**Pflichtenheft**

**Version:** 0.1

**Datum:** 05.02.2024

DOKUMENTVERSIONEN

|  |  |  |  |
| --- | --- | --- | --- |
| Versionsnr. | Datum | Autor | Änderungsgrund / Bemerkungen |
| 0.1 | 05.02.2024 |  | Ersterstellung |
| 0.2 |  |  |  |
|  |  |  |  |
|  |  |  |  |

INHALT

[DOKUMENTVERSIONEN 1](#_Toc158017293)

[INHALT 2](#_Toc158017294)

[1. Einleitung 3](#_Toc158017295)

[1.1 Allgemeines 3](#_Toc158017296)

[1.2 Verteiler und Freigabe 3](#_Toc158017297)

[1.3 Reviewvermerke und Meeting-Protokolle 3](#_Toc158017298)

[2. Konzept und Rahmenbedingungen 4](#_Toc158017299)

[2.1 Benutzer / Zielgruppe 4](#_Toc158017300)

[2.2 Ziele des Anbieters 4](#_Toc158017301)

[2.3 Ziele und Nutzen des Anwenders 4](#_Toc158017302)

[2.4 Systemvoraussetzungen 4](#_Toc158017303)

[2.5 Ressourcen 4](#_Toc158017304)

[2.6 Übersicht der Meilensteine 4](#_Toc158017305)

[3. Anforderungsbeschreibung 5](#_Toc158017306)

[3.1 1. Anforderung 5](#_Toc158017307)

[3.2 2. Anforderung 5](#_Toc158017308)

[3.3 3. Anforderung 6](#_Toc158017309)

[3.4 4. Anforderung 6](#_Toc158017310)

[3.5 5. Anforderung 7](#_Toc158017311)

[4. Technische anforderungen 8](#_Toc158017312)

[4.1 IntelliJ 8](#_Toc158017313)

[4.2 JDK 8](#_Toc158017314)

[4.3 GIT 8](#_Toc158017315)

[4.4 JAVAFX 8](#_Toc158017316)

[4.5 FXGL 8](#_Toc158017317)

[4.6 Paint.net 8](#_Toc158017318)

[5. Genehmigung 9](#_Toc158017319)

[6. Anhang 10](#_Toc158017320)

# Einleitung

## Allgemeines

### Ziel und Zweck dieses Dokuments

Dieses Pflichtenheft beschreibt wie die Forderungen für das 2D-Spiel NanoNode-Nexus aus dem Lastenhefts konkret umgesetzt werden sollen.

### Projektbezug

Es handelt sich bei dem Projekt „NanoNode-Nexus“ um ein alleinstehendes Projekt, welches in Form eines Schulprojekts bearbeitet wird.

### Abkürzungen

Im folgenden wird Pflichtenheft wird „NanoNode-Nexus“ als „NNN“ bezeichnet.

### Ablage, Gültigkeit und Bezüge zu anderen Dokumenten

Das Pflichtenheft bezieht sich auf die geforderten Funktionen des Lastenhefts. Phrasen wie „Sektion 4.4“ sind auf dem Lastenheft zurück zu führen.

## Verteiler und Freigabe

### Verteiler für dieses Lastenheft

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rolle | Name | Telefon | E-Mail | Bemerkungen |
| Projektleiter | Maikel/Jonas-Luca | - | - | - |
| Projekt-Schichtleiter | Maikel | - | - | - |
| Tastenhengst | Jonas-Luca | - | - | - |
| Firewall-Flüsterer | Maikel | - | - | - |

## Reviewvermerke und Meeting-Protokolle

### Erstes bis n-tes Review

Das erste Review findet in KW 6 statt.

# Konzept und Rahmenbedingungen

## Benutzer / Zielgruppe

Die Zielgruppe hinter dem Projekt NNN sind Computerspiel-intusiasten mit einem besonderen Anreiz an den Genren

* Tower-Defense
* Labyrinth

Es sind weitere Gruppen nicht auszuschließen.

## Ziele des Anbieters

Das Ziel hinter NNN ist ein vorallem simples und Benutzerfreundliches Spiel, welches langanhaltende Freude und Spielspaß im Geist hinterlässt.

## Ziele und Nutzen des Anwenders

Förderung des Strategischen denkens des Anwenders, durch entwicklung von effektiven Verteidiungsstrategien um die gegebenen Herausforderungen zu bewältigen. Zudem wird durch das Vorhersehen von gegnerischen Bewegungen und das proaktive Platzieren von Türmen vorausschauendes Denken erfordert welches erweitert werden könnte. Der Spieler soll natürlich auch sein durchhalte vermögen erweitern können durch das wiederhilte Versuchen von anspruchsvollen Leveln.

## Systemvoraussetzungen

Ein Computer mit mindestens Windows 7, sowie 2 GB RAM sind vorausgesetzt um das Spiel ohne Komplikationen genießen zu können.

## Ressourcen

Im besten Fall verfügt der Anwender über eine Maus.

## Übersicht der Meilensteine

|  |  |
| --- | --- |
| **Vorbereitungsphase** | |
| Geeignete Pakete für JavaFX zusammen suchen |  |
| Erste Testprojekte mit einzelnen Schwerpunkten erstellen |  |
| **Implementierung und Test** | |
| Klassen des UML-Klassendiagrammes entwickeln und Testen |  |
| Grafische Oberfläche mit passenden Grafiken und Audio Dateien versehen |  |
| **Einführung** | |
| Projektpräsentation |  |
| Early-Access im Steam-Store |  |
| **Voraussichtlicher Verkaufsstart** |  |

# Anforderungsbeschreibung

Detailierte Beschreibung der im Lastenheft geforderten Anforderungen.

## 1. Anforderung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | Td01 | **Nichttechnischer Titel** | | Platzierung und Funktion der Verteidigungsgebäude | | |
| **Quellen** | Gameplay/Konzeptdokument | | **Verweise** | Sektion 4.2 Spielmechanik | **Priorität** | Hoch |

### Beschreibung

Im Spiel sollen die Spieler in der Lage sein, verschiedene Verteidigungsgebäude strategisch auf dem Spielfeld zu platzieren. Diese Gebäude haben die Aufgabe, automatisch angreifende feindliche Einheiten zu bekämpfen und zu zerstören. Jede zerstörte Einheit soll Ressourcen freigeben, die der Spieler zum Bau neuer Gebäude und zur Verbesserung bestehender Strukturen verwenden kann. Die Mechanik soll sicherstellen, dass das Spiel fortlaufend dynamisch bleibt und die Spieler durch Ressourcenmanagement und strategische Platzierung der Gebäude involviert sindWechselwirkungen

Ihr Text

### Wechselwirkung

Die Verteidigungsgebäude interagieren direkt mit den feindlichen Einheiten. Die Ressourcen, die von zerstörten Einheiten gewonnen werden, beeinflussen das Wirtschaftssystem des Spiels und ermöglichen es dem Spieler, weitere Verteidigungsstrukturen zu errichten oder bestehende zu verbessern. Zudem soll das Sichtfeld (Fog of War) durch das Zerstören feindlicher Einheiten schrittweise erweitert werden, was strategisch wichtig ist, um die Lage der gegnerischen Fabrik aufzudecken

### Risiken

Es besteht das Risiko, dass die Balance zwischen der Schwierigkeit der angreifenden Wellen und der Verfügbarkeit von Ressourcen für den Spieler nicht optimal abgestimmt ist, was zu einer zu leichten oder zu schweren Spielerfahrung führen könnte. Eine sorgfältige Ausbalancierung ist erforderlich, um sicherzustellen, dass das Spiel herausfordernd bleibt, ohne frustrierend zu sein.

## 2. Anforderung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | Td02 | **Nichttechnischer Titel** | | Zerstörung der gegnerischen Farbik | | |
| **Quellen** | Gameplay/Konzeptdokument | | **Verweise** | Sektion 4.3 End-Ziel des Spiels | **Priorität** | Hoch |

### Beschreibung

Das primäre Ziel des Spiels ist es, die gegnerische Fabrik zu zerstören, von der aus feindliche Einheiten generiert werden. Der Weg zur Fabrik soll durch ein Labyrinth von Wegen führen, wobei der Spieler den richtigen Pfad finden muss, um die Fabrik zu erreichen und zu vernichten. Das erfolgreiche Zerstören der Fabrik führt zum Gewinn des Levels.

### Wechselwirkung

Die Spieler müssen ihre Verteidigungsgebäude und Ressourcenmanagement taktisch nutzen, um die Wellen der feindlichen Einheiten zu überstehen und den Weg zur Fabrik freizumachen. Die Interaktion mit dem Fog of War ist hierbei kritisch, da die Sicht auf den Weg zur Fabrik von der Zerstörung feindlicher Einheiten abhängt.

### Risiken

Ein schlecht gestaltetes Level-Design könnte es entweder zu einfach machen, die Fabrik zu erreichen, oder es könnte Spieler unverhältnismäßig schwer machen, was zu einer ungleichmäßigen Spielerfahrung führen könnte.

## 3. Anforderung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | Td03 | **Nichttechnischer Titel** | | Spielsteuerung und Benutzerinterface | | |
| **Quellen** | Gameplay/Konzeptdokument | | **Verweise** | Sektion 4.4 Benutzerinteraktion | **Priorität** | Hoch |

### Beschreibung

Das Spiel soll ausschließlich mit der Maus steuerbar sein. Die Benutzeroberfläche muss intuitiv sein, damit der Spieler leicht Gebäude platzieren und verwalten sowie Spielinformationen einsehen kann. Alle Aktionen sollen mit einfacher Interaktion durchgeführt werden können.

### Wechselwirkung

Die Interaktion des Spielers mit der Benutzeroberfläche muss flüssig und fehlerfrei sein, was eine klare Rückmeldung über die durchgeführten Aktionen beinhaltet. Die Maussteuerung muss präzise und reaktionsfähig sein.

### Risiken Eine nicht intuitive Steuerung oder Benutzeroberfläche kann zu einer schlechten Nutzererfahrung führen und den Spieler davon abhalten, das Spiel zu genießen oder fortzusetzen.

## 4. Anforderung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | Td04 | **Nichttechnischer Titel** | | KI der gegnerischen Einheiten | | |
| **Quellen** | Gameplay/Konzeptdokument | | **Verweise** | Keine | **Priorität** | Hoch |

### Beschreibung

Die KI der gegnerischen Einheiten muss in der Lage sein, autonom zu agieren. Dies beinhaltet das Navigieren durch das Level, das Umgehen von Hindernissen und das Anstreben der Zerstörung der Spielerbasis. Die KI soll auch in der Lage sein, auf Veränderungen der Umgebung zu reagieren, wie zum Beispiel neu platzierte Verteidigungstürme.

### Wechselwirkung

Die KI soll mit dem Levelgenerator und dem Fog of War-System interagieren, um sicherzustellen, dass die Einheiten sinnvolle Pfade wählen und auf die Entdeckungen des Spielers reagieren. Die KI soll auch mit dem Ressourcensystem zusammenarbeiten, um Belohnungen für den Spieler bei der Zerstörung von Einheiten zu gewährleisten.

### Risiken Eine KI, die zu vorhersehbar oder zu einfach zu besiegen ist, könnte das Spielerlebnis negativ beeinflussen und den Wiederspielwert des Spiels verringern. Ebenfalls könnte eine zu komplexe KI zu hohen Entwicklungszeiten oder -kosten führen.

## 5. Anforderung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nr. / ID** | Td05 | **Nichttechnischer Titel** | | Level Generator | | |
| **Quellen** | Gameplay/Konzeptdokument | | **Verweise** | Keine | **Priorität** | Mittel |

### Beschreibung

Das Spiel soll über einen Mechanismus verfügen, der die Level zufällig generiert. Dieser soll eine Vielzahl von Level-Layouts erstellen können, die sowohl strategische Tiefe bieten als auch die Anforderungen des Fog of War berücksichtigen. Jedes Level soll einzigartige Herausforderungen und Pfade zur gegnerischen Fabrik bieten.

### Wechselwirkung

Der Levelgenerator muss mit dem Spielbalancing-System interagieren, um sicherzustellen, dass keine unspielbaren oder unfairen Szenarien generiert werden. Er muss auch die Platzeigenschaften für die Gebäude des Spielers und die Startpunkte der gegnerischen Einheiten adäquat integrieren.

### Risiken

Ein Levelgenerator, der repetitive oder langweilige Level erzeugt, könnte das Spielerlebnis negativ beeinflussen. Zudem besteht das Risiko, dass die zufällig generierten Level nicht immer eine faire oder angemessene Schwierigkeitskurve bieten.

# Technische anforderungen

## IntelliJ

Die IDE zur Entwicklung des Projektes mit Java.

## JDK

Das Java Development Kit ist voraussetzung um mit Java zu programmieren.

## GIT

Eine Versionskontrolle zum kollaborierten Entwickeln.

## JAVAFX

Ein Java-Framework.

## FXGL

Eine Bibliothek die extra zum entwickeln von Spielen in JavaF vorgesehen ist.

## Paint.net

Eine Bildbearbeitungssoftware zur erstellen eigener Grafiken für das Projekt.

# Genehmigung

Die Genehmigung erfolgt...

|  |  |
| --- | --- |
| Datum: |  |
| Unterschrift Auftraggeber: |  |
| Unterschrift Projektleiter: |  |
| Weitere Unterschriften: |  |

# Anhang

## Erstes Konzeptbild:



## UML-Klassendiagramm



## Lastenheft

