versionen realisiert werden, ohne dass eine vollständige Fertigstellung angestrebt wird.

Spielbare, aber nicht vollendete Version

- Ziel ist es, ein spielbares, funktionales Spiel zu entwickeln, das zwar nicht in vollem Umfang ausgereift ist, aber eine abgeschlossene Gameplay-Loop bietet.
- Das Spiel wird kein fertiges Produkt mit vollständigem Feinschliff, sondern eine stabile Grundlage, auf der weiter experimentiert werden kann.
- Bugs und kleinere Mängel werden vorhanden sein, aber die Kernmechaniken funktionieren so, dass das Spiel tatsächlich spielbar ist.

NoTech-Produkt & Dokumentation

- Als abschliessendes NoTech-Produkt könnte eine Projekt-Dokumentation erstellt werden, die den Entwicklungsprozess, Designentscheidungen und technische Details festhält.
- Es wird ein Retro-Game-Booklet erstellt, das an klassische Anleitungen früherer Spiele angelehnt ist. Es enthält ein Tutorial, zusätzliche Informationen, Tipps, Lore und Hintergrundgeschichten zum Spiel.

Fazit

Das Projekt in PM4 dient als Grundlage für eine spielbare Version, die als Proof-of-Concept fungiert. Es wird nicht als vollwertiges fertiges Spiel entwickelt, sondern als funktionierender Prototyp, der es dem Team ermöglicht, weiter mit Spieleentwicklung zu experimentieren und neue Konzepte auszuprobieren.

8 Anhang

Game-Concept

Audio

Während des Spiels läuft im Hintergrund Musik

Während der Runde laufen Soundeffekte

Während des Spiels läuft Ambiance-Music

Artkonzepte

Artconcept für alles entweder durch Bibliothek oder durch Kevin Sonnek.

Lore

Hat unser Spiel eine Art von Lore

- HQ als Zeitreisestein -> Wiederholung eines Tages wenn er aufgeladen ist
- Game Booklet: Im Booklet kann die Lore vertieft werden

Grob-Ablauf

Spielstart:

1. Ladebildschirm
2. Titelscreen
 - a. Neues Spiel
 - b. Spiel Laden (oder wir machen es mehr Roguelike, ohne Laden...)
 - c. Settings
 - i. Keine optionale, Nur das Notwendigste
 - d. Mitwirkende (Optional)
 - e. Exit
3. Keine Schwierigkeitsstufenwahl -> Dynamisch im Spiel
4. Danach startet Spiel

Spielablauf

Laden der Karte oder laden der Overworld

Eine spielbare Karte - Selbst erstellt	Karte zufällig generieren Als Stretch Goal
Evtl. Mit Map-Builder	

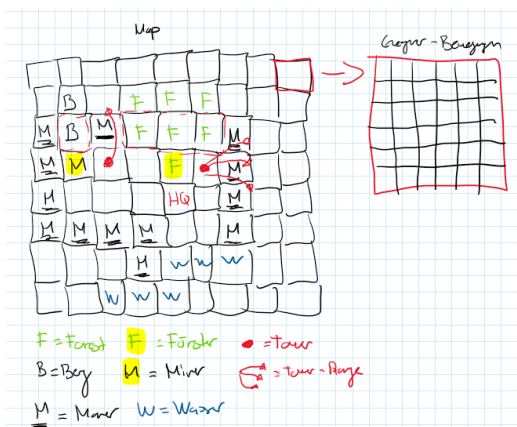
Optional (Tutorial) -> Im Booklet

Beispiel für Kartenlayout: Spielpixel in Quadraten

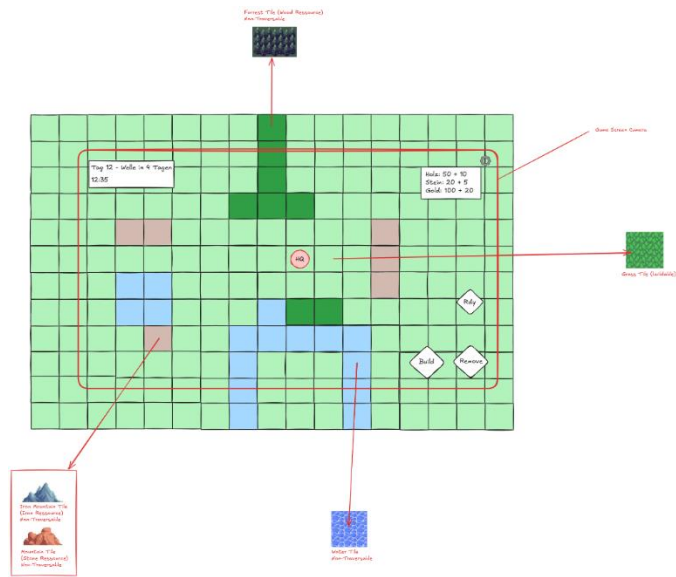
- Optional mit Fog of war
- Bauweite unklar
 - o Spieler kann überall bauen
 - o Oder nur gewisse Entfernung von Gebäude
 - o Oder nur gewisse Entfernung von HQ
 - o Gebäude sind effizienter, je näher sie an HQ sind

Zonenweise:

- Äusserster Ring: Gegner Spanbereich
- Innerer Ring: Baubereich + Ressourcenbereich + HQ

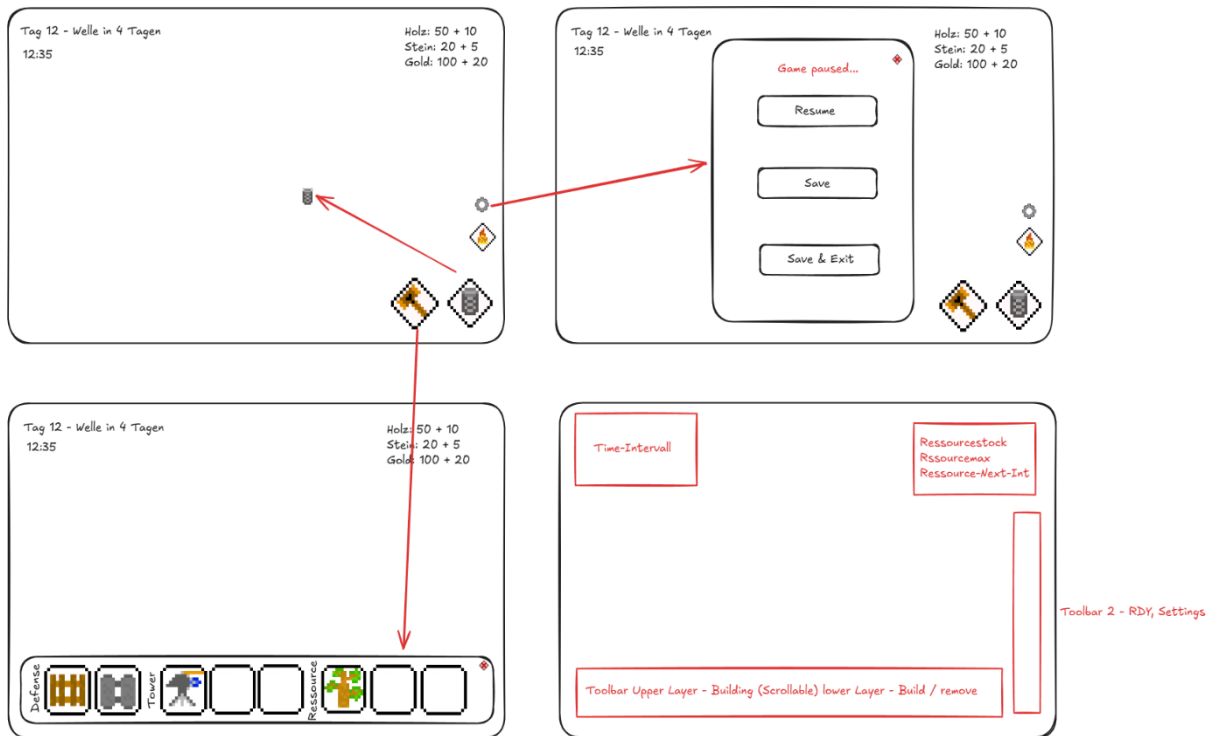


Beispiel Overlay Layout



Jeder Spielpixel ist jeweils nur etwas -> Nicht mehrere Dinge gleichzeitig (Optional: ausser bei Gegner) (Alle Varianten mit fixem Raster für Einfachheit)

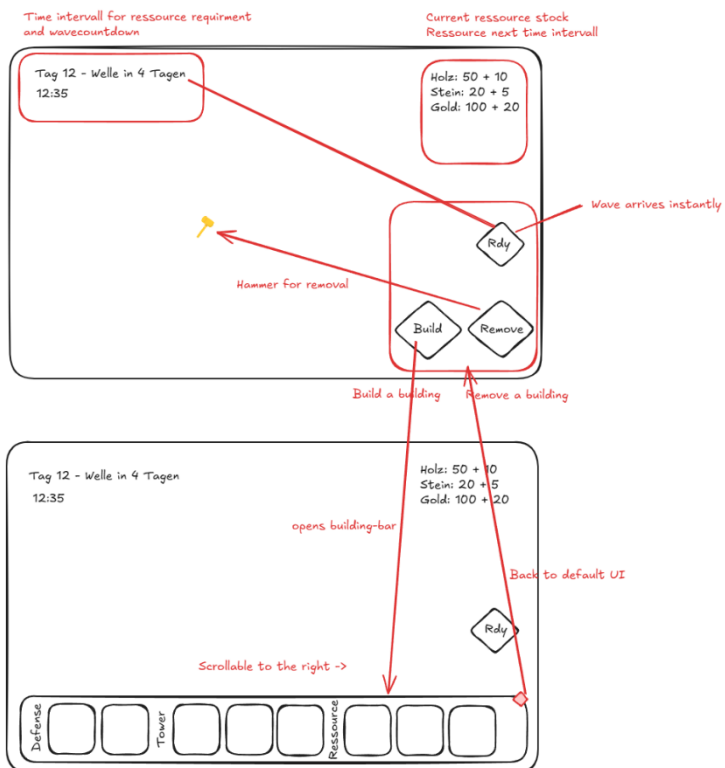
UI-Overview

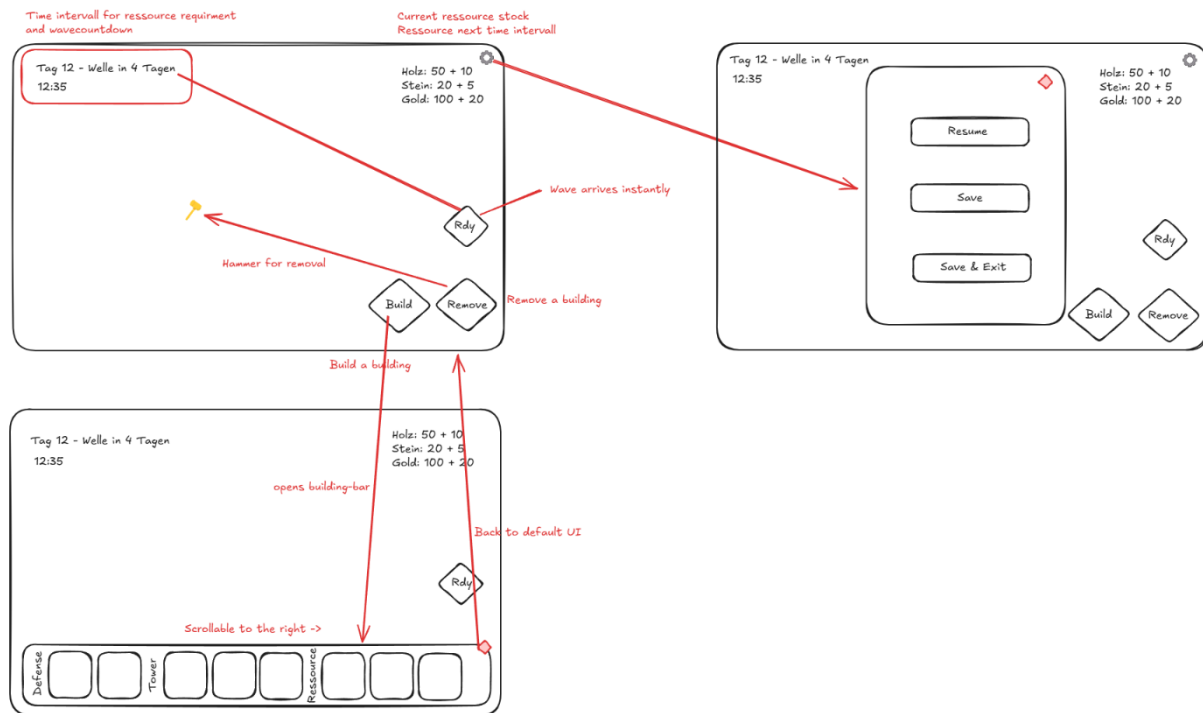


Setting:

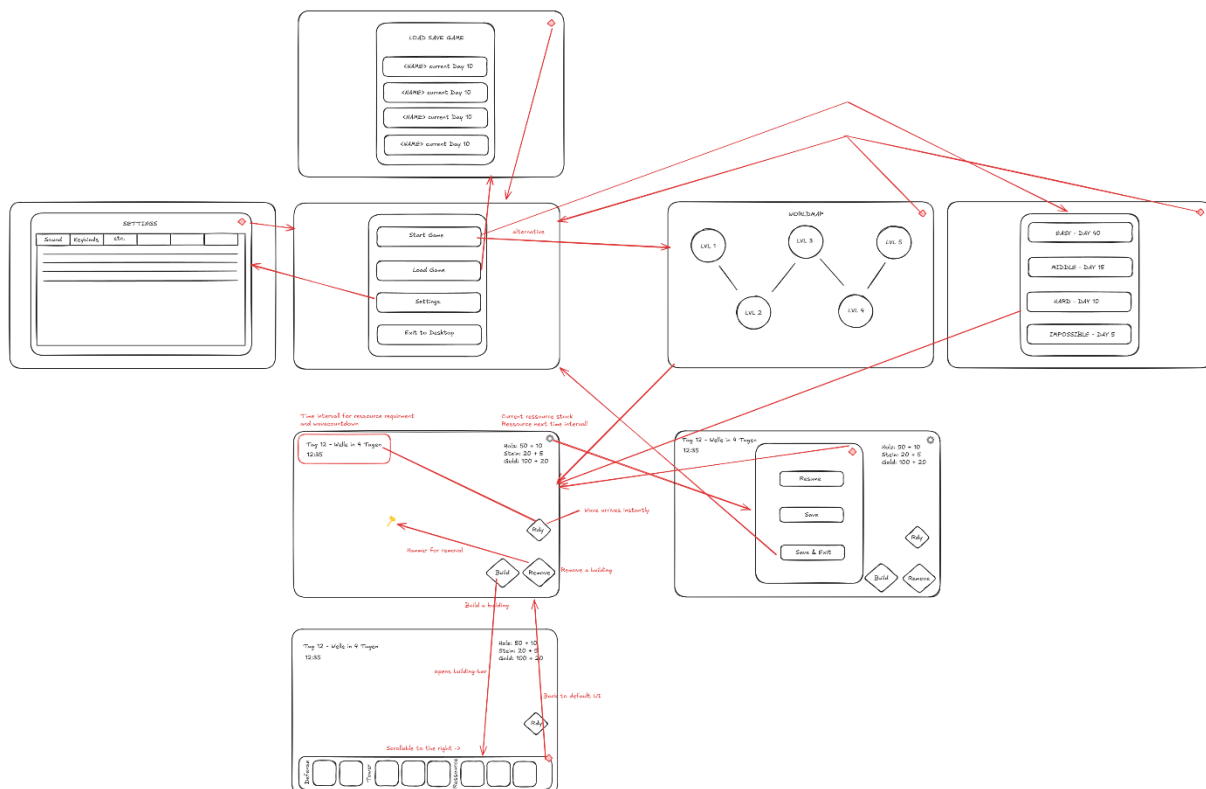
- Resume
- Save
- Load -> Wie kann geladen werden?
- Exit

Spulbutton fehlt -> Allenfalls auch mit pausieren... um zu reagieren...

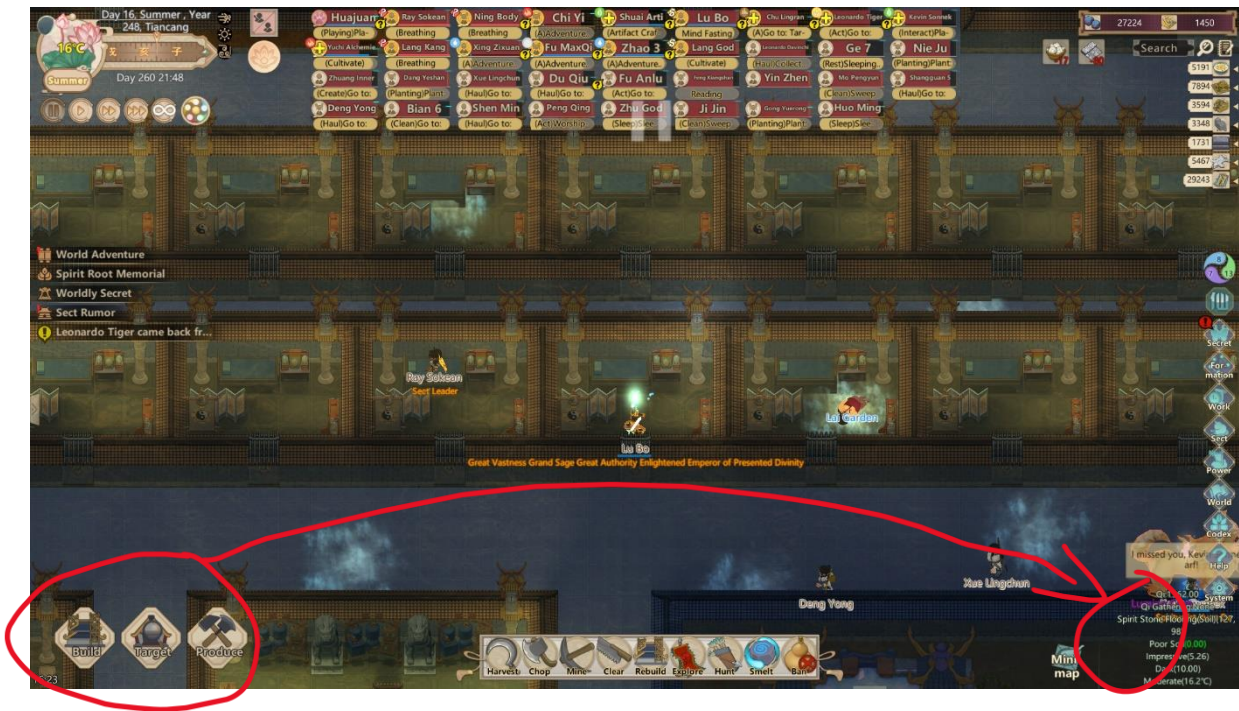




Gesamtübersicht



Inspiration für das UI aus :



Allgemeine Spielmechaniken

Vorhandene Ressourcen

- Währung -> Gold, Geld, Dollar,... (Wird von HQ produziert und verhindert softlock)
 - o Fürs Balancing notwendig
- Holz -> Grundlageressource
- Stein -> Advancedressource
- Eisen -> Endgameressource
 - o Max Ressource mit Abbaugebäude als Ressourcenhalter oder mit Lagerhäuser
 - o Oder einfach unendliche viele Ressourcen
- Optional (Techpoints)

Abbauen der Ressourcen erfolgt durch Gebäudeplatzierung, neben der Ressource -> Ressourcenfeld bleibt stets vorhanden -> Mit Multiplier oder durch effizienz der Entfernung

Gebäude

- HQ
- Ressourcen
 - o Holzfällerei
 - o Steinbruch
 - o Eisenmine
 - o Jedes Gebäude hat leben (Anzahl Gegner die das Gebäude erreichen)
 - Techtree pro Gebäude? -> Mehr leben, Effizienz
 - Max Ressourcen -> Lagerhäuser oder unendliche Ressourcen
- Verteidigungstürme -> Range, wie wird Range definiert -> Fixe Range oder upgradable
 - o Ein Turm -> Upgrade individuell
 - o Verschiedene Türme -> Werden mit Techpoints freigeschaltet, werden individuell verbessert
 - Haben Verteidigungsgebäude eine Techtree -> Range, Schaden, Leben oder Alles Fix
- Abgrenzung
 - o Gebüsch (mit Geld)
 - o Zaun (mit Holz)
 - o Mauer (Mit Stein)
 - o Eisenwand (Mit Eisen)

Wie haben Gebäude Lebenspunkte? Können Gebäude repariert werden mit Ressourcen/Geld

- Jede Runde fixe Zahl
- Oder werden über Runden hinaus Zustände gespeichert
- Oder haben Gebäude keine Lebenspunkte, sondern nur einen Lebenspunkt

Zeit/Gegnerwelle

Zeitintervall -> Spieler baut sich auf -> Ready Sperrt den Spieler vom Bau und Spieler schaut zu:

- Welle kommt erst mit Klick auf Ready

Optional: Während der Well kann Spieler vorspulen

Falls Spieler Welle nicht überlebt

- Spieler kehrt direkt nach Game over Bildschirm zum Titelschirm zurück
- Spieler kann den Tag beliebig wiederholen, aber kann nicht weiter zurück
- **Eventuell mit Lore fixiert -> HQ ist Zeitreisestein**

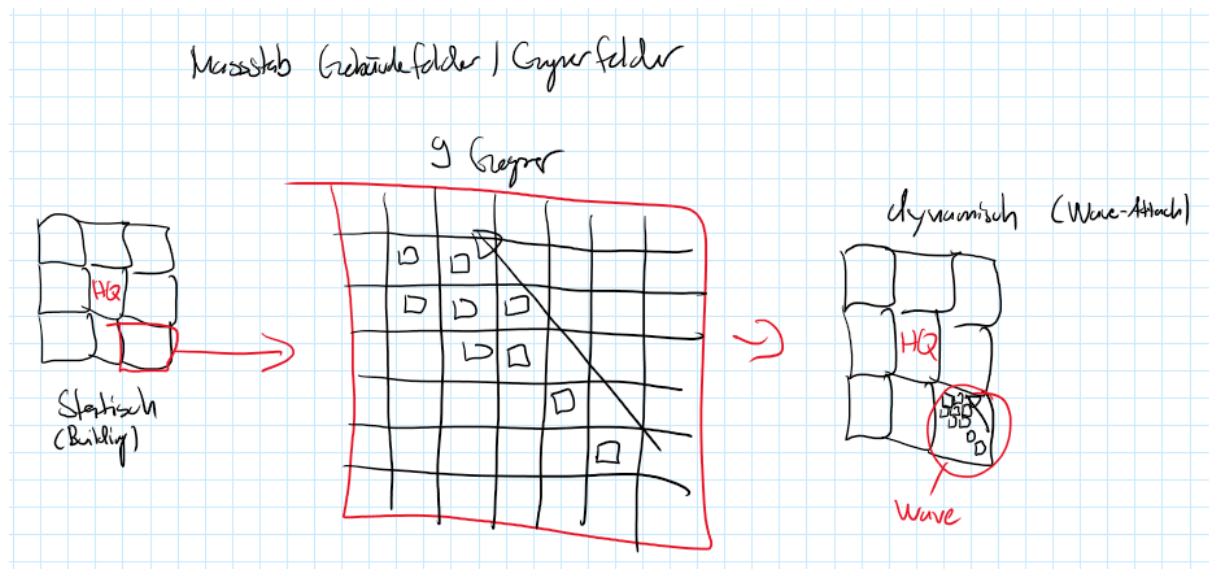
Gegner

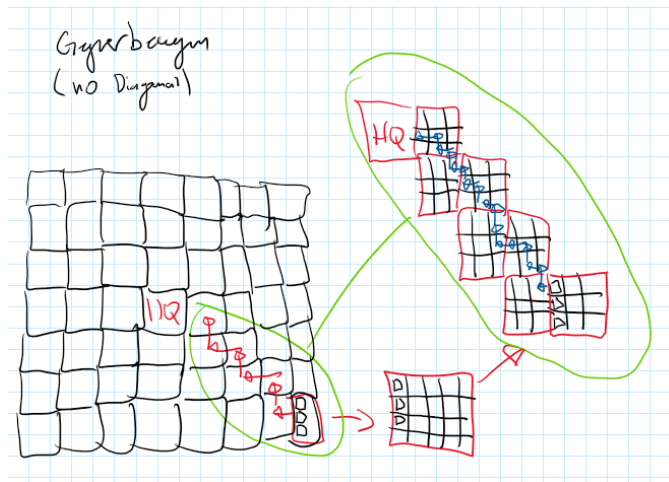
Gegner:

- Grosse Massen, Mehrere Gegner pro Feld
- Ein Gegner pro Feld
- Gegner greifen alles im Weg an -> Verschwinden beim anrichten von schaden oder laufen weiter bis zum tot
- Gegner Koordinaten im N (Spielpixelmatrix) oder R (Unabhängig von Spielpixelmatrix und mit eigenem Koordinatensystem) -> Beides in Richtung HQ

Gegner-KI:

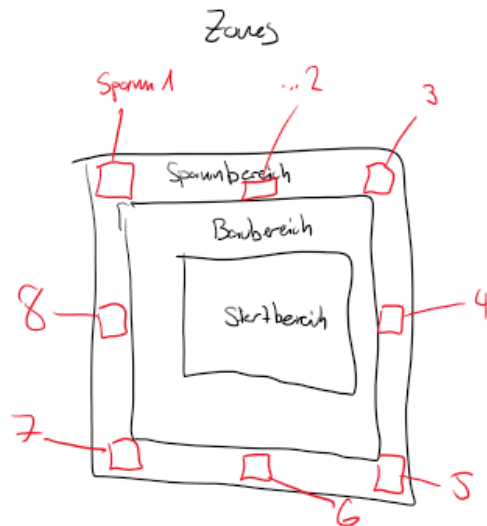
- Läuft auf das HQ, wird es erreicht -> schaden/defeat
- Gegner laufen gradlinig auf das HQ -> Alle vier wege müssen frei sein
- KI läuft um das Spielfeld herum und versucht intelligent das HQ zu erreichen
- Gegner laufen stets gerade auf HQ -> Gegnerwelle startet immer in einer Gerade zum HQ oder Gegner laufen unabhängig und Diagonal Richtung HQ





Spawnbereich:

- Entweder Random ausserhalb des Baubereichs
- Immer eine gerade zum HQ
- Fixe vordefinierte Orte



BALANCING!

Mittels mathematischer Formel:

- Kurvenfunktion -> Je flacher desto einfacher -> Stetiger Wachstum
- Mehrere leichte Wellen hintereinander und dann eine grössere Welle, z.B. alle 5 Runde grosse Welle

Alerts/Quality of Life

- Wo wird der Spieler angegriffen -> Hinweis auf der Kamera
- Ressourcen voll -> Ressource wird rot
- HQ nimmt schaden

Gameplayloop

(Balancing kann immer angepasst werden)

1. Tag

- Spieler hat Startressourcen
- HQ ist auf dem Spielfeld gesetzt
- Spieler hat nur HQ
- **Spieler kann Gebäude bauen (Geld)**
 - **Basisturm**
 - **Gebüsch**
 - **Holzfällerei**
- Welle startet (Entweder dynamisch oder durch klicken auf ready)
 - Gegner laufen auf HQ zu, und zerstören alles im Weg
 - Falls halbdynamisch: Spieler kann 10 Soldaten herumkoordinieren
 - 10 Soldaten machen einfach schaden, müssen auch nicht sterben, haben auch keine Hitbox
- Hat Spieler überlebt
 - Spieler erhält Tagesressourcen
 - Geld vom HQ
 - Holz von allen Holfällereien
 - Stein von allen... etc.
 - (Optional: Gebäude sind repariert)
- Nächster Tag beginnt automatisch oder dynamisch (Je nach dem hat Spieler eine Taste dafür oder er wartet)

2. Tag

- Spieler hat Ressourcen aus letzter Runde...
- Gebäude bleiben an Ort und Stelle
- Zerstörte Gebäude aus letzter Runde, werden nicht «wiederbelebt»
- **Spieler kann Gebäude bauen**
 - **Holzfällerei...etc.**
- Grössere Welle startet (oder mit multiplifier stärker und mehr leben -> Balancing) evtl neue Gegnerarten (je nach Zeit)

Gewinnbedingung

- Entweder keine
- Fixe Anzahl Tage
- Fixe Anzahl Tage, danach freies Spiel