UNIVERSITY OF LIVERPOOL

Application Development with C++ (ELEC362)

Lecture 13: Basics of Qt

mihasan@liverpool.ac.uk

Previous lecture

- The basics of software engineering were introduced and discussed.
- The waterfall model was discussed with its advantages and disadvantages.
- The requirement analysis, software design, validation and verification, and maintenance were discussed.
- The incremental model was presented and discussed.

This lecture

- What is covered in this lecture?

  1. Introduction to Qt library. 2. How to deploy a stand-alone app using Qt.

- Why it is covered?

  To understand how Qt is structured as a library and how an application can be made independent from Qt.

- How are topics covered in this lecture:

  2 live demonstrations

IMPORTANT NOTE

Material from this lecture forward are not included in the exam!!

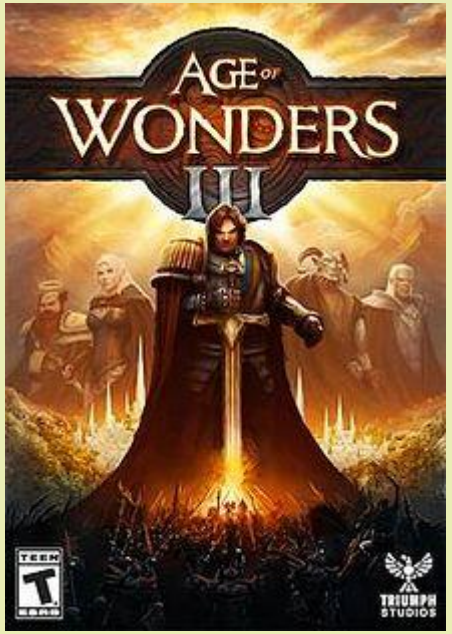
Why Qt??

- There are many C++ libraries aimed at GUI application development, including

  GTK+, Qt, MFC, and JUCE.
- Qt is one of the most widely supported GUI libraries, with a lot of online documentation, examples, tutorials, and supportive community.
- It can be integrated very well with other libraries such as OpenGL (graphics library) and Boost (math-based libraries).
- Qt is a cross-platform IDE that creates applications working on Windows, Linux, Mac-OS, Android, and many embedded systems.

Why Qt??

- Examples of known software built using Qt:



- Examples of known companies that use Qt for their software development:

Qt resources and learning outcomes

- Books: Learn Qt 5: N Sherriff (2018).

  C++ GUI Programming with Qt, J Blanchette & M Summerfield (2008).

- Online documentation: https://doc.qt.io/

  https://wiki.qt.io/Main

- Online developers community: https://forum.qt.io/

- What you will learn is:

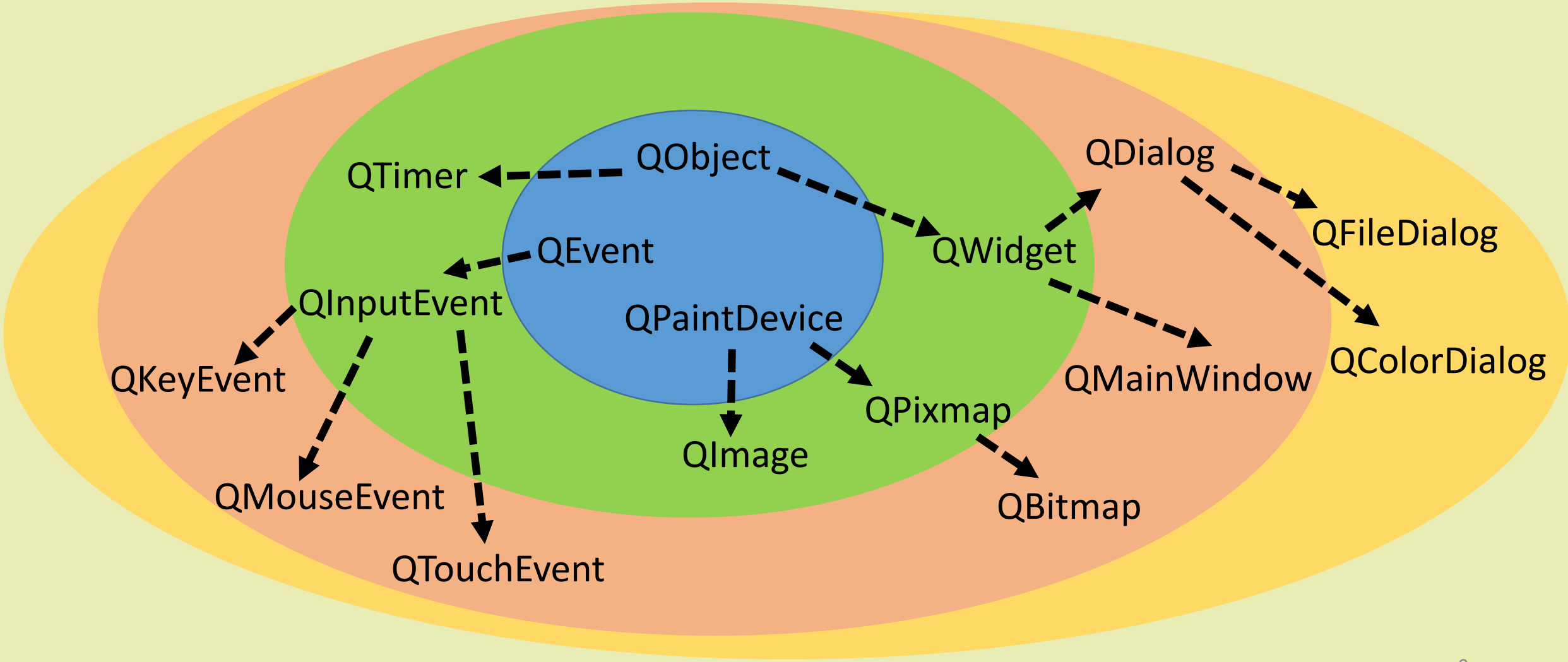
1. How to use Qt to develop Graphic User Interface (GUI)-based applications.

2. How to effectively use online documentations and resources for self-learning in programming context.

Qt basics

- Qt has been been created in 1991, since then it has been growing.
- Qt is a C++ Library used to develop GUI applications for cross-platform applications.
- It is available under a commersial license and in a open-source licensces.
- It consists of more than 2100 classes!! Each class has more than 7-10 functions!!
- Full list of classes can be found here: https://doc.qt.io/qt-5/classes.html
- The name of every class in Qt starts with a “Q”.
- The class names are carefully chosen to reflect their purpose.

Qt class hierarchy



- QUESTION: Do you notice anything about the names of the classes ??

Qt and C++ relation

- Qt was developed using C++.
- Qt follows C++ syntax rules.
- Many of the widely used classes/containers in C++ were replaced in Qt:

C++ name	Qt equivalent
`string`	QString
`vector`	QVector
`list`	QList

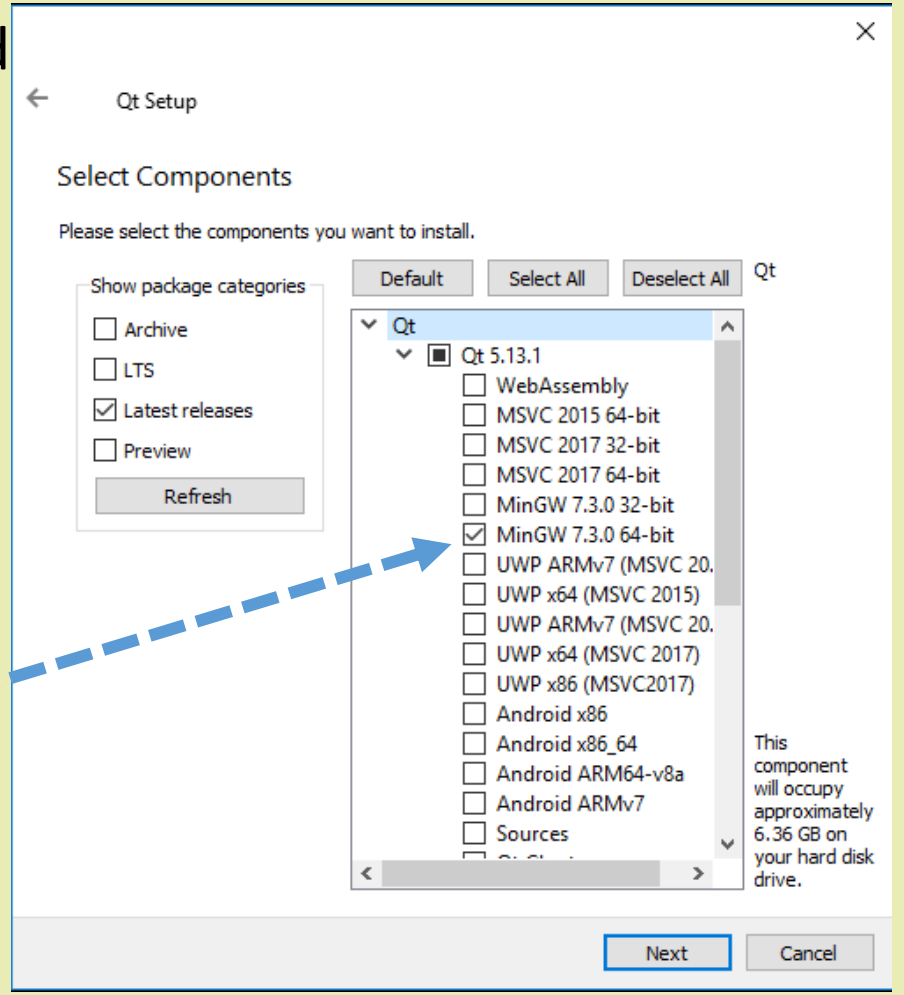
- To make the use of Qt more effecint, a new language called “QML“ was developed.

Installation of Qt

- In the labs and at the library, Qt is already pre-installed
- For downloading and installing Qt on your computer /laptop, go to:

https://www.qt.io/download-qt-installer?hsCtaTracking=99d9dd4f-5681-48d2-b096-470725510d34%7C074ddad0-fdef-4e53-8aa8-5e8a876d6ab4

Choose a package that matches the platform of the application you develop



Qt creator

- The name of IDE software for Qt is “Qt creator”.
- The name of the GUI editing part of Qt creator is known as “Qt designer”.
- A Qt creator walkthrough (demonstration).
- Considering that 2100 classes is a large number to handle, classes are categorised into “modules”, each module is particularly suited for the hardware of the platform it is developed for.
- A full list of modules is given here: https://doc.qt.io/qt-5/qtmodules.html

Deploying an application

- Qt creates the application easily within the IDE.
- The IDE generates an executable file that has dependencies. To make it a stand-alone application, the dependencies need to be put in the same folder as the exe file. Converting the application into a stand-alone app is known as “deploying”.
- For Windows application, one can use “Dependencies Walker” to determine which libraries an executable file need to run (http://www.dependencywalker.com/)
- Deploying stand-alone executable from Qt (demonstration).

Deploying an application

- The needed libraries can be found in the “bin” folder in the installation folder of the Qt compiler installed for the platform (in our case the MinGW 7.3-64).
- The required libraries also depend on the type of build:

Type of build	Libraries needed (all have .dll extension)
Debug	libgcc_s_seh-1, libstdc++-6, libwinpthread-1, Qt5Core, Qt5Gui, Qt5Widgets
Release	libgcc_s_seh-1, libstdc++-6, libwinpthread-1, Qt5Cored, Qt5Guid, Qt5Widgetsd

- Note that the list of libraries may change based on the module used, and the platform, and the architecture. Best use Dependency Walker to check.

Summary

- Justification for learning Qt was given.
- Basic structure of Qt was explained.
- An introduction to Qt Creator was given.
- Deploying an application using Qt was explained.