Projektdokumentation   
Snake-Core, SnakeFX, SnakeServer

Ostfalia Fachhochschule für angewandte Wissenschaften

Benjamin Wulfert

Leonard Reidel

Semester: Wintersemester 2020

23. Dezember 2020

Inhaltsverzeichnis

[Projektdokumentation 4](#_Toc57539720)

[Modul-Architektur 4](#_Toc57539721)

[Core 4](#_Toc57539722)

[SnakeFX - Frontend 4](#_Toc57539723)

[SnakeServer - Backend 4](#_Toc57539724)

[SnakeFX - Front End 5](#_Toc57539725)

[Login-Screen 5](#_Toc57539726)

[Registrierungs-Prozess 5](#_Toc57539727)

[Home-Screen 5](#_Toc57539728)

[Aktive Spieler 5](#_Toc57539729)

[Aktive Spiele 5](#_Toc57539730)

[Spielhistorie-Screen 5](#_Toc57539731)

[New-Game-Screen 6](#_Toc57539732)

[Snake-Implementierung 6](#_Toc57539733)

[SnakeServer 8](#_Toc57539734)

[Persistenz-Layer 8](#_Toc57539735)

[H2 Database Engine 8](#_Toc57539736)

[API-Layer / Schnittstelle 9](#_Toc57539737)

[Tasks / Aufgabenstellungen 9](#_Toc57539738)

[Realisiert 10](#_Toc57539739)

[UML-Diagramme 10](#_Toc57539740)

[Front-End Architektur 10](#_Toc57539741)

[User-Interfaces 10](#_Toc57539742)

[Initiale Asynchronität 10](#_Toc57539743)

[Initialer Konsum der Schnittstelle 10](#_Toc57539744)

[Planung und Realisierung der Bewegungsberechnung der Snake-Implementierung 10](#_Toc57539745)

[Snake-Implementierung – Mechanik - Wegfindung 11](#_Toc57539746)

[Integration der Design Patterns 11](#_Toc57539747)

[Überarbeitung der UML-Diagramme 11](#_Toc57539748)

[Ausstehende Arbeitspakete 11](#_Toc57539749)

[Absicherung der Schnittstelle (API) mittels Spring Security und JSON-Web-Tokens 11](#_Toc57539750)

[Vollständige Implementierung der Klassenrelationen 11](#_Toc57539751)

[Vollständiger Konsum der Schnittstellen 12](#_Toc57539752)

[Austausch der Spielereingaben (Implementierung von Web-Sockets) 12](#_Toc57539753)

[Komplexität des GameControllers reduzieren 12](#_Toc57539754)

[Regelwerk ausbauen 12](#_Toc57539755)

[Feedback aus der Präsenz 12](#_Toc57539756)

[RespawnSnake() 🡪 Raus 12](#_Toc57539757)

[Draw-Funktionalität in Klassen auslagern. 12](#_Toc57539758)

[Szenen-Wechsel 13](#_Toc57539759)

[Design-Patterns 14](#_Toc57539760)

[Composite-Pattern 14](#_Toc57539761)

[Factory-Pattern 14](#_Toc57539762)

[Observer-Pattern 15](#_Toc57539763)

[Anhang 17](#_Toc57539764)

[UML – Anwendungsfalldiagramm 17](#_Toc57539765)

[UML – Aktivitätsdiagramm 17](#_Toc57539766)

[UML – Klassendiagramm 19](#_Toc57539767)

[User-Interface – Frontend - SnakeFX 20](#_Toc57539768)

# Projektdokumentation

Dieses Dokument stellt die Dokumentation für das Projekt dar. Im Folgenden werden die verschiedenen Aspekte des Projekts beschrieben. Des Weiteren wird in dieser Lösungsdokumentation dargelegt welche Aspekte des Systems umgesetzt wurden sowie noch ausstehende Punkte erläutert, welche in den folgenden Wochen umgesetzt werden.

Der aktuelle Projektstand (Dokumente, UML-Diagramme, gesamter Projektcode, etc. ) kann dem folgenden GitHub-Repository entnommen werden:

<https://github.com/benjaminfoo/SnakeFX>

<https://github.com/Bummelnderboris/Patterns_and_Frameworks>

# STOMP-Server

Der folgende Abschnitt beschreibt die Integration des STOMP-Services sowie dessen Bereitstellung. Des Weiteren wird erläutert wie mithilfe eines STOMP-fähigen Clients die Kommunikation zwischen dem Front- und dem Backend hergestellt und verwendet werden kann.

ws://localhost:8080/chat

## StompClient

Der StompClient …

*CONNECTION\_URL* -> Die URL zum STOMP-Service.

## StompSessionHandler

Der StompClient …

## ChatController

Der StompClient …

## WebSocketConfig

Der StompClient …

Was ist ein Broker?

Ein Broker

Was ist ein Endpoint?

Neues Spiel in der Lobby

Authentisierter Client erzeugt neue Spiel definition mittels „Neues Spiel -> Erstellen“ schaltflächen. Im Anschluss dessen wird

/snakeserver  
Endpoint für http-Handshake und STOMP-Upgrade

/topic/  
Broker-Topic für Broadcast-Messages

/topic/messages  
Dient dem Austausch von Chat-Nachrichten

/topic/games  
Informationen über alle (laufenden) Spiele in der Lobby

/app/games/{gameId}

Ein Aufruf an /app/games/1 mit einer GameInputMessage hat zufolge, dass alle Clients welche bspw. die URL „ws://localhost:8080/app/game/1“ subscribed sind alle folgenachrichten der Clients erhalten – somit ist die Eingabe für Multiplayer gesichert =)