**Dokumentation einer beispielhaften Projektarbeit**

**im Rahmen der Abschlussprüfung zum Fachinformatiker/-in für Anwendungsentwicklung**

**„Game of life“**

**Bearbeitungszeitraum:**

[03.03.2021] bis [17.05.2021]

Inhalt

[1 Einleitung 3](#_Toc70422252)

[1.1 Projektumfeld 3](#_Toc70422253)

[2 Projektplanung 3](#_Toc70422254)

[2.1 Projektphasen 3](#_Toc70422255)

[3 Projektentwurf 3](#_Toc70422256)

[3.1 Ist-Analyse 3](#_Toc70422257)

[3.2 Design/Entwurf des Programms 3](#_Toc70422258)

[4 Projektdurchführung 4](#_Toc70422259)

[4.1 Implementierung/Realisierung 4](#_Toc70422260)

[4.2 Tests/Qualitätssicherungsmaßnahmen 4](#_Toc70422261)

[5 Reflexion 4](#_Toc70422262)

[6 Quellen 4](#_Toc70422263)

## 

## 1 Einleitung

### Das Spielfeld

Das Spielfeld ist in Zeilen und Spalten unterteilt und im Idealfall unendlich groß. Jedes Gitterquadrat ist ein [zellulärer Automat](https://de.wikipedia.org/wiki/Zellul%C3%A4rer_Automat) ([Zelle](https://de.wikipedia.org/wiki/Zelle_(Biologie))), der einen von zwei Zuständen einnehmen kann, welche oft als lebendig und tot bezeichnet werden. Zunächst wird eine Anfangs[generation](https://de.wikipedia.org/wiki/Generation) von lebenden Zellen auf dem Spielfeld platziert. Jede lebende oder tote Zelle hat auf diesem Spielfeld genau acht Nachbarzellen, die berücksichtigt werden ([Moore-Nachbarschaft](https://de.wikipedia.org/wiki/Moore-Nachbarschaft)). Die nächste Generation ergibt sich durch die Befolgung einfacher Regeln.

Das Spiel kann manuell auf einem Stück Papier oder mit [Computerhilfe](https://de.wikipedia.org/wiki/Computer) simuliert werden. Da ein reales Spielfeld immer einen Rand hat, muss das Verhalten dort festgelegt werden. Man kann sich den Rand zum Beispiel durch tote Zellen belegt denken, so dass manche Gleiter ihre Bewegungsrichtung dort ändern. Eine andere Möglichkeit ist ein [Torus](https://de.wikipedia.org/wiki/Torus)-förmiges Spielfeld, bei dem alles, was das Spielfeld nach unten verlässt, oben wieder hereinkommt und umgekehrt, und alles, was das Spielfeld nach links verlässt, rechts wieder eintritt und umgekehrt.

Alternativ kann man auch nur lebendige Zellen und ihre direkte Umgebung simulieren und bei Bedarf mehr Speicher allozieren, da große tote Flächen tot bleiben. So hat man zumindest ein quasi-unendliches Feld.

Anstatt auf einer quadratisch gerasterten Ebene kann die Simulation auch auf einer sechseckig gerasterten Ebene erfolgen. Dann beträgt die Zahl der Nachbarzellen nicht acht, sondern sechs. Es gibt auch dreidimensionale Game of Life Simulationen.

Eine weitere Variationsmöglichkeit ist, mehr als zwei mögliche Zustände der Gitterzellen einzuführen.

## 2 Projektplanung

### 2.1 Projektphasen

Zeitplanung

Analysephase  
 Ist-Analyse 1h

User Stories erstellen 3h

Entwurfsphase

Entwurf Benutzeroberfläche 1h

Entwurf Komponenten 1h

Implementierung

Implementierung Geschäftslogik 3h

Implementierung GUI 4h

Qualitätsmanagements

Unit Tests schreiben

Debugging, Funktionstests 8h

Dokumentation

Erstellen der Projektdokumentation 5h

Erstellen der Benutzerdokumentation 1h

Die Entwicklung erfolgt Scrum mit Sprints im 2-Wochen-Abstand.

Die Quellcodeverwaltung erfolgte über Github-Repository.

(https://github.com/AbdullahAltabbal/GameOfLife.git)

## 3 Projektentwurf

### 3.1 Ist-Analyse

Es gibt bereits bestehende online Webseiten, wo das Game of Life vorhanden ist.

Diese deckt jedoch nicht alle Anforderungen der Kunden ab, daher wird eine eigene Entwicklung erwünscht.

### 3.2 Design/Entwurf des Programms

* Ziel der Entwicklung ist eine GUI Anwendung, die das Game of Life visualisiert
* Die Entwicklung des Programms erfolgt in der Programmiersprache C#.
* Die gewünschten Funktionen des Programms sind im Anhang in den UserStories Datei dargestellt.
* Die Benutzeroberfläche ist in Anhang Überfläche Datei dargestellt.
* Unit Tests dient dazu, Die Anwendung Funktionen zu sichern.
* Die Speicherung der Datei erfolgt in XML-Dateien.
* Benutzeroberfläche geprüft.
* Eine Readme Datei wird bereitgestellt, die den Nutzer über die Funktionalität der Anwendung informiert.

## 4 Projektdurchführung

### 4.1 Implementierung/Realisierung

Die gesamte Anwendung ist in 2 Classen unterteilt.

Das Programm startet von den Classe ZubauZelleClass1.

Der Benutzer hat die 3 Optionen die Größe der Überfläche zu steuern, klein, mittel und groß.

Der Benutzer hat auch die Möglichkeit zufällige Zellen durch den Bauen Knopf generieren zu lassen

### 4.2 Tests/Qualitätssicherungsmaßnahmen

- zwei Testprotokolle sind im Anhang

## 5 Reflexion

Nach einer Weile von der zusammen Arbeit könnten wir gut unsere Ziele erreichen, die angegeben Zeit war dafür knapp, sodass wir nicht alle Wünsche des Benutzers erledigen könnten.

## 6 Quellen

* <https://fachinformatiker-anwendungsentwicklung.net/inhalte-der-projektdokumentation/>
* <https://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens>