

default title

default author
facculty

00/00/0000

1 Konzept

-GOGOL (kurz für Game of Game of Life) ist eine Desktop Applikation für das berühmte nullspieler-spiel Game of Life, welches vom britischen mathematiker John Horton Conway im jahr 1970 entwickelt wurde. - das grundkonzept des spiels besteht aus einem zweidimensionalen, rechteckigen gitter, wobei jedes feld in diesem gitter einer zelle entspricht - eine zelle kann pro generationsschritt einen von zwei zuständen haben: lebend, tot - der zustand einer zelle ist abhängig von den 8 nachbarn die, die zelle umgeben - bei einem generationsübergang wird nun der zustand einer zelle bestimmt: - Eine tote Zelle mit genau drei lebenden Nachbarn wird in der Folgegeneration neu geboren. - Lebende Zellen mit weniger als zwei lebenden Nachbarn sterben in der Folgegeneration an Einsamkeit. - Eine lebende Zelle mit zwei oder drei lebenden Nachbarn bleibt in der Folgegeneration am Leben. - Lebende Zellen mit mehr als drei lebenden Nachbarn sterben in der Folgegeneration an Überbevölkerung.

Die Grundidee war das entwickeln einer soliden desktop anwendung die weitere nützliche bedienfunktionen hat. Darüber hinaus soll das programm, neben dem standart spiel noch weitere modi zu implementieren, welche eine abwandlung des "vanilla"GoL bieten. Zu anfangs geplant waren: standard conway colormerge - das verschmelzen von zellen mit farbeigenschaften colorwar - kampf von zellen mit "teamfarben"probability of life - random warscheinlichkeit bei der geburt von zellen

23 zuletzt sollte noch ein lokaler player versus player modus implementiert
24 werden, welcher der anwendung seinen eigentlichen namen verleiht, da so auf
25 dem game of life ein tatsächliches spiel entsteht

26 **2 Planung**

27 - Game of Life nach Conway - Modi: ColorMerge, ColorWar, PvP - Spei-
28 chern/Laden des Spiels, laden bestimmter Presets - Modus: Bio - nicht mehr
29 Deterministisch - nutzung von Presets, finden von Strukturen nicht möglich -
30 kaum Korrelation zu Game of Life - Modus: PvP - Exterminate - gewonnen
31 hätte der Spieler der mehr Zellen des Gegners vernichtet, unabhaengig von
32 seiner eigenen Zellzahl - verworfen, da kaum sinnvoll zu bestimmen ob eine
33 Zelle "vernichtet" wurde oder nur abgestorben ist. - Button changeGridsize -
34 Funktionalität ist implementiert - Wird in Zukunft vermutlich noch aufge-
35 nommen - bei Aktuller GUI Struktur kein Platz - Webscraping - geplant war
36 aus <http://conwaylife.appspot.com/library> die bekannten Presets auszulesen
37 - scheitert momentan an der Interpretation von Strukturen, die Zeilen mit
38 ausschließlich toten Zellen enthalten - die große Menge an Presets ist für den
39 User schwer übersichtlich darstellbar - Wird in Zukunft ggf. noch aufgenom-
40 men - Custom Ruleset - geplant war die Regeln für das entstehen/absterben
41 von Zellen anpassbar zu machen - wurde verworfen, da Presets nicht mehr
42 sinnvoll verwendbar wären - Effizientes Berechnen der Zellzustaende - wur-
43 de aus Zeitmangel und wegen ausreichend guter performance verworfen -
44 Ueberlegung war die SVM zeilenweise in threads zu berechnen - repaint in
45 stepForward erst am ende des Steps ausfuehren, statt nach jeder berechneten
46 Zelle - SpeedSlider - Geschwindigkeit ist logarhitmisch - intuitiver wäre eine
47 lineare Funktion - aufgrund von geringer Wichtigkeit vorerst nicht angepasst -
48 Workload von Meilenstein "Rainbowßtark unterschaezt - Unklarheiten in Me-
49 thodendefinitionen des Backend - Methoden des jeweils anderen wurden falsch
50 verstanden und inkorrekt verwendet - Workload von Meilenstein "PvPüebers-
51 chaetzt - grossteil der noetigen Funktionalitaet wurde in "Rainbow" bereits
52 implementiert - Kolja Hopfmann - Frontend: all - Listener: all - Backend:
53 Player, Commandhandling - Planung: erstellung GANTT diagramm - Jonas
54 Sander - Test: all - Backend: Cells, Library, Saver, Ruler, Referee - Beide -
55 Controller, Referee, Planung
56 - readme
57 - verwendete Programme - Eclipse Neon - IntelliJ IDEA Ultimate 2017.1.2

58 - gimp - texmaker - GanttProject 2.7.1 - Funktionsweise Java.AWT.Graphics
59 war komisch - Es wurde kein Framework verwendet - alle relevanten Klassen
60 und Methoden wurden selber entwickelt und implementiert - wir haben uns
61 die Definition von Game of Life angelesen - Implementierung, Datentypen
62 und Darstellung wurden selber erarbeitet