

# QT Eğitimi - 3 : Menü

## Önder Arslan

<[onder@xcoders.net](mailto:onder@xcoders.net)>

Sürüm 1.0

### Özet

Bu belge qt öğrenimi için bir dizi şeklinde hazırlanmıştır. C++ bilmek ön koşuldur.

### Giriş

Grafik Kullanıcı Ara Yüzü (gui) programlarının önemli öğelerinden biri de "menü" dür. Menüler genel kullanımda araç çubuğu (toolbar) yada kısa yollar (short cuts) kadar sık kullanılmada yazı olması nedeniyle görsel programların vazgeçilmezidir. Özellikle yeni öğrenilmeye başlayan programların araç çubuklarındaki ikonlar ve kısayollar öğrenilene kadar çoğunlukla menü kullanılır. Yani şunu söylemeye çalışıyorum üçüncü dersimizin konusu : Menü ;)

### Menü

Bu dersimiz de önce ki iki ders gibi bir başlık dosyası(.h) ve iki kaynak dosyasından(.cpp) oluşuyor ;) .

```
/*--- form.h ---*/

#ifdef FORM_H
#define FORM_H

#include <QtGui>

class Form : public
QMainWindow
{
    Q_OBJECT

public:
    Form();

private slots:
    void menuItem1_func();
    void menuItem2_func();
    void altmenuItem_func();

private:
    QMenu *Menu;
    QMenu *altMenu;

    QAction *menuItem1;
    QAction *menuItem2;
    QAction *altmenuItem;
};
```

```
#endif
```

## form.h:

Include önışlemci komutuyla gerekli dosyaları başlık dosyamıza ekledikten sonra - dikkat edersiniz ki artık buralarda fazla açıklama yapmıyorum, neyi neden eklemeniz gerektiğini bilmiyorsanız önceki derslere bakın ;) - Form isimli sınıfımızı public türetmesi ile **QMainWindow** sınıfından türetiyoruz. QMainWindow uygulamanın ana penceresidir. Menü çubuğu (menu bar), araç çubuğu (tool bar), durum çubuğu (status bar) vb. kontrolleri sağlar. Bizim için önemli menüyü üzerinde oluşturacağımız çubuğu (bar) QMainWindow 'un menuBar() fonksiyonu ile oluşturacak olmamız. **QMenu** sınıfı ile oluşturmak istediğimiz menülerimizi tanıtırız ben bir menü ve bu menünün altmenüsü olmak üzere iki tane menü (Menu, altMenu) tanımladım. Menüme elemanları **QAction** sınıfı türünden ekliyoruz. QAction sınıfı menüler , araç çubuğu butonları, kısa yollarla kullanılmak için soyut bir sınıf sunuyor. Bu sayede elemanlarınız için etkilenmiş (enabled) , işaretlenmiş (checked) vb. gibi özellikleri tek bir arayüzle kullanabiliyorsunuz.

```
/*--- form.cpp ---*/

#include "form.h"

Form::Form()
{
    menuItem1 = new QAction(QIcon("disk.png"), "&menuItem1", this);
    menuItem1->setShortcut("Ctrl+A");
    connect(menuItem1, SIGNAL(triggered()), this, SLOT(menuItem1_func()));

    menuItem2 = new QAction("menuItem2", this);
    menuItem2->setShortcut("Ctrl+B");
    connect(menuItem2, SIGNAL(triggered()), this, SLOT(menuItem2_func()));

    altmenuItem = new QAction("altmenuItem", this);
    altmenuItem->setShortcut("Ctrl+C");
    connect(altmenuItem, SIGNAL(triggered()), this,
    SLOT(altmenuItem_func()));

    Menu = menuBar()->addMenu("&Menu");
    Menu->addAction(menuItem1);
    Menu->addAction(menuItem2);
    Menu->addSeparator();

    altMenu = Menu->addMenu("altMenu");
    altMenu->addAction(altmenuItem);

    resize(400, 200);
}

void Form::menuItem1_func()
{
    QMessageBox::information(this, "menuItem1_func",
    "menuItem1_func");
}

void Form::menuItem2_func()
```

```
{
    QMessageBox::information(this, "menuItem2_func",
"menuItem2_func");
}

void Form::altmenuItem_func()
{
    QMessageBox::information(this, "altmenuItem_func",
"altmenuItem_func");
}
```

## form.cpp

Yapıcı (constructor) fonksiyonumuza QAction nesnelimizi oluşturarak başlıyoruz. İlk oluşturulan action 'da menuItem1'e ikon eklediğimi farketmişsinizdir. Eğer program çalıştıktan sonra menuItem1 'in yanında resim görmek istiyorsanız çalıştığımız klasörde disk.png isminde bir ikonu aynı klasöre koyduğunuzdan emin olun ;) Ayrıca menü ve elemanların isimleri tanımlanırken (ör: &menuItem1) & işareti yazıldığına dikkat edin. & işareti bulunduğu konumun sağındaki harf ile Alt tuşuna basılarak bu menü yada elemanın tıklanması ile çalışacak fonksiyonun çalışmasını sağlar. & işareti ile kısa yol oluşturmak aynı isimli menülerin ve elemanların çok olması durumunda fazla kullanışlı değildir. Böyle durumlarda kısa yol oluşturmak için QAction sınıfının setShortcut fonksiyonu ile yukarıdaki gibi kısayol atanır. Action 'ları tanımladıktan sonra menümüzü menuBar sınıfının addMenu fonksiyonuyla oluşturuyoruz. addAction fonksiyonu ile de action 'larımızı menüye ekliyoruz. Dikkat etmeniz gereken ikinci menünün menubar 'a değil birinci menüye eklendiğidir.

```
/*--- main.cpp ---*/

#include <QApplication>

#include "form.h"

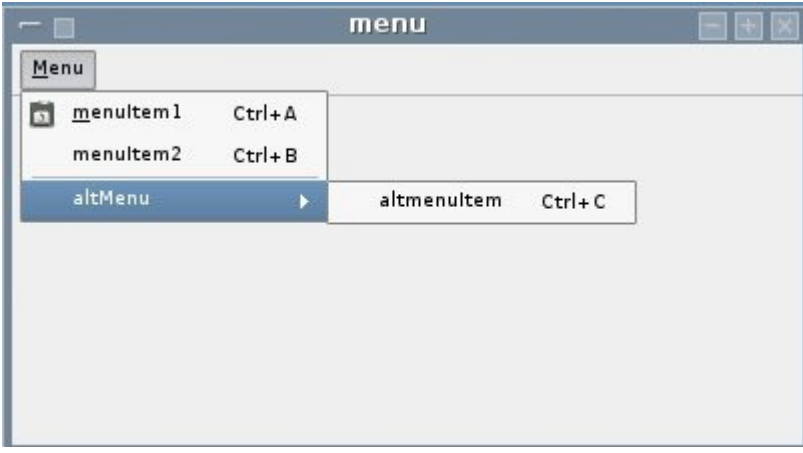
int main(int argc, char *argv[])
{
    QApplication app(argc,
argv);
    Form form;
    form.show();
    return app.exec();
}
```

## main.cpp

main.cpp kaynak kod dosyası içinde önceden açıklanmayan hiçbir kod parçası **yine** yok ;)

## Derleme ve Çalıştırma

Program çalıştığında menünüz aşağıdaki gibi görünür ;)



## Son Söz

Merak edenler için söyleyeyim menuItem1 'in yanındaki ikonda disket üzerinde bir tux var ;)

## Telif Hakkı ve Lisans

Bu belgenin, *QT Eğitimi - 3 : Menü*, 1.0 sürümünün **telif hakkı** © 2006 **Önder ARSLAN**'a aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Özgür Belgeleme Lisansının 1.1 ya da daha sonraki sürümünün koşullarına bağlı kalarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Bu Lisansın bir kopyasını <http://www.gnu.org/copyleft/fdl.html> adresinde bulabilirsiniz.

Linux, Linus Torvalds adına kayıtlı bir ticarî isimdir.

Qt, TrollTech adına kayıtlı bir ticari isimdir.

### Feragatname

Bu belgedeki bilgilerin kullanımından doğacak sorumluluklar, ve olası zararlardan belge yazarı sorumlu tutulamaz. Bu belgedeki bilgileri uygulama sorumluluğu uygulayana aittir.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim bir ticarî isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.