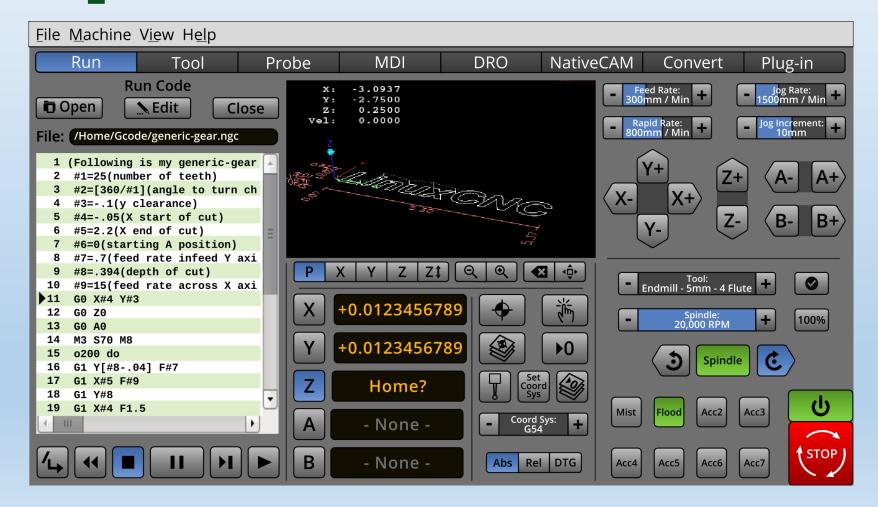
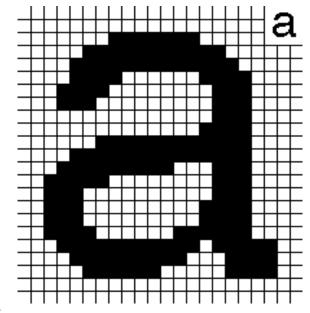
Graphical User Interface



Chapter 20

Early Monitors: Text based

- CPU writes a string of characters to the monitor
- Monitor is like a teletype:
 - New characters are written in order on the bottom line of the screen
 - When the bottom line fills up, scroll the screen up one line



- Each character is converted to pixels by the font library built in to the monitor
 - Each character is a rectangular matrix of pixels

Modern Graphics Basics

- Software controls the entire screen
- Your screen is a two dimensional array of pixels
- Think of each pixel as having an (x,y) coordinate
- The upper left hand side of the screen is (0,0)
- The X axis increases from left to right
- The Y axis increases from top to bottom

(0,0)(1920,0)0 eclipse - Java - cs140/src/SwingTutorial/TopLevelDemo.iava - Eclipse File Edit Source Refactor Navigate Search Project Run Window Help [a] [v] [t⟩ ▼ [v] ▼ [v] ▼ [w] Ø ▼ [w] Ø ▼ [v] Ø [v] Ø [v] [v] Ø [v] [v] [v] Ø [v] [v] Ø [Quick Access 🔡 📸 🐉 🎋 ■ TopLevelDemo.java Package Explorer ⋈ //Set the menu bar and add the label to the content pane. **☆** ▼ | % % √ 🔑 cs140 67 frame.setJMenuBar(greenMenuBar); ✓

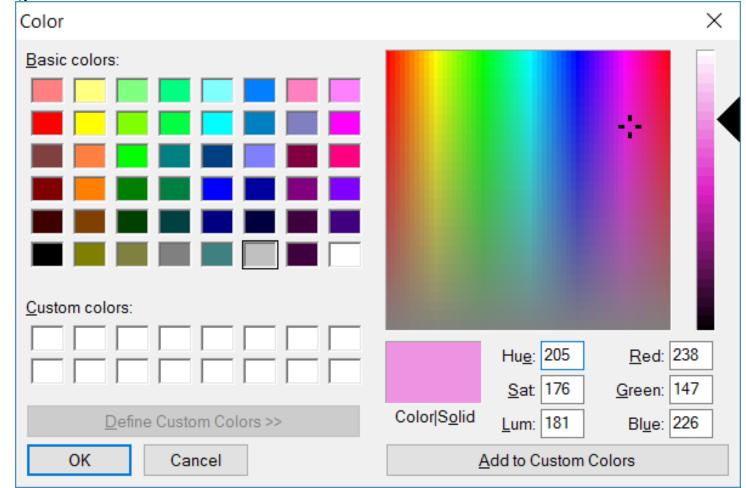
Æ src Find Q ▶ All frame.getContentPane().add(yellowLabel, BorderLayout.CENTER); > # assignment06 69 > # assignment06 sol Cygwin 70 //Display the window. > # assignment07 71 > # assignment08 frame.pack(); > # assignment08_sol 72 frame.setVisible(true); > # assignment08_template 73 ⊕ CelsiusConverter **2** 0 ⊠ □ □ 74 WinSCP > # lab08 5º 🖹 ↓ª 💆 🔊 8 75⊜ public static void main(String[] args) { > # lab09 //Schedule a job for the event-dispatching thread: 76 > # lab09 toolong # SwingTut 77 //creating and showing this application's GUI. > # lab10 ✓ O_▶ TopLevel →

■ SwingTutorial javax.swing.SwingUtilities.invokeLater(new Runnable() { 78 s creat > HelloWorldSwing.java MP Naviga △79⊝ public void run() { > 1 TopLevelDemo.java createAndShowGUI(); > **Q** r 80 > # xmp_constructor_inheritanc > # xmp_question_base > 🔠 xmp_question_constructor 🦹 Problems @ Javadoc 🖳 Declaration 📮 Console 🛭 > # xmp_question_inheritance <terminated> TopLevelDemo [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Nov 10, 2016, 1:41:28 PM) > # xmp_question_noInheritanc > # xmp_recurse ■ JRE System Library [JavaSE-1.8] students.txt Smart Insert 47:20 (0,1080)

Single Pixel

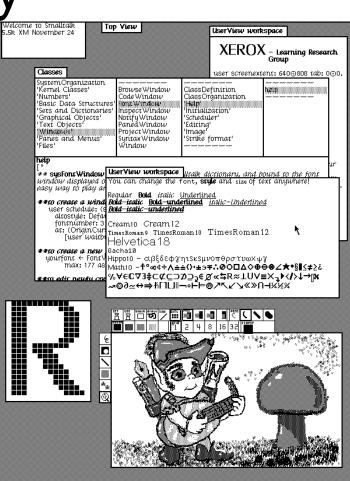
Each pixel is described by three numbers 0-255

- Amount of Red
- Amount of Green
- Amount of Blue
- Higher numbers are brighter
 - Black is 0,0,0
 - White is 255,255,255



Graphical User Interface History

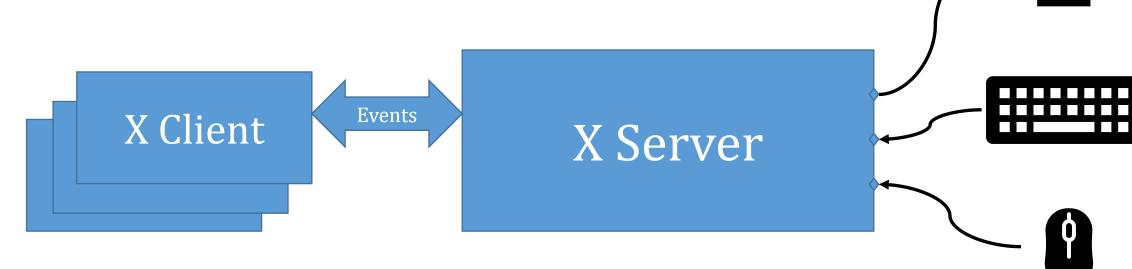
- Introduced in the 1960s
- Refined and implemented in early 1970 by Xerox
 - Palo Alto Research Center (PARC)
- "WIMP" paradigm
 - Windows, Icons, Menus, Pointers
- Xerox not interested
 - Demo to Steve Jobs and Bill Gates among others
 - Apple Lisa (1983)
 - Microsoft Windows (1985)



X Window System (X11)



- A platform independent layer of open-source software used to manage monitors
- Developed at MIT in 1984, version 11 (X11) 1987
- Maintained by the "X Org Foundation"



X Client Startup

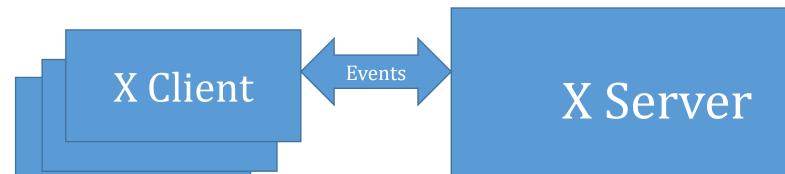


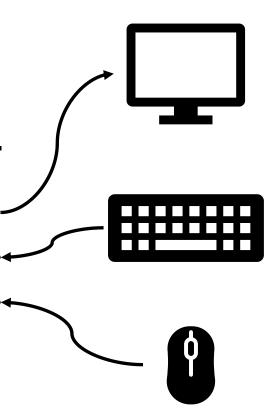
- X Client asks for a window on the screen
 - Defines what "widgets" go on that window
 - Provides functions to "draw" the window
 - Provides an "event loop" to handle "events" on that window **Events** X Client X Server

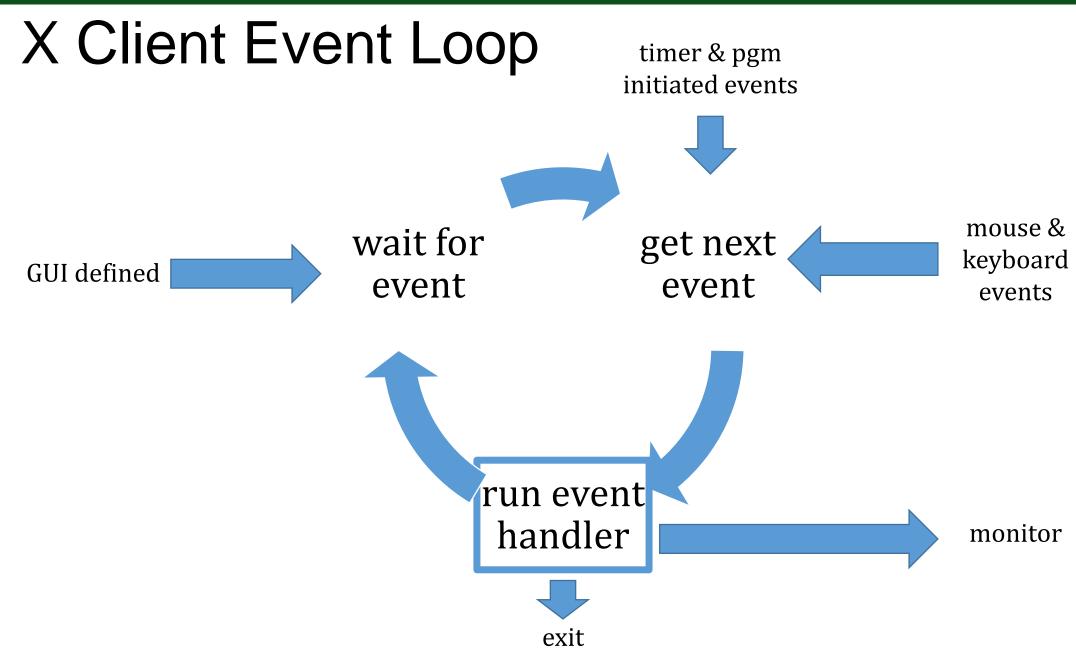
X Server



- Manages all windows for all clients
- Keeps track of an "active" window
 - The one the user is currently interacting with
 - Sends events from active window to appropriate client
- Manages window overlap / movement / minimize, etc.







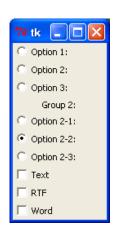
First Class Functions and Callbacks

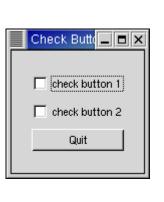
- X client "registers" functions to get invoked when specific events occur
 - "Call me when this button is pushed" (callback)
- Much easier using functions as parameters
 - Lambda functions simplified GUI processing SIGNIFICANTLY!



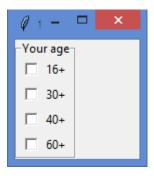
Java GUI Portability Problem

- Java loves the idea that your code produces the same results on any machine
- The underlying hardware and software of different machines has different GUI capabilities
- Different operating systems implement the same GUI concepts differently









Abstract Window Toolkit (AWT)

- First released in 1995 with Sun's first Java release
- Thin abstract layer between the Java view of a user interface and the underlying hardware/software
- For instance AWT checkbox:
 - maps to Windows checkbox on a Windows machine,
 - maps to X checkbox on Unix machine
- Simple, but not hardware independent
- Based on X11 style of programming (event loop)

Swing

- Built on top of AWT
- Coded in 1997 (Released Java 1.2)
- Name derived from code-name because Sun programmers liked swing music
- Deeper interface... less dependent on machine / operating system
 - More complicated (18 packages, 737 classes as of JDK 8)

JavaFX

- Built on top of Swing
- Makes the creation of a GUI "simpler"
- Requires a GUI building tool
 - 3rd party tool with different release levels, interactions with Eclipse, etc.
 - Hides the details of the GUI from the coder
 - Swing/Java code generator
- "Removes" capability
 - Simple Swing only, to simplify code generator
 - Override the GUI builder to get that capability back... if you know Swing
- No longer packages with JDK
- Use if you want, but we teach Swing