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

Installation und GUI

Installation:

- Sie können entweder mit einer Standard-Ubuntu-Installation oder der Virtuellen Maschine unseres IT-Teams (`https://sync.academiccloud.de/index.php/s/1mvVDsghUu2tg2W`) starten und einige Pakete dazu installieren.
- Installieren Sie mit dem Befehl `sudo apt-get install` folgende Libs: `g++`, `qt5-default` und `qtcreator`, da kommen natürlich noch einige Abhängigkeiten mit.
- Entpacken Sie `CG2Uebung.zip` an einen Ort ihrer Wahl und benennen Sie den Projektordner gemäß den Personen in ihrer Gruppe um. Füllen Sie auch die Datei `README.MD` aus.
- Öffnen Sie die Datei `imageviewer-qt5.pro` in QtCreator (ganz passable IDE, gut geeignet für Qt-Programme) Wählen sie die Option Desktop und klicken Sie auf Projekt konfigurieren, dann auf Play. Sie sollten ein kleines GUI mit verschiedenen Tabs auf der linken Seite und einem grauen Bereich auf der rechten Seite sehen, in dem später die Bilder angezeigt werden. Wenn das nicht funktioniert stimmt noch etwas an der Installation nicht.
- Um einige Fehlermeldungen zu vermeiden und die Autovervollständigen-Funktion in der IDE zu nutzen müssen Sie eine Einstellung vornehmen: Dazu im Menü Hilfe→plugins im Abschnitt C++ die Auswahl 'ClangCodeModel' entfernen und dann QtCreator neu starten.
- Damit wären Sie arbeitsbereit.

Fehlerbehebung:

Falls das Konfigurieren des QTCreator Projektes wie oben beschrieben nicht funktioniert können Sie folgendes probieren:

- Übersetzen Sie den Code zunächst *von Hand* durch `qmake && make clean && make` und starten Sie das Programm aus der Konsole. Das GUI sollte jetzt erscheinen.
- Öffnen Sie die Datei `imageviewer-qt5.pro` in QtCreator. Wählen sie dann aber die Option Desktop NICHT sondern die weiter unten stehende Option (Qt 5.12.8 oder ähnlich) in den Varianten Debug und Release. Klicken Sie auf Projekt konfigurieren, dann auf Play. Es sollte jetzt das Fester mit den GUI Elementen öffnen.

Überblick GUI

Das kleine Programm ist in der Lage Bild-Dateien einzulesen (jpeg und andere Formate) und anzuzeigen (im Hauptfenster rechts). Links daneben gibt es für jede Aufgabe vorgefertigte Buttons, Slider, Checkboxen etc., um die anzuwendenden Algorithmen zu steuern. Im ersten Tab 'Info' ist eine Liste von Shortcuts enthalten um Bilder zu laden, reset auszuführen, Zoom etc.

In den Übungsaufgaben sollen in der Regel ein oder mehrere Algorithmen zur Bildverarbeitung implementiert werden und die dazu gehörige GUI-Steuerung genutzt werden. Zunächst soll die Funktionsweise des vorhandenen Codes soweit notwendig verstanden werden. Es gibt dabei nur wenige Stellen an denen sie aktiv eingreifen müssen.

- Das input Bild heißt in allen Übungsaufgaben `image` und wird immer als Pointer `QImage * image` übergeben .
- Zusätzlich gibt es eine globale Instanz `QImage * backupimage` die für das resetten von Bildern auf das ursprüngliche Bild benutzt wird und auch in weiteren Aufgaben genutzt werden kann.
- Das workingImage (in `examples2.cpp/h`) ist ein Beispiel für zwei Dinge, erstens wie man sich eine zusätzliche `QImage` Instanz anlegen kann. Für die Filter braucht man ja schon zwei Bild-Instanzen. Das geht mit `workingImage = new QImage(*backupImage)`. Zweitens ist das workingImage ein Beispiel dafür, wie man sich ein QImage speichern kann und weiterhin gespeichert bleibt nach dem Eventaufruf und Methodenscope. Dafür muss man in die header Datei sich das Objekt anlegen: `inline QImage* workingImage;`. Dann muss man allerdings aufpassen, dass man den Speicher wieder frei gibt. Dafür gibt es eine `freeMemory.cpp` mit der Methode `cg2::freeMemory()`, welche im Destruktor der Imageviewer App aufgerufen wird. Die Methode `cg2::freeMemory()` kann also beliebig erweitert werden, und auch nach Belieben aufgerufen werden.
- Für jedes Aufgabenblatt gibt es je eine eigene Header- und Source-Datei, in denen die zu implementierende Funktionalität eingefügt werden soll. Die Interaktion mit den GUI Elementen ist schon gelöst und die entsprechenden Werte werden Ihnen als Übergabeparamter zur Verfügung gestellt. In der Regel werden die vorbereiteten leeren Methoden immer dann erneut aufgerufen, wenn eine Aktion im GUI passiert ist.
- Betrachten Sie die Beispiele auf dem Tab 'Bsp.' und die zugehörigen Header- und Source-Dateien im Verzeichnis Example. Machen Sie sich mit dem Programm vertraut und versuchen Sie die Funktionsweise nachzuvollziehen. Laden Sie ein Bild und wenden die Algorithmen an.
- Erstellen Sie kleine Änderungen, zeichnen Sie z.B. das Kreuz in einer anderen Farbe oder ändern Sie es so ab, dass die Balken diagonal verlaufen in X-Form.
- In `Example2.cpp` ist auch die Log-Funktionalität verwendet worden. 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.