Image Processing & GUI Programming with Qt

Edgardo Molina
CCNY

User Interaction

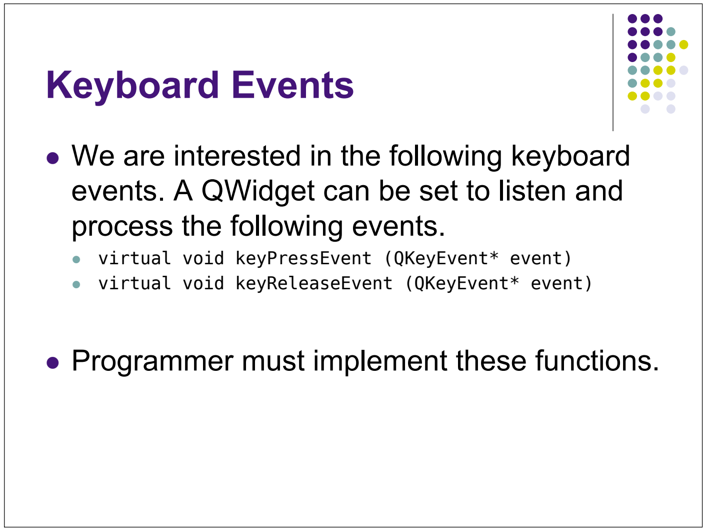
- Graphical User Interfaces allow us to interact with programs. The main input devices for interaction are:
  - Mouse
    - Mouse buttons
    - Scroll-wheel
    - Location
  - Keyboard
    - Keys
  - Tablet Pens
    - Like mouse with pressure sensor

Mouse Events

- We are interested in the following mouse and wheel events. A QWidget can be set to listen and process the following events.
  - `virtual void mouseDoubleClickEvent (QMouseEvent* event)`
  - `virtual void mouseMoveEvent (QMouseEvent* event)`
  - `virtual void mousePressEvent (QMouseEvent* event)`
  - `virtual void mouseReleaseEvent (QMouseEvent* event)`
  - `virtual void wheelEvent (QWheelEvent* event)`
- Programmer must implement these functions. The ‘`event`’ parameter contains information regarding the users input.

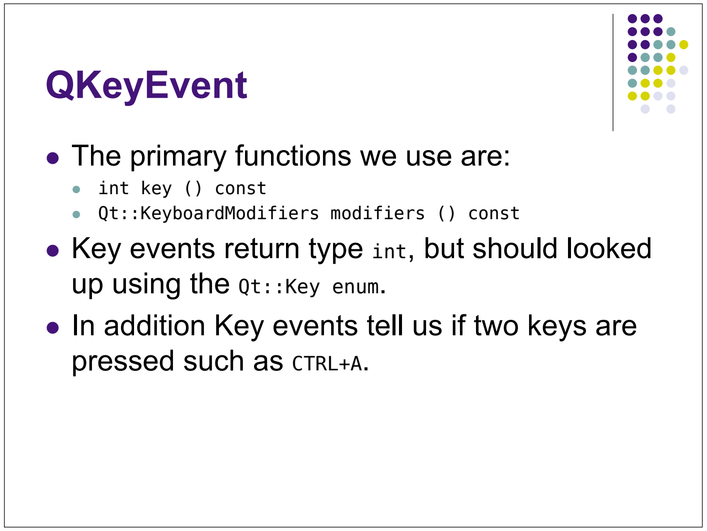
QMouseEvent

- The primary functions we use are:
  - `Qt::MouseButton button () const`
  - `int x () const`
  - `int y () const`
- We can set `QWidget::setMouseTracking()`, this will cause mouse events to occur without a user pressing a mouse button.
- Scroll-Wheels are usually found on a mouse, the functions it provides are:
  - `Qt::Orientation QWheelEvent::orientation () const`
  - `int QWheelEvent::delta () const`



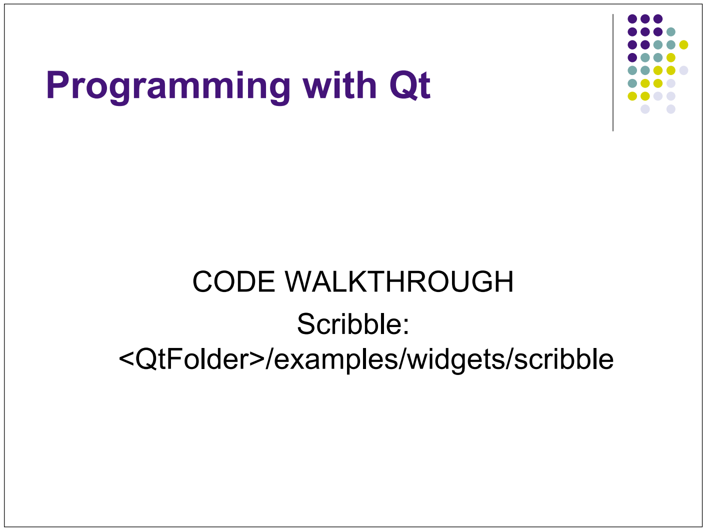
Keyboard Events

- We are interested in the following keyboard events. A QWidget can be set to listen and process the following events.
  - `virtual void keyPressEvent (QKeyEvent* event)`
  - `virtual void keyReleaseEvent (QKeyEvent* event)`
- Programmer must implement these functions.



QKeyEvent

- The primary functions we use are:
  - `int key () const`
  - `Qt::KeyboardModifiers modifiers () const`
- Key events return type `int`, but should looked up using the `Qt::Key enum`.
- In addition Key events tell us if two keys are pressed such as `CTRL+A`.



Programming with Qt

CODE WALKTHROUGH

Scribble:
<QtFolder>/examples/widgets/scribble