Is image everything?

Ceci n'est pas une pipe
Review

Computer Graphics technology enables GUIs and computer gaming.

GUI's are a fundamental enabling computer technology.

Without a GUI there would not be any, or much less:

- Computer based industry -- USA's major productivity advantage
- Computer systems in the homes (1997 18% → 2012 75%) with broadband
- Internet, Web, E-commerce
- before Mosaic -- telnet, ftp, gopher, lynx, WAIS (text UI)
- after Mosaic – Google, Facebook, Amazon.com, Ebay
- smart phones, tablets

How much software would you use / buy that does not have a GUI interface?

How many computer games without graphics would you play?

Where would Microsoft be without Windows? Apple w/o Macintosh?
GUI Concepts

Architecture

MVC  UIs often model / view-controller (touch changing ?)
Geometry  parent (container) / child (component)
            content hierarchy sets layout constraints

Interface Components  window, menu, dialog, label, button, scrollbar,
                      list, combobox, toolbar, table, pane, image, icon, font, ....

Events

  event driven program architecture "control flow"
  internal events: Timers, "event messages"
  user events are associated w/ UI component
    implicitly events (resize, expose)
    explicitly events: user clicks, selects, drags on UI component

Drawing

  automatic (UI components), borders, fore / back, focus, hover
  custom (application) draws: buttons, menus, cell renderers
Swing / Java

Containers: JFrame, JApplet, JDialog, JOptionPane, JColorChooser, JFileChooser, ...
JMenuBar, JMenu, JMenuItem, ...
ContentPane, JPanel, JScrollPane, JTabbedPane, JToolBar, JTree, JList, JTable,
JButton, JTextField, JTextArea, JEditorPane, ...

Geometry: Layout Managers: Border, Box, GridBag, Flow, Grid, ...
Containment hierarchy ← nested layout mgrs

Graphics: repaint & paintComponent
invoke with repaint();
Graphics class, g.draw*(...), g.fill*(...), Color, Font, images, icons
Event Handling

Events are classes: Action, Item, Adjustment, Window, Mouse, Timer.
Actions enable menuItems, buttons to share event handlers and labels.

Listeners & Adapters (for each listener w/ 2+ methods)
Action, Item, Mouse, Window, ...
MouseAdapter, WindowAdapter

Add a listener to component
```
abutton.addActionListener(new JButtonListener());
```

Implement the listener to define event handling
```
class JButtonListener implements ActionListener {
    public void actionPerformed(ActionEvent event) {
        // define behavior
    }
}
```

Use an anonymous inner class – adapter (singleton)
```
aJComponent.addActionListener ( new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        ...
    }
});
```
Data Models

Model/ViewController architecture in collection (selection and view oriented) UIs:

- JScrollPane, JList, JComboBox, JTable, JTree, JEditorPane, JTextPane.

Selection model: single, interval, multiple

Models are objects with collections: Object[], Vector
abstract models, default models extend abstract

JList(E[] listData)  // has static list items.
DefaultTableModel(Object[][] data, Object[] columnNames)
DefaultTableModel(Vector data, Vector columnNames)

DataModels provide get / set items, get / set model's data (Vectors), ...
DataModels facilitate persistence (local and remote file or database)
Collection UIs (JTable) provide default and custom cell editor and renders.
**Windows Presentation Foundation / C#**

C# class, partial class, interface, property, delegate, enum

XAML UI designer (Blend), compiled // see obj/debug/*.*.cs
"code behind" define behavior for an event

WPF Window holds a single element.
Layout containers are nested and can hold many elements
   Layouts derive from abstract System.Windows.Controls.Panel
   Layouts don’t scroll; use with a ScrollViewer

Properties:
FrameworkElement: Margin, Alignment, Width, Height, ...
Control: Fore / Background, Brush, Border, Cursor, ...
Panel: Children, ...
DockPanel: Dock {Top, Bottom, Left, Right }, LastChildFill, ..
Event Handling

5 Event categories: lifetime, mouse, keyboard, stylus, multiTouch

Bubbling events propagate up containment hierarchy from source
Tunneling events propagate down containment hierarchy to source

In XAML set event property for a Control

```xml
<Window x:Name ...
  ...
  <Button x:Name="aButton" ... Click="buttonPressed" ... />
  ...
</Window>
```

In code behind *.cs partial class define behavior to handle event

```csharp
private void buttonPressed(object sender, RoutedEventArgs e)
  ...
  // code for method's behavior
}
```

All event handlers have save argument (signature)
  Object the source of the event
  RoutedEventArgs args describing the event
Collection UIs

WPF collection UIs: ListView, GridView, TreeView, DataGrid

DataGrid provides built-in and custom (template) cell editor/renders

ADO.net provides database persistence

DataSet is an in-memory cache of data with properties:
Tables (collection of DataTable)
Relations (collection of DataRelation) properties.

navigation from one DataTable to another within a DataSet.
Design

HCI  human performance – humans are productive learners
   visual perception and memory are superior
   Error: intention (mistake), storage (lapse), action (slip)

GUIs  Functional > Aesthetic  Simplicity > realism
   usage is a visual search – pattern recognition task
   low syntactic demand (applications share "look && feel" syntax)
      images have semiotic "iconic representation"
      images / labels can embedded semantics into tasks.
   control and closure – UI objects are manipulated in "reversible" steps
      minimizes slip errors – application state show w/ enable / disable

graphic art principles:  reduction, regulation, scale, contrast, color use,
                        activity / visual search / spatial logic, grouping – distinction,
                        figure / ground – negative space, grids,

Interface Builders:  directional (?), layout ! design tool,
                    bind UI && events
GUI trends

2D Variable magnification of display
Allow search or large information space a low magnification see more
Select objects for detailed evaluation and zoom in increase magnification
  e.g. fisheye menus, table lens, hyperbolic trees

small GUIs  cell phones, wearable computers, smart jewelry
  serial-visual-presentation (word, character)

large GUIs  3D Navigation incorporates zoom.
  Volume > Area, depth == variable magnification
  e.g. data wall, cone trees, 3D desktops, immersive GUIs

Hieroglyphic language  emot-icons, text abbreviations (2mi, h2cus, ...)
  pictures (nouns, rebus), words (verbs), emoticons (emotions)

Scientific Visualization

  Big Information ↔ Big Data (sensor, databases: social media, ...)
What to do next wrt GUI?

Be more visually literate – visual thinking is a critical skill for graphical user interface design! **Graphic Design principles are important!**

Always evaluate and appreciate good (and bad) design in the world around you. Design ideas can come from other media.

Do not limit design to what you currently know how to do! Learn more about GUI APIs (design, controls, user interface, designers) how are events implemented, can they be combined (Actions), image processing, custom drawn components, data binding (**data models**), persistence, ...

One design means:
1. You can't think of another one.
2. If plan A doesn't work you are stuck
3. You will implement your first (not best) solution to the problem.

Design as an exercise ... Design "reps" strengthens your creativity ...