

Programmieren in C

Prof. Dr. Peter Thiemann
Hannes Saffrich
Sommersemester 2021

Universität Freiburg
Institut für Informatik

Übungsblatt 10

Abgabe: Montag, 05.07.2021, 9:00 Uhr morgens

In diesem Übungsblatt sollen Sie *Conway's Game of Life* mit der TUI-Library aus Blatt 6 implementieren.

Lesen Sie sich hierzu zunächst den zugehörigen Wikipedia-Artikel durch:

https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

Ein Demo-Video der Musterlösung inklusive Bonusaufgabe finden Sie hier:

<https://youtu.be/w5Qc0oo7RyU>

Der Aufbau dieses Übungsblattes ist analog zum Asteroiden-Spiel aus Blatt 7:

- in der `game_of_life.c` ist die `main`-Funktion vorgegeben;
- in der `game_of_life_lib.h` ist eine `GameState`-Struktur und Funktionsdeklarationen vorgegeben, die von der `main`-Funktion aufgerufen werden; und
- in der `bool_matrix.c/.h` finden Sie eine Implementierung einer einfachen Matrix von Werten mit Typ `bool`, die das Spielfeld aus lebenden (`true`) und toten (`false`) Zellen repräsentiert.

Ihre Aufgabe ist es die Funktionen in der `game_of_life_lib.c` zu implementieren.

Der vorgegebene Code darf *nicht* verändert werden. Es wird aber empfohlen weitere Funktionen zu definieren um die vorgegebenen Funktionen besser implementieren zu können.

Aufgabe 10.1 (Game of Life; 14 Punkte)

Implementieren Sie *Conway's Game of Life* und erfüllen Sie dabei die folgenden Kriterien:

- Wird 'q' gedrückt, so wird das Spiel verlassen
- Lebendige Zellen werden durch ein '#' gezeichnet, tote Zellen werden durch ein ' ' gezeichnet. In beiden Fällen mit weißer Text-Farbe und schwarzer Hintergrund-Farbe.
- Mit den Tasten "wasd" kann ein Cursor über das Spielfeld bewegt werden. Die aktuelle Position des Cursors (`gs->cursor_pos`) wird auf dem Spielfeld gezeichnet durch schwarze Text-Farbe und weiße Hintergrund-Farbe.

- (d) Wird 'n' gedrückt, so wechselt die Zelle an der Cursor-Position ihren Zustand: ist die Zelle lebendig, so ist sie danach tot, ist die Zelle tot, so ist sie danach lebendig.
- (e) Wird ' ' gedrückt, so wird die Simulation pausiert bzw. läuft weiter (`gs->simulation_active`).
- (f) Wird '+' bzw. '-' gedrückt so verschnellert bzw. verlangsamt sich die Simulation durch Verändern der `gs->speed` Variable.
- (g) Wird 'c' gedrückt, so sterben alle Zellen auf dem Spielfeld.
- (h) Die `update`-Funktion soll für alle Zellen einen Evolutionsschritt durchführen. Der aktuelle Zustand der Zellen steht in der Matrix `gs->m_cur`. Der neue Zustand der Zellen wird in die Matrix `gs->m_next` geschrieben. Nach einem Evolutionsschritt werden die Zeiger der beiden Matrizen vertauscht.¹

Für Zellen am Rand der Matrix soll ein "wrap-around"-Effekt entstehen: die Zelle an Position (0,0) hat also nicht nur die 3 Nachbar-Zellen (1,0), (1,1) und (0,1), sondern wie jede andere Zelle 8 Nachbar-Zellen. Hierzu werden die Zellen am gegenüberliegenden Spielfeldrand hinzugezogen: statt der nicht-existierenden Nachbar-Zelle bei (-1, -1) wird also die Zelle

`(gs->field_size.x - 1, gs->field_size.y - 1)`

verwendet.

Die `update`-Funktion soll nur alle `gs->speed` Zeitschritte einen neuen Evolutionsschritt durchführen und ansonsten nichts tun. Dies erlaubt es die Geschwindigkeit der Simulation zu kontrollieren.

- (i) Mit den Tasten 'u', 'i' und 'o' wird das aktuelle Spielfeld jeweils in den Dateien `save_u.gol`, `save_i.gol` und `save_o.gol` gespeichert.

Mit den Tasten 'U', 'I' und 'O' wird versucht ein gespeichertes Spielfeld jeweils aus den Dateien `save_u.gol`, `save_i.gol` und `save_o.gol` zu laden.

Der Inhalt der Dateien hat dabei die folgende Form:

```
5,4
.....
..#..
..#..
.....
```

In der ersten Zeile steht die Breite und Höhe des Spielfelds – in diesem Fall 5 Zellen breit und 4 Zellen hoch. Danach kommt der Inhalt des Spielfelds wobei lebende Zellen durch ein '#' und tote Zellen durch ein '.' repräsentiert werden.

¹Die Matrix `gs->m_next` dient nur als temporärer Speicherplatz während die `update`-Funktion ausgeführt wird und spielt ansonsten keine Rolle. Dies ist notwendig, da sonst beim Updaten einer Zelle, deren Nachbar-Zellen schon teilweise geupdated wären.

Die erste Zeile im Dateiformat des Spielfeldes erlaubt es beim Laden direkt eine Matrix mit der korrekten Größe zu initialisieren.

Hinweis. Beim Laden eines Spielfeldes bietet es sich an mit `fscanf` die erste Zeile auszulesen, und danach mit `fgetc` den Rest der Datei Zeile für Zeile auszulesen.

Für die Kriterien (a) bis (g) gibt es 2 Punkte und für (h) und (i) jeweils 6 Punkte.

Aufgabe 10.2 (Spawning Stuff; 4 **Bonus**-Punkte)

Erweitern Sie ihre Implementierung wie folgt:

- (a) Wird 'v' gedrückt, so aktiviert bzw. deaktiviert man einen Auswahlmodus (`gs->selection_active`).
- (b) Beim Aktivieren des Auswahlmodus' wird sich die aktuelle Cursor-Position gemerkt (`gs->selection_begin`).
- (c) Im Auswahlmodus wird nicht nur die Cursor-Position gezeichnet, sondern alle Zellen die sich innerhalb der Auswahl befinden.
- (d) Wird im Auswahlmodus mit 'u', 'i' oder 'o' gespeichert, so wird nur der ausgewählte Bereich des Spielfeldes in der Datei gespeichert.
- (e) Beim Laden eines Spielfeldes werden die Zellen beginnend an der aktuellen Cursor-Position eingefügt. Zellen, die außerhalb des Spielfeldes eingefügt werden würden, werden wahlweise ignoriert oder per "wrap-around" an der entgegengesetzten Spielfeld-Seite eingefügt.

Dies erlaubt es z.B. einen Glider zu speichern und während der Simulation per Tastendruck neue Glider zu spawnen ohne den Rest des Spielfeldes zu löschen.

Stellen Sie sicher, dass das Speichern auch dann korrekt funktioniert, wenn die Koordinaten der Endposition der Auswahl kleiner sind wie die der Startposition!

Aufgabe 10.3 (Erfahrungen; 2 Punkte)

Notieren Sie Ihre Erfahrungen mit diesem Übungsblatt in der Datei `erfahrungen.txt` (benötigter Zeitaufwand, Probleme, Bezug zur Vorlesung, Interessantes, etc.).

Der Zeitaufwand *muss* dabei in der ersten Zeile und in exakt dem folgenden Format notiert werden, da wir sonst nicht automatisiert eine Statistik erheben können:

Zeitaufwand: 3:30

<...Andere Erfahrungen...>

Die Angabe 3:30 steht hier für 3 Stunden und 30 Minuten.