

Debreceni Egyetem
Informatikai Kar

Robotautó világbajnokság

Témavezető: Dr. Bátfai Norbert

Beosztása: Adjunktus

Készítette: Fábián Kristóf – Szabolcs

Szak megnevezése: Programtervező
Informatikus BSc

Tartalomjegyzék

1. Bevezetés.....	4
2. Tárgyalás.....	5
2.1 Használt szoftver eszközök.....	5
2.1.1 Qt.....	5
2.1.1.1 qmake.....	5
2.1.1.2 Modulok.....	7
2.1.1.3 Slot-signal mechanizmus.....	8
2.1.1.4 Qt Creator.....	8
2.1.2 OpenCV.....	8
2.1.2.1 Telepítés.....	8
2.1.3 BGSLib.....	8
2.2 Előzmények.....	8
2.2.1 <megelőző forgalomszámáló szoftverről>.....	8
2.3 A Robotautó Világbajnokság.....	8
2.3.1 A platform koncepciója.....	8
2.3.2 <>.....	8
2.4 A szoftver elkészítésének folyamata.....	8
2.4.1 Tervezés.....	8
2.4.2 Fejlesztés.....	8
2.4.2.1 Videó források típusok.....	8
2.4.2.2 Adatbázis kapcsolatok kialakítása, kezelése.....	8
2.4.2.3 Szálkezelés.....	9
2.4.2.4 Videó feldolgozása.....	9
2.4.2.5 Járművek detektálása.....	9
2.4.3 Problémák a szoftver fejlesztése során.....	9

2.4.4 A szoftver tesztelése.....	9
2.4.5 Eredmények.....	9
2.5 A szoftver használata.....	9
2.5.1 Fordítása.....	9
2.5.2 Indítása.....	9
2.5.3 A grafikus felhasználói felület használata.....	9
2.5.3.1 Új forrás kiválasztása.....	9
2.5.3.2 Adatbázisok kezelése.....	9
2.5.3.3 Fájl videó forrás információi.....	9
2.5.4 A parancssoros felhasználói felület használata.....	9
2.5.4.1 Használható parancsok.....	9
3. Összefoglalás.....	9
4. Köszönetnyilvánítás.....	10
5. Irodalomjegyzék.....	11

1. Bevezetés

2. Tárgyalás

2.1 Használt szoftver eszközök

2.1.1 Qt

A Qt (kiejtése az angol “cute” szóval megegyező) egy számos platformra fordítható keretrendszer, aminek segítségével PC, mobil, illetve beágyazott rendszerekre is fejleszthetünk alkalmazásokat C++ programozási nyelven. A többi keretrendszerhez eltérően, a Qt rendelkezik egy MOC (Meta – Object Compiler) [MOC] nevű előfordítóval. Ez az előfordító feldolgozza a Qt- ban írt forrásfájlokat. Ha valamely osztály deklarációban a MOC megtalálja a Q_OBJECT makrót akkor meta-objektum kódot generál belőlük, ez a kód természetesen C++ nyelvű. A Qt egyik sajátos funkciója, a slot-signal (magyarul?) a MOC-nak köszönhető. A slot-signal mechanizmusról bővebben a 2.1.1.3- as, “Slot-signal mechanizmus” nevű fejezetben írok.

Mivel számomra feltétel volt, hogy Open Source szoftvereket használjak, így a Qt nagyon jó választásnak tűnt, mivel elérhető szabad szoftverként is, LGPL/GLP licenc alatt [qtlicensing]. Emellett a Qt megvásárolható kereskedelmi licenc alatt is, mely tartalmaz olyan további funkciókat, mint például virtuális billentyűzet, adatvizualizáció. Egy közösség alapú adatgyűjtés szerint jelenleg 206 könyvtár [include] használja a Qt- t, különböző, viszont legtöbbjük valamely szabad szoftver licenc alatt érhető el.

2.1.1.1 qmake

A qmake egy olyan eszköz, amely leegyszerűsíti számunkra a fordítás menetét eltérő platformokra. Egy projekt (.pro) fájlból generál számunkra egy “Makefile”- t, vagy akár projekt fájlt is tudunk a segítségével generálni, ha egyelőre csak a forrásfájlokkal rendelkezünk. Egy “Makefile” a GNU make által használt fájl, amely leírja, hogy a make hogyan fordítsa illetve kapcsolja össze az objektum fájlokat egy futtatható állománnyá.

Vegyük példának az alábbi forrásokat:

```
krajsz@krajsz-Lenovo-G500s:~/ $ ls -lh
total 16K
-rw-rw-r-- 1 krajsz krajsz 172 apr 21 14:52 main.cpp
-rw-rw-r-- 1 krajsz krajsz 219 apr 21 14:52 mainwindow.cpp
-rw-rw-r-- 1 krajsz krajsz 291 apr 21 14:52 mainwindow.h
-rw-rw-r-- 1 krajsz krajsz 630 apr 21 14:52 mainwindow.ui
```

Ezután a qmake- nek átadva a -project kapcsolót tudunk generálni egy projekt fájlt. Átadhatjuk a forrásfájlokat argumentumként a qmake- nek, viszont, ha nem adunk meg egy fájlt sem, akkor az aktuális könyvtárban található forrásfájlok alapján generálja a projekt fájlt:

```
krajsz@krajsz-Lenovo-G500s:~/ $ qmake -project
krajsz@krajsz-Lenovo-G500s:~/ $ ls -l
total 20
-rw-rw-r-- 1 krajsz krajsz 172 apr 21 14:52 main.cpp
-rw-rw-r-- 1 krajsz krajsz 219 apr 21 14:52 mainwindow.cpp
-rw-rw-r-- 1 krajsz krajsz 291 apr 21 14:52 mainwindow.h
-rw-rw-r-- 1 krajsz krajsz 630 apr 21 14:52 mainwindow.ui
-rw-rw-r-- 1 krajsz krajsz 354 apr 21 15:21 testProgram.pro
```

Láthatjuk, hogy generálva lett egy testProgram.pro nevű állomány, aminek a tartalma a következő:

```
krajsz@krajsz-Lenovo-G500s:~/ $ cat testProgram.pro

TEMPLATE = app
TARGET = testProgram
INCLUDEPATH += .
# Input
HEADERS += mainwindow.h
FORMS += mainwindow.ui
SOURCES += main.cpp mainwindow.cpp
```

Mivel a qmake nem csak Qt- s projektekhez használható, így, mint jelen esetben is, kiegészítésre szorul a generált projekt fájl. Jelen esetben ahhoz, hogy a megfelelő fájlokat kapcsolja a make, a kövező sorral kell kiegészítsük a projekt fájlunkat:

```
QT+= widgets
```

Ezután futtatjuk a qmake- et, átadva neki a generált projekt fájlunkat:

```
krajsz@krajsz-Lenovo-G500s:~/ $ qmake testProgram.pro
```

Ha nem kapunk semmi féle hibaüzenetet, akkor azt jelenti, sikeresen generálva lett egy Makefile. Láthatjuk, hogy mekkora segítség ez számunkra, mivel egy hét soros projekt fájlból egy 537 soros Makefile- et generált a qmake:

```
krajsz@krajsz-Lenovo-G500s:~/$ wc -l Makefile
537 Makefile
```

Ezután a make- et futtatva láthatjuk a fordítás menetét:

```
krajsz@krajsz-Lenovo-G500s:~/$ make
/usr/lib/x86_64-linux-gnu/qt5/bin/uic          mainwindow.ui          -o
ui_mainwindow.h
g++ -c -m64 -pipe -O2 -Wall -W -D_REENTRANT -fPIC -DQT_NO_DEBUG
-DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I. -I. -isystem
/usr/include/x86_64-linux-gnu/qt5 -isystem /usr/include/x86_64-
linux-gnu/qt5/QtWidgets -isystem /usr/include/x86_64-linux-
gnu/qt5/QtGui -isystem /usr/include/x86_64-linux-gnu/qt5/QtCore -I.
-I. -I/usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++-64 -o main.o
main.cpp
...
```

Ha hiba nélkül lefut a make, elkészült a futtatható állományunk.

2.1.1.2 Modulok

2.1.1.3 Slot-signal mechanizmus

2.1.1.4 Qt Creator

2.1.2 OpenCV

2.1.2.1 Telepítés

2.1.3 BGSLib

2.2 Előzmények

2.2.1 <megelőző forgalomszámáló szoftverről>

2.3 A Robotautó Világbajnokság

2.3.1 A platform koncepciója

2.3.2 \diamond

2.4 A szoftver elkészítésének folyamata

2.4.1 Tervezés

2.4.2 Fejlesztés

2.4.2.1 Videó források típusok

2.4.2.2 Adatbázis kapcsolatok kialakítása, kezelése

2.4.2.3 Szálkezelés

2.4.2.4 Videó feldolgozása

2.4.2.5 Járművek detektálása

2.4.3 Problémák a szoftver fejlesztése során

2.4.4 A szoftver tesztelése

2.4.5 Eredmények

2.5 A szoftver használata

2.5.1 Fordítása

2.5.2 Indítása

2.5.3 A grafikus felhasználói felület használata

2.5.3.1 Új forrás kiválasztása

2.5.3.2 Adatbázisok kezelése

2.5.3.3 Fájl videó forrás információi

2.5.4 A parancssoros felhasználói felület használata

2.5.4.1 Használható parancsok

3. Összefoglalás

4. Köszönetnyilvánítás

5. Irodalomjegyzék