

Übungsblatt 4

Abgabe bis Dienstag, den **29. Mai 2018** um **12:00 Uhr**

Bei diesem Übungsblatt geht es darum, eine einfache Version von John Horton Conways *Game of Life* mittels ASCII Grafik zu implementieren.

Aufgabe 1 (2 Punkte)

Schreiben Sie eine Funktion `void initializeGame()`, die das Terminal für das Malen mit *ncurses* vorbereitet und die Variablen für das Spiel geeignet setzt. Wie auch schon beim Ü3, sollten alle Variablen, die nicht nur lokal in einer Funktion gebraucht werden, global sein.

Insbesondere sollten Sie *zwei* jeweils zwei-dimensionale Felder gleicher Größe benutzen, wobei eins davon den aktuellen Zustand des Spiels enthält und das andere den vorherigen bzw. nächsten Zustand, mehr dazu in Aufgabe 3. Die Größe sollte in beiden Dimensionen eine Konstante `MAX_GRID_SIZE` sein, die so gewählt sein sollte, dass die Abmessungen eines typischen Terminals (auch mit kleinem Font) darin Platz finden und die Animation hinreichend schnell läuft.

Aufgabe 2 (3 Punkte)

Schreiben Sie eine Funktion `void showState()`, die den aktuellen Zustand des Spiels anzeigt. Achten Sie *optional* darauf, dass jede Zelle möglichst quadratisch dargestellt wird, wie in der Vorlesung erklärt. Oben links sollte angezeigt werden, in der wievielten Iteration sich das Spiel befindet. Diese Funktion brauchen Sie nicht zu testen (und die von Aufgabe 1 auch nicht).

Aufgabe 3 (5 Punkte)

Schreiben Sie eine Funktion `void updateState()`, die einen Schritt des *Game Of Life* nach den üblichen (auf den Vorlesungsfolien beschriebenen) Regeln ausführt. Es sollte zwei globale Zeiger geben, eins für jedes Feld. So müssen Sie nur nach jedem Schritt die Zeiger tauschen und nicht das ganze Feld kopieren (was Sie auch auf gar keinen Fall tun sollten).

Lagern Sie den Code zur Berechnung der Anzahl der lebendigen Nachbarn einer Zelle in eine Funktion `int aliveNeighbors(int x, int y)` aus, die man dann auch getrennt testen kann und soll.

[nach kurzer Kontemplation über den Sinn des Lebens umblättern]

Aufgabe 4 (3 Punkte)

Schreiben Sie eine Funktion *int processUserInput(int key)*, die die folgende Nutzeraktionen verarbeitet und einen Wert zurückgibt, nachdem sich das Hauptprogramm dann richten kann. Wenn die *Leertaste* gedrückt wird, soll das Spiel automatisch weiterlaufen bzw. ein automatischer Lauf angehalten werden. Wenn die Taste *s* gedrückt wird (für *step*), soll genau ein Schritt ausgeführt werden. Mit der Taste *ESC* (Keycode 27) soll das Spiel beendet werden können. Bei einem Mausklick soll der Zustand der Zelle, auf die der Mauszeiger zeigt, invertiert werden.

Optional können Sie auch noch folgende Aktion unterstützen (es ist nicht schwer und schön anzusehen): Wenn die Taste *g* gedrückt wird, soll ein einfacher Glider an der Stelle des letzten Mausklicks eingefügt werden. Wenn die Taste *G* gedrückt wird, soll eine Gliderfabrik an der Stelle des letzten Mausklicks eingefügt werden. Eine Deklaration eines statischen Feldes mit den relativen Positionen der beiden letztgenannten Figuren finden Sie auf dem Wiki. Die Figuren sollen die betreffenden Zellen einfach überschrieben. Wenn die Taste *r* gedrückt wird, soll jede Zelle mit Wahrscheinlichkeit 0,2 zum Leben erweckt werden.

Aufgabe 5 (2 Punkte)

Schreiben Sie ein Hauptprogramm (in einer Datei *GameOfLifeMain.cpp*), mit der man das Spiel spielen kann. Die Zellen sollten dabei zu Beginn leer sein und dann mit den in Aufgabe 4 beschriebene Aktionen gesteuert werden können.

Aufgabe 6 (5 Punkte)

Machen Sie ihr Makefile vollständig generisch. Das heißt, es sollte ohne Anpassungen funktionieren, egal wie die Dateien, in denen der Code steht heißen. Sie dürfen dabei annehmen, dass alle Hauptprogramme in Dateien mit einem Namen von der Form *...Main.cpp* stehen und alle Testprogramme in Dateien mit einem Namen von der Form *...Test.cpp*. Ansonsten enden alle Codedateien mit *.cpp* und alle Headerdateien mit *.h*.

Das Makefile sollte wie gehabt die Eigenschaft haben, dass immer nur so viel neu kompiliert / gelinkt werden wie nötig. Sie dürfen allerdings der Einfachheit halber annehmen, dass bei einer Änderung in einer beliebigen Headerdatei alles neu kompiliert werden muss (was der Realität in den meisten Projekten auch sehr nahe kommt).

Laden Sie wie gehabt alle Code-Dateien und das Makefile in unser SVN hoch, in einem neuen Unterverzeichnis *blatt-04*. Es gelten weiterhin die 10 Gebote von der letzten Seite des Ü1 und nehmen Sie die Ratschläge Ihres Tutors / Ihrer Tutorin ernst.

Laden Sie wie gehabt auch eine Datei *erfahrungen.txt* hoch (im Unterordner *blatt-04*), in der sie kurz Ihre Erfahrungen mit dem Ü4 und der Vorlesung dazu beschreiben.

Überlegen Sie, ob Sie nicht vielleicht auch nur eine Figur in einem zellulären Automaten sind.