

Programmieren in C

Prof. Dr. Peter Thiemann
Hannes Saffrich
Sommersemester 2021

Universität Freiburg
Institut für Informatik

Projekt

Abgabe: Montag, 09.08.2021, 9:00 Uhr morgens

Minesweeper ist ein simples Computerspiel, welches seit Windows 3.1 in allen Windows-Versionen vorinstalliert ist. Siehe hierzu:

<https://de.wikipedia.org/wiki/Minesweeper>

Ein Demo-Video des fertig implementierten Projekts finden Sie hier:

<https://youtu.be/B81dkgp0or4>

Sie dürfen in diesem Projekt beliebigen Code aus der Vorlesung und den Musterlösungen der Übungsblätter kopieren und abändern. Eine Quellenangabe ist nicht notwendig.

Sie dürfen annehmen, dass das Terminal mindestens eine bestimmte Höhe und Breite hat und sich diese bei der Ausführung auch nicht mehr verändert. Sie müssen aber beim Programmstart abfragen, ob das Terminal groß genug ist und ansonsten das Program mit einer informativen Fehlermeldung beenden.

Sie dürfen von den vorgegebenen Farben, Zeichen und dem visuellem Design im Demo-Video abweichen, aber unterschiedliche Zeichen und Farben müssen weiterhin erkennbar unterschiedlich sein.

Aufgabe 12.1 (Basic Game; 38 Punkte)

Implementieren Sie Minesweeper mit Hilfe der TUI-Library. Erfüllen Sie dabei mindestens die folgenden Kriterien:

- Zu Beginn des Spiels werden die Minen zufällig verteilt und alle Felder sind verdeckt. Das Spielfeld besteht stets aus 15×15 Feldern und jedes Feld hat eine Wahrscheinlichkeit von 15% eine Mine zu enthalten.
- Mit den Tasten **wasd** kann ein Cursor auf dem Spielfeld bewegt werden.
- Drücken der Leertaste deckt das Feld an der aktuellen Cursor-Position auf.
 - Enthält das Feld eine Mine, so ist das Spiel verloren.
 - Enthält das Feld keine Mine und alle Nachbarfelder enthalten ebenfalls keine Mine, so nennen wir das Feld ein *inneres Feld*. Wird ein inneres Feld aufgedeckt, so werden auch alle direkt und indirekt benachbarten inneren Felder aufgedeckt sowie deren direkte Nachbarfelder. Wurde das letzte minenlose Feld aufgedeckt, so ist das Spiel gewonnen.

- Wurde das Spiel gewonnen oder verloren, so werden alle Felder aufgedeckt und eine entsprechende Ausgabe angezeigt. Das Spielfeld kann dann nicht mehr verändert aber weiterhin betrachtet werden.
- Mit der Taste **f** wird eine Fahne an der aktuellen Cursor-Position gesetzt bzw. wieder entfernt. Ein Feld mit einer Fahne kann nicht aufgedeckt werden, bevor die Fahne wieder entfernt wurde.
- Die Felder werden wie folgt gezeichnet:
 - verdeckte Felder mit Fahne werden durch 'X' dargestellt;
 - verdeckte Felder ohne Fahne werden durch '.' dargestellt;
 - aufgedeckte Felder mit Fahne werden durch ein rotes oder grünes 'X' dargestellt, jenachdem ob sich in dem Feld eine Mine befindet;
 - aufgedeckte innere Felder werden durch ' ' dargestellt;
 - aufgedeckte Felder mit Minen und ohne Fahne werden durch '*' dargestellt; und
 - alle anderen aufgedeckten Felder werden durch die Anzahl der benachbarten Minen dargestellt – also den Ziffern '1' bis '8'.
- Mit der Taste **q** kann das Spiel abgebrochen/beendet werden.

Aufgabe 12.2 (Menü & Einstellungen; 20 Punkte)

Erweitern Sie das Spiel, sodass bei Programmstart ein Menü angezeigt wird, das mindestens folgende Auswahlmöglichkeiten bietet:

- *New Game*. Startet ein neues Spiel.
- *Settings*. Erlaubt es die Spielfeldgröße und die Minen-Wahrscheinlichkeit zu konfigurieren.
- *Exit*. Schließt das Programm.

Die Einstellungen, wie Spielfeldgröße und Minen-Wahrscheinlichkeit, sollen persistent sein – also zwischen mehreren Programmaufrufen erhalten bleiben.

Das Menü soll sich wie im Demo-Video verhalten.

Um das Programm als außenstehende Person einfach benutzen zu können, sollen die Tastenbelegungen zur Steuerung des Menüs und des Spiels auf dem Bildschirm angezeigt werden.

Aufgabe 12.3 (Highscores; 20 Punkte)

Erweitern Sie das Spiel um eine Top 10 Highscore-Tabelle und eine Zeitanzeige.

Diese soll wie im Demo-Video gezeigt über das Menü erreichbar sein und ist wie die Einstellungen persistent.

In der Highscore-Tabelle sollen die benötigte Zeit, die Größe des Spielfeldes, die Minen-Wahrscheinlichkeit und die Punktzahl zu sehen sein. Die Punkte ergeben sich durch folgende Formel:

```
points = width * height * probability * (1000.0 / time_in_seconds)
```

Die Highscore-Tabelle ist nach Punkten sortiert und enthält maximal die 10 besten Spielergebnisse.

Während des Spielens soll die bisher benötigte Zeit angezeigt werden (siehe Demo-Video). Die Zeit wird nicht mehr weiter gezählt, sobald das Spiel gewonnen oder verloren wurde.

Funktionen um die aktuelle Zeit abzufragen finden Sie in folgendem Abschnitt der Standardbibliothek:

<https://en.cppreference.com/w/c/chrono>

Aufgabe 12.4 (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.