

Übungsblatt 3

Abgabe bis Dienstag, den **10. Mai** um **12:00 Uhr**

Aufgabe 1 (15 Punkte)

Schreiben Sie ein Programm, das folgende Teilfunktionalität des Spiels *Tetris*, wie in der Vorlesung vorgestellt, realisiert:

1. Es gibt nur eine Sorte von Tetrominos. Suchen Sie sich dazu von den sieben möglichen eine aus (außer das Quadrat).
2. Zu jedem Zeitpunkt bewegt sich genau ein Tetromino auf dem Bildschirm. Es fällt von selber. Es kann mit den Pfeiltasten nach links und rechts bewegt werden. Mit Pfeil nach unten fällt es schneller als normal. Zum Drehen, siehe Aufgabe 3. Achten Sie darauf, dass das Tetromino nicht zu schnell fällt, aber trotzdem ohne spürbare Verzögerung auf einen Tastendruck reagiert wird.
3. Wenn das aktuelle Tetromino unten ankommt, soll es dort in rot gemalt werden. Im nächsten Zeitschritt soll dann von oben ein neues Tetromino herunterfallen.
4. Achten Sie darauf, dass die Tetrominos nie über den Rand des „Spielfeldes“ hinausgehen: die Standardmaße für Tetris sind 20 Zeilen und 10 Spalten. Optional können Sie einen Rand um das Spielfeld malen.
5. Alle Tetrominos fallen einfach bis ganz nach unten, ganz egal welche Tetrominos dort vorher schon liegen geblieben sind. Die Struktur der liegen gebliebenen vorherigen Tetrominos soll erst auf dem nächsten Übungsblatt berücksichtigt werden (dafür braucht man Felder und Zeiger).

Orientieren Sie sich an dem Programm aus der Vorlesung. Eine sehr ähnliche Struktur funktioniert auch für dieses Übungsblatt. Das Übungsblatt lässt sich mit relativ wenig zusätzlichem Code (gegenüber dem, was schon der Vorlesung gemacht wurde) realisieren.

In jedem Fall sollte ihr Programm sinnvoll in *.h* und *.cpp* Dateien aufgeteilt sein (*Tetris.h*, *Tetris.cpp*, *TetrisMain.cpp*). Außerdem soll der Code für das Malen in eigenen Funktionen stehen. Für diese Funktionen brauchen Sie keinen Unit Test zu schreiben.

Ansonsten gelten alle Hinweise auf der Rückseite vom ersten Übungsblatt unverändert.

Aufgabe 2 (5 Punkte)

Schreiben Sie ein Makefile, das so weit wie möglich Patterns und automatische Variablen einsetzt, wie in der Vorlesung erklärt. Funktionen (wie *wildcard*, *basename*, etc.) können Sie einsetzen, müssen es aber nicht. Sie können dabei davon ausgehen, dass es einen Basisnamen *<name>* gibt, und die Funktionen in *<name>.cpp* stehen, das Hauptprogramm in *<name>Main.cpp* und die Tests in *<name>Test.cpp*. Es soll dabei wie gehabt unter allen Umständen immer nur so viel neu kompiliert / gelinkt werden wie nötig.

Aufgabe 3 (freiwillig)

Wenn Sie möchten, implementieren Sie auch schon das *Drehen* eines Tetrominos (wenn die Taste Pfeil nach oben gedrückt wird). Das wird auf jeden Fall eine Aufgabe auf dem nächsten Übungsblatt. Wenn Sie es jetzt schon machen, brauchen Sie es dann nicht mehr machen bzw. haben dann mehr Zeit für andere Features.

Sie können außerdem eine Zelle eines Tetrominos mit mehreren Zeichen malen (zum Beispiel: 5 Zeichen breit und 3 Zeichen hoch), damit die Zellen quadratischer aussehen.

Sie können außerdem überlegen, wie man das Flackern (das durch schnell aufeinanderfolgendes Zeichnen und Löschen von Zeichen auf dem Bildschirm hervorgerufen wird) reduzieren kann.

Laden Sie wie gehabt alle Code-Dateien und das Makefile in unser SVN hoch, in einem neuen Unterverzeichnis *blatt-03*. Wie gehabt, soll nach dem letzten *svn commit* vor Abgabetermin auf Jenkins alles ohne Fehler durchlaufen.

Laden Sie dort wie gehabt auch eine Datei *erfahrungen.txt* in unser SVN hoch. Beschreiben Sie dort insbesondere, ob es Schwierigkeiten mit dem 3. Übungsblatt und der Vorlesung dazu gab, und wie viel Zeit Sie etwa benötigt haben.

Warum vergeht die Zeit langsamer, wenn man sich mit sehr großer Geschwindigkeit fortbewegt, und wie fühlt sich das an?