**Dokumentation des Projekts**

**2048 Multiplayer**

Der Klassen IT 8 i

an der G18

vorgelegt von Melanie Bär, Ricardo Kittmann und Jasper Lichte

Betreuer: Herr Rogge

Schuljahr: 2019/2020

# Inhalt

[1 Inhalt 2](#_Toc27567395)

[2 Einleitung 3](#_Toc27567396)

[2.1 Regelwerk 3](#_Toc27567397)

[2.2 Begriffserklärungen 3](#_Toc27567398)

[3 Voraussetzungen 4](#_Toc27567399)

[4 Architektur des Spiels - Design Pattern (MVC) 4](#_Toc27567400)

[5 Klassendiagramm 5](#_Toc27567401)

[6 Use Case Diagramm 5](#_Toc27567402)

[7 Sequenzdiagramm (exemplarisch) 7](#_Toc27567403)

[8 Testkonzept 8](#_Toc27567404)

[9 Fazit 9](#_Toc27567405)

[10 Abbildungsverzeichnis 10](#_Toc27567406)

# Abbildungsverzeichnis

Abbildung 1 Design Pattern (MVC) 4

Abbildung 2 Klassendiagramm 5

Abbildung 3 ERD Diagramm 6

# 

# Einleitung

Kurze Beschreibung des Spiels (Regelwerk ect)

Bei 2048 - Multiplayer handelt es sich um die Anwendung 2048 als Multiplayer. Ein Spielfeld besteht aus 4x4 Feldern. Sind mindestens zwei Spieler aktiv in einer Session, werden die Spielzüge zwischen den Spielern übertragen. Ist ein Spiel zu Ende erscheint eine nach den Ergebnissen geordnete Liste mit den Spielernamen und deren Ergebnissen. Es gibt einen Admin und normale User.

## Regelwerk

Der erste Spieler, der sich mit dem Server verbindet wird der Admin. Dieser muss warten, bis sich jeder Spieler mit einem Namen registriert hat. Jeder Name darf nur einmal vorhanden sein. Spieler ohne Namen werden vom Server zu Beginn des Spiels ausgeschlossen. Anschließend wird das Spiel vom Administrator gestartet, indem er sich selbst mit einem Namen registriert. Nur er hat diese Möglichkeit. Den Admin erkennt man an einem roten Banner mit der Aufschrift „Admin“ oben rechts.

## Begriffserklärungen

**REST**

REST steht für **RE**presentational **S**tate **T**ransfer. Gemeint ist eine Programmierschnittstelle, die sich an den Paradigmen und Verhalten des World Wide Web (WWW) orientiert und einen Ansatz für die Kommunikation zwischen Client und Server in Netzwerken beschreibt wobei Representational State Transfer die Übertragung von Repräsentationen beschreibt. Eine Repräsentation einer Ressource wird genutzt um den serverseitigen Zustand und andere Ressourcen in den Zustand der Applikation auf der Client-Seite zu übergeben.

**WebSocket**

WebSocket (ws) ist ein auf TCP basierendes Protokoll welches es möglich macht eine interaktive Kommunikations-Session zwischen dem Browser des Benutzers (Client) und dem Server herzustellen. Mit dieser API können Nachrichten zum Server gesendet werden und ereignisorientierte Antworten erhalten werden ohne beim Server die Antwort abzufragen.

# Voraussetzungen

* Einen Computer oder Laptop mit:
  + Betriebssystem deiner Wahl
  + ECMAScript2013 kompatibler Webbrowser deiner Wahl
* Ein Netzwerk, in dem sich alle erreichen können
* Mysql Datenbank

# Architektur des Spiels - Design Pattern (MVC)

Das Programm wurde anhand des Model-View-Controller aufgebaut. Die Daten werden am Ende einer Runde in einer MySQL gespeichtert (Model). Die Kommunikation übernimmt ein in C# .Net Core geschriebener Service (Controller). Eine kleine Übersicht, über die wichtigsten Klassen der Spiellogik, können dem Klassendiagramm in Abbildung 2 entnommen werden. Die View ist eine Webseite, welche mit TypeScript und SCSS erstellt wurde, und die Eingaben der User an den Controller kommuniziert.

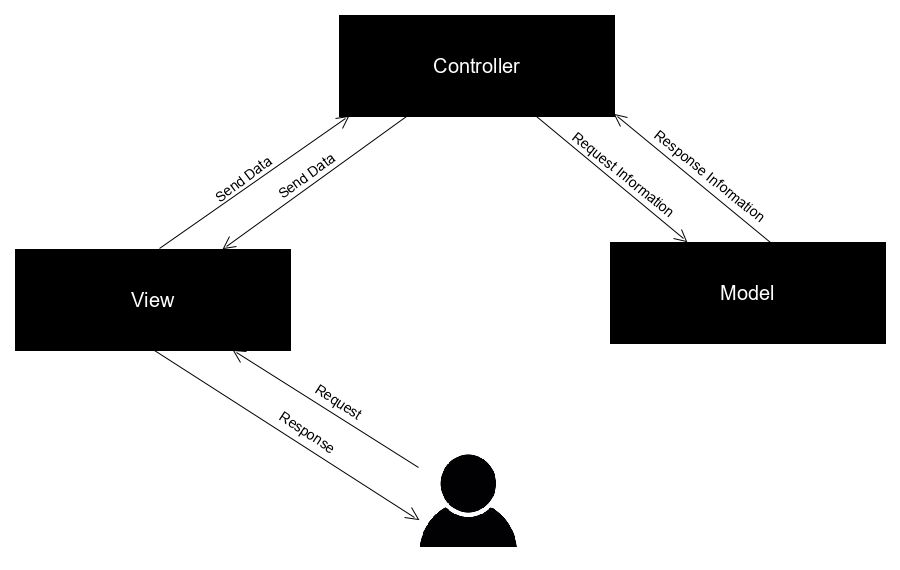


Abbildung Design Pattern (MVC)

# Klassendiagramm

In Abbildung 2 ist exemplarisch das Klassendiagramm der Klassen zu sehen, welche maßgeblich für die Funktion des Spiels und der Kommunikation zu einem Client verantwortlich sind.

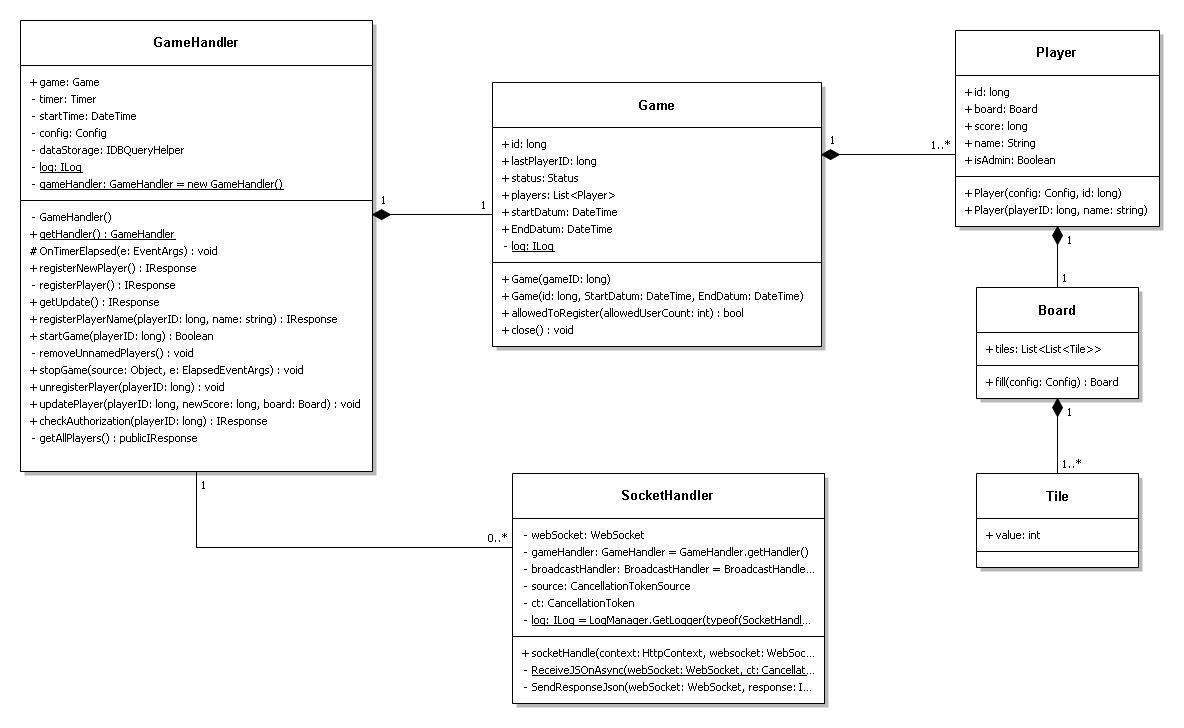


Abbildung Klassendiagramm

# Use Case Diagramm

Hier kommt eine kurze Einleitung hin.

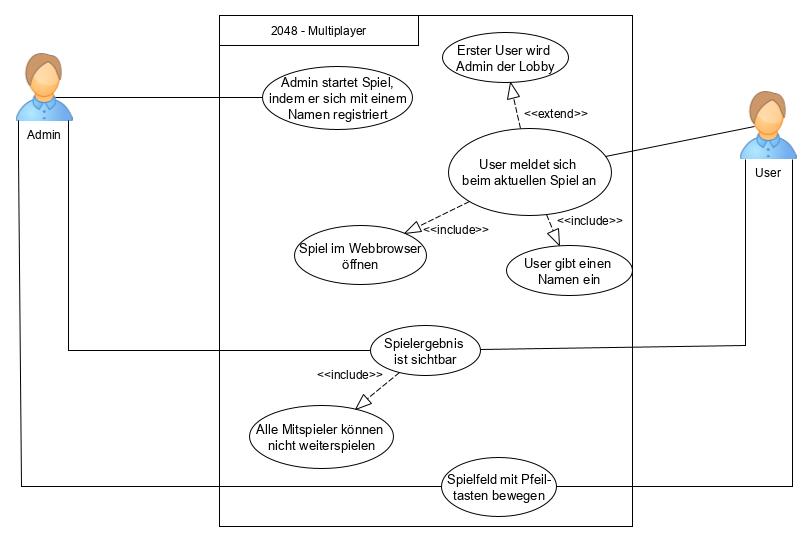


Abbildung Use Case Diagramm

# Sequenzdiagramm (exemplarisch)

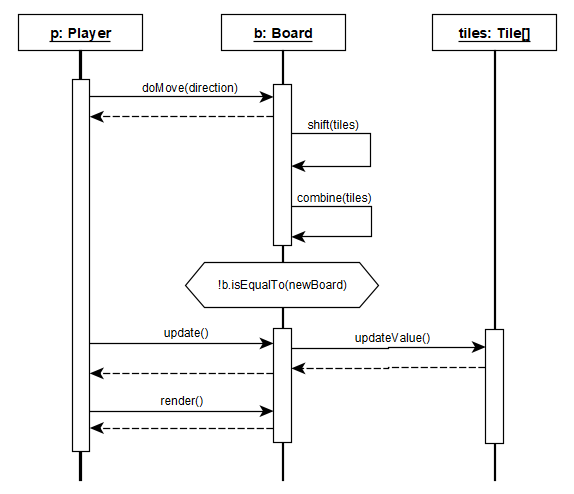


Abbildung ERD Diagramm

# Testkonzept

|  |  |  |  |
| --- | --- | --- | --- |
| Nummer | Vorbedingung | Aktion | Nachbedingung (Ergebnis) |
| 1 | Drei Mitspieler mit je drei bis vier Browserfenstern | 10 Mitspielern mit einer Minute Spielzeit durchführen | Absteigende Ergebnisliste mit allen Mitspielern |
| 2 | Drei Browserfenster im dem Opera Browser | Drei Browserfenstern und zwei Minuten Spielzeit durchführen | Absteigende Ergebnisliste mit allen Mitspielern |
| 3 | Drei Browserfenster im dem Microsoft Edge Browser | Drei Browserfenstern und zwei Minuten Spielzeit durchführen | Absteigende Ergebnisliste mit allen Mitspielern |
| 4 | Drei Browserfenster im dem Firefox 71.0 | Drei Browserfenstern und zwei Minuten Spielzeit durchführen | Absteigende Ergebnisliste mit allen Mitspielern |
| 5 | Drei Browserfenster im dem Firefox 70.0.1 | Drei Browserfenstern und zwei Minuten Spielzeit durchführen | Absteigende Ergebnisliste mit allen Mitspielern |
| 6 | Drei Browserfenster im dem Internet Explorer 11 | Zwei Browserfenstern und zwei Minuten Spielzeit durchführen | Kein Admin verfügbar. Das Spiel kann nicht gestartet werden |
| X | 24 Mitspieler mit je einem Browserfenster | 24 Mitspielern und zwei Minuten Spielzeit durchführen |  |

# Fazit

Schlussendlich sind die Gruppenmitglieder durchwegs zufrieden mit der Wahl ihres Projektes. So beinhaltet die Umsetzung einer Version vom bekannten Spiel 2048 als Mehrspieler Version viele und wichtige Konzepte und Prinzipien der (Web-)Entwicklung und der objektorientierten Programmierung im Allgemeinen. Hierdurch konnten die Gruppenmitglieder in vielen Hinsichten ihr Wissen und ihre Kompetenzen erweitern und ergänzen und hatten bei der Implementierung ihrer Ideen viel Spaß, sich kreativ auszutoben.



