JavaOne

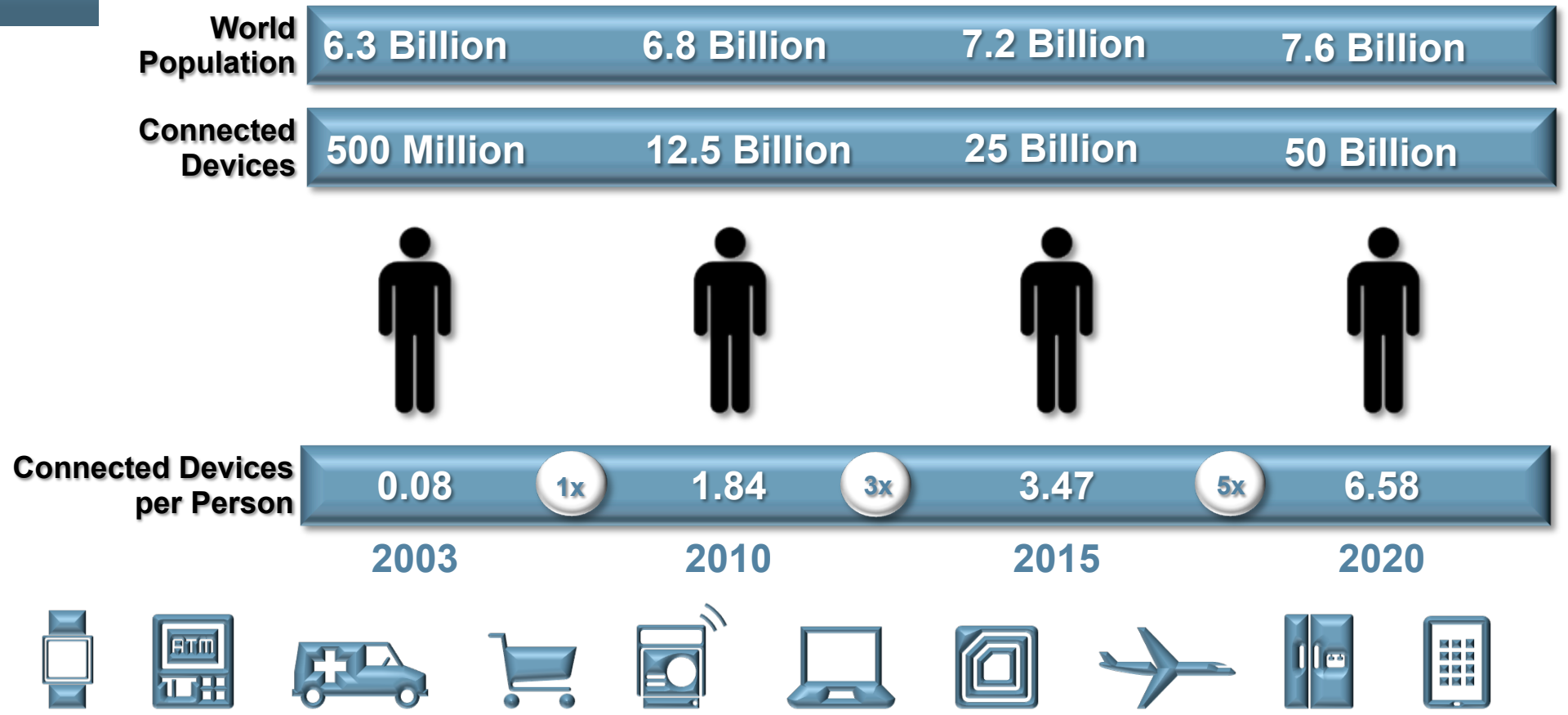
Java SE 8 for Tablets, Pis, and Legos

Stephen Chin / Jim Weaver
Java Technology Ambassadors

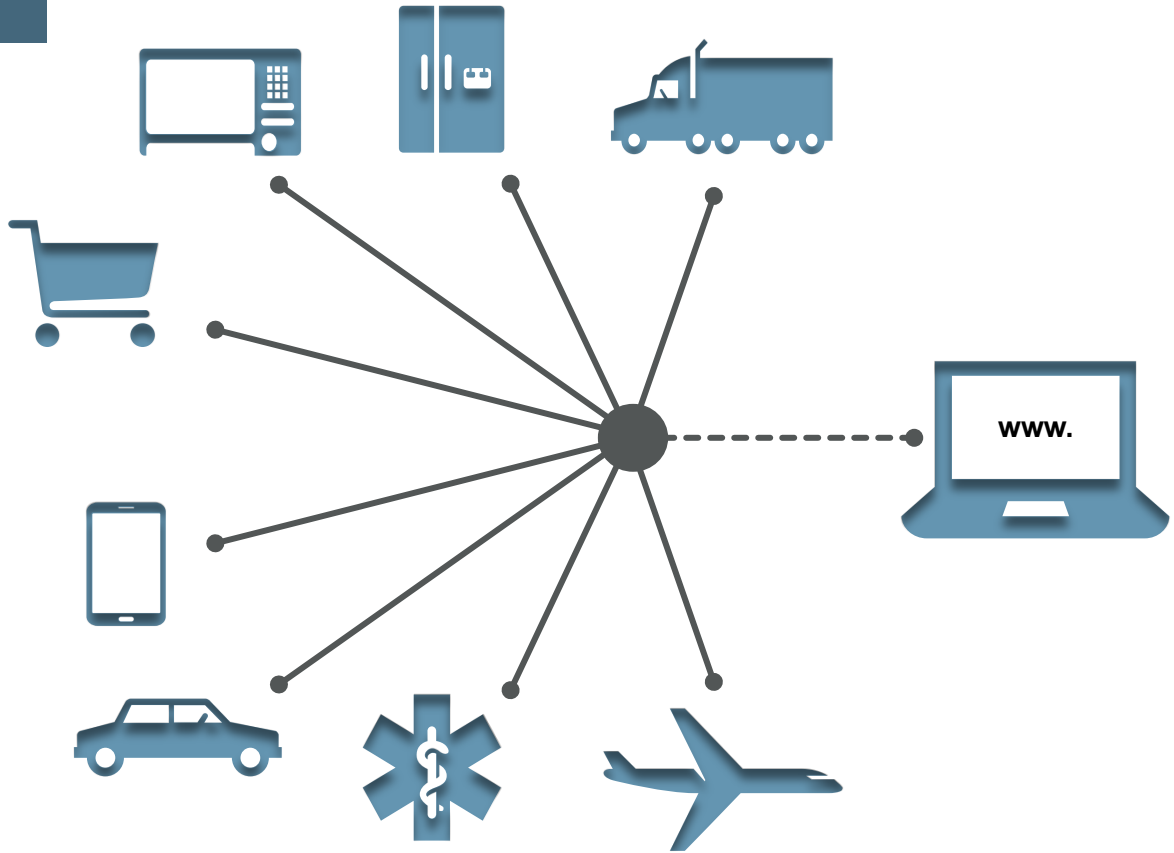
@steveonjava
@JavaFXpert

http://commons.wikimedia.org/wiki/File:Genting_Highlands_theme_park.jpg


Lego Mindstorms
Dev Boards
DukePad
Windows Tablet
iOS
Android

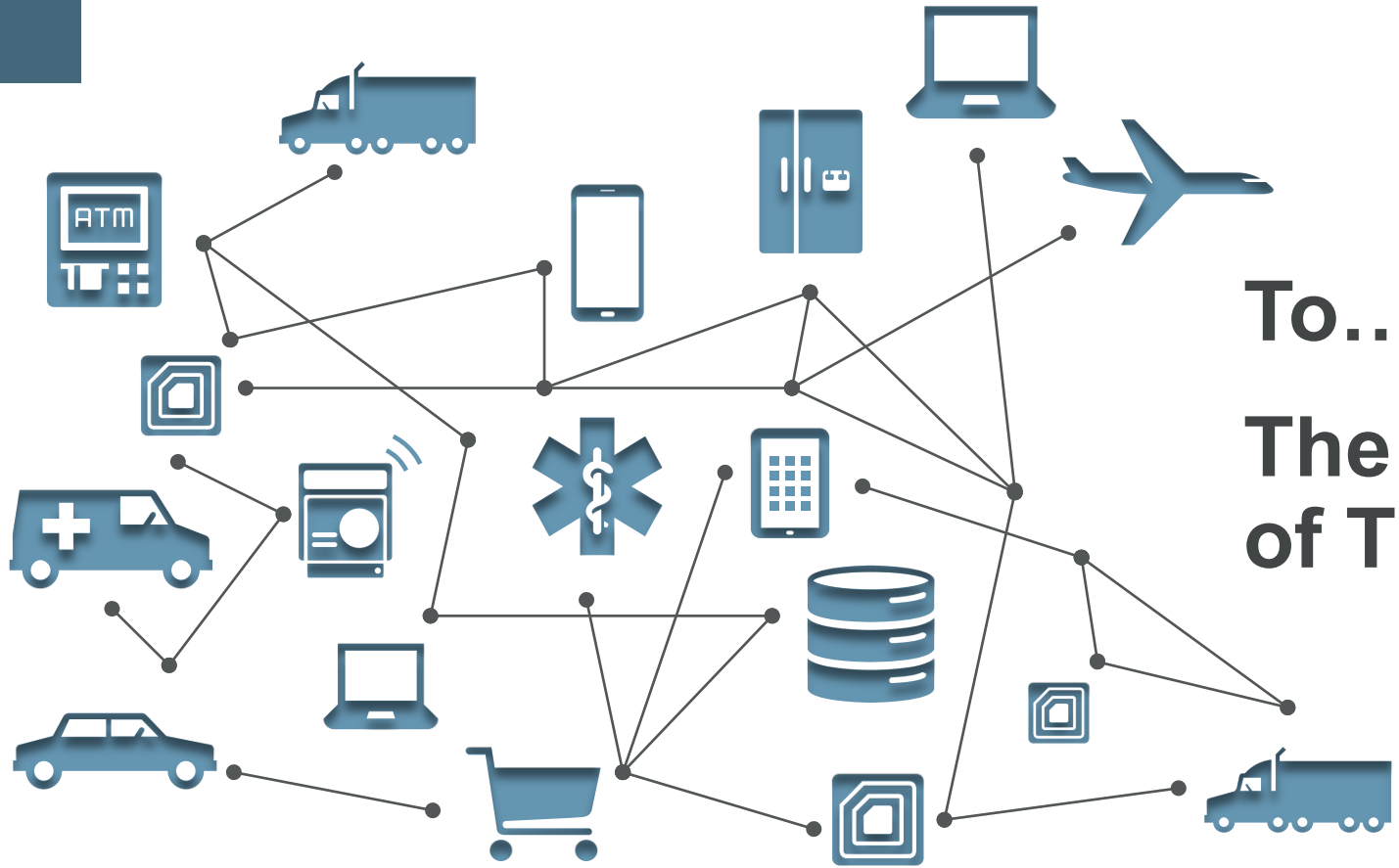


Source: Cisco



From…

Things Connected to the Internet

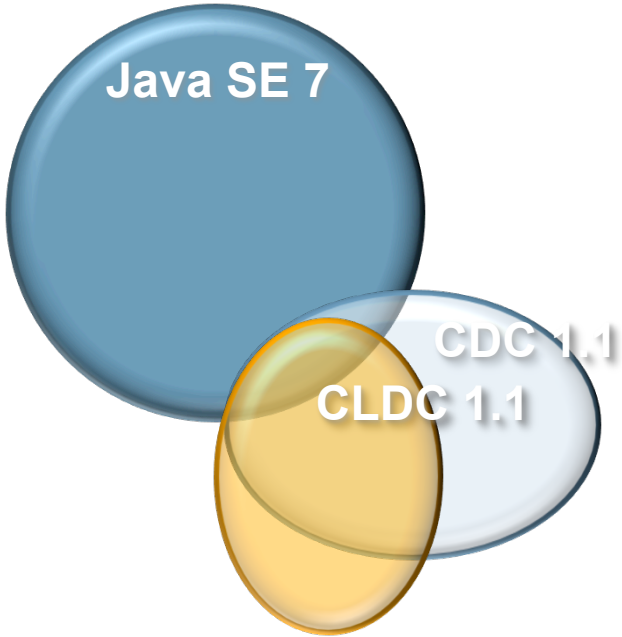


To…

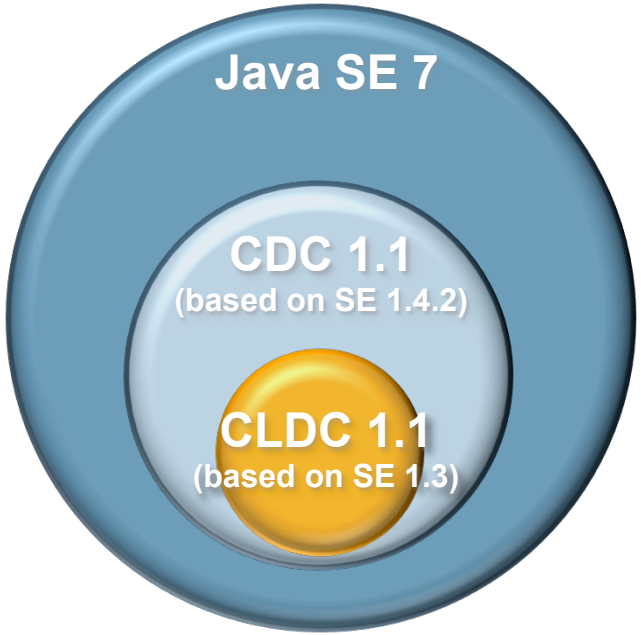
The Internet of Things

Today

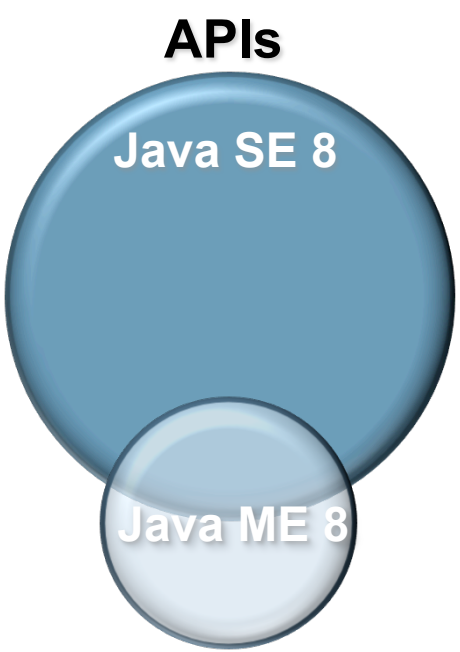
APIs

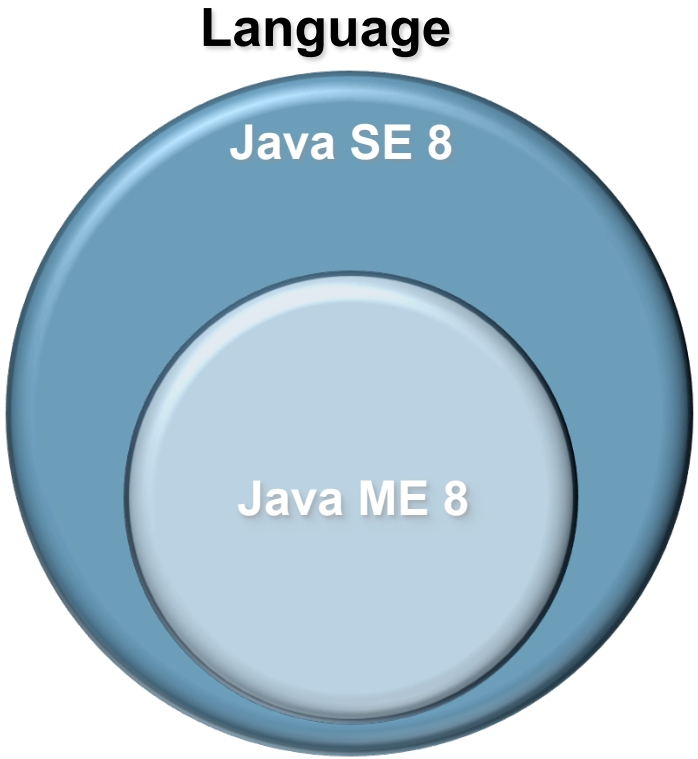


Language

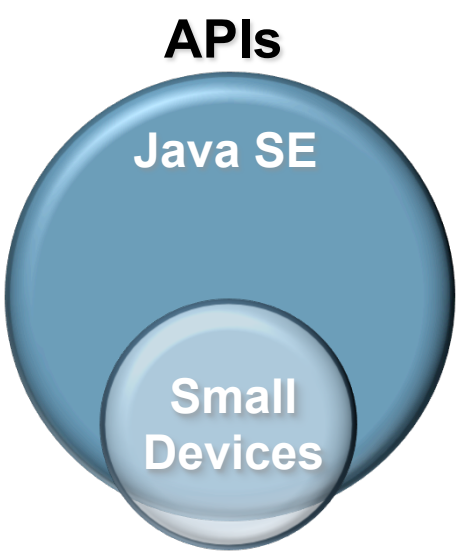


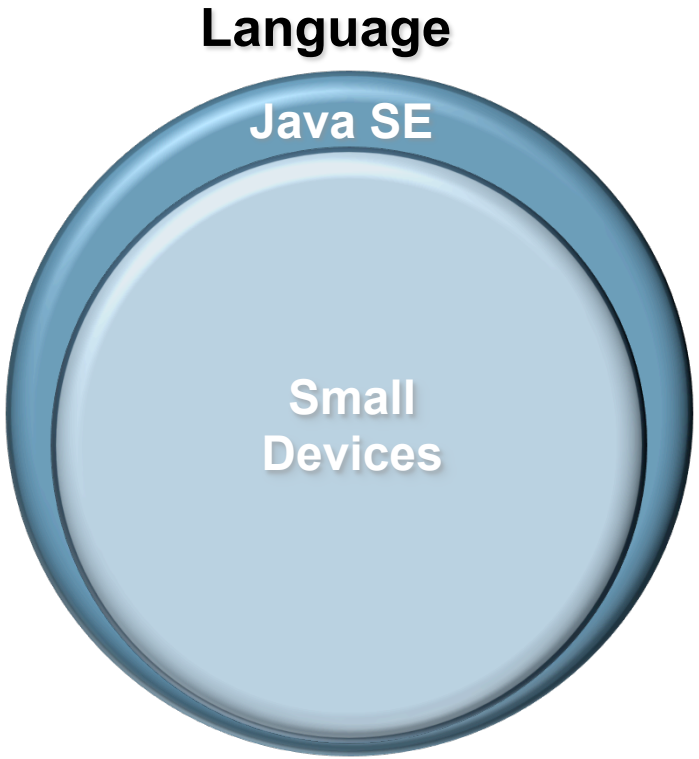
Java 8





Beyond Java 8





	Java Embedded	Native/C/C++
Cross-platform/multi-architecture/	Yes	No
Sandbox security model	Yes	No
Robustness	Excellent	Varied
Multi-tasking/multi-threading	Built-in	Add-on, platform-specific
Automatic memory management	Yes	No
Pre-integrated and customizable	Yes	Varied
Performance	Optimized	Varied
Code updatability	Excellent	Varied
Efficient, scalable development model from small embedded to large systems	Yes	No
Developer community	Large	Fragmented

LeJOS

How it works on the EV3

LEJOS
Java for LEGO Mindstorms

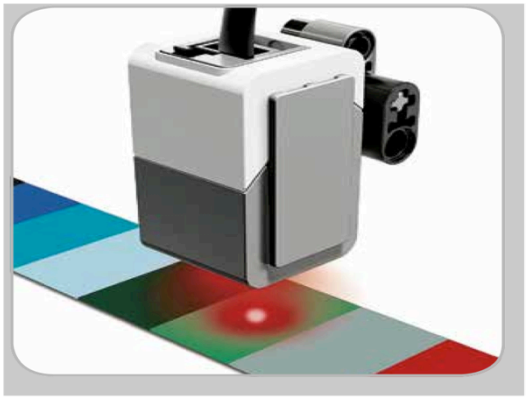
The Heart of the EV3

- TI Sitara AM1808
  - ARM9, 300Mhz
- 64MB RAM / 16MB Flash
- Analog to Digital Converter
- 4 Motor Ports
- 4 Sensor Ports
- Bluetooth / USB
- MicroSD

EV3 Motors

EV3 Sensors

Color and Light Sensor



Color Mode

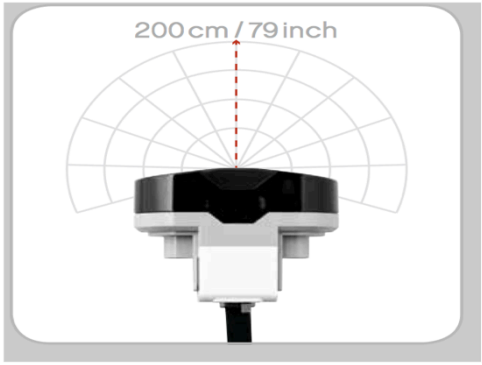
Reflected Light Intensity Mode

Ambient Light Intensity Mode

Ultrasonic Sensor

Measuring mode Vs Presence Mode

Infrared Sensor



Remote Control

Getting Started with LeJOS

Creating Your SD Card

- Micro SD Card (> 2GB)
- Compatible WIFI adapter
  - NetGear WNA1100
  - EDIMAX EW-7811Un
- Linux (or a Linux VM)

- Details here:

http://sourceforge.net/p/lejos/wiki/Home/

Getting Started with LeJOS

Setting Up Your Dev Environment

- IDE
  - Eclipse Kepler (needed to compile LeJOS source)
- Jars you need:
  - OpenJDK Java 7 Runtime
  - Java Native Access (libjna)
- Download/compile LeJOS Code
  - Repo: git://git.code.sf.net/p/lejos/ev3

Creates ev3classes.jar

Simple LeJOS Application

```
import lejos.nxt.Button;
import lejos.nxt.LCD;
public class EV3FirstProgram {
    public static void main(String[] args) {
        LCD.clear();
        LCD.drawString("First EV3 Program", 0, 5);
        Button.waitForAnyPress();
        LCD.clear();
        LCD.refresh();
    }
}
```

Running on Device

Copy your jar to device:

```
Desktop$ scp EV3FirstProgram.jar root@<ev3 ip>:~
```

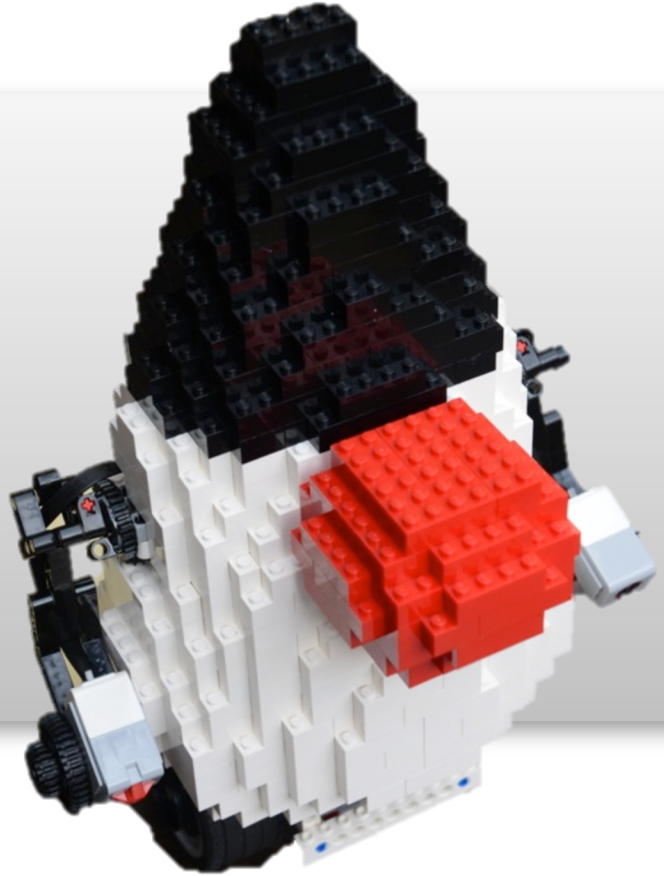
SSH to device (password is blank):

```
Desktop$ ssh root@<ev3 ip>
```

Run program:

```
EV3$ jrun –cp EV3FirstProgram.jar EV3FirstProgram
```

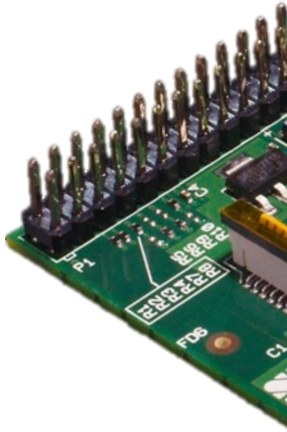
Lego Duke Segway

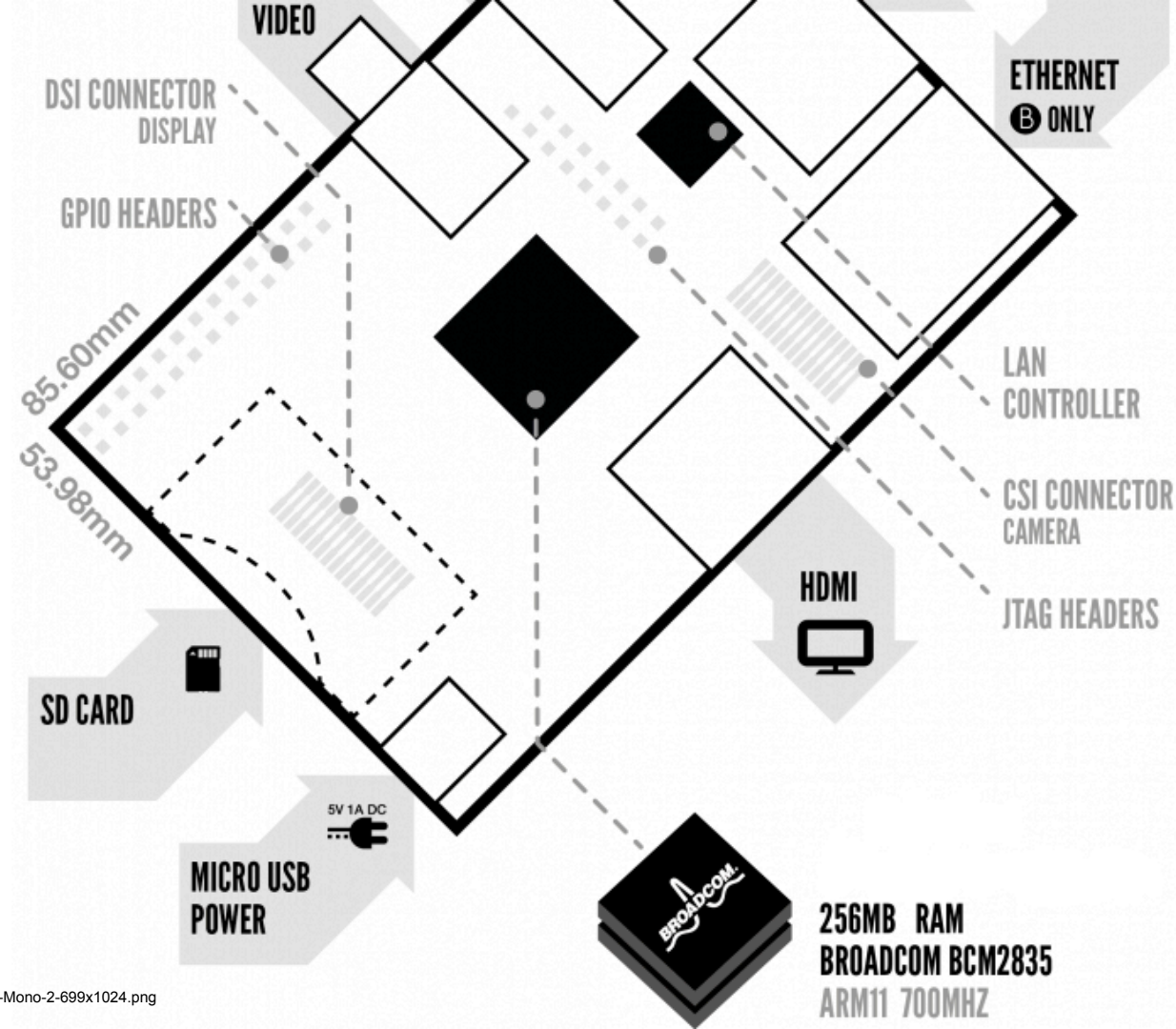


Have Java With Your Dessert

Raspberry Pi

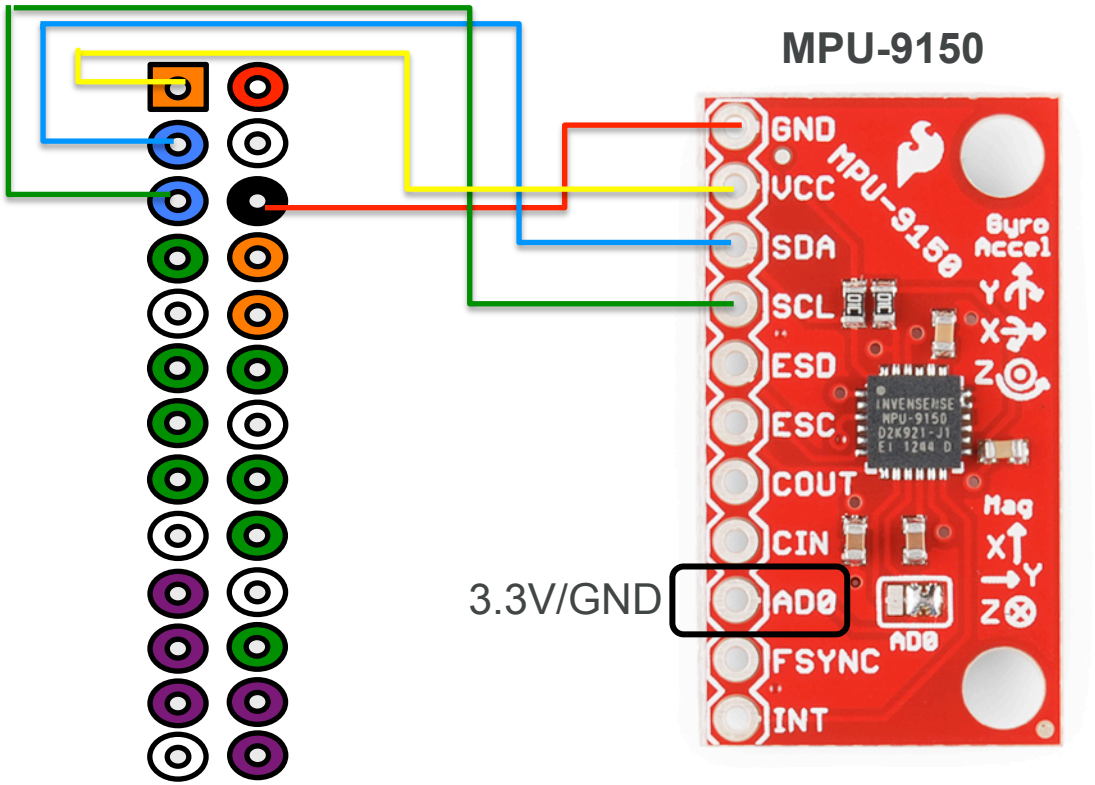
=



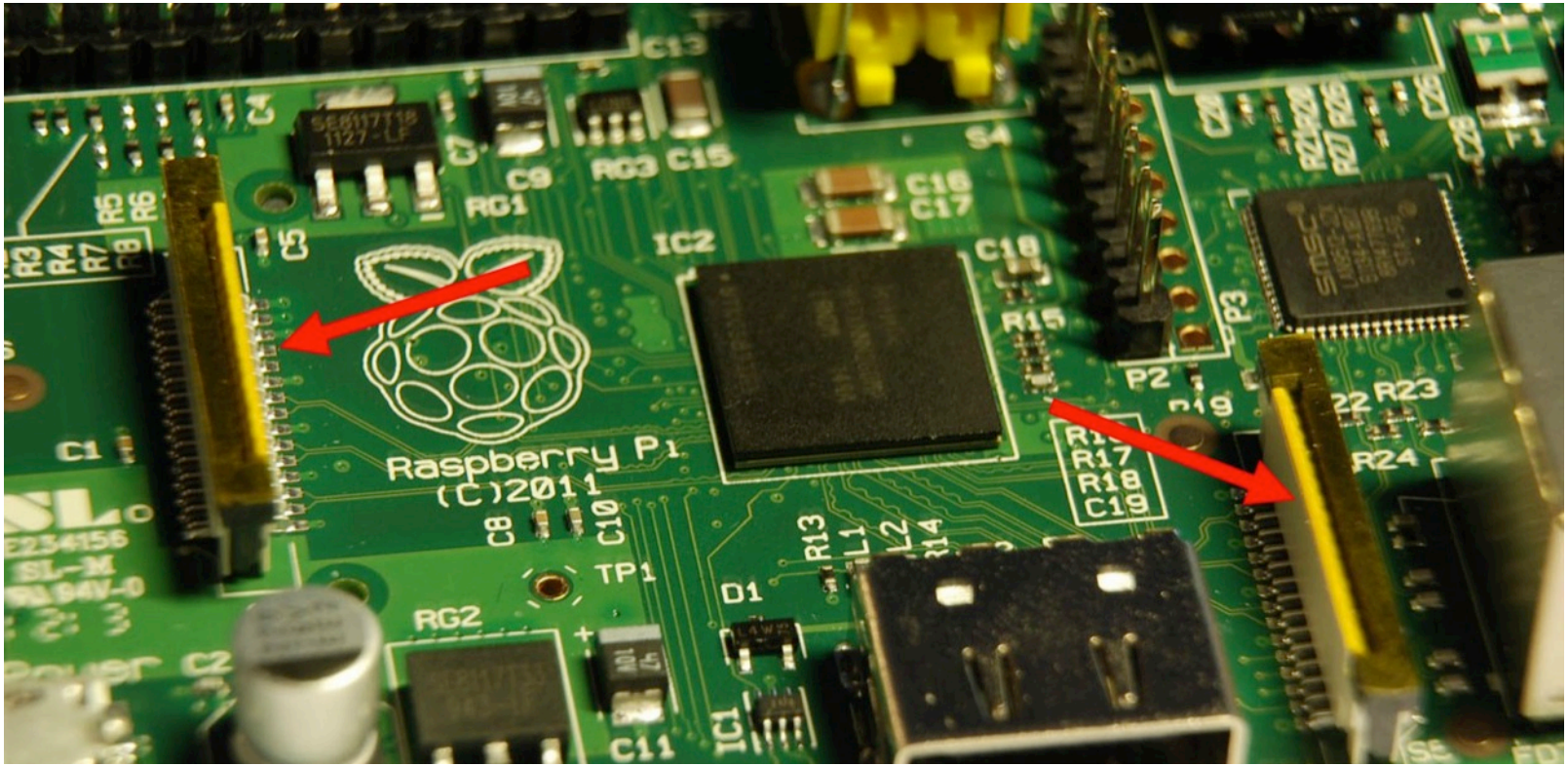


 http://elinux.org/File:Raspi-Model-AB-Mono-2-699x1024.png

I²C Hardware via Pi4J



And what are these for?



http://i.imgur.com/k0Puu.jpg

Chalkboard Electronics Touchscreen

- 10" or 7" Form Factor
- Connects via HDMI/USB
- Tested with JavaFX 8
- 10% Exclusive Discount:

  G1F0U796Z083

Chalkboard Electronics

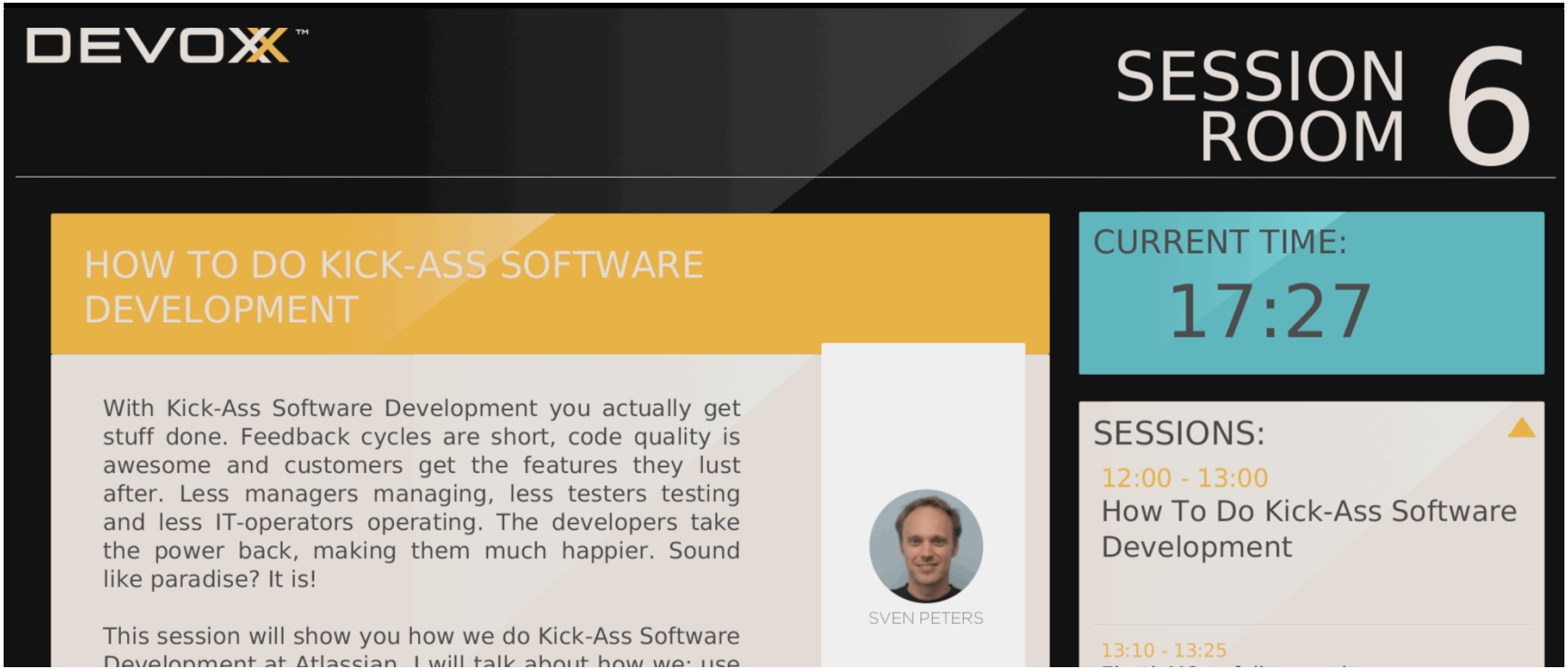
JavaFX on Raspberry Pi Gotchas

WebView

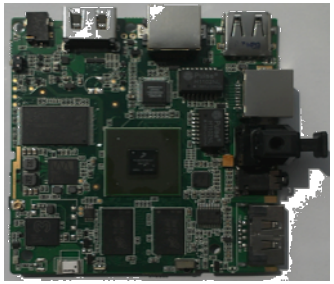
Media

And you can do cool stuff like this…

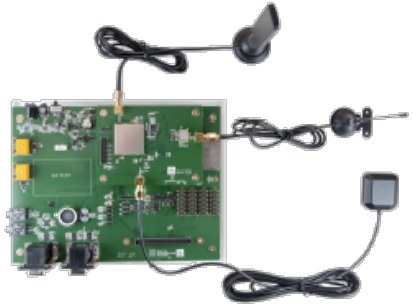
https://bitbucket.org/stephanj/tweetwall



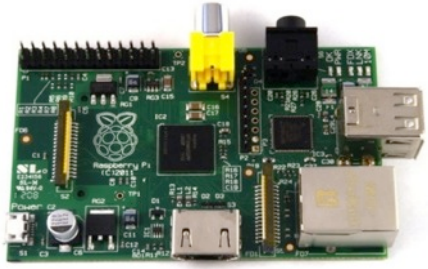
ARM



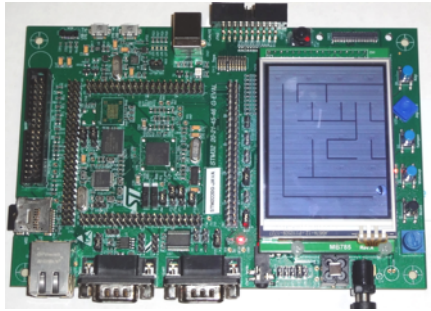
freescale™



QUALCOMM®



gemalto
security to be free



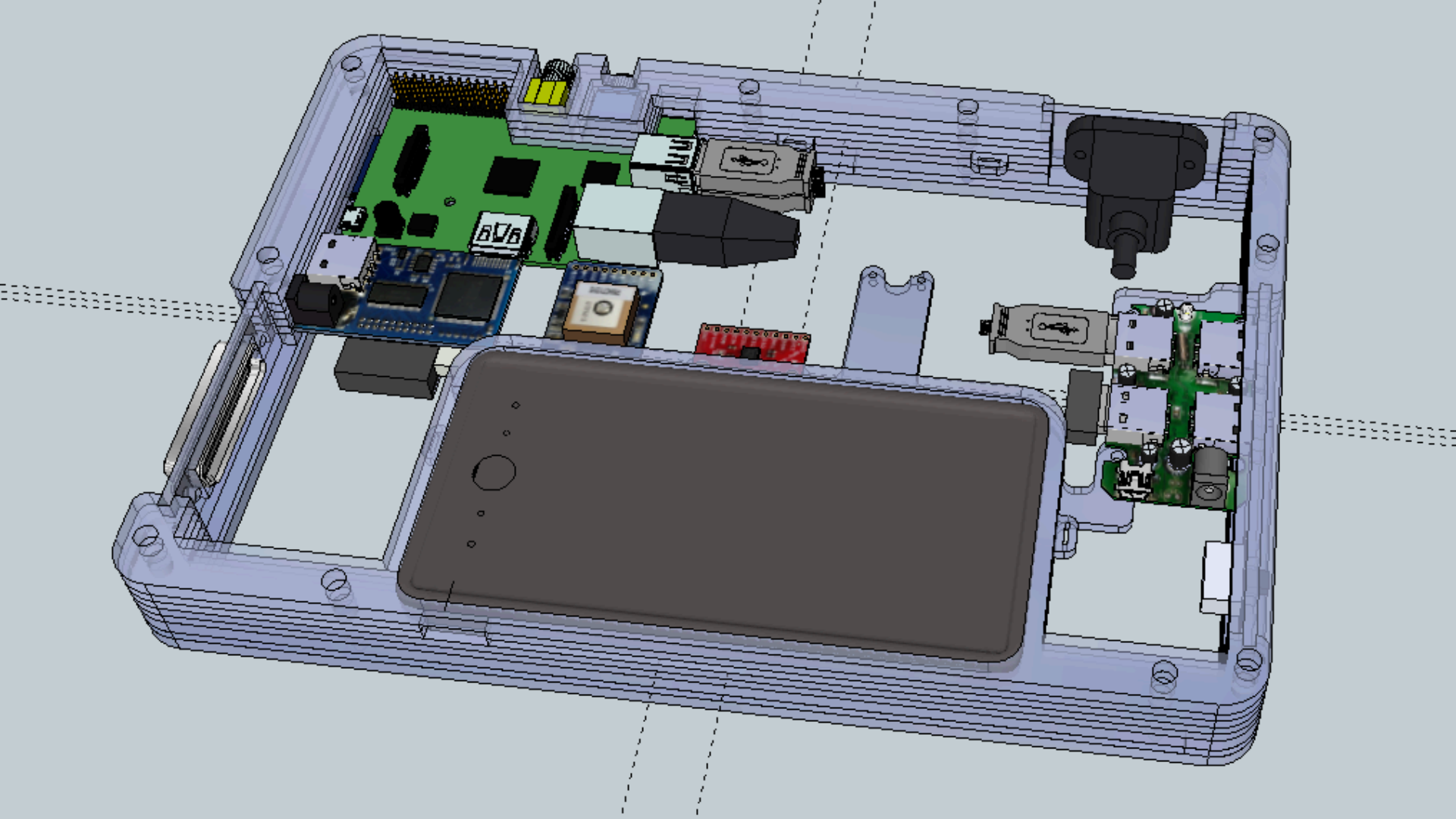
JavaOne™ ORACLE

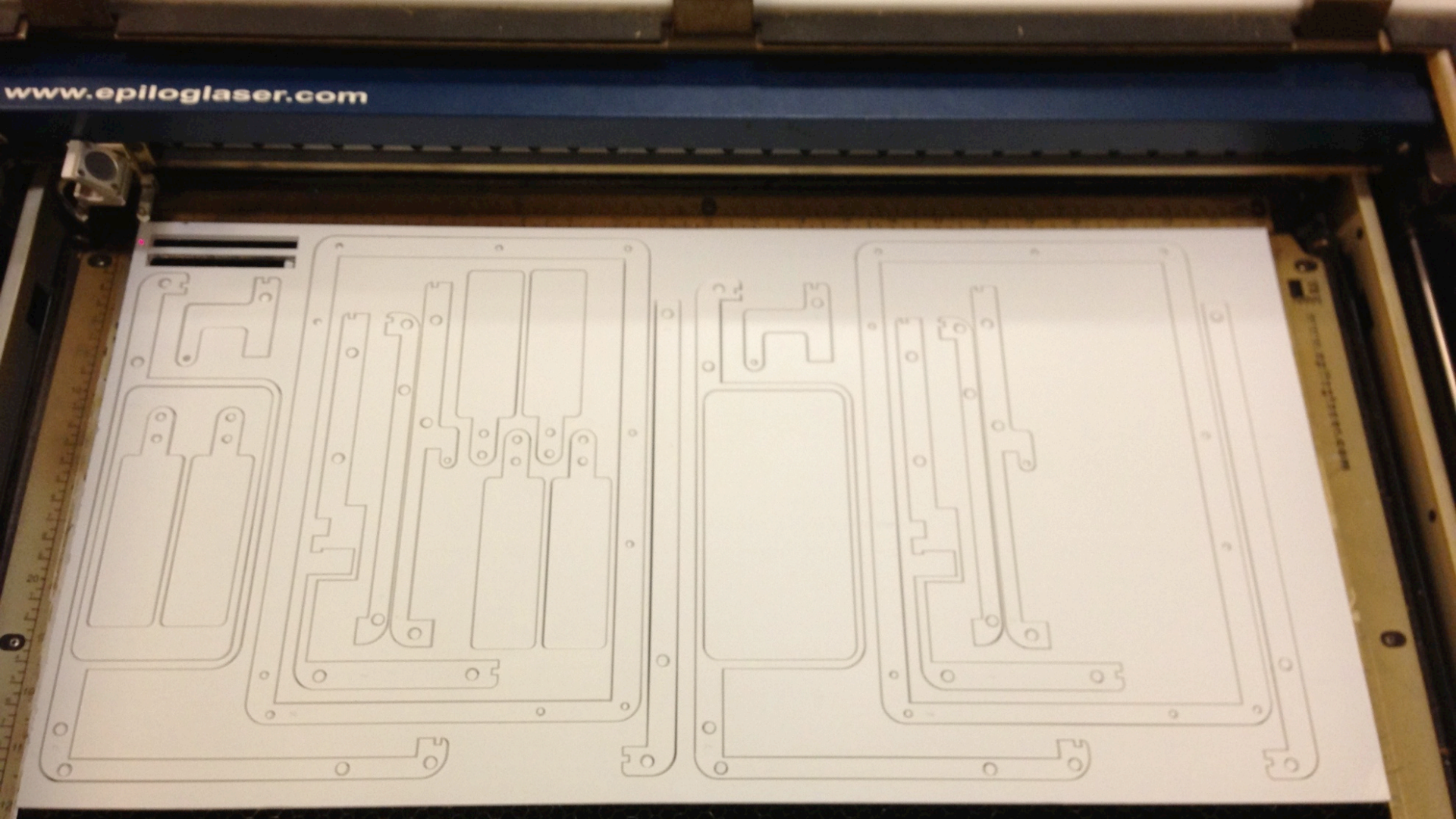
Example device that run Java SE Embedded

- Trimslice
- Plugcomputer
- Beaglebone Black
- Raspberry Pi
- HDMI Stick (white one)
- iMX6 Saberlite / Platform Kit
- iMX53 Tablet
- ST Micro Snowball

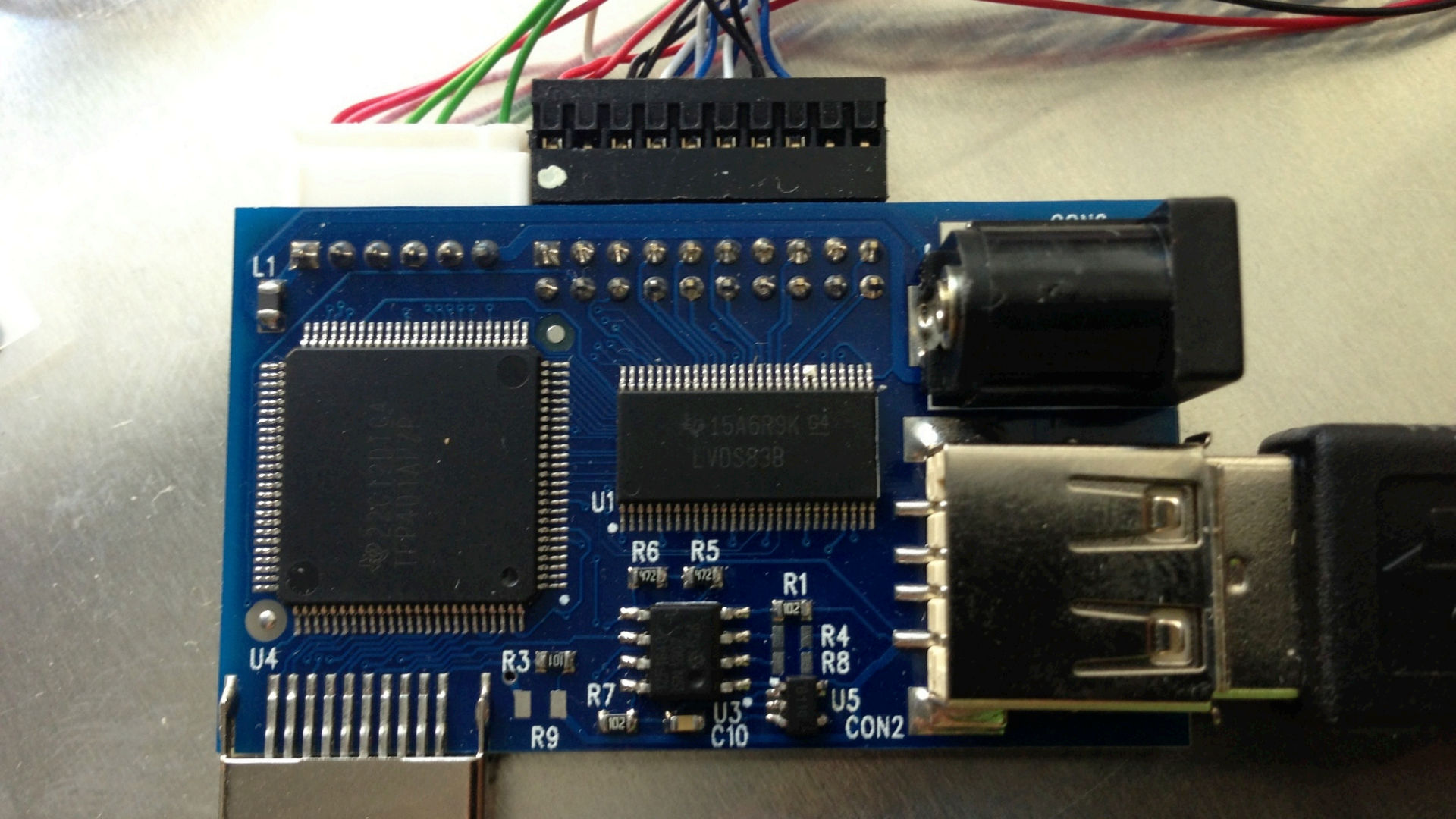
DukePad

http://j.mp/dukepad

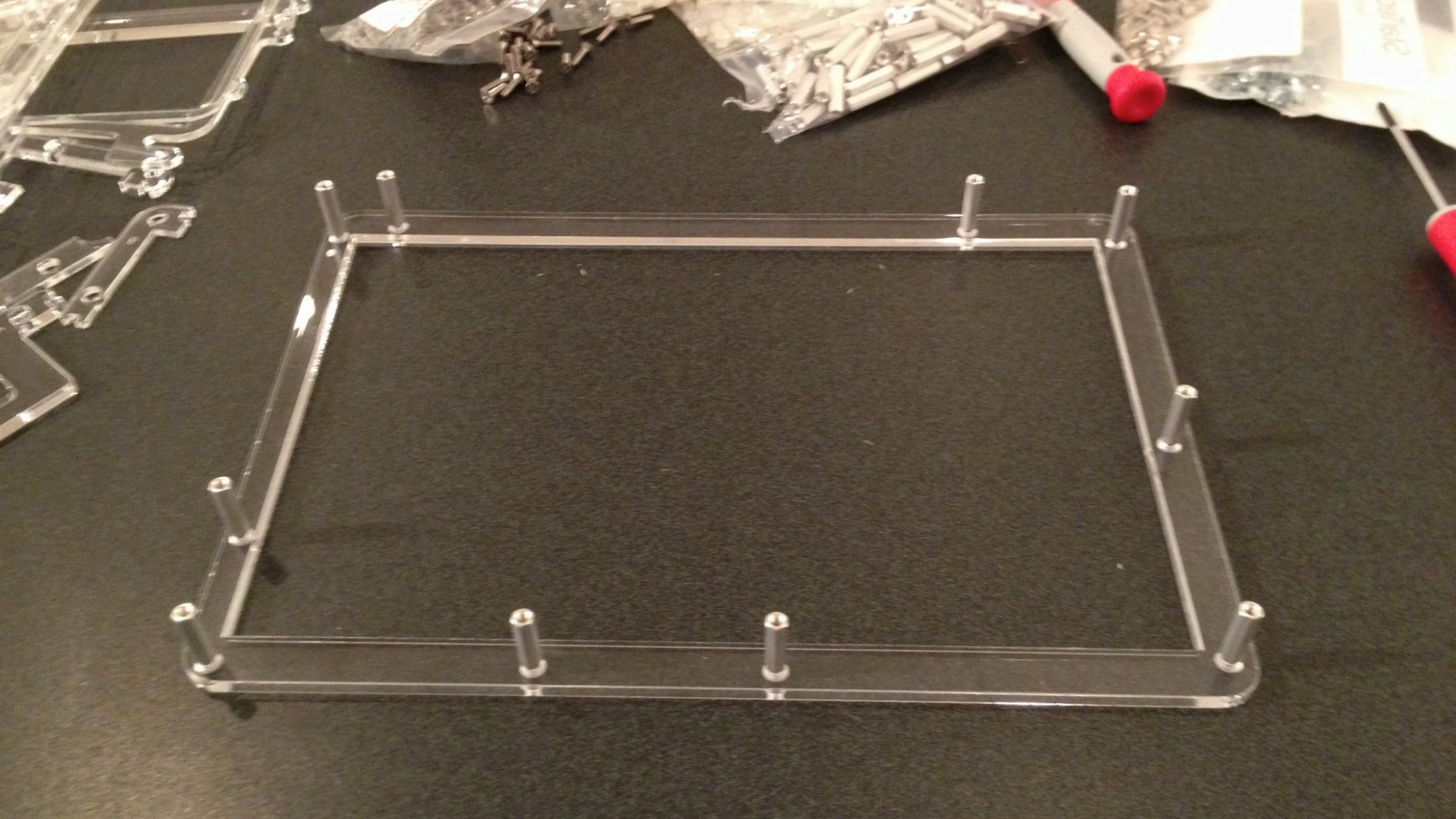


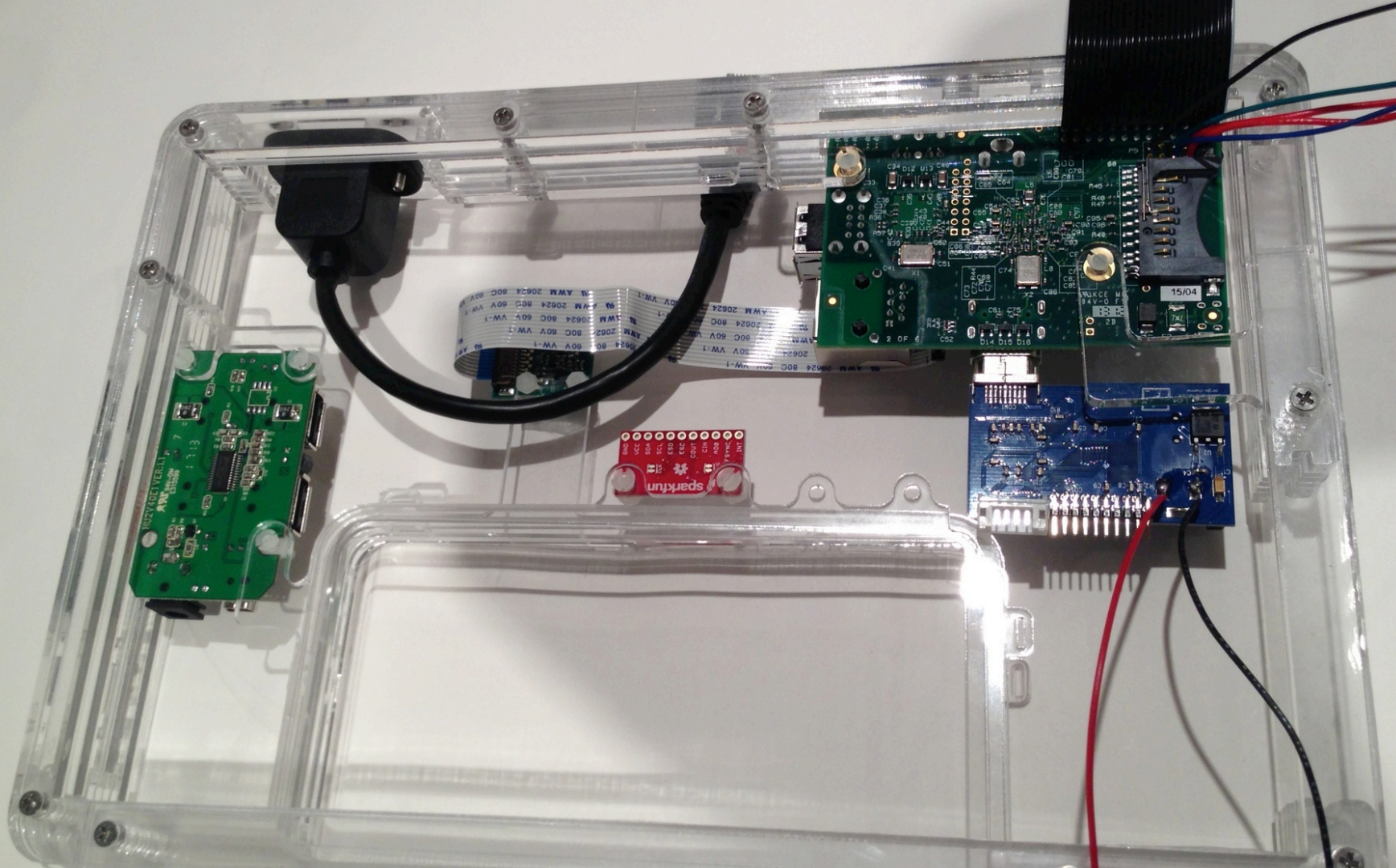

www.epiloglaser.com

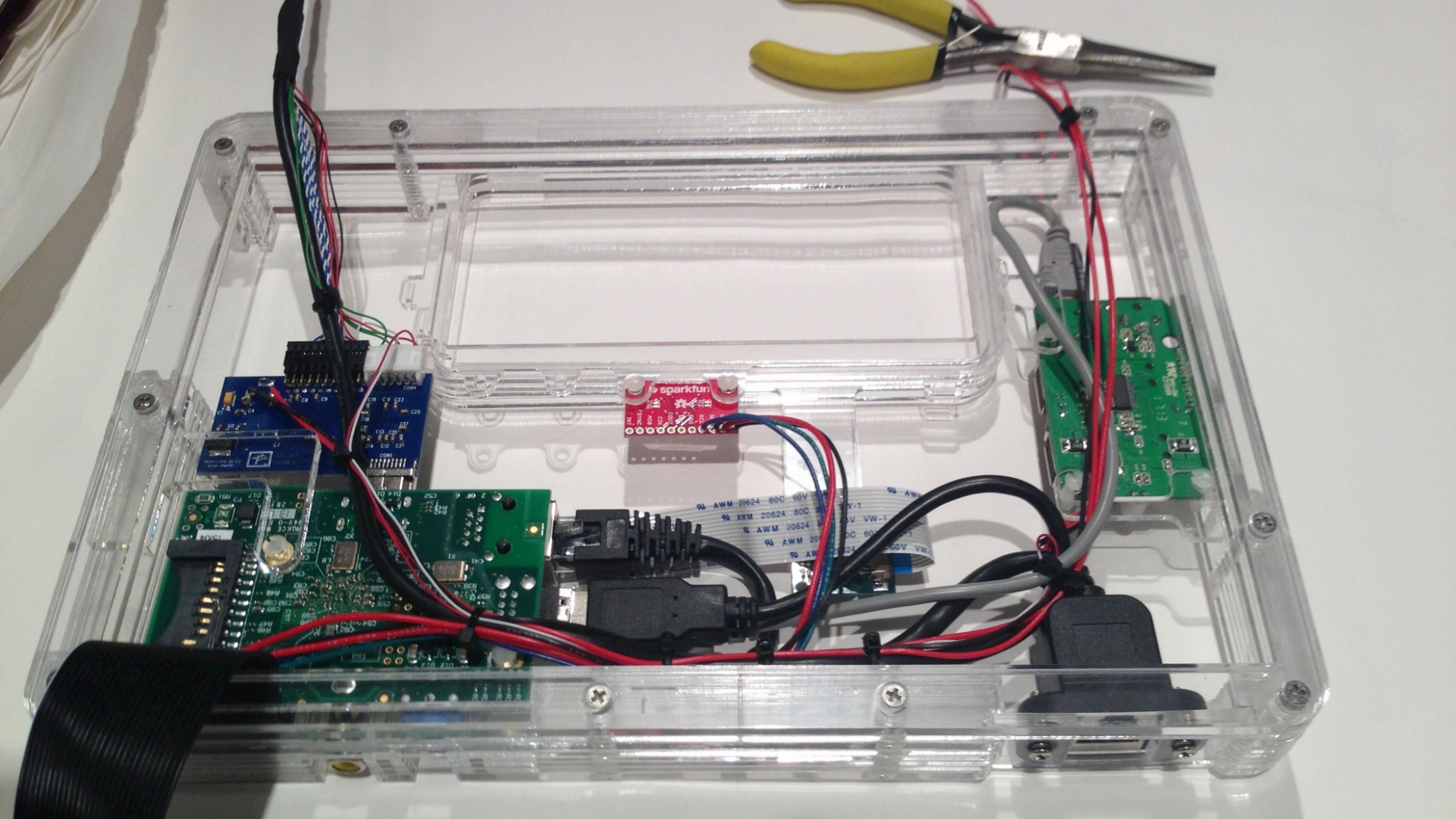
C17
CN1
C16
LED1
C7
C3
C1
R16
LED2
C8
C9
C13
C18
CN3


L1
U1
LVDS83B
R6
R5
R1
R4
R8
U4
R3
R7
U3
U5
R9
C10
CON2


ANKER®






sparkfun


Java™
Post-it

Getting Involved

http://j.mp/dukepad

OpenJDK Wiki

DukePad

Attachments: 7 • Added by Richard Bair, last edited by Richard Bair on Sep 21, 2013 (view change) • Labels None

Scratching the Surface with JavaFX

JavaFX multi-touch & 3D

- Touch Gestures
  - Swipe, Scroll, Rotate, Zoom
- Touch Event and Touch Points
- Introduction to JavaFX 3D
- Example app: ZenGuitar3D

Touch Gestures

Class GestureEvent

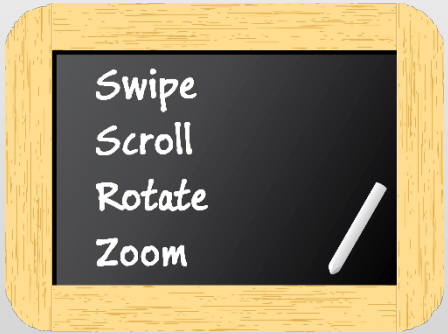
java.lang.Object
    java.util.EventObject
        javafx.event.Event
            javafx.scene.input.InputEvent
                javafx.scene.input.GestureEvent

All Implemented Interfaces:

java.io.Serializable, java.lang.Cloneable

Direct Known Subclasses:

RotateEvent, ScrollEvent, SwipeEvent, ZoomEvent

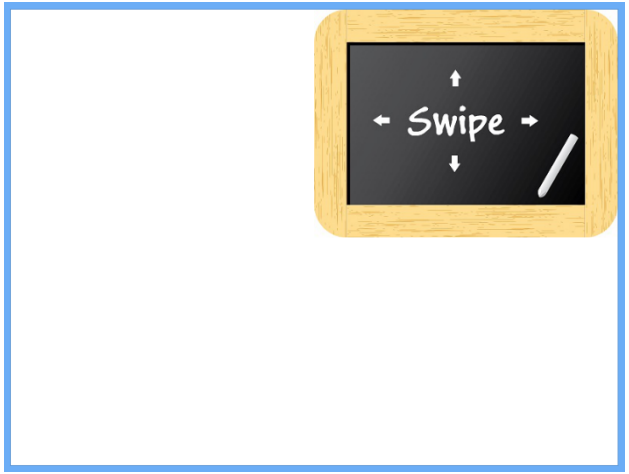


The Swipe Gesture

- Commonly a finger drag in one direction
- A single event is produced for the gesture
- May be left, right, up, or down

Handling the Swipe Gesture (`SwipeEvent`)

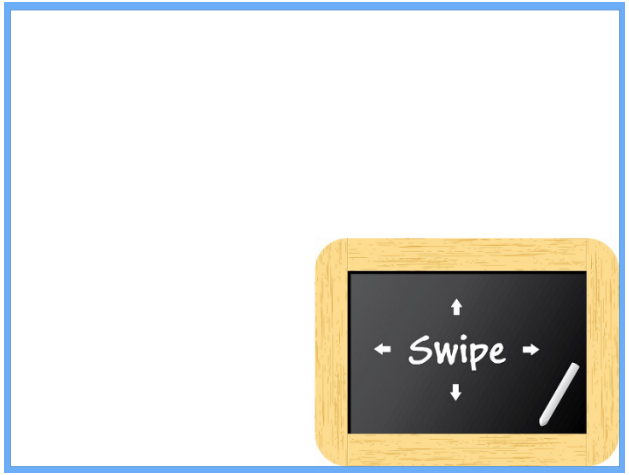

```
TranslateTransition rectTT =
  new TranslateTransition(dur, rect);

rect.setOnSwipeRight(e -> {
  if (rect.getTranslateX() <= 0) {
    rectTT.setByX(RECT_WIDTH);
    rectTT.setByY(0);
    rectTT.playFromStart();
  }
});
```



Note: Lambda expressions from JDK 8 are used here to simplify event handling

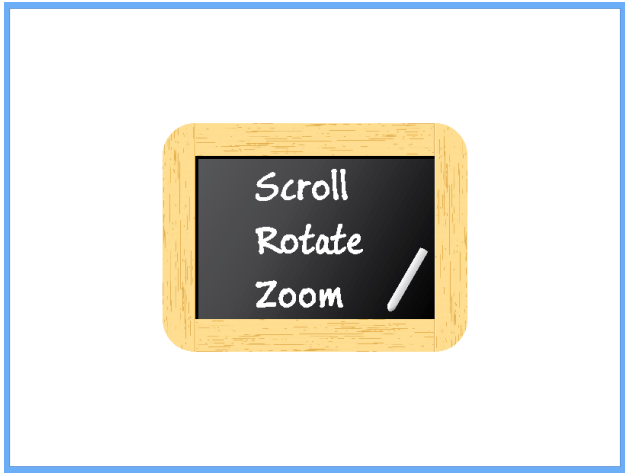
Handling the Swipe Gesture (`SwipeEvent`)


```
rect.setOnSwipeLeft(e -> {
  if (rect.getTranslateX() > 0) {
    rectTT.setByX(-RECT_WIDTH);
    rectTT.setByY(0);
    rectTT.playFromStart();
  }
});

rect.setOnSwipeDown(e -> {
  if (rect.getTranslateY() <= 0) {
    rectTT.setByX(0);
    rectTT.setByY(RECT_HEIGHT);
    rectTT.playFromStart();
  }
});
```



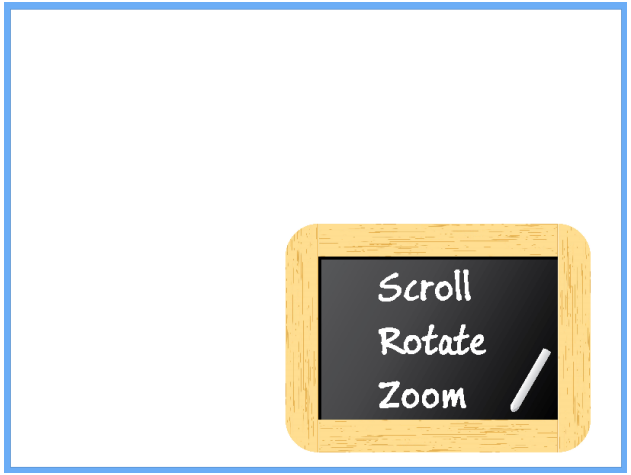
The Scroll Gesture

- User turns mouse wheel, drags finger on touch screen, etc.
- Scroll events are continuously generated, containing x/y position-related info
- Events are pixel-based or character/line-based
- If ***inertia*** is supported, scroll events may be generated after user quits scrolling



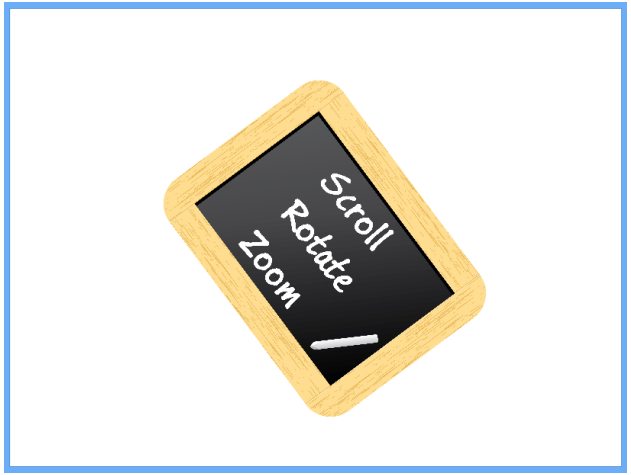
Handling the Scroll Gesture (`ScrollEvent`)


```
rect.setOnScrollStarted(e -> {
  curPosX = rect.getLayoutX();
  curPosY = rect.getLayoutY();
});

rect.setOnScroll(e -> {
  if (!e.isInertia()) {
    rect.setLayoutX(curPosX +
                    e.getTotalDeltaX());
    rect.setLayoutY(curPosY +
                    e.getTotalDeltaY());
  }
});
```



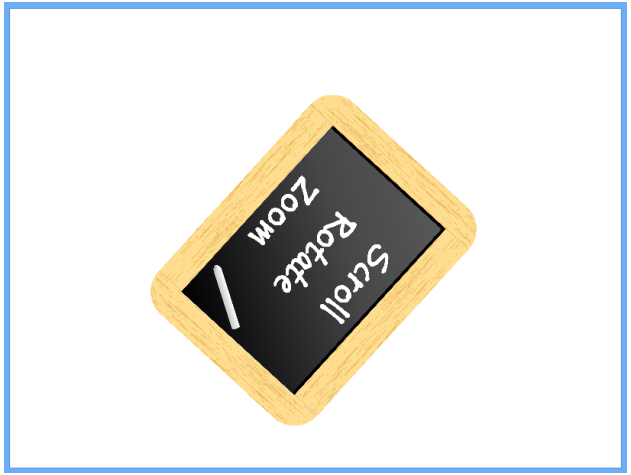
The Rotate Gesture

- User typically drags two fingers around each other
- Rotate events are continuously generated, containing angle-related info



Handling the Rotate Gesture (RotateEvent)


```
ImageView rect = new ImageView(gesturesImg);

rect.setOnRotationStarted(e -> {
  curRotateAngle = rect.getRotate();
});

rect.setOnRotate(e -> {
  rect.setRotate(e.getTotalAngle()
                 + curRotateAngle);
});
```

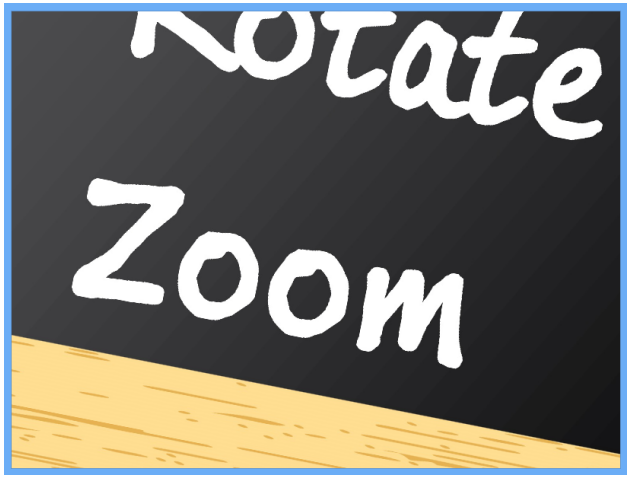


The Zoom Gesture

- User typically drags two fingers apart or closer together
- Zoom events are continuously generated, containing zoom factor-related info

Handling the Zoom Gesture (ZoomEvent)

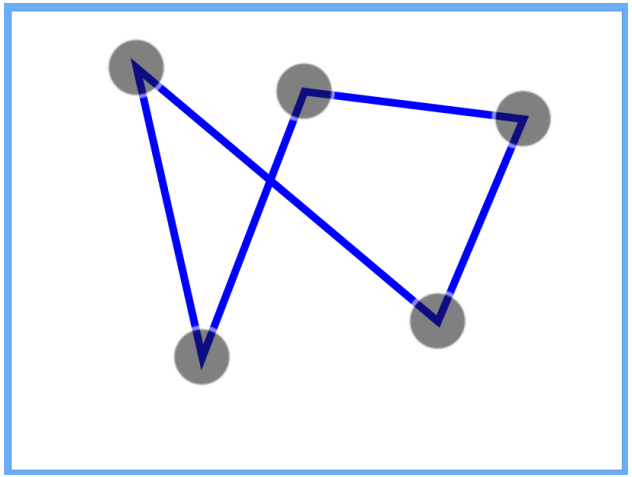

```
ImageView rect = new ImageView(gesturesImg);

rect.setOnZoomStarted(e -> {
  curZoomFactor = rect.getScaleX();
});

rect.setOnZoom(e -> {
  rect.setScaleX(e.getTotalZoomFactor()
                        * curZoomFactor);
  rect.setScaleY(e.getTotalZoomFactor()
                        * curZoomFactor);
});
```



Touch Event and Touch Points

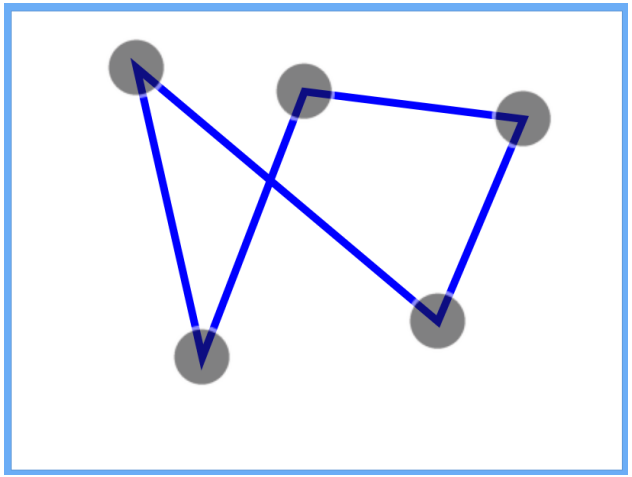
Handling Touch (`TouchEvent/TouchPoint`)

- A **`TouchEvent`** contains information about a touch, including:
  - Event type: Pressed, released, moved, or stationary
  - Touch points: The **`TouchPoint`** instances that represent each of the points that were touched
- Each **`TouchEvent`** has a unique ID to identify the events and touch points in a multi-touch action



Responding to Touch Events

```
    root.setOnTouchPressed(e -> updatePoly(e));
    root.setOnTouchReleased(e -> updatePoly(e));
    root.setOnTouchMoved(e -> updatePoly(e));
}

private void updatePoly(TouchEvent te) {
    poly.getPoints().clear();
    for (TouchPoint tp : te.getTouchPoints()) {
        poly.getPoints()
                .addAll(tp.getX(), tp.getY());
    }
}
```



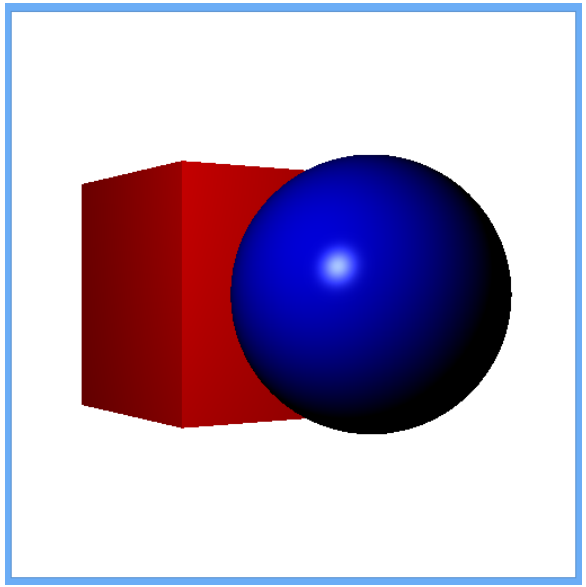
Introduction to JavaFX 3D

"JavaFX 3D gives you the ability to use ***3D geometry***, ***cameras***, and ***lights*** in JavaFX."

Jasper Potts
JavaFX Engineer,
Oracle Corporation

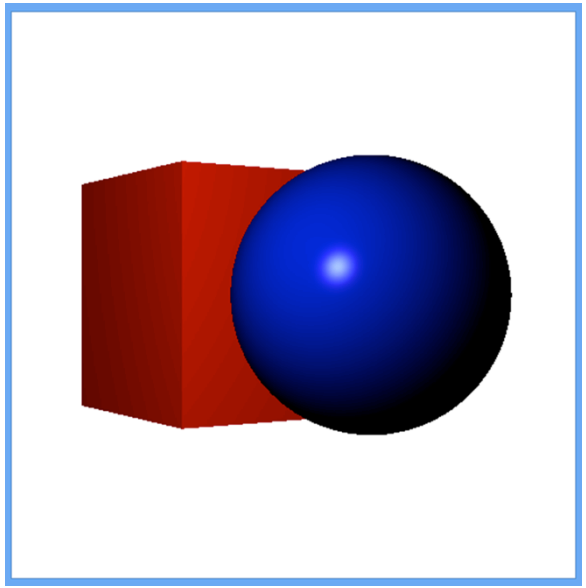
Mesh Geometry (3D Shapes)

- Predefined shapes
  - Box
  - Cylinder
  - Sphere
- User-defined shapes
  - Using `TriangleMesh` / `MeshView`



Creating Primitive Shapes and Materials

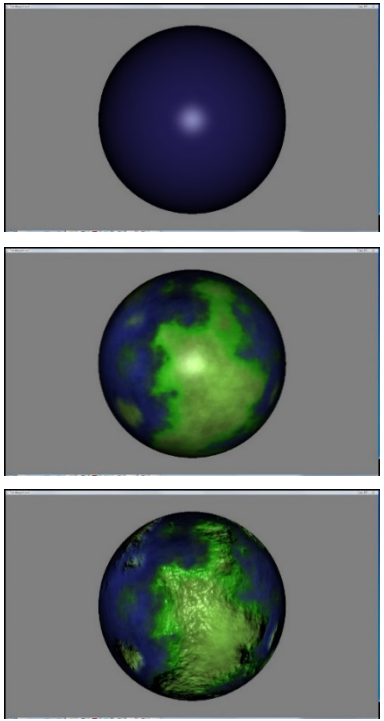
```
PhongMaterial mat =
    new PhongMaterial();
mat.setDiffuseColor(Color.BLUE);
mat.setSpecularColor(Color.LIGHTBLUE);

final Sphere blue = new Sphere(200);
blue.setMaterial(mat);
```



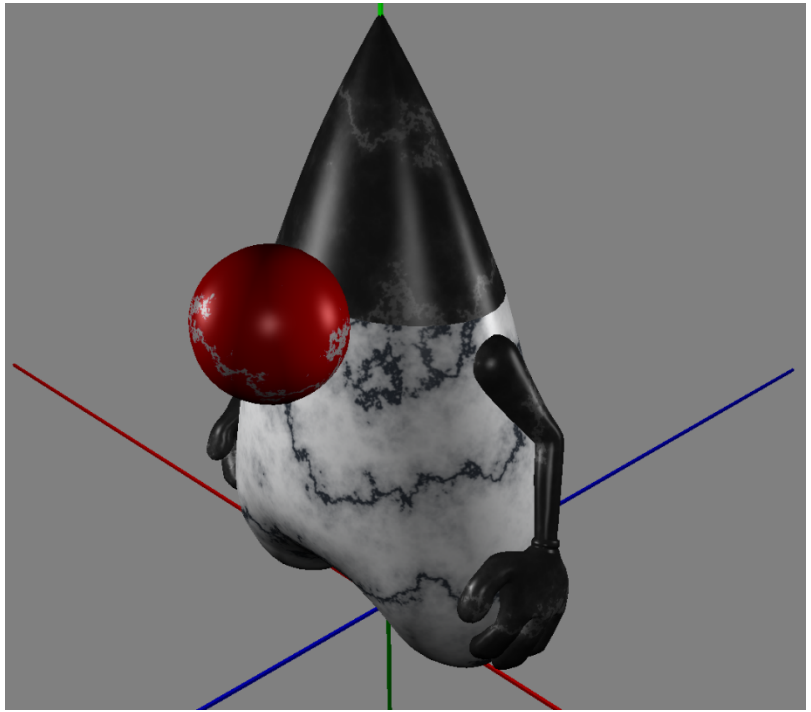
3D Materials and Textures

https://wikis.oracle.com/display/OpenJDK/3D+Features

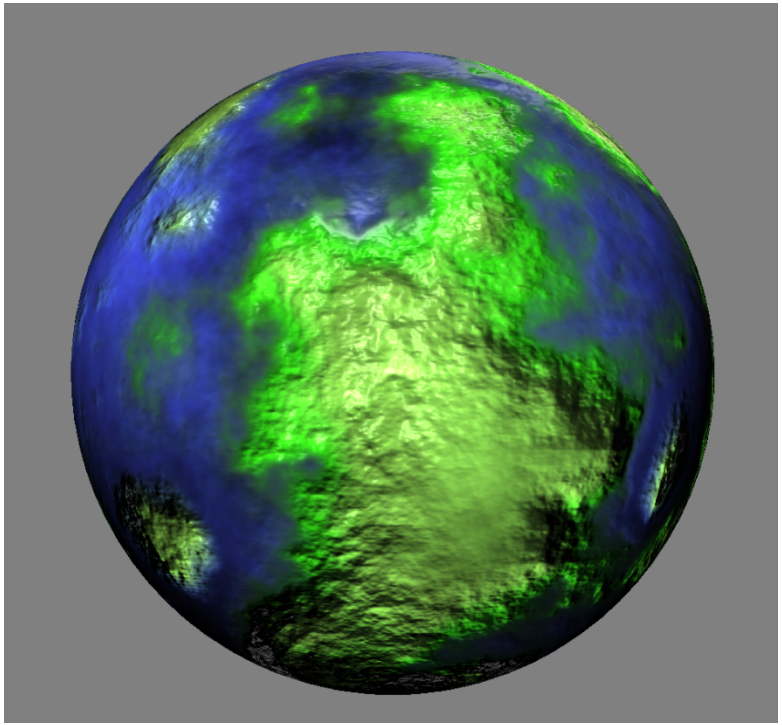
- **`PhongMaterial`** has these properties
  - Ambient color
  - Diffuse color, diffuse map
  - Specular color, specular map
  - Specular power
  - Bump map
  - Self-illumination map



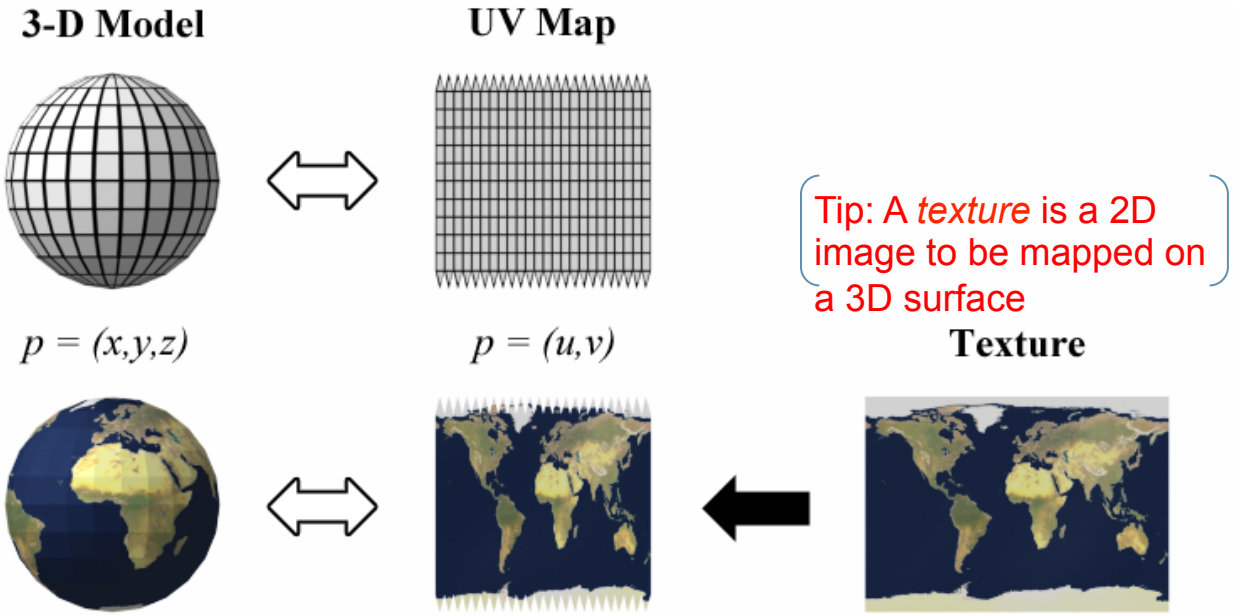
Duke's nose has a *Diffuse Map* texture



This planet has a *Bump Map* texture



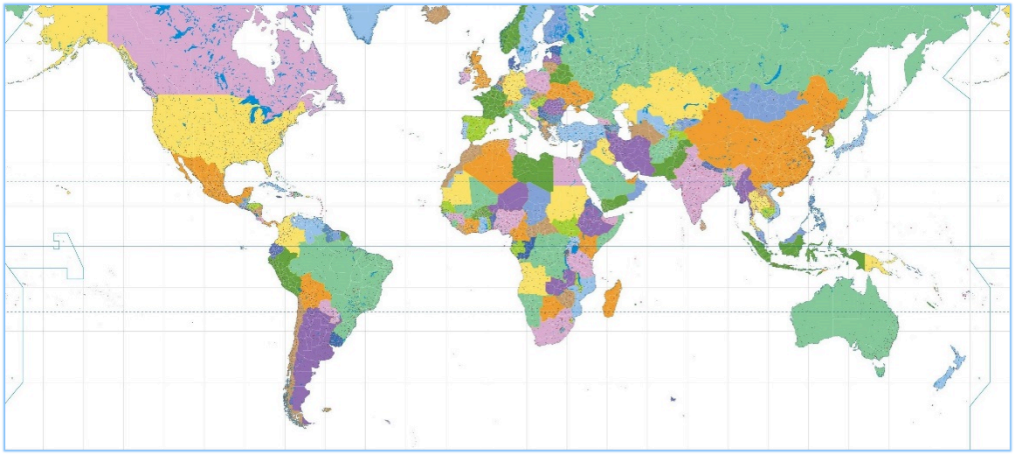
UV Mapping Textures to Shapes



Tip: A *texture* is a 2D image to be mapped on a 3D surface

Source: http://en.wikipedia.org/wiki/File:UVMapping.png

Placing a Texture on a Sphere

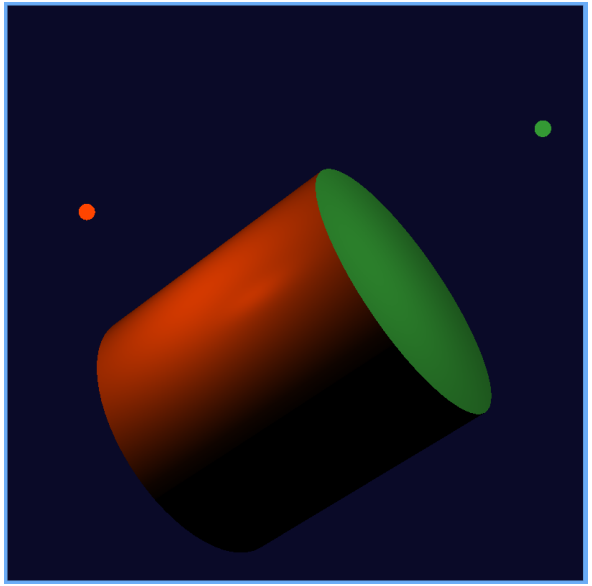


Placing a Texture on a Sphere

```
Material earthMaterial =
   new PhongMaterial(Color.TRANSPARENT,
                     diffuseMap, // Image
                     null, null, null);

final Sphere earth = new Sphere(400);
earth.setMaterial(earthMaterial);
```

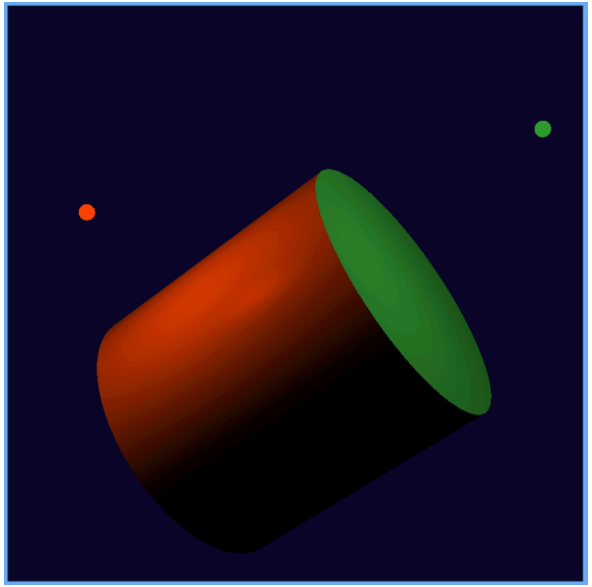
3D Lights

- Lights are nodes in the scene graph
  - `PointLight`
  - `AmbientLight`
- Default light provided if no active lights

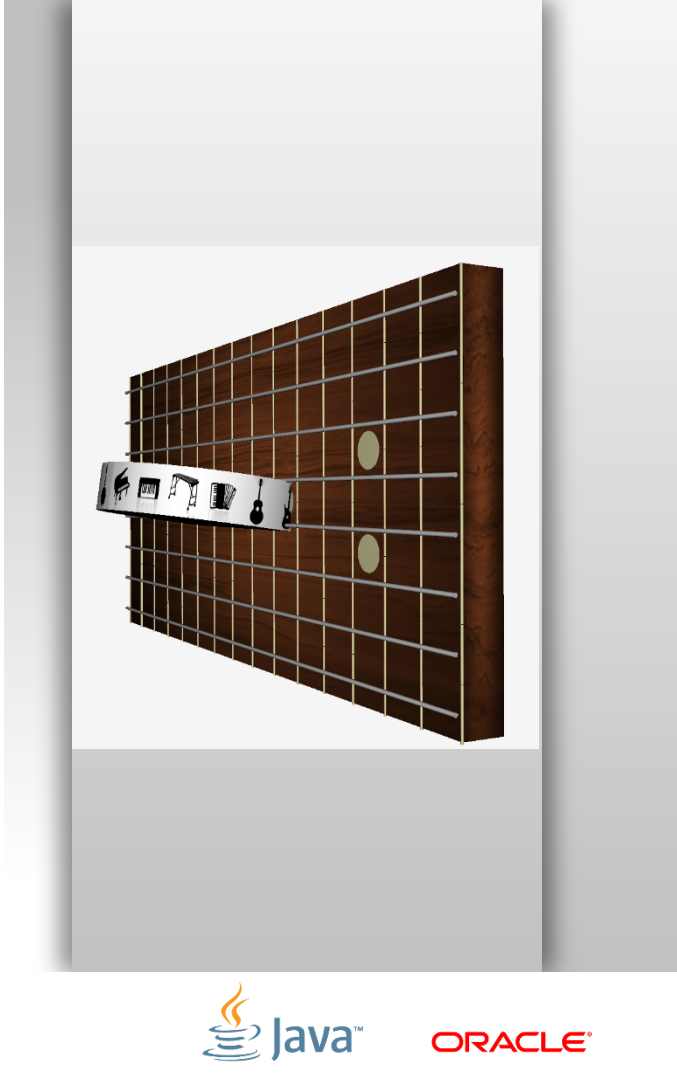


Lights, Camera, Action!

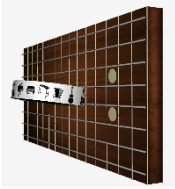
```
PointLight lightA =
  new PointLight(light1Color);
lightA.setTranslateX(700);
lightA.setTranslateY(200);
lightA.setTranslateZ(-200);

PerspectiveCamera camera =
  new PerspectiveCamera();
camera.setTranslateZ(-10);
```



Example multi-touch app: ZenGuitar3D

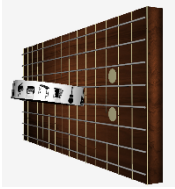


ZenGuitar3D Uses the JFugue5 Library

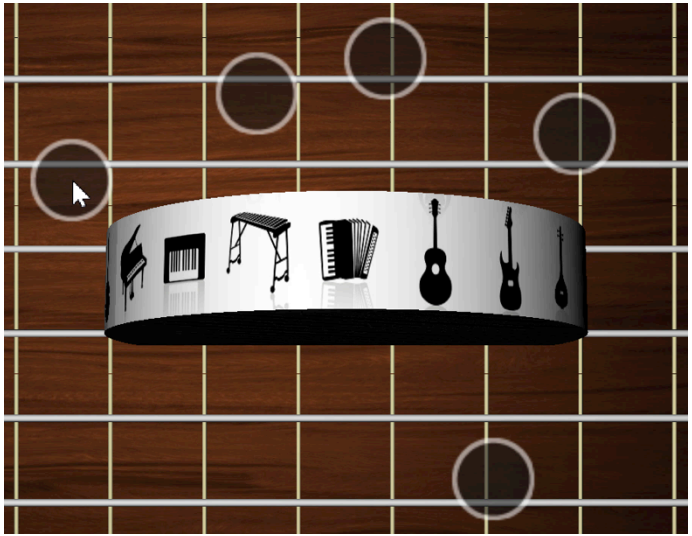


- An open-source Java API for programming music without the complexities of MIDI
- Developed by David Koelle
- Available at http://JFugue.org

Showing the Picker (`TouchEvent`)



```
setOnTouchPressed(te -> {
    if (te.getTouchCount() >= 5) {
        _zenGuitar3D.showMidiPicker(false);
    }
});


_showMidiPicker =
    new TranslateTransition(
        new Duration(2000),
        _midiPicker
    );
_showMidiPicker.setFromZ(400);
_showMidiPicker.setToZ(-100);
```

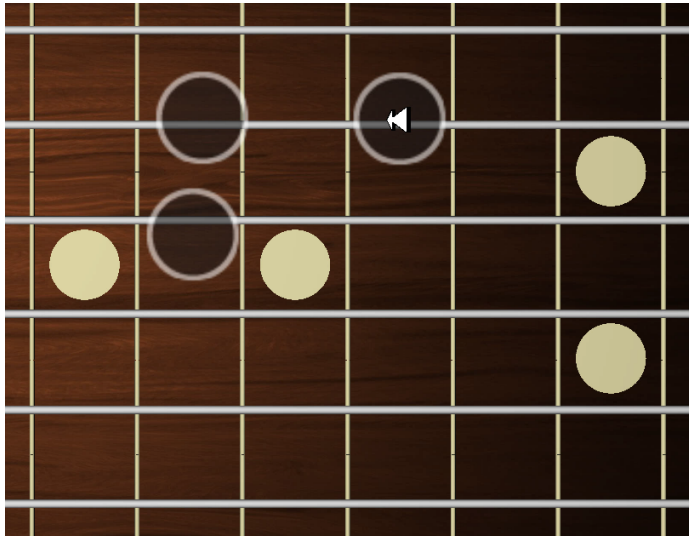


Rotating Instrument Picker with Scroll Gesture

```
_cylinder.setOnScrollStarted(e -> {
  _curRotateAngle =
    (_cylinder.getRotate() + 360) % 360;
});

_cylinder.setOnScroll(e -> {
  if (!e.isInertia()) {
    _cylinder.setRotate(
      _curRotateAngle - e.getTotalDeltaX() / 7
    );
  }
});
```

Playing Strings (TouchEvent / TouchPoint)

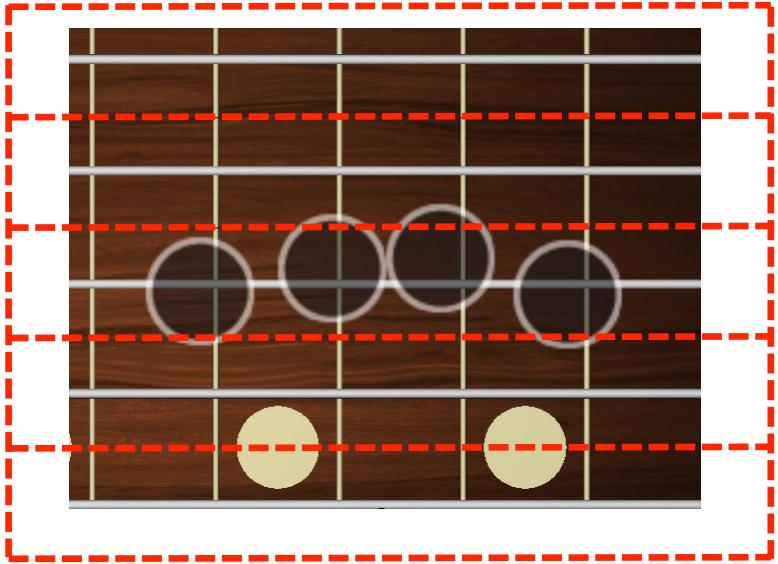
```
setOnTouchPressed(te -> {
  // Tapping / hammer-ons
});
setOnTouchReleased(te -> {
  // Removing finger
});
setOnTouchMoved(te -> {
  // Slides / bends
});
setOnTouchStationary(te -> {
  // Pull-offs
});

computeNoteValue(te.getTouchPoint().getX());
```



Switching Modes (`TouchPoint#belongsTo`)

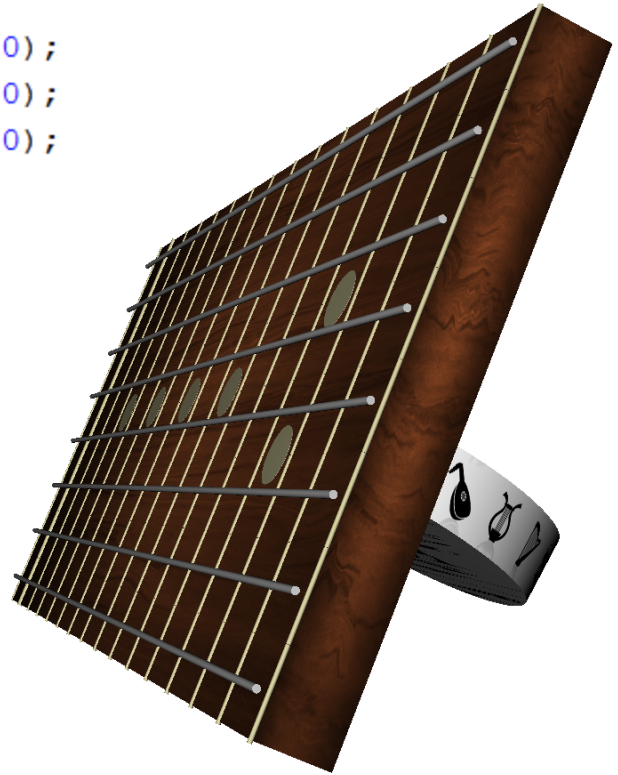
```
int numTouchPointsOnString(TouchEvent te) {
  int numTouchPoints = 0;
  for (TouchPoint tp : te.getTouchPoints()) {
    if (tp.belongsTo(this)) {
      numTouchPoints++;
    }
  }
  return numTouchPoints;
}
```

GuitarString3D instance



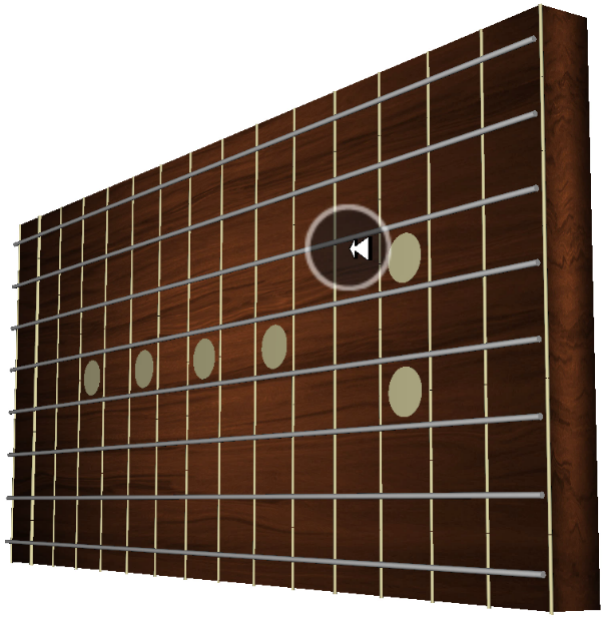
Setting up to Rotate on Three Axes

```
DoubleProperty _guitarAngleX = new SimpleDoubleProperty(0);
DoubleProperty _guitarAngleY = new SimpleDoubleProperty(0);
DoubleProperty _guitarAngleZ = new SimpleDoubleProperty(0);

Rotate _guitarRotateX;
Rotate _guitarRotateY;
Rotate _guitarRotateZ;

_guitar.getTransforms().setAll(
    _guitarRotateX = new Rotate(0, Rotate.X_AXIS),
    _guitarRotateY = new Rotate(0, Rotate.Y_AXIS),
    _guitarRotateZ = new Rotate(0, Rotate.Z_AXIS)
);

_guitarRotateX.angleProperty().bind(_guitarAngleX);
_guitarRotateY.angleProperty().bind(_guitarAngleY);
_guitarRotateZ.angleProperty().bind(_guitarAngleZ);
```



Using Scroll Gesture for X/Y Rotate

```
_guitar.setOnScroll(e -> {
  if (_muteMode) {
    if (!e.isInertia()) {
      _guitarAngleX.set(_guitarAngleX.get()
                        + e.getDeltaY() / 3);
      _guitarAngleY.set(_guitarAngleY.get()
                        - e.getDeltaX() / 3);
    }
  }
});
```

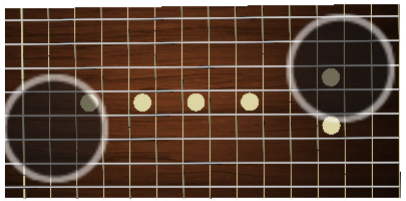


Using Rotate Gesture for Z Rotate

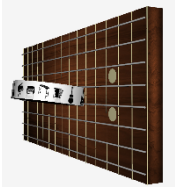
```
_guitar.getTransforms().setAll(
    _guitarRotateX = new Rotate(0, Rotate.X_AXIS),
    _guitarRotateY = new Rotate(0, Rotate.Y_AXIS),
    _guitarRotateZ = new Rotate(0, Rotate.Z_AXIS)
);

_guitarRotateZ.angleProperty().bind(_guitarAngleZ);

_guitar.setOnRotate(e -> {
    if (_muteMode) {
        _guitarAngleZ.set(_guitarAngleZ.get()
                          + e.getAngle());
    }
});
```

Using Zoom Gesture for Scaling

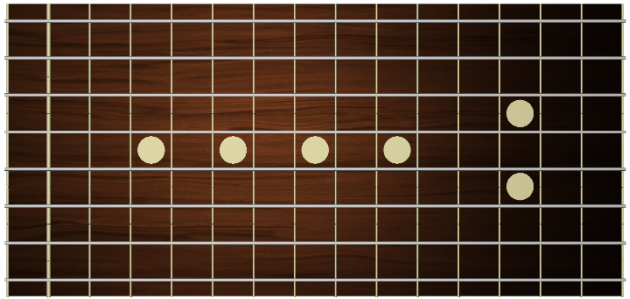
```
_guitar.setOnZoomStarted(e -> {
  _curZoomFactor = _guitar.getScaleX();
});

_guitar.setOnZoom(e -> {
  _guitar.setScaleX(e.getTotalZoomFactor() * _curZoomFactor);
  _guitar.setScaleY(e.getTotalZoomFactor() * _curZoomFactor);
  _guitar.setScaleZ(e.getTotalZoomFactor() * _curZoomFactor);
  if (_guitar.getScaleX() < .25) {
    goHomePos();
  }
});
```



Using `Timeline` to Transform to Home Position



```
_goHomeAnim = new Timeline(
  new KeyFrame(
    new Duration(1000),
    new KeyValue(_guitarAngleX, 0),
    new KeyValue(_guitarAngleY, 0),
    new KeyValue(_guitarAngleZ, 0),
    new KeyValue(_guitar.scaleXProperty(), 1.0),
    new KeyValue(_guitar.scaleYProperty(), 1.0),
    new KeyValue(_guitar.scaleZProperty(), 1.0),
    new KeyValue(_guitar.translateXProperty(),
                                STRING_WIDTH / 2),
    new KeyValue(_guitar.translateYProperty(),
                                 NECK_HEIGHT / 2),
    new KeyValue(_guitar.translateZProperty(),
                                GUITAR_INITIAL_Z)
  )
);
```



RoboVM

JavaFX on iOS

Twisted Flowers
Margarita Leskova

Candy Popper
Margarita Leskova

GDX Super Jumper
SpamDrain AB

Formula Car Game Premium for iPh...
Matti Vilola

Gravity Robot
Nemanja Komar

AntiVirus -The super duper accur...
Vision90

Dark Night Avenger : Magic Ride
CreatioSoft Solutions Private Limited

Space Bubble Shooter
Tomasz Kucza

123 Kids Fun Christmas Tree
RosMedia

Oh My Goat
CremaGames S.L.

Gesto: Levels
Huxi Games

Ichigu
Mehmet Atas

Tiny Hope Free
Blyts

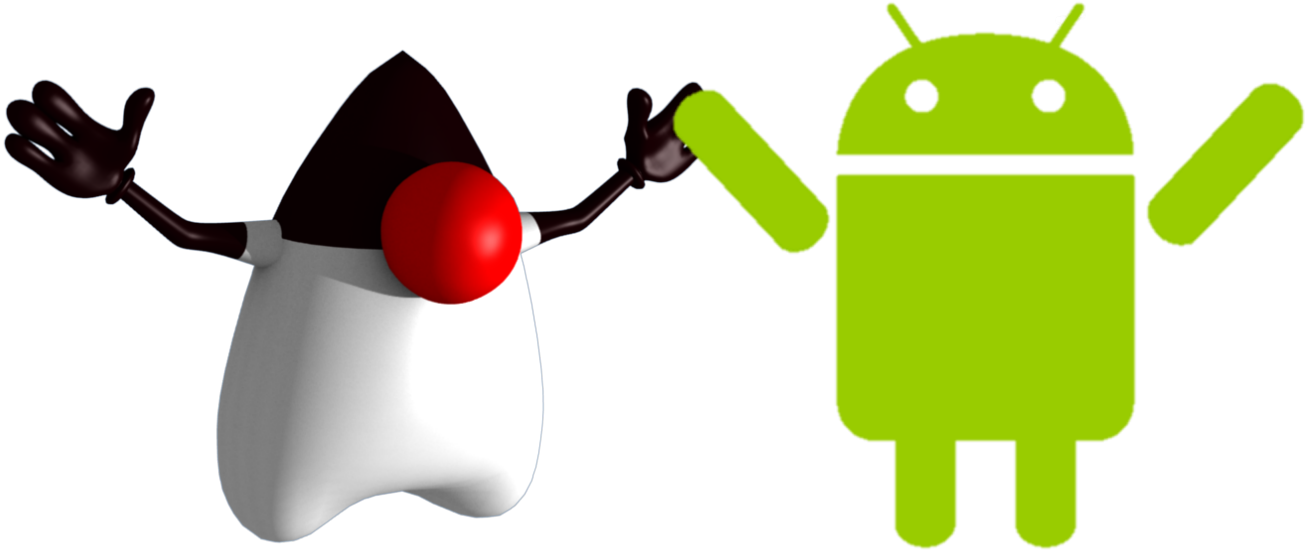
NagiQ 2: Treasure Hunt
Jose Brito

Max the Gold Hunter Plus Free
Joao Ignacio da Silva Neto

Dark Night the Game
Joao Ignacio da Silva Neto

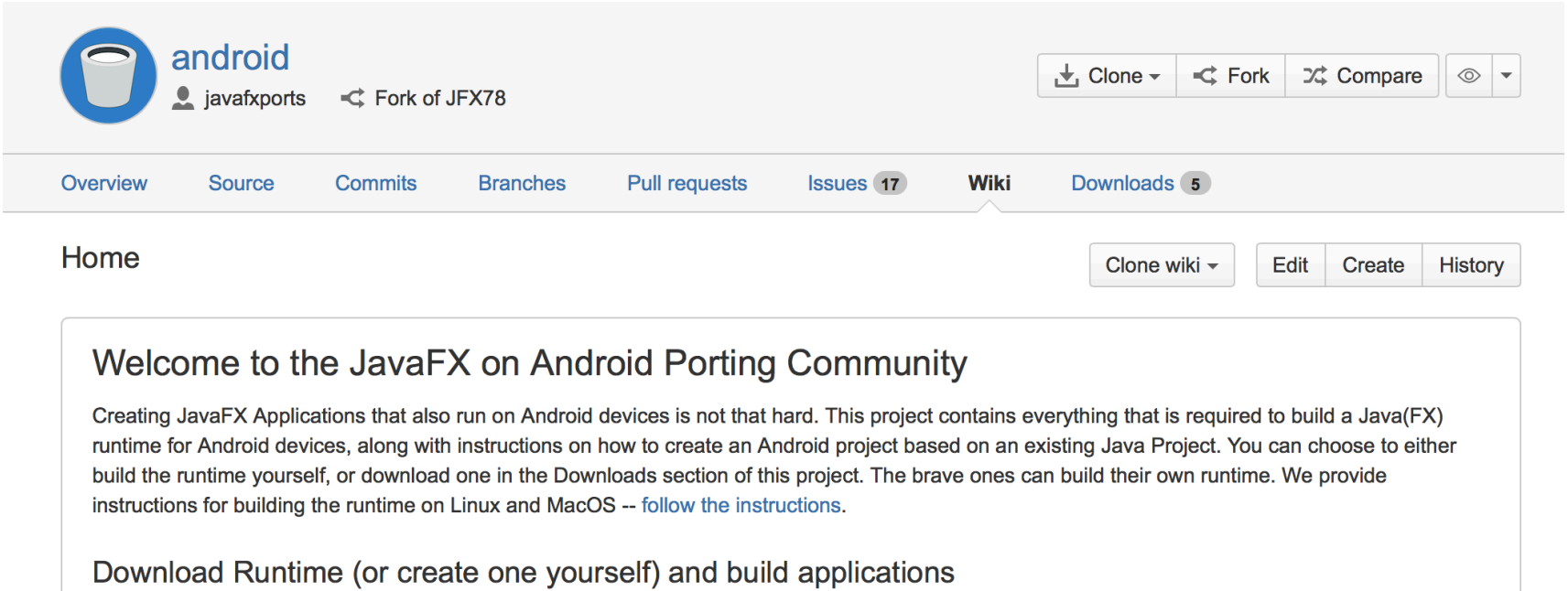
JavaFX on Android

It is about time!



Open Source Effort

https://bitbucket.org/javafxports/android/wiki/Home



MemeQuest: End-to-end JavaFX

- Animations, Images, Binding, Controls on the Client
- DataFX (http://www.javafxdata.org) for communication with DaliCloud back-end
- User Interface: We need to stop Johan from doing User Interfaces. Who wants to help?
- Code + downloads at http://bitbucket.org/lodgon/memequestfx

JavaOne™

Stephen Chin
tweet: @steveonjava
blog: http://steveonjava.com

NightHacking Tour

Real Geeks
Live Hacking

nighthacking.com

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.