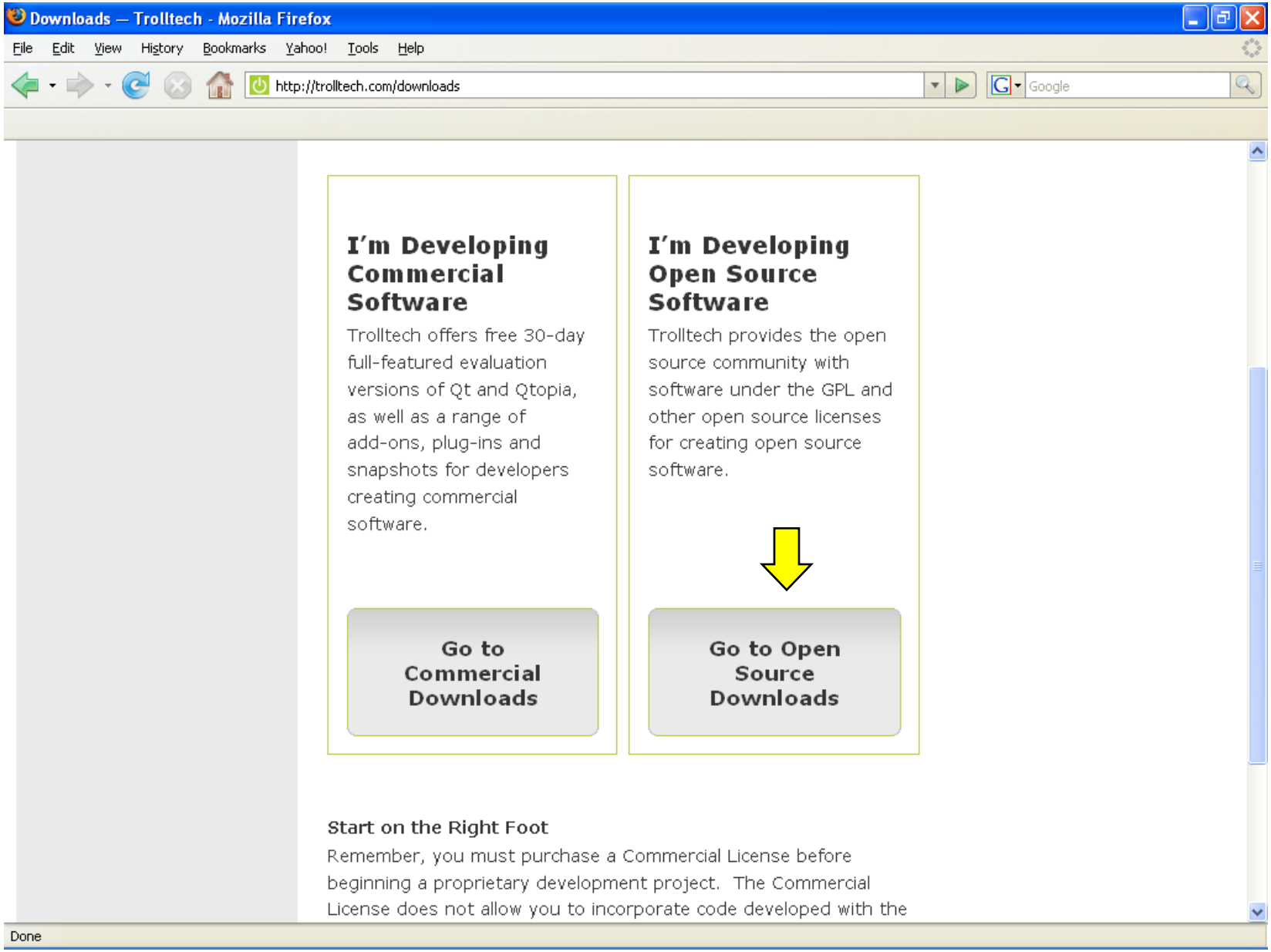
ECE 462 C++ and Java

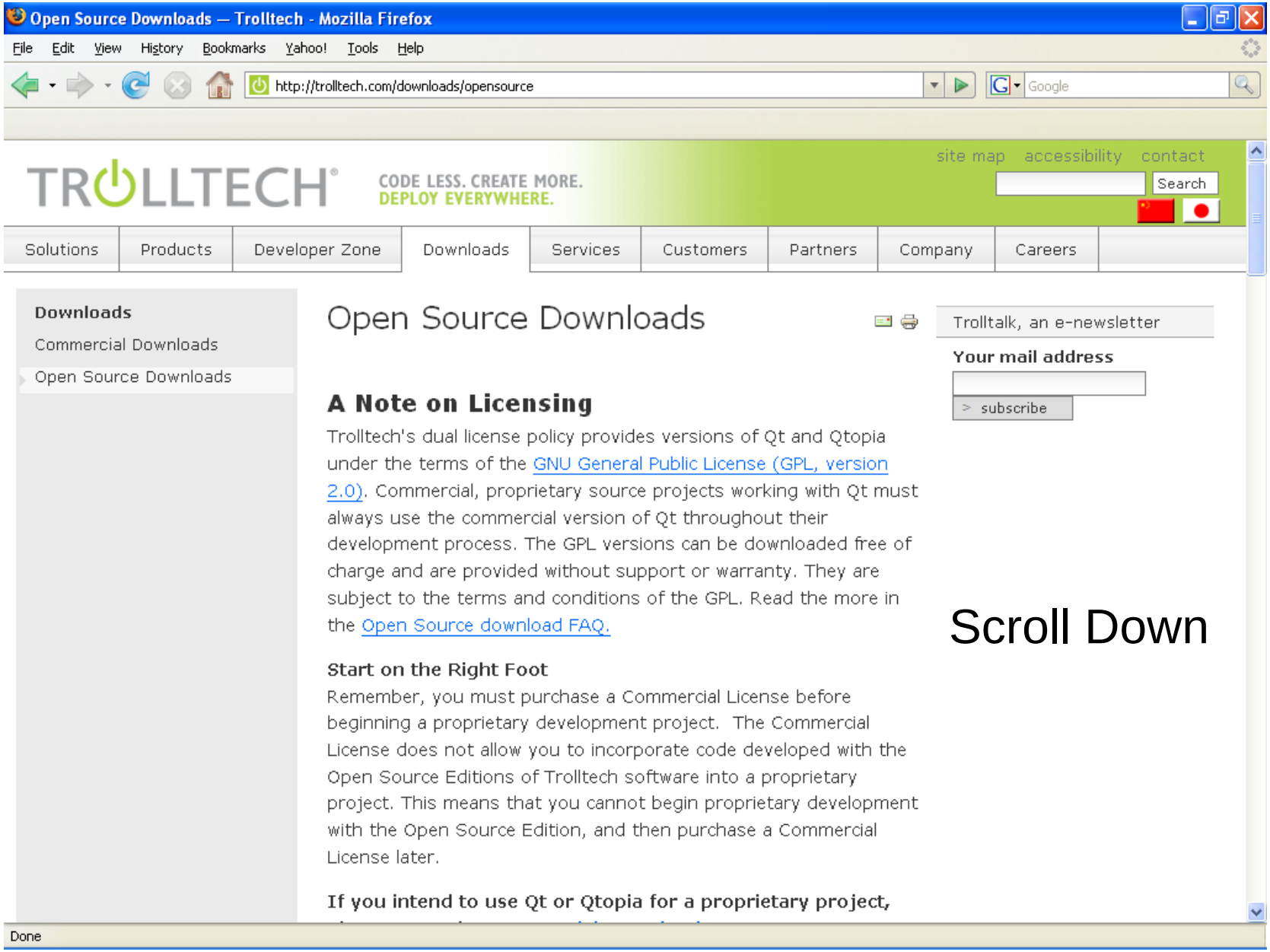
Lab Exercise 03
Graphical User Interface using C++ and Qt

Yung-Hsiang Lu
yunglu@purdue.edu

Install Qt in Your Computer


Code Less, Create More — Trolltech - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://trolltech.com/
Google
TROLLTECH
CODE LESS. CREATE MORE.
DEPLOY EVERYWHERE.
site map accessibility contact
Search
Solutions Products Developer Zone Downloads Services Customers Partners Company Careers
Trolltech offers solutions fo
Cross-Platform Developers
Industrial Embedded Developers
Consumer Electronics & Mobile Device Vendors
Mobile Application Developers
Software Development Managers
Open Source Developers
Discover how our CUSTOMER
are su
Qt
News
Trolltech and Red Benc
Over-the-Air Software
Management to Linux
ICS and Froglogic Form
Provide Superior Testir
Developers (Jan 22)
Trolltech's Qt to be lice
(Jan 18)
Trolltech Congratulatin
(Jan 11)
Zi Corporation License
to Trolltech (Jan 10)
More news
Qt JAMBI GETS HIGH MARKS FOR JAVA DEVELOPMENT
"... DELIVERS EXCEPTIONAL EASE OF DEVELOPMENT" – InfoWorld
READ THE FULL REVIEW
Qt
Qt is a cross-platform application framework. It includes:
DOWNLOAD NOW
Qtopia
Qtopia is an application platform and UI for Linux-based mobile, CE and embedded devices. Qtopia
LEARN MORE
Done

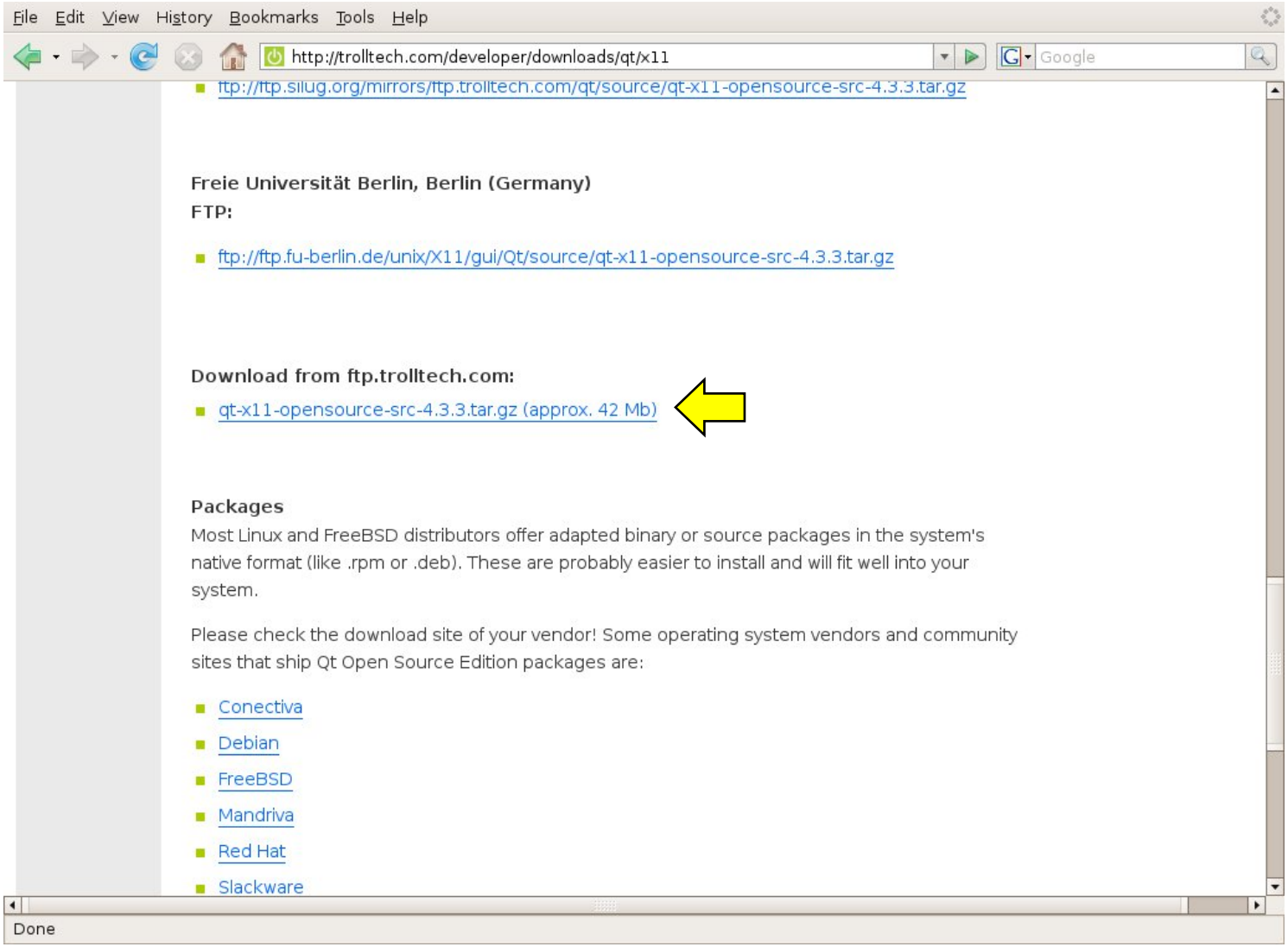

Downloads — Trolltech - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://trolltech.com/downloads
Google
I'm Developing Commercial Software
Trolltech offers free 30-day full-featured evaluation versions of Qt and Qtopia, as well as a range of add-ons, plug-ins and snapshots for developers creating commercial software.
Go to Commercial Downloads
I'm Developing Open Source Software
Trolltech provides the open source community with software under the GPL and other open source licenses for creating open source software.
Go to Open Source Downloads
Start on the Right Foot
Remember, you must purchase a Commercial License before beginning a proprietary development project. The Commercial License does not allow you to incorporate code developed with the
Done

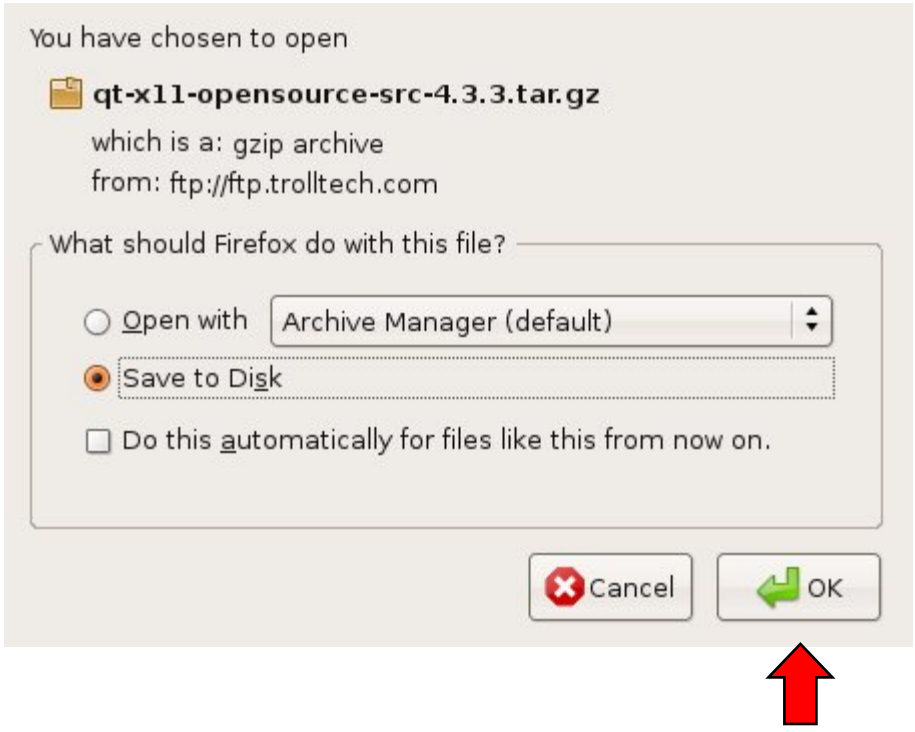

Open Source Downloads — Trolltech - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://trolltech.com/downloads/opensource
Google
TROLLTECH
CODE LESS. CREATE MORE. DEPLOY EVERYWHERE.
site map accessibility contact
Search
Solutions Products Developer Zone Downloads Services Customers Partners Company Careers
Downloads
Commercial Downloads
Open Source Downloads
Open Source Downloads
A Note on Licensing
Trolltech's dual license policy provides versions of Qt and Qtopia under the terms of the GNU General Public License (GPL, version 2.0). Commercial, proprietary source projects working with Qt must always use the commercial version of Qt throughout their development process. The GPL versions can be downloaded free of charge and are provided without support or warranty. They are subject to the terms and conditions of the GPL. Read the more in the Open Source download FAQ.
Start on the Right Foot
Remember, you must purchase a Commercial License before beginning a proprietary development project. The Commercial License does not allow you to incorporate code developed with the Open Source Editions of Trolltech software into a proprietary project. This means that you cannot begin proprietary development with the Open Source Edition, and then purchase a Commercial License later.
If you intend to use Qt or Qtopia for a proprietary project,
Trolltalk, an e-newsletter
Your mail address
subscribe
Scroll Down
Done

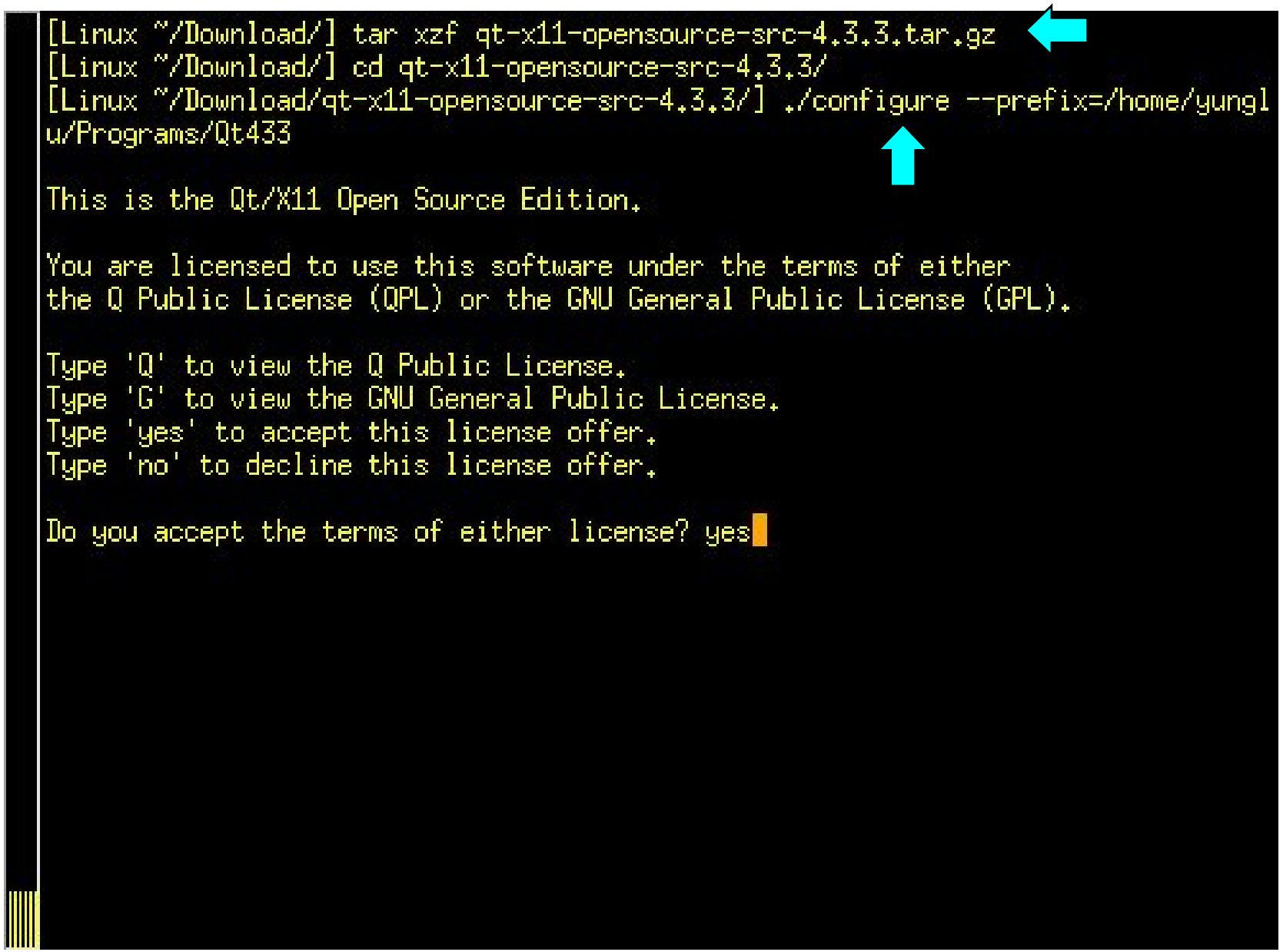
Install Qt in Linux


Open Source Downloads — Trolltech - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://trolltech.com/downloads/opensource
Google
Qt Open Source Edition for C++ Developers
download
description
Qt Open Source Edition for C++ Developers: Windows Download
Qt is a cross-platform application development framework. The Open Source Edition of Qt is available to open source developers under the terms of the GPL version 2.0, and is not intended for commercial, proprietary source projects.
Qt Open Source Edition for C++ Developers: Mac Download
Qt is a cross-platform application development framework. The Open Source Edition of Qt is available to open source developers under the terms of the GPL version 2.0, and is not intended for commercial, proprietary source projects.
for Linux
Qt Open Source Edition for C++ Developers: X11 Download
Qt is a cross-platform application development framework. The Open Source Edition of Qt is available to open source developers under the terms of the GPL version 2.0, and is not intended for commercial, proprietary source projects.
Provide Feedback on Pre-Release Software
download
description
Qt 4.4 Technology Preview
Done

Install in Linux




You have chosen to open
qt-x11-opensource-src-4.3.3.tar.gz
which is a: gzip archive
from: ftp://ftp.trolltech.com
What should Firefox do with this file?
Open with
Archive Manager (default)
Save to Disk
Do this automatically for files like this from now on.
Cancel
OK


[Linux ~/Download/] tar xzf qt-x11-opensource-src-4.3.3.tar.gz
[Linux ~/Download/] cd qt-x11-opensource-src-4.3.3/
[Linux ~/Download/qt-x11-opensource-src-4.3.3/] ./configure --prefix=/home/yungl
u/Programs/Qt433
This is the Qt/X11 Open Source Edition.
You are licensed to use this software under the terms of either
the Q Public License (QPL) or the GNU General Public License (GPL).
Type 'Q' to view the Q Public License.
Type 'G' to view the GNU General Public License.
Type 'yes' to accept this license offer.
Type 'no' to decline this license offer.
Do you accept the terms of either license? yes

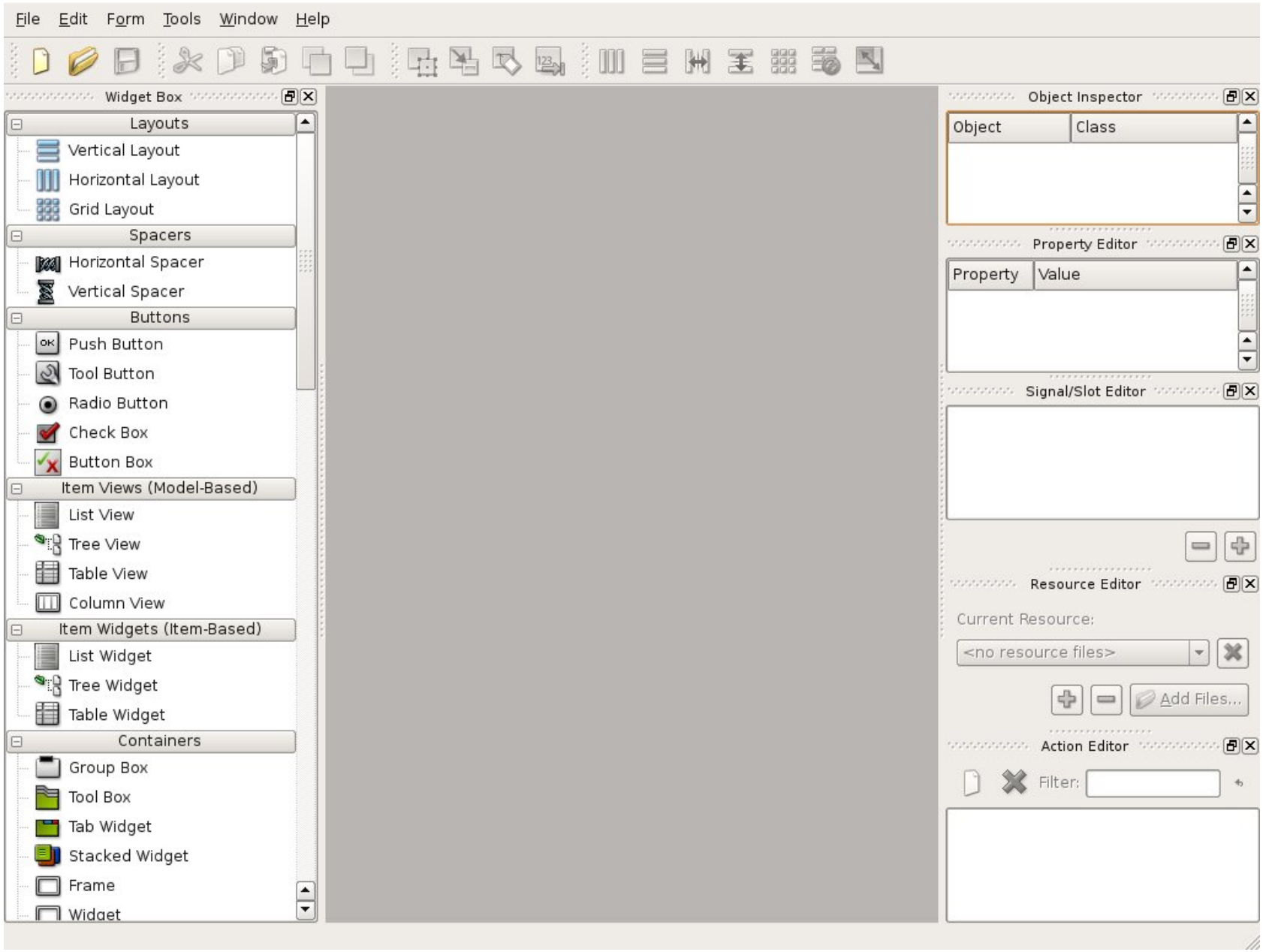
```
  for /home/yunglu/Download/qt-x11-opensource-src-4.3.3/src/plugins/codecs/cn/cn
.pro
  for /home/yunglu/Download/qt-x11-opensource-src-4.3.3/src/plugins/codecs/jp/jp
.pro
  for /home/yunglu/Download/qt-x11-opensource-src-4.3.3/src/plugins/codecs/tw/tw
.pro
  for /home/yunglu/Download/qt-x11-opensource-src-4.3.3/src/plugins/codecs/kr/kr
.pro
  for /home/yunglu/Download/qt-x11-opensource-src-4.3.3/src/plugins/imageformats
/gif/gif.pro
  for /home/yunglu/Download/qt-x11-opensource-src-4.3.3/src/plugins/imageformats
/tiff/tiff.pro
  for /home/yunglu/Download/qt-x11-opensource-src-4.3.3/src/plugins/imageformats
/jpeg/jpeg.pro
  for /home/yunglu/Download/qt-x11-opensource-src-4.3.3/src/plugins/imageformats
/svg/svg.pro
  for /home/yunglu/Download/qt-x11-opensource-src-4.3.3/src/plugins/imageformats
/mng/mng.pro
  for /home/yunglu/Download/qt-x11-opensource-src-4.3.3/src/plugins/inputmethods
/imsw-multi/imsw-multi.pro

Qt is now configured for building. Just run 'make'.
Once everything is built, you must run 'make install'.
Qt will be installed into /home/yunglu/Programs/Qt433

To reconfigure, run 'make confclean' and 'configure'.

[Linux ~/Download/qt-x11-opensource-src-4.3.3/]
```

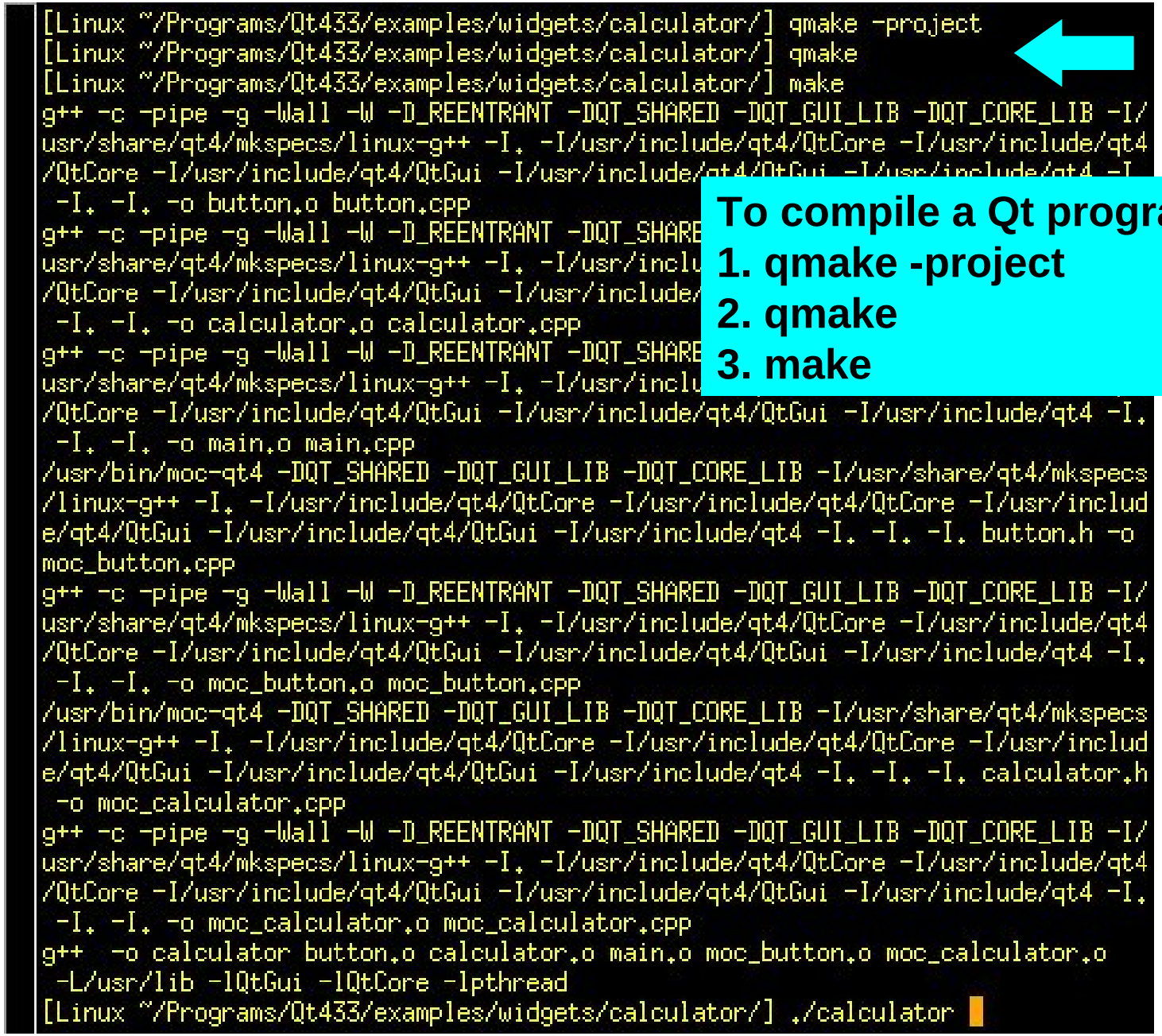
```
emo.cpp
g++ -Wl,-rpath,/home/yunglu/Programs/Qt433/lib -Wl,-rpath,/home/yunglu/Programs/
Qt433/lib -o ../../bin/qtdemo .obj-release-shared/main.o .obj-release-shared/dem
oscene.o .obj-release-shared/mainwindow.o .obj-release-shared/demoitem.o .obj-re
lease-shared/score.o .obj-release-shared/demoitemanimation.o .obj-release-shared
/itemcircleanimation.o .obj-release-shared/demotextitem.o .obj-release-shared/he
adingitem.o .obj-release-shared/dockitem.o .obj-release-shared/scanitem.o .obj-r
elease-shared/letteritem.o .obj-release-shared/examplecontent.o .obj-release-sha
red/menucontent.o .obj-release-shared/guide.o .obj-release-shared/guideline.o .o
bj-release-shared/guidecircle.o .obj-release-shared/menumanager.o .obj-release-s
hared/colors.o .obj-release-shared/textbutton.o .obj-release-shared/imageitem.o
.obj-release-shared/moc_mainwindow.o .obj-release-shared/moc_demoitemanimation.o
 .obj-release-shared/moc_menumanager.o .obj-release-shared/qrc_qtdemo.o    -L/ho
me/yunglu/Download/qt-x11-opensource-src-4.3.3/lib -L/usr/X11R6/lib -lQtAssistan
tClient -lQtXml -L/home/yunglu/Download/qt-x11-opensource-src-4.3.3/lib -pthread
 -pthread -lQtOpenGL -L/usr/X11R6/lib -pthread -pthread -pthread -pthread -pthre
ad -pthread -lQtGui -lpng -lSM -lICE -pthread -pthread -lXi -lXrender -lXrandr -
lXfixes -lXcursor -lXinerama -lfreetype -lfontconfig -lXext -lX11 -lQtNetwork -p
thread -pthread -lQtCore -lz -lm -pthread -lgthread-2.0 -lrt -lglib-2.0 -ldl -lG
LU -lGL -lpthread
(test -z "../../bin/" || cd "../../bin/" ; targ=`basename ../../bin/qtdemo`; obj
copy --only-keep-debug "$targ" "$targ.debug" && objcopy --strip-debug "$targ" &&
 objcopy --add-gnu-debuglink="$targ.debug" "$targ" && chmod -x "$targ.debug" ) ;
make[2]: Leaving directory `/home/yunglu/Download/qt-x11-opensource-src-4.3.3/de
mos/qtdemo'
make[1]: Leaving directory `/home/yunglu/Download/qt-x11-opensource-src-4.3.3/de
mos'
[Linux ~/Download/qt-x11-opensource-src-4.3.3/] make install
```

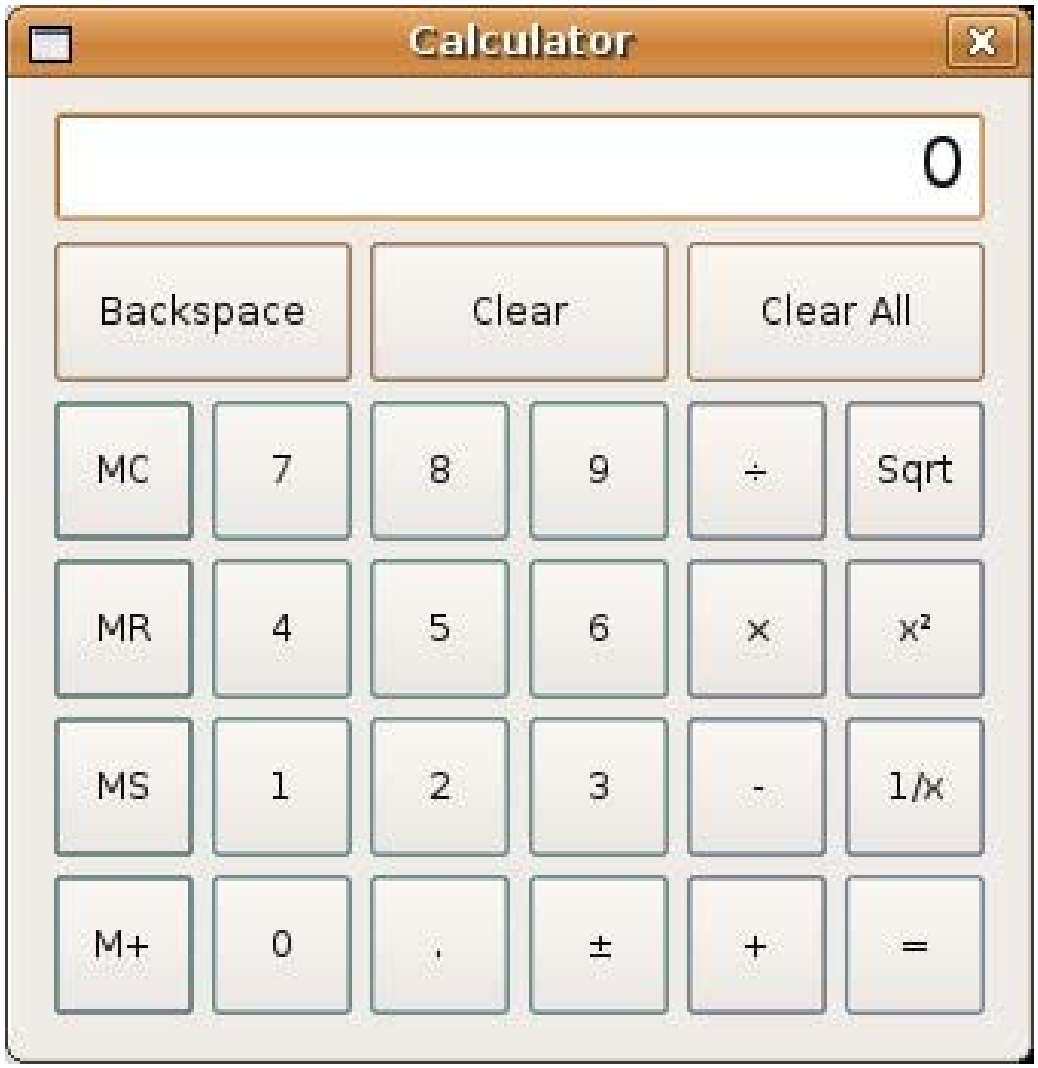
```
/guidecircle.h /home/yunglu/Programs/Qt433/demos/qtdemo/
install -m 644 -p /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo
/menumanager.h /home/yunglu/Programs/Qt433/demos/qtdemo/
install -m 644 -p /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo
/colors.h /home/yunglu/Programs/Qt433/demos/qtdemo/
install -m 644 -p /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo
/textbutton.h /home/yunglu/Programs/Qt433/demos/qtdemo/
install -m 644 -p /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo
/imageitem.h /home/yunglu/Programs/Qt433/demos/qtdemo/
install -m 644 -p /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo
/qtdemo.qrc /home/yunglu/Programs/Qt433/demos/qtdemo/
install -m 644 -p /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo
/qtdemo.pro /home/yunglu/Programs/Qt433/demos/qtdemo/
cp -f -r /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo/images /
home/yunglu/Programs/Qt433/demos/qtdemo/
cp -f -r /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo/xml /hom
e/yunglu/Programs/Qt433/demos/qtdemo/
install -m 644 -p /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo
/qtdemo.ico /home/yunglu/Programs/Qt433/demos/qtdemo/
install -m 644 -p /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo
/qtdemo.icns /home/yunglu/Programs/Qt433/demos/qtdemo/
install -m 644 -p /home/yunglu/Download/qt-x11-opensource-src-4.3.3/demos/qtdemo
/qtdemo.rc /home/yunglu/Programs/Qt433/demos/qtdemo/
make[2]: Leaving directory `/home/yunglu/Download/qt-x11-opensource-src-4.3.3/de
mos/qtdemo'
make[1]: Leaving directory `/home/yunglu/Download/qt-x11-opensource-src-4.3.3/de
mos'
[Linux ~/Download/qt-x11-opensource-src-4.3.3/]
```

installation complete


[Linux ~/Programs/Qt433/bin/] ./designer
execute Qt designer

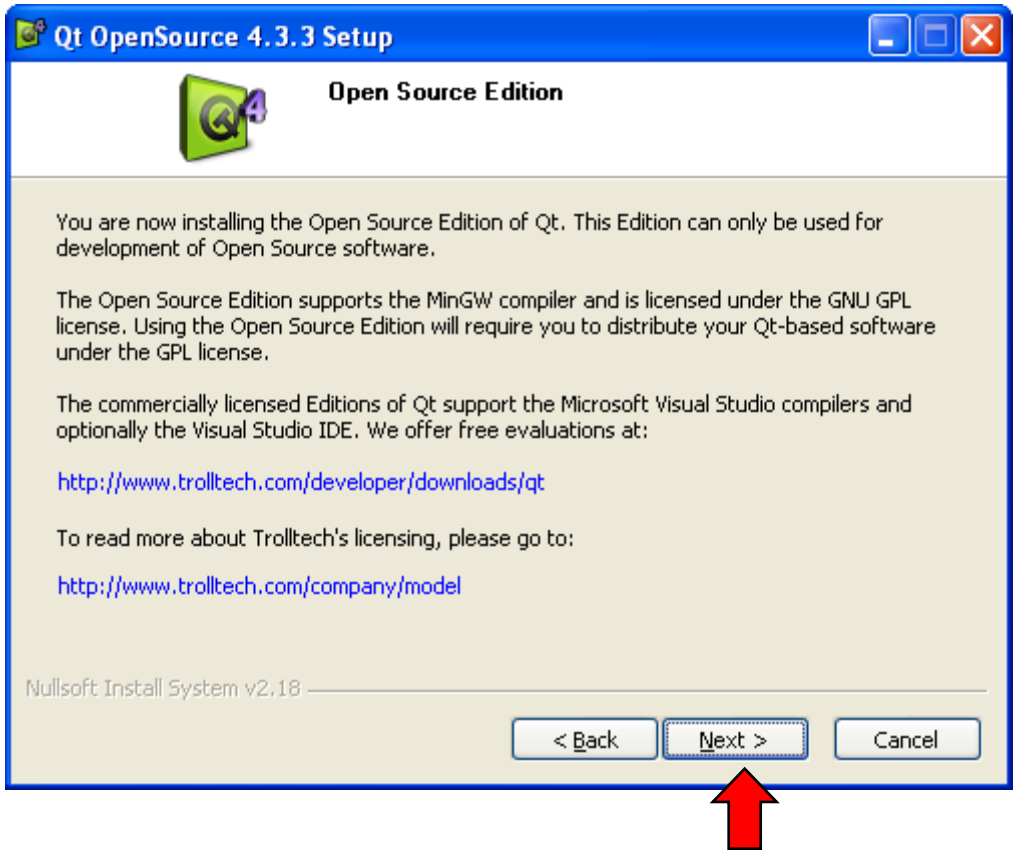

File
Edit
Form
Tools
Window
Help
Widget Box
Layouts
Vertical Layout
Horizontal Layout
Grid Layout
Spacers
Horizontal Spacer
Vertical Spacer
Buttons
Push Button
Tool Button
Radio Button
Check Box
Button Box
Item Views (Model-Based)
List View
Tree View
Table View
Column View
Item Widgets (Item-Based)
List Widget
Tree Widget
Table Widget
Containers
Group Box
Tool Box
Tab Widget
Stacked Widget
Frame
Widget
Object Inspector
Object
Class
Property Editor
Property
Value
Signal/Slot Editor
Resource Editor
Current Resource:
<no resource files>
Add Files...
Action Editor
Filter:

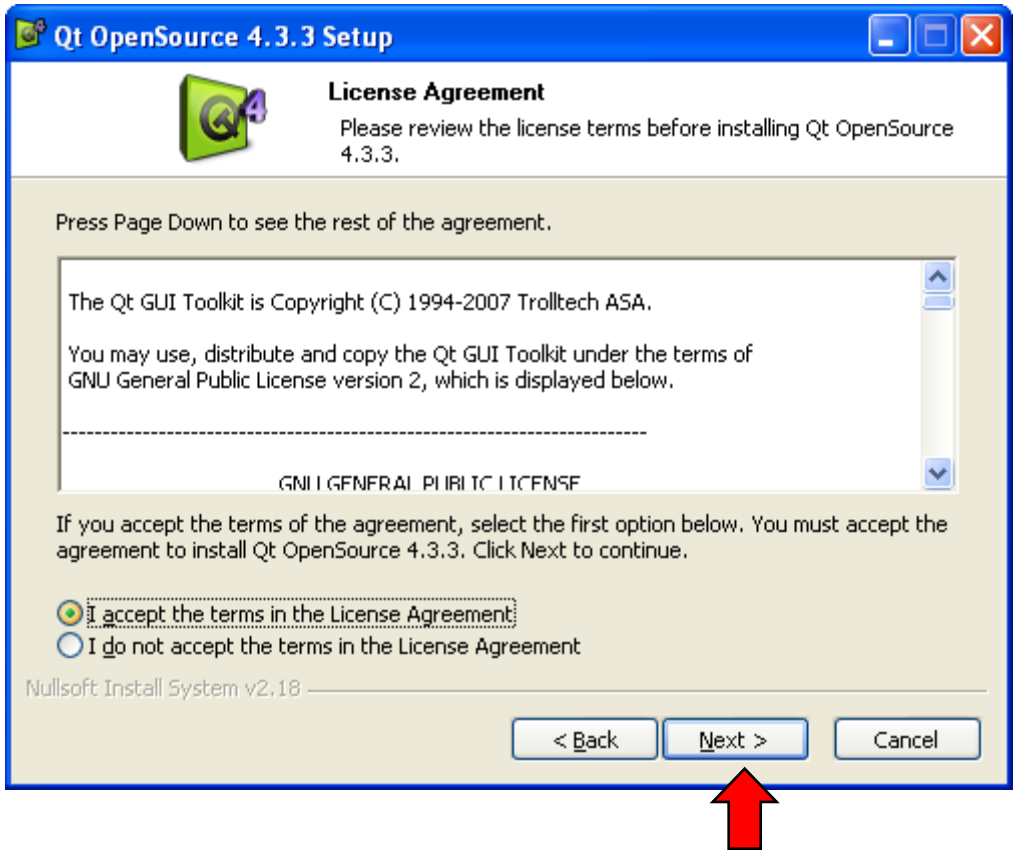

[Linux ~/Programs/Qt433/examples/widgets/calculator/] qmake -project
[Linux ~/Programs/Qt433/examples/widgets/calculator/] qmake
[Linux ~/Programs/Qt433/examples/widgets/calculator/] make
g++ -c -pipe -g -Wall -W -D_REENTRANT -DQT_SHARED -DQT_GUI_LIB -DQT_CORE_LIB -I/usr/share/qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/usr/include/qt4/QtGui -I/usr/include/qt4 -I. -I. -I. -o button.o button.cpp
g++ -c -pipe -g -Wall -W -D_REENTRANT -DQT_SHARED -DQT_GUI_LIB -DQT_CORE_LIB -I/usr/share/qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/usr/include/qt4/QtGui -I/usr/include/qt4 -I. -I. -I. -o calculator.o calculator.cpp
g++ -c -pipe -g -Wall -W -D_REENTRANT -DQT_SHARED -DQT_GUI_LIB -DQT_CORE_LIB -I/usr/share/qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/usr/include/qt4/QtGui -I/usr/include/qt4 -I. -I. -I. -o main.o main.cpp
/usr/bin/moc-qt4 -DQT_SHARED -DQT_GUI_LIB -DQT_CORE_LIB -I/usr/share/qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/usr/include/qt4/QtGui -I/usr/include/qt4 -I. -I. -I. button.h -o moc_button.cpp
g++ -c -pipe -g -Wall -W -D_REENTRANT -DQT_SHARED -DQT_GUI_LIB -DQT_CORE_LIB -I/usr/share/qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/usr/include/qt4/QtGui -I/usr/include/qt4 -I. -I. -I. -o moc_button.o moc_button.cpp
/usr/bin/moc-qt4 -DQT_SHARED -DQT_GUI_LIB -DQT_CORE_LIB -I/usr/share/qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/usr/include/qt4/QtGui -I/usr/include/qt4 -I. -I. -I. calculator.h -o moc_calculator.cpp
g++ -c -pipe -g -Wall -W -D_REENTRANT -DQT_SHARED -DQT_GUI_LIB -DQT_CORE_LIB -I/usr/share/qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/usr/include/qt4/QtGui -I/usr/include/qt4 -I. -I. -I. -o moc_calculator.o moc_calculator.cpp
g++ -o calculator button.o calculator.o main.o moc_button.o moc_calculator.o -L/usr/lib -lQtGui -lQtCore -lpthread
[Linux ~/Programs/Qt433/examples/widgets/calculator/] ./calculator
To compile a Qt program:
1. qmake -project
2. qmake
3. make

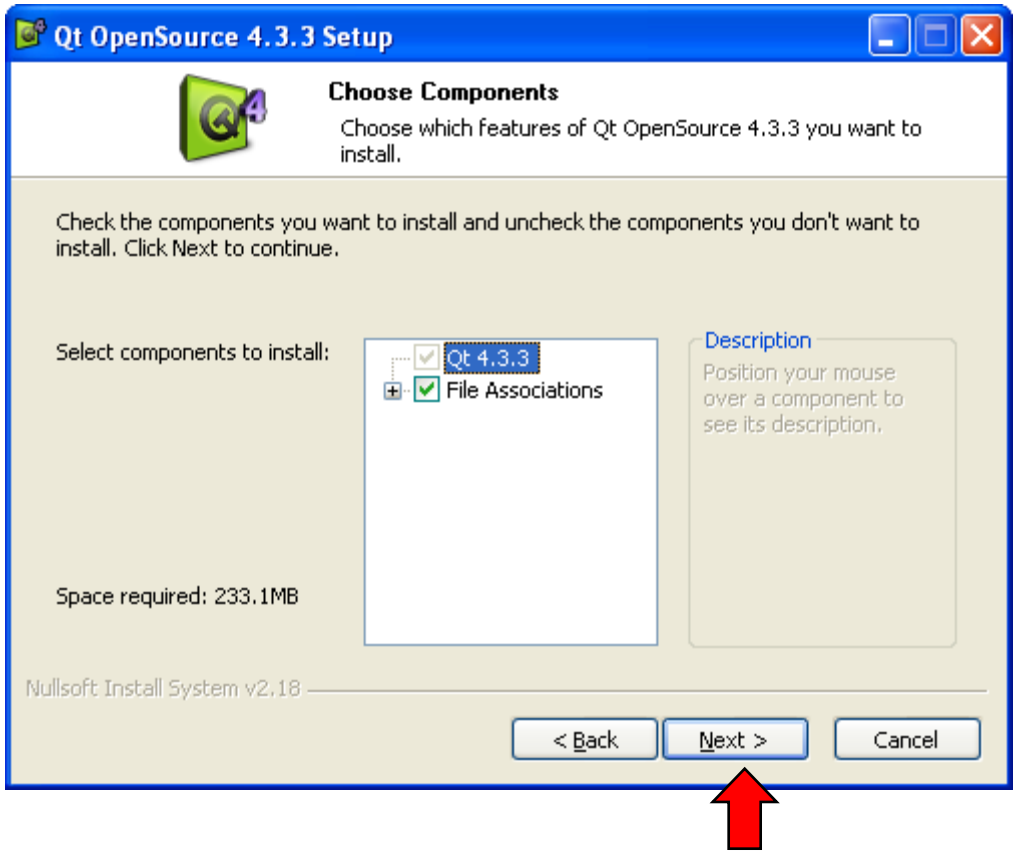

Calculator
0
Backspace
Clear
Clear All
MC
7
8
9
÷
Sqrt
MR
4
5
6
×
x²
MS
1
2
3
-
1/x
M+
0
,
±
+
=

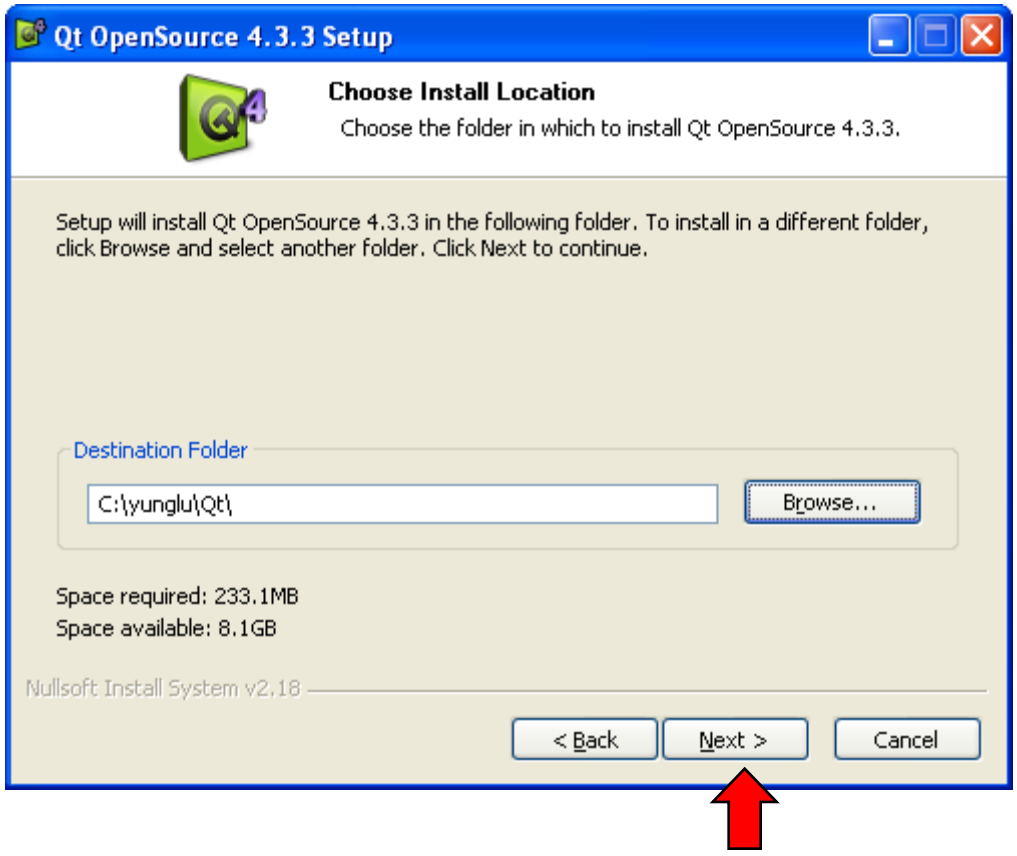
Install Qt in Windows

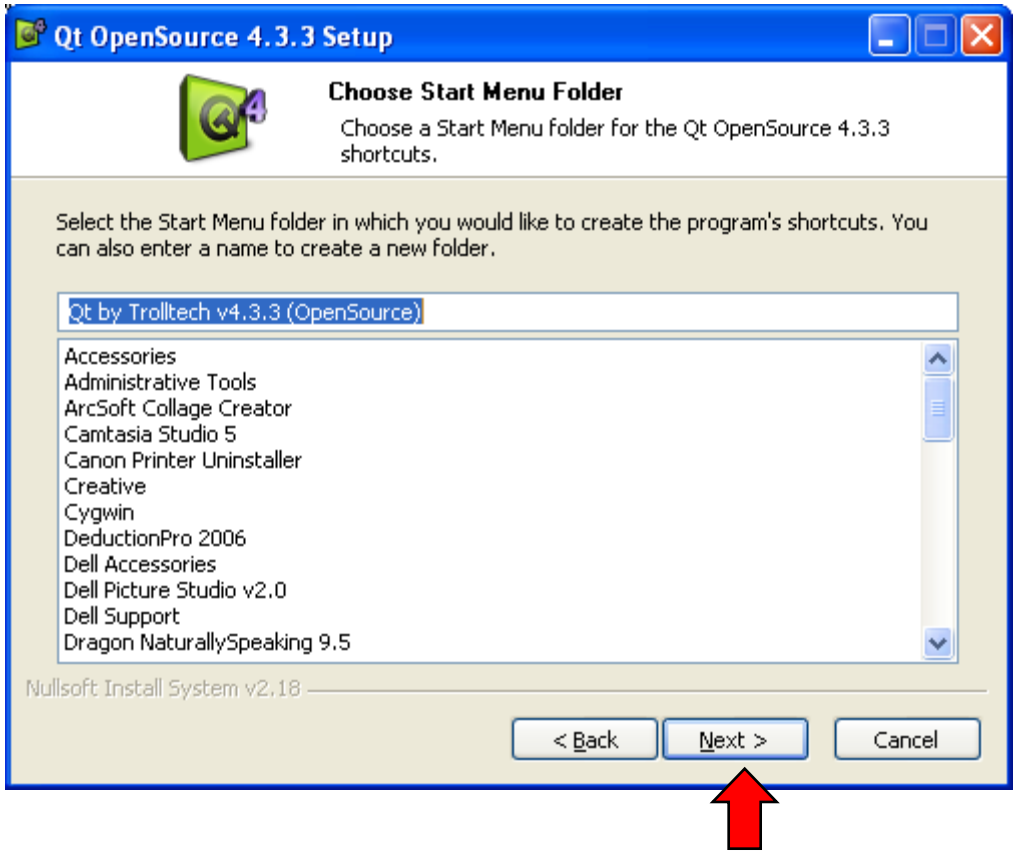
Install in Windows

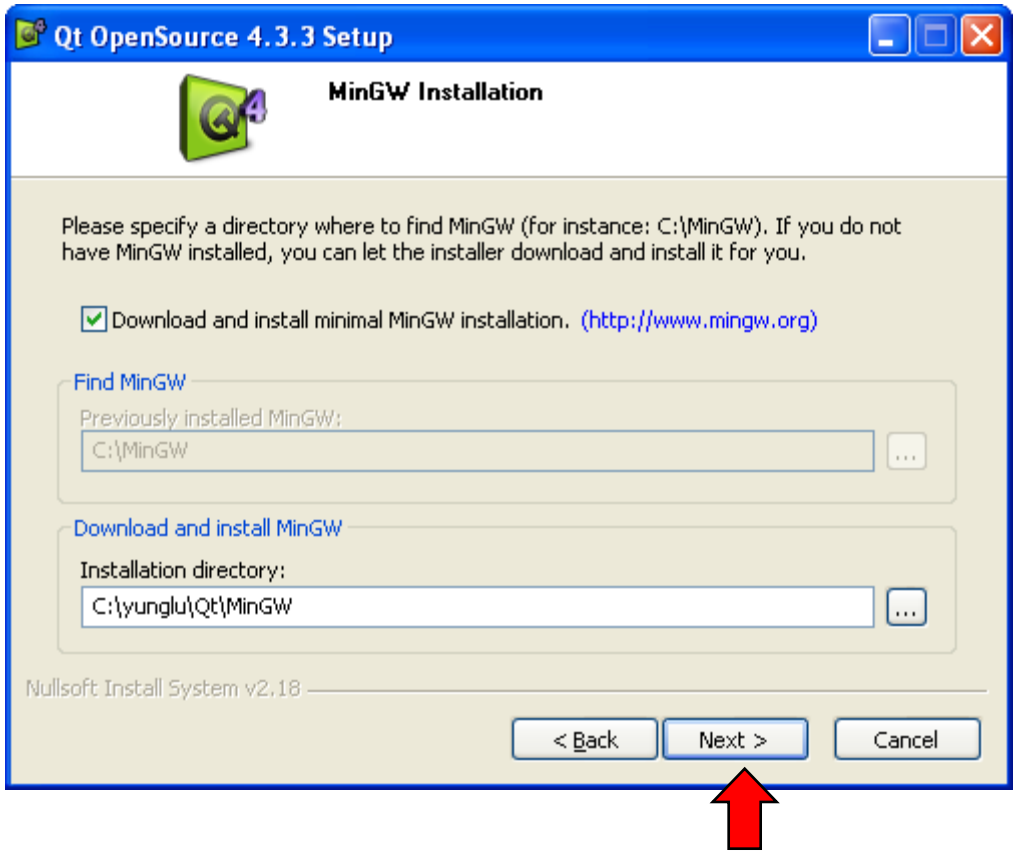

Qt OpenSource 4.3.3 Setup
Open Source Edition
You are now installing the Open Source Edition of Qt. This Edition can only be used for development of Open Source software.
The Open Source Edition supports the MinGW compiler and is licensed under the GNU GPL license. Using the Open Source Edition will require you to distribute your Qt-based software under the GPL license.
The commercially licensed Editions of Qt support the Microsoft Visual Studio compilers and optionally the Visual Studio IDE. We offer free evaluations at:
http://www.trolltech.com/developer/downloads/qt
To read more about Trolltech's licensing, please go to:
http://www.trolltech.com/company/model
Nullsoft Install System v2.18
< Back
Next >
Cancel

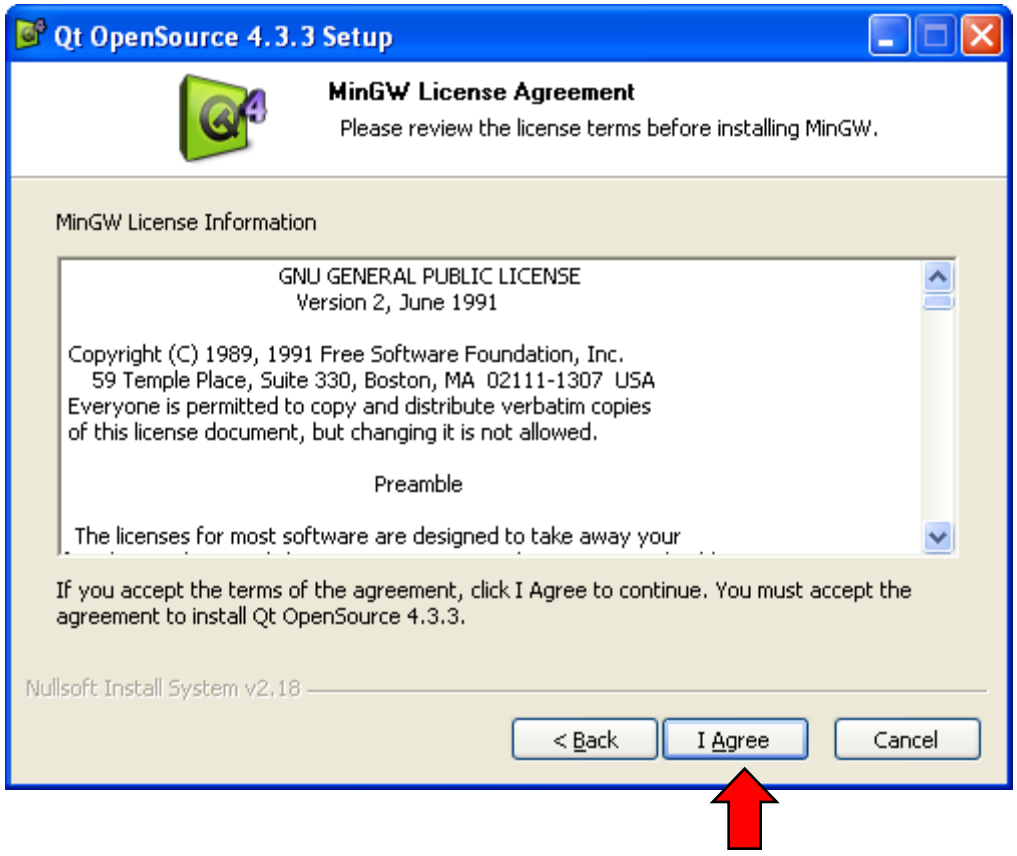

Qt OpenSource 4.3.3 Setup
License Agreement
Please review the license terms before installing Qt OpenSource 4.3.3.
Press Page Down to see the rest of the agreement.
The Qt GUI Toolkit is Copyright (C) 1994-2007 Trolltech ASA.
You may use, distribute and copy the Qt GUI Toolkit under the terms of GNU General Public License version 2, which is displayed below.
GNU GENERAL PUBLIC LICENSE
If you accept the terms of the agreement, select the first option below. You must accept the agreement to install Qt OpenSource 4.3.3. Click Next to continue.
I accept the terms in the License Agreement
I do not accept the terms in the License Agreement
Nullsoft Install System v2.18
< Back
Next >
Cancel

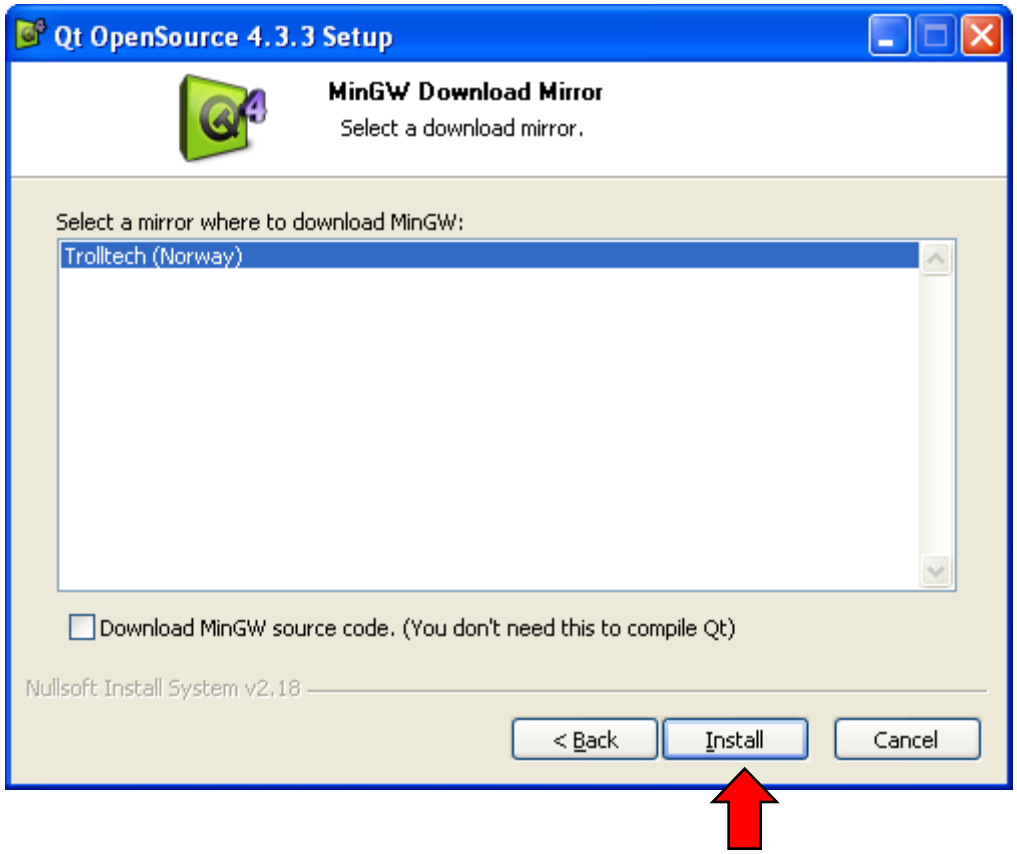

Qt OpenSource 4.3.3 Setup
Choose Components
Choose which features of Qt OpenSource 4.3.3 you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
Qt 4.3.3
File Associations
Description
Position your mouse over a component to see its description.
Space required: 233.1MB
Nullsoft Install System v2.18
< Back
Next >
Cancel

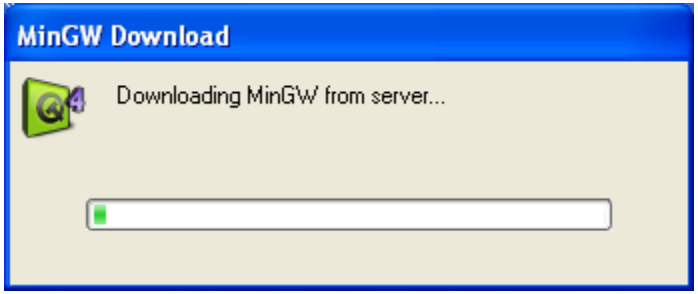

Qt OpenSource 4.3.3 Setup
Choose Install Location
Choose the folder in which to install Qt OpenSource 4.3.3.
Setup will install Qt OpenSource 4.3.3 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\yunglu\Qt\
Browse...
Space required: 233.1MB
Space available: 8.1GB
Nullsoft Install System v2.18
< Back
Next >
Cancel

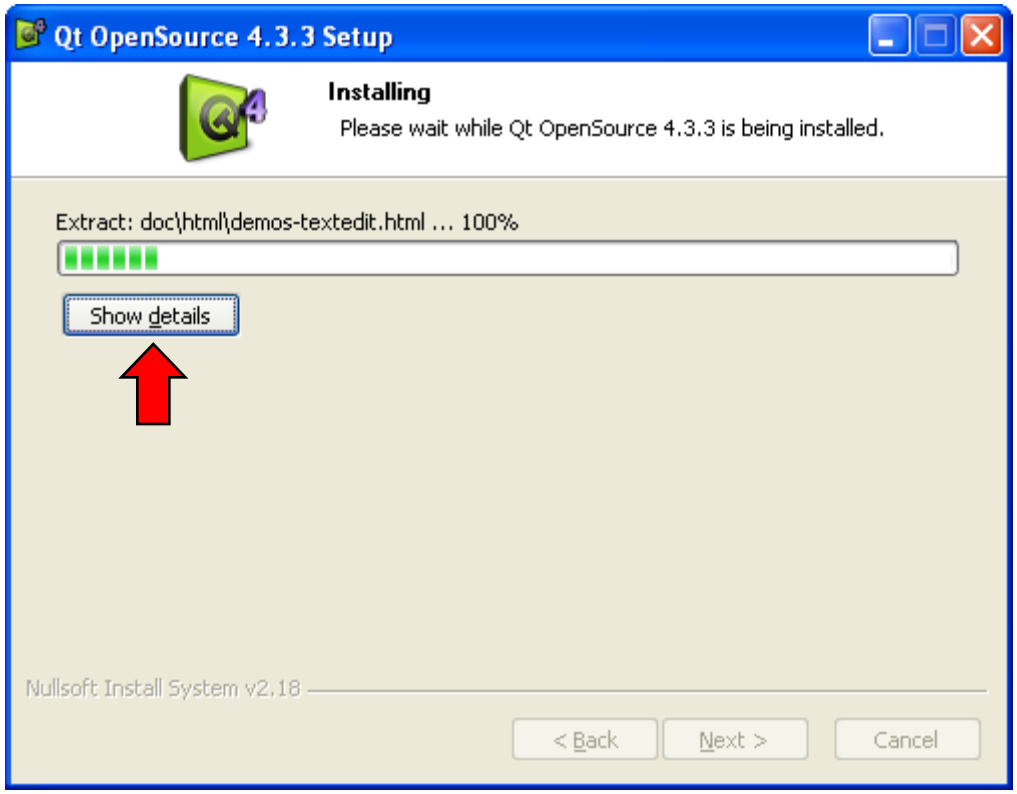

Qt OpenSource 4.3.3 Setup
Choose Start Menu Folder
Choose a Start Menu folder for the Qt OpenSource 4.3.3 shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
Qt by Trolltech v4.3.3 (OpenSource)
Accessories
Administrative Tools
ArcSoft Collage Creator
Camtasia Studio 5
Canon Printer Uninstaller
Creative
Cygwin
DeductionPro 2006
Dell Accessories
Dell Picture Studio v2.0
Dell Support
Dragon NaturallySpeaking 9.5
Nullsoft Install System v2.18
< Back
Next >
Cancel

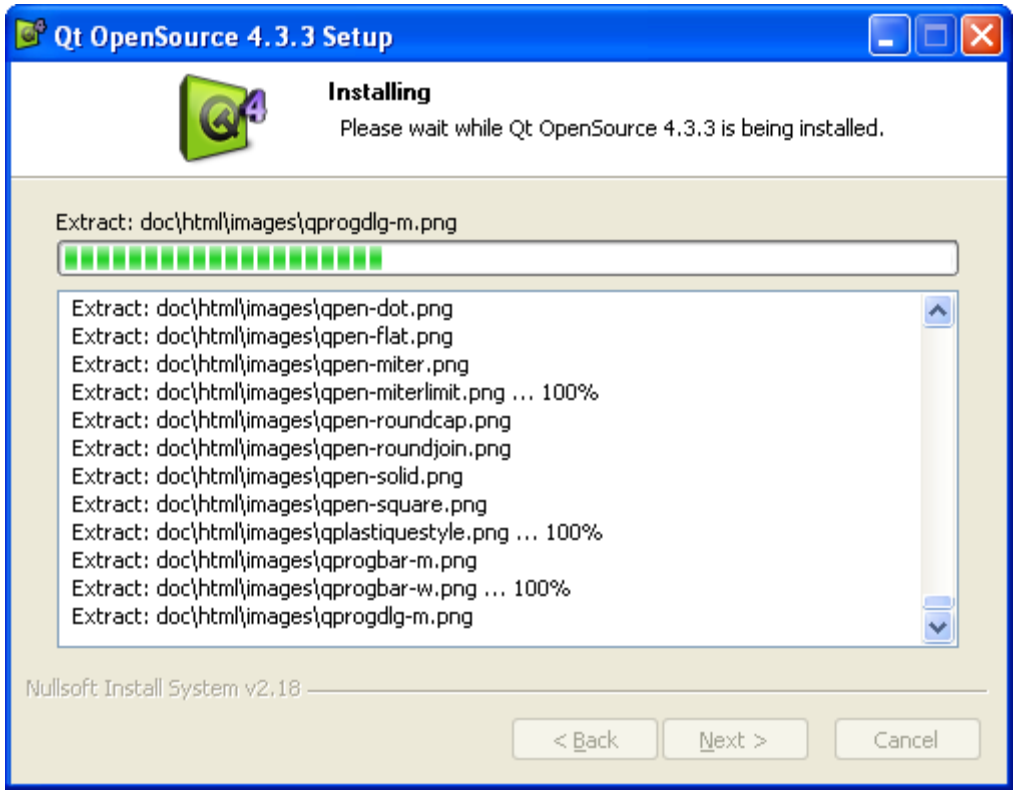

Qt OpenSource 4.3.3 Setup
MinGW Installation
Please specify a directory where to find MinGW (for instance: C:\MinGW). If you do not have MinGW installed, you can let the installer download and install it for you.
Download and install minimal MinGW installation. (http://www.mingw.org)
Find MinGW
Previously installed MinGW:
C:\MinGW
Download and install MinGW
Installation directory:
C:\yunglu\Qt\MinGW
Nullsoft Install System v2.18
< Back
Next >
Cancel

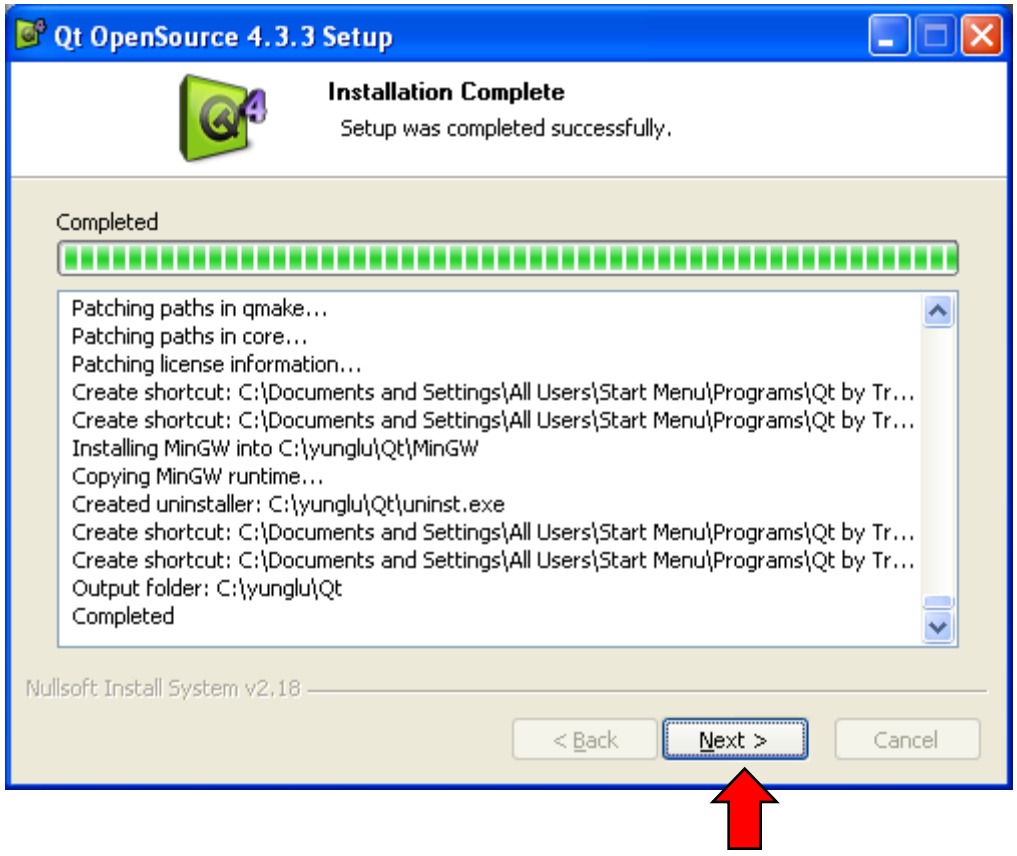

Qt OpenSource 4.3.3 Setup
MinGW License Agreement
Please review the license terms before installing MinGW.
MinGW License Information
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The licenses for most software are designed to take away your
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Qt OpenSource 4.3.3.
Nullsoft Install System v2.18
< Back
I Agree
Cancel


Qt OpenSource 4.3.3 Setup
MinGW Download Mirror
Select a download mirror.
Select a mirror where to download MinGW:
Trolltech (Norway)
Download MinGW source code. (You don't need this to compile Qt)
Nullsoft Install System v2.18
< Back
Install
Cancel

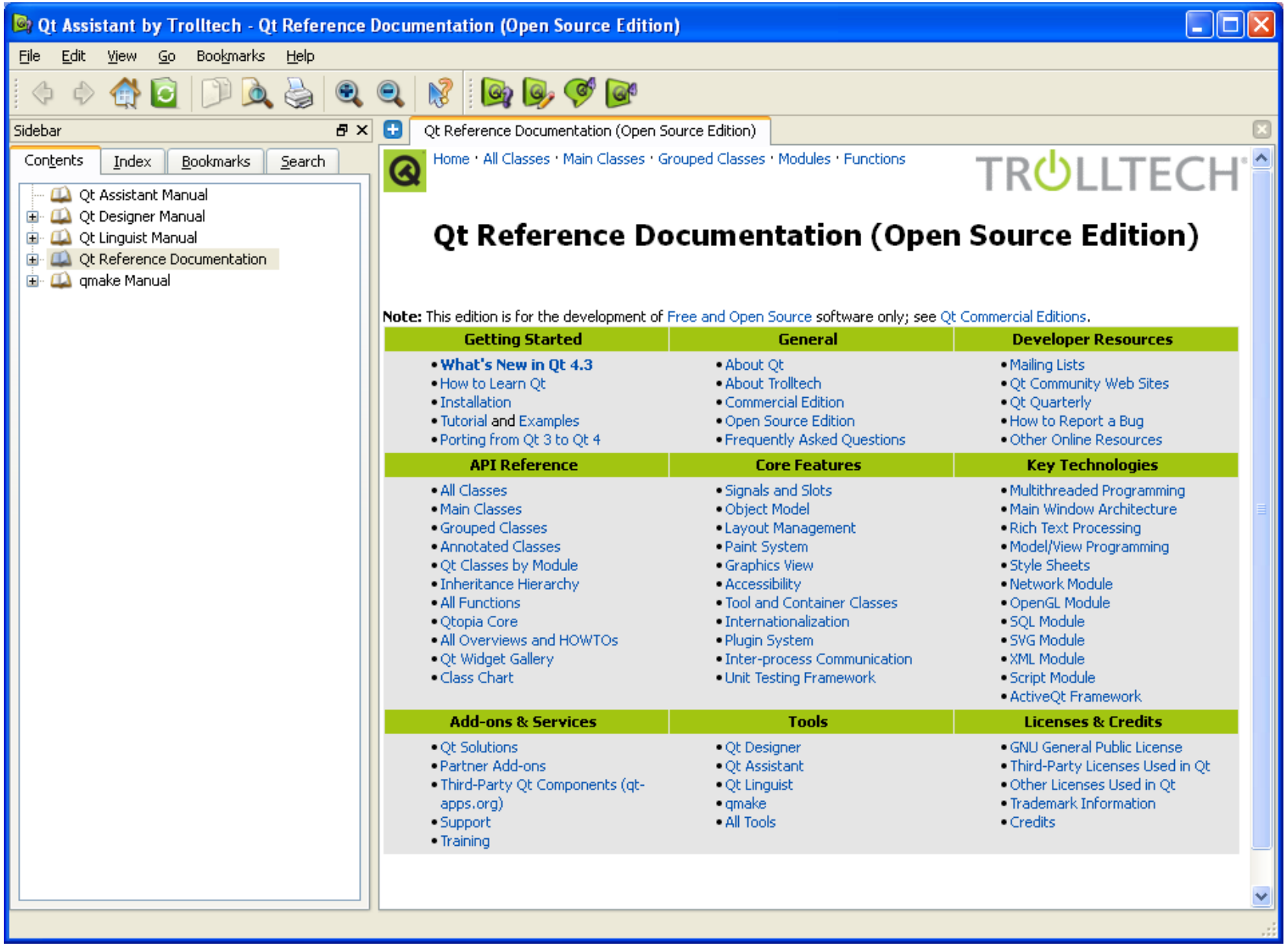

MinGW Download
Downloading MinGW from server...

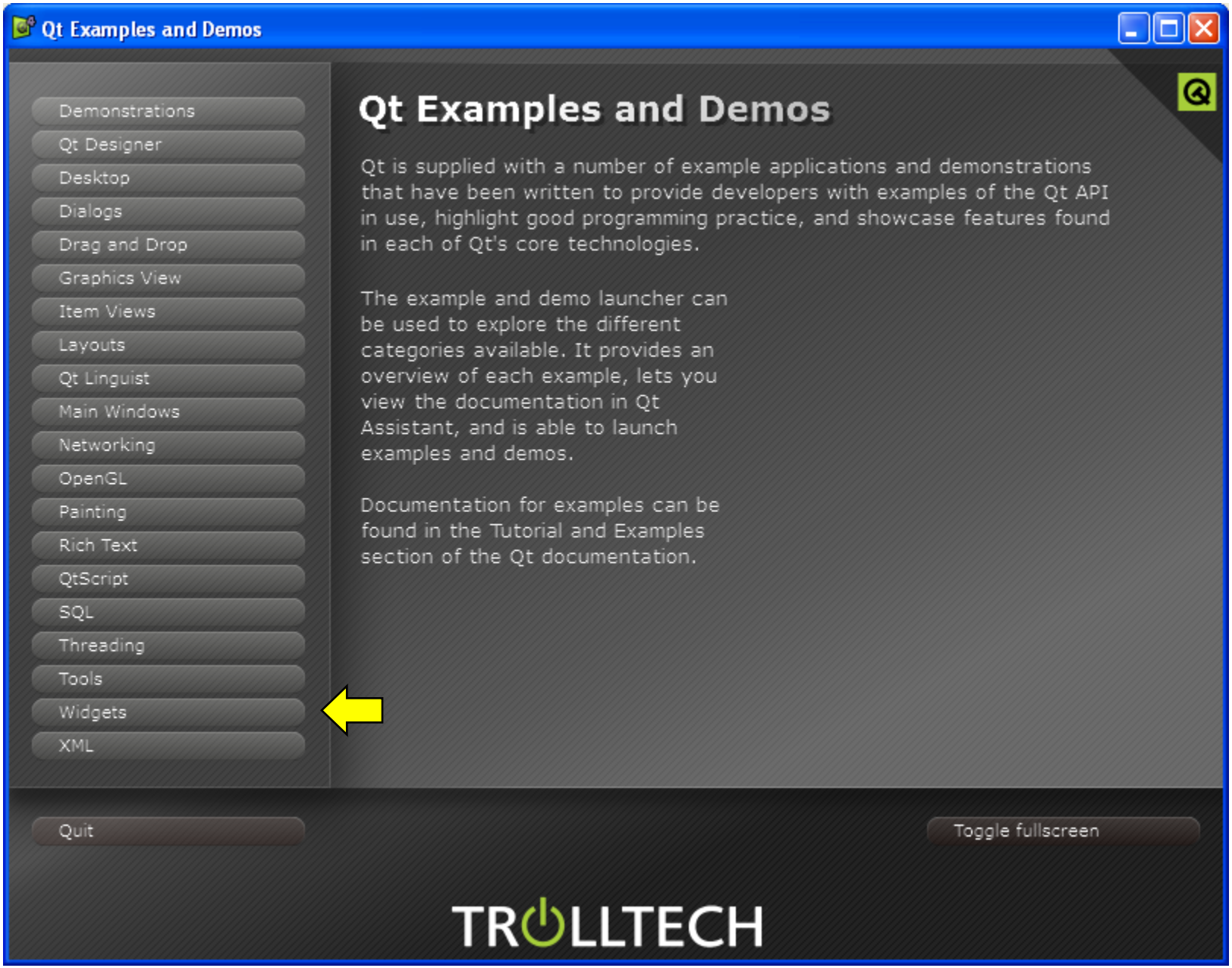

Qt OpenSource 4.3.3 Setup
Installing
Please wait while Qt OpenSource 4.3.3 is being installed.
Extract: doc\html\demos-textedit.html ... 100%
Show details
Nullsoft Install System v2.18
< Back
Next >
Cancel


Qt OpenSource 4.3.3 Setup
Installing
Please wait while Qt OpenSource 4.3.3 is being installed.
Extract: doc\html\images\qprogdlg-m.png
Extract: doc\html\images\qpen-dot.png
Extract: doc\html\images\qpen-flat.png
Extract: doc\html\images\qpen-miter.png
Extract: doc\html\images\qpen-miterlimit.png ... 100%
Extract: doc\html\images\qpen-roundcap.png
Extract: doc\html\images\qpen-roundjoin.png
Extract: doc\html\images\qpen-solid.png
Extract: doc\html\images\qpen-square.png
Extract: doc\html\images\qplastiquestyle.png ... 100%
Extract: doc\html\images\qprogbar-m.png
Extract: doc\html\images\qprogbar-w.png ... 100%
Extract: doc\html\images\qprogdlg-m.png
Nullsoft Install System v2.18
< Back
Next >
Cancel

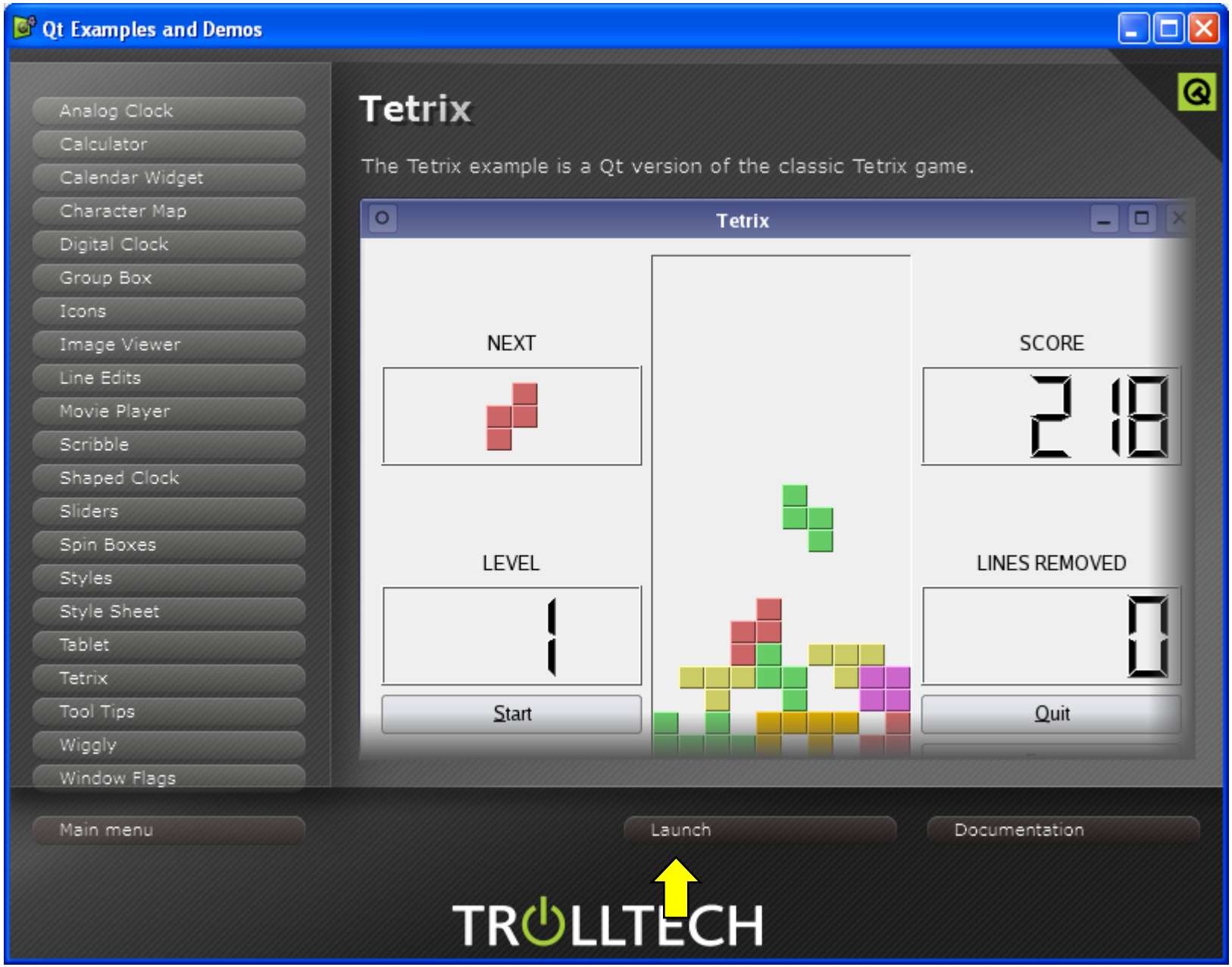

Qt OpenSource 4.3.3 Setup
Installation Complete
Setup was completed successfully.
Completed
Patching paths in qmake...
Patching paths in core...
Patching license information...
Create shortcut: C:\Documents and Settings\All Users\Start Menu\Programs\Qt by Tr...
Create shortcut: C:\Documents and Settings\All Users\Start Menu\Programs\Qt by Tr...
Installing MinGW into C:\yunglu\Qt\MinGW
Copying MinGW runtime...
Created uninstaller: C:\yunglu\Qt\uninst.exe
Create shortcut: C:\Documents and Settings\All Users\Start Menu\Programs\Qt by Tr...
Create shortcut: C:\Documents and Settings\All Users\Start Menu\Programs\Qt by Tr...
Output folder: C:\yunglu\Qt
Completed
Nullsoft Install System v2.18
< Back
Next >
Cancel

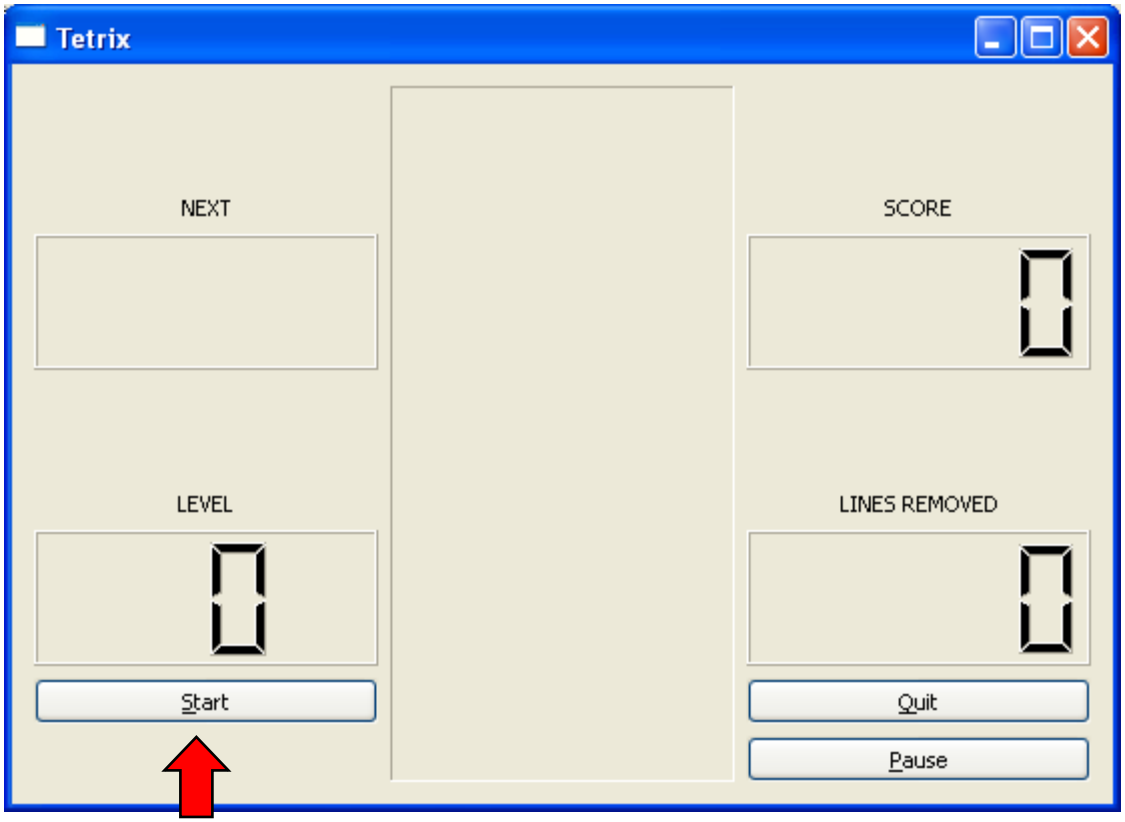

Qt OpenSource 4.3.3 Setup
TROLLTECH
Code Less. Create More.
Completing the Qt OpenSource 4.3.3 Setup Wizard
Qt OpenSource 4.3.3 has been installed on your computer.
Click Finish to close this wizard.
Run Examples and Demos
Show Documentation
< Back
Finish
Cancel

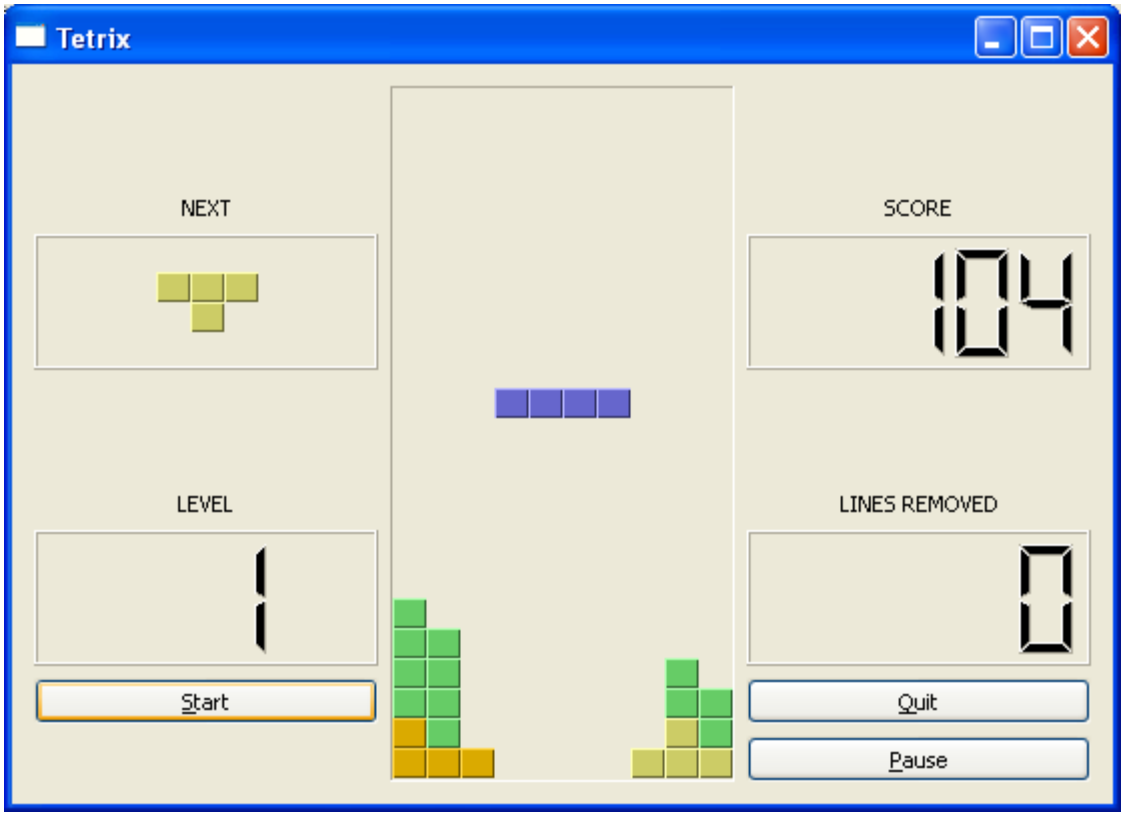

Qt Assistant by Trolltech - Qt Reference Documentation (Open Source Edition)
File Edit View Go Bookmarks Help
Sidebar
Contents Index Bookmarks Search
Qt Assistant Manual
Qt Designer Manual
Qt Linguist Manual
Qt Reference Documentation
qmake Manual
Qt Reference Documentation (Open Source Edition)
Home · All Classes · Main Classes · Grouped Classes · Modules · Functions
TROLLTECH
Qt Reference Documentation (Open Source Edition)
Note: This edition is for the development of Free and Open Source software only; see Qt Commercial Editions.
Getting Started
What's New in Qt 4.3
How to Learn Qt
Installation
Tutorial and Examples
Porting from Qt 3 to Qt 4
General
About Qt
About Trolltech
Commercial Edition
Open Source Edition
Frequently Asked Questions
Developer Resources
Mailing Lists
Qt Community Web Sites
Qt Quarterly
How to Report a Bug
Other Online Resources
API Reference
All Classes
Main Classes
Grouped Classes
Annotated Classes
Qt Classes by Module
Inheritance Hierarchy
All Functions
Qtopia Core
All Overviews and HOWTOs
Qt Widget Gallery
Class Chart
Core Features
Signals and Slots
Object Model
Layout Management
Paint System
Graphics View
Accessibility
Tool and Container Classes
Internationalization
Plugin System
Inter-process Communication
Unit Testing Framework
Key Technologies
Multithreaded Programming
Main Window Architecture
Rich Text Processing
Model/View Programming
Style Sheets
Network Module
OpenGL Module
SQL Module
SVG Module
XML Module
Script Module
ActiveQt Framework
Add-ons & Services
Qt Solutions
Partner Add-ons
Third-Party Qt Components (qt-apps.org)
Support
Training
Tools
Qt Designer
Qt Assistant
Qt Linguist
qmake
All Tools
Licenses & Credits
GNU General Public License
Third-Party Licenses Used in Qt
Other Licenses Used in Qt
Trademark Information
Credits


Qt Examples and Demos
Demonstrations
Qt Designer
Desktop
Dialogs
Drag and Drop
Graphics View
Item Views
Layouts
Qt Linguist
Main Windows
Networking
OpenGL
Painting
Rich Text
QtScript
SQL
Threading
Tools
Widgets
XML
Qt Examples and Demos
Qt is supplied with a number of example applications and demonstrations that have been written to provide developers with examples of the Qt API in use, highlight good programming practice, and showcase features found in each of Qt's core technologies.
The example and demo launcher can be used to explore the different categories available. It provides an overview of each example, lets you view the documentation in Qt Assistant, and is able to launch examples and demos.
Documentation for examples can be found in the Tutorial and Examples section of the Qt documentation.
Quit
Toggle fullscreen
TROLLTECH

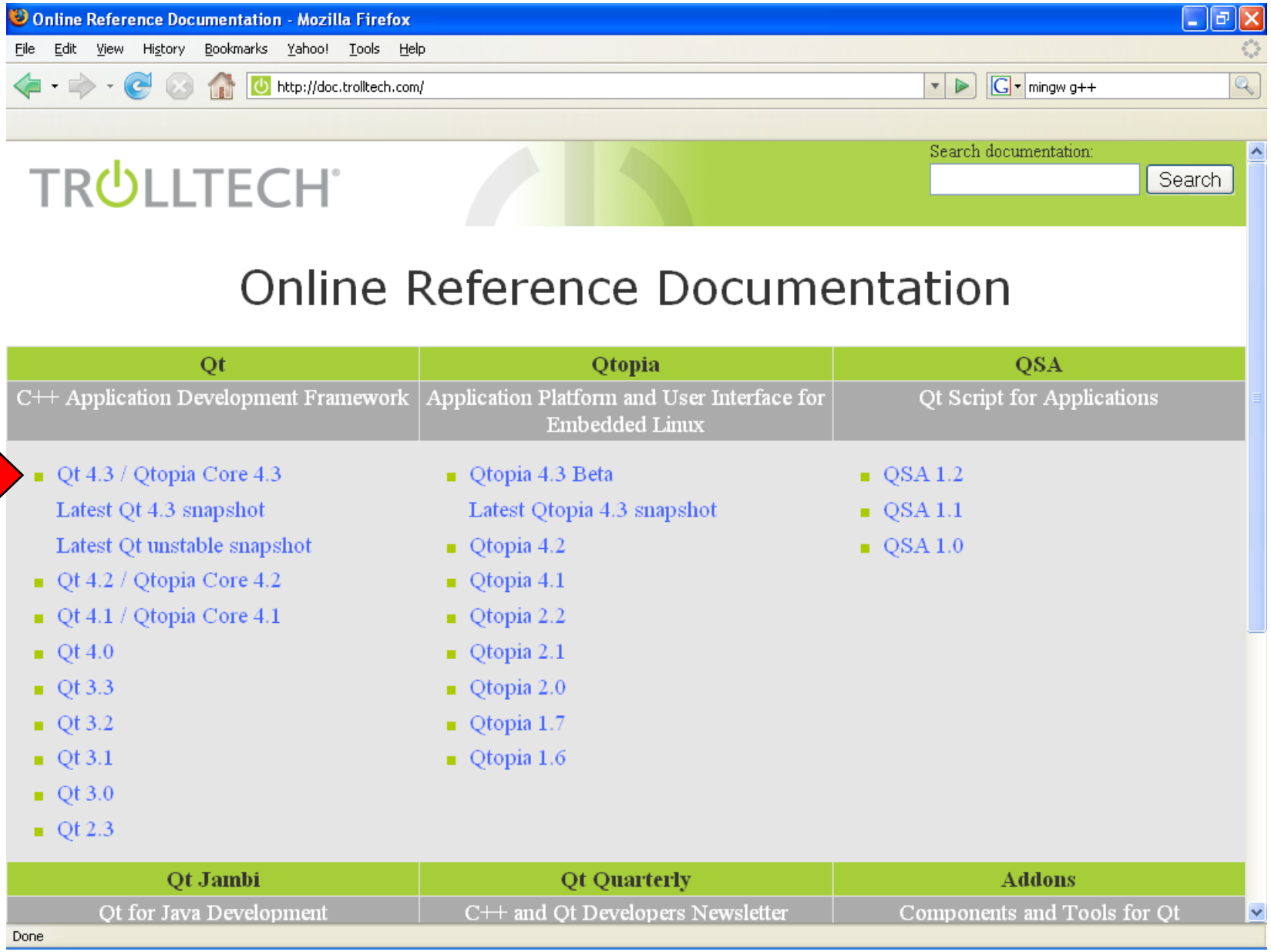

Qt Examples and Demos
Analog Clock
Calculator
Calendar Widget
Character Map
Digital Clock
Group Box
Icons
Image Viewer
Line Edits
Movie Player
Scribble
Shaped Clock
Sliders
Spin Boxes
Styles
Style Sheet
Tablet
Tetrix
Tool Tips
Wiggly
Window Flags
Main menu
Widgets
Qt comes with a large range of standard widgets that users of modern applications have come to expect.
You can also develop your own custom widgets and controls, and use them alongside standard widgets.
It is even possible to provide custom styles and themes for widgets that can be used to change the appearance of standard widgets and appropriately written custom widgets.
TROLLTECH

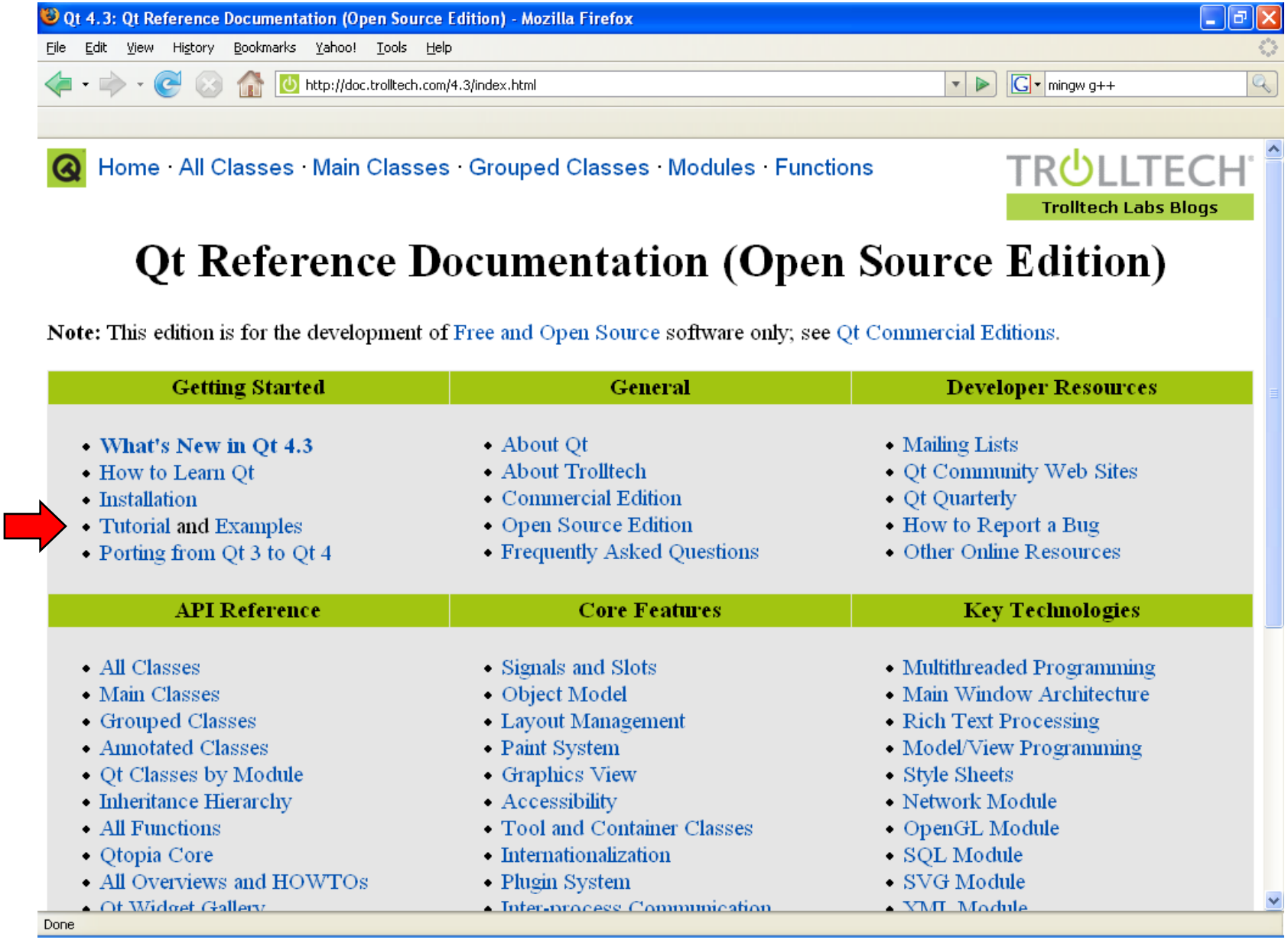

Qt Examples and Demos
Analog Clock
Calculator
Calendar Widget
Character Map
Digital Clock
Group Box
Icons
Image Viewer
Line Edits
Movie Player
Scribble
Shaped Clock
Sliders
Spin Boxes
Styles
Style Sheet
Tablet
Tetrix
Tool Tips
Wiggly
Window Flags
Tetrix
The Tetrix example is a Qt version of the classic Tetrix game.
Tetrix
NEXT
SCORE
218
LEVEL
1
LINES REMOVED
0
Start
Quit
Main menu
Launch
Documentation
TROLLTECH

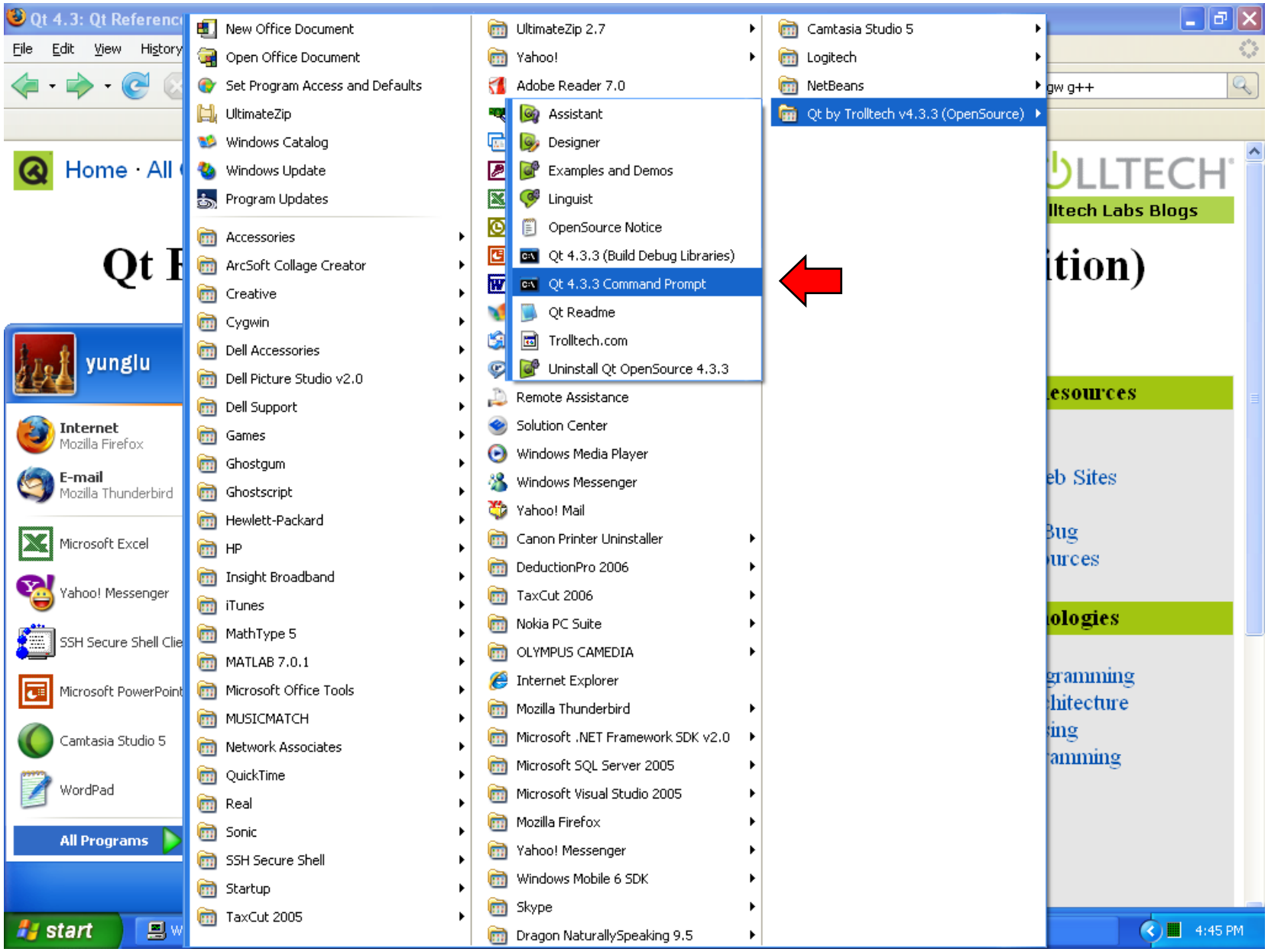

Tetrix
NEXT
SCORE
0
LEVEL
0
LINES REMOVED
0
Start
Quit
Pause

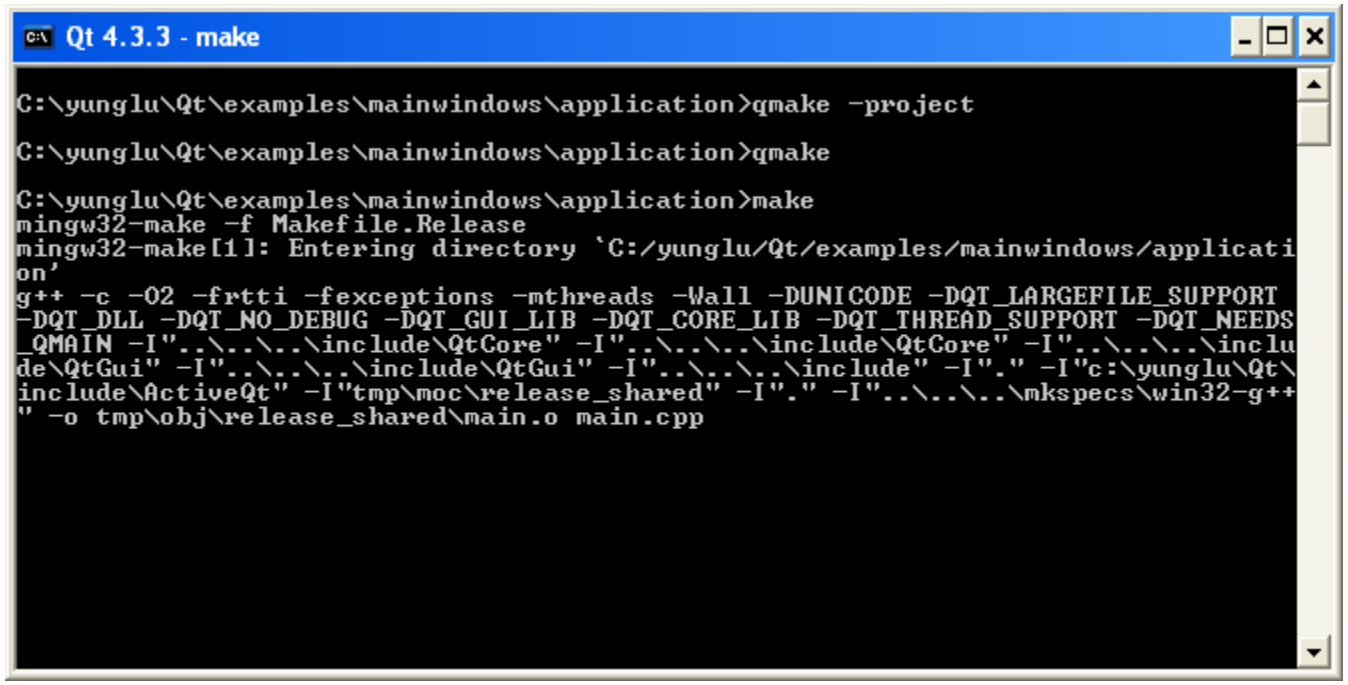

Tetrix
NEXT
SCORE
104
LEVEL
1
LINES REMOVED
0
Start
Quit
Pause

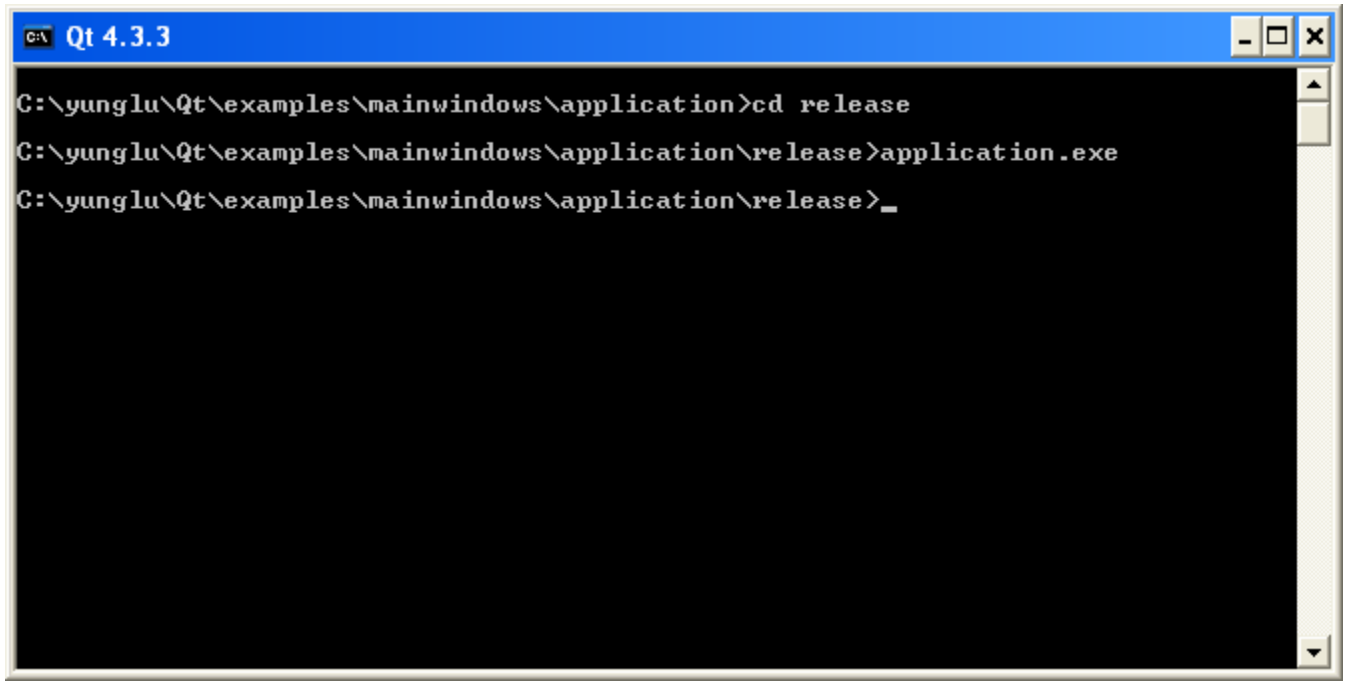
Qt Documentation

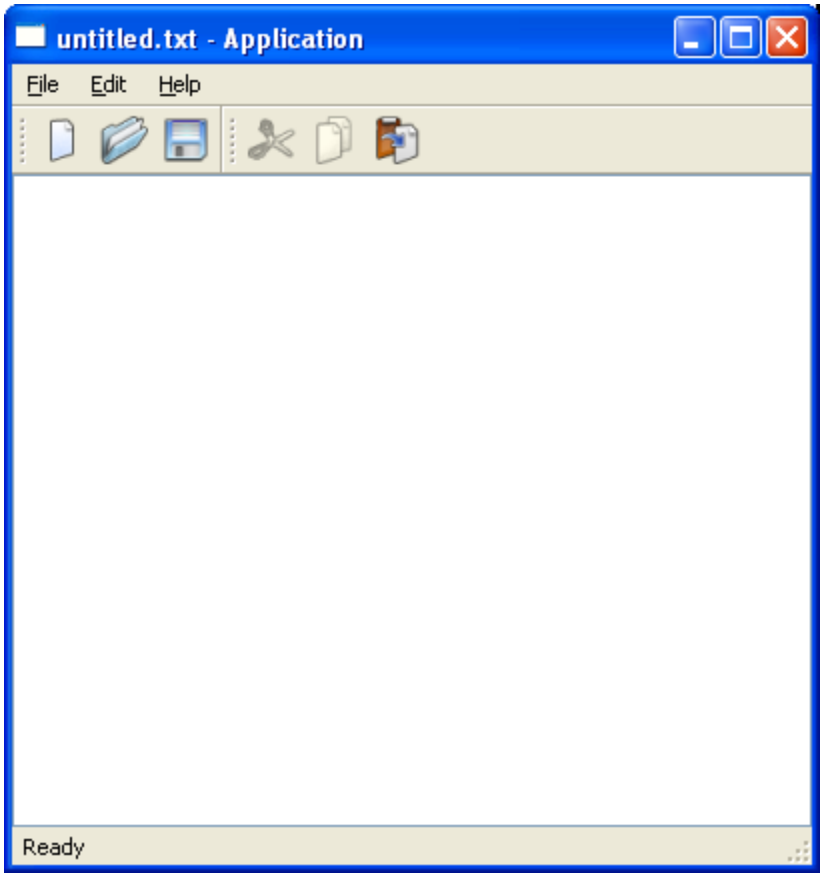

Online Reference Documentation - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://doc.trolltech.com/
mingw g++
TROLLTECH
Search documentation:
Search
Online Reference Documentation
Qt
C++ Application Development Framework
Qt 4.3 / Qtopia Core 4.3
Latest Qt 4.3 snapshot
Latest Qt unstable snapshot
Qt 4.2 / Qtopia Core 4.2
Qt 4.1 / Qtopia Core 4.1
Qt 4.0
Qt 3.3
Qt 3.2
Qt 3.1
Qt 3.0
Qt 2.3
Qtopia
Application Platform and User Interface for Embedded Linux
Qtopia 4.3 Beta
Latest Qtopia 4.3 snapshot
Qtopia 4.2
Qtopia 4.1
Qtopia 2.2
Qtopia 2.1
Qtopia 2.0
Qtopia 1.7
Qtopia 1.6
QSA
Qt Script for Applications
QSA 1.2
QSA 1.1
QSA 1.0
Qt Jambi
Qt for Java Development
Qt Quarterly
C++ and Qt Developers Newsletter
Addons
Components and Tools for Qt
Done


Qt 4.3: Qt Reference Documentation (Open Source Edition) - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://doc.trolltech.com/4.3/index.html
mingw g++
Home · All Classes · Main Classes · Grouped Classes · Modules · Functions
TROLLTECH
Trolltech Labs Blogs
Qt Reference Documentation (Open Source Edition)
Note: This edition is for the development of Free and Open Source software only; see Qt Commercial Editions.
Getting Started
What's New in Qt 4.3
How to Learn Qt
Installation
Tutorial and Examples
Porting from Qt 3 to Qt 4
General
About Qt
About Trolltech
Commercial Edition
Open Source Edition
Frequently Asked Questions
Developer Resources
Mailing Lists
Qt Community Web Sites
Qt Quarterly
How to Report a Bug
Other Online Resources
API Reference
All Classes
Main Classes
Grouped Classes
Annotated Classes
Qt Classes by Module
Inheritance Hierarchy
All Functions
Qtopia Core
All Overviews and HOWTOs
Qt Widget Gallery
Core Features
Signals and Slots
Object Model
Layout Management
Paint System
Graphics View
Accessibility
Tool and Container Classes
Internationalization
Plugin System
Inter-process Communication
Key Technologies
Multithreaded Programming
Main Window Architecture
Rich Text Processing
Model/View Programming
Style Sheets
Network Module
OpenGL Module
SQL Module
SVG Module
XML Module
Done

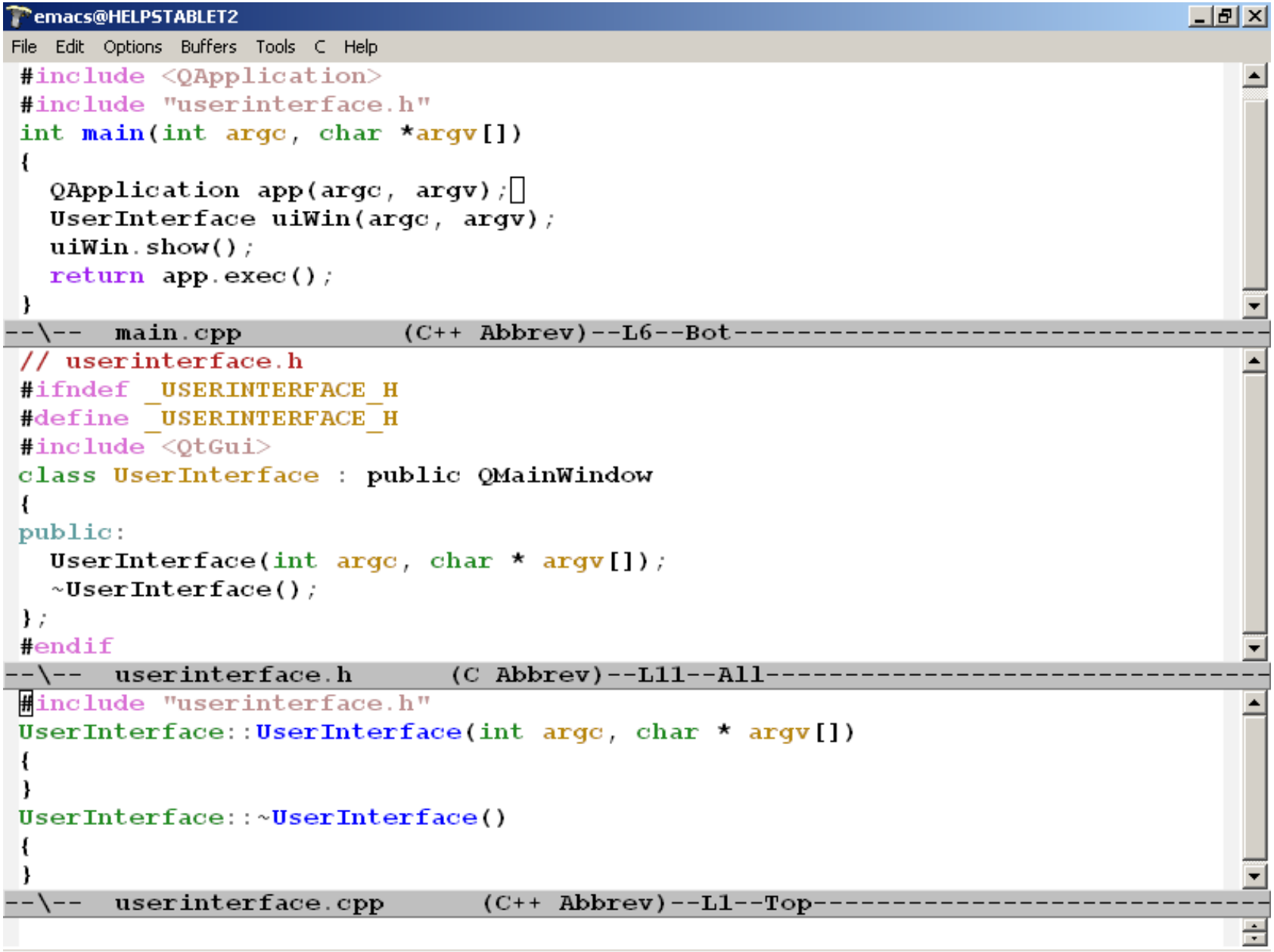

Qt 4.3: Qt Reference
File Edit View History
Home · All
Qt H
yunglu
Internet
Mozilla Firefox
E-mail
Mozilla Thunderbird
Microsoft Excel
Yahoo! Messenger
SSH Secure Shell Clie
Microsoft PowerPoint
Camtasia Studio 5
WordPad
All Programs
start
New Office Document
Open Office Document
Set Program Access and Defaults
UltimateZip
Windows Catalog
Windows Update
Program Updates
Accessories
ArcSoft Collage Creator
Creative
Cygwin
Dell Accessories
Dell Picture Studio v2.0
Dell Support
Games
Ghostgum
Ghostscript
Hewlett-Packard
HP
Insight Broadband
iTunes
MathType 5
MATLAB 7.0.1
Microsoft Office Tools
MUSICMATCH
Network Associates
QuickTime
Real
Sonic
SSH Secure Shell
Startup
TaxCut 2005
UltimateZip 2.7
Yahoo!
Adobe Reader 7.0
Assistant
Designer
Examples and Demos
Linguist
OpenSource Notice
Qt 4.3.3 (Build Debug Libraries)
Qt 4.3.3 Command Prompt
Qt Readme
Trolltech.com
Uninstall Qt OpenSource 4.3.3
Remote Assistance
Solution Center
Windows Media Player
Windows Messenger
Yahoo! Mail
Canon Printer Uninstaller
DeductionPro 2006
TaxCut 2006
Nokia PC Suite
OLYMPUS CAMEDIA
Internet Explorer
Mozilla Thunderbird
Microsoft .NET Framework SDK v2.0
Microsoft SQL Server 2005
Microsoft Visual Studio 2005
Mozilla Firefox
Yahoo! Messenger
Windows Mobile 6 SDK
Skype
Dragon NaturallySpeaking 9.5
Camtasia Studio 5
Logitech
NetBeans
Qt by Trolltech v4.3.3 (OpenSource)
gw g++
LLTECH
lltech Labs Blogs
ition)
esources
eb Sites
Bug
urces
ologies
gramming
hitecture
ing
amming
4:45 PM

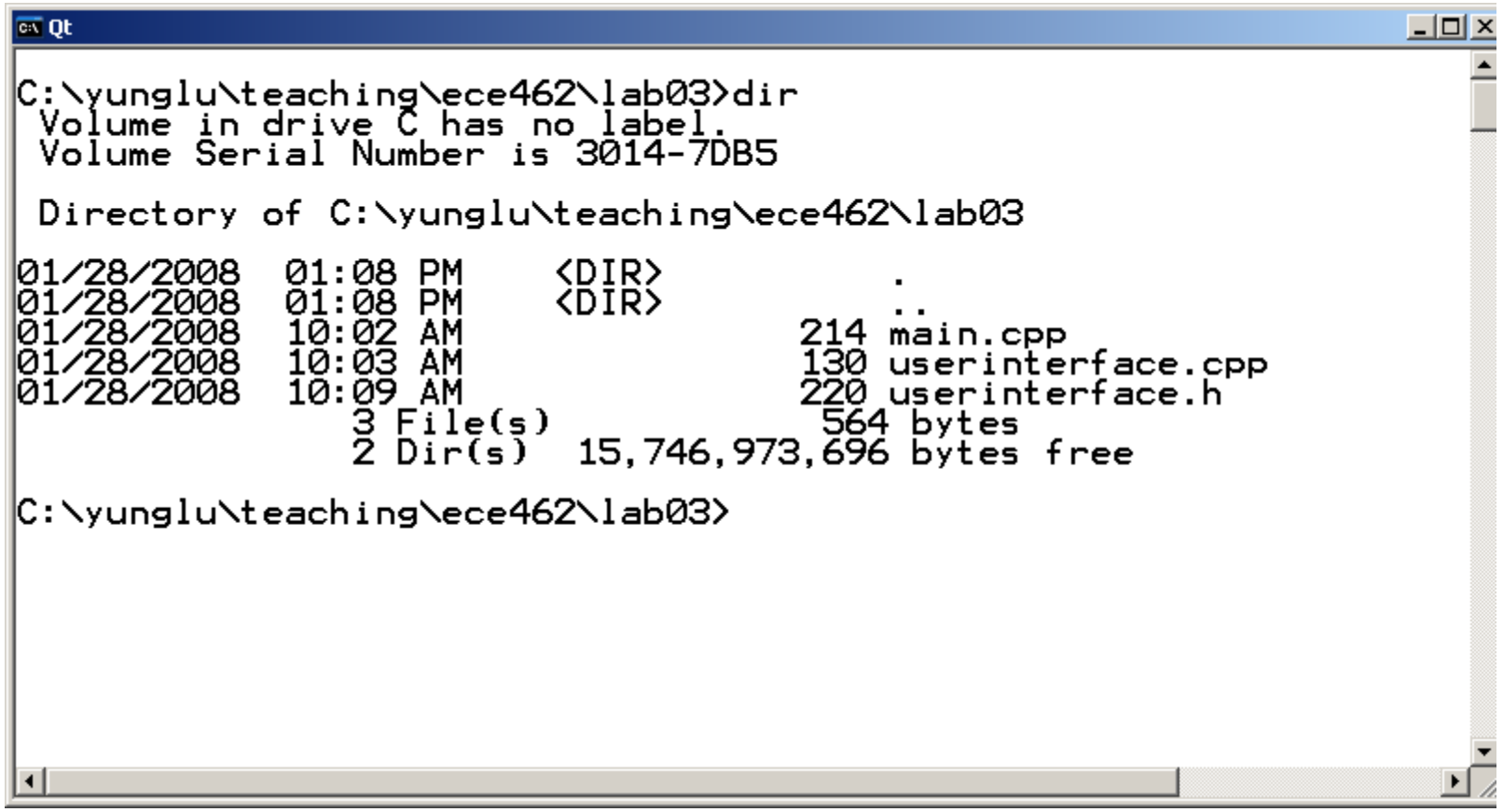

Qt 4.3.3 - make
C:\yunglu\Qt\examples\mainwindows\application>qmake -project
C:\yunglu\Qt\examples\mainwindows\application>qmake
C:\yunglu\Qt\examples\mainwindows\application>make
mingw32-make -f Makefile.Release
mingw32-make[1]: Entering directory `C:/yunglu/Qt/examples/mainwindows/applicati
on'
g++ -c -O2 -frtti -fexceptions -mthreads -Wall -DUNICODE -DQT_LARGEFILE_SUPPORT
-DQT_DLL -DQT_NO_DEBUG -DQT_GUI_LIB -DQT_CORE_LIB -DQT_THREAD_SUPPORT -DQT_NEEDS
_QMAIN -I"..\..\..\include\QtCore" -I"..\..\..\include\QtCore" -I"..\..\..\inclu
de\QtGui" -I"..\..\..\include\QtGui" -I"..\..\..\include" -I"." -I"c:\yunglu\Qt\
include\ActiveQt" -I"tmp\moc\release_shared" -I"." -I"..\..\..\mkspecs\win32-g++
" -o tmp\obj\release_shared\main.o main.cpp

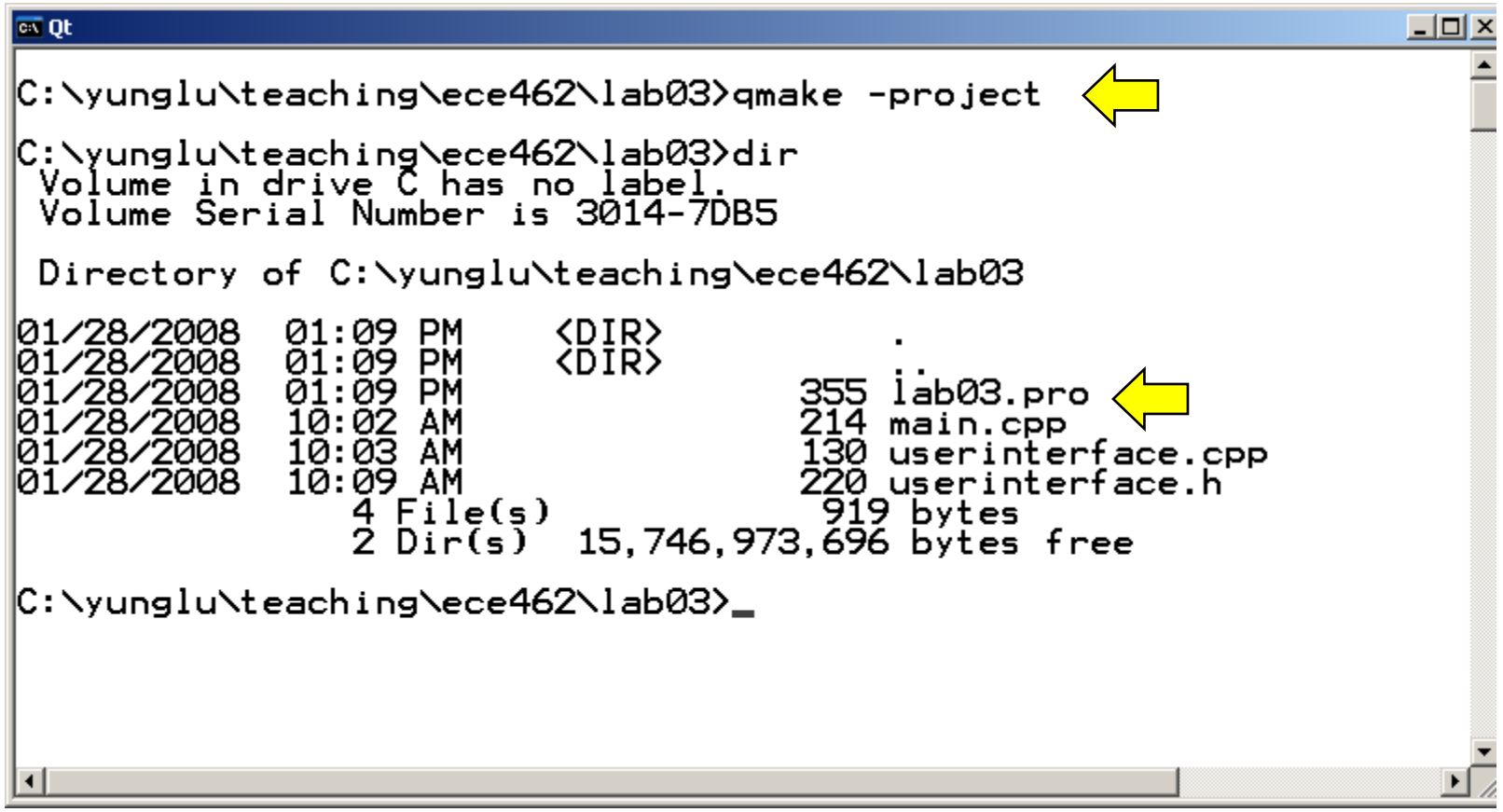

Qt 4.3.3
C:\yunglu\Qt\examples\mainwindows\application>cd release
C:\yunglu\Qt\examples\mainwindows\application\release>application.exe
C:\yunglu\Qt\examples\mainwindows\application\release>

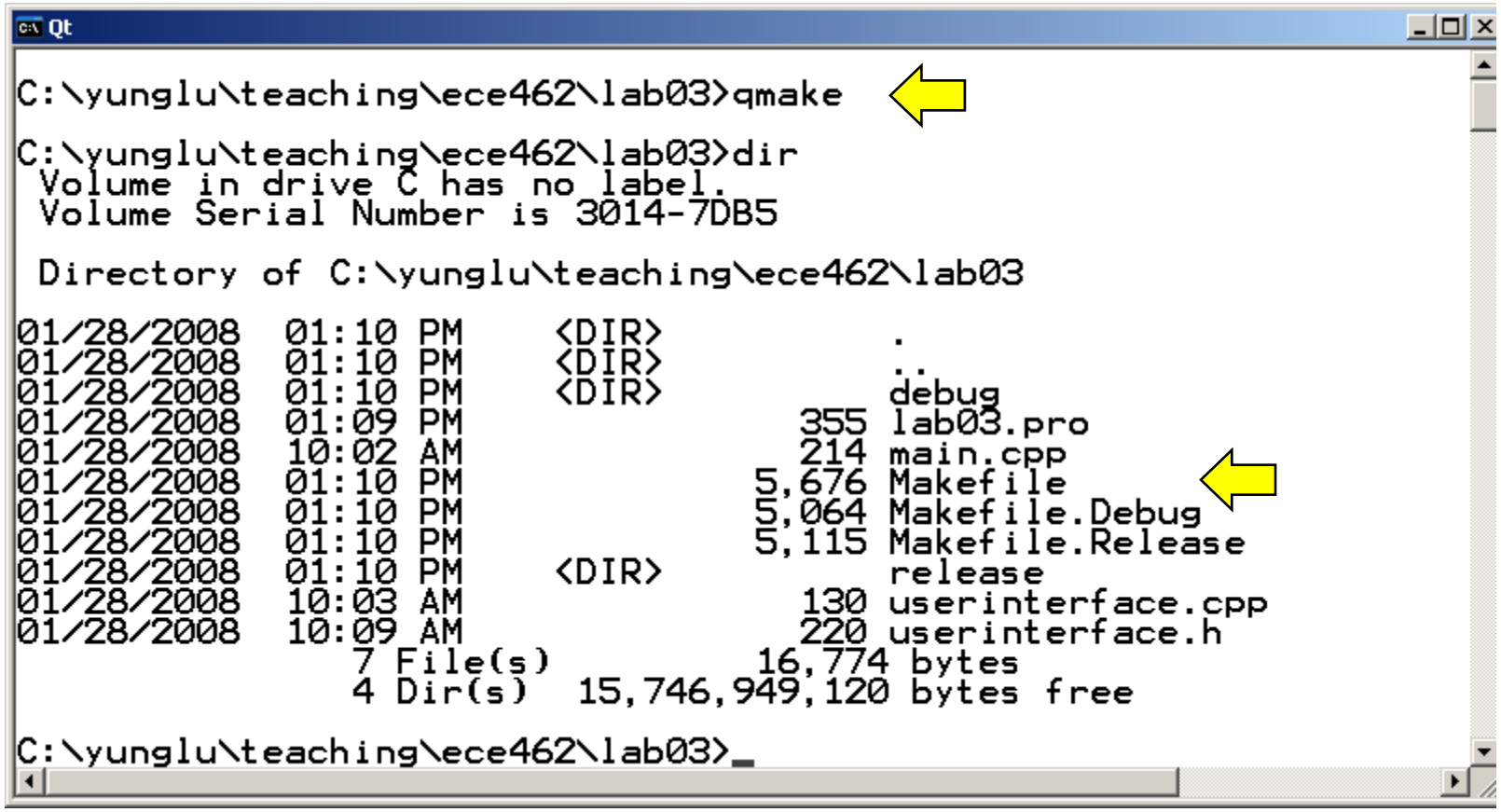

untitled.txt - Application
File
Edit
Help
Ready

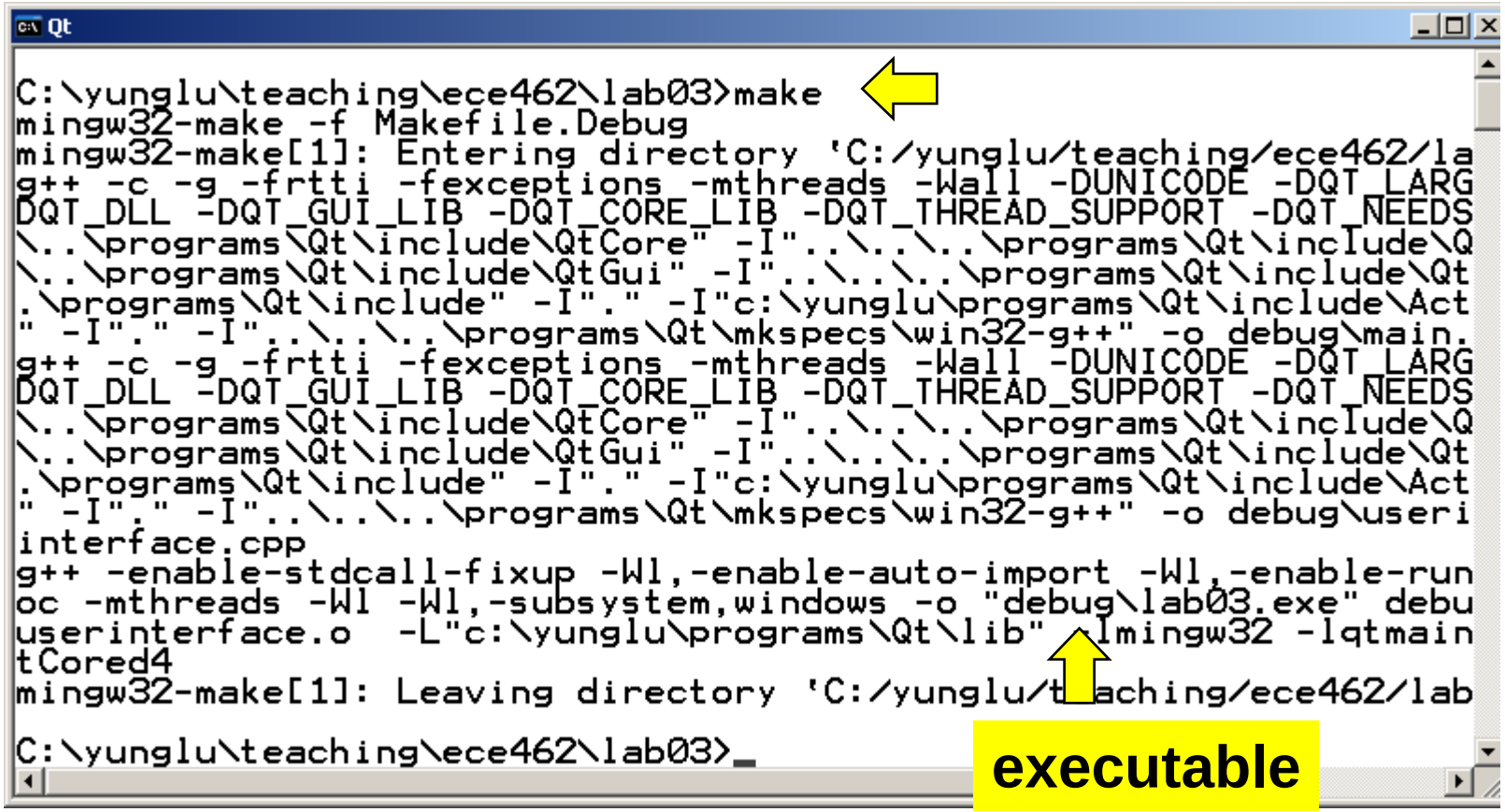
Sample Qt Program


emacs@HELPSTABLET2
File Edit Options Buffers Tools C Help
#include <QApplication>
#include "userinterface.h"
int main(int argc, char *argv[])
{
QApplication app(argc, argv);
UserInterface uiWin(argc, argv);
uiWin.show();
return app.exec();
}
--\-- main.cpp (C++ Abbrev)--L6--Bot
// userinterface.h
#ifndef _USERINTERFACE_H
#define _USERINTERFACE_H
#include <QtGui>
class UserInterface : public QMainWindow
{
public:
UserInterface(int argc, char * argv[]);
~UserInterface();
};
#endif
--\-- userinterface.h (C Abbrev)--L11--All
#include "userinterface.h"
UserInterface::UserInterface(int argc, char * argv[])
{
}
UserInterface::~UserInterface()
{
}
--\-- userinterface.cpp (C++ Abbrev)--L1--Top

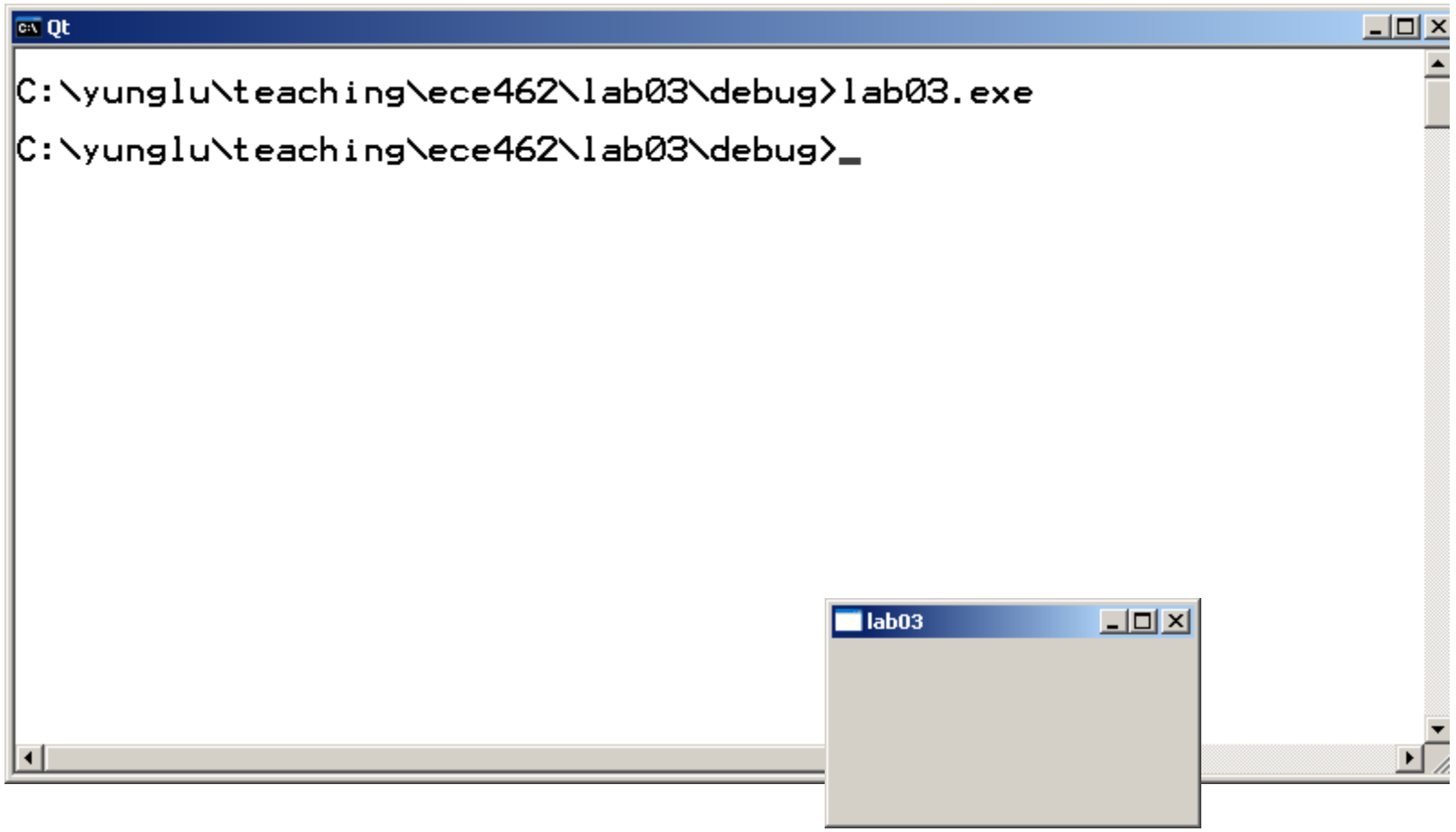

Qt
C:\yunglu\teaching\ece462\lab03>dir
Volume in drive C has no label.
Volume Serial Number is 3014-7DB5
Directory of C:\yunglu\teaching\ece462\lab03
01/28/2008 01:08 PM <DIR> .
01/28/2008 01:08 PM <DIR> ..
01/28/2008 10:02 AM 214 main.cpp
01/28/2008 10:03 AM 130 userinterface.cpp
01/28/2008 10:09 AM 220 userinterface.h
3 File(s) 564 bytes
2 Dir(s) 15,746,973,696 bytes free
C:\yunglu\teaching\ece462\lab03>


Qt
C:\yunglu\teaching\ece462\lab03>qmake -project
C:\yunglu\teaching\ece462\lab03>dir
Volume in drive C has no label.
Volume Serial Number is 3014-7DB5
Directory of C:\yunglu\teaching\ece462\lab03
01/28/2008 01:09 PM <DIR> .
01/28/2008 01:09 PM <DIR> ..
01/28/2008 01:09 PM 355 lab03.pro
01/28/2008 10:02 AM 214 main.cpp
01/28/2008 10:03 AM 130 userinterface.cpp
01/28/2008 10:09 AM 220 userinterface.h
4 File(s) 919 bytes
2 Dir(s) 15,746,973,696 bytes free
C:\yunglu\teaching\ece462\lab03>

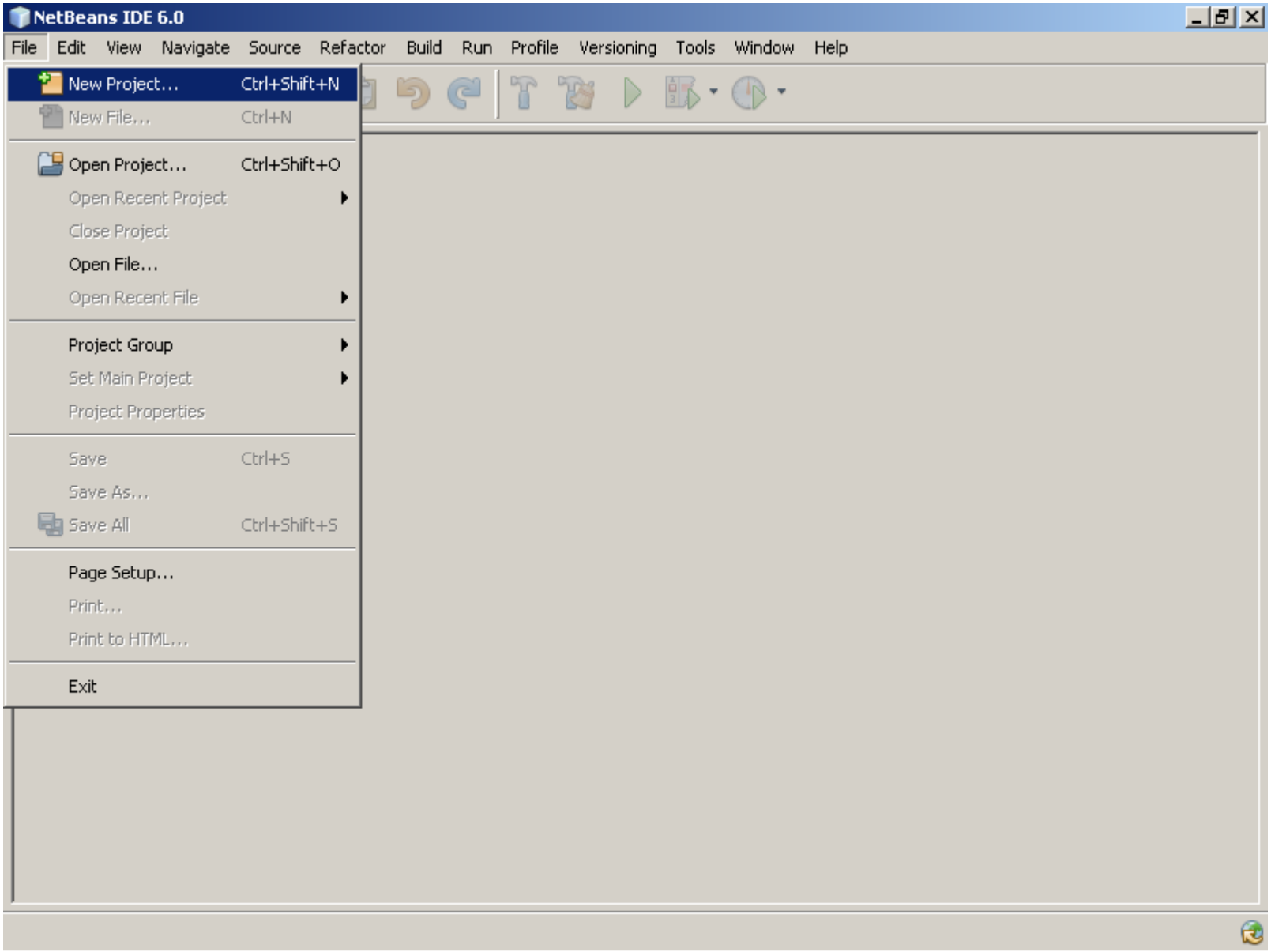

Qt
C:\yunglu\teaching\ece462\lab03>qmake
C:\yunglu\teaching\ece462\lab03>dir
Volume in drive C has no label.
Volume Serial Number is 3014-7DB5
Directory of C:\yunglu\teaching\ece462\lab03
01/28/2008 01:10 PM <DIR> .
01/28/2008 01:10 PM <DIR> ..
01/28/2008 01:10 PM <DIR> debug
01/28/2008 01:09 PM 355 lab03.pro
01/28/2008 10:02 AM 214 main.cpp
01/28/2008 01:10 PM 5,676 Makefile
01/28/2008 01:10 PM 5,064 Makefile.Debug
01/28/2008 01:10 PM 5,115 Makefile.Release
01/28/2008 01:10 PM <DIR> release
01/28/2008 10:03 AM 130 userinterface.cpp
01/28/2008 10:09 AM 220 userinterface.h
7 File(s) 16,774 bytes
4 Dir(s) 15,746,949,120 bytes free
C:\yunglu\teaching\ece462\lab03>

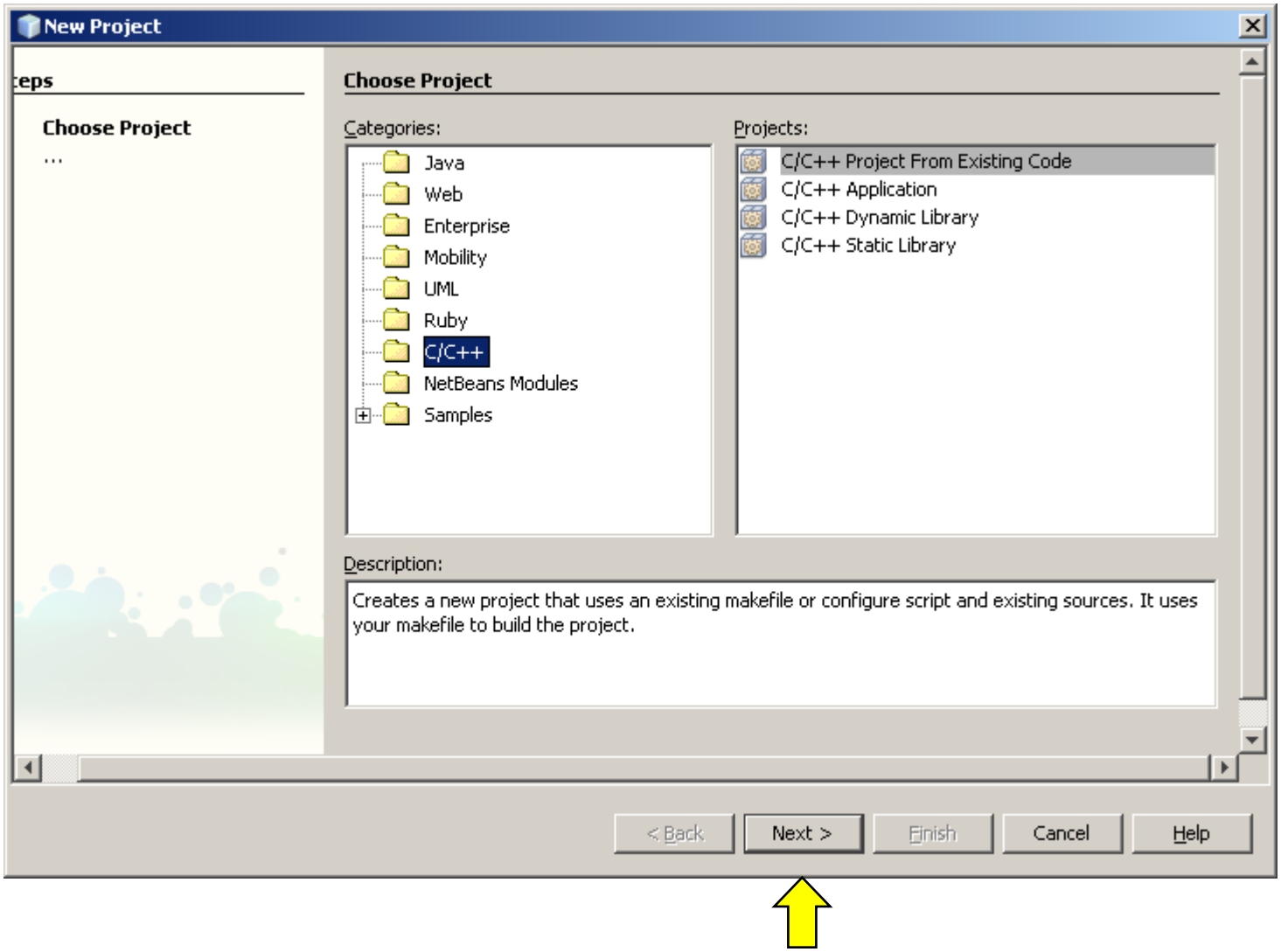


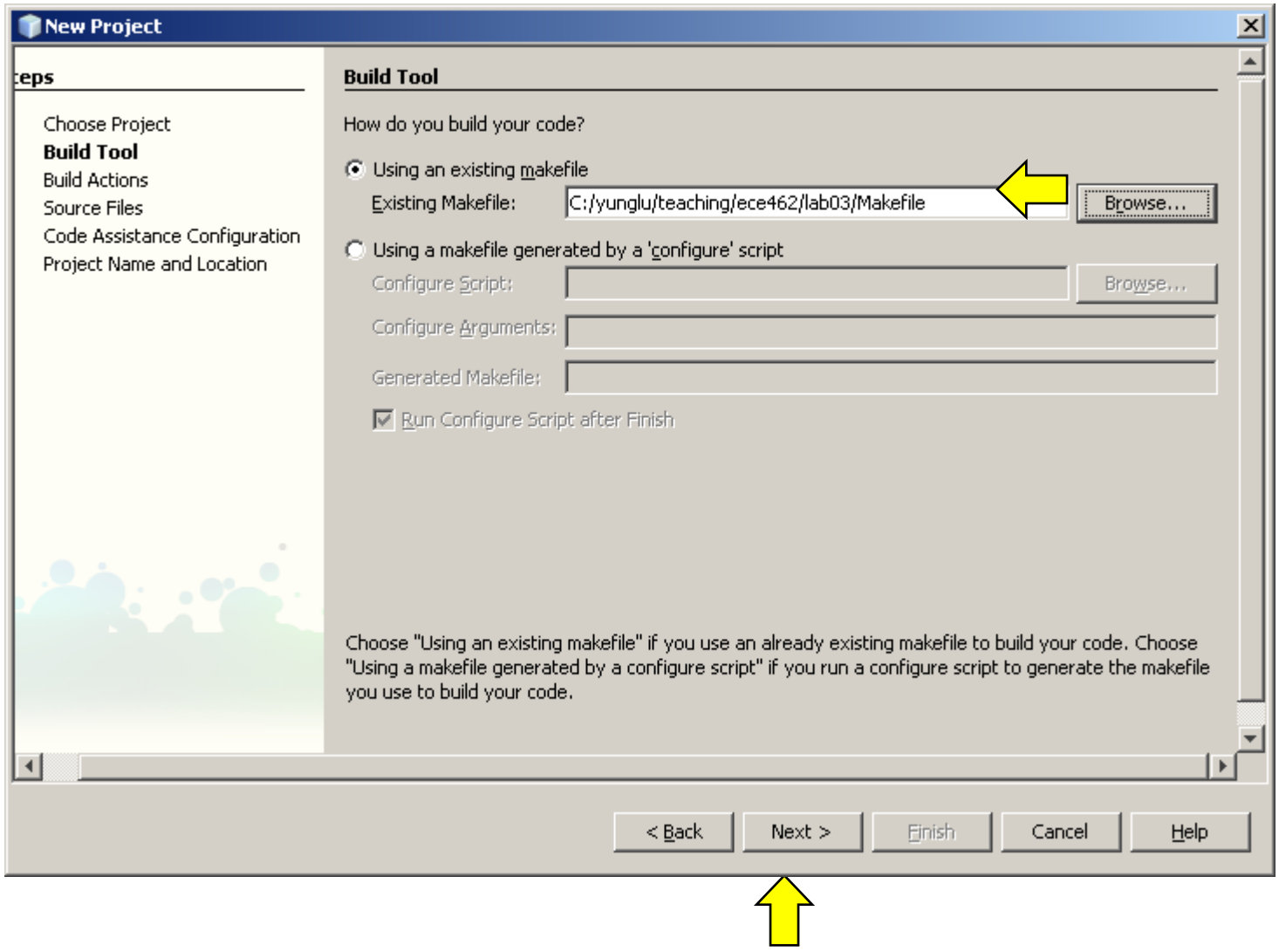
1. **qmake –project**
2. **qmake**
3. **make**

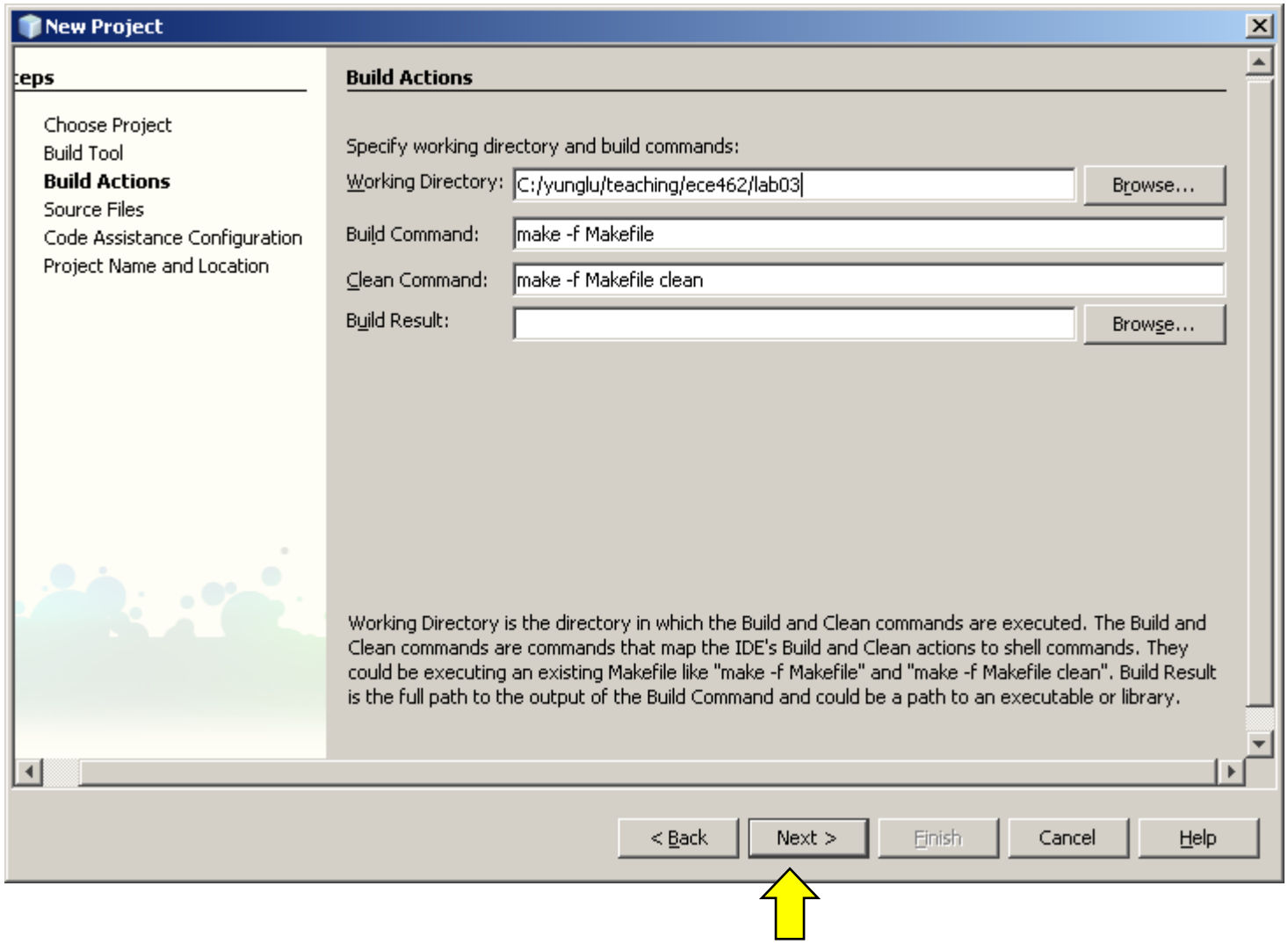

Qt
C:\yunglu\teaching\ece462\lab03\debug>lab03.exe
C:\yunglu\teaching\ece462\lab03\debug>_
lab03

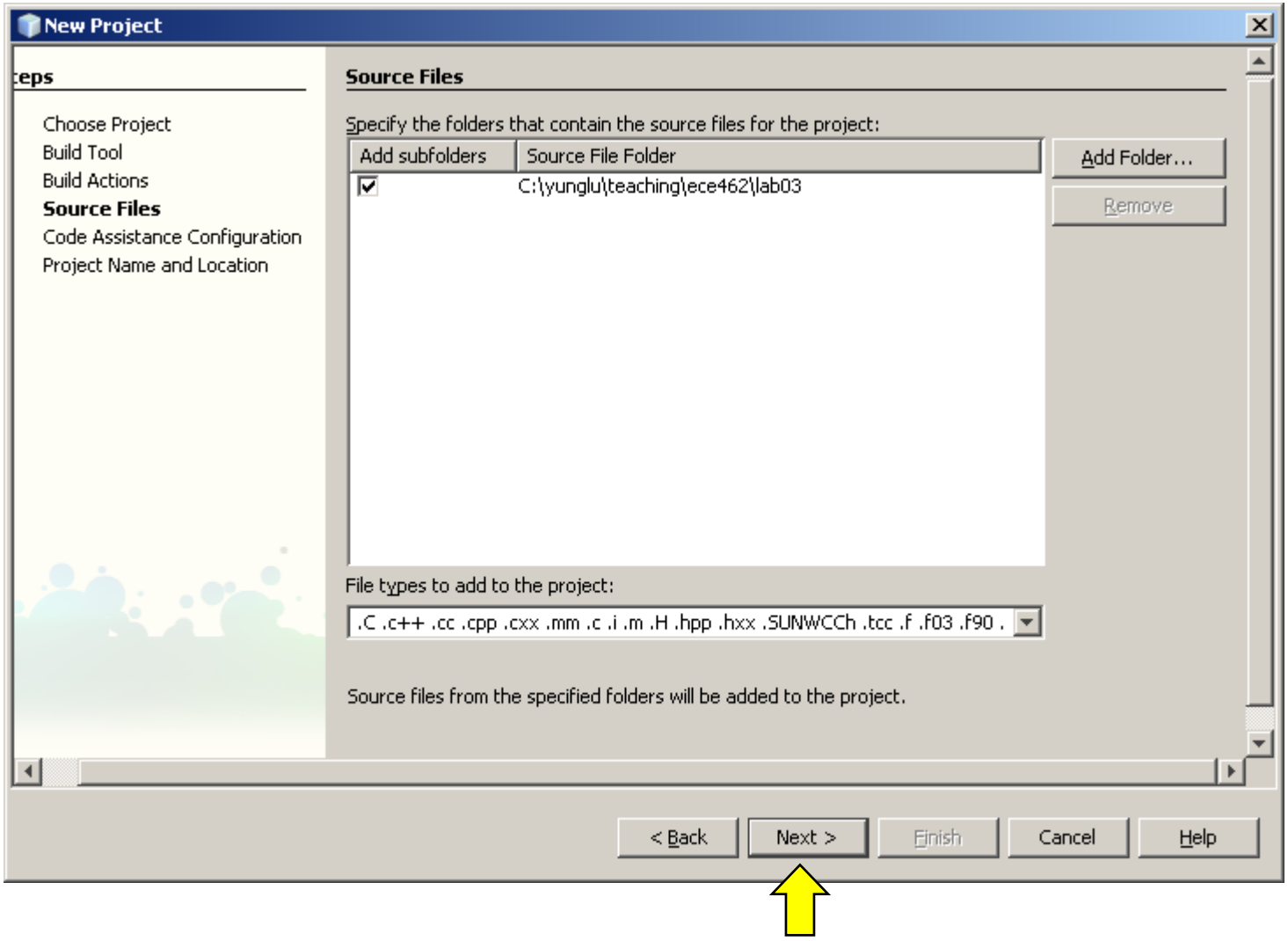
Develop Qt Project in Netbeans

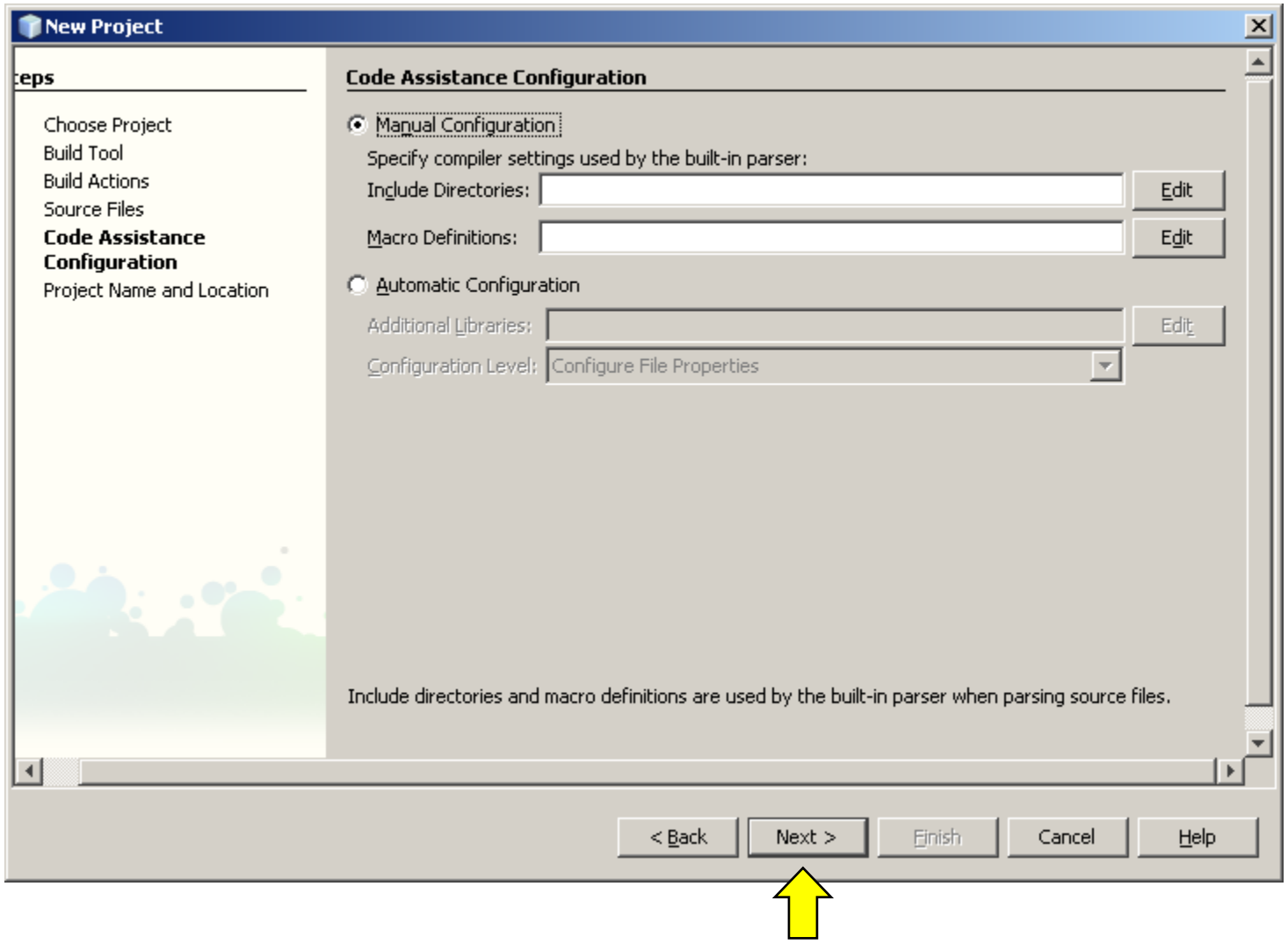

NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
New Project... Ctrl+Shift+N
New File... Ctrl+N
Open Project... Ctrl+Shift+O
Open Recent Project
Close Project
Open File...
Open Recent File
Project Group
Set Main Project
Project Properties
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Page Setup...
Print...
Print to HTML...
Exit

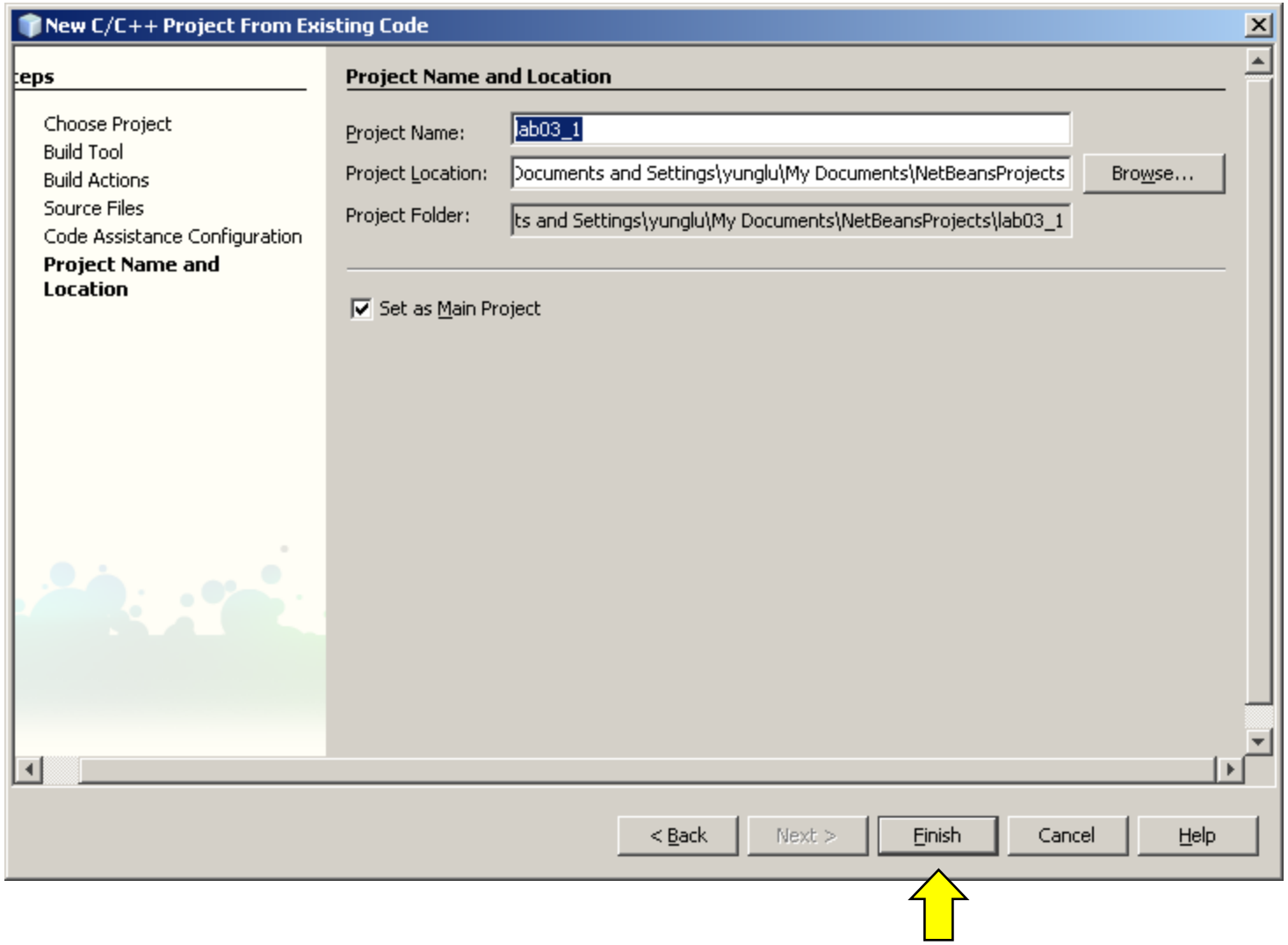

New Project
eps
Choose Project
...
Choose Project
Categories:
Java
Web
Enterprise
Mobility
UML
Ruby
C/C++
NetBeans Modules
Samples
Projects:
C/C++ Project From Existing Code
C/C++ Application
C/C++ Dynamic Library
C/C++ Static Library
Description:
Creates a new project that uses an existing makefile or configure script and existing sources. It uses your makefile to build the project.
< Back
Next >
Finish
Cancel
Help

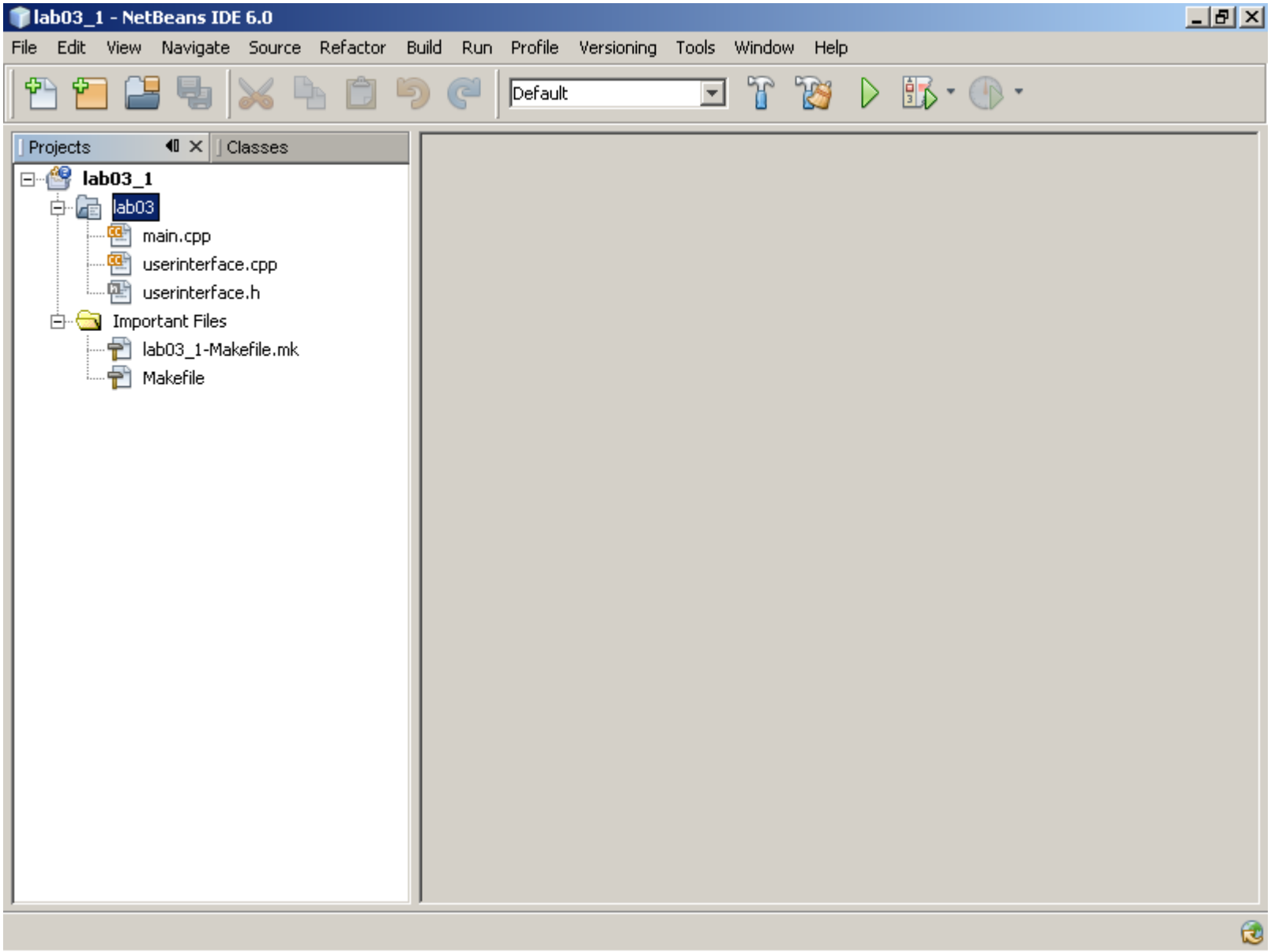

New Project
eps
Choose Project
Build Tool
Build Actions
Source Files
Code Assistance Configuration
Project Name and Location
Build Tool
How do you build your code?
Using an existing makefile
Existing Makefile:
C:/yunglu/teaching/ece462/lab03/Makefile
Browse...
Using a makefile generated by a 'configure' script
Configure Script:
Browse...
Configure Arguments:
Generated Makefile:
Run Configure Script after Finish
Choose "Using an existing makefile" if you use an already existing makefile to build your code. Choose "Using a makefile generated by a configure script" if you run a configure script to generate the makefile you use to build your code.
< Back
Next >
Finish
Cancel
Help

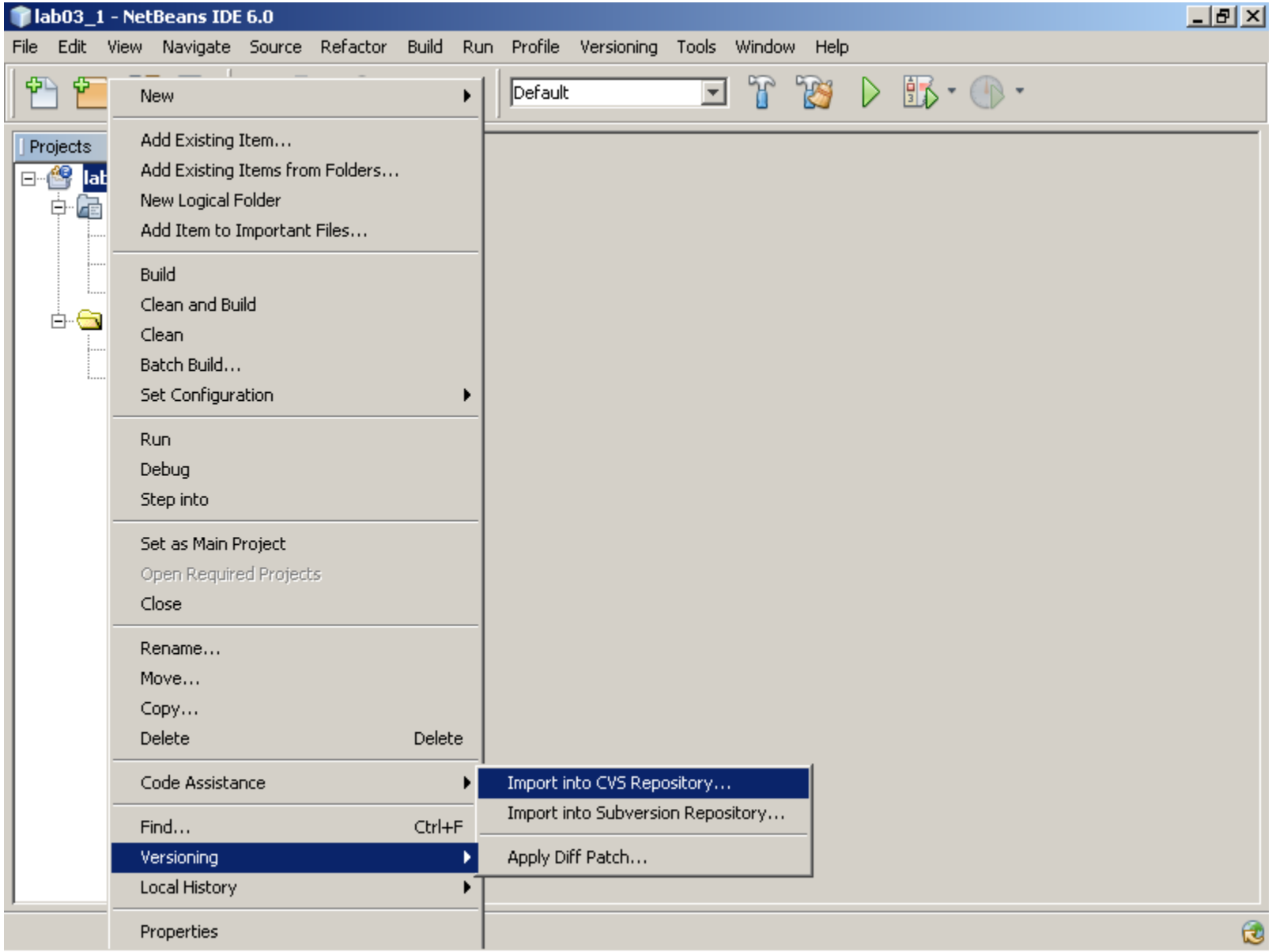

New Project
eps
Choose Project
Build Tool
Build Actions
Source Files
Code Assistance Configuration
Project Name and Location
Build Actions
Specify working directory and build commands:
Working Directory:
C:/yunglu/teaching/ece462/lab03
Browse...
Build Command:
make -f Makefile
Clean Command:
make -f Makefile clean
Build Result:
Browse...
Working Directory is the directory in which the Build and Clean commands are executed. The Build and Clean commands are commands that map the IDE's Build and Clean actions to shell commands. They could be executing an existing Makefile like "make -f Makefile" and "make -f Makefile clean". Build Result is the full path to the output of the Build Command and could be a path to an executable or library.
< Back
Next >
Finish
Cancel
Help


New Project
eps
Choose Project
Build Tool
Build Actions
Source Files
Code Assistance Configuration
Project Name and Location
Source Files
Specify the folders that contain the source files for the project:
Add subfolders
Source File Folder
C:\yunglu\teaching\ece462\lab03
Add Folder...
Remove
File types to add to the project:
.C .c++ .cc .cpp .cxx .mm .c .i .m .H .hpp .hxx .SUNWCCh .tcc .f .f03 .f90 .
Source files from the specified folders will be added to the project.
< Back
Next >
Finish
Cancel
Help

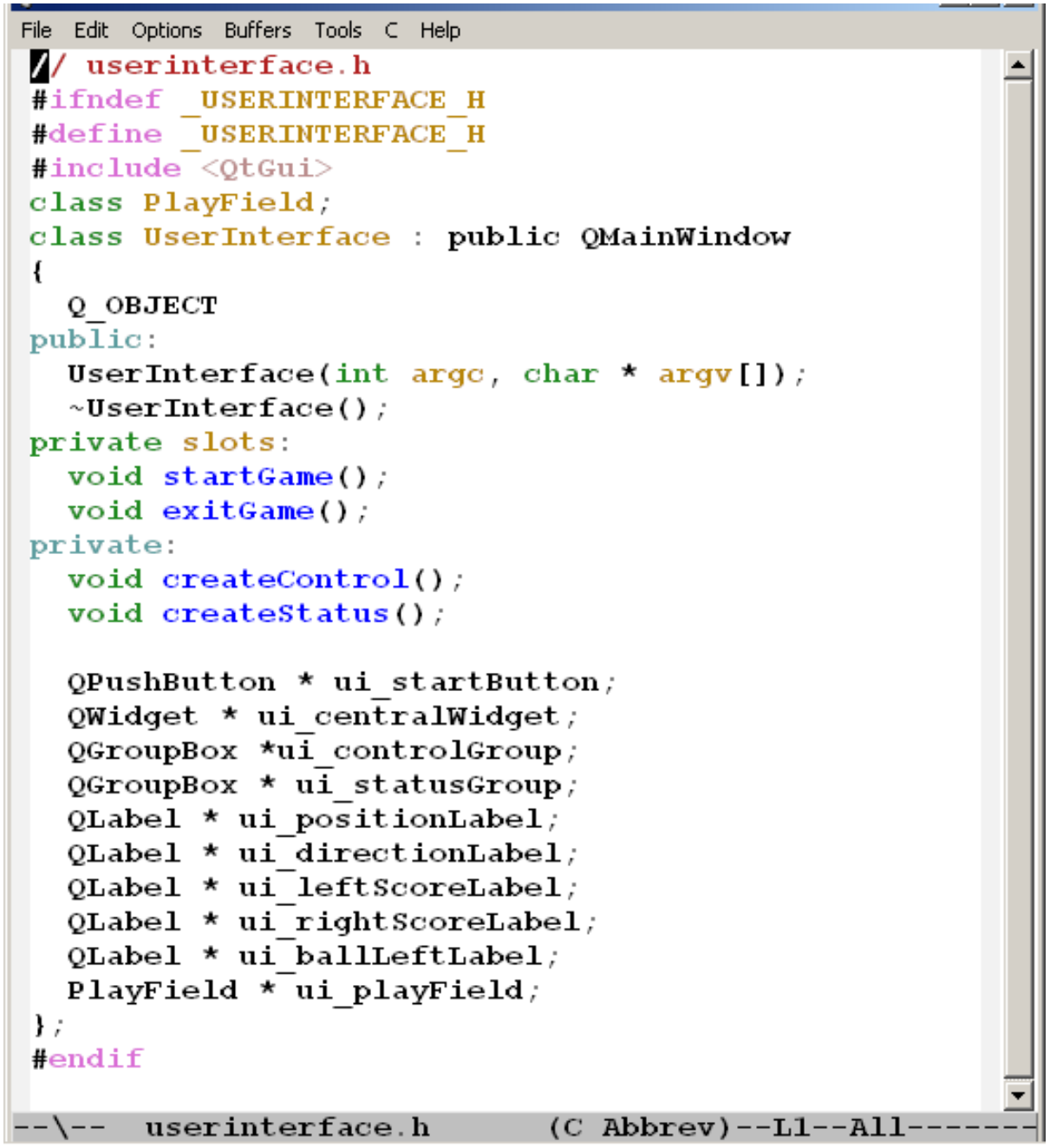

New Project
eps
Choose Project
Build Tool
Build Actions
Source Files
Code Assistance Configuration
Project Name and Location
Code Assistance Configuration
Manual Configuration
Specify compiler settings used by the built-in parser:
Include Directories:
Edit
Macro Definitions:
Edit
Automatic Configuration
Additional Libraries:
Edit
Configuration Level:
Configure File Properties
Include directories and macro definitions are used by the built-in parser when parsing source files.
< Back
Next >
Finish
Cancel
Help


New C/C++ Project From Existing Code
eps
Choose Project
Build Tool
Build Actions
Source Files
Code Assistance Configuration
Project Name and Location
Project Name and Location
Project Name:
lab03_1
Project Location:
Documents and Settings\yunglu\My Documents\NetBeansProjects
Browse...
Project Folder:
ts and Settings\yunglu\My Documents\NetBeansProjects\lab03_1
Set as Main Project
< Back
Next >
Finish
Cancel
Help


lab03_1 - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Default
Projects
Classes
lab03_1
lab03
main.cpp
userinterface.cpp
userinterface.h
Important Files
lab03_1-Makefile.mk
Makefile


lab03_1 - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Default
Projects
New
Add Existing Item...
Add Existing Items from Folders...
New Logical Folder
Add Item to Important Files...
Build
Clean and Build
Clean
Batch Build...
Set Configuration
Run
Debug
Step into
Set as Main Project
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Code Assistance
Find... Ctrl+F
Versioning
Local History
Properties
Import into CVS Repository...
Import into Subversion Repository...
Apply Diff Patch...

Add Buttons and Labels

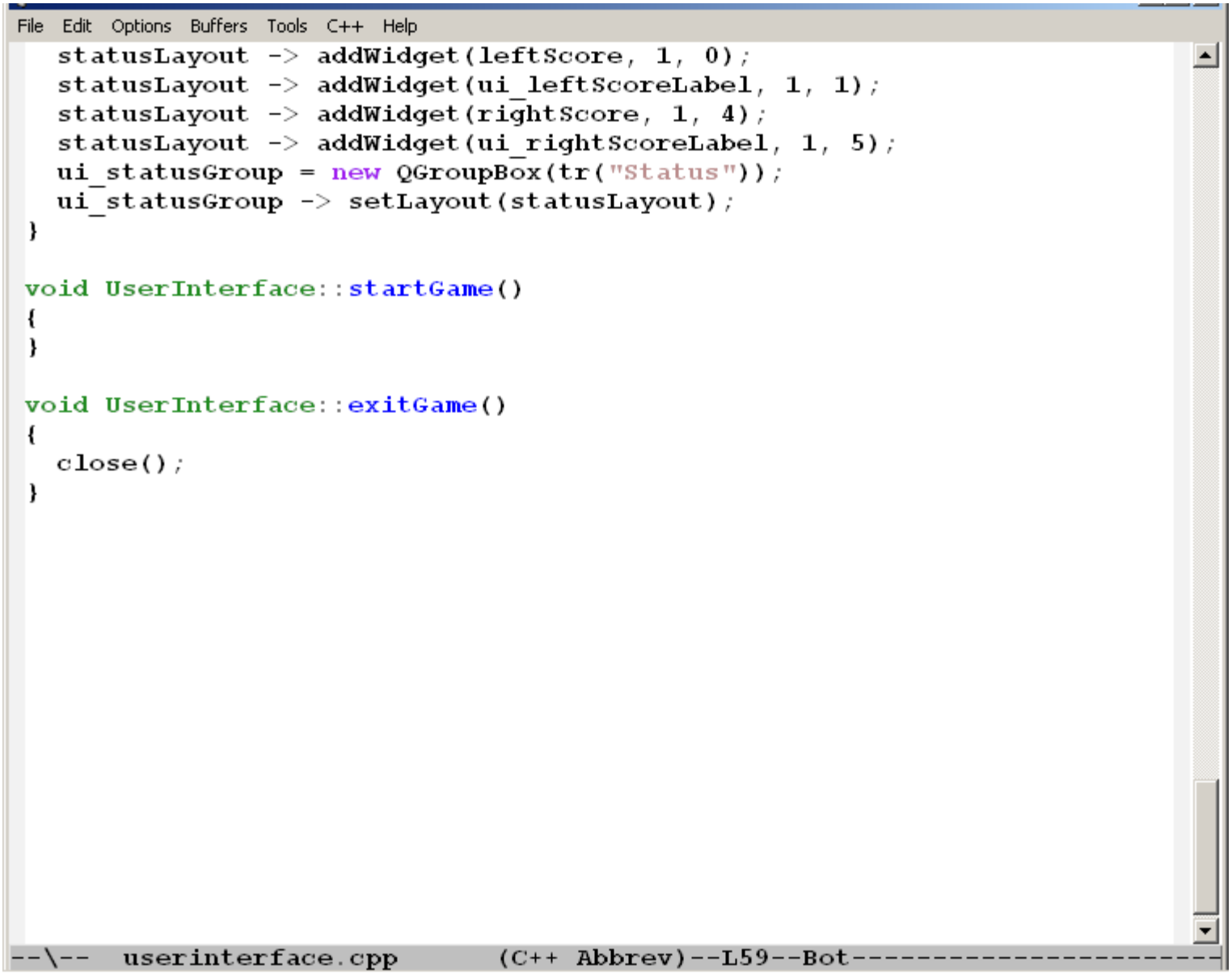

File Edit Options Buffers Tools C Help
// userinterface.h
#ifndef _USERINTERFACE_H
#define _USERINTERFACE_H
#include <QtGui>
class PlayField;
class UserInterface : public QMainWindow
{
Q_OBJECT
public:
UserInterface(int argc, char * argv[]);
~UserInterface();
private slots:
void startGame();
void exitGame();
private:
void createControl();
void createStatus();
QPushButton * ui_startButton;
QWidget * ui_centralWidget;
QGroupBox *ui_controlGroup;
QGroupBox * ui_statusGroup;
QLabel * ui_positionLabel;
QLabel * ui_directionLabel;
QLabel * ui_leftScoreLabel;
QLabel * ui_rightScoreLabel;
QLabel * ui_ballLeftLabel;
PlayField * ui_playField;
};
#endif
--\-- userinterface.h (C Abbrev)--L1--All-------

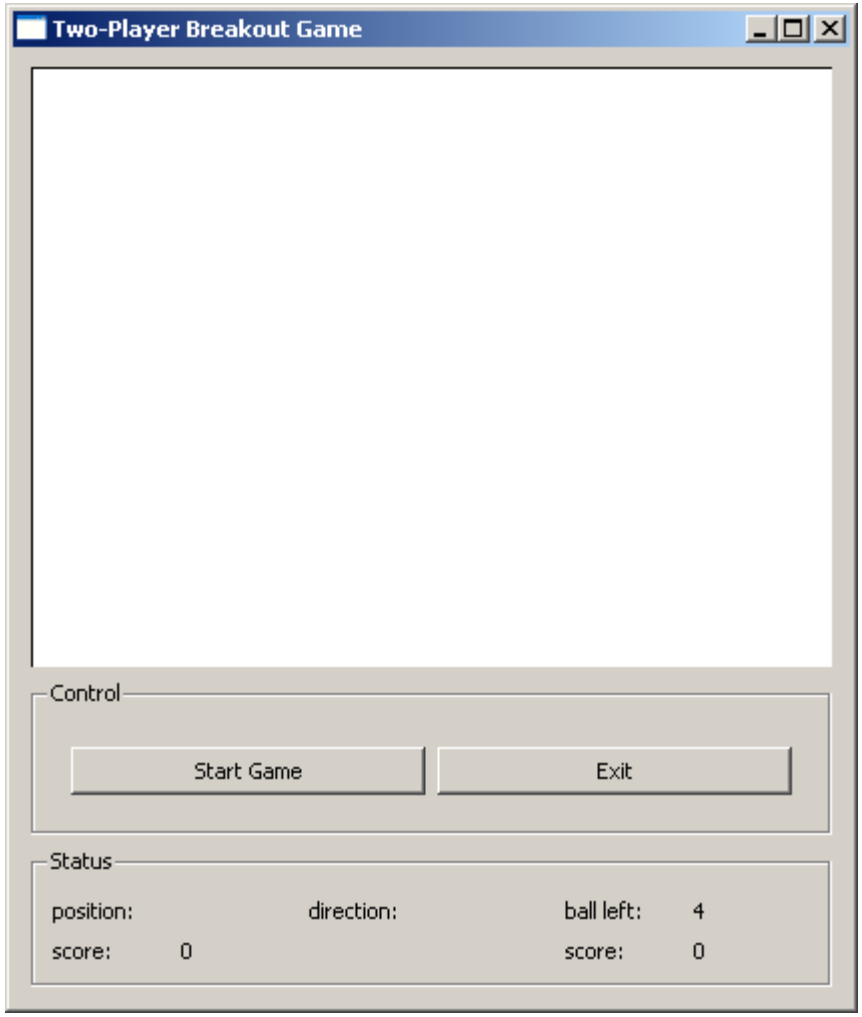
File Edit Options Buffers Tools C++ Help

```
#include "userinterface.h"
#include "playfield.h"
#include <iostream>
using namespace std;
UserInterface::UserInterface(int argc, char * argv[])
{
  createControl();
  createStatus();
  ui_centralWidget = new QWidget;
  QVBoxLayout * mainLayout = new QVBoxLayout;
  ui_playField = new PlayField;
  ui_centralWidget -> setLayout(mainLayout);
  mainLayout -> addWidget(ui_playField);
  mainLayout -> addWidget(ui_controlGroup);
  mainLayout -> addWidget(ui_statusGroup);
  setCentralWidget(ui_centralWidget);
  setWindowTitle(tr("Two-Player Breakout Game"));
}
UserInterface::~UserInterface()
{
}

void UserInterface::createControl()
{
  QVBoxLayout * controlLayout = new QVBoxLayout;
  QWidget *startExitWidget = new QWidget;
  QHBoxLayout * startExitLayout = new QHBoxLayout;
  ui_startButton = new QPushButton("Start Game");
  QPushButton * exitButton = new QPushButton("Exit");
  startExitLayout -> addWidget(ui_startButton);
  startExitLayout -> addWidget(exitButton);
```

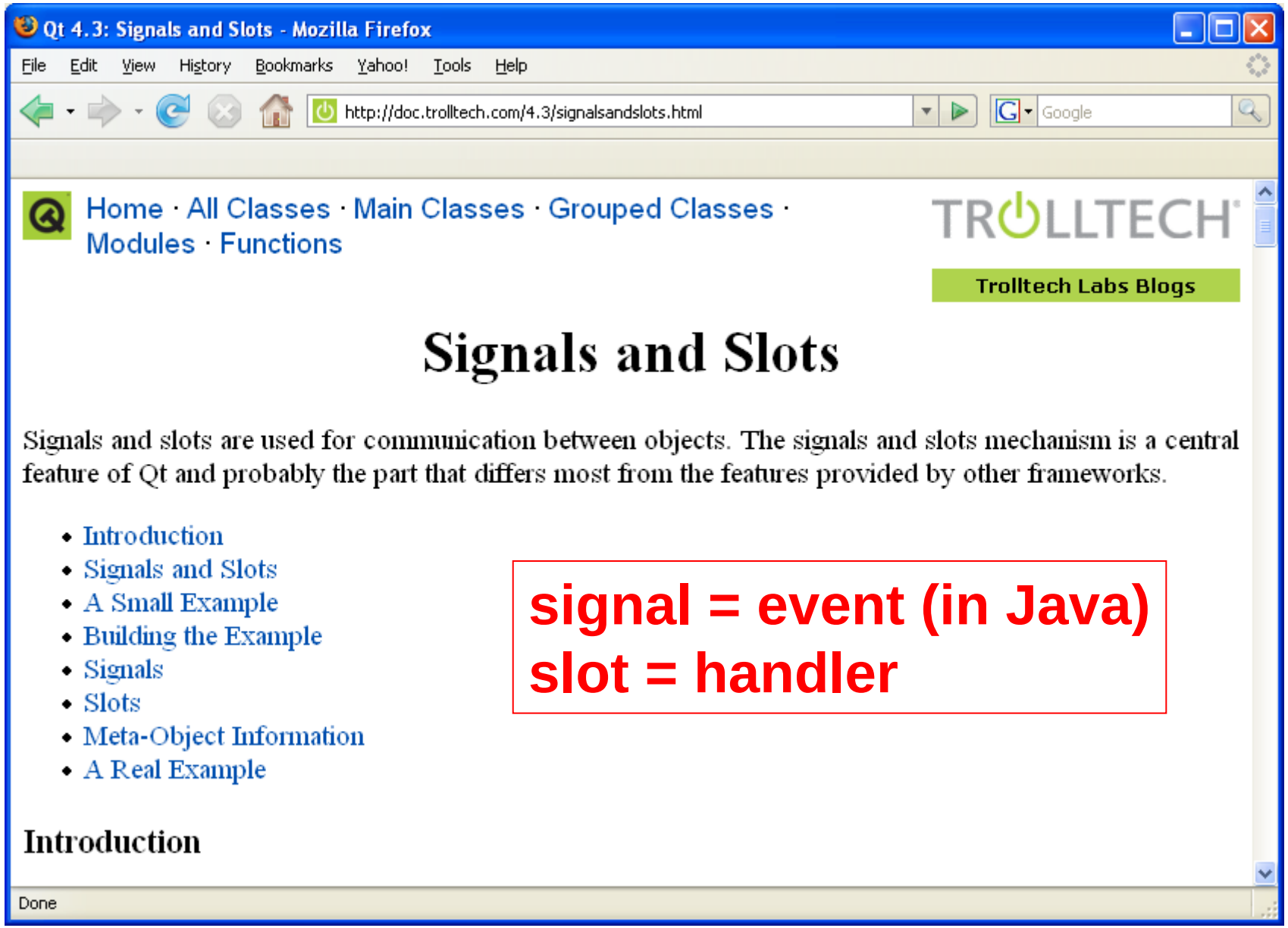
--\-- userinterface.cpp (C++ Abbrev)--L1--Top----------

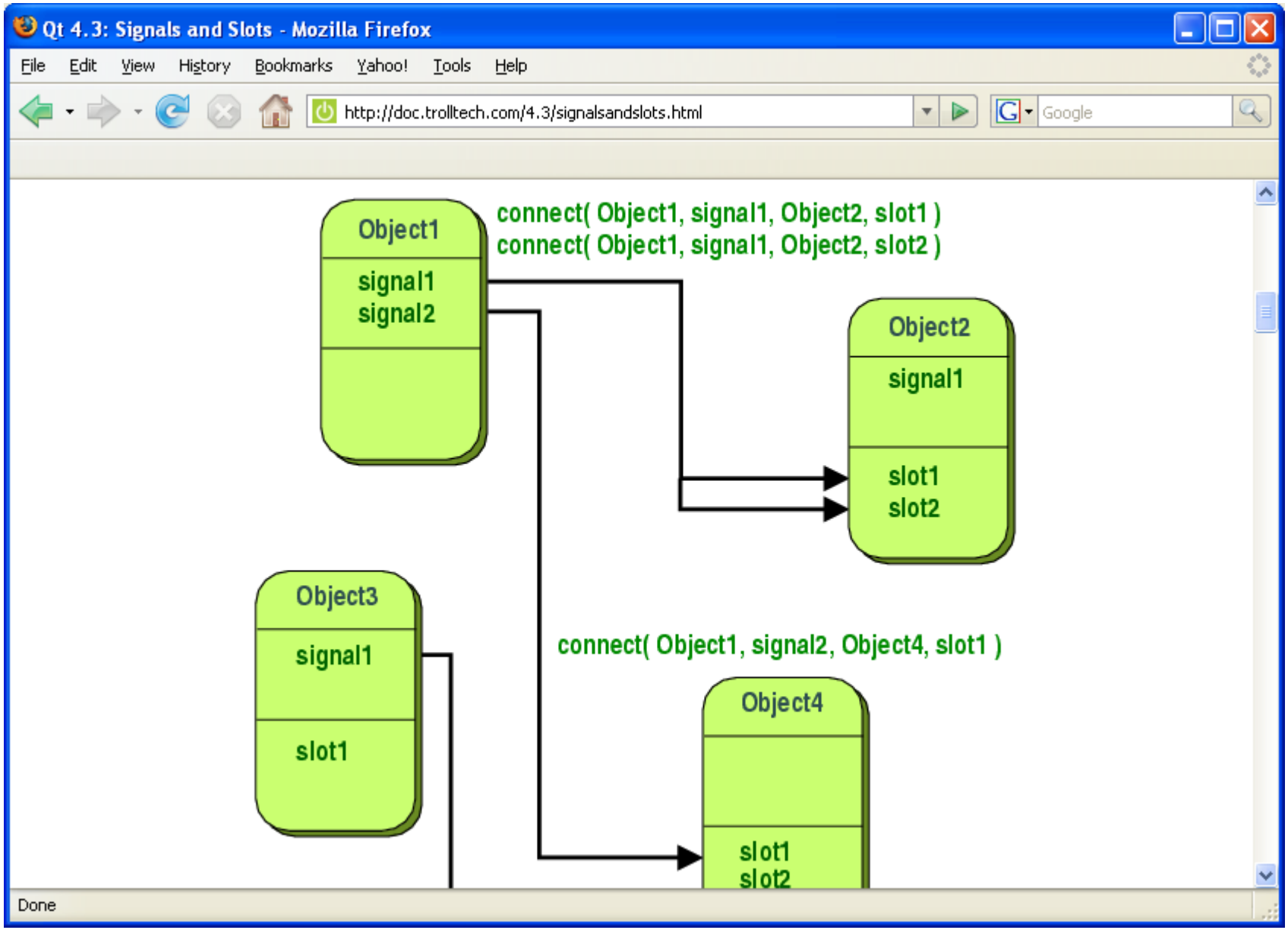
```
File  Edit  Options  Buffers  Tools  C++  Help
  startExitLayout -> addWidget(ui_startButton);
  startExitLayout -> addWidget(exitButton);
  startExitWidget -> setLayout(startExitLayout);
  connect(ui_startButton, SIGNAL(clicked()), this, SLOT(startGame()));
  connect(exitButton, SIGNAL(clicked()), this, SLOT(exitGame()));
  controlLayout -> addWidget(startExitWidget);
  ui_controlGroup = new QGroupBox(tr("Control"));
  ui_controlGroup -> setLayout(controlLayout);
}

void UserInterface::createStatus()
{
  QGridLayout * statusLayout = new QGridLayout;
  QLabel * position = new QLabel("position: ");
  ui_positionLabel = new QLabel("      ");
  QLabel * direction = new QLabel("direction: ");
  ui_directionLabel = new QLabel("      ");
  QLabel * leftScore = new QLabel("score: ");
  ui_leftScoreLabel = new QLabel("0");
  QLabel * ballLeft = new QLabel("ball left: ");
  ui_ballLeftLabel = new QLabel("4");
  QLabel * rightScore = new QLabel("score: ");
  ui_rightScoreLabel = new QLabel("0");
  statusLayout -> addWidget(position, 0, 0);
  statusLayout -> addWidget(ui_positionLabel, 0, 1);
  statusLayout -> addWidget(direction, 0, 2);
  statusLayout -> addWidget(ui_directionLabel, 0, 3);
  statusLayout -> addWidget(ballLeft, 0, 4);
  statusLayout -> addWidget(ui_ballLeftLabel, 0, 5);
  statusLayout -> addWidget(leftScore, 1, 0);
  statusLayout -> addWidget(ui_leftScoreLabel, 1, 1);
--\--  userinterface.cpp      (C++ Abbrev)--L30--36%----------------------
```


File Edit Options Buffers Tools C++ Help
statusLayout -> addWidget(leftScore, 1, 0);
statusLayout -> addWidget(ui_leftScoreLabel, 1, 1);
statusLayout -> addWidget(rightScore, 1, 4);
statusLayout -> addWidget(ui_rightScoreLabel, 1, 5);
ui_statusGroup = new QGroupBox(tr("Status"));
ui_statusGroup -> setLayout(statusLayout);
}
void UserInterface::startGame()
{
}
void UserInterface::exitGame()
{
close();
}
--\-- userinterface.cpp (C++ Abbrev)--L59--Bot

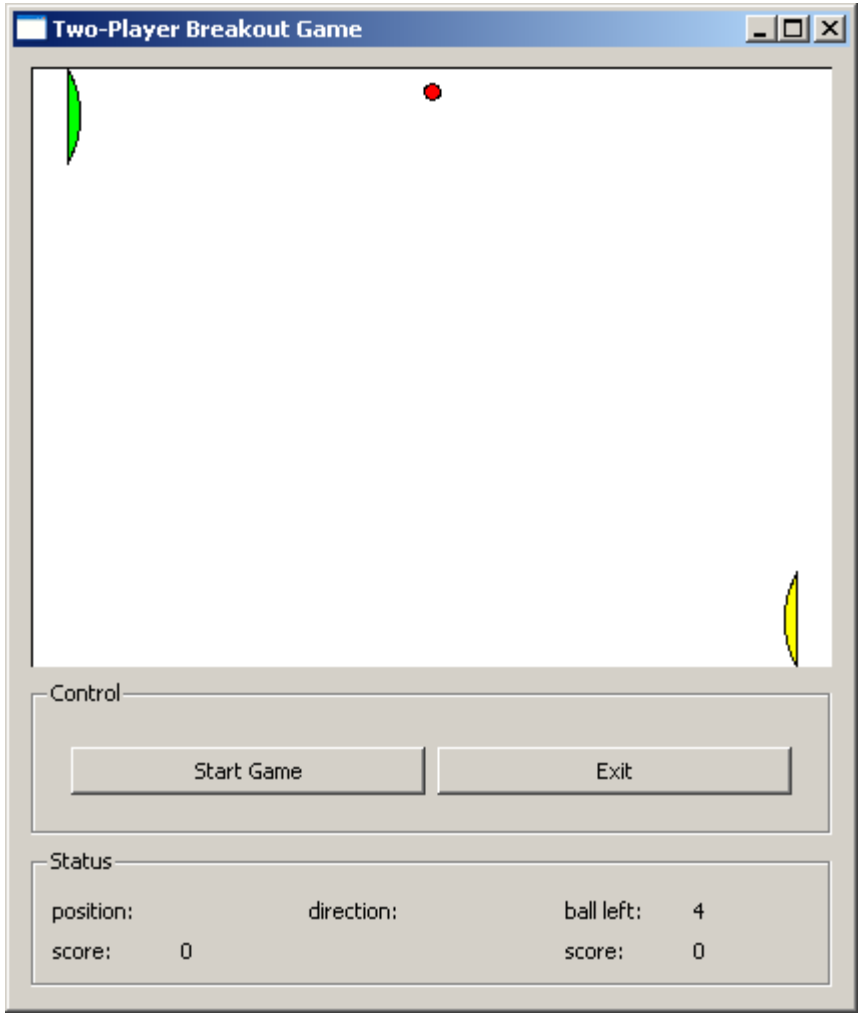

Two-Player Breakout Game
Control
Start Game
Exit
Status
position:
direction:
ball left: 4
score: 0
score: 0

Handle User Inputs (Signals)


Qt 4.3: Signals and Slots - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://doc.trolltech.com/4.3/signalsandslots.html
Google
Home · All Classes · Main Classes · Grouped Classes · Modules · Functions
TROLLTECH
Trolltech Labs Blogs
Signals and Slots
Signals and slots are used for communication between objects. The signals and slots mechanism is a central feature of Qt and probably the part that differs most from the features provided by other frameworks.
Introduction
Signals and Slots
A Small Example
Building the Example
Signals
Slots
Meta-Object Information
A Real Example
signal = event (in Java)
slot = handler
Introduction
Done


Qt 4.3: Signals and Slots - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://doc.trolltech.com/4.3/signalsandslots.html
Google
Object1
signal1
signal2
connect(Object1, signal1, Object2, slot1)
connect(Object1, signal1, Object2, slot2)
Object2
signal1
slot1
slot2
Object3
signal1
slot1
connect(Object1, signal2, Object4, slot1)
Object4
slot1
slot2
Done

Add Paddles and a Bouncing Ball


Two-Player Breakout Game
Control
Start Game
Exit
Status
position:
direction:
ball left: 4
score: 0
score: 0

File Edit Options Buffers Tools C Help

```
#ifndef _TWODVECTOR_H_
#define _TWODVECTOR_H_
// This class handles two-dimensional analytic geometry. C++ or
// Qt's vector classes are container classes.
//
class TwoDVector
{
public:
  TwoDVector(double x = 1.0, double y = 1.0, bool unit = true);
  void reflect(const TwoDVector & vin, TwoDVector & vout) const;
  // the object is the unit vector, given the incident vector, find
  // the reflective vector. using passing-by-reference to reduce the
  // overhead of copying objects
  double innerProduct(const TwoDVector & v2) const;
  double getX() { return tdv_x; }
  double getY() { return tdv_y; }
  void addVector(const TwoDVector & vec);
  void print();
private:
  void makeUnit(); // keep it a unit vector
  double tdv_x;
  double tdv_y;
  bool unitVector;
};
extern void testVectorReflect();
// generate a set of vin and normal to test vout

#endif
```

--\-- twodvector.h (C Abbrev)--L20--All----------

File Edit Options Buffers Tools C++ Help

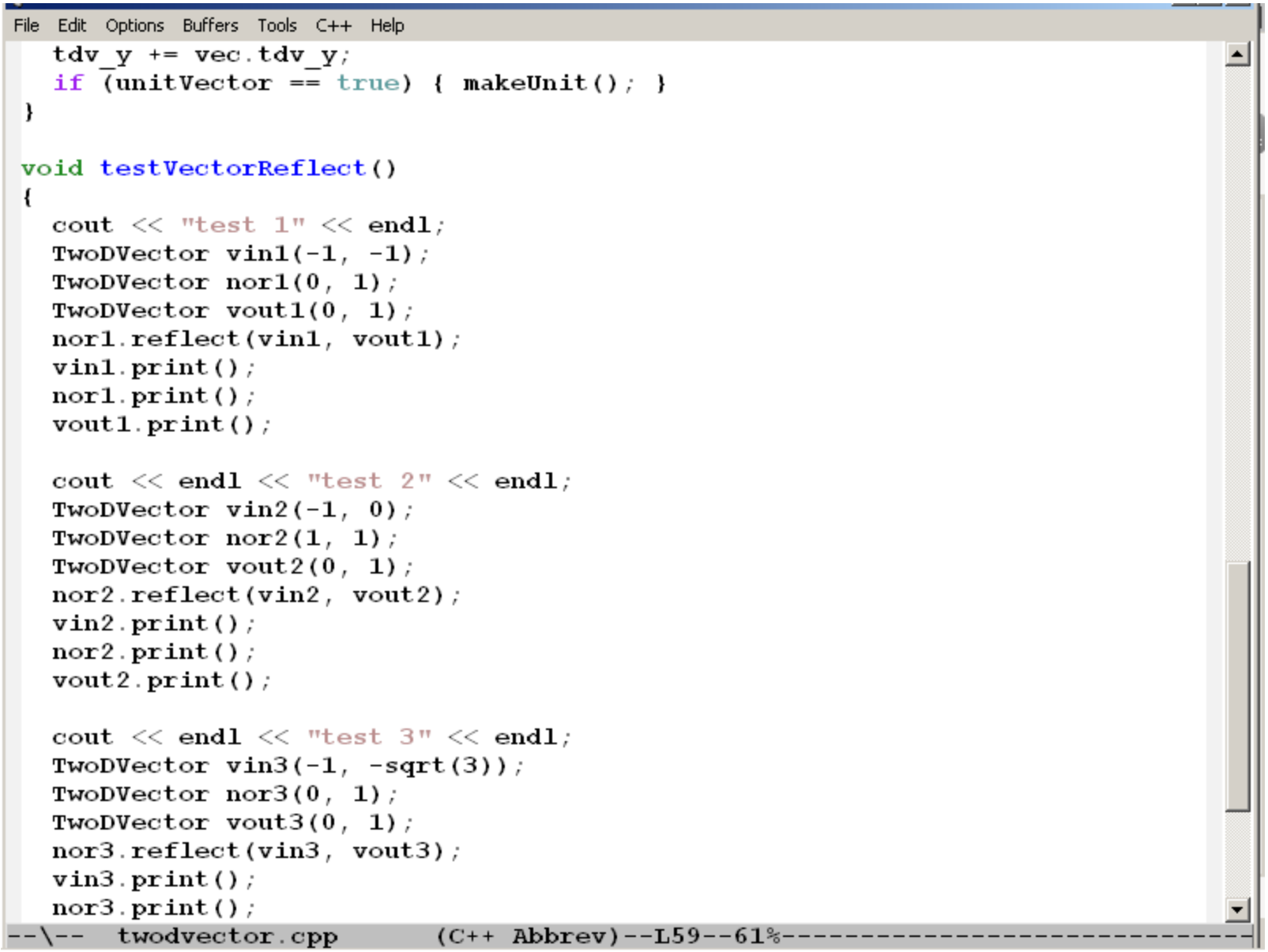
```
#include <iostream>
#include <math.h>
#include "twodvector.h"
using namespace std;
TwoDVector::TwoDVector(double x, double y, bool unit)
{
  tdv_x = x;
  tdv_y = y;
  unitVector = unit;
  if (unitVector == true) { makeUnit(); }
}

void TwoDVector::makeUnit()
{
  if ((tdv_x == 0) && (tdv_y == 0))
    { cerr << "ERROR, a (0, 0) vector" << endl; return; }
  if (tdv_x == 0)
    {
      if (tdv_y > 0) { tdv_y = 1.0; return; }
      tdv_y = -1.0;
      return;
    }
  if (tdv_y == 0)
    {
      if (tdv_x > 0) { tdv_x = 1.0; return; }
      tdv_x = -1.0;
      return;
    }
  double magnitude = sqrt(tdv_x * tdv_x + tdv_y * tdv_y);
  tdv_x /= magnitude;
  tdv_y /= magnitude;
```

--\-- twodvector.cpp (C++ Abbrev)--L7--Top----------

File Edit Options Buffers Tools C++ Help

```
  tdv_x /= magnitude;
  tdv_y /= magnitude;
}

double TwoDVector::innerProduct(const TwoDVector & v2) const
{
  return (tdv_x * v2.tdv_x + tdv_y * v2.tdv_y);
}

void TwoDVector::reflect(const TwoDVector & vin, TwoDVector & vout) const
{
  if ((unitVector == false) ||
      (vin.unitVector == false) ||
      (vout.unitVector == false))
    { cout << "ERROR! vector must unit" << endl; }
  double s = -2 * innerProduct(vin);
  vout.tdv_x = vin.tdv_x + s * tdv_x;
  vout.tdv_y = vin.tdv_y + s * tdv_y;
  vout.makeUnit();
}

void TwoDVector::print()
{
  cout << "(" << tdv_x << "," << tdv_y << ")" << endl;
}

void TwoDVector::addVector(const TwoDVector & vec)
{
  tdv_x += vec.tdv_x;
  tdv_y += vec.tdv_y;
  if (unitVector == true) { makeUnit(); }
```

--\-- twodvector.cpp (C++ Abbrev)--L30--30%----------


File Edit Options Buffers Tools C++ Help
tdv_y += vec.tdv_y;
if (unitVector == true) { makeUnit(); }
}
void testVectorReflect()
{
cout << "test 1" << endl;
TwoDVector vin1(-1, -1);
TwoDVector nor1(0, 1);
TwoDVector vout1(0, 1);
nor1.reflect(vin1, vout1);
vin1.print();
nor1.print();
vout1.print();
cout << endl << "test 2" << endl;
TwoDVector vin2(-1, 0);
TwoDVector nor2(1, 1);
TwoDVector vout2(0, 1);
nor2.reflect(vin2, vout2);
vin2.print();
nor2.print();
vout2.print();
cout << endl << "test 3" << endl;
TwoDVector vin3(-1, -sqrt(3));
TwoDVector nor3(0, 1);
TwoDVector vout3(0, 1);
nor3.reflect(vin3, vout3);
vin3.print();
nor3.print();
--\-- twodvector.cpp (C++ Abbrev)--L59--61%

File Edit Options Buffers Tools C Help

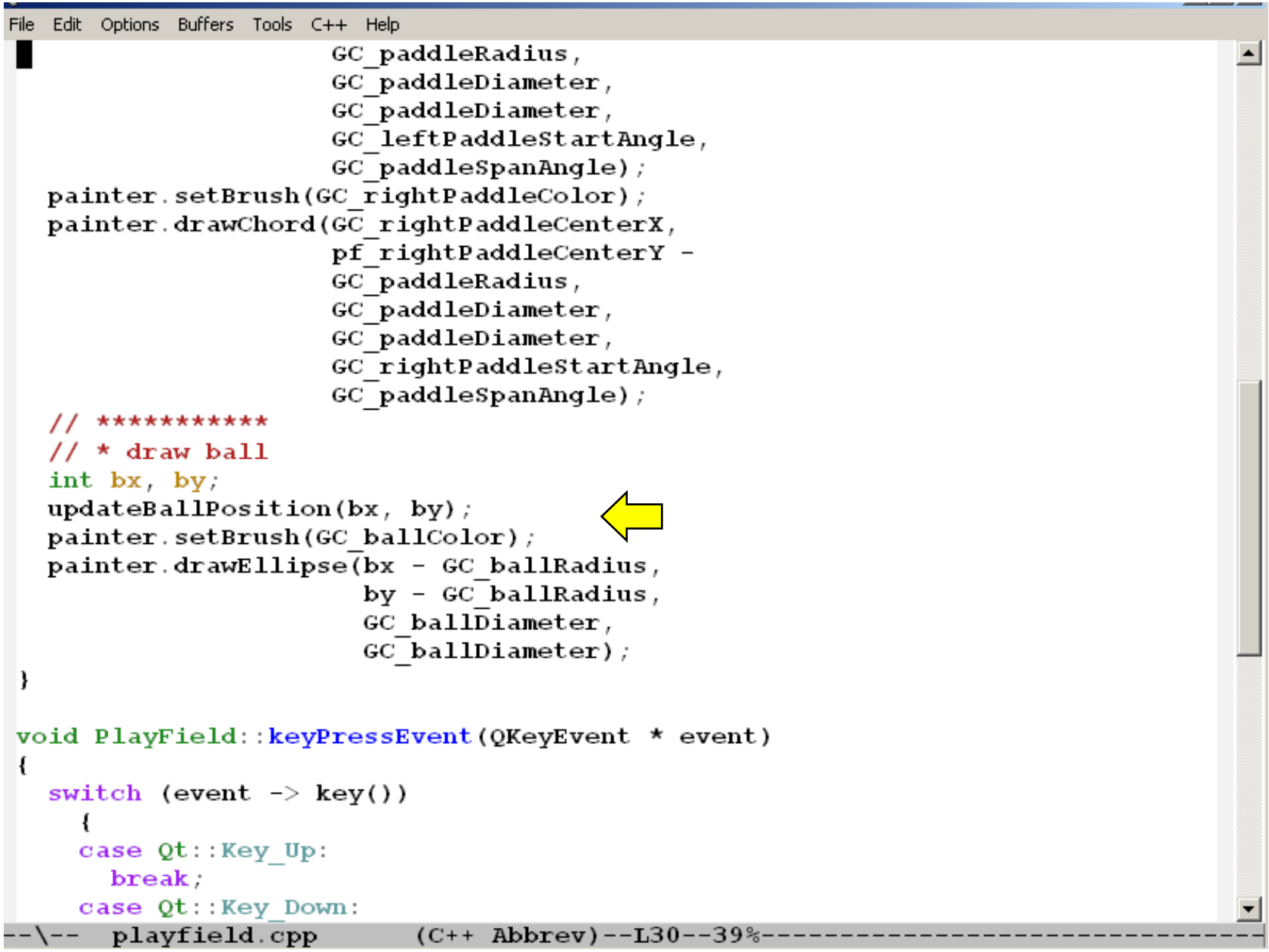
```
// playfield.h
#ifndef _PLAYFIELD_H
#define _PLAYFIELD_H
#include <QtGui>
#include "twodvector.h"
class PlayField: public QWidget
{
public:
  PlayField(QWidget *parent = 0);
  // no destructor since no attributed are created using new
protected:
  void paintEvent(QPaintEvent *event);
  void keyPressEvent(QKeyEvent * event);
  void mouseMoveEvent(QMouseEvent * event);
private:
  int pf_leftPaddleCenterY;
  int pf_rightPaddleCenterY;
  TwoDVector pf_ballPosition; // convert to int only for drawing
  TwoDVector pf_ballVelocity; // unit vector
  void updateBallPosition(int & bx, int & by);
};
#endif
```

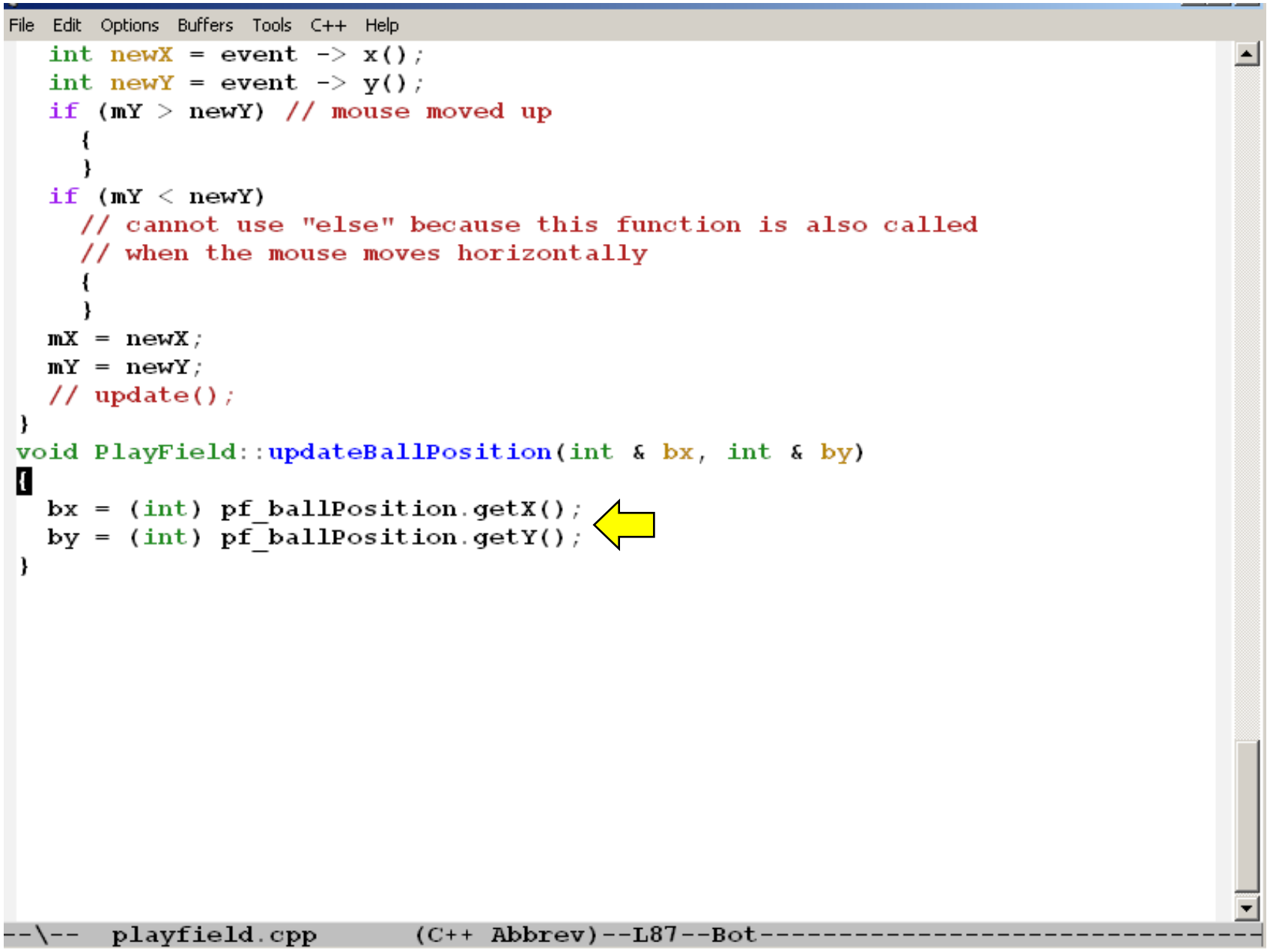
--\-- playfield.h (C Abbrev)--L5--All----------

File Edit Options Buffers Tools C++ Help

```
// playfield.cpp
#include <QtGui>
#include <iostream>
#include "playfield.h"
#include "constant.h"
using namespace std;
PlayField::PlayField(QWidget *parent): QWidget(parent)
{
  setMinimumSize(GC_fieldWidth, GC_fieldHeight);
  setMaximumSize(GC_fieldWidth, GC_fieldHeight);
  setMouseTracking(true);
  pf_leftPaddleCenterY = GC_paddleHalfHeight;
  pf_rightPaddleCenterY = GC_fieldHeight - GC_paddleHalfHeight;
  pf_ballPosition = TwoDVector(GC_fieldWidth/2, GC_ballRadius * 3, false);
}

void PlayField::paintEvent(QPaintEvent *event)
{
  int ww = width(); // widget's width
  int wh = height(); // widget's height
  // draw background
  QPainter painter(this);
  painter.setBrush(GC_backgroundColor);
  painter.drawRect(0, 0, ww, wh);
  // ***************
  // * draw paddles
  painter.setBrush(GC_leftPaddleColor);
  painter.drawChord(GC_leftPaddleCenterX,
                    pf_leftPaddleCenterY -
                    GC_paddleRadius,
                    GC_paddleDiameter,
```

--\-- playfield.cpp (C++ Abbrev)--L1--Top----------


File Edit Options Buffers Tools C++ Help
GC_paddleRadius,
GC_paddleDiameter,
GC_paddleDiameter,
GC_leftPaddleStartAngle,
GC_paddleSpanAngle);
painter.setBrush(GC_rightPaddleColor);
painter.drawChord(GC_rightPaddleCenterX,
pf_rightPaddleCenterY -
GC_paddleRadius,
GC_paddleDiameter,
GC_paddleDiameter,
GC_rightPaddleStartAngle,
GC_paddleSpanAngle);
// ***********
// * draw ball
int bx, by;
updateBallPosition(bx, by);
painter.setBrush(GC_ballColor);
painter.drawEllipse(bx - GC_ballRadius,
by - GC_ballRadius,
GC_ballDiameter,
GC_ballDiameter);
}
void PlayField::keyPressEvent(QKeyEvent * event)
{
switch (event -> key())
{
case Qt::Key_Up:
break;
case Qt::Key_Down:
--\-- playfield.cpp (C++ Abbrev)--L30--39%


File Edit Options Buffers Tools C++ Help
int newX = event -> x();
int newY = event -> y();
if (mY > newY) // mouse moved up
{
}
if (mY < newY)
// cannot use "else" because this function is also called
// when the mouse moves horizontally
{
}
mX = newX;
mY = newY;
// update();
}
void PlayField::updateBallPosition(int & bx, int & by)
{
bx = (int) pf_ballPosition.getX();
by = (int) pf_ballPosition.getY();
}
--\-- playfield.cpp (C++ Abbrev)--L87--Bot

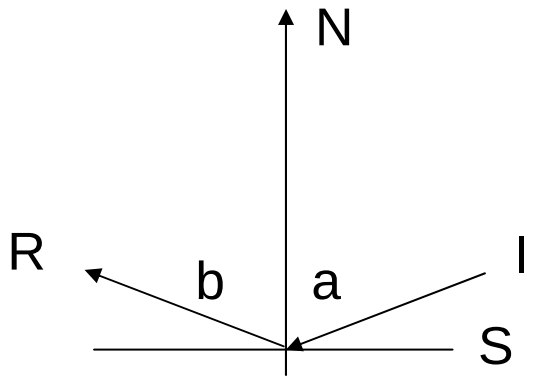
Move Ball
Track Mouse
Detect Collision

Law of Reflection

- S: surface of collision
- N: normal vector, N $\perp$ S
- I: incident vector
- R: reflection vector
- a: angle between I and N
- b: angle between R and N
- law of reflection: a = b
- If N, I, and R are unit vectors (|N| = |I| = |R| = 1),

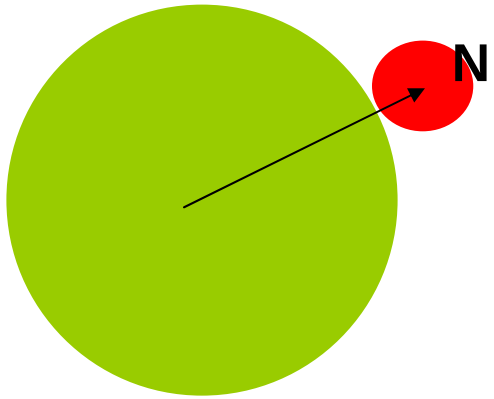
  R - I = 2 N $\cdot$ (-I), here $\cdot$ is the inner product.

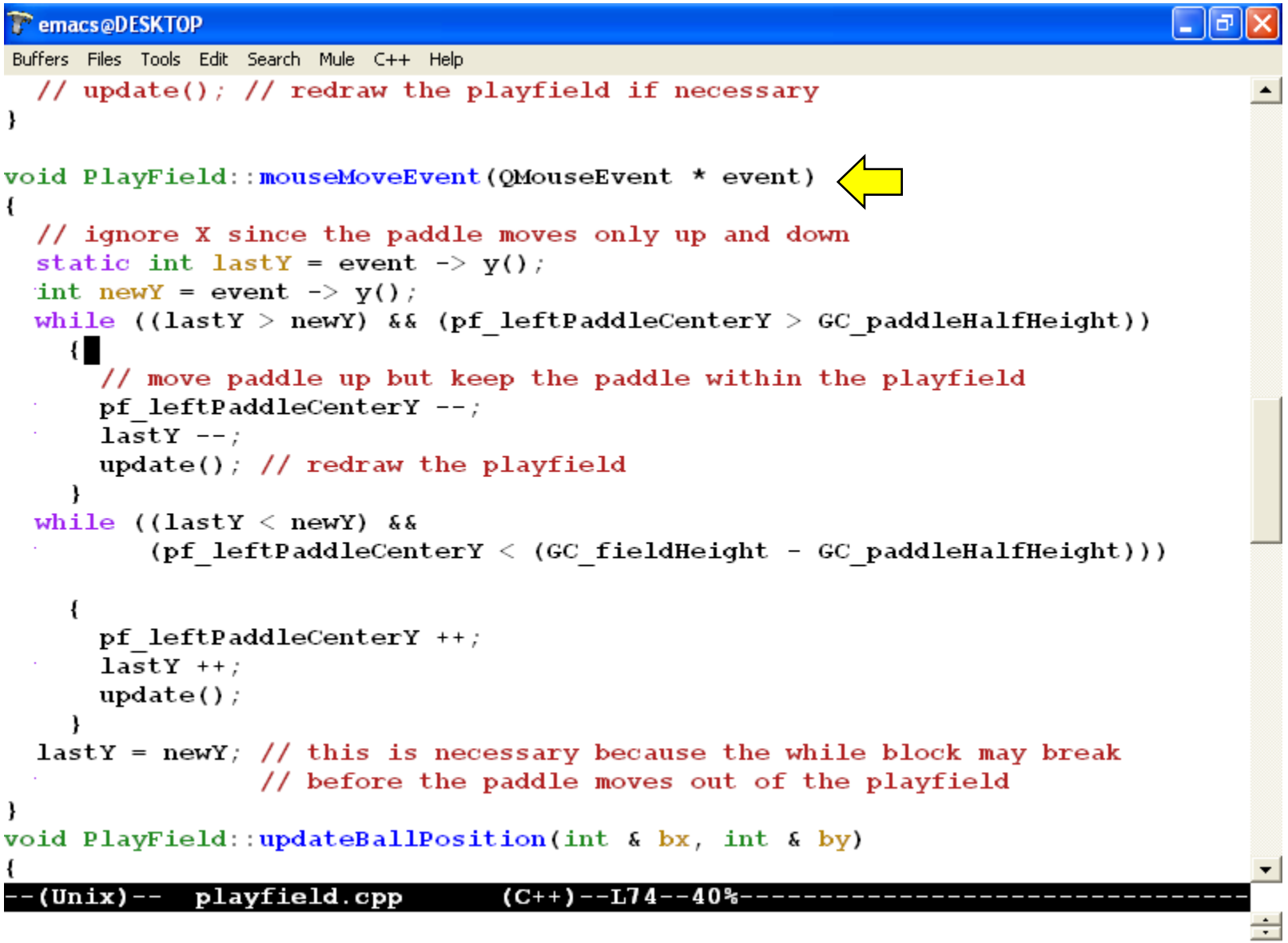
  $\Rightarrow$ R = I - 2 N $\cdot$ I

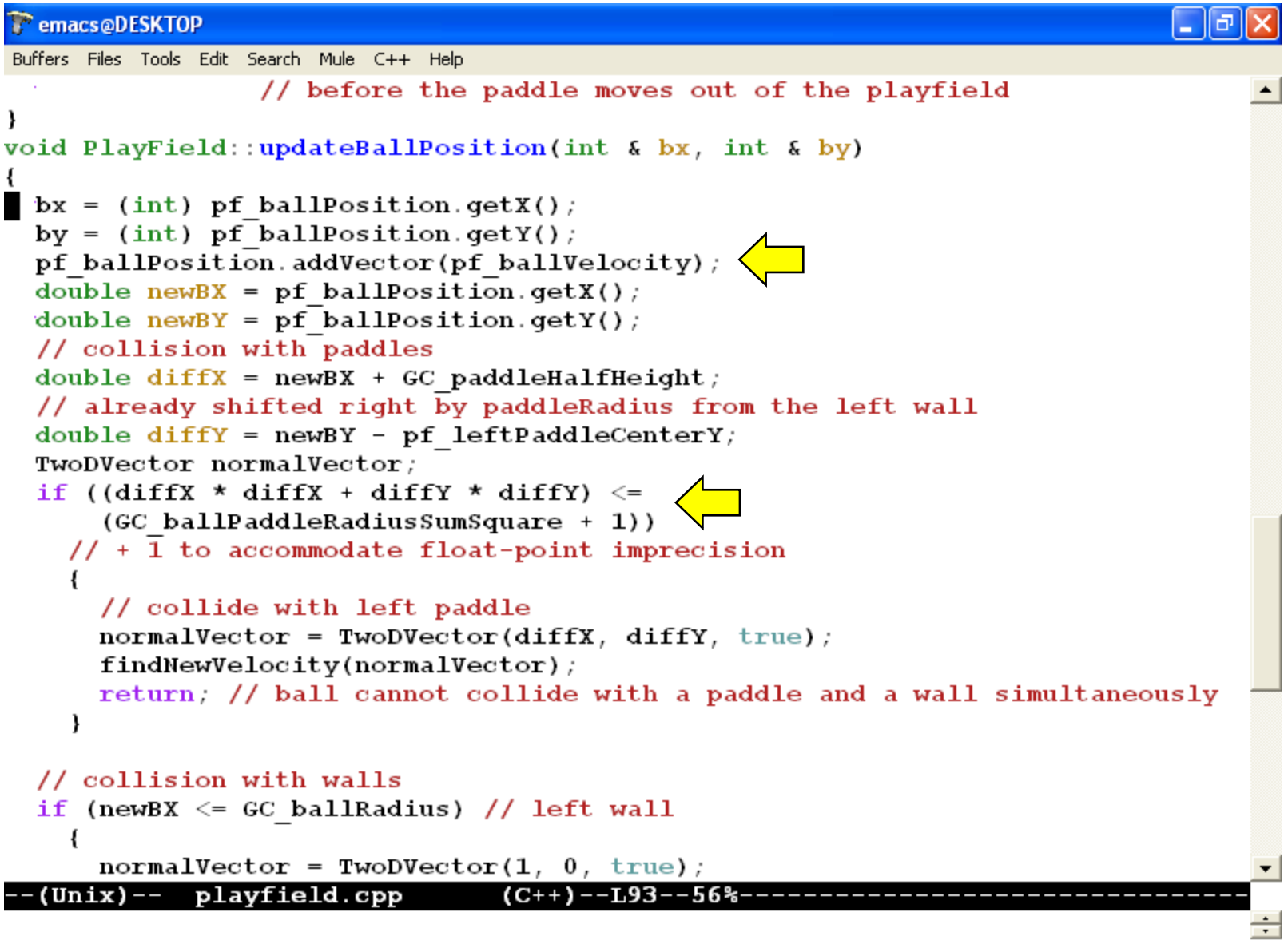


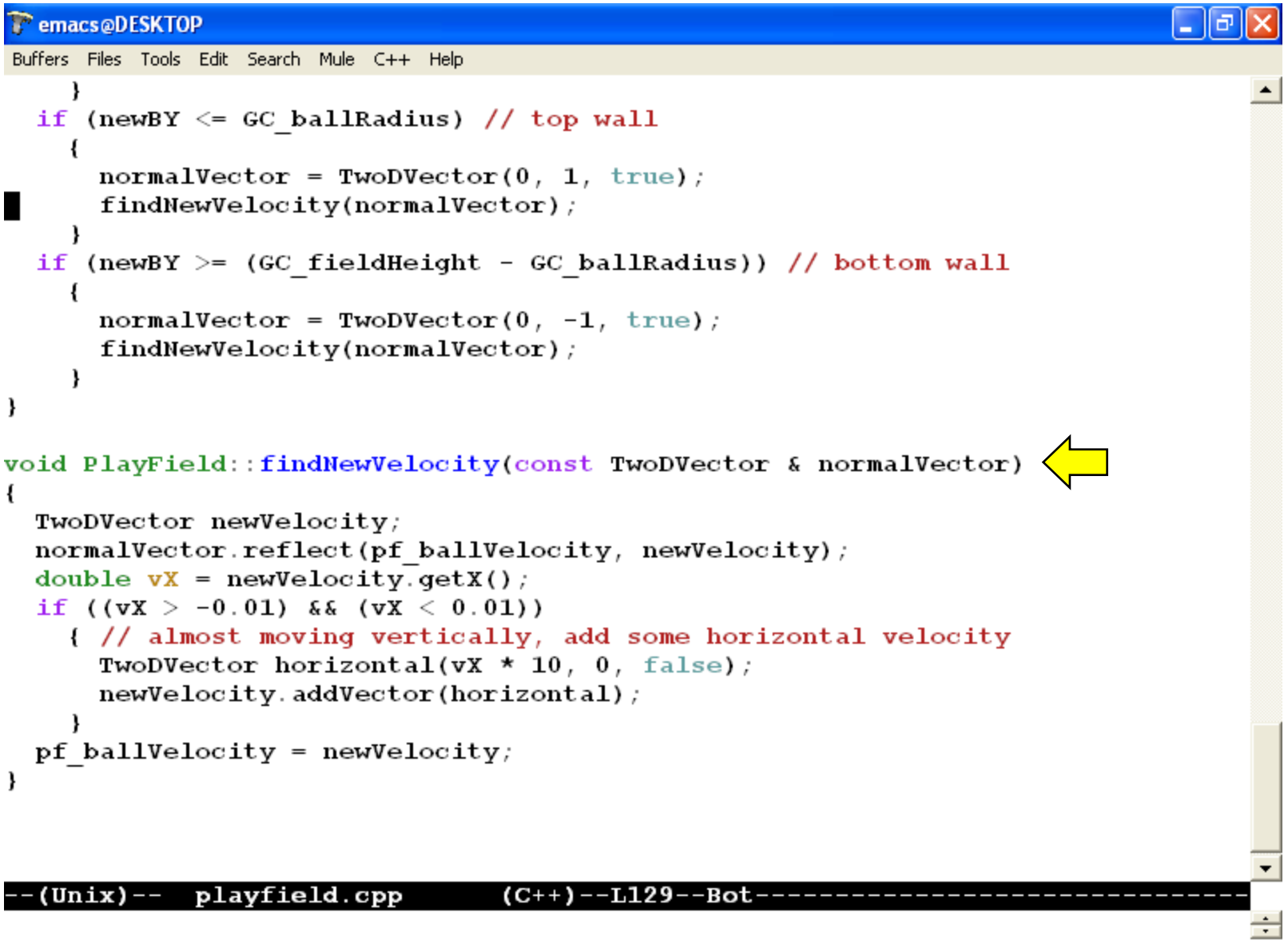
Ball Hits Paddle

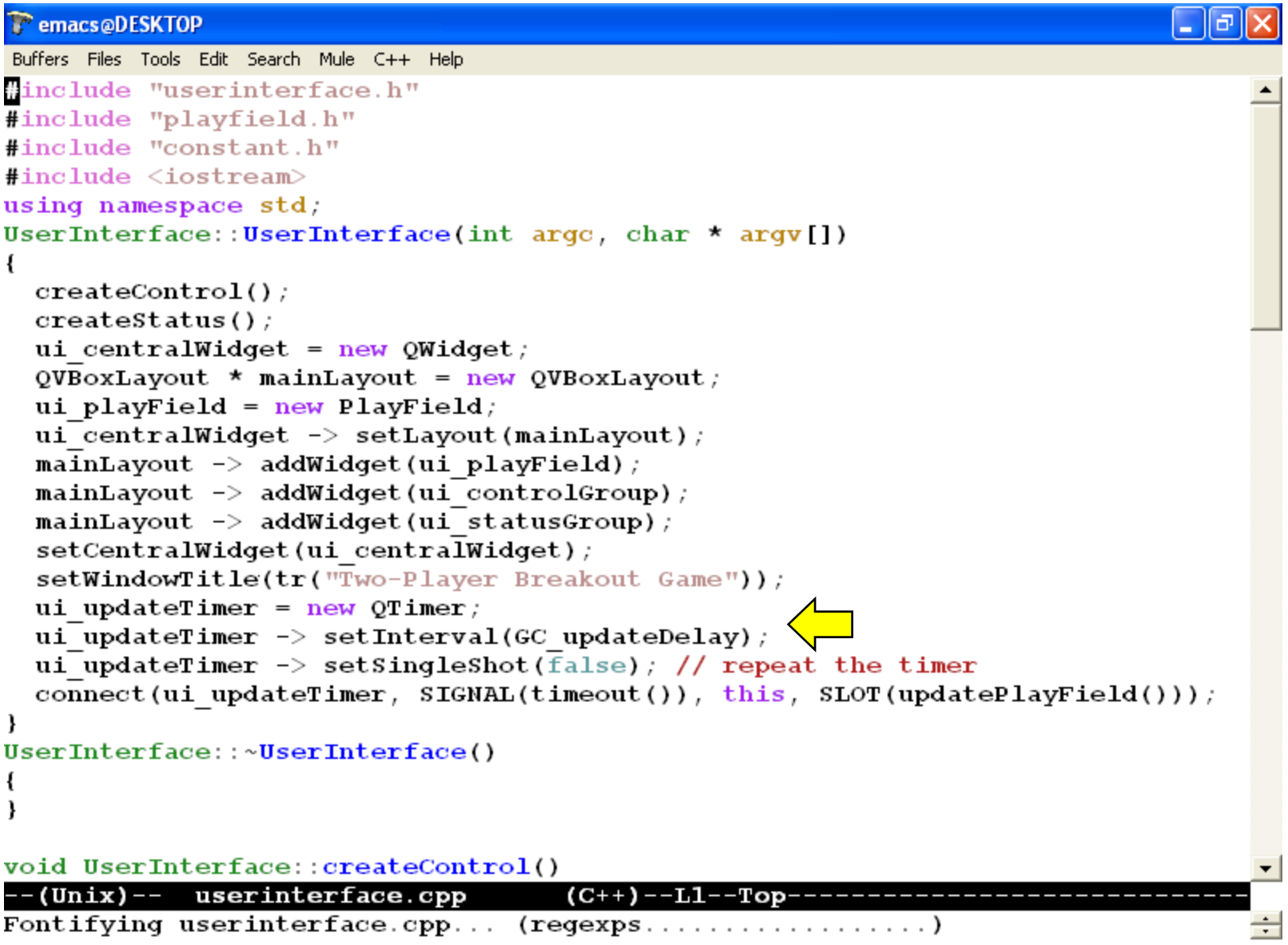
- Both are circles. The normal vector is the vector connecting the center of the ball and the center of the paddle.
- Collision occurs when the distance between the two centers is smaller than the sum of the radii.

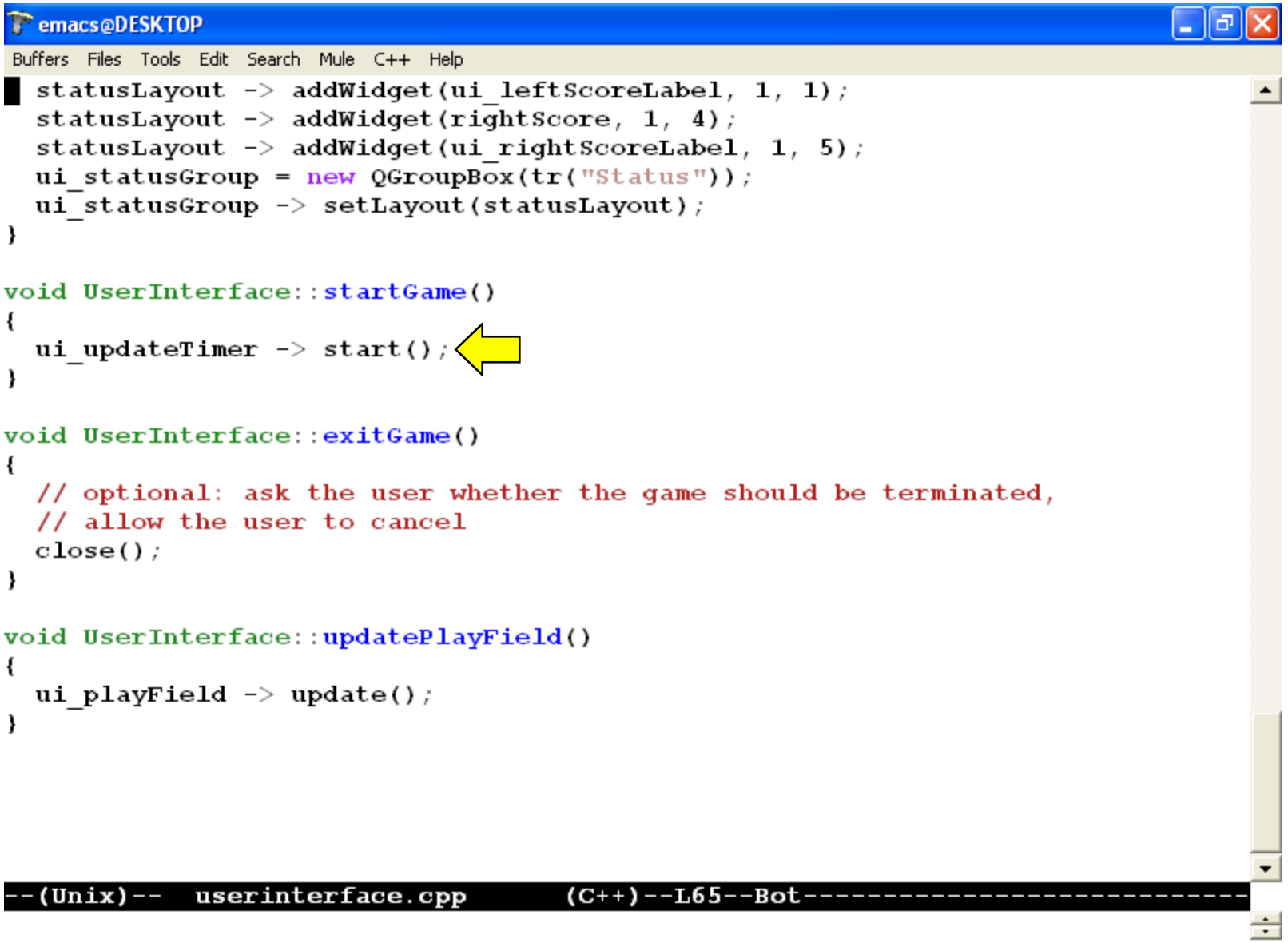



emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
// update(); // redraw the playfield if necessary
}
void PlayField::mouseMoveEvent(QMouseEvent * event)
{
// ignore X since the paddle moves only up and down
static int lastY = event -> y();
int newY = event -> y();
while ((lastY > newY) && (pf_leftPaddleCenterY > GC_paddleHalfHeight))
{
// move paddle up but keep the paddle within the playfield
pf_leftPaddleCenterY --;
lastY --;
update(); // redraw the playfield
}
while ((lastY < newY) &&
(pf_leftPaddleCenterY < (GC_fieldHeight - GC_paddleHalfHeight)))
{
pf_leftPaddleCenterY ++;
lastY ++;
update();
}
lastY = newY; // this is necessary because the while block may break
// before the paddle moves out of the playfield
}
void PlayField::updateBallPosition(int & bx, int & by)
{
--(Unix)-- playfield.cpp (C++)--L74--40%


emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
// before the paddle moves out of the playfield
}
void PlayField::updateBallPosition(int & bx, int & by)
{
bx = (int) pf_ballPosition.getX();
by = (int) pf_ballPosition.getY();
pf_ballPosition.addVector(pf_ballVelocity);
double newBX = pf_ballPosition.getX();
double newBY = pf_ballPosition.getY();
// collision with paddles
double diffX = newBX + GC_paddleHalfHeight;
// already shifted right by paddleRadius from the left wall
double diffY = newBY - pf_leftPaddleCenterY;
TwoDVector normalVector;
if ((diffX * diffX + diffY * diffY) <=
(GC_ballPaddleRadiusSumSquare + 1))
// + 1 to accommodate float-point imprecision
{
// collide with left paddle
normalVector = TwoDVector(diffX, diffY, true);
findNewVelocity(normalVector);
return; // ball cannot collide with a paddle and a wall simultaneously
}
// collision with walls
if (newBX <= GC_ballRadius) // left wall
{
normalVector = TwoDVector(1, 0, true);
--(Unix)-- playfield.cpp (C++)--L93--56%

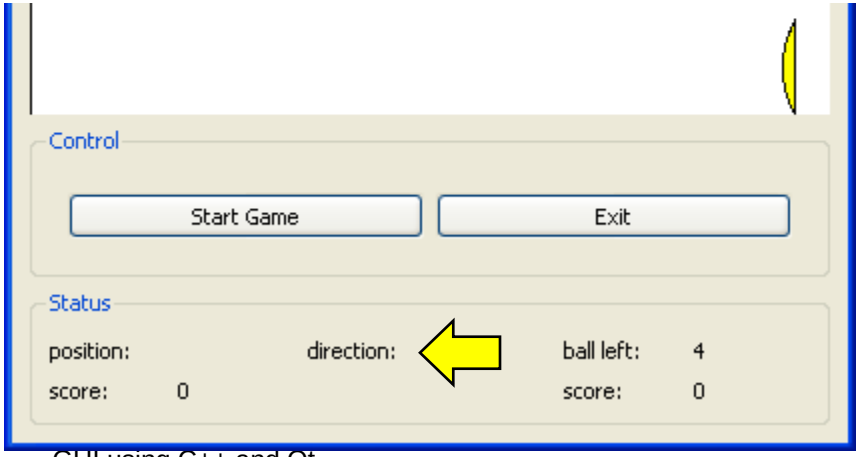

emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
}
if (newBY <= GC_ballRadius) // top wall
{
normalVector = TwoDVector(0, 1, true);
findNewVelocity(normalVector);
}
if (newBY >= (GC_fieldHeight - GC_ballRadius)) // bottom wall
{
normalVector = TwoDVector(0, -1, true);
findNewVelocity(normalVector);
}
}
void PlayField::findNewVelocity(const TwoDVector & normalVector)
{
TwoDVector newVelocity;
normalVector.reflect(pf_ballVelocity, newVelocity);
double vX = newVelocity.getX();
if ((vX > -0.01) && (vX < 0.01))
{ // almost moving vertically, add some horizontal velocity
TwoDVector horizontal(vX * 10, 0, false);
newVelocity.addVector(horizontal);
}
pf_ballVelocity = newVelocity;
}
--(Unix)-- playfield.cpp (C++)--L129--Bot


emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
#include "userinterface.h"
#include "playfield.h"
#include "constant.h"
#include <iostream>
using namespace std;
UserInterface::UserInterface(int argc, char * argv[])
{
createControl();
createStatus();
ui_centralWidget = new QWidget;
QVBoxLayout * mainLayout = new QVBoxLayout;
ui_playField = new PlayField;
ui_centralWidget -> setLayout(mainLayout);
mainLayout -> addWidget(ui_playField);
mainLayout -> addWidget(ui_controlGroup);
mainLayout -> addWidget(ui_statusGroup);
setCentralWidget(ui_centralWidget);
setWindowTitle(tr("Two-Player Breakout Game"));
ui_updateTimer = new QTimer;
ui_updateTimer -> setInterval(GC_updateDelay);
ui_updateTimer -> setSingleShot(false); // repeat the timer
connect(ui_updateTimer, SIGNAL(timeout()), this, SLOT(updatePlayField()));
}
UserInterface::~UserInterface()
{
}
void UserInterface::createControl()
--(Unix)-- userinterface.cpp (C++)--L1--Top----------
Fontifying userinterface.cpp... (regexps.....)


emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
statusLayout -> addWidget(ui_leftScoreLabel, 1, 1);
statusLayout -> addWidget(rightScore, 1, 4);
statusLayout -> addWidget(ui_rightScoreLabel, 1, 5);
ui_statusGroup = new QGroupBox(tr("Status"));
ui_statusGroup -> setLayout(statusLayout);
}
void UserInterface::startGame()
{
ui_updateTimer -> start();
}
void UserInterface::exitGame()
{
// optional: ask the user whether the game should be terminated,
// allow the user to cancel
close();
}
void UserInterface::updatePlayField()
{
ui_playField -> update();
}
--(Unix)-- userinterface.cpp (C++)--L65--Bot

Exercise Requirements

1. Make the right paddle follow the ball's vertical coordinate.
2. Detect collision between the right paddle and the ball and make the ball bounce.
3. Update the ball's position and direction.



Submission: A zip file of the CVS repository

Remember to commit all changes first.

Submit this exercise only.

Do not submit any other exercise.

Do not submit a wrong zip file.