

## Übungsblatt 3

Abgabe bis Dienstag, den **7. Mai 2024** um **12:00 Uhr**

Bei diesem Übungsblatt geht es darum, eine stark vereinfachte Version des Spiels *Snake* zu implementieren. Dabei bewegt sich ein „Pixel“ (im wirklichen Spiel ist es eine Kette von mehreren Pixeln, die an eine Schlange erinnert) mit langsam steigender Geschwindigkeit über den Bildschirm und kann mithilfe der Pfeiltasten gesteuert werden. Das Spiel ist zu Ende, wenn der Pixel an den Bildschirmrand stößt oder die *Escape* Taste gedrückt wird.

### Aufgabe 1 (10 Punkte)

Schreiben Sie Dateien *Snake.h* und *Snake.cpp* mit den folgenden Funktionen. Für einige der Funktionen wurde bereits ganz oder teilweise Code in der Vorlesung geschrieben und über das SVN bereitgestellt. Die Aufgabe ist deswegen weniger Arbeit, als es auf den ersten Blick den Anschein haben mag.

Benutzen Sie für die folgenden Werte globale Variablen: die Dimensionen des Spielfelds, die aktuelle Richtung, und die aktuelle Position des Pixels auf dem Bildschirm.

*void initTerminal()*: Initialisierung des Terminals und der Spielfelddimensionen mittels *ncurses*. Letztere sollten in globalen Variablen gespeichert werden.

*void initGame()*: Initialisierung des Spiels (insbesondere: Setzen der Anfangsposition des Pixels und der Anfangsrichtung).

*void drawPixel(int row, int column, int color)*: Malen eines Pixels an die gegebene Position mit der gegebenen Farbe. Ein Pixel soll dabei zwei Zeichen breit sein, damit er hinreichend quadratisch aussieht.

*void drawBorder(int color)*: Malen des Spielfeldrands in der gegebenen Farbe (mittels *drawPixel*).

*void drawSnake(int color)*: Malen der „Schlange“ an der aktuellen Position in der gegebenen Farbe (mittel *drawPixel*).

*bool collidesWithBorder()*: Berechnen, ob die „Schlange“ an den Rand gestoßen ist.

*void moveSnake()*: Bewegen des Pixels in die aktuelle Richtung um genau einen Pixel.

[Mit den Pfeiltasten umblättern ohne gegen den Rand zu stoßen]

*bool handleKey(int key)*: Verarbeiten des gegebenen Keycodes. Bei einer Pfeiltaste soll die Richtung entsprechend angepasst und *false* zurückgegeben werden. Eine Bewegung in die Gegenrichtung, zum Beispiel nach links wenn der Pixel sich gerade nach rechts bewegt, sollte nicht möglich sein (der Tastendruck sollte dann einfach keinen Effekt haben). Bei der Taste *ESC* soll *true* zurückgegeben werden, sonst *false*.

Schreiben Sie für jede Funktion, die nichts auf den Bildschirm malt, einen Unit Test gemäß dem 6. Gebot. Diese Tests sollten in einer Datei *SnakeTest.cpp* stehen. Benutzen Sie für die Tests kein *ncurses*. Setzen Sie die Höhe und Breite des Bildschirms „von Hand“.

### Aufgabe 2 (5 Punkte)

Schreiben Sie eine Datei *SnakeMain.cpp* mit einer *main* Funktion, die das Spiel auf die eingangs beschriebene Weise realisiert. Benutzen Sie dabei eine Variable *speed* für die aktuelle Geschwindigkeit in Pixel pro Sekunde und eine Variable *acceleration* für die Beschleunigung in Pixel pro Sekunde Quadrat. Wenn das Spiel zu Ende ist, sollte der Bildschirminhalt nicht sofort verschwinden, sondern erst nach (erneutem) Drücken der *Escape* Taste. Der Code für die *main* Funktion sollte relativ kurz sein und vor allem die Funktionen benutzen, die Sie für Aufgabe 1 geschrieben haben.

### Aufgabe 3 (0 Punkte)

Wenn Sie Spaß an der Aufgabe haben, können Sie das Programm auf vielerlei Arten erweitern. Sie können beispielsweise eine Schlange aus mehreren Gliedern bauen oder Sie können die Zeit anzeigen, die bereits vergangen ist. Lassen Sie Ihrer Fantasie freien Lauf.

### Aufgabe 4 (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 schon für das Übungsblatt 2 unter allen Umständen immer nur so viel neu kompiliert / gelinkt werden wie nötig.

Laden Sie wie gehabt alle Code-Dateien und das Makefile in unser SVN hoch, in einem neuen Unterverzeichnis *blatt-03*. Es gelten weiterhin die 10 Gebote auf dem Wiki und bitte nehmen Sie die Ratschläge Ihres Tutors oder Ihrer Tutorin ernst. Laden Sie wie gehabt auch eine Datei *erfahrungen.txt* hoch (im Unterordner *blatt-03*), in der Sie kurz Ihre Erfahrungen mit dem Ü3 und der Vorlesung dazu beschreiben.