31.5.2018

Florian Fechner

f\_fech03@uni-muenster.de

**Cells**

Eine Conway’s Game of Life Implementation

Inhaltsverzeichnis

[Vorwort 3](#_Toc514879370)

[Design der Software 4](#_Toc514879371)

[Verwendete Werkzeuge und Softwarepakete 4](#_Toc514879372)

[Electron 4](#_Toc514879373)

[ndarray 5](#_Toc514879374)

[Polymer 5](#_Toc514879375)

[Gestaltung der Benutzeroberfläche 6](#_Toc514879376)

[Window Frame 6](#_Toc514879377)

[Hauptmenü 6](#_Toc514879378)

[Sandbox Modus 7](#_Toc514879379)

[Tutorial 8](#_Toc514879380)

[Einstellungen 8](#_Toc514879381)

[Impelemtierung 9](#_Toc514879382)

[Spielfeld 9](#_Toc514879383)

[Zeitintervall 9](#_Toc514879384)

# Vorwort

Bei dem folgenden Dokument handelt es sich um eine deutsche Dokumentation der Software „Cells“, welche für die Code Competition von [it-talents.de](https://www.it-talents.de/) im Mai 2018 erstellt wurde.

# Design der Software

## Verwendete Werkzeuge und Softwarepakete

Bei der Implementierung von „Cells“ wurden eine Reihe von Frameworks und Softwarepaketen verwendet. Die Basis hierbei bildet das Softwarepaket [Node.js](https://nodejs.org/en/) sowie die beiden Paket-Management-Services [NPM](https://www.npmjs.com/) und [Bower](https://bower.io/).

Durch den Paket-Management-Service NPM werden Abhängigkeiten installiert, die sich über das Node.js Modulsystem integrieren lassen. Eine zentrale Abhängigkeit stellt hierbei das Modul [Electron](https://electronjs.org/) dar. Durch Electron ist es möglich eine Desktop Applikation zu erstellen, welche intern für die Darstellung der Benutzeroberfläche den Webbrowser [Chromium](https://www.chromium.org/) verwendet. Hierdurch wird es für den Ersteller einer Desktop Applikation möglich die komplette Darstellungslogik seiner Applikation in HTML, CSS und JavaScript zu implementieren. Des Weiteren wurde das NPM-Paket [rough.js](https://roughjs.com/) verwendet, um SVG-Elementen den Charme von selbstgezeichneten Bildern zu verleihen. Für effizientere Operationen auf mehrdimensionalen Arrays in JavaScript wurde das NPM-Paket [ndarray](https://github.com/scijs/ndarray) sowie dessen Untermodul [zeros](https://github.com/scijs/zeros) verwendet.

Durch den Paket-Management-Service Bower lassen sich sogenannte Web Components installieren. Bei Web Components handelt es sich um Elemente einer Website, welche bestimmte W3C-Standards erfüllt und dadurch ein gekapseltes System bilden. Hierdurch wird die technische Verschuldung bei der Verwendung unterschiedlicher Elemente möglichst klein gehalten. Nähere Informationen lassen sich auf <https://www.webcomponents.org/> nachlesen. Als Bibliothek zur praktischeren Erstellung von Web Components wurde [Polymer](https://www.polymer-project.org/) verwendet. Des Weiteren wurden einige Web Components der [„Paper Elements“ Collection](https://www.webcomponents.org/collection/PolymerElements/paper-elements) und der der [„Iron Elements“ Collection](https://www.webcomponents.org/collection/PolymerElements/iron-elements) verwendet. Die „Paper Elements“ Collection enthält visuelle Web Components, welche die von Google entworfene Designrichtlinien von [Material Design](https://material.io/) realisieren. Bei der „Iron Elements“ Collection handelt es sich um eine Sammlung von Basis-Elementen für den grundlegenden Aufbau einer Website.

### Electron

Die vom W3C definierten Web-Standards bilden in meinen Augen eines der besten visuellen Darstellungssysteme. Die Programmiersprache JavaScript erfreut sich außerdem großer Beliebtheit in der Entwickler Community, weshalb es eine überwältigende Fülle an Bibliotheken für eben diese Sprache gibt. Deshalb sehe ich Electron als eine gute Wahl an, wenn der Entwickler das Ziel hat eine Desktopapplikation zu implementieren, wenn sie nicht grade ein dreidimensionales Videospiel oder eine komplexe Simulation werden soll. In solchen Fällen kann die interpretierte Sprachnatur von JavaScript nämlich ein Problem darstellen. Eine Implementierung von Conway’s Game of Life sehe ich allerdings nicht als solch einen kritischen Fall an, wenn der Spieler nicht unbedingt wünscht auf ein 1000x1000 Feld zu spielen. Solch ein Spieler kann sich ja mit eines der anderen Einsendungen für die Coding Challange vergnügen, die mehr auf Performance ausgelegt ist ;)

### ndarray

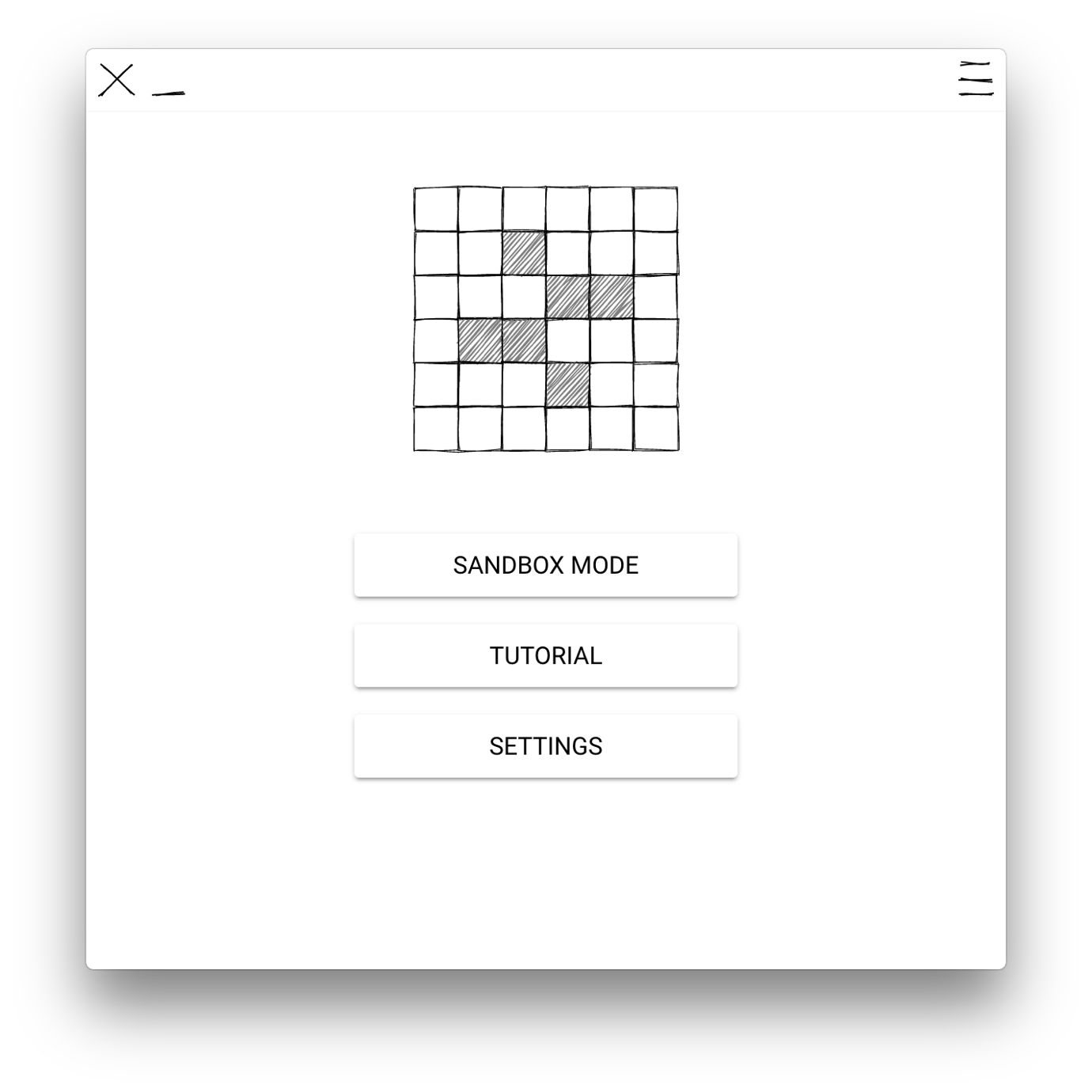
Bei Conway’s Game of Life werden Berechnungen auf einem zweidimensionalen Feld durchgeführt. Für diesen Fall bietet die Verwendung vom NPM-Modul ndarray einen Geschwindigkeitsvorteil.

### Polymer

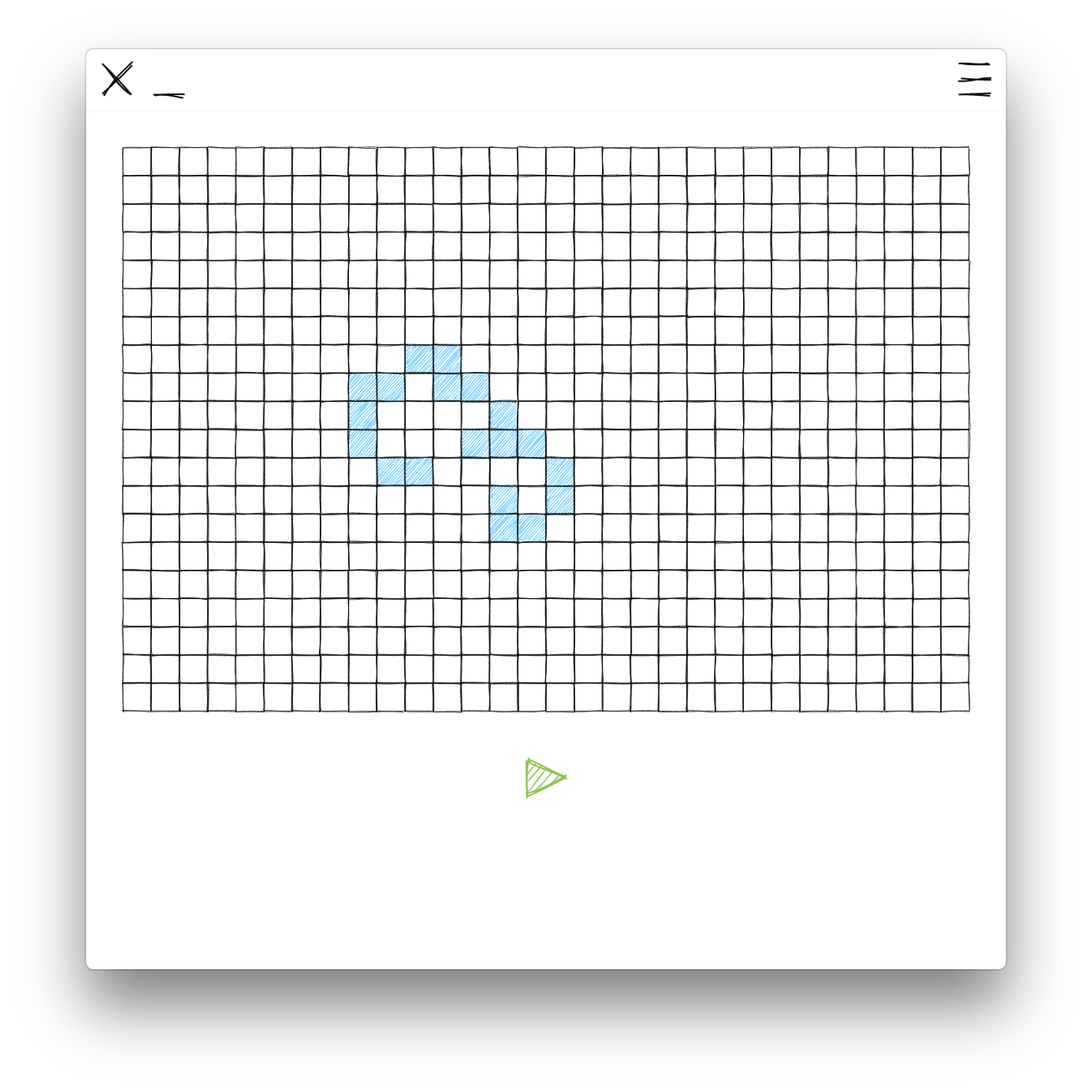
# Gestaltung der Benutzeroberfläche

## Window Frame

## Hauptmenü



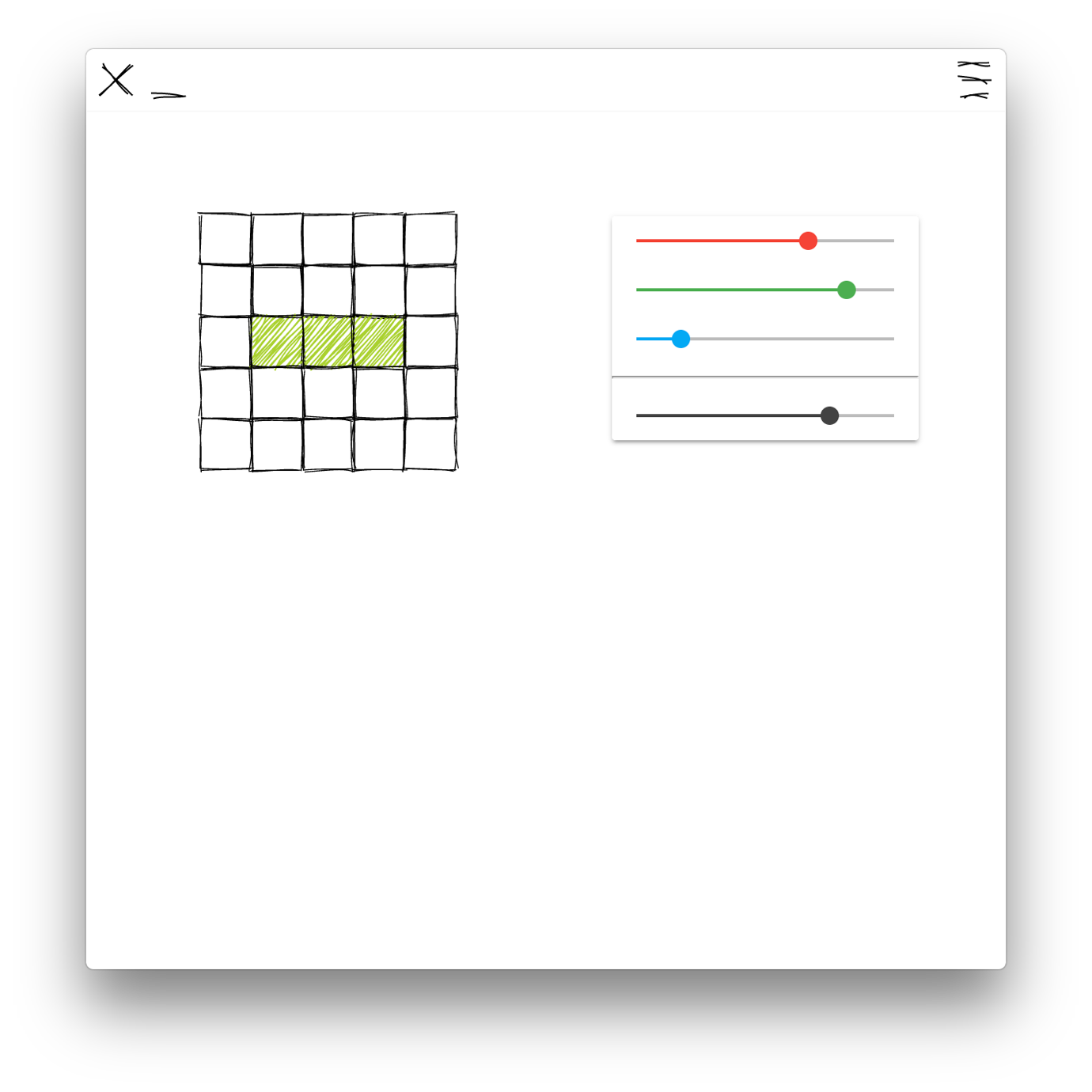
## Sandbox Modus



## Tutorial

## Einstellungen

* Zellfarbe
* Rauheit der Zeichnungen
* Breite der Linien
* Höhe und Breite des Spielfeldes
* Belebungs- und Sterberegeln



# Impelemtierung

## Spielfeld

## Zeitintervall