Game Design Document: Giranimo

Inhaltsverzeichnis

[1. Kurzfassung 4](#_Toc25770455)

[2. Spielkonzept 4](#_Toc25770456)

[2.1 Story 4](#_Toc25770457)

[2.2 Spielablauf/Level 4](#_Toc25770458)

[2.3 Game Over 4](#_Toc25770459)

[2.4 Mechanik 4](#_Toc25770460)

[2.5 Menü/Szenen 4](#_Toc25770461)

[2.6 Logik 5](#_Toc25770466)

[3. Artdesign & Musik 5](#_Toc25770470)

[3.1 Charakter 5](#_Toc25770471)

[3.2 Collectibles 5](#_Toc25770472)

[3.3 Cut Scenes 6](#_Toc25770473)

[3.3.1 Start 6](#_Toc25770474)

[3.3.2 Ende (satte, dicke Giraffe) 6](#_Toc25770475)

[3.4 Sound 6](#_Toc25770476)

[3.4.1 Startgeräusch 6](#_Toc25770477)

[3.4.2 Essgeräusch 6](#_Toc25770478)

[3.4.3 Verloren 6](#_Toc25770479)

[3.4.4 Gewonnen 6](#_Toc25770480)

[3.4.5 Ebenenübergang 6](#_Toc25770481)

[3.5 Musik 6](#_Toc25770482)

[3.5.1 Erste Ebene 6](#_Toc25770483)

[3.5.2 … 6](#_Toc25770484)

[4. Technische Details 6](#_Toc25770485)

[4.1 Programmiersprache (Naming Conventions) 6](#_Toc25770486)

[4.2 Programmierhardware & -software 6](#_Toc25770487)

[4.2.1 Game Engine 6](#_Toc25770488)

[4.2.2 Audiotool 6](#_Toc25770489)

[4.2.3 Snippingtool/Traingertool 6](#_Toc25770490)

[4.2.4 Plattformen/ Zielgeräte 6](#_Toc25770491)

[5. Spielerbasierend 7](#_Toc25770492)

[5.1 Target Audience 7](#_Toc25770493)

[5.2 Spielgefühl (Psyche) 7](#_Toc25770494)

[6. Management 7](#_Toc25770495)

[6.1 Detailed Scheduele 7](#_Toc25770496)

[6.2 Budget (extra Plan) ! 7](#_Toc25770497)

[6.3 Lizenzen ! 7](#_Toc25770498)

[6.4 Riskanalysis ! 7](#_Toc25770499)

[6.5 Testplan 7](#_Toc25770500)

TODO!

2.7 Mathematische Berechnungen James

3. Art & Design Marty + (Artist = alle)

3.3 Cut Scenes Markus

3.4 Sound Markus

3.5 Musik Markus

4. Technische Details James

4.2.2 Audiotool Markus

4.2.3 Snippingtool/Traingertool

4.3 Plattformen/ Zielgeräte James

6. Management Alle

# **1. Kurzfassung**

Ein Rogue-Like/Endless-Runner Mobile Game für unterwegs, bei dem der Spieler eine stetig wachsende Giraffe manövriert und versucht so hoch zu kommen wie möglich.

# **2. Spielkonzept**

## 2.1 Story

Eine kleine Giraffe erfreut sich an kühlem Wasser in einer lauen Sommernacht. Als sie in den klaren Himmel aufblickt und die funkelnden Sterne sieht, fragt sie sich, wie wohle die Sterne und der Mond schmecken würden. Sie streckt ihren Hals und beginnt zu essen was ihr in den Weg kommt, um eines Tages die Sterne zu erreichen.

## 2.2 Spielablauf

Bei jedem Spiellauf fängt die Giraffe vom Boden den Hals zu strecken. Der Spieler steuert die horizontale Bewegung des Giraffenkopfs und versucht dabei Nahrung aufzunehmen so wie Hindernissen auszuweichen. Der Wachstum des Halses hängt von der Nahrungsaufnahme ab.

(Skizze)

## 2.3 Game Over

Im Sinne des Endless-Runner Prinzip gibt es keine Winning-condition in diesem Spiel.

Die Bedingungen für das Game-Over auslösen sind:

1. Giraffe stößt gegen Hindernis
2. Giraffe gerät an den Bildschirmrand (oben, unten, links, rechts)

## 2.4 Mechanik

Die horizontale Bewegung der Giraffe wird kontrolliert durch Touch-Input auf den beiden Bildschirmhälften. Die Besonderheit in Giranimo ist, dass die Bildschirmkamera unabhängig des Giraffenwachstums steigt. Somit muss der Spieler eine passende Nahrungsaufnahme managen um nicht aus dem Screen zu fallen.

## 2.5 Szenen/Menü

2.5.1 Startszene: Beim Start des Spiels „Berühre den Bildschirm um zu starten“

2.5.2 Charaktermenü: (Skizze)

Zeigt den Charakter mit seinem aktuellen Aussehen und seinen Statistiken: Wächst x schnell, Anzeige aller Währungen (Gegessene Nahrung). Mit dem Klick auf den Charakter kommt der Spieler auf ein Menü mit Charakterskins.

2.5.3 Optionen: (Skizze)

Hier gibt es mehrere Regler zum Einstellen von Musik und Soundeffekte, sowie die allgemeine Spielerstatistik: Spielzeit, insgesamte Währung, Anzahl der Run, Rekord

### 2.5.4 Weitere Ideen: Integration von Online-System z.B. Game-Center (IOS-Systeme) mit Global Highscore und Freundesliste

## 2.6 Logik/Implementierung

### 2.6.1 Kamerabewegung: Eine konstante *k\_fallSpeed* bestimmt die generelle Fallgeschwindigkeit der Consumables und des Bildschirmhintergrunds -> Simulation von steigender Kamera. Beim Spawnen der Consumables wird ein zufälliger Wert [.1f, 3f] an der FallSpeed multipliziert um so Monotonie zu verhindern.

### 2.6.2 Wachstum: Ein konstanter 2DVector.down *k\_growthVel* am Player Objekt sorgt dafür, dass sie langsam nach unten fällt -> Erweckt den Anschein, dass der Giraffenhals langsamer als die Kamera wächst. Jede Unterklasse von Consumables hat einen *m\_pushFactor* welches beim Konsumieren als Positive Force nach oben auf den Spieler angewandt wird -> Schubeffekt.

### 2.6.3 Preise & Upgrades: Die „gegessenen“ Objekte pro Run werden abgespeichert und zählen dann als Spielwährung. Unterschiedliche Upgrades/Skins erfordern eine bestimmte Anzahl von verschiedenen Nahrungsstücken, die gezahlt werden müssen.

# 3. Artdesign & Musik

## 3.1 Charakter

|  |  |
| --- | --- |
| Charakter | Bild |
| Standard |  |
| Alien |  |
| Party |  |
| Bauarbeiter |  |
| Astronaut |  |
| Drache |  |
| Verrückte Farben |  |
| Gothicstyle |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 3.2 Collectibles

|  |  |
| --- | --- |
| Währung | Bild |
| Laubblatt (Birke) |  |
| Laubblatt (Ahorn) |  |
| Hütte |  |
| Haus |  |
| Wolkenkratzer |  |
| Wolke |  |
| Flugzeug |  |
| Satellit |  |
| Mond |  |
| Planet |  |
| Sonne |  |
| Sonnensystem |  |
| Glaxie |  |
|  |  |
|  |  |
|  |  |

## 3.3 Cut Scenes

### 3.3.1 Start

### 3.3.2 Ende (satte, dicke Giraffe)

## 3.4 Sound

### 3.4.1 Startgeräusch

### 3.4.2 Essgeräusch

### 3.4.3 Verloren

### 3.4.4 Gewonnen

### 3.4.5 Ebenenübergang

## 3.5 Musik

### 3.5.1 Erste Ebene

### 3.5.2 …

# 4. Technische Details

## 4.1 Programmiersprache (Naming Conventions)

Das Spiel wird mit Unity gemacht und die Skripte in C# programmiert. Naming Convention sind im GitHub Ordner unter *naming\_conventions.txt* zu finden.

## 4.2 Programmierhardware & -software

### 4.2.1 Game Engine

Unity 2019.2.12f1

Visual Studio 2019

### 4.2.2 Audiotool

### 4.2.3 Snippingtool/Traingertool

### 4.2.4 Plattformen/ Zielgeräte

Vorerst für mobile Endgeräte: Android, IOS

TODO: welche Versionen!

Weiter dann für PC über Steam (Windows), Nintendo Switch

# 5. Spielerbasierend

## 5.1 Target Audience

Kinder und Jugendliche und Gelegenheitsspieler. Für Menschen, welche nicht viel spielen, aber z.B. in öffentlichen Verkehrsmitteln fahren und das Spiel nebenbei spielen.

## 5.2 Spielgefühl (Psyche)

# 6. Management

## 6.1 Detailed Scheduele

## 6.2 Budget (extra Plan) !

## 6.3 Lizenzen !

## 6.4 Riskanalysis !

Copyrightprobleme?

Konkurrierende Produkte: die meisten Idle-Games

Abgrenzung von den anderen Spielen: TODO

## 6.5 Testplan

Für jede Plattform ein Plan bzw. die selben Punkte auf allen Plattformen und Versionen testen!

TODO