Game Design Document: Giranimo

TODOS!

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

3.3 Cut Scenes Markus

3.4 Sound Markus

3.5 Musik Markus

4.1.2 Audiotool Markus

4.1.3 Snippingtool/Traingertool Markus

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 namens Giranimo erfreut sich am kühlem Wasser in einer lauen Sommernacht. Als sie in den klaren Himmel aufblickt und die funkelnden Sterne sieht, fragt sie sich, wie wohl die Sterne und der Mond schmecken würden. Aus Neugier streckt sie ihren Hals und verzehrt dabei einen Apfel der an einem Baum hängt. Überraschenderweise wächst ihr Hals mit einem Schub in die Höhe. Ob es denn möglich ist, dass Giranimo bis zu den Sternen wächst?

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:

* Giraffe stößt gegen Hindernis
* 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 Charaktere

|  |  |
| --- | --- |
| 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 |  |
| Galaxie |  |

3.3 Cut Scenes

3.3.1 Start

Giraffe schaut in den Himmel, ihr Magen knurrt und das Spiel beginnt.

3.3.2 Ende

GameOver: Giraffen Kopf mit heraus hängender Zunge. Sternchen kreisen um ihren Kopf.

3.4 Sound

3.4.1 Startgeräusch

Mindestens 3 hohe aufsteigende Piano Noten schnell hintereinander gespielt.

3.4.2 Essgeräusch

Mit Microfon eingesprochenes "chump".

3.4.3 Verloren

Kurze traurige Melodie , dann Regenplätschern.

3.4.4 Gewonnen (neue Ebene freigeschaltet)

Trompete spielt

3.4.5 Ebenenübergang

TODO: entfernen, dass spiel kann nicht "gewonnen werden", der einzige "Sieg" ist ein Ebenenfreischalten

3.5 Musik

3.5.1 Erste Ebene

"Oldschool" Game Music / Smooth Jazz Piano / LoFi - HipHop

3.5.2 Zweite Ebene und Folgende

Musik abhängig von der Thematik der Ebene. (Bsp. "Funky Space Musik" in der Weltall Ebene"

4. Technische Details

4.1 Hardware & Software

4.1.1 Programming: Die Gameengine ist Unity Version 2019.2.12f1. Skripte sind in C# geschrieben. Naming Conventions sind im GitHub Ordner unter *naming\_conventions.txt* zu finden.

4.1.2 Audiotool:

Studio One

4.1.3 Snippingtool/Traingertool:

Magix Video Pro X5

4.2 Plattformen/ Zielgeräte

Das Spiel wird auf AppStore und Google Play gepublished. Systemrequirements sind IOS-Version 11 und Android 9 Pie oder darüber.

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