

HTW Berlin
Fachbereich 4 - Informatik, Kommunikation und Wirtschaft
Angewandte Informatik

Wintersemester 2021/2022

Dokumentation zum semesterbegleitendem Android Projekt
“Grundlagen mobiler Anwendungen”

Veranstaltung: B35.1 MA Grundlagen mobiler Anwendungen

Dozent: Prof. Dr. Thomas Schwotzer

Bearbeitet von: Florian Symmank

Matrikel Nummer: 578767

GitHub: <https://github.com/FlorianSymmank/ConwaysGameOfLifeApp>

1.) Erörterung der Projektidee - Conway's Game of Life

Conway's Game of Life, auch bekannt als Life, ist ein lokales Null-Spieler-Spiel bei dem der Ausgangszustand die Spielentwicklung bestimmt, es sind keine weiteren Eingaben notwendig. Die zu entwickelnde mobile Anwendung soll dieses Spiel auf einem Android Smartphone nachbilden und zusätzliche Funktionen, die nicht mit dem Spielablauf zusammenhängen anbieten.

Conway's Game of Life wird üblicher Weise als ein 2-Dimensionales Grid (Spielfeld) dargestellt, welches aus vielen Zellen besteht. Zellen haben zwei Zustände, lebendig oder tot. Die Spielregeln sind sehr einfach, trotzdem ist eine sehr komplexe Entwicklung zu betrachten.

Vorgehensweise:

- 1.) Der Spieler interagiert mit dem Grid und legt für jede Zelle ihren Zustand fest.
- 2.) Eine neue Generation des Grids wird durch die Anwendung der Regeln auf alle Zellen festgelegt.
- 3.) Punkt 2 wird solange fortgesetzt, bis eine Generation des Grids einer beliebigen vorhergehenden gleicht.

Regeln für die Zellen:

- Eine Zelle ist lebendig, wenn sie drei lebendige Nachbarn hat, oder sie selbst lebendig ist und zwei lebendige Nachbarn hat.
- Eine Zelle ist tot, wenn sie mehr als drei lebendige Nachbarn hat, wenn sie null oder einen lebendigen Nachbarn hat, falls eine Zelle tot ist und zwei lebendige Nachbarn hat bleibt sie tot.

Die Regeln werden auf alle Zellen gleichzeitig angewendet.

Nachbarn sind alle diagonal, horizontal und vertikal angrenzende Zellen.

Neben dem reinen Auflauf des Spiels, soll mit diesem über eine Benutzeroberfläche zur Interaktion implementiert werden, sodass einzelne Zelle manipuliert werden können. Zur Laufzeit des Spiels soll Statistik über die Anzahl der Generation, die Anzahl der lebendigen Zellen und toten Zellen geführt werden. Das Spiel kann vom Benutzer gestartet, angehalten, neugestartet und zurückgesetzt werden. Außerdem sollen Zustände des Spiels auf Anfrage des Nutzers gespeichert werden können, sodass sie zu einem späteren Zeitpunkt fortgeführt werden können. Am Ende eines jeden Spiels soll der Nutzer gefragt werden, ob er sein Ergebnis mit anderen Teilen möchte. Die gespeicherten Zustände, sowie die Ergebnisse des Nutzers und anderer sollen eingesehen werden können.

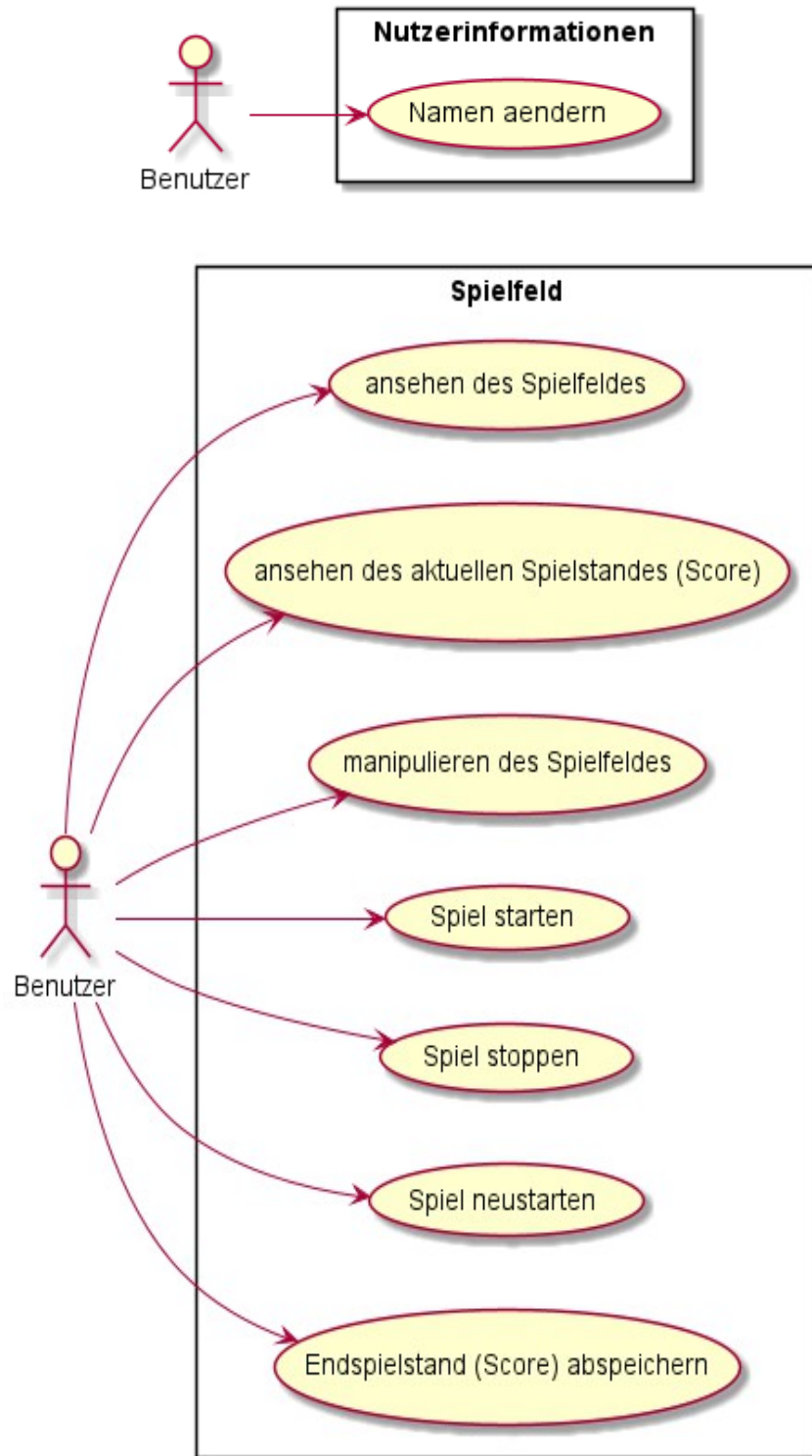
Es ist geplant die gespeicherten Daten lokal vorzuhalten, sodass keine Internetverbindung benötigt wird.

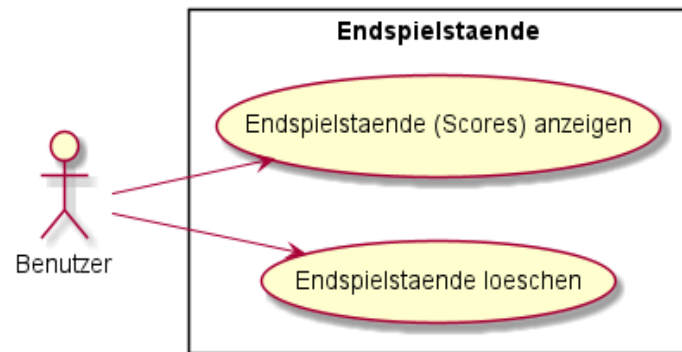
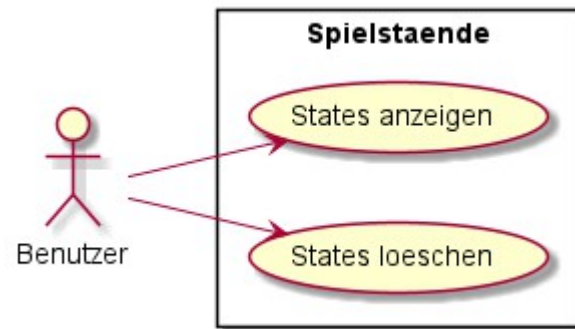
Zusätzlich zur deutschen Lokalisierung soll auch eine englische zur Verfügung stehen

Es soll eine Möglichkeit geben zwischen einem Dark- und Light-Mode zu wechseln.

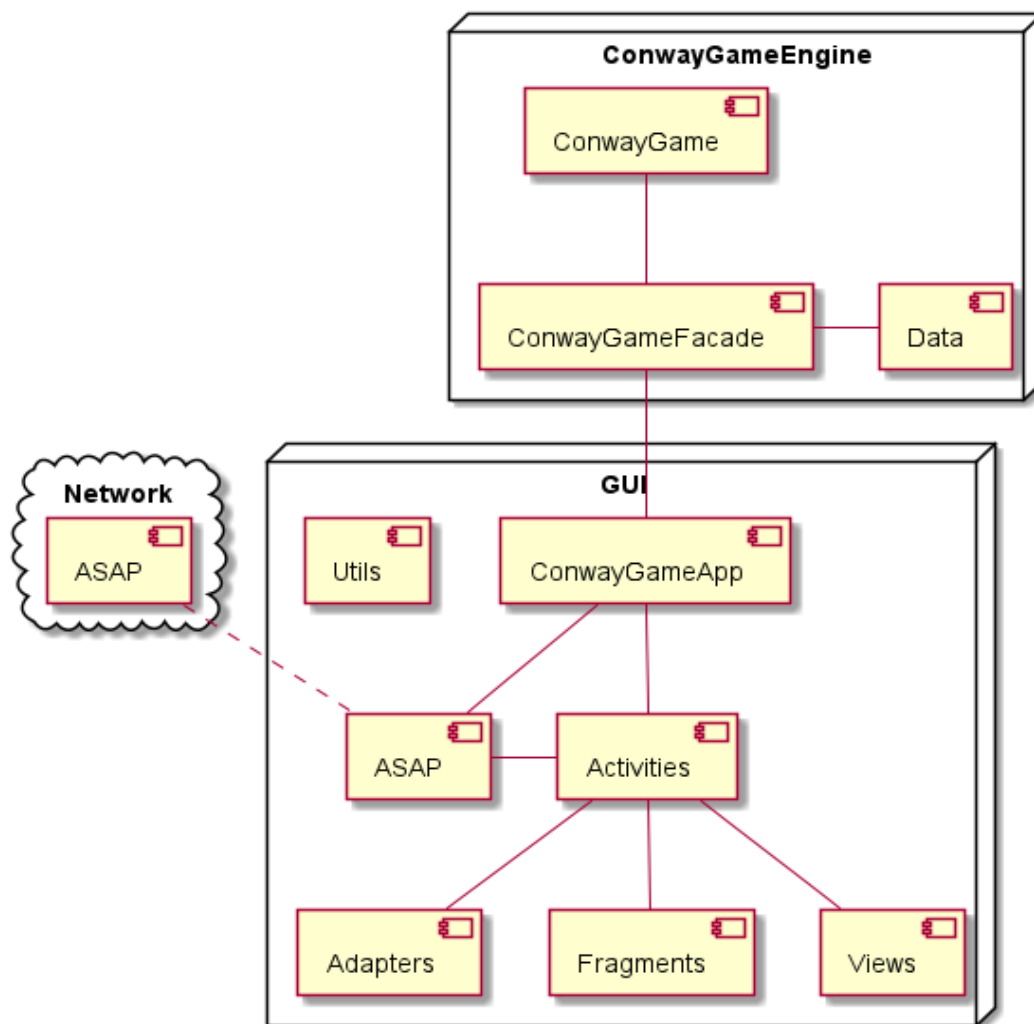
2.) Anwendungsfalldiagramme

Die Anwendungsfalldiagramme stellen die externe Anwendersicht auf das System dar und gibt uns einen Überblick über die Anforderungen an die Anwendung.





3.) Übersicht Komponenten



4.) Komponenten Beschreibung

I. GUI

I.I. Zusammenfassung

Stell die graphische Benutzeroberfläche für Android zur Verfügung. Die Komponenten sind im Sinne von MVC aufgebaut.

I.II. Struktur/Technik

Es sind folgende Aktivitäten entstanden: **GameActivity**, **IntialActivity**, **SavedGamesListActivity**, **SavedScoresActivity** und **SettingsActivity**.

Nach dem Start der App ist entweder die **GameActivity** oder, im Fall des ersten Starts der App, die **SettingsActivity** zu sehen.

GameActivity ist die Aktivität in der sich hauptsächlich aufgehalten wird, hier wird das Spiel angezeigt und gespielt.

In der **SettingsActivity** können der Spielername, die Anzahl der Zeilen und Spalten des Spiels sowie der Zeitintervall zwischen den einzelnen Generationen angepasst werden. Es ist auch möglich die Einstellungen auf den Standard zurückzusetzen.

SavedGamesListActivity und **SavedScoresActivity** sind sehr ähnlich und bieten jeweils eine Auflistung der Scores oder gespeicherten Spiele an. Die Auflistung ist durch RecyclerViews sowie den dazugehörigen Adaptern geregelt. Beide bieten die Möglichkeit an, den gesamten Inhalt zu löschen. In der **SavedGamesListActivity** kann ein Spiel ausgewählt werden welches danach in der **GameActivity** fortgesetzt werden kann.

Die **IntialActivity** ist die Startaktivität die bei jedem App Start aufgerufen wird und das System (**ConwayGameApp**) initialisiert.

Alle Aktivitäten, bis auf die **IntialActivity** werden von **ASAPActivity** abgeleitet und können somit Bluetooth Funktionalitäten anbieten.

I.III Schnittstelle

Es liegt keine Schnittstelle gegenüber anderen Komponenten vor.

I.IV Testkonzept mit Espresso

Testfall	Testname	Implementiert	Erfolg
Test ob Klick auf die Spielfläche (Zelle invertieren, Darstellung aktualisieren)	GameOfLifeViewTest.InTouchEvent	Ja	Ja
Klick auf Start soll das Spiel starten	GameActivityTest.fabClickStartsGame	Ja	Ja
Klick auf Halt soll das Spiel anhalten	GameActivityTest.fabDoubleClickStopsGame	Ja	Ja
Klick auf Neustart soll das Spiel neustarten (Scores zurücksetzen)	-	Ja	Nein
Geladene Spielstände sollen den letzten Spielstatus korrekt anzeigen	-	Ja	Nein
vorhanden Spielstände laden	-	Ja	Nein
Langer Klick Spielstand gelöscht werden soll	-	Nein	Nein
Scores werden absteigend angezeigt	-	Nein	Nein

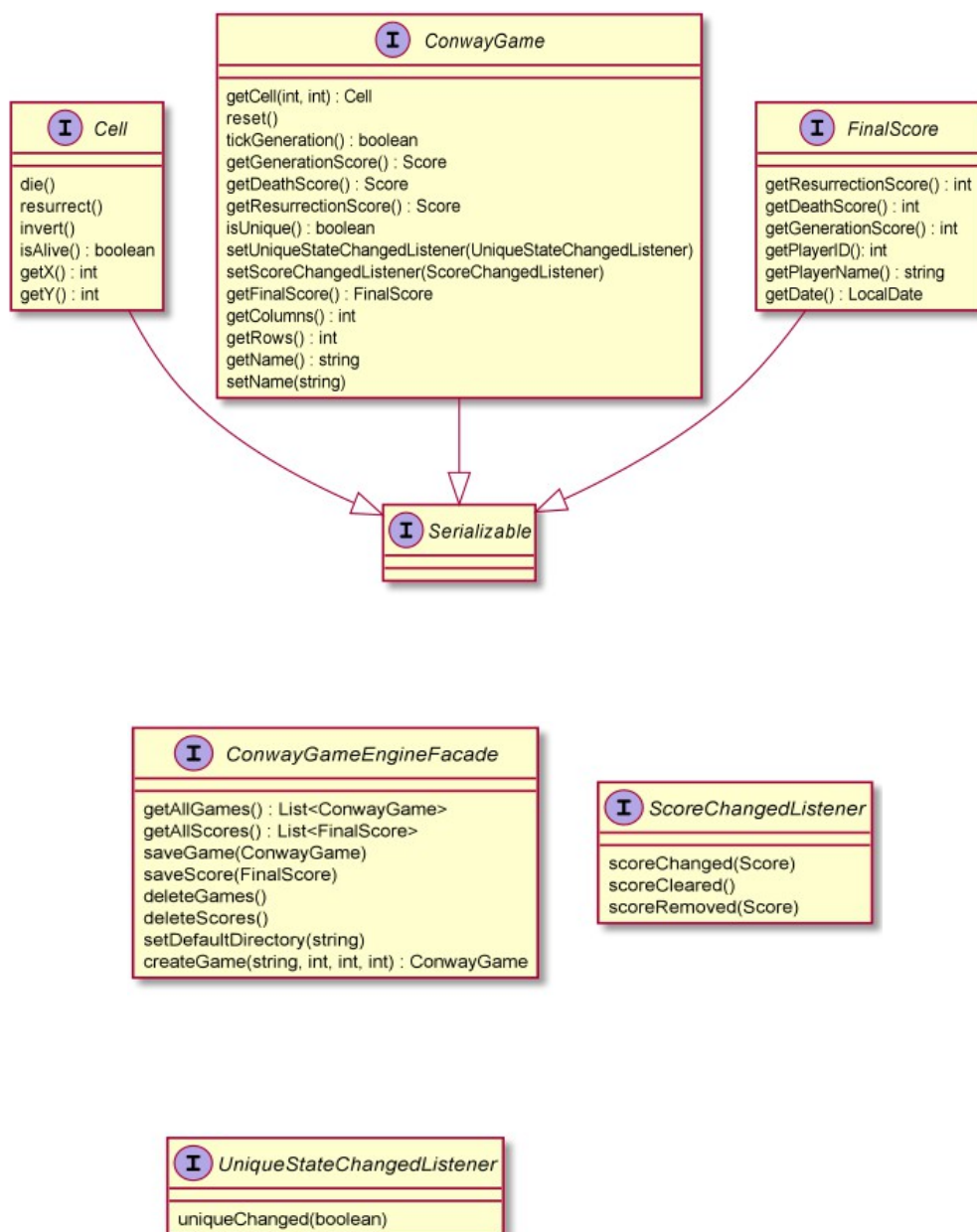
Speicherndialog akzeptieren sichert Zustand	-	Nein	Nein
Beendigung bringt des Score teilen Dialog	-	Nein	Nein

Fehlende Tests aufgrund von fehlender Zeit nicht implementiert. Funktionalitäten durch manuelle Tests aber vollständig überprüft und für funktional befunden.

II. ConwayGameEngine

II.I Zusammenfassung

Realisiert die gesamte Spiellogik und die Persistierung der Daten. Nachfolgend sind die interessanteren Interfaces, welche innerhalb der Komponente implementiert wurden aufgelistet.



II.II Struktur/Technik

Cell repräsentiert hier eine Zelle des Conway Spiels und wird in der Implementation des **ConwayGame** Interfaces verwaltet.

Das **ConwayGame** Interface stellt implementiert die gesamte Ablauflogik des Spiels bereit. z.B.: erzeugen neuer Generationen. Die Verwaltung der Scores wird über eine andere Klasse innerhalb des **ConwayGame** realisiert.

FinalScore stellt den Endpunktestand des Spiels bei Beendigung dar.

Cell, **ConwayGame** und **FinalScore** implementieren jeweils Serializable, dadurch wird erreicht das ein Objekt dieser Klassen, ohne größere Umstände in der Implementierende Klasse des Interfaces **ConwayGameEngineFacade** persistiert werden kann. Die **ConwayGameEngineFacade** übernimmt das Verwalten der **ConwayGames** und **FinalScores**.

Rückblickend könnte man die Persistierung der Daten von der Verwaltung trennen. In die eigentliche Implementation ist allerdings sehr trivial, daher habe ich dies nicht als notwendig angesehen.

II.III Schnittstelle

Die **ConwayGameEngineFacade** wird als Hauptauffangstelle für andere Komponenten genutzt.

II.IV Testkonzept – Junit Tests

Testfall	Testname	Erfolg
Korrekten Spielzustand durch das GameMemento erhalten	GameMementoTest.getState()	Ja
alle vorhandenen Spielstände erhalten	ConwayGameEngineFacadeImpl.getAllGames()	Ja
alle vorhandenen Scores erhalten	ConwayGameEngineFacadeImplTest.getAllScores()	Ja
Spielzustand lokal abspeichern	ConwayGameEngineFacadeImplTest.saveGame()	Ja
Spielscore lokal abspeichern	ConwayGameEngineFacadeImplTest.saveScore()	Ja
lokale Spielzustände löschen	ConwayGameEngineFacadeImplTest.deleteGames()	Ja
lokale Spielscores löschen	ConwayGameEngineFacadeImplTest.deleteScores()	Ja
korrekte Zustandsübergänge in der ConwayWorld	ConwayGameImplTest.changedListenerCalled()	Ja
Erkennung sich wiederholender Zustände	ConwayGameImplTest.isUnique()	Ja
Scores des Spiels temporär merken	ConwayGameImplTest.changedListenerCalled()	Ja

Diesmal sind alle Tests implementiert.

5.) Liste der Umgesetzten Features

1. Spiellogik für Conways's Game of Life → vollständig umgesetzt
2. Zellenmanipulation über die Benutzeroberfläche → vollständig umgesetzt
3. Statistik während des Spiel → vollständig umgesetzt
4. Kontrolle über das Spiel über die Benutzeroberfläche (Start, Stopp, Neustart) → Logik vollständig vollständig umgesetzt, teilweise fehlende GUI Tests
5. Spielstände abspeichern/Spielstände laden und fortführen → Logik vollständig vollständig umgesetzt, fehlende GUI Tests
6. Teilen des Spielscore nach Spielende → Logik vollständig vollständig umgesetzt, fehlende GUI Tests
7. Ansehen aller Spielscores → Logik vollständig vollständig umgesetzt, fehlende GUI Tests
8. Daten lokal vorhalten → vollständig umgesetzt
9. Deutsche und Englische Lokalisierung → vollständig umgesetzt
10. Dark- und Light-Mode → verworfen

6.) GitHub-Repositorys

Conway's Game of Life App für Android:

<https://github.com/FlorianSymmank/ConwaysGameOfLifeApp>

Spiellogik:

<https://github.com/FlorianSymmank/ConwayGameEngine>

Bluetooth Messaging durch:

<https://github.com/SharedKnowledge/ASAPAndroid>