

Fachprojekt - Digital Entertainment Technologies

WiSe 2021/2022
TU Dortmund

E. Almsouti, C. Kolbe, L. Labbert, A. Montag

01.02.2022

TuttiFrutti - Spielidee

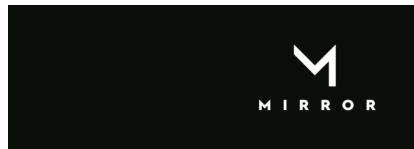
- Multiplayer Spiel
- Einfach und schnelle Runden mit unterschiedlichen Leveln/Modi
- Vor allem für Freunde-Gruppen gedacht als "party game"
- Free-for-all

TuttiFrutti - Infos

- Multiplayer via IP oder Steam (keine eigene Game-SteamID)
- 3rd Person
- Keine hohen Performance-Anforderungen
- Ein Spiel besteht aus 4 Runden zwischen 90-150s
- Kein finales Ausscheiden aus einer Lobby möglich
- Punkte werden auf die Spieler-Anzahl normiert

TuttiFrutti - Schwerpunkte

- Multiplayer
 - IP-basiert über kcp
 - Online über Steam



Mirror Networking

Steam Matchmaking & Lobbies

Steamworks Documentation > Features > Multiplayer > Steam Matchmaking & Lobbies

Overview

Steam's peer-to-peer matchmaking is built around the concept of a lobby. A lobby is an entity that lives on the Steam back-end servers that is a lot like a chat room. Users can create a new lobby; associate data with a lobby; search for lobbies based on that data; join lobbies; and share information with other users in the lobby. A single lobby can have up to 250 users in it, although typically most games have at most 2-16 players. Skill-based matchmaking is built on top of this system.

The Steam peer-to-peer matchmaking API is a set of functions that enable users to find other users to play a game with. The matchmaking functions all live in [SteamMatchmaking](#), which contains more details on the parameters for each function. Lobbies are uniquely identified by Steam ID, like users or game servers. The Steamworks Example has a full working implementation of lobbies.

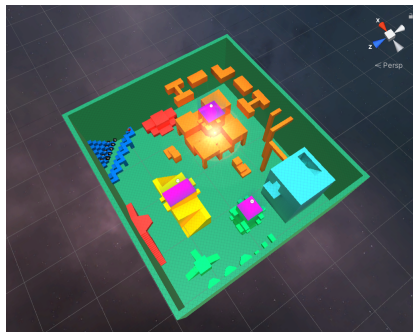
TuttiFrutti - Schwerpunkte 2

- Presentation
 - Eigener Player character
 - Player-Animations
 - Particle effects
 - Animiertes Menu
 - Player Konfiguration



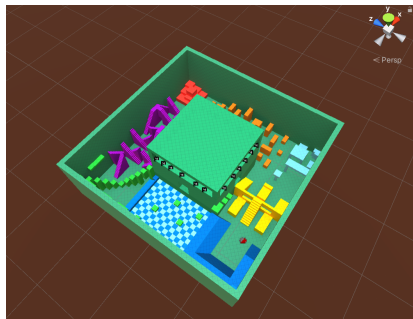
Level - HillKing

- Aktive Plattform wechselt alle 30" zufällig
- Wenn man auf der Plattform steht bekommt man Punkte -> andere Spieler durch Schubsen an Punkten hindern

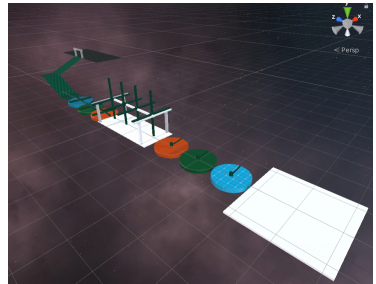


Level - Crown

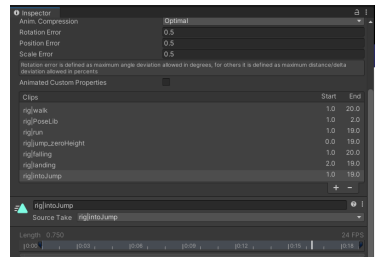
- Abhängig von der Spieler-Anzahl werden einzelne Spieler mit Kronen gespaunt
- Man bekommt Punkte wenn man eine Krone hat -> Kronen stehlen



Level - RunTheLine

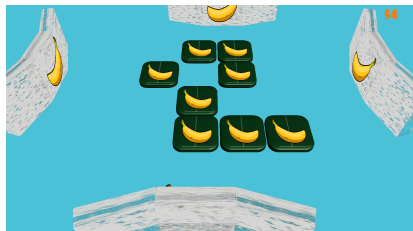


- Rennen zum Ziel!
- Der Spieler muss mehreren Hindernissen ausweichen und als erstes ins Ziel kommen.
- Die Punkte werden abhängig



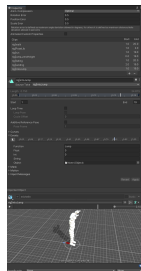
Level - PerfectMatch

- Halte alle drei Runden durch!
- Spielfeld: 16 Plattformen, die zu einem 4x4 Raster aufgebaut sind.
- Der Spieler muss sich, die auf den Plattformen angezeigten, Früchte merken und sich auf die richtige Plattform stellen wenn die Eliminierung stattfindet.
- Punkte werden anhand der durchgehaltenen Runden verteilt.



Sounds

- Hintergrund Themen für einzelne Level und das Hauptmenü über die Unity AudioSource und den Audiolistener.
- Level abhängige Sounds z.B.:
 - RunTheLine: wenn Spieler das Ziel durchquert
 - Crown: Kronenwechsel
 - HillKing: Plattformwechsel
 - PerfectMatch: wenn ein Spieler von der Karte fällt
- Playersounds werden über Events ausgelöst die mit den Animationen des Playercharacters verknüpft sind.



```
0 public class PlayerSound : MonoBehaviour //NetworkBehaviour
1 {
2     [SerializeField]
3     private AudioClip[] clips;
4
5     private AudioSource audioSource;
6
7     private void Awake(){
8         audioSource = GetComponent<AudioSource>();
9     }
10
11     private void Jump(){
12         audioSource.PlayOneShot(clips[0]);
13     }
14
15     private void RunningOne(){
16         audioSource.PlayOneShot(clips[1]);
17     }
18     private void RunningTwo(){
19         audioSource.PlayOneShot(clips[2]);
20     }
21 }
22
23
24
25
26
27
28
29
30
31
32
33
34
35
```

Probleme

- Multiplayer
 - Auch mit Framework aufwendig bis alle Details verstanden sind
 - Scripte an Synchronisation anpassen macht wenig Spaß
 - Testen schwierig/aufwendig, da Fehler oftmals erst nach Host/Client Wechsel, Rundenwechsel, etc. auftauchen
- Probuilder
 - (Unity-)Koordinaten
 - Snapping local/global
- Git
 - Zeitaufwendig merge-conflicts auf Grund von Parameter Änderungen auf Objekten durchzugehen
- C-Sharp : keine HashMaps, etc. (nur key und nicht value sorting)
- Blender zusammen mit Unity
- Unity selbst!

TuttiFrutti - Planung und Umsetzung

KW	45 08. Nov	46 15. Nov	47 22. Nov	48 29. Nov	49 06. Dez	50 13. Dez	51 20. Dez	52 27. Dez	1 03. Jan	2 10. Jan	3 17. Jan	4 24. Jan	5 31. Jan
Alex													
Colin													
Emad													
Louis													
Abgabe													
	Idee-Findung	Idee-Ausarbeitung		Planung updaten			Zw.-Präsi		Noch offen		Final Product	Fixes	
Team Fokus													
Presentation, GUI, Levelbuilding		Character Prototyping, Controller											
		Placeholder components											
Team Fokus													
Server/Client		Server/Client											
Game logic				Game UI		Game logic							
						Alle/nicht festgelegt							

- Tiefere Integration von (Steam)Multiplayer Komponenten
 - (Voice-)Chat
 - (Steam-)Achievements
- Seasons
 - Level + Modi Updates
 - Where are the coins?
- Globales player-ranking
- Weitere Player-Moves, wie Dinge Werfen

Let's play TuttiFrutti!