PM4 – Gameprojekt «Fortress Forge»

## Kurzbeschreibung

**Art**: Echtzeit-Strategiespiel, Action

**Ziel**: Ressourcen fördern und diese sichern

**Sekundäres** **Ziel**: Gegenspieler ausfindig machen und eliminieren

**Spielvorgang**: Der Spieler wählt zuerst auf der Karte einen Startpunkt aus, auf dem die zentrale Einheit der Festung gesetzt wird. Einige Startpunkte haben mehr Ressourcenvorkommen als andere. 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 termininert. 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.

## Bauen

Die Karte ist in ein 3D hexagonales Gitter aufgeteilt. Jedes Gebäude hat eine Form, die in Hexagon unterteilt sind (wie Tetris-Blöcke). Neue Gebäude können nur an bereits existierende Gebäude platziert werden.  
**Wichtig:** Gebäude können gestapelt werden.

## Schiessen

Der Action-Aspekt liegt im manuellen Abfeuern jeder platzierten Waffe. Dabei kommt der Spieler in eine Zielansicht (die je nach Waffentyp anders aussehen kann) die nach folgendem Schema verläuft:

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.
2. **Munitionstyp wählen**Brandgeschoss, hochexplosiv, panzerdurchdringend, …
3. **Feuer**Alle Geschosse werden auf einem Radar angezeigt. Beim ballistischen Flug zum Ziel wird das Geschoss von Wind und Wetter beeinflusst.

## Implementationsziele

**Hexgrid**Implementation eines dreidimensionalen Hexgrids, das das Rückgrat für das Platzieren der Gebäude darstellt.

**Gebäude Blueprints (Tetrisblöcke)**Ein Gebäude hat eine fixe Grösse und Höhe, sowie Attribute. Dafür können ScriptableObjects verwendet werden.

**Gamecontroller**Verwaltet den Stand des aktuellen Spiels und alle Spieler.

**Inputcontroller**Regelt die Userinputs für Kameraführung und das Bauen.

**Buildmode**Mit Raycast Position der Maus bestimmen und Drag and Drop Logik implementieren. HUD mit Tastenbelegung entwerfen.

**Kameraführung**Mit der Maus kann man um die Festung rotieren oder frei bewegen. Die Kamera kann nur in einem begrenzten Bereich kontrolliert werden.

**Zielsystem entwerfen (direkt und indirekt)**Mit First-Person oder der Eingabe von Koordinaten kann eine Waffe ausgerichtet werden. Eine Vorhersage des Einschlags wird im HUD angezeigt.

**Logik der Geschosse**Die Flugbahn der Geschosse müssen simuliert werden (ballistisch für Artillerie, Zielsuchend für Raketen, …)  
Auf Hinsicht des Moduls «Physics Engine» kann auch Wind simuliert werden.

Diese Aspekte bilden den Grundstein des Spiels, auf dem weiter aufgebaut werden kann. Da dies alles in der Unity-Gameengine implementiert wird und die Entwickler noch eingeführt werden müssen, sind die Ziele vorerst simpel gehalten.

## Frameworks

* Unity

## Programmiersprachen

* C#