HOCHSCHULE
HANNOVER
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS
–
Fakultät IV
Wirtschaft und
Informatik

Übungen zur Vorlesung Computergrafik 2

Wintersemester 2020/2021

Prof. Dr. Ingo Ginkel

Aufgabenblatt 1

19.10.2020

Hinweise:

- Abgabe der Aufgaben bis zum 27.11.2020.
- Support bezieht sich nur auf Probleme mit der VM. Alles weitere, wie Installation auf dem eigenen Notebook, kann gerne gemacht werden, wird aber von mir nicht aktiv unterstützt.
- Sie können gerne - wie in CG1 - die QtCreator IDE benutzen, dazu müssen Sie nur die `.pro` Datei importieren (in der VM schon gemacht und lauffähig).

Aufgabe 1: Kennenlernen der zur Verfügung gestellten Funktionalität (0,5 Punkte)

Das kleine Programm ist in der Lage Bild-Dateien einzulesen (jpeg und andere Formate) und anzuzeigen (im Hauptfenster rechts). Links daneben gibt es einen Bereich in dem entsprechende Buttons, Slider, Checkboxen etc. erstellt werden können um die anzuwendenden Algorithmen zu steuern. Es ist sinnvoll hier jeweils ein Tab für jede Aufgabe zu verwenden. Wie Sie dies genau ausgestalten ist aber ihnen überlassen, letztlich können Sie das optisch so machen wie sie mögen.
In den Übungsaufgaben sollen inder Regel ein oder mehrere Algorithmen zur Bildverarbeitung implementiert werden und die dazu gehörige GUI-Steuerung erstellt werden. In diesem Übungsblatt soll die Funktionsweise des vorhandenen Codes soweit notwendig verstanden werden. Es gibt dabei nur wenige Stellen an denen sie aktiv eingreifen müssen. Ich habe versucht das so zu sortieren, dass alles was nicht notwendig ist zum Verständnis irgendwie nach hinten verschwindet. Das eingelesene Bild wird als Instanz der Klasse `QImage` erstellt und ist mittels dem Pointer `image` zugreifbar.
Es gibt im Wesentlichen 3 Stellen im Code an denen Sie aktiv werden müssen.

- Zum einen müssen alle GUI-Elemente, die Sie erzeugen als private Member deklariert werden. Das passiert in der Datei `imageviewer-qt5.h`. Die Stelle ist mit einem Kommentar markiert. Einige Beispiel-Buttons und ein Slider sind dort schon vorhanden und können als Vorlage genutzt werden.
- Die zweite Stelle ist die Instanziierung der GUI und die Verknüfung der Elemente mit der Aktion (d.h. dem Funktionsaufruf) der passieren soll wenn z.B. ein Button gedrückt wurde. Dies passiert in `imageviewer-qt5.cpp` in der Methode `void generateControlPanels()`. Hier werden 2 Tabs angelegt und mit einigen GUI Elementen sowie einem Layout befüllt. Sie können im Verlaufe des Semester mit dieser Vorlage weitere GUI-Funktionalität nach und nach ergänzen. Hier ist auch gezeigt wie man z.B. das Drücken eines Buttons mit der dann auzurufenden Methode mit dem SINGAL-SLOT Konzept von Qt bewerkstelligt.

- Die letzte größere Baustelle ist dann die Methode in der der eigentliche Algorithmus implementiert werden soll. Als Beispiel ist die Methode `void applyExampleAlgorithm()` in `imageviewer-qt5.cpp` vorhanden. Hier wird dann die jeweilige Funktionalität implementiert. Es macht natürlich Sinn sich pro Algorithmus eine solche Methode zu erstellen und mit den GUI-Elementen zu verbinden und besser nicht alles in eine Methode zu packen. In dem Beispiel wied ein schwarzer Strich diagonal durch das Bild gezeichnet. Ebenso wird hier die Log-Funktionalität benutzt. Das funktioniert wie `std::cout` nur statt `cout << \Mein Text\` schreiben Sie `log << \Mein Text\`. Dann wird der entsprechende Text in die Datei `logFile.txt` geschrieben zusätzlich in dem Konsolenfenster im Programm angezeigt wenn Sie `renewLogging()` aufrufen.

a) Machen Sie sich mit dem Programm vertraut und recherchieren Sie die Signal-Slot Funktionalität von Qt (es gibt genug Online-Quellen).

b) Ergänzen sie die GUI-Elemente durch einige eigene Elemente. In der Online-Doku zu Qt finden Sie viele weitere Möglichkeiten wie z.B. Checkboxen, Radio-Buttons, Slider etc. Probieren Sie möglichst viele davon aus uns verknüfen Sie die Objekte mit jeweils einer Methode die dann aufgerufen wird. Hier gibt es jeweils verschiedene Möglichkeiten wann das passieren soll, etwa bei Click auf das GUI-Element, oder nur wenn sich der Wert (z.B. bei einem Slider) ändert etc. probieren Sie hier einiges aus, so dass Sie einen Fundus an Bedienmöglichkeiten haben um später ihre Programme zu steuern.

c) Das Setzen einer Pixelfarbe ist in Qt etwas gewöhnungsbedürftig. Suchen Sie auch hier in Doku heraus, welche Möglichkeiten es gibt und verändern sie den Beispiel-Code so, dass statt des schwarzen Linie ein rotes Kreuz gezeichnet wird, dessen Größe Sie mit einem Slider von einem Pixel bis zur Bildbreite bzw. Bildhöhe erhöhen und auch wieder verringern können.

Bei dem Verringern der Strichbreite soll der ursprüngliche Inhalt des Bildes an den entsprechenden Pixel-Positionen wieder sichtbar werden. Überlegen Sie sich einen Mechanismus, der dies möglichst einfach gewährleistet.

Hinweis: Der Sinn der Aufgabe ist nicht das rote Kreuz (es ist egal ob das senkrecht oder diagonal oder sonstwie ist...), sondern die entsprechend benötigten Werte aus dem Slider und dem Bild abzufragen und die Farbe zu setzen, sowie die Bildinformation beim "Überzeichnen" nicht zu verlieren. Auch muss hier aus dem Sliderwert, der glaube ich standardmäßig von 0 bis 100 geht, die entsprechende Pixelbreite des Kreuzes berechnet werden, da bei verschieden großen Bilden mit dem gleichen Slider immer von 0 bis Bildbreite skaliert werden soll.