Information Visualization

Jing Yang

Spring 2010

1

InfoVis Programming

2

Outline

- Look at increasing higher-level tools
  - 2D graphics API
  - Graphicial User Interface (GUI) toolkits
  - Visualization framework
- You decide which tool to use

3

2D Graphics APIs

- OpenGL
  - We will have a class on this Thursday
- Java2D, GDI+(win32), Quartz(MacOS X)
  - Platform specific 2D graphics APIs
- Processing
  - Aims to get non-programmers started with programming, through the instant gratification of visual feedback
  - Built for electronic arts and visual design communities

4

Graphical User Interface Toolkits

- Built on top of a 2D graphics library
- Major additions:
  - Input event processing and handling
    - Typically mouse and keyboard events
  - Encapsulation and organization of widgets
  - Bounds management
    - Only redraw areas in need of updating

5

GUI Toolkits

- Examples:
  - Visual C++, .NET, Visual Basic, Delphi
  - Java Swing, Tcl/Tk, QT, GT/K, wxWidgets
- Many GUI toolkits are supplied as frameworks
  - Framework – half-baked bread
  - Large chunks of the control flow are hidden inside the canned machinery of the framework and are invisible

6

Terminology

- Window
  - An area of the screen controlled by an application.
  - Is usually rectangular
  - Can contain other windows

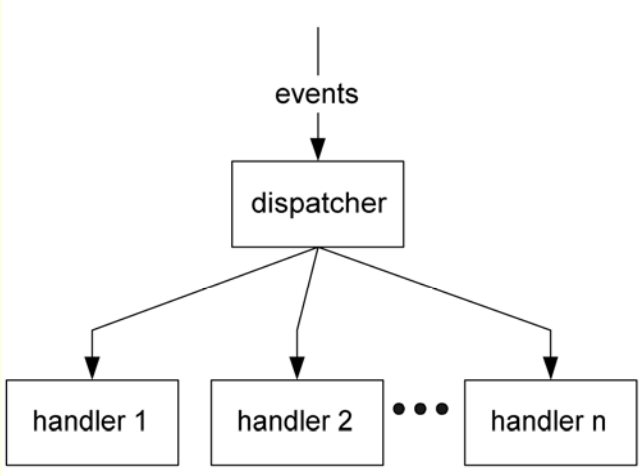
7

Terminology

- Control
  - Is a GUI object used for controlling the application
  - Has properties and usually generates events.
  - Corresponds to application level objects
  - Events are coupled to methods of the corresponding object

8

Event-Driven Programming



9

Terminology

- Widget:
  - Visible control that can be manipulated by user or programmer

- Frame
- Label
- Button
- Text Entry
- Message boxes Frame
- Label
- Button
- Text Entry
- Message boxes
- Text box

- Radio Button
- Canvas
- Check button
- Image
- listbox - for lists!
- Menu/Menu Button Scale/Scrollbar

10

Terminology

- Frame
  - A type of widget used to group other widgets together.
  - Often a Frame is used to represent the complete window and further frames are embedded within it.

11

Terminology

- Layout
  - Controls are laid out within a Frame according to a particular form of Layout
  - The Layout may be specified in a number of ways, either using on-screen coordinates specified in pixels, using relative position to other components (left, top etc) or using a grid or table arrangement

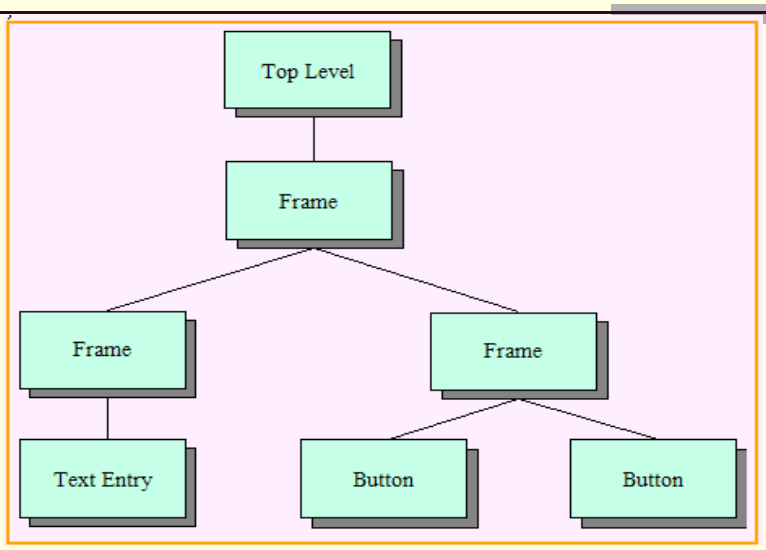
12

Terminology

- Child
  - GUI applications tend to consist of a hierarchy of widgets/controls.
  - The top level Frame comprising the application window will contain sub frames which in turn contain still more frames or controls.
  - These controls can be visualized as a tree structure with each control having a single parent and a number of children.

13

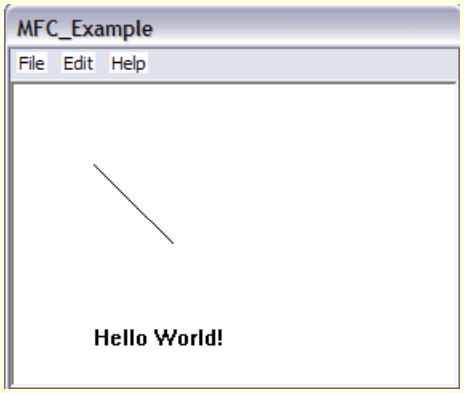
The Containment tree



14

Example 1

- Visual C++ 2005



15

Example 2

- wxWidget
- Example project: wxVisTool

16

wxVisTool Overview

- GUI (Graphical User Interface) platform: wxWidgets
- IDE: VC 2005 (you can also use other IDEs that support C++)
- Graphics: OpenGL

17

What is wxWidgets?

- A C++ framework providing GUI and other facilities on more than one platforms
- Version 2 currently supports all desktop versions of MS Windows, Unix with GTK+, Unix with Motif, and MacOS
- Free, open source
- Powerful

18

To begin

- To set a wxWidgets application going, you will need to derive a wxApp class and override wxApp::OnInit.
- An application must have a top-level wxFrame or wxDialog window. Each frame may contain one or more instances of classes such as wxPanel, wxSplitterWindow or other windows and controls.
- Example: MainApplication

19

Program Entrance

```
bool MainApplication::OnInit()
{
    MyFrame* frame = new MyFrame("Visualization
    Tool Demo", 50, 50, 450, 300);
    frame->Show(TRUE);

    return TRUE;
}
```

- MainApplication is derived from wxApp
- An application must have a top-level wxFrame or wxDialog window

20

MyFrame and wxFrame

- MyFrame is derived from wxFrame
  - Each frame may contain one or more instances of classes such as wxPanel, wxSplitterWindow or other windows and controls.
  - A frame can have a wxMenuBar, a wxToolBar, a status line, and a wxIcon for when the frame is iconized.
- MyFrame contains a canvas where you can draw using OpenGL

21

GLCanvas and wxGLCanvas

- GLCanvas is derived from wxGLCanvas
- wxGLCanvas is a class for displaying OpenGL graphics

22

Event Table

- Event table
  - Header file:
    - Declare the handler
    - DECLARE_EVENT_TABLE()
  - Source file:
    - Declare an event (such as a menu click or a mouse click) in the event table, associate it with its handler
    - Define the handler

23

Dialogs

- Instances of wxDialog can also be used for controls and they have the advantage of not requiring a separate frame.
- Instead of creating a dialog box and populating it with items, it is possible to choose one of the convenient common dialog classes, such as wxFileDialog.

24

SceneGraph-Based Approaches

- Commonly used in 3D toolkits, also used in 2D
- Models visual elements, properties, and groups in a semantic directed acyclic graph
- Groups specified related to their own coordinate systems
- Can include object grouping, multiple cameras
- Well suited for panning and zooming

25

SceneGraph-Based Toolkits

Adobe Flash

- A multimedia platform that is popular for adding animation and interactivity to web pages.
- Commonly used to create animation, advertisements, and various web page Flash components, to integrate video into web pages, and more recently, to develop rich Internet applications.

26

SceneGraph-Based Toolkits

Piccolo

- http://www.cs.umd.edu/hcil/jazz/
- Support Zoomable User Interface
- Three versions:
  - Piccolo.Java
    - Built on Java 2 and rely on Jave2D API
  - Piccolo.NET
    - Built on the .NET Framework and relies on the GDI+ API
  - PocketPiccolo.NET (for the .NET Compact Framework)

27

InfoVis Toolkits

- Most GUI toolkits provide unified structures for graphics and interaction
- InfoVis frameworks must also consider:
  - Data modeling and manipulation
  - Mappings from data to visuals
- Higher-level constructs also possible
  - Layout techniques
  - Interactive techniques
  - Visual transformations

28

InfoVis Toolkits

- Jean-Daniel Fekete's InfoVis Toolkit
- Katy Borner's InfoVis CyberInfrastructure
- UC Berkeley's User Interface Research Group's Prefuse , ProtoVis
- University of Maryland's Piccolo Toolkit
- Penn State Departement of Geography's GeoVISTA Studio

29

For network visualization or graph layout

- UC Irvine's Java Universal Network/Graph Framework (JUNG)
- Dimitris Kalamaras's Social Network Visualizer
- AT&T's GraphViz
- University of Ljubljana's Pajek
- David Auber's Tulip

30

InfoVis Toolkit [Fekete]

- http://ivtk.sourceforge.net/
- A Java library and software architecture relying on the Swing GUI
- Extensible collection of infovis “widgets”
  - Scatterplot, treemaps, graph visualizations, etc
- General interactive components
  - Dynamic queries, distortion lenses, excentric labels

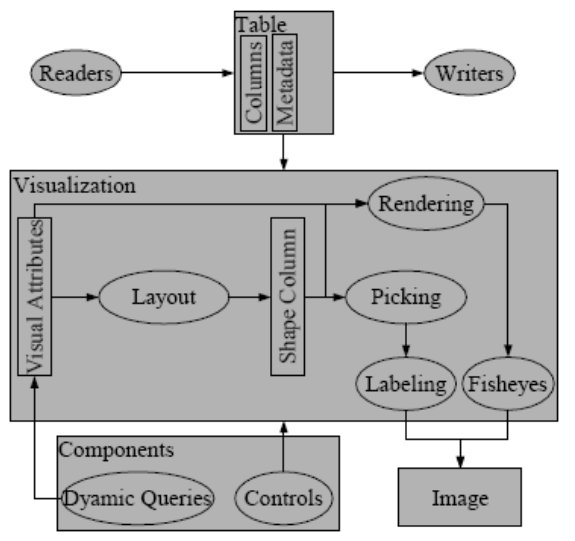
31

Data Model

- Table-based data model, similar to database
  - A table is a list of named columns plus metadata and user data
  - To represent a tree: add “parent”, “first child”, “next sibling “ columns and other columns on demand

32

Internal Structure



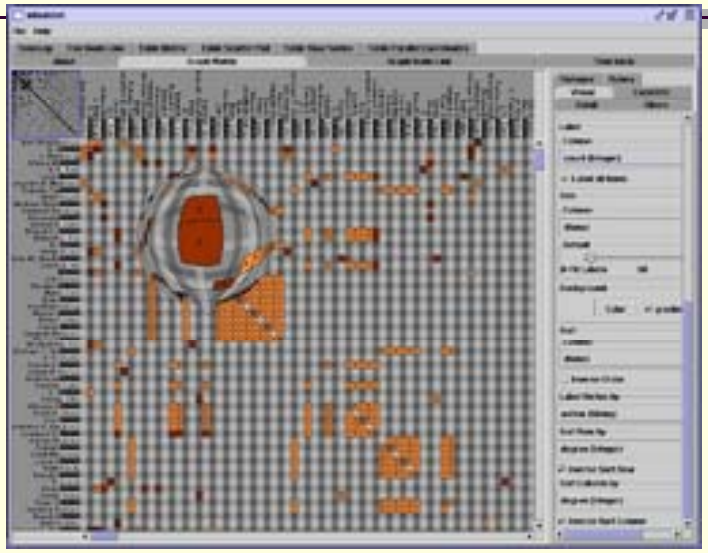
33

Sample Code

```
public class Example1 {
    public static void main(String args[]) {
        String fileName =
            (args.length == 0) ? "data/salivary.tqd" : args[0];
        DefaultTable t = new DefaultTable();
        t.setName(fileName);
        AbstractReader reader =
            TableReaderFactory.createReader(fileName, t);
        if (reader == null || !reader.load()) {
            System.err.println("cannot load " + fileName);
            return;
        }
        TimeSeriesVisualization visualization =
            new TimeSeriesVisualization(t);
        VisualizationPanel panel =
            new VisualizationPanel(visualization);

        JFrame frame = new JFrame(fileName);
        frame.getContentPane().add(panel);
        frame.setVisible(true);
        frame.pack();
    }
}
```

Example



35

Prefuse [Heer et al]

- http://prefuse.org/
- Prefuse toolkit provides a visualization framework for the Java programming language using the Java2D graphics library.
- Prefuse flare toolkit provides visualization and animation tools for ActionScript and the Adobe Flash Player.
- Supports node-link diagrams, containment diagrams, collections, scatterplots, timelines

36

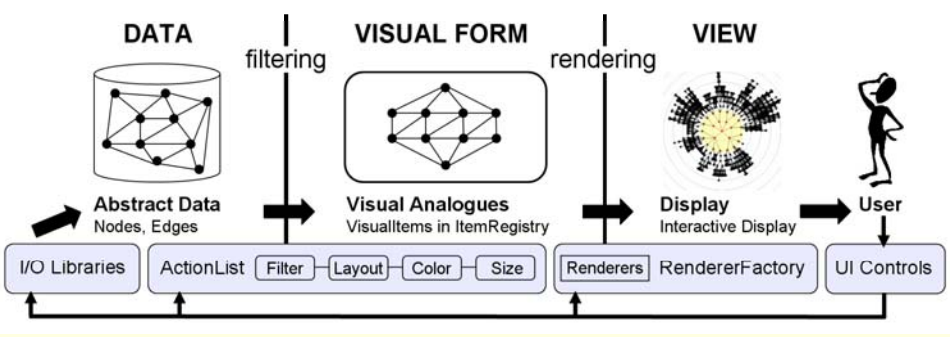
Features

- Data structures and I/O libraries
- Multiple visualizations, multiple views
- Application design through composable modules
- A library of provided layout and distortion techniques
- Animation and time-based processing
- Graphics transforms, including panning and zooming
- A full force simulator for physics-based interfaces
- Interactor components for common interactions
- Integrated color maps and search functionality
- Event logging to support visualization evaluation

- Demonstration video

37

System Architecture



38

Table-Based Data Representation

- Use edge tables and node tables to represent graphs and trees
- Tables can be indexed and queried
- Visual items

39

ProtoVis [Bostock et al]

A graphical toolkit designed for visualization

- Led by Mike Bostock and Jeff Heer of the Stanford Visualization Group.
- Free and open-source, provided under the BSD License.
- Uses JavaScript and SVG for web-native visualizations; no plugin required
  - JavaScript is an object-oriented scripting language; SVG is a language for describing two-dimensional graphics and graphical applications in XML.

40

Features

- Composes custom views of data with simple marks such as bars and dots.
- Defines marks through dynamic properties that encode data, allowing inheritance, scales and layouts to simplify construction.
- Designed to be learned by example.

41

Reference

- Jeffrey Heer: Software Architectures
  - http://vis.berkeley.edu/courses/cs294-10-sp06/WWW/lectures-WWW/frameworks/

42