Resurrect your Java Applets

Ioannis Kostaras

JCRETE 2015

Introduction

- A long time ago in a galaxy far, far away … the Internet was created
- It only served static web pages based on HTML (HyperText Markup Language) **Boring**
- After a while, interactivity appeared:
  - JavaScript
  - Adobe Flash
  - Java Applet

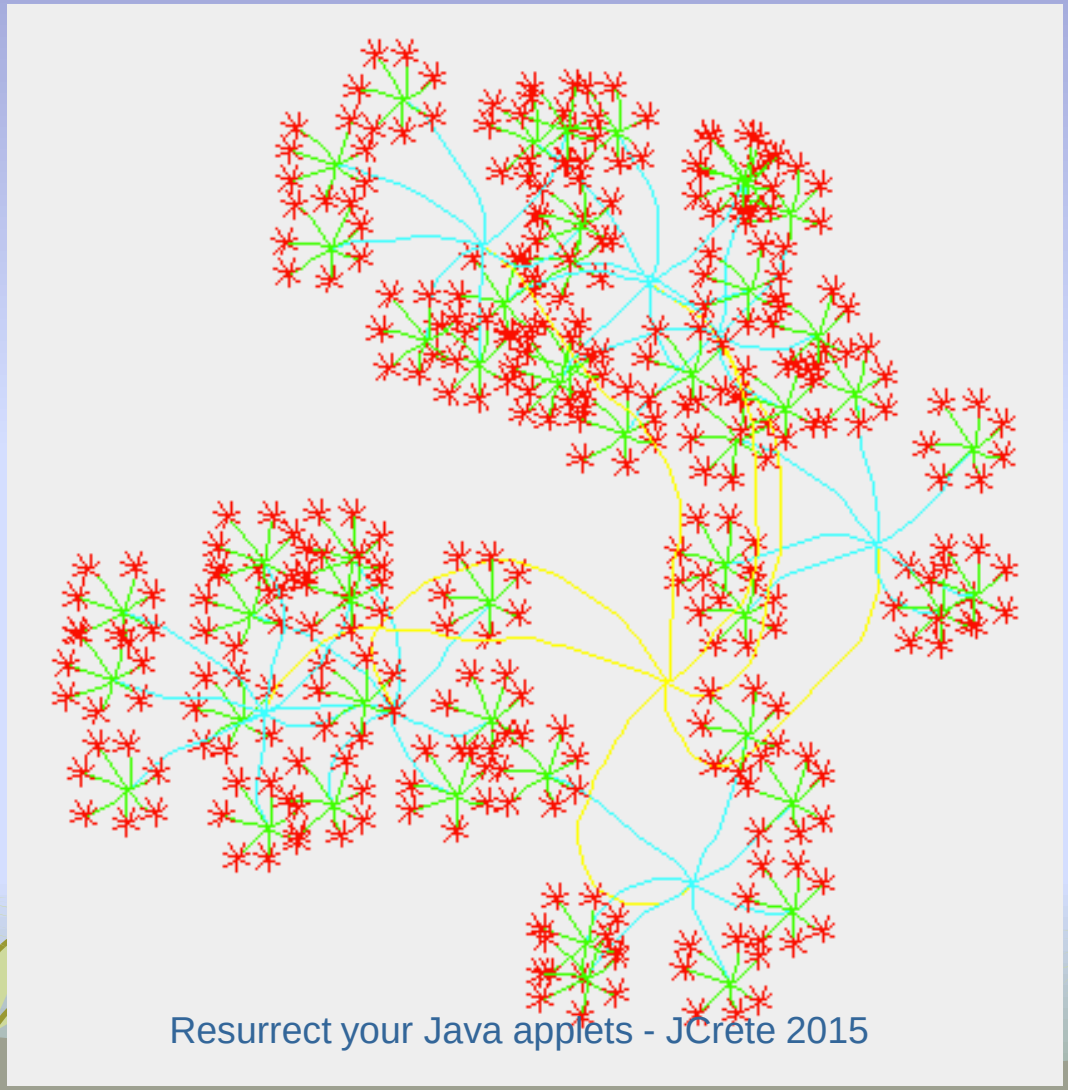
Introduction

- There are many Java (AWT) Applets on the Internet
  - E.g. for teaching Physics, Chemistry, Math but not only
- But, they don't work in modern browsers

Solutions?

- JavaFX
- DukeScript
- HTML 5 / Javascript
- ?

A JAVA APPLET



Skeleton of a Java Applet

```
public class MyApplet extends java.applet.Applet {

  public void init() { ... }

  public void start() { ... }

  public void paint (java.awt.Graphics g) { ... }

  public void stop() { ... }

  public void destroy() { ... }
}
```

```
<APPLET code="MyApplet.class" width=600
height=600></APPLET>
```

Tendrils Java Applet

```
public class Tendrils extends
java.applet.Applet {
  private static final int X = 300;
  private static final int Y = 300;
  // max number of segments
  private static final int SEGS =
60;
  private Cluster cluster;

  public void init() {
    setSize(600, 600);
    cluster = new Cluster();
  }

  public void start() {  }

  public void paint(Graphics
g) {
    cluster.display(g, SEGS,
X, Y);
  }

  public void stop() { }

  public void dispose() {  }
...
}
```

Tendrils Java Applet (cont.)

```
private static class Cluster {
    // Number of helixes in a cluster
    private static final int NUMBER = 7;

    public void display(Graphics g, int size, int
x, int y) {
      for (int i = 0; i < NUMBER; i++) {
         double theta = i * 2 * Math.PI / NUMBER;
         Tendril t = new Tendril();
         t.display(g, size, theta, x, y);
      }
    }
}
```

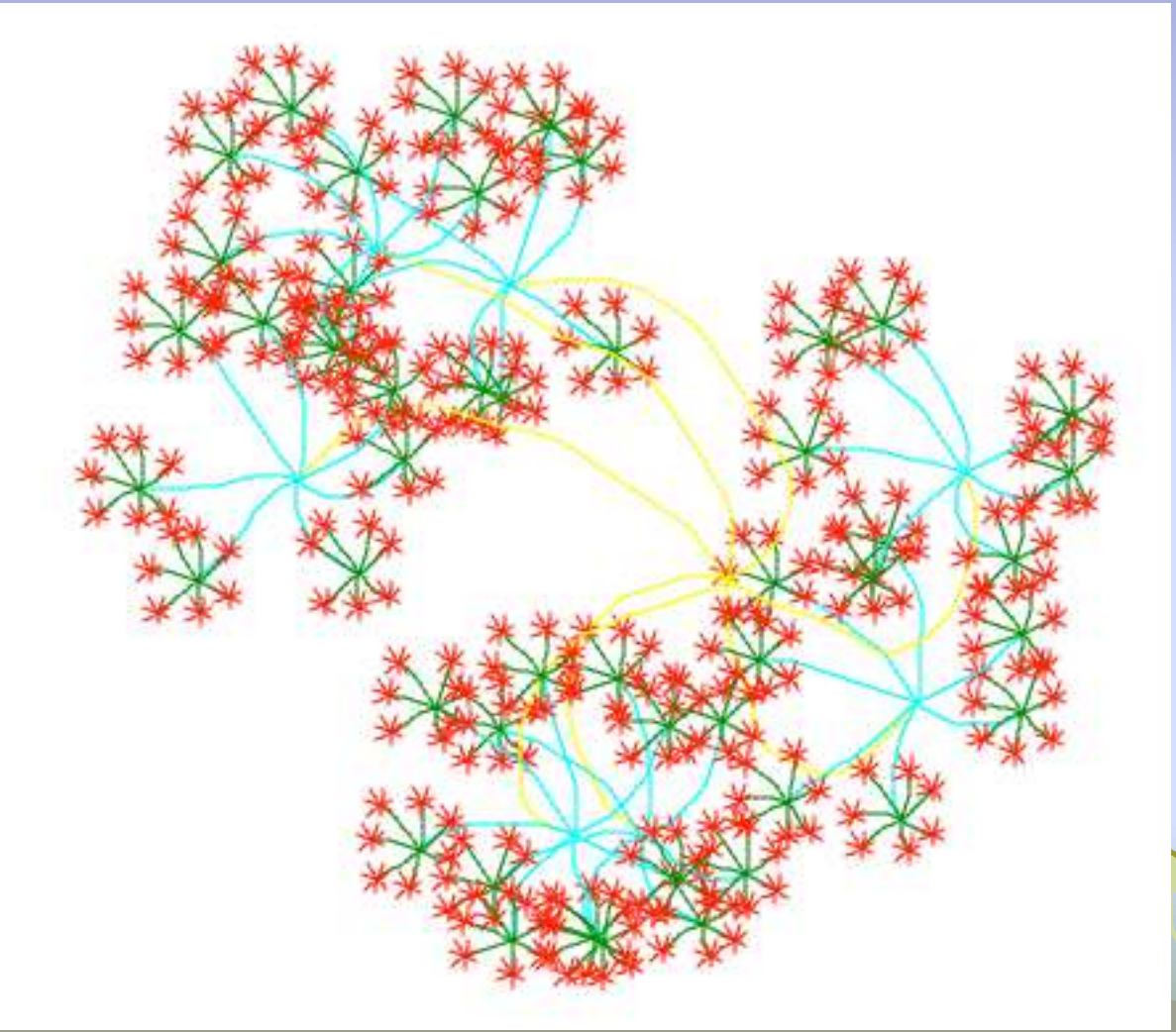
Tendrils Java Applet (cont.)

```
private static class Tendril {
    // Numerator of probability
    private static final int NUME = 30;
    // Denominator of probability
    private static final int DENOM = 100;
    // length of line segments
    private static final double RAD = 3.0;
    // change in angle theta (in rad)
    private static final double DELTHETA = 0.1;
    // division of number of helixes
    private static final int REDUX = 3;
    // min number of segments
    private static final int MIN = 1;
```

Tendrils Java Applet (cont.)

```
public void display (Graphics
g, int size, double theta,
int x, int y) {
  for(int j=0; j<size; j++) {
  int chng = ((int) (DENOM *
Math.random()) < NUME) ? -1 :
1;
  theta += chng * DELTHETA;
  int x0 = x;
  int y0 = y;
  x += RAD*Math.sin(theta);
  y += RAD*Math.cos(theta);
  if (size < 4) {
    g.setColor(Color.red);
  } else if (size < 13) {
    g.setColor(Color.green);
  } else if (size < 40) {
    g.setColor(Color.cyan);
  } else {
    g.setColor(Color.yellow);
  }
  g.drawLine(x0, y0, x, y);
}
if (size > MIN) {
    Cluster c = new Cluster();
    int newsize = size / REDUX;
    c.display(g, newsize, x, y);
}
}
}
```

JAVAFX



Pros & Cons

- + Write in Java
- + A lot of documentation
- + Rich UI widgets
- - jars do ***not*** run in the browser
- - Need to learn a new API if you are not familiar with JavaFX before, but part of Java (AWT → Swing → JavaFX)

Create a new JavaFX application

- In NetBeans:
  - File → New Project
    - Categories: JavaFX
    - Projects: JavaFX Application

TendrilsFX

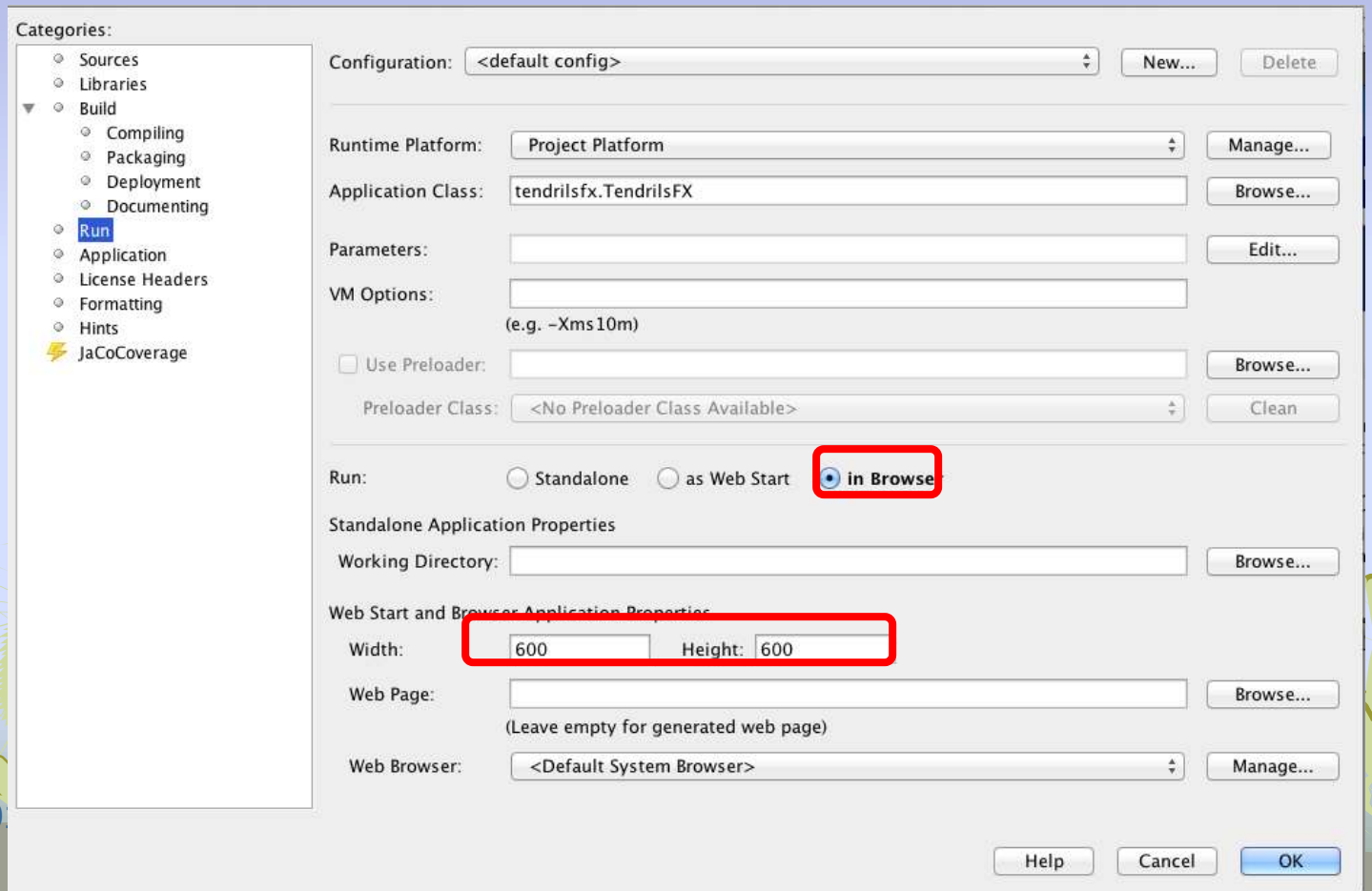
```
public class TendrilsFX extends Application {
  // max number of segments
  private static final int SEGS = 60;
  private static final int X = 600;
  private static final int Y = 600;
  private final Cluster cluster = new Cluster();
  @Override
  public void start(Stage primaryStage) {
    Canvas canvas = new Canvas(X, Y);
    GraphicsContext g2D = canvas.getGraphicsContext2D();
    cluster.display(g2D, SEGS, X/2, Y/2);
    StackPane root = new StackPane();
    root.getChildren().add(canvas);
    Scene scene = new Scene(root, X, Y);
    primaryStage.setTitle("Tendrils!");
    primaryStage.setScene(scene);
    primaryStage.show();
  }
```

TendrilsFX (cont.)

```
if (size < 4) {
g.setStroke(javafx.scene.paint.Color.RED);
} else if (size < 13) {
g.setStroke(javafx.scene.paint.Color.GREEN)
;
} else if (size < 40) {
g.setStroke(javafx.scene.paint.Color.CYAN);
} else {
g.setStroke(javafx.scene.paint.Color.YELLOW
);
}
g.strokeLine(x0, y0, x, y);
```

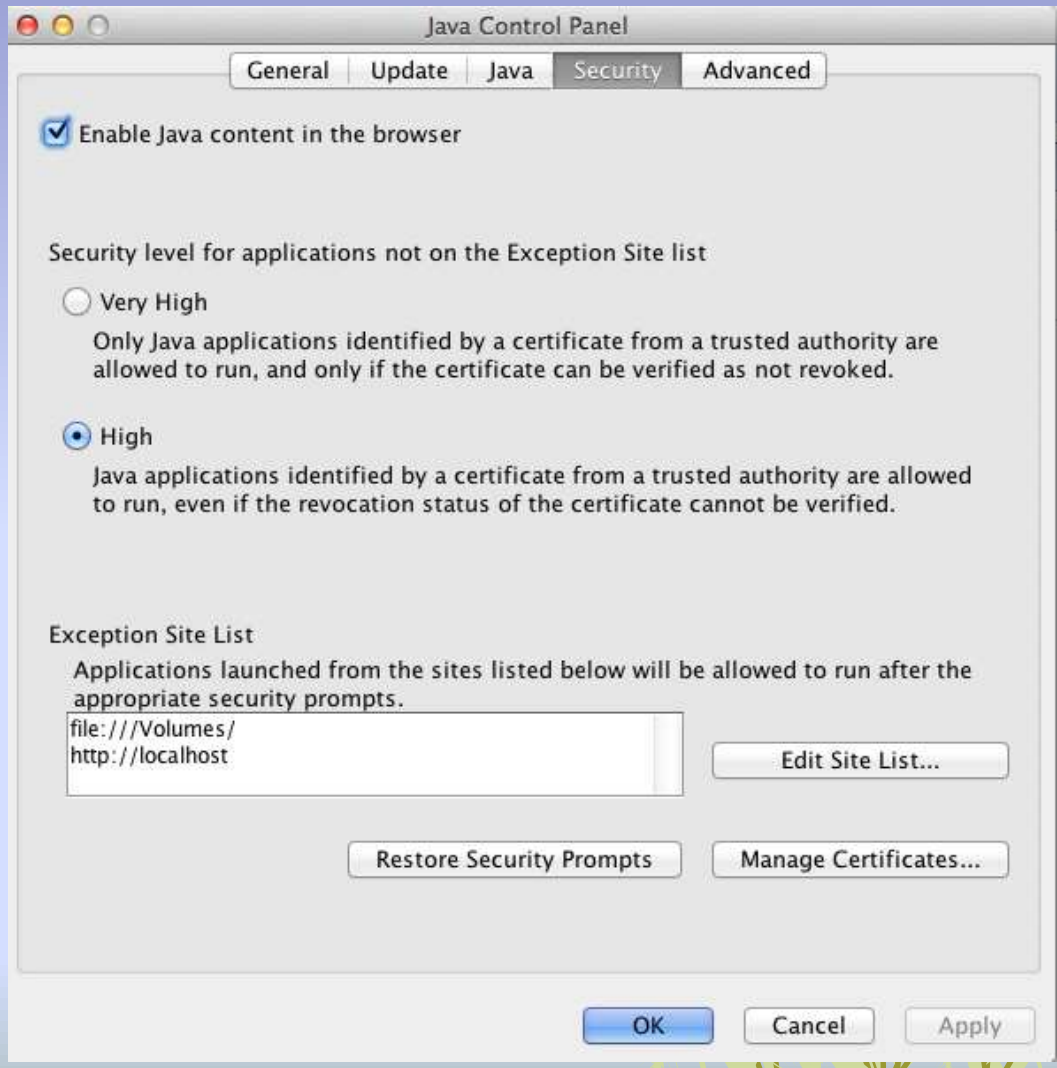
Deploy (1)

- In NetBeans, right-click on the TendrilsFX project and select **Properties → Run**
- Clean & build (uses Deployment Toolkit `dtjava.js`)



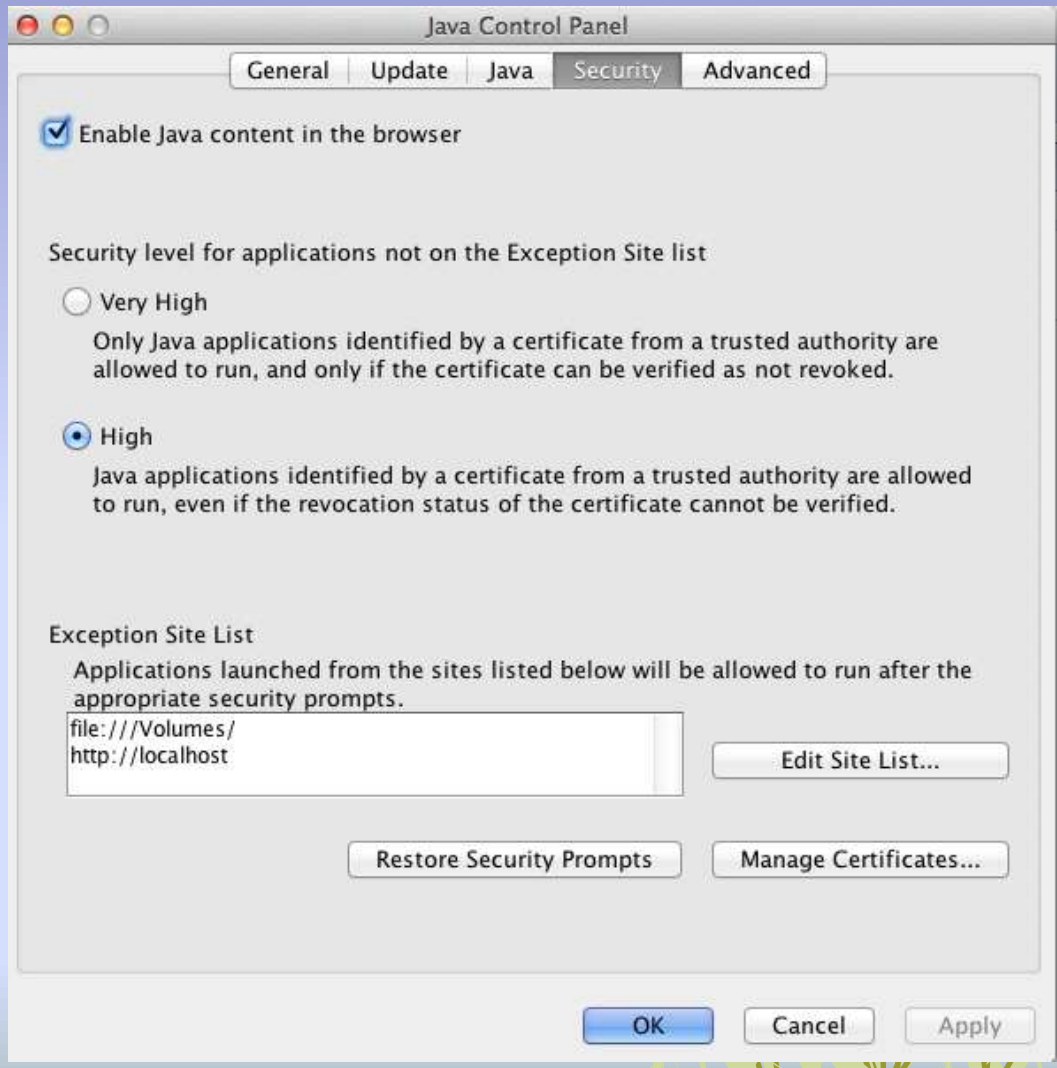
Deploy (2) - Firefox

- Firefox
  - Open **Java Control Panel**
  - Click on **Edit Site List**
  - Add the sites that contain the JavaFX applet
  - Click on **OK**
- In Firefox enable Java Applet plugin



Deploy (3) - Safari

- Safari doesn't allow to execute an applet from the filesystem, i.e. with URL file:///
- Deploy your JFX applet to a web server, e.g. Apache
- In **Java Control Panel** allow **http://localhost**



Deploy (4) – Opera, Chrome, IE

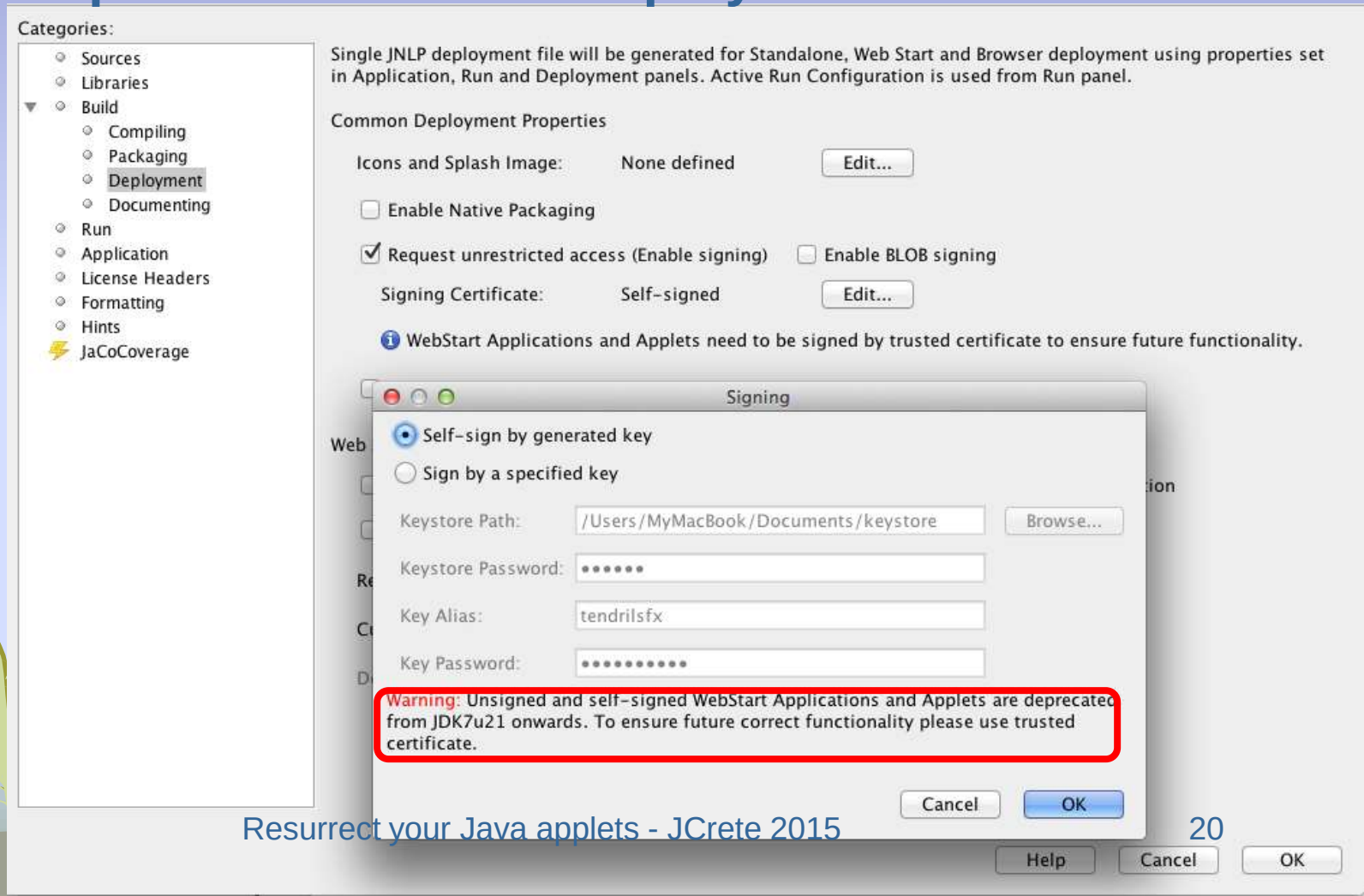
- Opera & Chrome do not allow execution of JavaFX applets

```
Content can not be displayed using
your Web browser. Please open this
page using another browser.
```

- Internet Explorer (I couldn't make it run)

Deploy (5) – Sign the jar

- In NetBeans, right-click on the TendrilsFX project and select **Properties → Build → Deployment**
- Clean & build



DUKESCRIPT

What is DukeScript

- DukeScript is a new technology that allows you to write your logic in Java and render the result to a number of clients, which can be web browser, portable devices etc.
- It achieves this by transforming Java code to HTML 5 (Javascript) which can run in any device.

How does it work

Pros & Cons

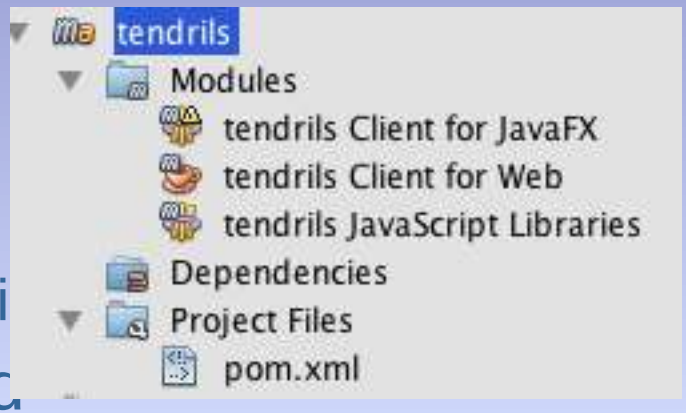
- + Write in Java
- + Write once run everywhere (web, JavaFX, Android, iOS, …)
- + API similar to JavaFX
- - not a lot of documentation available
- - Need to learn a new API

Installation

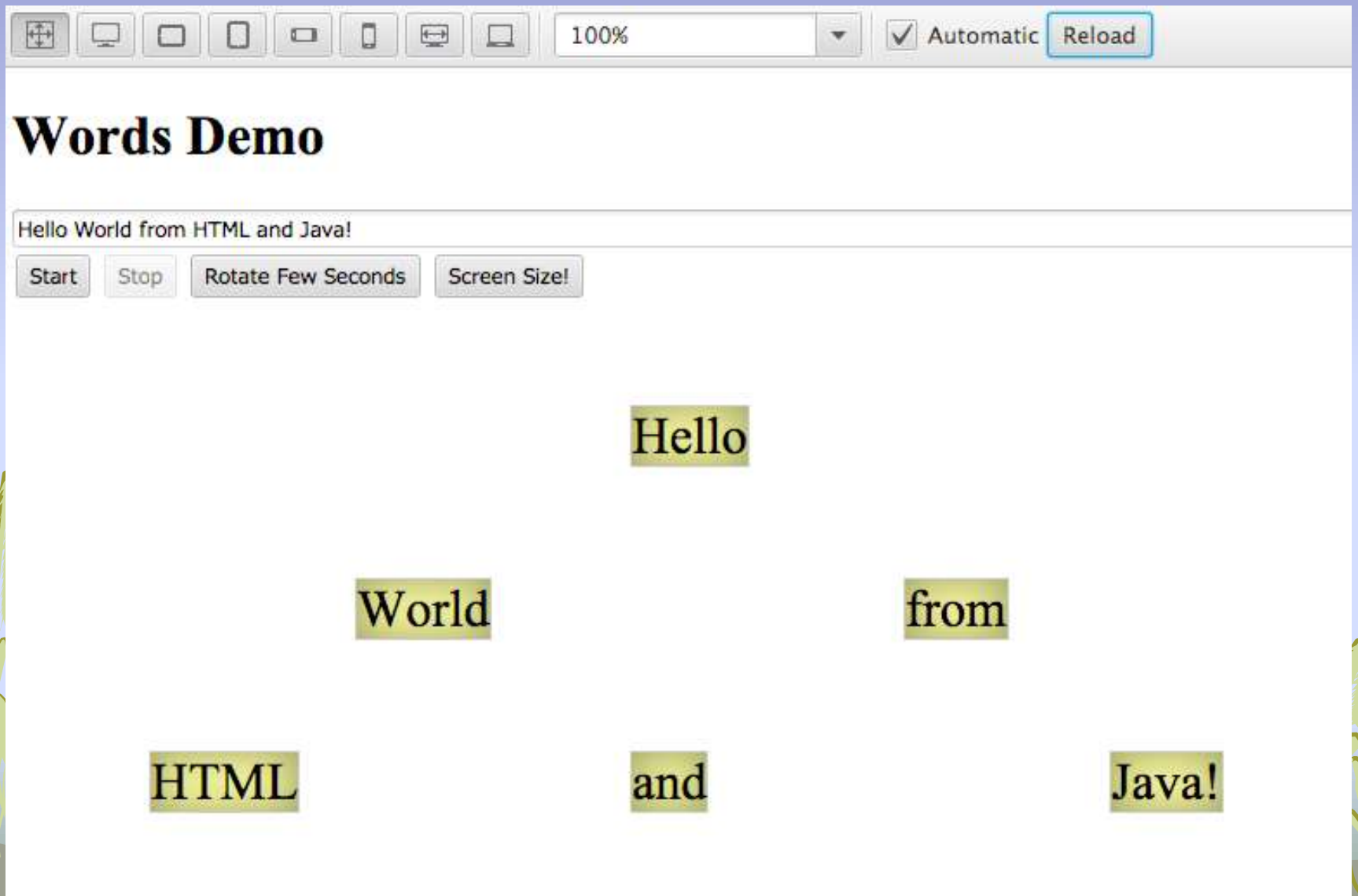
- In NetBeans install the DukeScript plugin
  - Tools → Plugins → Available Plugins
    - DukeScript Project Wizard
    - Mine Sweeper

Create a new DukeScript application

- In NetBeans:
  - File → New Project
    - Categories: HTML5
    - Projects: HTML5 with Applicati
  - Provide an artifact & group ic
  - Choose your platform: Browser
  - Select *Knockout 4 Java Maven Archetype* template
  - Right click on it and choose **Build with Dependencies**.
  - Execute *tendrils Client for JavaFX*
  - Execute *tendrils Client for Web*



Words Demo


100%
Automatic
Reload
Words Demo
Hello World from HTML and Java!
Start
Stop
Rotate Few Seconds
Screen Size!
Hello
World
from
HTML
and
Java!

Modify the Words Demo

- You only need to modify *tendrils Client for JavaFX*
- Delete `DataModel` class and any test classes
- Cleanup the body of static `onPaneLoad()` method of `Main`
- Clean up `index.html`
- Copy `Tendril` and `Cluster` inner classes to `Main`
- Copy the contents of `paint()` to `onPageLoad()` and any constant fields

index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
    </head>
    <body>
        <h1>Words Demo</h1>
        <!-- boot bck2brwsr -->
        <script type="text/javascript" src="bck2brwsr.js"></script>
        <script>
            var vm = bck2brwsr('tendrils.js');
            var c = vm.loadClass('tendrils.tendrils.BrowserMain');
            c.invoke('main');
        </script>
    </body>
</html>
```

Modify the Words Demo

- Convert `java.awt.Graphics` to its DukeScript equivalent `com.dukescript.api.canvas.GraphicsContext2D`

```
public static void onPageLoad()
throws Exception {
   GraphicsContext2D g =
HTML5Graphics.getOrCreate("tendri
ls");
   cluster.display(g, SEGS, X/2,
Y/2);
}
```

Modify the Words Demo

- Convert `java.awt.Color` to its DukeScript equivalent `g.setStrokeStyle`

```
if (size < 4) {
  g.setStrokeStyle(g.getWebColor("#ff0000"));  //
Color.RED
} else if (size < 13) {
  g.setStrokeStyle(g.getWebColor("#00ff00"));  //
Color.GREEN
} else if (size < 40) {
  g.setStrokeStyle(g.getWebColor("#00FFFF"));  //
Color.CYAN
} else {
 g.setStrokeStyle(g.getWebColor("##FFFF00"));  //
Color.YELLOW
}
```

Modify the Words Demo

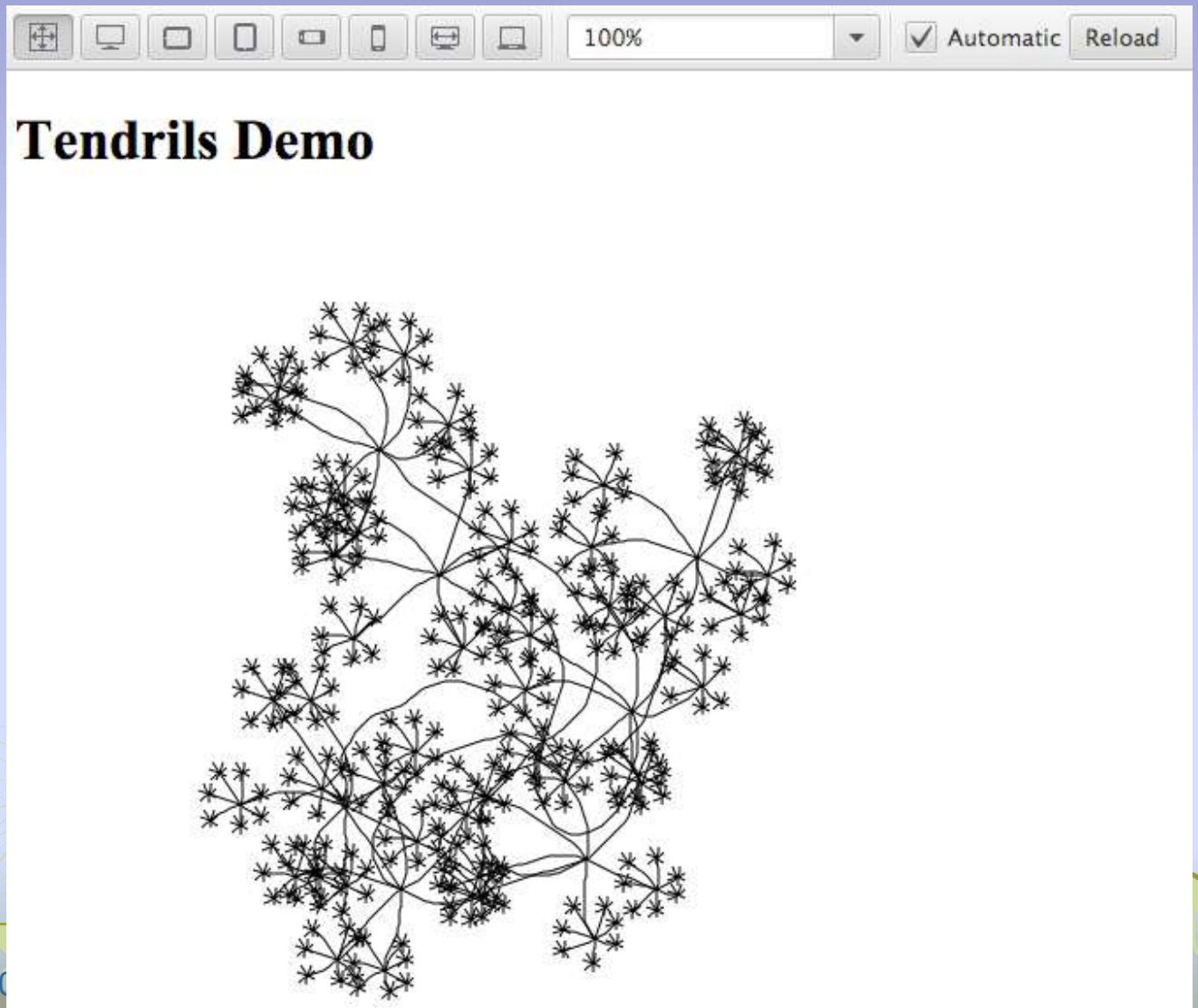
- Convert `g.drawLine(x0, y0, x, y)` to its DukeScript equivalent

```
g.beginPath();
g.moveTo(x0, y0);
g.lineTo(x, y);
g.stroke();
```

- Modify `index.html`

```
<h1>Tendrils Demo</h1>
<canvas width="600" height="600"
id="tendrils"></canvas>
<!-- boot bck2brwsr -->
```

Run the Tendrils Demo



Deployment

- contents of `tendrils/client-web/target/tendrils-web-1.0-SNAPSHOT-bck2brwsr/public_html/`:
  - `lib/`
  - `bck2brwsr.js`
  - `index.html`
  - `tendrils.js`
- That's it! Open `index.html` in your modern browser and let the magic work!
- No security exceptions any more!

Knockout.js

- Knockout.js lets you create a direct connection between the underlying data and its presentation
- After linking an HTML element with a particular data object, any changes to that object are *automatically* reflected in the DOM (*automatic dependency tracking*)
- E.g. deleting an object from a JS array deletes it automatically from the corresponding HTML elements

HTML 5

Pros & Cons

- + A lot of documentation
- + Easy deployment
- - write in Javascript
- - Need to learn a new language if not already familiar with

Create a new HTML5 application

- In NetBeans:
  - File → New Project
    - Categories: HTML5
    - Projects: HTML5 Application

index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    </head>
    <body>
        <div>TODO write content</div>
    </body>
</html>
```

Add a canvas

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tendrils</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <script type="text/javascript">
            function loader() {
                var canvas = document.getElementById('canvas');
                var context = canvas.getContext('2d');
            }
        </script>
    </head>
    <body onload="loader()">
         <h1>Tendrils Demo!</h1>
         <canvas id="canvas" width="600" height="600">
         </canvas>
    </body>
</html>
```

Keep in mind

- local variables are defined with the keyword `var`; don't forget this; otherwise JavaScript assumes it is global variable

```
int newsize = size / REDUX;
```
➔
```
var newsize = size / REDUX;
```

- Define classes like so:

```
var Custer = new function() {}
```

- Define member methods like so:

```
Cluster.prototype.display = function
(g, size, x, y) {}
```

Ported code

```
<script type="text/javascript”>
   // Cluster
   var Cluster = function() {
       var.self = this;
       var NUMBER = 7;      // Number of helixes in a cluster
       self.display = function (g, size, x, y) {
          for (var i = 0; i < NUMBER; i++) {
              var theta = i * 2 * Math.PI / NUMBER;
              var t = new Tendril();
              t.display(g, size, theta, x, y);
          }
       };
   };
```

Ported code

```
// Tendril
var Tendril = function() {
   var.self = this;
   var NUME = 30;          // Numerator of probability
   var DENOM = 100;        // Denominator of probability
   var RAD = 3.0;          // length of line segments
   var DELTHETA = 0.1;     // change in angle theta (in rad)
   var REDUX = 3;          // division of number of helixes
   var MIN = 1;            // min number of segments
   self.display = function(g, size, theta, x, y) {
      for (var j = 0; j < size; j++) {
      var chng = (((DENOM * Math.random()) | 0) < NUME) ? -1 :
1;
      theta += chng * DELTHETA;
      var x0 = x;
      var y0 = y;
      x += RAD * Math.sin(theta);
      y += RAD * Math.cos(theta);
```

Ported code

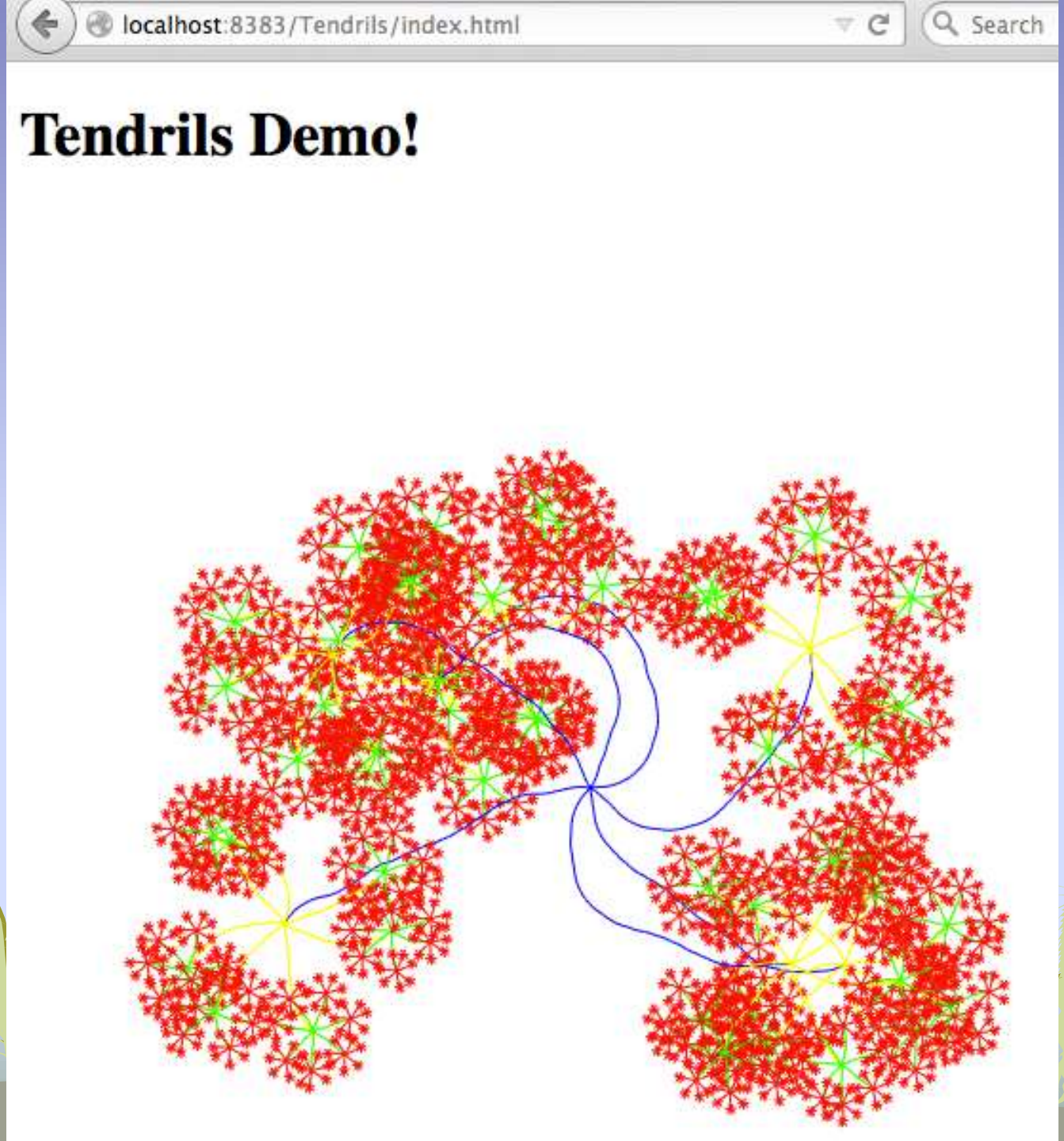
```
    if (size < 4) {
       g.strokeStyle = '#ff0000';  // Color.red
    } else if (size < 13) {
       g.strokeStyle = '#00ff00';  // Color.green
    } else if (size < 40) {
       g.strokeStyle = '#ffff00';  // Color.cyan
    } else {
       g.strokeStyle = '#0000ff';  // Color.yellow
    }
    g.beginPath();
    g.moveTo(x0, y0);
    g.lineTo(x, y);
    g.stroke();
}
```

Ported code

```
 if (size > MIN) {
    var c = new Cluster();
    var newsize = size / REDUX;
    c.display(g, newsize, x, y);
 }
};

var X = 300;
var Y = 300;
var SEGS = 60;          // max number of segments

function loader() {
   var canvas = document.getElementById('canvas');
   var context = canvas.getContext('2d');
   var cluster = new Cluster();
   cluster.display(context, SEGS, X, Y);
}
</script>
```


localhost:8383/Tendrils/index.html
Search
Tendrils Demo!

Deployment

- Just one file:

`Tendrils/public_html/index.html`

ADD INTERACTIONS

Java Applet - Tendrils

```
public class Tendrils extends Applet {
  ...
 public void init() {
    setSize(600, 600);
    cluster = new Cluster();
    this.addMouseListener(new MouseAdapter() {
      public void mouseClicked(MouseEvent e) {
         repaint();
      }
    });
 }
```

JavaFX - TendrilsFX

```
public class TendrilsFX extends Application {
   ...
   @Override
   public void start(Stage primaryStage) {
      ...
 scene.addEventHandler(MouseEvent.MOUSE_CLICKED,
(MouseEvent event) -> {
                g2D.clearRect(0, 0, X, Y);
                cluster.display(g2D, SEGS, X/2, Y/2);
                event.consume();
          });
      ...
   }
```

HTML 5 - Tendrils

```
var X = 300;
var Y = 300;
var SEGS = 60;              // max number of segments
function loader() {
 var canvas = document.getElementById('canvas');
 onMouseClick(null);
 canvas.addEventListener("click", onMouseClick, true);
}
function onMouseClick(evt) {
  var canvas = document.getElementById('canvas');
  var context = canvas.getContext('2d');
  context.clearRect(0, 0, canvas.width,
canvas.height);
 var cluster = new Cluster();
 cluster.display(context, SEGS, X, Y);
}
```

DukeScript - tendrils

index.html

```
<canvas width="600" height="600" id="tendrils"
data-bind="click: $root.refresh()"></canvas>
```

```
public final class Main {
    private Main() {  }
    public static void
main(String... args) throws
Exception {
  BrowserBuilder.newBrowser().
   loadPage("pages/index.html").
  loadClass(Main.class).
     invoke("onPageLoad", args).
       showAndWait();
  System.exit(0);
 }
```

```
    /**
     * Called when the page is ready.
     */
    public static void onPageLoad()
throws Exception {
        //DataModel.onPageLoad();
        Data d = new Data();
        d.applyBindings();
    }

}
```

DukeScript - tendrils

```
@Model(className = "Data", targetId = "",
properties = {})
final class DataModel {
    private static final int X = 600;
    private static final int Y = 600;
    private static final int SEGS = 60;          //
max number of segments
    private static final Cluster cluster = new
Cluster();
    private static final GraphicsContext2D g =
HTML5Graphics.getOrCreate("tendrils");
    @Function
    static void refresh() {
        g.clearRect(0, 0, X, Y);
        cluster.display(g, SEGS, X / 2, Y / 2);
    }
```

References

- Kostovarov D. (2013), *Deployment in the Browser*, Oracle
- DukeScript official site
- Development Environment for Web
- WebFX
- JavaFX in the Browser

References

- Lafore R. (1991), *Object Oriented Programming in Turbo C++*, Waite Group.
- Stefanov S. (2008), *Object Oriented JavaScript*, Packt.
- *Introduction to Object-Oriented JavaScript*, Mozilla Development Network.
- Geary D. (2012), *Core HTML5 Canvas,* Prentice Hall
- Hawkes R. (2011), *HTML5 Canvas,* Friendsofed.