Anforderungsanalyse

Created by	Ⓝ Noah
Created time	@February 28, 2025 5:36 PM
Category	Planning Software

Einführung & Projektziel

Unser Spiel wird, wie im Dokument Spielidee.pdf beschrieben, ein rundenbasiertes Ressourcenmanagement-Strategie-Spiel mit einem Setting in der nordischen Mythologie sein.

Das Projektziel ist neben dem Erlernen von neuen Fähigkeiten und Wissen unsererseits, ein spaßiges und strategisch tiefes Spiel zu entwickeln, welches gewaltfreie Konkurrenz ermöglicht und einen Fokus auf gute Ressourcenverwaltung legt.

Im Folgenden werden die Anforderungen beschrieben. Nicht essenzielle Vorschläge werden dabei mit “(?)” markiert.

Funktionale Anforderungen

Spielmechanik

- Hexfeld-Interaktionen
  - Platzieren und Entfernen (?) von Gebäuden
  - Ressourcenabbau
  - Upgrades
  - Rituale
    - Göttersegen

        - Flüche
    - Übersicht über Gebäude-Output (bspw. mit Mauszeiger drüberfahren) (?)
    - Prozedurale Generierung der Karte (?)
        - Felder (Layout und nutzbare Runen)
        - Strukturen (Chance auf einem Feld generiert zu werden
            - Fallen
            - Strukturen (bspw. Ruinen → höhere Chance auf Artefakte)
- Rundenbasiertes System (Reihum)
    - Felder erwerben (0-3)
    - Gebäude Bauen / Abreißen (?) / Upgraden / Benutzen
- Mehrspieler Funktionalität
    - Netzwerkkommunikation
    - Synchronisation der globalen Karte
    - genauer: siehe Client/Server

Ressourcen- & Wirtschaftssystem

- Ressourcentypen: Runen, Energie
- Balancing in Bezug auf Preise von Gebäuden und Ritualen
- Upgrade Möglichkeiten von Gebäuden

Spielerinteraktion (?)

- Kooperative Elemente, wie bspw. Zusammenarbeit an bestimmten Ritualen (?)
- Kompetitive Mechaniken, wie bspw. Sabotage

Benutzeroberfläche

- Globale Hexfeld-Karte
- Seitenleiste mit Gebäuden, welche auf der Karte platziert werden können
- Anzeige über aktuellen, persönlichen Runen- und Energiehaushalt

- Leaderboard, auf welches jederzeit zugegriffen werden kann (?)
  - Übersicht über Punkte / Ressourcen / Götter / ...
- Chat (sowohl in Lobby als auch im Spiel)
- Title Screen (?)
- Menü
  - Einstellungen
  - Spiel erstellen / beitreten
  - Tutorial (?)
  - Übersicht über bereits kennengelernte Götter und Gaben (?)

Client-Anforderungen (Java)

- Spielbrett: dynamisch gezeichnetes Hexagonalraster; klickbare Felder zum Platzieren von Objekten (z.B. Gebäuden)
- Spieler-Panel 1: Anzeige von Spielername, Farbe?, Ressourcen und Siegespunkte?
- Spieler-Panel 2: Buttons für Aktionen wie z.B. Kaufen, Handeln usw)
- Zug-Anzeige: markiert den aktuellen Spieler
- weitere UIs: Handel, Artefakte (?)
- Lobby und Matchmaking: Liste mit offenen Servern und Spielern, Option zum Erstellen / Beitreten eines Spiels
- Chat-System: In-Game-Chat zur Kommunikation mit anderen Spielern

Server-Anforderungen (Java)

Spielmechanik:

- Zugverwaltung: Sicherstellen der Reihenfolge der Spieler
- Würfeln? / Ressourcenverteilung: Evtl. korrekte Verteilung der Ressourcen sicherstellen
- Bauen / Handeln: Überprüfen der Platzierungsregeln für Gebäude (u.Ä.), Validierung von Handelsangeboten

- Spezielle Aktionen: Verwalten von speziellen Effekten (Karten, Artefakte, was weiß ich)

Verwaltung

- Server sendet Updates an Clients, wenn sich der Spielzustand ändert
- Clients senden Aktionen (bauen, handeln, würfeln, etc.)
- Spielzustandsverwaltung im Speicher: GameInstance zur Speicherung vom Spielzustand
- Behandlung von Verbindungsabbrüchen: Spieler können wieder beitreten und ihren Zustand wiederherstellen
- Verwalten mehrerer Spielinstanzen und Lobbies gleichzeitig

Nicht-funktionale Anforderungen

Performance & Skalierbarkeit

- Das Spiel soll maximal 4 Spieler gleichzeitig unterstützen.
- Eine Spielrunde soll nicht länger als eine Minute für einen Spieler dauern. (Hard Limit)
- Der Server muss 4 gleichzeitigt Sitzungen verarbeiten können, ohne spürbare Verzögerungen.
- Ladezeiten für Karten und Menüs dürfen 10 Sekunden nicht überschreiten (angestrebt 5)

Sicherheit

- Das Spiel muss sicherstellen, dass Spieler keine unerlaubten Aktionen durch direkte API-Manipulation oder Netzwerkpaket-Manipulation durchführen können.
- Falls ein Spieler die Verbindung verliert, muss eine Re-Join-Funktion vorhanden sein, ohne dass der Spielstand verloren geht.

Benutzerfreundlichkeit & UX (User Experience)

- Die UI sollte intuitiv und leicht verständlich sein, auch für neue Spieler.
- Es soll eine **eingängige und einfache Steuerung** geben (z. B. Maussteuerung mit klarer Interaktion auf Hexfeldern).

- Ein Tutorial-Modus (?) oder eine Hilfefunktion soll Anfängern den Einstieg erleichtern.
- Low Priority: **Sound- und Grafikeinstellungen** sollen anpassbar sein (z. B. Lautstärke) (?)

Wartbarkeit und Erweiterbarkeit

- Der Code soll modular aufgebaut sein, sodass neue Spielmechaniken leicht integriert werden können.
- Es sollen **detaillierte Logs** existieren, um Fehler einfacher zu debuggen.
- Das Spiel soll so entwickelt sein, dass **zukünftige Erweiterungen** (z. B. neue Gebäude, Karten, Events) ohne große Änderungen implementiert werden können.

Zuverlässigkeit & Verfügbarkeit

- Der Server sollte eine **Uptime von mindestens 99%** haben.
- Abstürze sollen **automatisch erkannt und geloggt** werden.
- Falls ein Spiel crasht, soll eine **Auto-Speicherfunktion** einen letzten Stand wiederherstellen können.

Kompatibilität

- Das Spiel soll auf Windows, macOS und Linux spielbar sein.
- Es soll mit **verschiedenen Bildschirmauflösungen** kompatibel sein (z. B. 1080p, 1440p, 4K).
- Es sollen **keine High-End-PCs notwendig sein**, das Spiel soll auf durchschnittlicher Hardware flüssig laufen.