

Drawing Attributes

- All drawn objects have attributes
 - Color
 - Line widths
 - Fill
- These are often specified in a structure
 - Qt: QPainter Object
 - X: Graphics Context (GC)
 - Windows: Device Context (CDC)

1




Color

[see <http://doc.trolltech.com/qcolor.html>]

- Specified by electron gun settings
 - Red, Green, and Blue
 - Great for hardware, poor for the user?
- Predefined X color names (known by Qt)
 - Predefined names for standard colors
 - Look for rgb.txt
 - `dpkg --search rgb.txt`
 - `find /usr -name rgb.txt`
 - Mapped to RGB values
- QColor (Qt color class)
- Other color models are available

2



Palettes

- Systems allow lots of colors
- Images use many fewer
 - Why store so many bits/pixel?
 - When most colors are not used
- Solution
 - Only allow a subset of the colors
 - At any one time

3

Palette Example

- Typical modern system
 - 24 bits/pixel → 16 M colors
- Limited palette
 - 16 colors at a time → 4 bits/color ID

```

    graph LR
      FB[Frame Buffer] -- 4 --> P[Palette]
      P -- 24 --> EG[Electron Gun]
  
```

Often implemented in hardware

4

Palettes and Qt

- A QPalette is not a true palette
- It describes a group of related colors
 - Provide for a consistent look
- Based on a few colors
 - *e.g.*, Foreground and Background
- Similar to the Windows appearance schemes

5

Line Attributes

[see <http://doc.trolltech.com/qpen.html>]

- Width
 - How many pixels wide is a line
 - Draw rectangle or lines parallel to ideal line
- Style
 - Solid, Dashed, etc.
 - Skip pixels while drawing

6

Line Cap Styles

<http://doc.trolltech.com/qt.html#PenCapStyle-enum>

Not last		← Line endpoint
Butt		
Round		
Projecting		

Not all styles are supported by Qt.

7



Line Join Styles

<http://doc.trolltech.com/qt.html#PenJoinStyle-enum>

Bevel	Miter	Round

Qt 3 supports all 3 styles.

8




Polygon Attributes

[see <http://doc.trolltech.com/qbrush.html>]

- Filling
 - Inside-Outside Rule
 - Even-Odd
 - Non-Zero Winding Number
 - Pattern
 - Etc.

9






Modern Windowing Systems

- Program flow is **NOT** sequential
- Order of execution is up to the user
 - When...
 - Buttons pressed
 - Keys typed
 - Mouse moved over a window
 - Etc.


10



Events and Callbacks

- GUI and mouse activity is registered
 - Actions are associated with functions
 - Events – user actions
 - Callbacks – registered functions
- Each action can trigger a callback
 - Similar to interrupt service routines


11



Callback Features

- Should return in a reasonable period of time
 - Don't wait for additional user input
 - Stalls application
- How do we keep track?
(e.g., press-drag-release)
 - View as state machine
 - State information in member variables
 - Messages (callbacks from the windowing system to your objects) trigger response and change of state


12



X Window System

- Network aware windowing system
 - Distributed processes
- Client-Server model
 - E.g., Defiant code is the client
 - Laptop is the server
 - Multiple servers can be used
 - More than one "terminal"
- Qt – Runs on top of X

13



Server Software: XFree86

- XFree86 is X Server software for *nix
 - xterm – a terminal window that runs under X
 - ssh – secure login to and forwarding of X traffic between local machine and defiant
 - TCP/IP session (address, port, ...) between client and server
 - Client trades messages with server
- Efficient use of network
 - Data structures stored by client and server
 - Pens, colors, etc.
- Device independent graphics!

14
