Flappy Hunter Extreme: The Return

Anforderungsanalyse

|  |  |
| --- | --- |
| **Autor(en):** | Nando Gmünder |
| **Version:** | 1.0 |
| **Erstellt am:** | 3. Dezember 2021 |
| **Letzte Änderung:** | 09.12.2021 |
| **Speicherort:** | Dokumente |

Inhalt

[1 Einleitung 3](#_Toc89250474)

[1.1 Systemidee 3](#_Toc89250475)

[Die wichtigsten Funktionen sind: 3](#_Toc89250476)

[1.2 Management Summary 3](#_Toc89250477)

[1.3 Projektteam 3](#_Toc89250478)

[1.4 Glossar 3](#_Toc89250479)

[2 Ausgangslage (IST), Problembereiche 4](#_Toc89250480)

[2.1 Beschreibung der Ausgangslage 4](#_Toc89250481)

[2.2 Problembereiche und Schwachstellen 4](#_Toc89250482)

[3 Ziele (SOLL) 5](#_Toc89250483)

[3.1 Beschreibung der Ziele 5](#_Toc89250484)

[3.2 Produktperspektive, Nutzen 5](#_Toc89250485)

[3.3 Zielkonflikte 5](#_Toc89250486)

[3.4 Abgrenzung 5](#_Toc89250487)

[4 Anforderungsanalyse 6](#_Toc89250488)

[4.1 Identifizierung der Akteure 6](#_Toc89250489)

[4.2 Anforderungskatalog 7](#_Toc89250490)

[A: Funktionale Anforderungen 7](#_Toc89250491)

[NF: Nichtfunktionale Anforderungen 8](#_Toc89250492)

[5 Systemablaufmodelle (Aktivitäten) 9](#_Toc89250493)

[5.1 Aktivität "Spiel spielen" 9](#_Toc89250494)

# Einleitung

## Systemidee

In diesem Spiel kann man eine Spielfigur durch eine Welt bewegen. Das Ziel ist es in einer bestimmten Zeit, möglichst viele Gegnerische Figuren zu «fressen», welche in der Welt herumfliegen oder laufen. Jedes Mal, wenn man einen Feind «gefressen» hat, wird die Spielfigur ein bisschen grösser. Nach der abgelaufenen Zeit wird das Spiel beendet und die Anzahl der «gefressenen» Feinden wird angezeigt.

## Die wichtigsten Funktionen sind:

* Funktion 1: Die Spielfigur soll nach links bewegt werden können.
* Funktion 2: Die Spielfigur soll nach rechts bewegt werden können.
* Funktion 3: Die Spielfigur soll hüpfen können
* Funktion 4: Es sollen feindliche Figuren automatisch und zufällig in der Spielwelt herumfliegen oder laufen.

**Management Summary**

In diesem Spiel kann man eine Spielfigur durch eine Welt bewegen. Das Ziel ist es in einer bestimmten Zeit, möglichst viele Gegnerische Figuren zu «fressen», welche in der Welt herumfliegen oder laufen. Jedes Mal, wenn man einen Feind «gefressen» hat, wird die Spielfigur ein bisschen grösser. Nach der abgelaufenen Zeit wird das Spiel beendet und die Anzahl der «gefressenen» Feinden wird angezeigt.

## Projektteam

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Kürzel** | **Funktion** | **Kontakt (Telefon, Mail...)** |
| Nando Gmünder | 001 | Projektleiter/Entwickler | gmuender.nando@gmail.com |

# Ausgangslage (IST)

In diesem Kapitel wird die Ausgangslage beschrieben und es werden die Problembereiche identifiziert, die sich in der heutigen Situation zeigen.

## Beschreibung der Ausgangslage

Die Ausgangslage ist es, ein Projekt innerhalb von 9 Tagen zu machen. Ich möchte ein kleines Spiel machen. Da ich nicht weiss, mit was und wie ich das Spiel machen möchte, muss ich zuerst wahrscheinlich ein bisschen recherchieren.

## Problembereiche und Schwachstellen

Ein Problem wird es sicher sein, die richtige Entwicklungsumgebung zu finden.

# Ziele (SOLL)

In diesem Kapitel werden die übergeordneten Ziele beschrieben, die mit dem zu entwickelnden System erreicht werden sollen.

## Beschreibung der Ziele

Das übergeordnete Ziel ist es, dass das Spiel läuft.

Weitere Ziele sind:

* Das die Spielfigur sich bewegen kann.
* Das feindliche Figuren herumfliegen und laufen.
* Das die Spielfigur die feindlichen Figuren fressen kann.

## Produktperspektive, Nutzen

Der Nutzen dieses Spieles ist es, Langeweile zu vertreiben.

# Anforderungsanalyse

Die Anforderungen an das zu entwickelnde System definieren alle zu erfüllenden Eigenschaften oder die zu erbringende Leistung, sowie allfällige technische Vorgaben und weitere Forderungen des Kunden im Zusammenhang mit dem (den) zu erstellendem Produkt(en).

## Identifizierung der Akteure

Ein Akteur bezeichnet eine Rolle, die mit dem System interagiert. Es werden 2 Akteure unterschieden:

* **Kunde:** Der Kunde kann nur das Spiel spielen.

## Kontextdiagramm



Spieler

Spiel

## Anforderungskatalog

Der Anforderungskatalog ist eine priorisierte Liste, die alles enthält, was im zu entwickelnden Produkt enthalten sein soll. Es wird unterschieden zwischen funktionalen und nicht-funktionalen Anforderungen.

Jede Anforderung ist in Alltagssprache, in der Form einer "User-Story", formuliert, mit einer ID zur späteren Identifizierung versehen sowie mittels 3 verschiedener Kategorien priorisiert: 1 = hohe Prorität, 2 = mittlere Priorität, 3 = keine Priorität. Diese drei Prioritäten repräsentieren die Verpflichtungen "must", "should" und "nice to have".

Im Verlauf der Zeit können neue Anforderungen hinzukommen und/oder bestehende Anforderungen können wegfallen. Anforderungen, die wegfallen, sind im Dokument zu belassen und als ~~gestrichen~~ zu markieren.

Hinter einer Anforderung kann ihn eckigen Klammern in der Form [Z#] eine Zusatzinformation hinterlegt werden und mit [F#] bzw. [F#, F#, …] können eine oder mehrere offene Fragen referenziert werden.

## A: Funktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| **ID** | **Anforderung** | **Prio** |
| A001 | Als Spieler möchte ich eine schön gestaltete Welt sehen. [Z1] | 1 |
| A002 | Als Spieler möchte ich die Spielfigur nach links bewegen können. | 1 |
| A003 | Als Spieler möchte ich die Spielfigur nach rechts bewegen können. | 1 |
| A004 | Als Spieler möchte ich das die Spielfigur hüpfen kann. | 1 |
| A005 | Als Spieler möchte ich auf laufende Gegner treffen können. | 1 |
| A006 | Als Spieler möchte ich auf fliegende Gegner treffen können. | 1 |
| A007 | Als Spieler möchte ich bei Kontakt mit den Gegnern, dass die Spielfigur die Gegner frisst. | 1 |
| A008 | Als Spieler möchte ich, dass die Spielfigur grösser wird, wenn ein Gegner gefressen wird. | 1 |
| A009 | Als Spieler möchte ich, dass die gefressenen Gegner nach kurzer Zeit wieder generiert werden. [Z2] | 2 |
| A010 | Als Spieler möchte ich, dass das Spiel nach einer bestimmten Zeit abgebrochen wird. [Z3] | 2 |
| A011 | Als Spieler möchte ich, dass nach Spielabbruch die Anzahl der gefressenen Gegner angezeigt wird. | 2 |
| A012 | Als Spieler möchte ich das Spiel mit einem Button neustarten. | 2 |
| A013 | Als Spieler möchte ich einen Account erstellen. [Z4] | 3 |
| A014 | Als Spieler möchte ich, dass die Anzahl der gefressenen Gegner gespeichert wird. | 3 |

## NF: Nichtfunktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| **ID** | **Anforderung** | **Prio** |
| NF001 | Der Code soll übersichtlich sein. | 1 |
| NF002 | Beim Erstellen eines Accounts sollen die Account-Daten geprüft werden. | 1 |

[Z1] = Die Welt soll Bäume, Steine, Höhen und Tiefen enthalten.

[Z2] = Die Gegner sollen nach 3 Sekunden wieder generiert werden.

[Z3] = Das Spiel soll nach 3 Minuten abgebrochen werden.

[Z4] = Der Account wird benötigt, um den Highscore zu speichern.

# Systemablaufmodelle (Aktivitäten)

Dieses Kapitel zeigt die wichtigsten und/oder komplexesten funktionalen Anforderungen in ihrem Ablauf. Dazu werden die einzelnen Aktivitätsschritte detailliert analysiert. Für die Darstellung der einzelnen Aktivitätsschritte werden Aktivitätsdiagramme nach UML verwendet.

## Aktivität "Spiel spielen (Grob Durchlauf)"

