Introduction to yFiles

www.despair.com

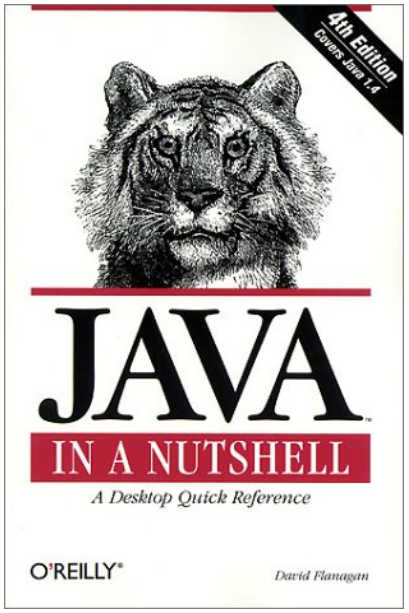
Problems: No matter how great and destructive your problems may seem now, remember, you've probably only seen the tip of them.

Overview

- yFiles
- Basic data structure
- Breadth First Search
- Simple layout algorithm
- cvs

Tools

Books: Java in a Nutshell



IDE: Eclipse

yFiles

- Basic: graph, geometry, ...
- Layout: abstract layout model, implementation
- View: graphical representation, I/O, ...
- Extensions: GraphML

Libraries: `y.jar`, `graphml.jar` and `crimson.jar`

Simple Example – Linear List

List member functions:

- add element
- remove element
- get iterator

⇝ API YList

Iterator functions:

- get current element
- get next element
- is valid position

⇝ API YCursor

Example – Breadth First Search

Input : graph $G = (V, E)$, root node $s \in V$

Data : queue Q (BFS front)
markers for nodes and edges

```
mark s;
append Q ← s;
while Q not empty do
    pop v ← Q;
    foreach e = (v, w) ∈ E not marked do
        mark e;
        if w is not marked then
            mark w;
            append Q ← w;
```

Datastructures

- graph $\longleftrightarrow$ Graph
- queue $\longleftrightarrow$ YList
- marker $\longleftrightarrow$ Node/Edge-Map

```
Graph graph;
Node startNode;

NodeList queue;

NodeMap isVisited;
EdgeMap isTraversed;
```

BFS — Constructor

```
public BFS( Graph argG , Node args ){
  graph = argG;
  startNode = args;
  queue = new NodeList();
  isVisited = graph.createNodeMap();
  isTraversed = graph.createEdgeMap();

  for(NodeCursor V = G.nodes(); V.ok(); V.next()){
    isVisited.setBool(V.node(), false);
  }
  for(EdgeCursor E = G.edges(); E.ok(); E.next()){
    isTraversed.setBool(E.edge(), false);
  }
}
```

BFS — Simple Auxiliary Functions

```
void markNode(Node v){
  isVisited.setBoolean(v,true);
}
boolean isMarked(Node v){
  return isVisited.getBoolean(v);
}

...

void init(){
  markNode(startNode);
  queue.add(startNode);
}
```

BFS — Iteration

```
void iteration(){
  while( ! queue.isEmpty() ){
    Node v = queue.popNode();
    for(EdgeCursor N = v.edges(); N.ok(); N.next()){
      Edge e = N.edge();
      if( ! isMarked(e) ){
        markEdge(e);
        Node w = e.opposite(v);
        if( ! isMarked(w) ){
          markNode(w);
          queue.add(w);
        }
      }
    }
  }
}
```

BFS — Algorithm

```
public void run(){
  init();
  iteration();
}

public void run(Node otherStartNode){
  startNode = otherStartNode;
  init();
  iteration();
}
```

⇝ API: Graph Node NodeMap

Layouts

Restricted scenario:

- assign coordinates to every node
- assign coordinates to every edge (polylines)

$\rightsquigarrow$ API: CanonicMultiStageLayouter

Important Methods of CanonicMultiStageLayouter

- canLayoutCore
- doLayoutCore

```
protected boolean canLayoutCore(LayoutGraph arg0) {
  // test requirements
  return true;
}
```

Simple Layout – Diagonal Placement

```
protected void doLayoutCore(LayoutGraph G) {

  double offset = 0.0;
  for(NodeCursor V = G.nodes() ; V.ok() ; V.next()){
    Node v = V.node();
    G.setLocation(v,offset,offset);
    offset += 10.0;
    offset += Math.max(G.getWidth(v),G.getHeight(v));
  }
```

Simple Layout – Diagonal Placement (2)

```
    for(EdgeCursor E = G.edges() ; E.ok() ; E.next()){
      Edge e = E.edge();
      EdgeLayout el = G.getLayout(e);
      el.clearPoints();
      el.addPoint(
         G.getCenterX(e.source()),
         G.getCenterY(e.target()));
    }
  }
```

⇝ API: LayoutGraph

Integration of New Algorithm

YModule replaces 'public static void main(String args[])'

```
public class DiagonalLayoutModule extends LayoutModule {

  public DiagonalLayoutModule() {
    super("Demo","MG","DiagonalLayout");
  }

  protected void mainrun() {
    DemoLayouter layouter = new DemoLayouter();
    launchLayouter(layouter);
  }
}
```

Integration of New YModule

- derive class from

  `demo.yext.graphml.ViewActionDemo`

- override `createMenuBar`

```
protected JMenuBar createMenuBar(){
   JMenuBar bar = super.createMenuBar();
   JMenu menu = new JMenu("Layout");
   menu.add(new LaunchModule(new DiagonalLayoutModule()));
   bar.add(menu);
   return bar;
}
```

Integration of New YModule (2)

- put *path_to_yfiles/demo/yed/yed.jar* into `CLASSPATH`
- create *tools.pkg* in ∼/`.yed` with

```
<PACKAGE name="PACKAGE_TOOLS">
  <MODULE name="DiagonalLayoutModule"
      class="package.DiagonalLayoutModule">
  </MODULE>
</PACKAGE>
```

- use `java yed.Launcher`

CVS

CVS = Concurrent Versions System

- secure tunnel:

  ```
  ssh -N -L 2401:i11raid.ira.uka.de:2402 algoprakt@i11raid.ira.uka.de
  ```

- repository:

  ```
  /home/algo/lehre/cvsroot
  ```

- commands:

  ```
  add, checkout, commit, login, update
  ```

Teamwork: A few harmless flakes working together can unleash an avalanche of destruction.