

Game Development 2

Technical Design Document

Among Teddy

-

Sozial Deduction Game

Team : Sozial Deduction Game
Feature : Sprachsteuerung

Kai Frieze
Jasmin Knott
Bryan Fichtel
Daniel Sun
Dennis Tombers

Inhaltsverzeichnis

| | |
|--|----------|
| Abschnitt 1: Liste der Funktionen | 2 |
| Abschnitt 2: Entwicklungsumgebung | 2 |
| Abschnitt 3: Zeitplan | 3 |
| Abschnitt 4: Diagramme zur Illustration des Software Designs | 3 |
| Abschnitt 5: Art Tools | 4 |
| Abschnitt 6: Objekte, Terrain und Szenen Management | 6 |
| Abschnitt 7: Kollisionsdetektion, Physik & Interaktion | 7 |
| Abschnitt 8: Spiellogik & AI | 7 |
| Abschnitt 9: Networking | 7 |
| Abschnitt 10: Audioeffekte und visuelle Effekt | 7 |
| Abschnitt 11: Zielplattform & Hardware/Software Voraussetzungen | 8 |
| Abschnitt 12: Graphical User Interface (GUI) | 8 |

Abschnitt 1: Liste der Funktionen

- Top-Down Ansicht
- Spielwelt:
 - Räume
 - Versammlungsraum
 - Sabotage Reparatur Orte
 - Charaktere
 - Aufgaben
- Windows
- Hintergrundgeschichte
- Multiplayer
- Minispiele
 - Flappy Bird
 - Simon Says
 - Farben bestimmen
 - Gegenstände bestimmen
 - Puzzle
 - 2 Player Code
 - Schieberätsel
 - Schiffe versenken
 - Snake
- Sabotage
 - Türen verschließen
 - Lichter ausschalten
 - Kommunikation blockieren
 - GPS deaktivieren
- Menü
- Spieleinstellungen
- Audio und Sound Effekte

Abschnitt 2: Entwicklungsumgebung

Als Entwicklungsumgebung verwenden wir für die Programmierung Unity, da alle benötigten Spielelemente darin entwickelt werden können und bereits viele Funktionen zur Verfügung stehen, auf welchen aufgebaut werden kann. Dies beinhaltet vor allem Netzwerkfunktionen und Spracheingaben.

Zum Austausch von Daten, Grafiken, Animation und Code verwenden wir GitHub, ein Tool mit welchem man Online als Team seine Dateien austauschen und synchronisieren kann.

Abschnitt 3: Zeitplan

Zeitplan für Programmierer

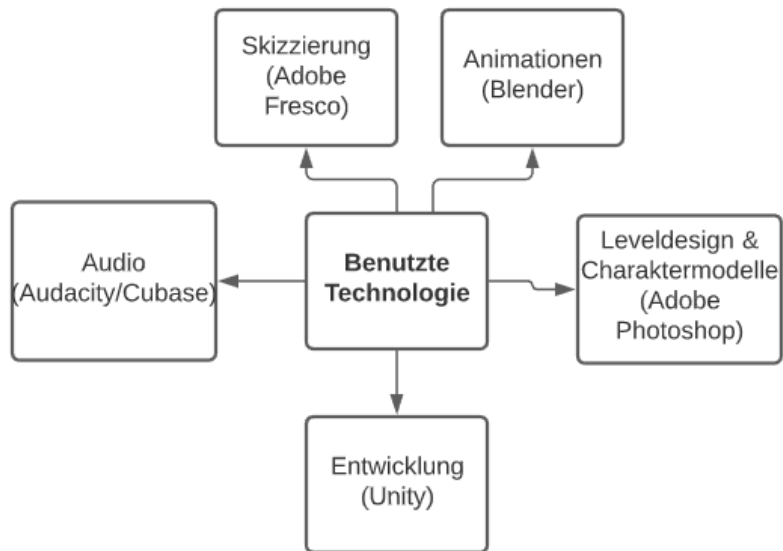
| Aufgaben | Woche 1 02.11.2020 | Woche 2 09.11.2020 | Woche 3 16.11.2020 | Woche 4 23.11.2020 | Woche 5 30.11.2020 | Woche 6 07.12.2020 | Woche 7 14.12.2020 | Woche 8 21.12.2020 | Woche 9 28.12.2020 | Woche 10 04.01.2020 | Woche 11 11.01.2020 | Woche 12 18.01.2020 | Woche 13 25.01.2020 |
|-----------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|------------------------|------------------------|------------------------|
| | | | | | | | | | | | | | |
| Game Design Dokument | | | | | | | | | | | | | |
| Technisches Design Dokument | | | | | | | | | | | | | |
| Menü erstellen | | | | | | | | | | | | | |
| Level einbinden | | | | | | | | | | | | | |
| Charaktere einbinden | | | | | | | | | | | | | |
| Objekte einbinden | | | | | | | | | | | | | |
| Trigger & Events einbinden | | | | | | | | | | | | | |
| Animationen einbinden | | | | | | | | | | | | | |
| Sprachsteuerung | | | | | | | | | | | | | |
| Servererstellung | | | | | | | | | | | | | |
| Anleitung erstellen | | | | | | | | | | | | | |
| Walkthrough-Video erstellen | | | | | | | | | | | | | |

Zeitplan für Grafiker/Designer

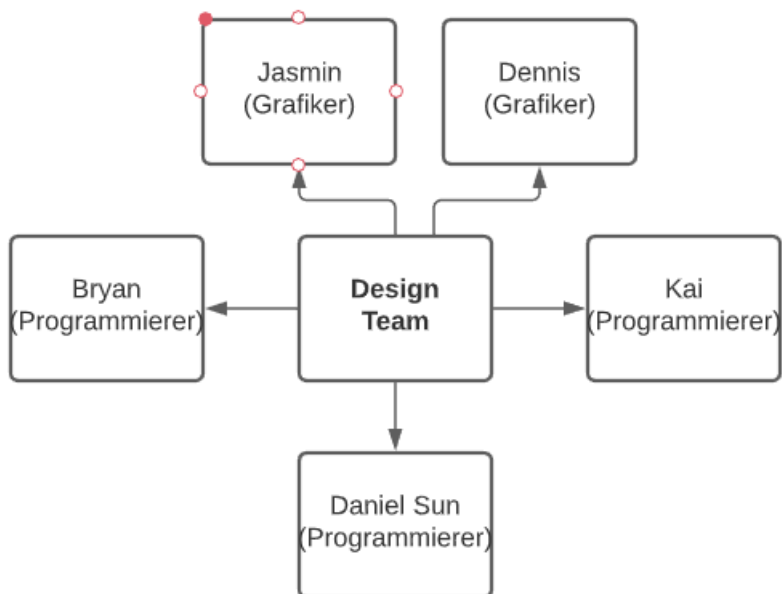
| Aufgaben | Woche 1 02.11.2020 | Woche 2 09.11.2020 | Woche 3 16.11.2020 | Woche 4 23.11.2020 | Woche 5 30.11.2020 | Woche 6 07.12.2020 | Woche 7 14.12.2020 | Woche 8 21.12.2020 | Woche 9 28.12.2020 | Woche 10 04.01.2020 | Woche 11 11.01.2020 | Woche 12 18.01.2020 | Woche 13 25.01.2020 |
|------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|------------------------|------------------------|------------------------|
| | | | | | | | | | | | | | |
| Game Design Dokument | | | | | | | | | | | | | |
| Technisches Design Dokument | | | | | | | | | | | | | |
| Entwurf des Menüdesigns | | | | | | | | | | | | | |
| Entwurf des Leveldesigns | | | | | | | | | | | | | |
| Entwurf des Charakterdesigns | | | | | | | | | | | | | |
| Entwurf des Objektdesigns | | | | | | | | | | | | | |
| Entwurf der Animationen | | | | | | | | | | | | | |
| Anleitung erstellen | | | | | | | | | | | | | |
| Walkthrough-Video erstellen | | | | | | | | | | | | | |

Abschnitt 4: Diagramme zur Illustration des Software Designs

Technologie-Diagramm



Design-Diagramm



Implementationsdiagramm

Gameplay-Diagramm

Abschnitt 5: Art Tools

Adobe Photoshop

- Nutzen wir für
 - 2D charakter erstellung
 - Map Design
 - Minispiele Design
 - Spritesheet erstellen
- Version
 - Photoshop (22.0.1)
- Welche möglichkeiten haben wir
 - Pixelbasierte Bildbearbeitungs software
 - Erstellung und Manipulation von Grafiken
 - Erstellung Transparenter Grafiken
 - 2D- und 3D-Bildbearbeitung und -Compositing

Blender

- Nutzen wir für
 - Visuelle Effekte
 - 3D Modelle
 - 2D animationen
- Version
 - Blender (2.91.0)
- Welche möglichkeiten haben wir
 - 3D-Modelle erstellen und bearbeiten
 - 3D Animationen Erstellen
 - Bilder Rendern
 - Grafiken erstellen
 - Texturierung von Objekten

Audacity

- Nutzen wir für
 - Sound Design
 - Sound Effekte
 -
- Version
 - Audacity (2.4.2)
-
- Welche möglichkeiten haben wir
 - Audio Editor
 - Audio Mixer
 - Audio Aufnahmen
 - Sounddateien Erstellen

Adobe Fresco

- Nutzen wir für
 - 2D charakter Skizzierung
 - Map Design Skizzierung
 - Minispiele Design Skizzierung
- Version
 - Fresco (2.0.1)
- Welche möglichkeiten haben wir
 - Zeichentool mit Stifteingabe
 - Pixel- und Vektor-Pinsel
 - Verschiedene Pinsel zum Skizzieren
 - 2D Grafiken erstellen

Abschnitt 6: Objekte, Terrain und Szenen Management

Objekte

- Spieler
 - Teddybär
 - Einhorn
 - Dinosaurier
 - Hund
 - Hase
 - Kanguru
 - ...
 - ...

Terrain

- Karten
 - Kinderzimmer
 - Schule
 - Labor
 - Küche
 - Flur
 - Garage
 - Wohnzimmer
 - Treppenhaus
 - Hauswirtschaftsraum

- ...
- ...
- ...

Szenen

- Intro : Kurze Einführung in das Spiel, erzählt die Anfangsgeschichte, dient ebenfalls als preload screen.
- Menü : Hier werden die grundsätzlichen Einstellungen wie Auflösung, Audio und Tastaturbelegung festgelegt. Hier kann man auch das Spiel starten und das Spiel beenden
- Lobby : Wenn ein Spiel gestartet wird befindet man sich Anfangs in der start Lobby hier bekommt man eine IP-Adresse mitgeteilt womit die anderen Spieler dem Server beitreten können. Ebenfalls werden hier Einstellungen vorgenommen wie die Spieleranzahl, Anzahl der Infizierten und Anzahl der Aufgaben. Die Spieler können sich hier ein Charakter aussuchen
- Minigames
 - FlappyBird
 - Simon says
 - Farben bestimmen
 - Gegenstände bestimmen
 - Puzzle
 - 2 Player Code
 - Schieberätsel
 - Schiffe versenken
 - Snake
- Ending Screen : Der Gewinner wird bekannt gegeben
- Credits : Die Spielersteller, Musik Komponisten und die Assets werden erwähnt.

Abschnitt 7: Kollisionsdetektion, Physik & Interaktion

Die Kollisionsdetektion und Physik der Spielcharaktere und Objekte wird mittels der Rigidbody- und Collider-Funktionen von Unity dargestellt.

Die Spieler können dabei mittels Maus und Tastatureingabe mit der Spielwelt interagieren. Wodurch zum Beispiel Minispiele, oder Abstimmungen aufgerufen werden. Diese können dann in Kombination von Maus- und Spracheingabe gesteuert werden.

Abschnitt 8: Spiellogik & AI

Künstliche Intelligenz wird ausschließlich in Minispielen benötigt und beschränkt sich auf simple Zufallsalgorithmen zur Verteilung der Aufgaben auf die Spieler, sowie zur Generierung von Elementen, Reihenfolgen und Mustern innerhalb der Minispiele. Dazu

kommt je nach Aufgabe eine eventuell erweiterte Logik, zum Beispiel beim Schiffe versenken, um nach einem Treffer weiter auf dieselbe Region zu feuern.

Abschnitt 9: Networking

Ein Spieler öffnet einen lokalen Serverraum und ist damit Host und Client, die restlichen Spieler können diesem Serverraum als Client beitreten. Die maximale Anzahl der Spieler kann dabei vom Host beschränkt werden.

Abschnitt 10: Audioeffekte und visuelle Effekt

Audio Effekte

Für die Audiodateien nutzen wir lizenzfreie Soundbibliotheken. Zudem nutzen wir für die Erstellung von individuellen Sounds Audacity.

Visuelle Effekte

Zur Erstellung von visuellen Effekten nutzen wir Blender. Mit dem Partikelsystem lassen sich einfach 2D effekte erstellen. Ebenfalls nutzen wir Photoshop um die Spritesheets zu erstellen.

Abschnitt 11: Zielplattform & Hardware/Software Voraussetzungen

Zielplattform:

- Windows Betriebssystem (vorzugsweise 8 oder höher)

Hardware Voraussetzungen:

1. Für den PC:

- Maus
- Tastatur
- Monitor
- Headset bzw. Lautsprecher und Mikrofon

2. Für den Laptop:

- Maus (oder Verwendung des integrierten Touchpads)
- Headset (oder Verwendung der integrierten Lautsprecher und des Mikrofon)

3. Allgemeine Voraussetzungen:

- Es wird eine stabile Internet Konnektivität gefordert

- Zeitgemäße PC Hardware

Software Voraussetzungen:

- Windows 8 oder höher

Abschnitt 12: Graphical User Interface (GUI)

Abschnitt 13: Spielmechaniken

Der Infizierte kann folgende Aktionen ausführen:

- Anstecken : Hierbei muss der Spieler fest in sein Mikrofon husten. Desto lauter das Husten, desto größer ist der Infektionsradius.
-

Die nicht Infizierten können folgenden Aktionen ausführen:

-

Abschnitt 14: Umsetzung der Minispiele

Alle Minigames im Überblick

1. FlappyBird
2. Simon says
3. Farben bestimmen
4. Gegenstände bestimmen
5. Puzzle
6. 2 Player Code
7. Schieberätsel
8. Schiffe versenken
9. Snake

FlappyBird :

Simon says :

Farben bestimmen :

Gegenstände bestimmen :

Puzzle :

2 Player Code :

Schieberätsel :

Schiffe versenken :
Snake :

Abschnitt 15: Musik

Wir werden für unser Spiel auf kostenlose und lizenzfreie Musik Bibliotheken zurückgreifen. Ebenfalls nutzen wir eigens komponierte Musikstücke.

Abschnitt 16: Namenskonventionen

Die

Abschnitt 17: Datenstruktur

Im Unity-Projekt haben wir verschiedene selbst erstellte Strukturordner

Resources

Resources/Objects

Resources/Player

Resources/

Scripts

Scripts/PlayerController

Scripts/NetzwerkManager

Scene

Scene/Intro

Scene/Start

Abschnitt 18: Versionsverwaltung GitHub

Zur Versionsverwaltung nutzen wir GitHub und das GitHub Asset für Unity

Bei jedem neuen Implementierungsschritt erschaffen wir einen neuen Branch. Bei fertiger Entwicklung mergen wir diesen mit unserem Developer Branch.

Wenn das Spiel am Ende fertig ist, mergen wir den Developer Branch mit dem Master Branch und erstellen ein fertiges, ausführbares Spiel.

Abschnitt 19: Probleme / Konflikte

Aufgrund der beschränkten Zeit schaffen wir es vielleicht nicht alle Minispiele zu implementieren. Ebenfalls müssen wir uns noch weiter mit der serverseitigen Erstellung eines Multiplayer Spiels auseinandersetzen.

Abschnitt 20: Referenzen