



Universität  
Basel

# **Meilenstein 1 - Gruppe 6**

## **Hurricane Entertainment**

### **CS108 Programmierprojekt, FS17**

**Benjamin Park**

**Cedrik Schimschar**

**Gowthaman Gobalasingam**

**Michel Borer**



HE

# Spielbeschreibung: Brändi Dog

- *Brettspiel, in der Regel 2 Gruppen à 2 Personen*
- *Ziel: seine eigene Murmeln aus dem Zwinger vom Start ins Ziel bringen*
- *Während des Spiels vom Partner unterstützt und vom Gegner „heimgeschickt“*
- *Murmeln werden bewegt durch das Ausspielen der Karten*



# Spielregeln

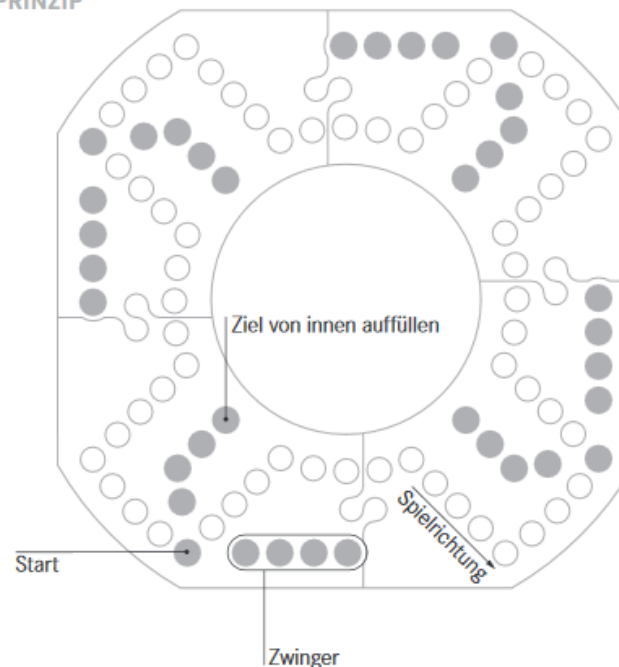
- *Partner setzen sich gegenüber*
- *Jeder hat 4 Murmeln im Zwinger*
- *Eine Person/Server ist der „Verteiler“*
- *Kartenset 2 x 55 Karten im Gegenuhrzeigersinn verteilt*
- *In jeder Runde tauschen die Partner je eine Karte verdeckt*
- *Spieler deckt eine Karte auf und macht dementsprechend den Zug*
- *Runde zu Ende, wenn alle Spieler alle Karten aufgedeckt haben*
- *Spiel zu Ende, wenn der Spieler ihre Murmeln ins Ziel gebracht hat*



# Züge & Karten

- *Start: ASS, KÖNIG, JOKER*
- *ASS = 1 oder 11 Pkt., KÖNIG = 13 Pkt. DAME = 23 Pkt.*
- *BUBEN: eigene Murmel mit einer vom Gegner oder vom Partner austauschen*
- *VIER: vor- oder rückwärts spielen*
- *SIEBEN: kann auf eigene Murmeln aufgeteilt werden*
- *Zwei Murmeln aufs gleiche Feld: 1. Murmel „heimgeschickt“*
- *Mit einer ganzen oder aufgeteilten Sieben überholt und „heimgeschickt“*
- *Überholen der Murmeln erlaubt, Ausnahme: Murmel neu auf Start*
- *Jede Karte muss gespielt werden*
- *Ziel von innen nach aussen gefüllt*

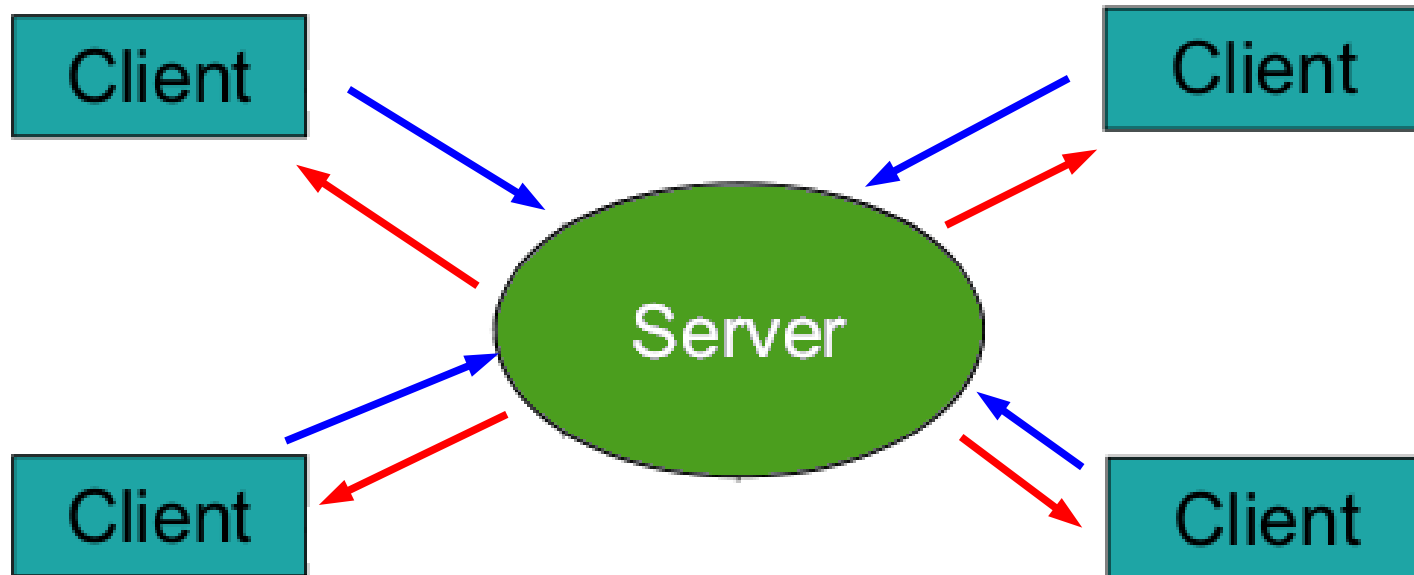
SPIELPRINZIP



# Anforderungen

- *Server*
  - *Kennt die Spielregeln*
  - *Empfängt Spielzüge von den Clients*
  - *Verteilt Karten*
  - *Verteilt Gamestate an die Clients*
- *Client*
  - *Zeichnet das Brett/Spiel*
  - *Lässt User interagieren*
  - *Schickt Spielzüge an den Server*
  - *Zeichnet die Änderungen vom Server*

# Schnittstellen zwischen Client und Server



Blau: Spielzüge  
Rot: Spielstatus

Synchron: Spielablauf

Asynchron: Chat

## Clients

**SEND:** Spieler wählt Karte

**GET:** Ablegestapel rendern

**GET:** Spielfeld rendern

**GET:** Anzeige aktualisieren

**SEND:** Nächster Spieler führt  
Aktion aus

...

## Server

- **SEND:** Karte an andere Clients
  - **GET/SEND:** Führt Karte aus und aktualisiert Status
  - **SEND:** Bestimmt nächsten Spieler
  - **GET:** Reagiert auf Spielzug
-

# Projektplan

Vorgangsname	Dauer	Beginn	Ende	# Zuständige
Client/Server System	14 Tage	06.03.17	20.03.17	2
Netzwerk-Kommunikation	14 Tage	06.03.17	20.03.17	2
Kernkomponenten	16 Tage	28.03.17	13.04.17	2
Lobby GUI	16 Tage	28.03.17	13.04.17	2
Spielregelimplementierung	12 Tage	21.04.17	03.05.17	2
Spiel Grafik 1	12 Tage	21.04.17	03.05.17	2
Spiel Grafik 2	7 Tage	04.05.17	11.05.17	2
Handbuch und Dokumentation	7 Tage	04.05.17	11.05.17	2
Fehlerbehebung & Refactoring	13 Tage	04.05.17	18.05.17	4