

Übungsblatt 5

Abgabe bis Dienstag, den **31. Mai 2022** um **12:00 Uhr**

Aufgabe 1 (15 Punkte)

Machen Sie den Code vom letzten Übungsblatt objekt-orientiert, so wie es in der Vorlesung für eine vereinfachte Version des Spiels *Snake* vorgemacht wurde. Sie können dabei von Ihrem eigenem Code für das Ü4 ausgehen (sofern er korrekt war oder Sie ihn für dieses Übungsblatt korrekt machen) oder von der Musterlösung.

Sie können und sollen dabei die Klassen *TerminalManager* und *UserInput* aus der Vorlesung wiederverwenden. Die ganze Logik des Game of Life sollte in einer Klasse *GameOfLife* stehen und die Implementierung dieser Klasse sollte ganz ohne *ncurses* auskommen, sowohl die *.h* Datei als auch die *.cpp* Datei. Das heißt, sämtliche Funktionalität für die Ausgabe auf den Bildschirm und das Lesen der Eingabe (Tastatur oder Maus) sollte über die Klassen *TerminalManager* und *UserInput* geschehen. Es soll keine globalen Funktionen oder Variablen mehr geben.

Es sollte, analog zum letzten Übungsblatt, jede Funktion der Klasse *GameOfLife* außer *play* entweder Spiellogik implementieren oder mit dem Terminal interagieren (Eingabe oder Ausgabe, nun über die Klasse *TerminalManager*). Für die Funktionen, die nur Spiellogik implementieren, soll wie gehabt ein Unit Test geschrieben werden. Für alle anderen Funktionen sollen keine Tests geschrieben werden.

Die Funktion *main* sollte wie in der Vorlesung nur aus drei Zeilen bestehen: einer geeigneten Deklaration eines *TerminalManager* Objektes, einer geeigneten Deklaration eines *GameOfLife* Objektes und eines geeigneten Aufrufs der *play* Funktion der Klasse *GameOfLife*.

Aufgabe 2 (5 Punkte)

Machen Sie Ihr *Makefile* ganz generisch. Das heißt, in dem Makefile sollte sich kein Hinweis auf konkrete Dateinamen mehr befinden. Wie gehabt sollte bei einer Dateiänderung immer nur genauso viel kompiliert werden wie nötig (bei einer Änderung in einer Headerdatei darf alles neu kompiliert werden). Sie können weiterhin davon ausgehen, dass es nur eine Datei *...Main.cpp* und eine Datei *...Test.cpp* gibt, die denselben Basisnamen (der Teil von dem ...) haben.

Tipp: Erweitern Sie in einem ersten Schritt das *Makefile* vom letzten Übungsblatt so, dass es für Aufgabe 1 funktioniert und sämtliche Hinweise auf die Dateinamen in Makefile-Variablen *MAIN_BINARY*, *TEST_BINARY* und *OBJECTS* stehen. Setzen Sie diese Variablen dann automatisch mittels der bereits auf den Folien von Vorlesung 3 aufgeführten Makefile-Funktionen.

Aufgabe 2 (optional)

Erweitern Sie Ihr Spiel nach Belieben, siehe die Anregungen dazu vom letzten Übungsblatt.

Aufgabe 3 (optional)

Experimentieren Sie mit einer Funktion *godMode*, die den Wesen in Ihrem Game of Life Bewusstsein vorgaukelt und sie glauben lässt, sie seien etwas Besonderes, wodurch sie sich selbst und alles um sie herum sehr ernst nehmen, worüber man sich von außen betrachtet nur wundern kann. Implementieren Sie dann auch ein paar Funktionen, mit denen Sie per Knopfdruck verschiedene Sachen entstehen lassen können, z.B. ein Virus oder narzisstische Präsidenten. Aktivieren Sie die Funktionen nach Belieben und schauen Sie, wie die Kästchen plötzlich alle durchdrehen.

Laden Sie alle Code-Dateien und das Makefile in unser SVN hoch, in einem neuen Unterverzeichnis *blatt-05*. Es gelten weiterhin die 10 Gebote und nehmen Sie die Ratschläge Ihrer Tutorin und Ihres Tutors ernst. Und für die wenigen, die es betrifft: Bitte nicht schummeln, wir merken es. So gut zu plagieren, dass wir es nicht merken, ist ungefähr so schwer wie die Aufgaben zu lösen.

Laden Sie wie gehabt auch eine Datei *erfahrungen.txt* hoch (im Unterordner *blatt-05*), in der Sie kurz Ihre Erfahrungen mit dem Ü5 und der Vorlesung dazu beschreiben.

Sie wissen jetzt, dass Sie auch nur eine Figur in einem zellulären Automaten sind. Was hat das für Konsequenzen für Ihr Leben?