

# Dokumentation Projektarbeit Game Console App

## Einleitung

Auf dem kommerziellen Markt gibt es diverse Hardware-Spielkonsolen (HSK). Ziel dieses Projektes war es, analog zu den HSK, eine Software-Spielkonsole zu programmieren, welche sich mit verschiedenen Spielen modular erweitern lässt und flexibel bezüglich dem User Interface (UI) ist.

## Methoden & Technologien

**Spring Boot (SB) & Maven:** Zum Zweck der Modularität wurde ein SB Framework (FW) auf der Basis von Maven aufgesetzt. Dabei wurden die Applikationskomponenten innerhalb des FW als unabhängige Spring Boot/Maven-Module implementiert.

**UI: Console (MVP), Web (<http://localhost:8080/>) und GUI (JavaFX)** wurden als mögliche UI's in die Game-Console-App implementiert, wobei alle UI's prinzipiell für alle Games zur Verfügung stehen. Jedes UI wird daher durch eine eigene Main-Datei gestartet:

MainConsole.java, MainWeb.java, respektive MainGui.java, wobei diese mittels "Transitive Dependencies" die jeweilige Programmlogik der entsprechenden Games einbinden.

**Eingesetzte Technologien & Programmierkonzepte:** Java (Amazon Corretto 11.0.4); Spring Boot & Spring Boot Maven Plugin; Console-App (Spring Boot Starter); Web-App (Spring Boot Web, Thymeleaf & Spring Boot Developer Tools); GUI-App (JavaFX Maven Dependencies (11.0.2) & JavaFX Maven Plugin Maven Mojo); Modularisierung (Spring Module: Games/Core/GUI/Web); Exception Handling; Annotation Interfaces; Unit Testing (Spring Boot Starter Test mit JUnit 5), Javadoc (Apache Maven Javadoc Plugin).

**Java API Docs & UML-Diagramme:** Die kompletten API Docs, sowie die UML-Diagramme der einzelnen Komponenten befinden sich im Projektordner "docs/\*\*/\*".

**GitHub Repository:** <https://github.com/albiurs/game-console-app>

## Resultate & Implementierung

Die total vier Games wurden in den UI's wie folgt implementiert:

- **Console:** Tic-Tac-Toe, Guess the Number Game, Random Joke Crawler
- **GUI:** Game of Life
- **Web:** Guess the Number Game

## Start-Module

**Was das Modul macht:** Die Start-Module (start-core, start-gui und start-web) implementieren die Startseite des jeweiligen UI. Bis zum Zeitpunkt des Abgabetermins der Projektarbeit wurde aus Zeitgründen nur für das Modul start-core eine Startseite implementiert.

## Game of Life (Conway)

**Was das Modul macht:** Das Game of Life ist eine Computersimulation. Aktive Zellen verteilen sich auf einem begrenzten Feld. Je nach Nachbarschaft bleiben sie am Leben, sterben oder bringen neues Leben hervor. Das Spiel des Lebens wird durch vier Regeln kontrolliert:

- Eine lebendige Zelle stirbt, wenn sie weniger als zwei lebendige Nachbarzellen hat.
- Eine lebendige Zelle mit zwei oder drei lebenden Nachbarn lebt weiter.
- Eine lebendige Zelle mit mehr als drei lebenden Nachbarzellen stirbt im nächsten Zeitschritt.
- Eine tote Zelle wird wiederbelebt, wenn sie genau drei lebende Nachbarzellen hat.

## Tic Tac Toe

**Was das Modul macht:** Tic Tac Toe ist ein Spiel, welches im Terminal gespielt werden kann. Das Spielfeld hat 3x3 Felder. Der Computer setzt zufälligerweise das Symbol 'O' auf dem Spielfeld, nachdem der Spieler sein Feld ausgewählt hat. Das Programm findet über eine ArrayList heraus, ob der Spieler gewonnen, verloren oder ein Unentschieden gemacht hat.

## Guess the Number Game

**Was das Modul macht:** Eine Zahl im Bereich zwischen 0 und 100 muss erraten werden. Nach einer Eingabe einer Zahl gibt der Computer einen Hinweis zurück, ob die Zahl zu hoch oder zu niedrig ist, respektive ob sie erraten wurde.

## Random Joke Crawler

**Was das Modul macht:** Das Modul zeigt einen zufälligen Witz aus verschiedenen Online-Quellen an.

## Diskussion

Die Ziele des Projektes, eine Game Console App mit verschiedenen UI's zu programmieren wurde erreicht.

Aufgrund der zeitlich begrenzten Projektdauer konnten nicht alle Komponenten / Module in alle UI's implementiert werden. Zusätzlich wurden auch die JUnit tests aufgrund der zeitlichen Begrenzung nicht vollständig implementiert.

Weitere Schritte und mögliche Erweiterungen sind demzufolge die Implementierung der Start- und Spiel-Module in alle drei UI's, sowie die vollständige Entwicklung der zugehörigen Unit Tests. Zusätzlich wäre eine Benutzerverwaltung wünschenswert, mit der Option den Spielstand und die jeweiligen Top-Scores zu speichern.

# Referenzen

Dieses Projekt wurde auf zahlreichen Wissensressourcen abgestützt:

1. Apache Maven Project Documentation  
<https://maven.apache.org/guides/index.html>
2. Spring Boot Reference Documentation  
<https://docs.spring.io/spring-boot/docs/current/reference/html/>
3. Spring Initializr  
<https://start.spring.io/>
4. MVN Repository  
<https://mvnrepository.com/>
5. Udemy: Java Programming Masterclass for Software Developers  
<https://www.udemy.com/course/java-the-complete-java-developer-course/>
6. Udemy: Spring Framework 5: Beginner to Guru  
<https://www.udemy.com/course/spring-framework-5-beginner-to-guru/>
7. Udemy: Java Spring Tutorial Masterclass - Learn Spring Framework 5  
<https://www.udemy.com/course/java-spring-framework-masterclass/>
8. Udemy: Master Java Unit Testing with Spring Boot & Mockito  
<https://www.udemy.com/course/learn-unit-testing-with-spring-boot/>
9. JavaFX: Gluon  
<https://gluonhq.com/products/javafx/>
10. JavaFX: JavaFX+Maven+Spring – Step by step  
<https://drogago.net/javafxmavenspring-step-by-step/>
11. Game of Life: Beispiele von Source Codes als Idee für eigenen Code  
<https://github.com/vanillaSlice/GameOfLife>  
<https://github.com/prakashsur/game-of-life>  
<https://math.hws.edu/javanotes/source/chapter7/Life.java>