

Gründe für die Wahl von Qt für die UI-Entwicklung

19. Februar 2025 von Mairita

Qt ist wohl das am besten geeignete Framework für den kommerziellen Einsatz als plattformübergreifende GUI-Bibliothek. Es ist für C++, Python, Go, Haskell und verschiedene andere Programmiersprachen verfügbar. Natürlich stehen Entwicklern bei der Erstellung von Benutzeroberflächen auch viele andere Frameworks zur Verfügung, wie z.B. wxWidgets, JUCE, CEGUI, Tk oder sogar GTK. Jedes dieser Frameworks hat jedoch seine eigenen Einschränkungen und Unzulänglichkeiten.

Auch Qt ist nicht perfekt, aber wir glauben, dass es weniger Nachteile hat als jedes andere GUI-Framework. Es hat sich bewährt und ist wahrscheinlich das bekannteste, am besten dokumentierte und gut unterstützte Framework auf dem Markt. Qt wurde entwickelt, um ein konsistentes Werkzeug für die Erstellung plattformübergreifender Benutzeroberflächen zu sein, oder, wie es die Qt Company ausdrückt, „flüssige, leistungsstarke und intuitive Benutzeroberflächen“.

QML

Was Qt zum mächtigsten Werkzeug für die plattformübergreifende Entwicklung macht, ist QML. Dabei handelt es sich nicht einfach um ein weiteres Qt-Modul, sondern um die Qt-Meta- (oder Modellierungs-) Sprache, mit der Entwickler mithilfe einer relativ einfachen JSON-basierten Syntax leistungsstarke Benutzeroberflächen erstellen können. Der Inline-JavaScript-Code übernimmt die imperativen Aspekte.

QML bietet eine API, mit der Entwickler die QML-Sprache um benutzerdefinierte Typen erweitern und QML-Code in JavaScript und C++ integrieren können. QML bietet sowohl eine QML-API als auch eine C++-API.

QML wird vor allem bei der Entwicklung mobiler Anwendungen eingesetzt, wo Aspekte wie Berührungseingabe, flüssige Animationen (60 FPS) und nahtlose Benutzererfahrung entscheidend sind.

Qt Controls

Qt Controls Eines der Hauptprobleme von Qt waren in der Vergangenheit seine UI-Steuererelemente. Dieses Problem wurde jetzt mit der Einführung von Qt Quick Controls 2.0 erfolgreich gelöst. Viele Entwickler haben sich darüber beschwert, dass es unmöglich war, eine Benutzeroberfläche zu erstellen, die genau wie eine native Anwendung aussieht, insbesondere auf iPhones und macOS.

Ursprünglich funktionierte das Qt-Framework, indem es Schnappschüsse von nativen Steuererelementen machte, diese optimierte und sie als Vorder- oder Hintergrundelemente in

seinen eigenen UI-Komponenten verwendete. Dieser Ansatz funktionierte jedoch nicht gut für Animationen und Übergänge.

Um dieses Problem zu lösen, haben die Qt-Entwickler die Qt Quick Controls 2.0 Bibliothek in Qt 5.8 veröffentlicht. Das Ziel war es, einen alternativen Ansatz zu bieten, indem untersucht wurde, ob es möglich ist, tatsächliche native Steuerelemente in eine plattformübergreifende Qt-API zu verpacken. Das Qt Quick Controls-Modul wurde in erster Linie entwickelt, um die UI-Steuerelemente für mobile und eingebettete Systeme zu optimieren.

Die Version Qt Controls 2.1 enthält neue Funktionen für Desktop-, mobile und eingebettete Systeme, darunter:

- Hover-Effekte.
- Ausstrahlungseffekte.
- Eine neue Themenoption für Material- und Universal-Stile.
- Ein neues Dialog-Popup.
- Ein RoundButton, der als Floating Action Button verwendet werden kann.

Diese Funktionen wurden hinzugefügt, um die Google Material Design Richtlinien und die Windows Phone Design Richtlinien besser zu erfüllen.

Außerdem gibt es separate Qt-Erweiterungen, darunter:

- QML-Themen- und Komponentenbibliothek für Android.
- Material Design Komponenten mit Qt/QML.
- QML-Themen- und Komponentenbibliothek für iOS.

Andere Qt-Funktionen

Qt bietet viele weitere Funktionen, die es zum bevorzugten Framework für die GUI-Entwicklung machen.

Windowing System

Das Qt GUI-Modul bietet Klassen, die die Integration mit Windowing System, Ereignisbehandlung, OpenGL und OpenGL ES, 2D-Grafiken, Bildbearbeitung, Schriftarten und Text ermöglichen. Diese Klassen können direkt verwendet werden – zum Beispiel durch das Schreiben von Anwendungen, die Low-Level OpenGL ES Grafik-APIs verwenden.

Für Anwendungsentwickler, die Benutzeroberflächen entwerfen, bietet Qt APIs auf höherer Ebene, wie z.B. Qt Quick. Das 2D-Grafikmodul bietet Klassen für Grafiken, Bildbearbeitung, Schriftarten und erweiterte Typografie. Die Klasse QWindow unterstützt das Rendering von Grafiken über OpenGL und OpenGL ES, je nachdem, was die Plattform unterstützt. Mit dem GUI-Modul können Entwickler Benutzeroberflächen mit erweiterten Funktionen wie Karten, Diagrammen und Datenvisualisierung erstellen.

Qt Quick und andere Tools

Qt verfügt über ein eigenes UI-Design-Tool (Qt Designer), mit dem Sie ganz einfach eine Benutzeroberfläche mit Spacern, Schaltflächen, einem HTML-Editor und anderen Elementen

erstellen können. Die Qt Quick-Bibliothek reduziert den Arbeitsaufwand im Vergleich zur traditionellen nativen UI-Entwicklung in C oder C++ erheblich. Die Einführung von Qt Controls und Qt Layouts hat die Entwicklungszeit weiter verkürzt, indem vorgefertigte UI-Steuer-elemente und Layouts angeboten werden.

Animationen

Qt Quick und QML erleichtern die Animation von UI-Komponenten und sorgen für ein flüssigeres und natürlicheres Erlebnis. Anstatt dass UI-Elemente plötzlich auftauchen oder verschwinden, sorgen Animationen für sanfte Übergänge und ein besseres Benutzererlebnis.

Benutzererfahrung (UX)

Qt 5.x unterstützt Touch-, Maus- und Tastatureingaben. Die Module Qt Sensors und Qt Positioning ermöglichen berührungsbasierte Interaktionen, die sich an den üblichen Verhaltensweisen auf tragbaren Geräten orientieren. Die Qt Sensors API bietet Zugriff auf Sensor-Hardware, darunter Beschleunigungsmesser, Kompass und Gyroskope.

In der Zwischenzeit ruft Qt Positioning Positionsdaten ab, wie z.B. den Standort eines Objekts auf einer Karte oder eine gebietsbezogene Überwachung. Die Qt Sensors API kann sowohl in QML- als auch in C++-Umgebungen verwendet werden, so dass Entwickler flexibel den besten Ansatz für ihre Projekte wählen können.

Web-Engine

Qt WebEngine ist eine Rendering-Engine für Webinhalte, die wie viele beliebte Browser auf dem Open-Source-Projekt Chromium basiert. Sie unterstützt HTML5, CSS-Filter, Animationen, Video und Canvas. Sie lässt sich nahtlos mit Qt Quick und Qt Widgets integrieren.

Das Qt WebChannel-Modul überbrückt die Lücke zwischen QML/C++ und JavaScript/HTML und ermöglicht die gemeinsame Nutzung von QObjects aus QML/C++ mit HTML/JavaScript-basierten Clients.

Multimedia

Qt verfügt über starke Multimedia-Fähigkeiten mit einem umfangreichen Satz von QML-Typen und C++-Klassen für die Handhabung von Audio-, Video-, Radio- und Kamerafunktionen.

- Qt Multimedia Classes bieten Multimedia-Unterstützung für Qt-Anwendungen.
- Qt Multimedia Widgets ermöglichen die Integration von Multimedia-Funktionen in traditionelle Qt Widget-Anwendungen.
- Die Multimedia-Module sind plattformübergreifend und funktionieren auf allen wichtigen Betriebssystemen.

Unterstützte Betriebssysteme

Qt kann kommerziell auf fast allen gängigen Desktop- und Mobilbetriebssystemen verwendet werden, darunter Windows, Linux, macOS, iOS und Android. Dies garantiert jedoch nicht immer eine Benutzeroberfläche, die genau wie eine native App aussieht. Einige Anpassungen können dennoch erforderlich sein.

Um diesen Prozess zu vereinfachen, bietet Qt:

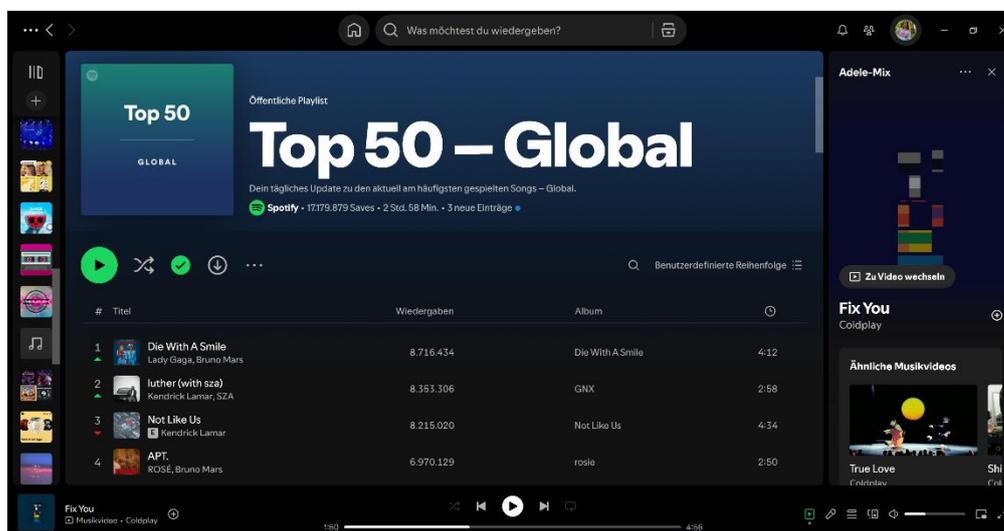
- Qt Extras – Plattformspezifische Funktionen zur Erstellung einer nativen GUI.
- Qt Android Extras – Ermöglicht die Integration von Java-Code in Qt-Anwendungen.
- Qt Mac Extras – Bietet Zugang zu macOS- und iOS-spezifischen Funktionen.
- Qt Windows Extras – Enthält Windows-spezifische Funktionen wie Sprunglisten und Fortschrittsanzeigen in der Taskleiste.
- Qt X11 Extras – Ermöglicht den Zugriff auf die Bildschirmkonfiguration in Linux (X11).

Qt kann auch zur Entwicklung für tvOS (Apple TV) und watchOS (Apple Watch) verwendet werden. Qt für tvOS ähnelt zu 95 % iOS, wobei der Hauptunterschied in der Handhabung der Eingaben besteht (Touch-basiert für iOS vs. Focus-basiert für tvOS). Allerdings unterstützt Qt für watchOS weder QML noch eine andere Qt-basierte Benutzeroberfläche, da Apples WatchKit API erforderlich ist.

UI Beispiele

Einige großartige UI-Beispiele, die mit dem Qt-Framework entwickelt wurden, sind:

- Telegramm (Desktop-Version).
- Google Earth (Desktop-Version).
- Autodesk Maya.
- Mercedes-Benz Multimedia-System (MBUX).
- Spotify für Windows und viele mehr.



Mit seiner breiten Palette an Funktionen bleibt Qt eines der besten Frameworks für die plattformübergreifende UI-Entwicklung! Das Qt-Team von helloQt hilft Ihnen gerne bei der Implementierung Ihres plattformübergreifenden Qt-Projekts.