How To Build a Sketch Based Interface

Tevfik Metin Sezgin, Michael Oltmans, Randall Davis

Consider This Device...
Our Vision

Our Model

- The designer sketches with pen and paper
- The observer interprets the sketch
- The observer and designer interact
Demo

- Conceptual mechanical design
- Client
  - Low level sketch understanding
  - Recognize sketch as mechanical device
- Server
  - Simulate the recognized device

How did we get here?

- Low level sketch processing
- Domain level recognition
- Connection to existing design tools
Today’s goals

• Learn to use the low level recognition toolkit
• Learn how to build a simple sketching interface with the toolkit
• Build your own interface to Xfig

And now, on to the Toolkit...

Toolkit functionality

Given a freehand stroke, generate a geometric primitive

• Lines
  ![Line Example]

• Circles
  ![Circle Example]

• Curves
  ![Curve Example]
The Toolkit doesn’t do…

- Higher level recognition (i.e., can’t recognize squares, rectangles, domain specific shapes)

- Gesture recognition

Terminology

Sketch: Informal, messy diagrams consisting of several strokes

Stroke: Array of timestamped points collected between pen down and pen up events
Structure of an application

SketchPanel

StrokeData

StrokeDataListener

SimpleClassifier

Application

Geometric Object

Package hierarchy

- edu.mit.sketch.fig (SketchFig.java)
- edu.mit.sketch.toolkit
  - Recognition related classes are here
- edu.mit.sketch.geom
  - The geometry package
- edu.mit.sketch.ui
  - User interface related classes
### Classes you will use most often

**edu.mit.sketch.ui**
- SketchPanel

**edu.mit.sketch.toolkit**
- StrokeData
- SimpleClassifier

**edu.mit.sketch.geom**
- GeneralPath
- Polygon
- Line
- Ellipse
- Point
- Rectangle

---

**edu.mit.sketch.ui**

- SketchPanel
  - extends javax.swing.JPanel
  - Gathers stroke data
  - Displays raw strokes as they are drawn
  - Has methods for adding and removing StrokeDataListeners
edu.mit.sketch.toolkit.StrokeData

- This class holds and computes stroke related information such as points in the stroke, pen speed, curvature...
- The constructor takes an array of points
- SketchPanel creates this object after each mouse up event

edu.mit.sketch.toolkit.SimpleClassifier

- Constructor takes a StrokeData object
- Has a method int classify()
- This method returns an int indicating the type of the approximation generated by the toolkit
- Compare the result against the following using a switch statement
  - SimpleClassifier.LINE
  - SimpleClassifier.ELLIPSE
  - SimpleClassifier.POLYGON
  - SimpleClassifier.COMPLEX
edu.mit.sketch.toolkit.SimpleClassifier

- One can also check for a particular type by
  - SimpleClassifier.isLine()
  - SimpleClassifier.isEllipse()
  - SimpleClassifier.isPolygon()
  - SimpleClassifier.isComplex()

edu.mit.sketch.toolkit.SimpleClassifier

- Once the type is determined, the approximation can be accessed by
  - Line getLineApproximation()
  - Ellipse getEllipseApproximation()
  - Polygon getPolygonApproximation()
  - GeneralPath getComplexApproximation()
public class XFigFrontend
extends SketchPanel
implements StrokeDataListener {

    public XFigFrontend () {
        addStrokeDataListener( this );
    }

    public void handleStroke( StrokeData stroke_data ) {
        SimpleClassifier classifier = new SimpleClassifier( stroke_data );
        switch( classifier.classify() ) {
            case SimpleClassifier.LINE:
                ...
            ...
        }
    }
}

How to get started

Compile and run
- “make classpath” – only need to do this once for each shell window
- “make” – compiles the java files
- java SketchTest

See how it uses the toolkit and the SimpleClassifier

Best strategy for understanding the control flow: find the handleStroke() method in TicTacToe.java
How to get started

• Compile and run
  – javac SketchTest.java
    • Java is not Python! **ALWAYS** recompile after making changes
    – <correct any errors!>
    – java SketchTest
      • don’t include the “.class” part of the name when running

• Try compiling and running MyTTT (assuming I can find it)

• See how it uses the toolkit and the SimpleClassifier

• Best strategy for understanding the control flow: find the handleStroke() method in TicTacToe.java

Resources

• Javadoc documentation is included with toolkit
  – mozilla ~/assist/doc/index.html

• Questions: contact Metin Sezgin
  mtsezgin@ai.mit.edu