

Un nuovo protagonista per la strumentazione medica

La progettazione di interfacce utente sta diventando un elemento sempre più importante nello sviluppo della strumentazione a uso medico; alcuni consigli per realizzare interfacce utente accattivanti e sempre più funzionali

Sami Makkonen
Senior product manager
Digia, Qt Commercial

La crescita esponenziale nell'utilizzo di smartphone e tablet ha creato aspettative sempre maggiori per quel che riguarda le interfacce utente utilizzate negli apparecchi elettronici. Lo stesso fenomeno si è esteso all'ambito medicale; in questo segmento, se da un lato i componenti utilizzati per realizzare questi sistemi sono gli stessi di quelli di tanti altri settori, dall'altro il target di utenti è tipicamente meno esperto, dal punto di vista tecnico, pur essendo comunque molto esigente.

Accade ormai sempre più spesso che la strumentazione medica si stia "trasferendo" dai reparti ospedalieri all'interno delle mura domestiche; le apparecchiature portatili vengono trasportate dal personale medico che effettua visite a domicilio, a volte vengono lasciate in custodia ai pazienti che così possono monitorare i propri progressi. Ciò comporta una serie di considerazioni nella progettazione di questi prodotti. L'affidabilità e l'efficienza delle batterie sono sempre questioni importanti ma, oltre a ciò, la semplicità e la facilità d'uso assumono un aspetto critico. Non ci si può certo aspettare che l'utente medio si metta a leggere un manuale tecnico dettagliato prima di rilevare la misura di un valore fisiologico da trasmettere al proprio medico. In altre parole, i dispositivi dovrebbero essere intuitivi e accattivanti. Tutti questi fattori spingono i progettisti a usare display TFT, spesso abbinati a sensori tattili (touch) per il controllo; ciò apre



Fig. 1 - Interfaccia utente per il paziente realizzata tramite Qt

la strada alla realizzazione di controlli soft programmabili e alla possibilità di effettuare l'aggiornamento dei sistemi tramite software, slegandosi dai limiti fisici imposti dai dispositivi di manovra. Un altro vantaggio dell'uso delle interfacce touchscreen in ambito medico - soprattutto negli ospedali - è il miglioramento del livello di igiene. Ogni interruttore, pulsante o manopola rappresenta un habitat ideale per i batteri e costituisce un reale problema per quel che concerne la pulizia. Uno schermo touch, per contro, presenta una superficie liscia e semplice da pulire.

La sfida per i progettisti

In ogni progetto tutte le componenti coinvolte - designer, i progettisti hardware e software, i programmatori - devono lavorare in completa sinergia. A turbare questo equilibrio si presentano

alla ribalta nuove problematiche legate all'integrazione nel prodotto di interfacce grafiche "accattivanti" per l'utente.

In primo luogo, non tutti coloro che sono coinvolti nel processo di sviluppo hanno familiarità con le tecniche e i tool necessari per la realizzazione di una GUI di nuova generazione. A ciò si aggiunge la pressione imposta dal time to market e dalla divisione marketing, che spesso richiede demo e prototipi da mostrare a clienti potenziali o da testare su un campione di utenti finali. Inoltre, può succedere che l'hardware effettivo non sia ancora disponibile. Questo può portare a un ritardo - imputabile all'attesa dell'hardware definitivo - oppure alla necessità di collaudare il software su altre piattaforme per poi effettuare il porting sul prodotto finale, con tutti gli svantaggi che ciò comporta.

Infine viene spesso richiesto di progettare un'interfaccia grafica professionale e di qualità, e potrebbe essere la prima volta che il gruppo di lavoro si trova a dover affrontare una sfida simile.

È noto che la diffusione capillare degli smartphone ha alzato la posta in gioco. Produrre elementi grafici, effetti e animazioni a partire da zero è un'operazione che richiede tempi lunghi.

C'è poi da considerare che lo sviluppo di un'interfaccia grafica non riguarda solo l'aspetto prettamente visivo; anche la codifica deve essere eseguita in maniera efficiente, così da non richiedere un'ampiezza di banda eccessiva del processore di sistema. Senza dimenticare che il software dell'interfaccia grafica deve integrarsi perfettamente con il sistema operativo del dispositivo, che potrebbe anche essere del tipo real-time (RTOS), nel qual caso diventa necessario non compromettere i tempi di risposta o le prestazioni del sistema.

L'ambiente di sviluppo Qt

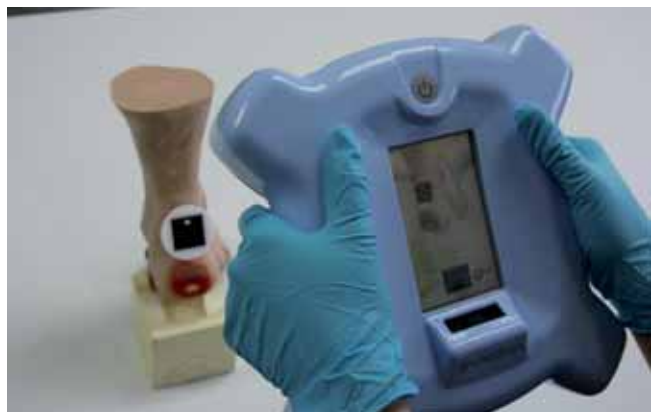
Qt è un framework cross-platform per la realizzazione di interfacce grafiche molto diffuso e ampiamente utilizzato in

Qt in pratica

Eykona Technologies è una delle aziende che hanno scelto di adottare Qt per introdurre la propria soluzione sul mercato. Nata nel 2007 presso l'Università di Oxford, Eykona si è occupata dello sviluppo di uno strumento per la misura delle ferite basato su di una ricerca fatta al Dipartimento di Ingegneria dell'Università. Eykona Wound Measurement System fornisce al personale sanitario uno strumento obiettivo capace di analizzare in maniera replicabile il volume di una ferita, permettendo il confronto e mantenendo sotto controllo il processo di guarigione.

Eykona ha individuato un'effettiva esigenza di misure accurate dal punto di vista clinico di ferite dalla guarigione difficile, quali ad esempio le ulcere diabetiche e da pressione. Le tecniche esistenti stimavano il volume delle ferite (la cui evoluzione rappresenta un indicatore importante per il processo di guarigione) moltiplicando un'area stimata con un misura in profondità ottenuta tramite sonda. Questo metodo si è però rivelato essere impreciso e difficile da replicare. Eykona si è così servita della sua tecnologia per sviluppare un sistema di diagnostica per immagini portatile che riproduce un modello 3D a colori di una ferita, che può essere archiviato nella cartella personale del paziente e condiviso per controllo e valutazione da remoto.

Dopo aver valutato una serie di interfacce grafiche e ambienti applicativi, Eykona ha scelto Qt



Wound Measurement System è uno strumento per la misura della pressione sviluppato da Eykona

per via della sua flessibilità, della possibilità di operare su più piattaforme e della sua diffusione. Eykona ha poi scelto di adottare Qt Commercial per usufruire del supporto e della licenza commerciale invece di prendere in considerazione l'alternativa open source; in ogni caso, gli sviluppatori hanno comunque attinto frequentemente alla base di conoscenze creata dalla community Qt.

Gli ingegneri di Eykona hanno avuto riscontri positivi dalle caratteristiche multi-piattaforma di Qt, poiché queste hanno permesso loro di sviluppare un'applicazione Linux embedded prima che il supporto hardware fosse disponibile: all'inizio hanno sviluppato in ambiente Windows, poi hanno importato direttamente in ambiente Linux. Senza Qt sarebbero stati necessari 6-9 mesi in più per poter rilasciare la versione iniziale del prodotto.

ambito medicale, aeronautico, automobilistico e nell'automazione industriale. È stato adottato da più di 450.000 sviluppatori in tutto il mondo ed è disponibile con licenza commerciale Digia Qt Commercial e con licenza open source LGPL.

Qt permette di sviluppare le interfacce utente in un ambiente grafico che offre ai progettisti lo stesso aspetto e gli stessi strumenti solitamente disponibili in ambiente desktop. Il motore di layout avanzato aiuta a semplificare il processo di adattamento delle interfacce per diverse risoluzioni e orientamenti degli schermi.

Grazie all'ambiente di programmazione C++ e all'approccio intuitivo orientato agli oggetti, Qt aumenta la produttività fornendo estensioni predeterminate insieme con classi e moduli che riprendono le funzionalità più comuni dei sistemi operativi.

Qt contiene una libreria modulare di oltre 800 classi C++, che offrono più di una semplice interfaccia grafica utente. Grazie alle classi per XML, networking, IPC, SVG, threading, SQL, internazionalizzazione e multimedia, Qt fornisce un ambiente di sviluppo completo che permette agli sviluppatori di concentrarsi sulle loro competenze e sulle caratteristiche distintive del prodotto. Può venire configurato per includere solo i moduli necessari così da limitare l'occupazione di memoria.

Ogni componente delle interfacce utente di questo tipo può venire personalizzato in svariati modi, dal livello di classe per la gestione degli eventi fino alla definizione del colore o tema. Qt riesce a fare tutto ciò rendendo virtuali le funzioni principali, ossia uno sviluppatore può sostituire parte delle funzionalità di una classe con una propria implementazione.

Prestazioni e portabilità

Grazie alla minima dipendenza dall'hardware, Qt offre la possibilità di migrare con semplicità tra piattaforme Linux e Windows Embedded, così come tra sistemi operativi Linux, Windows, Mac e Unix. Sono inoltre supportati anche numerosi sistemi operativi real-time tra cui quelli di Wind River, Green Hills Software e QNX. La portabilità cross platform diventa importante quando un prodotto viene sviluppato per un sistema operativo, ma nelle sue versioni successive verrà utilizzato un sistema operativo differente, oppure quando il prodotto verrà distribuito su piattaforme multiple: in questo caso il codice applicativo Qt può semplicemente venire ricompilato per i diversi sistemi operativi. Qt è stato anche ottimizzato per funzionare su una vasta gamma di piattaforme hardware, comprese ARM, x86, PowerPC e MIPS.

La libreria multithread consente al processore di usare thread

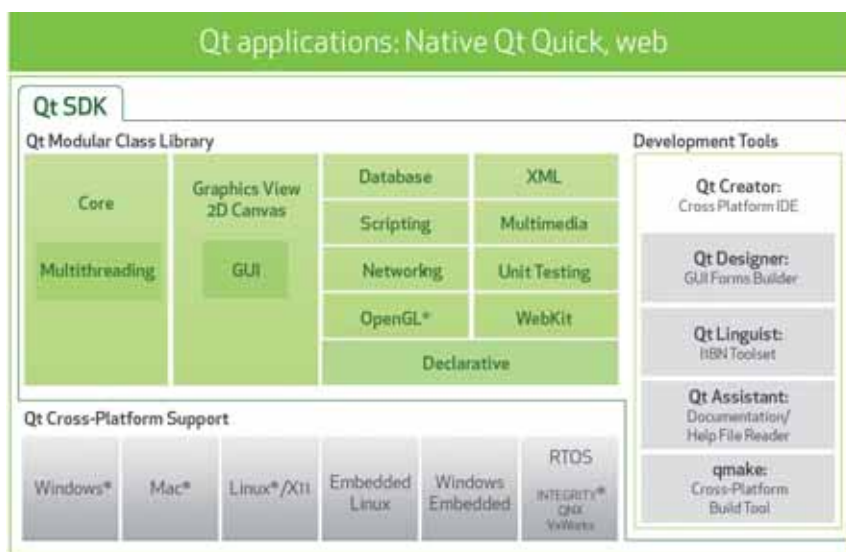


Fig. 2 - Schema a blocchi dello stack del framework Qt

separati per l'interfaccia utente e per altre funzioni. Ciò contribuisce a migliorare la velocità di risposta dell'interfaccia utente, con risultati positivi in termini di fruizione da parte dell'utente. Qt ha anche il suo sistema di windowing, eliminando la necessità di installare X11 sui dispositivi embedded.

Spesso i progetti vengono avviati prima che l'hardware sia disponibile, così gli sviluppatori devono trovare il modo di collaudare i loro prototipi; anche quando l'hardware è disponibile, i cicli di compilazione e di download possono rallentare l'avanzamento dell'intero progetto. Per superare questo ostacolo, Qt per Linux embedded presenta un'implementazione con frame buffer virtuale, che funziona sulla piattaforma hosting Linux di sviluppo. Il frame buffer virtuale accelera lo sviluppo simulando la risoluzione e profondità di colore nativi. Inoltre, gli sviluppatori possono servirsi di una skin per il frame buffer virtuale e definire la collocazione dei pulsanti, permettendo di verificare il comportamento delle applicazioni anche senza l'hardware fisico del sistema embedded.

Velocità elevate e grafica accattivante

L'ambiente Qt consente di creare interfacce utente accattivanti per dispositivi embedded tramite una palette di colori avanzata, che permette di lavorare sulla risoluzione fino al singolo pixel con linee, curve e gradienti, di mescolare e sfumare e di utilizzare metodi di composizione a canali alfa (per la semitrasparenza).

Inoltre permette di effettuare rendering avanzati di testo, che comprendono la possibilità di realizzare il pre-rendering dei font, così da ridurre il carico per il processore. Qt supporta OpenGL ES per le operazioni di grafica 3D e fast painting, oltre a OpenVG per la grafica 2D accelerata.