

## **XML Praktikum**

# Mancala

### **Team XML Schema**

Michael Conrads  
Andreas Eichner  
Emiliyana Kalinova  
Ha Ngan Nguyen  
Son Nguyen

# Outline

1. Mancala
2. Technologien
3. Architektur
4. Objekt Design
5. Demo
6. Frontend
7. Backend
8. Ausblick

# Outline

- 1. Mancala**
2. Technologien
3. Architektur
4. Objekt Design
5. Demo
6. Frontend
7. Backend
8. Ausblick

# Mancala

Version:

- Kalah

Rahmen:

1. 2 Spieler
2. 6 Häuser pro Spieler
3. 1 Kalah pro Spieler

Sonderregeln:

1. Nochmal ziehen
2. In beide Kalahs legen
3. Klauen (Future Work)

# Outline

1. Mancala
- 2. Technologien**
3. Architektur
4. Objekt Design
5. Demo
6. Frontend
7. Backend
8. Ausblick

# Technologien

## Standards:

1. XML
2. Docbook
3. XQuery
4. SVG
5. CSS

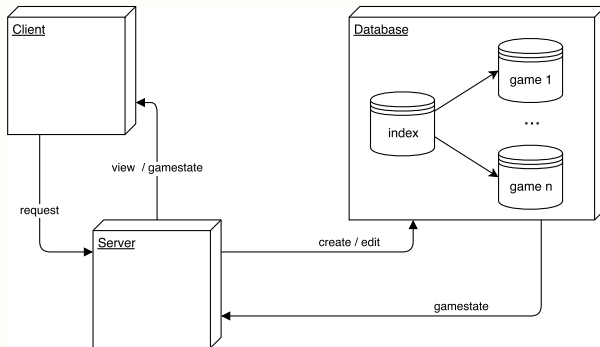
## Anwendung:

1. Trello (Tickets)
2. Git
3. BaseX & RESTXQ

# Outline

1. Mancala
2. Technologien
3. **Architektur**
4. Objekt Design
5. Demo
6. Frontend
7. Backend
8. Ausblick

# Architektur





# Outline

1. Mancala
2. Technologien
3. Architektur
4. **Objekt Design**
5. Demo
6. Frontend
7. Backend
8. Ausblick

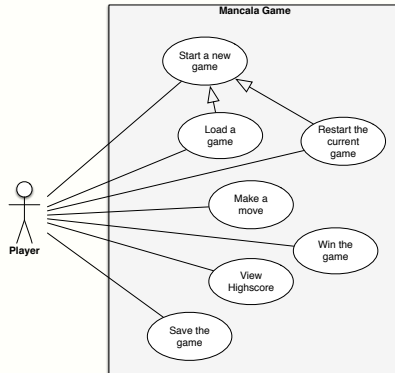
# Outline

1. Mancala
2. Technologien
3. Architektur
4. **Objekt Design**
  - 1 Anwendungsfalldiagramm
  - 2 Klassendiagramm
  - 3 Sequenzdiagramm
5. Demo
6. Frontend
7. Backend
8. Ausblick

# Outline

1. Mancala
2. Technologien
3. Architektur
4. Objekt Design
  - 1 **Anwendungsfalldiagramm**
  - 2 Klassendiagramm
  - 3 Sequenzdiagramm
5. Demo
6. Frontend
7. Backend
8. Ausblick

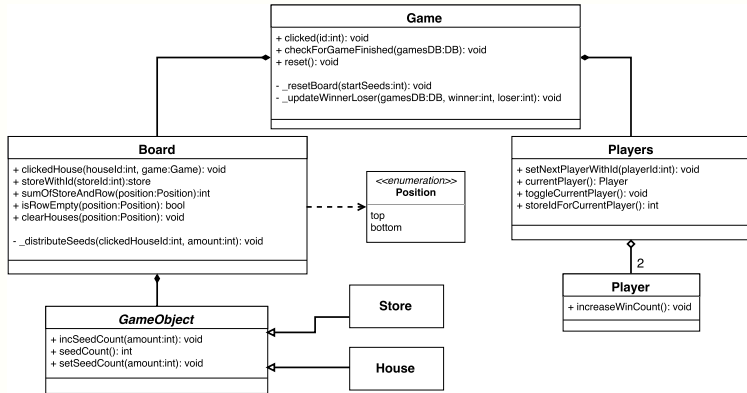
# Anwendungsfalldiagramm



# Outline

1. Mancala
2. Technologien
3. Architektur
4. Objekt Design
  - 1 Anwendungsfalldiagramm
  - 2 **Klassendiagramm**
  - 3 Sequenzdiagramm
5. Demo
6. Frontend
7. Backend
8. Ausblick

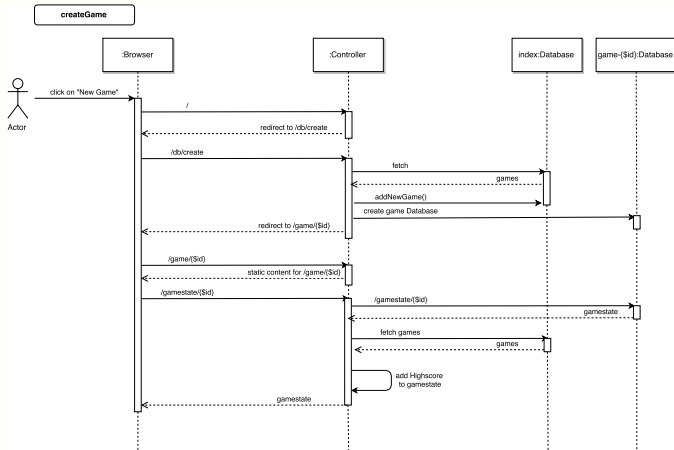
# Klassendiagramm



# Outline

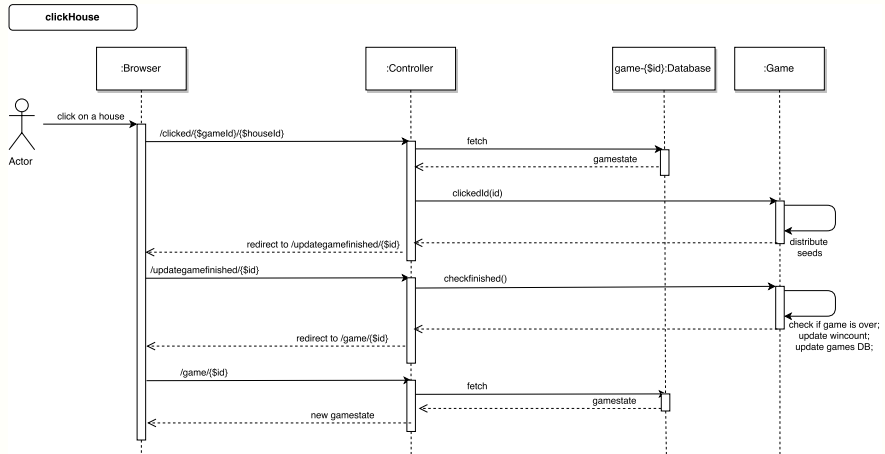
1. Mancala
2. Technologien
3. Architektur
4. Objekt Design
  - 1 Anwendungsfalldiagramm
  - 2 Klassendiagramm
  - 3 **Sequenzdiagramm**
5. Demo
6. Frontend
7. Backend
8. Ausblick

# Sequenzdiagramm





# Sequenzdiagramm



# Outline

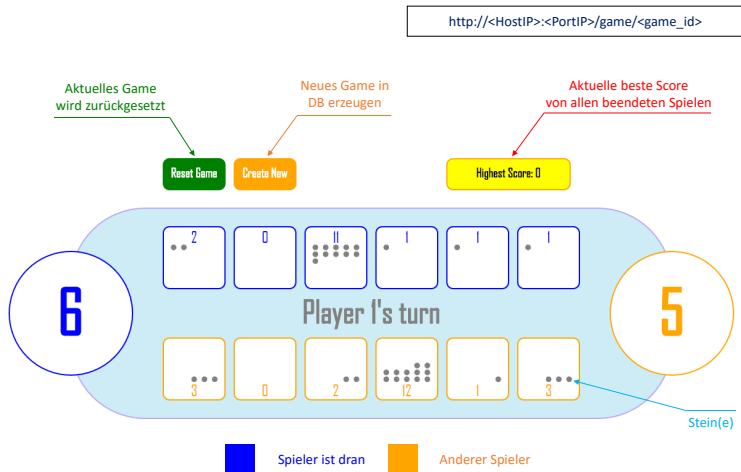
1. Mancala
2. Technologien
3. Architektur
4. Objekt Design
5. **Demo**
6. Frontend
7. Backend
8. Ausblick

# Demo

# Outline

1. Mancala
2. Technologien
3. Architektur
4. Objekt Design
5. Demo
6. **Frontend**
7. Backend
8. Ausblick

# Frontend



# Outline

1. Mancala
2. Technologien
3. Architektur
4. Objekt Design
5. Demo
6. Frontend
7. **Backend**
8. Ausblick

## Backend

- OOP
  - one xqm per class
  - one namespace per class
  - private methods prefixed with `_`
- Databases
  - one index database
  - one database per game
- Problems
  - FLWOR vs update:
    - Use getter instead of `let`
    - Use recursion instead of loop
    - Redirect frontend to other url for updated database
  - Single node updates:  
Precompute every value before updating a node

# Outline

1. Mancala
2. Technologien
3. Architektur
4. Objekt Design
5. Demo
6. Frontend
7. Backend
8. **Ausblick**



## Ausblick

- “Klauen” Regel
- Distributed Multiplayer
- KI Gegner
- User Ranking

Vielen Dank!

---