ZHAW School of Engineering

Bachelor of Science in Informatik

Software-Projekt PM4 FS25

**Projektskizze**

**Fortress Forge**

16.03.2025

Fabio Geisser

Levin Hofer

Julien Wyss

Patrick Zürcher

Diego Esposito

Leona Kryeziu

Inhaltsverzeichnis

[1. Projektidee 1](#_Toc193058176)

[2. Stand der Technik / Konkurrenzanalyse 2](#_Toc193058177)

[2.1. Forts (Earthwork Games) 2](#_Toc193058178)

[2.2. Redcon (Hexage) 3](#_Toc193058179)

[2.3. From the Depths (Brilliant Skies) 3](#_Toc193058180)

[2.4. Fortress Forge im Vergleich 4](#_Toc193058181)

[3. Kontextszenario 5](#_Toc193058182)

[3.1. Hauptmenü 5](#_Toc193058183)

[3.2. Vorbereitung 5](#_Toc193058184)

[3.3. Festung bauen 5](#_Toc193058185)

[3.4. Kampf 5](#_Toc193058186)

[4. Weitere Anforderungen 6](#_Toc193058187)

[4.1. Multiplayer (Client-Server-Kommunikation) 6](#_Toc193058188)

[4.2. Balancing 6](#_Toc193058189)

[4.3. UI-Design 6](#_Toc193058190)

[4.4. Soundeffekte 6](#_Toc193058191)

[5. Optionale Anforderungen 7](#_Toc193058192)

[5.1. Kampagne 7](#_Toc193058193)

[6. Ressourcen und Zeitplanung 7](#_Toc193058194)

[6.1. Ressourcen 7](#_Toc193058195)

[6.2. Zeitplanung 7](#_Toc193058196)

[7. Risiken 8](#_Toc193058197)

[7.1. Fehlendes Wissen über die Unity-Gameengine 8](#_Toc193058198)

[7.2. Multiplayer-Kompatibilität 8](#_Toc193058199)

[7.3. Abhängigkeit von Drittsystemen 8](#_Toc193058200)

[7.4. Leistungs- und Skalierungsprobleme 8](#_Toc193058201)

[8. Ausblick 8](#_Toc193058202)

[8.1. Spiel ausbauen 8](#_Toc193058203)

[8.2. Publisher suchen 8](#_Toc193058204)

[8.3. Veröffentlichung auf Steam (und anderen Plattformen) 9](#_Toc193058205)

[9. Quellenverzeichnis 10](#_Toc193058206)

[10. Glossar 11](#_Toc193058207)

# Projektidee

**Art**: Echtzeit-Strategiespiel, Action

**Ziel**: Gegenspieler ausfindig machen und eliminieren

**Sekundäres** **Ziel**: Ressourcen fördern und diese sichern.

**Spielvorgang**: Der Spieler wählt zuerst auf der Karte einen Startpunkt aus, auf dem die zentrale Einheit der Festung gesetzt wird. Es kann nur in einem begrenzten Bereich gestartet werden.

Wenn das Spiel beginnt, kann der Spieler Gebäude platzieren, um Ressourcen wie Metall und Gestein zu gewinnen. Diese kann er nutzen, um Mauern, Fabriken und Waffen herzustellen.

Das Spiel ist zeitlich terminiert. Am Ende des Spieles siegt der Spieler, der die meisten Ressourcen im Speicher gesammelt hat. Doch es führt auch ein anderer Weg zum Sieg: alle gegnerischen Festungen vernichten. Mit Waffenfabriken können Artillerie, Mörser und Laserwaffen hergestellt werden. Mauern und Tore beschützen die Waffen vor feindlichem Beschuss.

Um all diese Gebäude zu betreiben, wird Energie benötigt, welche einen eigenen Typ von Gebäude darstellt (Windräder, Solar, Erdsonden). Wenn der benötigte Energiehaushalt nicht erreicht wird, deaktivieren sich Gebäude und Waffen.

Das Spiel endet, wenn der Timer abläuft oder nur noch ein Spieler übrig ist.

**Allgemeines Spielerlebnis**: Das Spiel soll leicht zu lernen sein und gleichzeitig viel Potenzial und Abwechslung bieten. Der Fokus liegt auf dem möglichst schnellen und effizienten Aufbau einer Festung, sowie der Vielfältigkeit des Kampfes. Es ist primär ein Kampf gegen die Zeit, um die Oberhand über die Gegenspieler zu gewinnen.

# Stand der Technik / Konkurrenzanalyse

Es gibt einige Videospiele, die diesen Aspekt ebenfalls anschneiden.

## Forts (Earthwork Games)

"Forts" ist ein Echtzeit-Strategiespiel, in dem Spieler Festungen entwerfen, bauen und im Kampf erproben. Dabei kommt es auf geschickte Ressourcenverwaltung, taktische Platzierung von Waffen und kluges Verteidigungsverhalten an, um gegnerische Anlagen zu zerstören. Das Spiel setzt auf Physik-basierte Mechaniken, die ein dynamisches Bau- und Zerstörungssystem ermöglichen. Durch stetiges Anpassen und Verbessern der eigenen Konstruktionen entsteht ein vielseitiges strategisches Erlebnis [1].

**Besonderheiten**:

* 2D
* Setzt stark auf Physik-Elemente beim Bau der Festung
* Sehr kompetitiv

Ein Bild, das PC-Spiel, Strategievideospiel, Spielesoftware, Adventure-Spiel enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 1: Screenshot von Forts

## Redcon (Hexage)

"REDCON" ist ein 2D-Strategiespiel, in dem man eine Festung befehligt und sie mit verschiedenen Kanonen, Raketen und anderen Waffensystemen ausstattet, um gegnerische Stellungen zu zerstören. Das Spiel legt grossen Wert auf geschicktes Ressourcen- und Energiemanagement, da Waffen und Verteidigungsanlagen vor dem Kampf sorgfältig platziert und aufeinander abgestimmt werden müssen. Es ist kein Echtzeit-Strategiespiel im herkömmlichen Sinne, da regelmässig mit der Pausenfunktion die Waffen neu eingestellt, und die Besatzung kommandieren werden muss. Daher gibt es keine Multiplayerfunktion [2].

**Besonderheiten**:

* 2D
* Singleplayer
* Ausgeprägte Story

Ein Bild, das PC-Spiel, Spielesoftware, Strategievideospiel, Digitales Compositing enthält.

KI-generierte Inhalte können fehlerhaft sein.Abbildung 2: Screenshot von Redcon

## From the Depths (Brilliant Skies)

From The Depths ist ein komplexes Konstruktions- und Strategiespiel in einer offenen 3D-Welt. Man baut Schiffe, Fluggeräte oder schwimmende Festungen Stein für Stein aus unterschiedlichen Bauteilen zusammen und rüstet sie mit Waffen, Antrieben sowie Verteidigungssystemen aus. Dank eines umfangreichen Physik- und Schadensmodells spielen Statik, Panzerung und Treibstoffversorgung eine grosse Rolle. Im weiteren Spielverlauf kann man die Welt erkundet, verteidigt Konstruktionen gegen feindliche Fraktionen und führt taktische Schlachten [3].

**Besonderheiten**:

* 3D
* Extensiver Baueditor
* Realistische Schadenmodelle und Physik

## Fortress Forge im Vergleich

Fortress Forge kombiniert die Besonderheiten dieser Videospiele, besonders von *Forts* und *Redcon*. Die fundamentalen Game-Mechaniken und -Designs dienen als Grundlage für die Implementationen. Die wichtigsten Aspekte sind:

**3D**: Die Einführung einer zusätzlichen Dimension ermöglicht vielfältige Strategien für die Spieler und Spielerinnen.

**Schneller und einfacher Baueditor:** Anders als in *Redcon* und *From the Depths* wird während dem Spiel die Festung aufgebaut. *Forts* legt grossen Wert auf realistische Physik, sodass der Bau einer stabilen Festung für neue Spieler eine grössere Herausforderung darstellt als das Kämpfen selbst.  
Fortress Forge fasst den Ansatz von *Redcon* auf und führt diesen weiter in den dreidimensionalen Raum. Waffen und Räume (allg. Gebäude) haben fixe Formen und werden der Festung wie Bausteine hinzugefügt. Das Design dieses Baueditor ist intuitiv und einfach zu bedienen, sodass Bauen und Schiessen gleichzeitig bewältigt werden kann.

**Grösse der Karte beeinflusst Spielverlauf:** Das Gelände und die Startposition der Festungen spielt eine zentrale Rolle im Spielverlauf. Starten zwei Spieler nahe beieinander, so gestaltet sich das Spiel kurz und intensiv. Bei weit entfernten Startpositionen haben die Spieler mehr Zeit, diverse Strategien zu verfolgen.

Ein Bild, das Spielzeug enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 3: Screenshot eines Fortress Forge Mockups

# Kontextszenario

Spieler Jens ist ein begeisterter Echtzeit-Strategiespieler. Er kennt grundlegende Konzepte wie Ressourcenmanagement und Tech-Trees.

## Hauptmenü

Nach dem Starten des Videospiels wird Jens von einem Hauptmenü begrüsst. Hier hat er unter anderem die Möglichkeit, einem Online-Spiel beizutreten oder selbst ein Spiel zu hosten. Jens möchte sein Können mit anderen Spielern weltweit messen und eröffnet eine neue Lobby.

## Vorbereitung

Sobald genügend Spieler in Jens‘ Lobby beigetreten sind, definiert Jens, welche Spieler in welchem Team spielen, oder ob alle gegen alle spielen. Jens hat ausserdem die Möglichkeit, eine Spielkarte aus einer Bibliothek auszuwählen, sowie die verfügbaren Gebäude festzulegen. Sobald alle Spieler bereit sind, startet Jens das Spiel.

## Festung bauen

Jens Strategie für dieses Spiel ist „Full-Economy“, also zuerst eine stabile Ressourcenversorgung aufbauen und sich vorerst den Kämpfen enthalten. Im UI werden die Gebäude angezeigt, die er auf einem hexagonalen Spielfeld platzieren kann. Mit den geeigneten Gebäuden kann Jens seine Gestein-, Metall- und Energieproduktion ausbauen. Wichtige Gebäude beschützt er mit Verteidigungsanlagen wie Mauern, Schutzwälle und Luftabwehr. Damit ist der Grundstein für eine mächtige Festung gelegt.

## Kampf

Jens hat eine breite Auswahl an verfügbaren Waffen wie Artillerie, Mörser, Raketen und Laserwaffen. Diese kann er wie die anderen Gebäude vom UI auswählen und auf dem Spielfeld platzieren. Munitionsfabriken stellen Munition aller Art für die Waffen her.  
Jens hat eine feindliche Festung mit einem Aufklärungsgebäude ausgemacht und möchte diese nun angreifen. Dies gestaltet sich wie folgt:

1. **Waffe ausrichten**

Je nach freier Schussbahn kann auf zwei Arten das Ziel ins Visier genommen werden.

* 1. **Direkt**  
     Die Waffe hat direkte Sicht auf das Ziel. Der Spieler kann in einem Fernrohr das Ziel ins Visier nehmen. Dabei werden wichtige Informationen wie Windgeschwindigkeit oder Waffentemperatur angezeigt.
  2. **Indirekt**  
     Die Sicht wird vom Gelände verdeckt. Auf einer Radaransicht kann die Zielkoordinaten eingegeben werden.

1. **Munitionstyp wählen**

Brandgeschoss, hochexplosiv, panzerbrechend, …

1. **Feuer**Jens kann alle bereitgemachten Waffen geordnet abfeuern. Alle Geschosse werden auf einem Radar angezeigt. Beim ballistischen Flug zum Ziel wird das Geschoss von Wind und Wetter beeinflusst.

Im UI sind alle gebauten Waffen mit Statusinformationen übersichtlich dargestellt, sodass Jens schnell und unkompliziert eine Waffe abfeuern kann.

# Weitere Anforderungen

Damit das Spiel auch wirklich spielbar wird, müssen folgende weitere Anforderungen erfüllt sein.

## 4.1. Multiplayer (Client-Server-Kommunikation)

* Synchronisation: Konsistentes Spielgeschehen zwischen allen Teilnehmern (z. B. gleiche Ressourcenzählung, gleiche Kartenupdates).
* Server-Performance: Bei steigender Spielerzahl oder sehr grossen Karten muss der Server stabil und skalierbar laufen.
* Netzwerk-Latenz: Techniken wie Client-Side-Prediction oder Interpolation können das Spielerlebnis trotz Verzögerungen angenehm halten.
* Sicherheit: Schutz vor Cheating, etwa durch Validierung wichtiger Aktionen auf dem Server.

## Balancing

* Ressourcen- und Baukosten: Müssen angepasst sein, damit sich weder defensive noch offensive Strategien unverhältnismässig lohnen.
* Waffenschaden und -Reichweite: Feintuning, um kein „OP“-Waffensystem zuzulassen und dennoch Unterschiedlichkeit zu bewahren.
* Iteratives Vorgehen: Sammeln von Feedback (z. B. durch interne Tests, Beta-Tests), um kontinuierlich nachzusteuern.

## UI-Design

* Klarheit und Übersicht: Ein gut strukturiertes Interface (Menüs, Icons, Tooltipps) ermöglicht schnelle Entscheidungen.
* Feedback und Kontext: Wichtige Zustände (z. B. Energiemangel) und Warnungen (Angriff!) sollten sofort erkennbar sein.
* Einstiegshilfe: Tooltips, Tutorials und ggf. ein „Assistent“, der Spielern bei wichtigen Schritten hilft, senken die Einstiegshürde.

## Soundeffekte

* Akustisches Feedback: Aktionen wie Ressourcenabbau, Baufortschritt oder Schüsse sollten passend vertont sein, damit der Spieler wichtige Ereignisse nicht verpasst.
* Atmosphäre: Hintergrundmusik und Umgebungsgeräusche (Wind, Wetter, Maschinen) steigern die Immersion.
* Sprach- und Signal-Hinweise: Warnmeldungen oder Einheiten-Rückmeldungen (z. B. „Feuerbereit!“) für schnelle Orientierung.

# Optionale Anforderungen

Nicht unbedingt notwendige Features, die aber für ein komplettes Spiel notwendig sind.

## 5.1. Kampagne

* Ein Einzelspieler-Modus, der keine Internetverbindung benötigt.
* Erarbeitung einer Story mit einer ausführlichen Hintergrundgeschichte.

# Ressourcen und Zeitplanung

## 6.1. Ressourcen

Das Entwicklerteam besteht aus sechs Vollzeitstudierenden. Da das PM4 Modul 4 ETCs gibt (entspricht 4 \* 25 Stunden), steht für die Umsetzung des Projekts ungefähr 600 Arbeitsstunden zur Verfügung.

**Benötigte Software**

* Unity 6 (Version 6000.0.37f)
* Blender
* IDE für C#

## Zeitplanung

Um den Fortschritt des Projekts zu verfolgen, dient folgende grobe Zeitplanung:

|  |  |  |
| --- | --- | --- |
| Datum | Sprint | Meilenstein |
| 06.03.2025 – 20.03.2025 | #1 | Mechanik |
| 20.03.2025 – 03.04.2025 | #2 | Funktionalität |
| 03.04.2025 – 17.04.2025 | #3 | Spielbar |
| 17.04.2025 – 01.05.2025 | #4 | Online (Peer2Peer) |
| 01.05.2025 – 15.05.2025 | #5 | Reserve |
| 15.05.2025 – 29.05.2025 | #6 | Produkt fertigstellen |

# Risiken

## 7.1. Fehlendes Wissen über die Unity-Gameengine

Die meisten Entwickler haben wenig bis kein Wissen über die Unity-Engine und ihre Funktionalitäten. Darum muss viel in Recherche und Ausprobieren investiert werden, was zu erheblichen Zeitverzögerungen führen kann.

## Multiplayer-Kompatibilität

Es kann sein, dass die Firewall des Clients die Verbindung zum Server trennt, wenn diese als unsicher erkannt wird. Da wir keinen Einfluss über das Heimnetzwerk jedes Clienten haben, kann das zum Problem werden.

## Abhängigkeit von Drittsystemen

Für den Multiplayer verwenden wir das Fishnet-Framework. Für das Soundsystem kann das FMOD-Framework genutzt werden. Dies stellt zusätzliche Komplexität für die Entwicklung des Spieles dar und braucht zusätzlich Zeit, um die Frameworks in das Spiel zu integrieren.

## Leistungs- und Skalierungsprobleme

Bei vielen gleichzeitigen Spielern oder sehr grossen Karten kann es zu hoher CPU-/RAM-Auslastung kommen, was Lags oder Ausfälle zur Folge haben kann. Aufwändige Grafikeffekte, Kollisionsabfragen und komplexe Berechnungen (z.B. ballistische Flugbahnen) können schwächere Systeme überfordern.

# Ausblick

Das primäre Ziel dieses Projekts ist es, ein robustes und skalierbares Fundament für ein grosses Videospiel zu schaffen. Aspekte wie Grafik, Gameplay oder Progression werden innerhalb des PM4-Scopes nicht berücksichtigt. Das bedeutet folglich, dass am Ende des Semesters kein Spiel entsteht, das veröffentlicht werden kann.

## 8.1. Spiel ausbauen

Das Spiel wird stetig erweitert, um zusätzliche Inhalte, neue Gameplay-Elemente und mehr strategische Tiefe zu bieten. Dabei steht weiterhin das schnelle und effiziente Bauen im Mittelpunkt, während neue Features für Abwechslung und langfristige Motivation sorgen.

Dazu zählen bspw. das Hinzufügen neuer Gebäudearten, Karten oder Gamemodes. Auch die Umsetzung einer Einzelspieler-Kampagne hilft der Langlebigkeit eines Spieles.

## Publisher suchen

Ein Publisher wird benötigt, um das Projekt langfristig zu unterstützen und zu vermarkten. Ein geeigneter Publisher kann Ressourcen bereitstellen und das Spiel einem noch grösseren Publikum zugänglich machen, während das Entwicklerteam sich weiter auf Feinschliff und Erweiterungen konzentrieren kann.

Ein schweizerischer Publisher (wie zum Beispiel *StrayFawn Studio* [4]) wird bevorzugt, um den lokalen Markt zu unterstützen.

## Veröffentlichung auf Steam (und anderen Plattformen)

Schliesslich ist eine Veröffentlichung auf Steam [5] geplant, um das Spiel einer breiten Spielerschaft zugänglich zu machen. Dabei wird vor allem auf die technische Stabilität, Benutzerfreundlichkeit und ansprechende Präsentation geachtet.

# 9. Quellenverzeichnis

[1] «Earthwork Games Webseite» [Online]. Stand: 03.2025

URL: <https://www.earthworkgames.com>

[2] «Hexage Official Website» [Online]. Stand: 03.2025

URL: <https://hexage.net/redcon/index.html>

[3] «Brilliant Skies – The sky’s the limit» [Online]. Stand: 03.2025

URL: <https://www.brilliantskies.co.uk/#OurGamesFTD>

[4] «Stray Fawn Studio» [Online]. Stand: 03.2025

URL: <https://strayfawnstudio.com>

[5] «Steam» [Online]. Stand: 03.2025

URL: [https://store.steampowered.com](https://store.steampowered.com/?l=german)

# 10. Glossar

* **Blender**: Ein kostenloses Open-Source-Programm für 3D-Modellierung, Animation, Rendering und Videobearbeitung, das in vielen Bereichen – von der Erstellung von Videospielen bis zur Filmproduktion – eingesetzt wird.
* **Echtzeitstrategiespiel (RTS)**: Ein Videospiel-Genre, bei dem Spieler ihre Einheiten und Ressourcen gleichzeitig (in Echtzeit) managen und strategische Entscheidungen treffen, um Gegner zu besiegen oder bestimmte Ziele zu erreichen.
* **Game-Design**: Der Prozess der Konzeption, Planung und Umsetzung aller Spielelemente (z. B. Story, Mechaniken, Level) mit dem Ziel, ein fesselndes und ausgewogenes Spielerlebnis zu schaffen.
* **Game-Mechanik**: Einzelne Regeln oder Systeme in einem Videospiel, die das Spielerlebnis prägen, etwa Bewegung, Sammeln von Ressourcen, Kampf oder Interaktionen mit der Spielwelt.
* **Gamemode**: Voreingestellte Variante oder Regelwerk in einem Spiel, das den Ablauf oder die Ziele verändert. Beispiele sind etwa „Team Deathmatch“, „Capture the Flag“ oder „Survival“-Modi.
* **Gameplay**: Das Zusammenspiel aus Steuerung, Spielmechaniken und Spielerinteraktionen, das letztlich das Spielgefühl und den Spielspass definiert.
* **Lobby**: Ein virtueller Warteraum in einem Onlinespiel, in dem sich Spieler vor einem Match einfinden, Gruppen bilden oder Einstellungen anpassen, bevor das eigentliche Spiel beginnt.
* **Progression (Fortschritt)**: Ein System in Videospielen, das den fortlaufenden Entwicklungsprozess des Spielers oder der Spielfigur beschreibt. Durch das Sammeln von Erfahrung, das Erreichen von Levelaufstiegen und das Freischalten neuer Fähigkeiten, Gegenstände oder Bereiche entsteht eine stetige Weiterentwicklung, die den Spielreiz und die Langzeitmotivation erhöht.
* **Publisher**: Ein Unternehmen, das für die Finanzierung, Vermarktung und den Vertrieb eines Videospiels verantwortlich ist und oft eng mit den Entwicklerstudios zusammenarbeitet.
* **Steam**: Eine digitale Vertriebsplattform für Computerspiele, die eine grosse Sammlung von Titeln zum Kauf und Download anbietet, sowie Multiplayer-Funktionen und soziale Features unterstützt.
* **Tech-Tree (Technologiebaum)**: Ein Fortschrittssystem in Spielen, insbesondere in Strategiespielen, bei dem durch Forschung oder Entwicklung neue Einheiten, Gebäude, Fähigkeiten oder Technologien freigeschaltet werden können.
* **Unity**: Eine beliebte, plattformunabhängige Spiele-Engine, mit der Entwickler Videospiele für verschiedene Systeme (PC, Konsole, Mobile) erstellen können.