

Qt - RAZVOJNA OKOLINA ZA FORMIRANJE APLIKACIJA Qt - A CROSS - PLATFORM APPLICATION UI FRAMEWORK

Cerovina Dajana, Elektrotehnički fakultet, Istočno Sarajevo

Sadržaj -Qt je integrisana razvojna okolina (IDE) koja omogućava korištenje alata za dizajniranje i razvoj aplikacija posredstvom korisničkog interfejsa (UI). Koristeći Qt, jednom napisane aplikacije se raspoređuju u mnoge ugrađene operativne sisteme bez prepisivanja izvornog koda čime se smanjuju troškovi razvoja. Qt-Creator se razlikuje od tekst editora po tome što zna kako da izradi i pokrene aplikaciju. On razumije C++ i QML jezike kao kod, ne samo kao puni tekst. Qt-Creator pruža podršku za izgradnju i rad Qt aplikacija za desktop okruženje (Windows, Linux, Mac OSX) i mobilne uređaje (Symbian i Maemo). Da bi nam omogućio da izradimo i pokrenemo aplikacije, Qt-Creator treba iste informacije kao i što bi kompajler trebao. Ove informacije su specifikovane u izgradnji projekta i podešavanjima vezanim za pokretanje. U ovome radu će biti objašnjena Qt razvojna okolina i korisnički interfejs.

Abstract -Qt is an integrated development environment (IDE) that allows the use of tools for designing and developing application through the user interface (UI). Using Qt, a written application is allocated to many embedded operating systems without rewriting the source code, which deduces development posts. Qt-Creator is different from text editor in that it knows how to develop and launch the application..He understands C++ and QML language as code, not just as full text. Qt-Creator provides support for construction and operation of Qt applications for the desktop environment (Windows, Linux, Mac OSX) and mobile (Symbian and Maemo). To enable us to develop and launch applicatins, Qt-Creator will need the the same information as compiler would need. This information is specified in the construction project and adjustments related to the launch. The paper will explain the Qt development environment and user interface.

1. KRATAK ISTORIJSKI PREGLED Qt



Qt okvir je prvi put postao javno dostupan u maju 1995. godine. Inicijalno su ga razvili Haavard Nord i Eirik Chambe-Eng. Haavar i Eirik su se upoznali na Norveškom institutu za tehnologiju gdje su diplomirali kompijuterske nauke. Havaard-ov interes za razvoj C++ GUI okruženja je počeo 1988. godine kada je radio u Švedskoj kompaniji za razvoj GUI-a.

Nekoliko godina kasnije, u leto 1990. godine Haavard i Eirik su radili zajedno na C++ aplikacijama za bazu podataka ultrazvučnih slika. Sistem je trebao da radi sa GUI na Unix-u, Macintosh-u i Windows-ovom okruženju. Jednoga dana u ljeto su izašli napolje da uživaju u suncu kada je Haavard rekao "Mi trebamo objektno orjentisani ekranski sistem"!

Ovaj rzgovor je postavio temelj za intelektualno objektno-orjentisane platforme GUI okvira na čijoj će se izradi dosta raditi. Odmah nakon toga je Haavard započeo da razvija klase koje će činiti Qt. Eirik je dobio ideju za razvoj jednostavne ali moćne programibilne paradigme GUI, a to je "signals and slots" i ručno kodiranih implementacija. Već 1993. godine su bili sposobni da sami razviju svoje sopstvene widgets (dodatke). Krajem godine oni osnivaju kompaniju. Slovo "Q" je izabrano kao klasni prefiks jer je izgledalo lijepo u Haavardovom fontu, a slovo "t"

je dodato kao oznaka za "toolkit" "inspirisan sa Xt, the X Toolkit. Kompanija je osnovana marta 1994. godine kao Quasar Technologies, zatim kao Trolltech. Ubrzo se razvija i softverska podrška za Qt. Prvo je formirana Qt 0.90 verzija. Svake godine su se razvijale sve naprednije verzije do one koja je razvijena 2005. godine a to je Qt 4.0.

Qt 4.0 je prvi Qt editor koji je bio dostupan za komercijalni i otvoren razvoj na svim platformama koje podržava. Od formiranja ove kompanije, popularnost Qt-a stalno raste. Uspjeh Qt-a je garantovan njegovim kvalitetom i njegovom ugodnošću za korištenje.

2. ŠTA SE SVE MOŽE URADITI SA Qt

Qt je jedna integrisana razvojna okolina (IDE) koja omogućava korištenje alata za dizajniranje i razvoj aplikacija posredstvom korisničkog interfejsa (UI). On uključuje unakrsno-platformsku biblioteku klasa, integrisane razvojne alate za pravljenje aplikacija koje se mogu rasporediti u mnoge ugrađene sisteme bez prepisivanja izvornog koda. Pomoću Qt-a mogu da se pišu i web skripte.

-Jednom napisani kod povezuje više platformi.

Qt dozvoljava da se napiše napredna aplikacija i UI-s za prikaz na desktop- u i ugrađenom operativnom sistemu bez ponovnog prepisivanja izvornog koda. Ovo ušteduje vrijeme i troškove razvoja.

-Razvija nevjerovatno korisničko iskustvo.

Bez obzira da li preferiramo C++ ili Java jezik, Qt pruža gradivne blokove- širok skup gradivnih datoteka koje su prilagodljive, grafiku okruženja, stil pokretanja i još mnogo

toga sto je potrebno da se razvije moderan korisnički interfejs. Za konkurentno kreiranje aplikacija poželjno je koristiti 3D grafiku, multimedijalne audio i video efekte, vizuelne efekte i animacije.

-Uraditi više sa manje vremena.

Sa novim Qt-Creator unakrsnom platformom IDE ili na sopstvenoj Qt je lako i brzo dizajnirati. Zahvaljujući Qt bibliotečkim klasama više vremena se posvećuje vodeći računa o inovacijama, a manje vremena o infrastrukturi kodiranja. Zbog ovoga se brzo dobija softver na tržištu za razvoj Qt-a.

-Mješavina Web-a i prirodnog koda u jednoj aplikaciji. Integracija Qt-a sa Web okruženjem znači da se mogu brzo inkorporirati sadržaji i usluge sa interneta na matičnim aplikacijama. To se postiže tako što web brzo dostavlja svoje usluge i funkcionalnost korisnicima.

3. KO PRAVI I KORISTI Qt ?!

-Nokia to radi.

Qt se prvenstveno razvija i održava od strane neustrašivih programera koji rade u "Qt Development Frameworks", jedinica unutar Nokie. Oni su radili na razvju Qt-a preko 14. godina i danas hiljade kompanija širom svijeta koristi Qt za razvoj dobro poznate desktop, ugrađenih i mobilnih aplikacija i korisničkog interfejsa.

-Zajednica koja posjeduje licencu.

U početku je korištenje Qt-a bilo ograničeno licencama (GPL i LGPL) u komercijalne svrhe. To je omogućilo da se prikupi mnogo korisničkih povratničkih informacija i da se Qt razvije brže.

-Ti možeš da koristiš Qt!

Za razvoj sopstvene Qt aplikacije može se koristiti Qt mape puteva i riznice izvornog koda koje su otvorene za javnost.

Ovi možeš doprinjeti

Budućem razvoju Qt-a, razvojem koda, prevodenjem i raznim implementacijama vezanim za projekat.

Qt se koristi od strane hiljadu kompanija. On im omogućava da razviju moćan softver za korištenje uređaja koji će biti potrebni velikom broju ljudi.

-Samsung frame (Samsung SPF-105V Digital Photo Frame)

Samsung je bio sposoban da razvije Web-skriptni digitalni ram sa dinamičkim informacijama sa Qt GUI i aplikacionim okvirom.

-Rocu

Ovaj uređaj omogućava da se direktno prenese sadržaj sa Netflix-a do televizora potrošača korištenjem Qt bežičnog interfejsa.

-Barco

Komunikator dodira-Barco je digitalni projektor koji je izgrađen korištenjem Qt-a. On omogućava razvoj snažnog korisničkog interfejsa (GUI) i jedan izvorni kod multiplatformske primjene.

-ASUS EeePC

Asus je izdao sveobuhvatni komplet programa (SDK) Za mobilne internet Asus-ove uređaje. Linux verzija koristi KDE desktop okruženaj bazirana na Qt-u.

-Next Limit Technologies

Next Limit Technologies je osvojio Oskara za tehniku i za jednu od svojih Qt zasnovanih aplikacija, RealFlow.

Koristili su Qt da bi dobili odgovarajuću tačnost korištenjem dinamičke i svjetleće simulacije softvera.

-DAZ 3D

DAZ 3D je razvio moćan softver za kreiranje aplikacija sa digitalnim sadržajem. Qt je omogućio pojednostavljen razvoj platformama i znatno skratio ciklus rada korištenjem programibilnih pristupa.

4. RAZVOJ PLATFORME

Jedna od najvećih prednosti Qt-Creator-a kao jedne integrisane razvojne okoline je što omogućava timu ljudi da dijele projekat, onaj projekat na čijem su razvoju radili preko različitih razvojnih platformi sa uobičajnim alatima za razvoj i debagovanje. Preporučljiv način za razvoj projekta je korištenje "version control system". Ovom komandom Qt-Creator omogućava korisniku da pristupi spremištu. Podržani su sledeći sistemi kontrole verzije:

-Gid,

-Subversion,

-Mercurial,

-CVS;

Osnovna funkcije su dostupne za sve podržane sisteme i sve verzije upoređuju svoje funkcije sa najnovijom verzijom. Qt-Creator dopušta da se specificiraju odvojena gradivna podešavanja za svaku razvojnu platformu. Može se koristiti qmake scopes da se označi fajl koji će se izvršavati zavisno od platforme na kojoj se qmake izvršava. Stavke kao što su otvorene datoteke, tačke prekida su smeštene u sesijama. One se ne smatraju kao dio informacija koji je djeljiv među različitim platformama.

5. KREIRANJE PROJEKATA

Da bi nam omogućio da izgradimo i pokrenemo aplikaciju, Qt Creator treba iste informacije koje bi i kompajler trebao. Ove informacije su specificirane u izgradnji projekta i podešavanjima vezanim za pokretanje.

Kreiranje projekta omogućava:

-grupisanje datoteka,

-dodavanje koraka za izradu aplikacije,

-uključi obrsce i resurse fajlova,

-precizira postavke za pokretanje programa,

Postavljanje novog projekta u Qt-Creator-u je potpomognuto čarobnjakom koji vodi korak po korak kroz proces kreiranja projekta. U prvom koraku se bira tip projekta iz kategorije Qt C++ Project ili Qt Quick Project ili neki drugi projekti. Dalje se izabira lokacija za projekat i preciziraju se postavke za to. Kada se završe ovi koraci, Qt-Creator automatski generiše projekat sa potrebnim zaglavljljima, izvornim fajlovima, opisom korisničkog interfejsa i projekat datoteke izabran čarobnjakom. Na primjer, ako se odabere da se kreira Qt Quick application, Qt-Creator generiše QML fajl koji se može mijenjati sa integrisanim Qt Quick Designer. Qt-Creator daje mogućnost i za dodavanje drugih datoteka koje nisu Qt tipa u projekat. Mogu se povezati aplikacije sa bibliotekama kao što su sistemske biblioteke ili naše sopstvene. Da bi bili spremni za kompajliranje, potrebno je dodati biblioteke u projekat. Ovo omogućava kod završetka i isticanje sintakse za biblioteke. Postupak dodavanja biblioteke projektu zavisi od sistema za izgradnju koji se koristi.

6. KREIRANJE KORISNIČKOG INTERFEJSA

Qt Creator omogućava dva integrisana vizuelna editora, Qt Quick Designer i Qt Designer. Ekran velike rezolucije, unos podataka dodirom i značajna grafika su postali uobičajni u prenosnim potrošačkim uređajima kao što su mobilni telefoni, medijski plejeri, set top box i prenosivi računari. Da bi se u potpunosti iskoristile ove funkcije i da bi se stvorio korisnički interfejs modernog izgleda, koristi se Qt Quick. On se sastoji od velikog skupa elemenata korisničkog interfejsa kao i od deklarativnog jezika za opisivanje ovoga interfejsa. Kolekcija C++ API se koristi da integriše te visoke funkcije sa klasičnom Qt aplikacijama. Može se editovati QML kod u integrisanom Qt Quick Designer-u. Integracija obuhvata upravljanje projektom i kod završetka. Ako je potreban tradicionalni korisnički interfejs koji je jasno struktuiran, može se koristiti Qt Designer. Moguće je napisati i prilagoditi datoteke i dijaloge i testirati ih korištenjem različitih stilova Qt Designer-a.

7. KODIRANJE

Kao IDE, Qt-Creator se razlikuje od tekst Editora u tome što zna kako da izgradi i pokrene aplikaciju. On razumije C++ i QML jezik kao kod a ne samo kao puni tekst. Ovo omogućava sledeće:

- omogućava da se napiše dobro oblikovan kod,
- predviđa šta namjeravamo da napišemo i kompajlira kod,
- prikazuje unutar linije greške i poruke upozorenja,
- u kontekstu nam omogućuje pomoć vezanu za klase, funkcije i simbole,
- preimenovanje simbola na inteligentan način tako da ostali simboli sa istim imenom koji pripadaju drugim dometima nisu preimenovani,
- da prikaže lokacije u kodu gdje je proglašena funkcija ili gdje se poziva.

Koristi se kod editor za pisanje koda u C++ ili QML deklarativnim jezicima. QML je proširenje Java Script-a koja obezbeđuje deklarativnu izgradnju QML elemenata. QML poboljšava integraciju između Java Script-a i Qt postojećih objekata i samim tim omogućava transparentnost na nivou jezika.

8. IZGRADNJA, TESTIRANJE I OTKLANJANJE GREŠAKA

Qt je integrisan sa platformskim sistemima za automatizovanu izgradnju qmake i CMake. Pored toga je dozvoljeno korištenje i generičkih projekata koji ne koriste ove automatizacije izgradnje i da Qt ignoriše izgradbeni sistem. Qt -Creator pruža podršku za izgradnju irad Qt aplikacija za desktop okruženja (Windows, linux, Mac OSX) i mobilne uređaje (Symbian i Maemo). Podešavanje za izgradnju omogućavaju brze prelaze između izgradbenih meta. Kada se instalira Qt SDK, izgradnja i pokretanje podešavanja su postavljena za Maemo i Symbian automatski. Međutim, potrebno je da se instaliraju i konfigurišu dodatni softveri na uređajima da bi se mogli povezati na mobilne uređaje iz razvojnih računara. Jedini podržani sistem za mobilne aplikacije u Qt-Creator-u je qmake.

Prilikom instaliranja Qt-Creatora-a kao dijela Qt SDK, GNU Symbolic Debugger je instaliran automatski i treba biti spreman na otklanjanje grešaka nakon kreiranja projekta. Podešavanja vezana za otklanjanje grešaka mogu da se promjene u Windows okruženju. To se radi tako što se mobilni uređaji povežu na razvojni računar i tako se otklanjaju greške procesa na ovim uređajima. Aplikacije koje su namjenjene ovim uređajima se mogu testirati u Qt Simulator-u Maemo Emulator-u ali su potrebna i dodatna testiranja aplikacije. Qt je integrisan sa nekoliko spoljnih maternih debagera:

-GNU Symbolic Debugger (gdb),

-Microsoft Console debugger (CDB),

-Internal Java Script Debbuger;

Možemo interaktivno sa otklanjanjem grešaka na nekoliko načina:

-idući kroz program liniju po liniju,

-postavljanjem breakpoints.

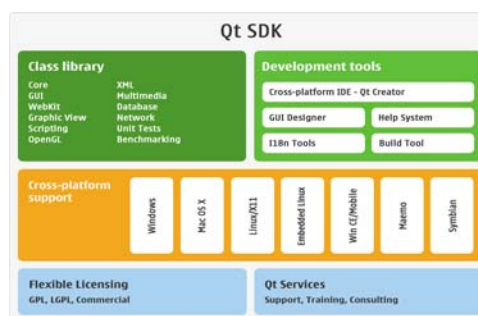
-ispitivanjem liste učitanih biblioteka,

-napravimo snimke od trenutnih stanja debagera Dok

aplikacija radi, može se koristiti i QML Observer pogled da iskaže strukturu objekata, otklanjanje grešaka, animacija, može se koristiti za skok na određeni dio koda.

9. Qt OKVIR

Qt omogućava razvoj aplikacija za otključavanje hardvera i softvera i da usluge raznih uređaja učini dostupnim. Koristeći Qt SDK-Nokia mogu se kreirati aplikacije sa jednim baznim kodom za milione Symbian, MeeGo uređaja i desktop platformi što je i prikazano na slici br. 1.



slika br. 1

Zato što je Qt okvir a ne runtime, nasa aplikacija se pokreće sa izvršnim performansama.

Qt okvir omogućava kreiranje privlačnih aplikacija. Qt SDK tehnologija daje osnovni pregled alatki i funkcija za konačnu verziju. Vrhunac karakteristike ove tehnologije prikaza je deklarativna podrška za Qt i Qt Quick:

-programeri mogu da kreiraju atrktivne i dinamične UIs korisničke zahtjeve korištenjem Qt Quick Designera, Qt Meta Quick Language (QML)-Qt 4.7 APIs je za brz razvoj bogatih visokofunkcionalnih aplikacija sada sa značajnim poboljšanjem performansi i dubljih web integracija.

-Qt Mobility 1.1 APIs sa dodatnim funkcijama za integraciju sa uređajima na nivou hardvera i softvera upotrebom kamere, kalendara, lokacija i navigacija.

-Alati za Symbian, Maemo i desktop aplikacije uključujući i maternje okruženje.

10. WIDGETS

Widgets su osnovni blokovi za grafički korisnički interfejs (GUI) koji je ugrađen aplikacijama u Qt-u. Postoji više vrsta Widget-a koji su određeni podklasom QWidget-a koja je sama podklasa klase QObject koja je inače glavna klasa za rukovanje objektima u Qt-u. može da se koristi kao skladište za druge dodatke i može da se subklasira da stvori nove prilagodljive dodatke. QWidget se koristi za kreiranje prozora glavnog roditeljskog dodatka u kome se nalaze djeca widget. Kao i kod QObject i QWidget može biti kreiran sa matičnim objektima za indicaciju vlasništva ali se mora voditi računa da se predmeti koji su obuhvaćeni tim klasama brišu kada se više ne koriste. Ovaj odnos roditelj-dijete ima dodatna značenja. Kada se obriše prozor roditeljskog dodatka, brišu se i svi dodaci djeteta koje taj roditelj dodatak sadrži. Mnogi GUI primjeri Qt-a slijede paterne ili obrazce koji se nalaze u main.cpp fajlu koji sadrži standardni kod. Za inicijalizaciju se koristi heder fajl i izvorni fajl. Oni sadrže aplikacionu logiku i uobičajne GUI komponente. QWidget kao što korisničkog interfejsa interaktuje sa mišem, tastaturom i drugim događajima iz sistema prozora. Svaki dodatak je pravougaonog oblika. Onaj dodatak koji nije ugrađen u roditeljski dodatak se naziva prozor. Obično prozori imaju okvir i naslov. Svaki konstruktor widget-a prihvata jedan ili dva standardna argumenta, npr.:

-QWidget* parent=0 je roditelj novog dodatka kako je deklarirana vrijednost 0 novi widget će biti prozor. Ako ne, onda će biti dijete i biće ograničen geometrijom roditelja.

-Qt::windowflag f=0 predstavlja zastavu i pogodan je za skoro sve dodatke ali da bi se dolil widgets bez okvira trebaju da se koriste neke druge zastave.

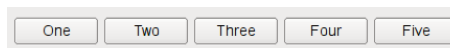
QWidget ima mnogo funkcija članica ali malo njih direktno utiče na ovu klasu kao što je slučaj što QWidget ima font karakter ali je nikada ne primjenjuje na sebi. Postoje mnoge podklase koje omogućavaju stvarnu funkcionalnost kao što su: QLabel, QPushButton, QListWidget i QLabWidget. Dodatak bez roditeljskog widget-a je uvijek nezavisan prozor. Za ove widgets, setWindowTitle () i setWindowIcon () predstavljaju naslovnu traku i izgled. Bez okvira dodaci se nazivaju dijete dodaci i prikazani su u okviru svojih roditelja. Većina dodataka se u Qt-u koriste kao djeca. Npr., moguće je prikazati dugme kao najviši nivo prozora ali većina ljudi voli da stavi tastere i druge dodatke te tako kreiraju QDialog. Kada se koristi widget kao skladište za više drugih widgeta onda se naziva kompozicioni dodatak. Prilikom implementacije novog widget-a skoro je uvijek poželjno da se podesi setSizeHint () i setSizePolicy () da bi se odredila veličina prostora koje djeca zauzimaju u roditeljskom prozoru. Dodaci reaguju na događaje koji su izazvani korisničkom interakcijom Qt i da isporučuje ove događaje dodacima preko upravljača ili handler-a čije su funkcije članice klase QEvent. Npr., ako želimo da detektujemo klik miša u dječijem dodatku koriste se undermouse () funkcije. Neki od poznatih upravljača su: closeEvent (), moveEvent (), leaveEvent (), enterEvent ().

11. UPRAVLJANJE RASPOREDOM (layout management)

Qt layout sistem omogućava običan i moćan način za automatsko redanje djece widgets unutar roditeljskog widget-a i omogućava iskorištenost dostupnog prostora. Qt uključuje grupu layout klasa za upravljanje da opišu kako su dodaci raspoređeni unutar aplikacionog korisničkog interfejsa. Ovi rasporedi daju automatsku poziciju i promjenu veličine prostora koji je dostupan, obezbjeđujući da dodaci dosledno raspoređeni i da korisnički interfejs bude upotrebljiv kao cjelina. Sve QWidget podklase mogu da koriste layout-e da upravljaju djecom pomoću funkcije QWidget::setLayout (). Kada je layout postavljen na widget-u onda vrši sledeće poslove:

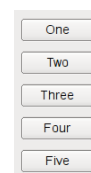
- pozicioniranje djeteta dodatka,
- pravilne podrazumjevanje veličine za prozor,
- pravilne minimalne veličine za prozor,
- promjena veličine rukovanja,
- automatsko ažuriranje kada se promijeni sadržaj,
- veličina font, tekst ili drugi sadržaji djeteta widget-a,
- skrivanje ili prikazivanje djeteta, uklanjanje djeteta

Layout se prikazuje pozivom funkcije setLayout (). Rasporedi su vidljivi samo samo kroz uticaje koji imaju na dodatke i odgovorni su za upravljanje. Baš kao što dodaci mogu da sadrže druge dodatke tako i rasporedi mogu da obezbjeđe različite nivoe grupisanja. QVBoxLayout zauzima mjesto koje dobije (od roditeljskog layout-a ili od parentWidget (), dobijajući redove kutija i omogućava im da svaki radeći widget popuni odgovarajuću kutiju. Ako je orijentacija QVBoxLayout-a, Qt::Horizontal, kockice su postavljene u odgovarajućem red sa odgovarajućom veličinom. To je prikazano na slici broj 2. Svaki widget ili drugi box će najmanje zauzeti minimalnu veličinu i najviše svoju maksimalnu veličinu. Bilo kakvo dodatno dodjeljivanje prostora se ostvaruje pomoću faktora za istezanje (stretch factors).



slika br. 2

Ako je orijentacija QVBoxLayout-a, Qt::Vertical, kvadratići su postavljeni u kolone ponovo sa odgovarajućim veličinama što je prikazano na slici broj 3.



slika br. 3

Najlakši način da se kreira Qboxayout je da se koristi jedna od dostupnih klasa kao što je QHBoxLayout ili QVBoxLayout. Može se koristiti QVBoxLayout direktno specificirajući direkciju. Ako QVBoxLayout nije vrhovni raspoređivač, ne pokriva svu djecu, onda se mora dodati svom roditeljskom raspoređivaču prije nego što se može da uradi ista sa njima. Normalan način za dodavanje layout-a je pomoću addLayout. Kada se ovo uradi mogu se dodavati kutije u QVBoxLayout korištenjem jedne od sledećih pet funkcija:

- addWidget () za dodavanje dodataka u QVBoxLayout i podesi se faktor za istezanje.
- addSpacing () da se doda prazan box, ovo je jedna od funkcija za lijep prostor dijalog,
- addStretch za kreiranje kutije za istezanje,

-add Layout () da se doda box koji sadrži drugi Qlayout u redu i podesi faktor istezanja:

Sve ove stavke su ostvarive preko funkcije insert ().

12. KAKO KORISTITI Qt DESIGNER?!

Korištenje Qt Designer-a uključuje četiri osnovna koraka:

1. izabrati svoju formu i objekte.
2. postaviti objekat na obrascu,
3. povezati signale i mjesta djelovanja,
4. pregled formulara;

Potrebno je istaći da Qt aplikaciona klasa upravlja kontrolom protoka i omogućava GUI aplikaciona podešavanja. Ona sadrži glavne događaje i petlje gdje se svi događaji od prozora sistema i drugih izvora obrađuju. Ova klasa omogućava inicijalizaciju, finalizaciju i sesiju za upravljanje. Za svaku GUI aplikaciju koja koristi Qt postoji precizirano jedan QApplication objekat bez obzira da li aplikacija ima više prozora. QApplication pokreće aplikaciju sa desktop korisnika, služi kao neka vrsta kontrolnog panela. Ona vrši obradu događaja što znači da prima događaje iz osnovnog prozora sistema i dopunjava ih relevantnim dodacima. Korištenjem sendEvent () i postEvent () možemo dodavati svoje događaje dodacima. QApplication definiše aplikacioni izgled, određuje lokalizacije karaktera, omogućava određivanje pozicije widget-a korištenjem WidgetAt () i upravljanje kursorom. Aplikacioni objekat se mora izgraditi prije bilo kojeg drugog objekta.

Na slici br. 4 je prikazan jedan jednostavan primjer formiranja aplikacije posredstvom korisničkog interfejsa koji obezbjeđuje Qt. Ova aplikacija ima ulogu da na ekranu ispiše određeni tekst, u ovom slučaju to je tekst "Hello world ". Ispod slike je ispisan kod koji podržava korisnički interfejs za ispis teksta. Na početku koda je formiran objekat Rectangle, to je aplikacioni objekat i njemu je dodjeljeno ime preko id, page. Ovaj objekat ima svoje attribute a to su: width, color koji imaju vrijednosti.

Unutar ovoga aplikacionog objekta nalazi se objekat tekst koji će se koristiti za ispis naznačenog teksta, i to u centralnom dijelu stranice.



slika br. 4

```
import QtQuick 1.0
Rectangle {
    id: page
    width: 500; height: 200
    color: "lightgray"
    Text {
        id: helloText
        text: "Hello world!"
        y: 30
        anchors.horizontalCenter: page.horizontalCenter
        font.pointSize: 24; font.bold: true
    }
}
```

13. ZAKLJUČAK

Qt-Creator je jedna integrisana razvojna okolina (IDE) Koja nas snadbjeva alatkama za dizajniranje i razvoj aplikacija sa Qt aplikacionim okvirom. On omogućava alatke za uspješno izvršavanje zadataka kroz cijeli aplikacioni životni krug, od kreiranja podataka do raspoređivanja aplikacija na određenoj platformi.

14. LITERATURA

[1] Internet

[2] Jasmin Blanchette, C++ GUI Programming with Qt 4