JavaOne™

ORACLE®

# Creating Our Robot Overlords

## Autonomous Drone Development with Java and IoT

CREATE
THE
FUTURE

Mark Heckler / James Weaver
Software Engineer / Java Ambassador
Oracle

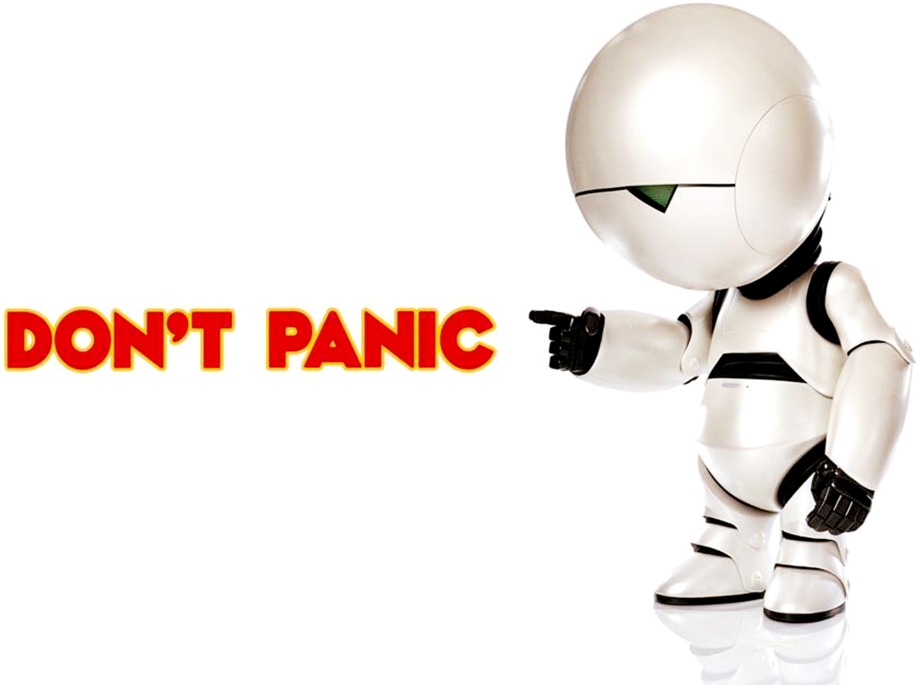
Sean Phillips
Software Engineer
Ai Solutions

# Program Agenda

1. Achieving autonomous flight (and tearing up the house)
2. Anatomy of a quadcopter
3. Raspberry Pi for brains
4. Leveraging IoT concepts and tools
5. 3D flight simulation
6. Next step: Self-aware drone (implementing a control loop)

# Achieving Autonomous Flight!

- First things first
  - Choosing a drone
  - Finding or writing a foundational library
  - Determining level of autonomous ops
- Equipment list
  - Drone
  - Brain
  - Power
- Making it work
  - And **this** is where the story gets really interesting



# Achieving Autonomous Flight!
# Making the Tough Choices

- Which drone?
  - Published API
  - Community
  - Price of equipment
  - **Parrot AR.Drone 2.0 (Parrot.com)**
- Which library?
  - Capability
  - Reliability
  - Responsibility
  - **Parrots On Java/Parroteer (ParrotsOnJava.com)**

# Achieving Autonomous Flight!
# Assembling the Pieces

- Equipment list
  - Parrot AR.Drone 2.0
  - Brain
    - Raspberry Pi Model B with case
    - Two (2) Edimax EW-7811un wifi adapters
    - 16G Class 10 SD card
  - Power
    - dodocool 2600 mAh mini power bank/charger
    - Cablejive microStubz extra short USB to micro USB cable
  - Anything else?
    - Duct/gaffer’s tape! (Just kidding, we used Velcro)

# Achieving Autonomous Flight!
# Configuring the Positronic Brain

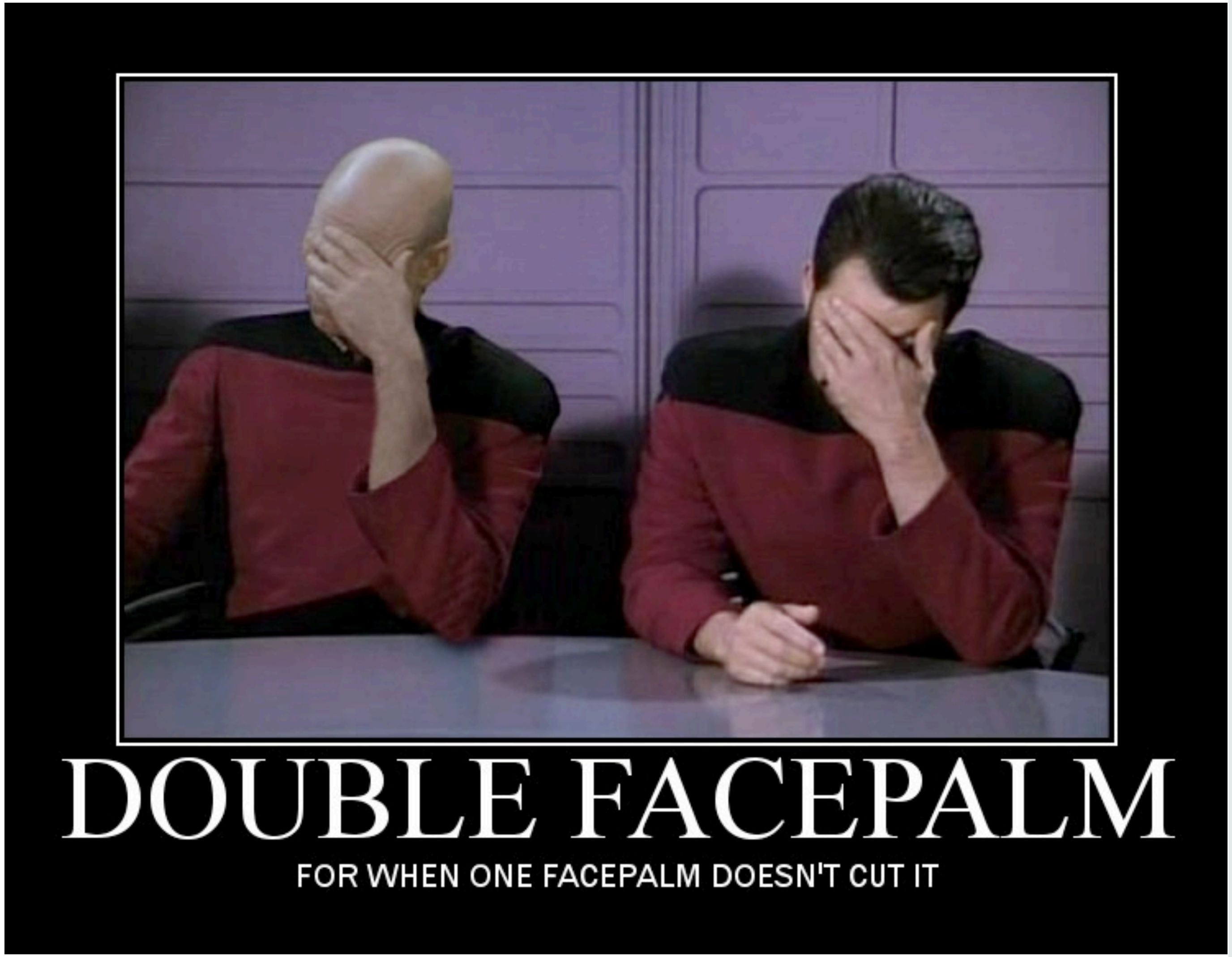
- Central piece of the puzzle
- Configure one wifi adapter to connect as a client to the drone
- Configure other with Pi running
  - Wireless Access Point
  - DHCP server
- More details in a bit…

# Achieving Autonomous Flight!
# Powering the Positronic Brain

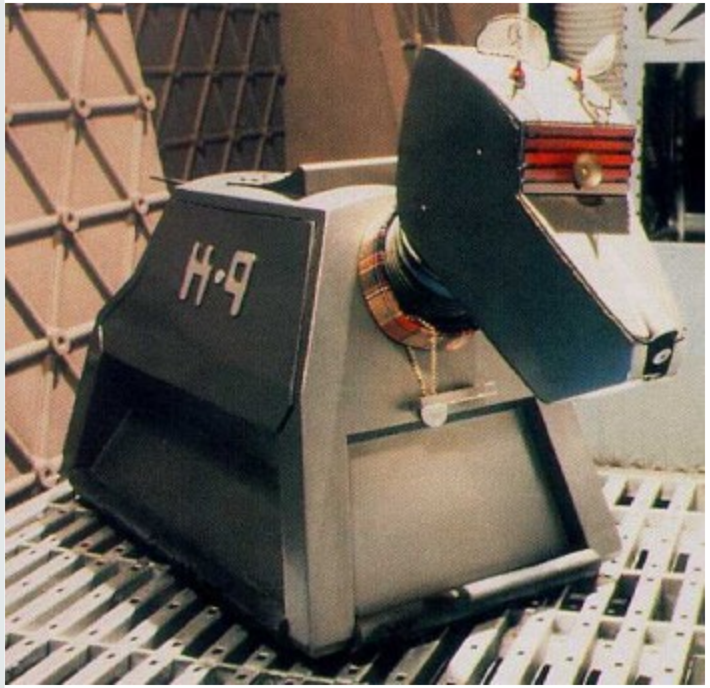
- No straightforward means of powering other devices from drone
- How to fix?
- Add a power source!

# Achieving Autonomous Flight!
# Making it Work

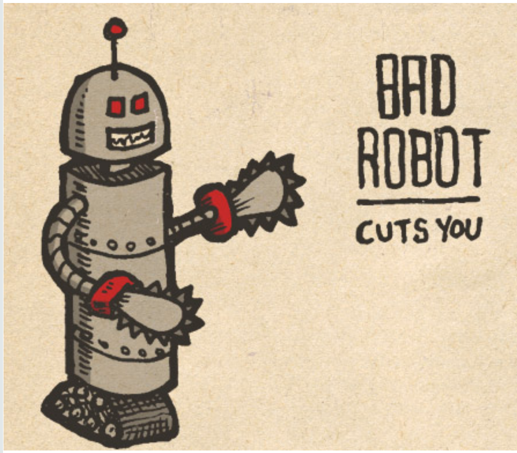
- Didn't anticipate many issues
- In hindsight, that was just silly
- Firmware challenges
- API challenges
- Equipment challenges
- Sensing a pattern?

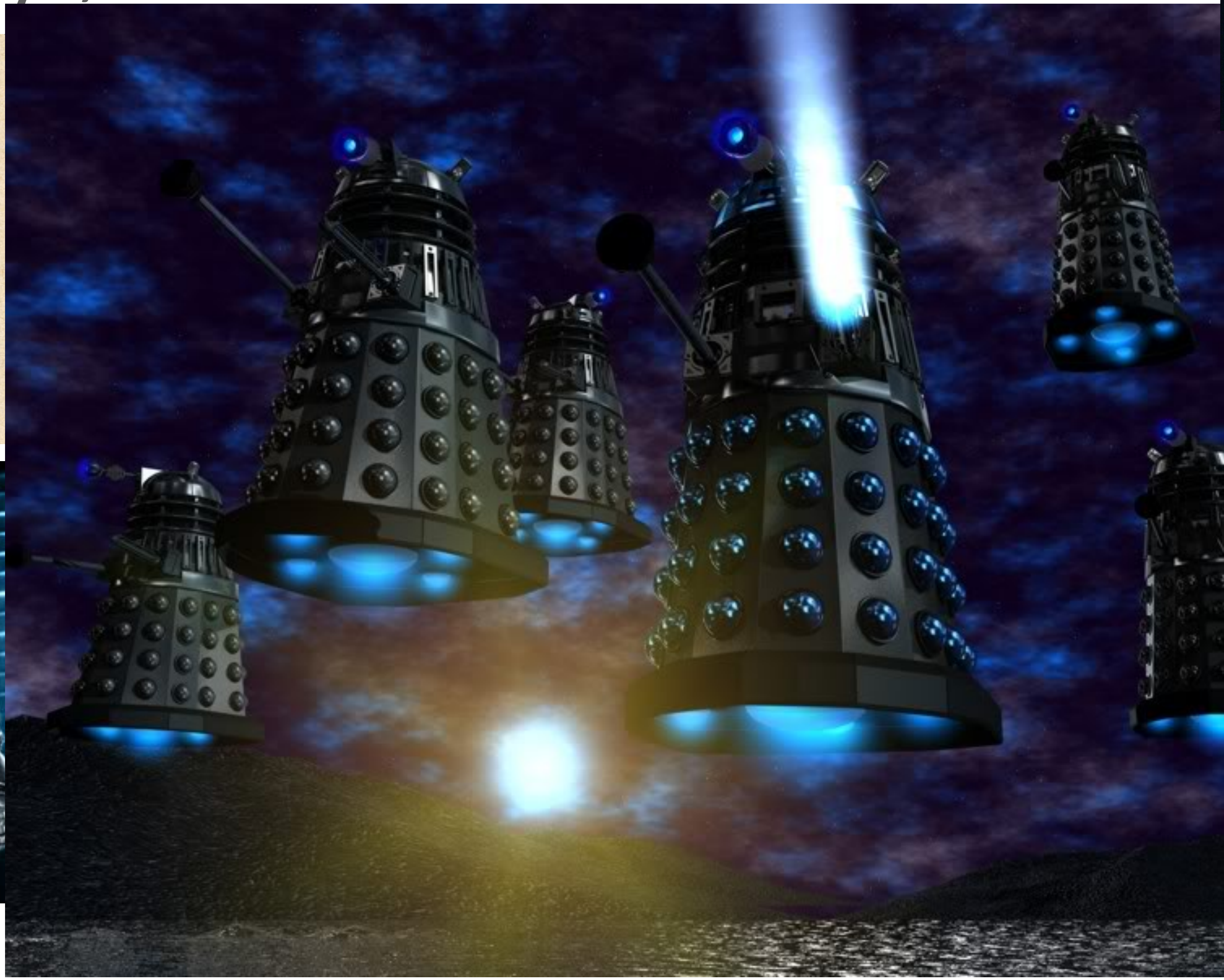


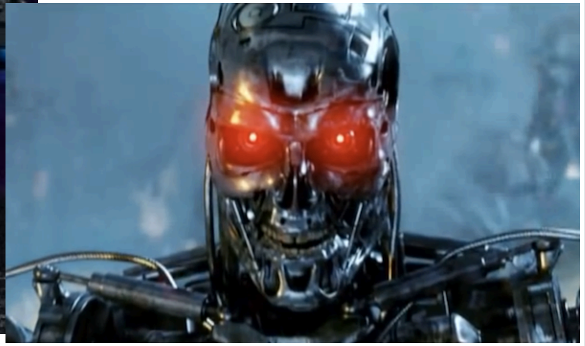
# Achieving Autonomous Flight!
(Unrealistic) Expectations



# Achieving Autonomous Flight!
# In some ways, it’s more like…



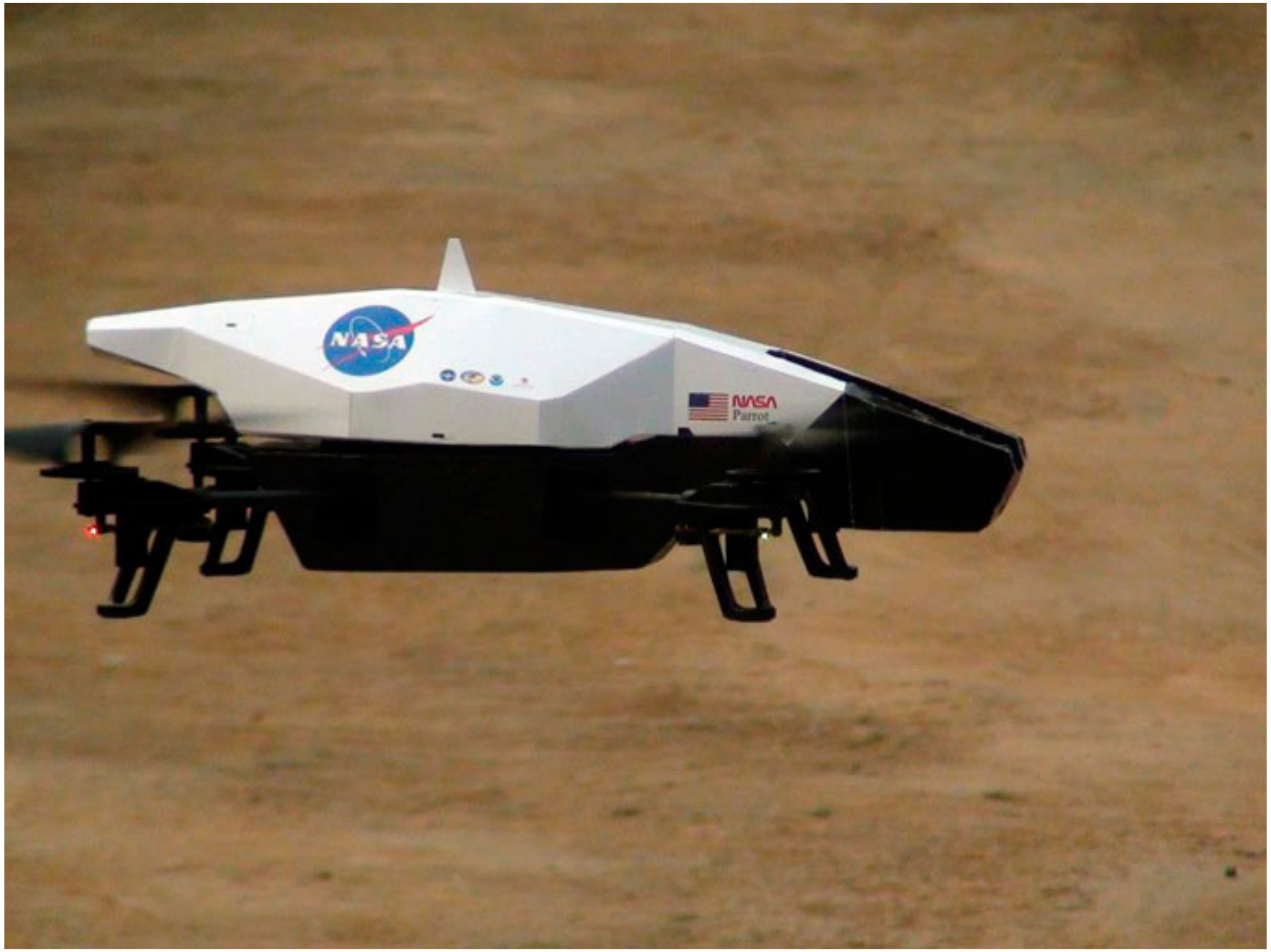




# Achieving Autonomous Flight!
# With a bit of this thrown in for good measure…

# Anatomy of a Quadcopter

# Anatomy of a Quadcopter



# Anatomy of a Quadcopter

- Four spinning blades of doom (!)
- Arms
- Onboard Wireless Access Point
- Firmware/Controller
- Cameras (forward and downward)
- Battery holder
- Shell (indoor or outdoor)

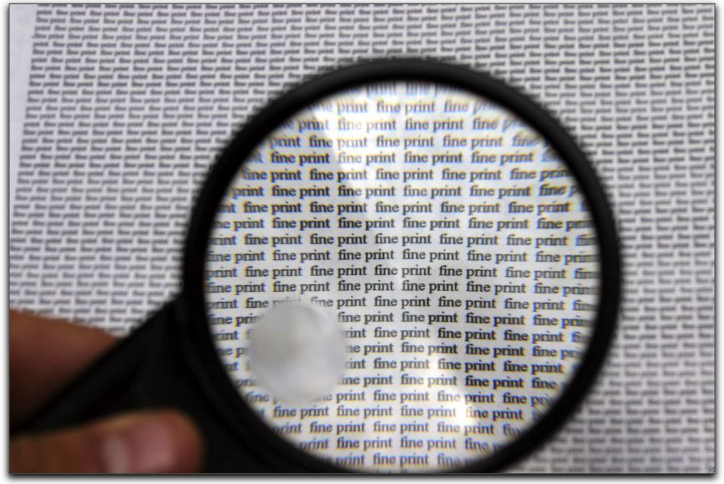
# Raspberry Pi for Brains

# Raspberry Pi for Brains
# Bill of Materials

- One (1) Raspberry Pi Model B
- One (1) Raspberry Pi case (smaller/lighter is better)
- One (1) SD card, Class 10, minimum 8G
- Two (2) Edimax EW-7811un wifi adapters
- One (1) Raspberry Pi power adapter (for initial configuration steps)
- One (1) portable USB mobile phone charger
- One (1) ethernet cable (for initial configuration steps)
- Parts list with links in appendix

# Raspberry Pi for Brains
# Configuring the Software Stack

- Raspbian, typical configuration
- Configure as Wireless Access Point (hostapd, DHCP server) – 1st adapter
- Configure to connect to drone, get IP address (DHCP client) – 2nd adapter
- Tweak ifplugd to maintain two concurrent connections
- Startup script, timing (initiating network connections, DHCP server a bit fiddly initially)
- Full documentation available upon request



# Leveraging IoT Concepts and Tools

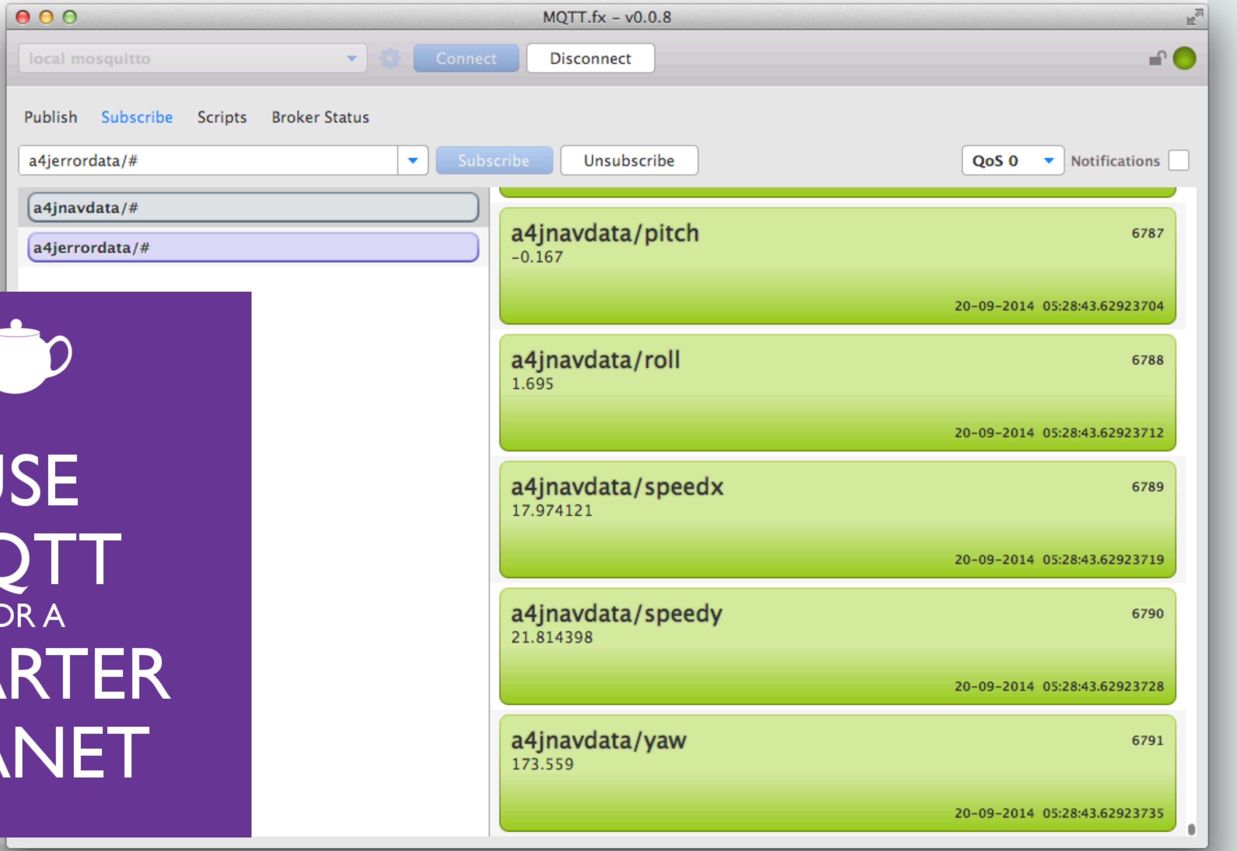
NetBeans

Java

Mosquitto

An Open Source MQTT v3.1/v3.1.1 Broker

USE
MQTT
FOR A
SMARTER
PLANET



```
a4jnavdata/pitch                                   6787
-0.167
                          20-09-2014  05:28:43.62923704

a4jnavdata/roll                                    6788
1.695
                          20-09-2014  05:28:43.62923712

a4jnavdata/speedx                                  6789
17.974121
                          20-09-2014  05:28:43.62923719

a4jnavdata/speedy                                  6790
21.814398
                          20-09-2014  05:28:43.62923728

a4jnavdata/yaw                                     6791
173.559
                          20-09-2014  05:28:43.62923735
```

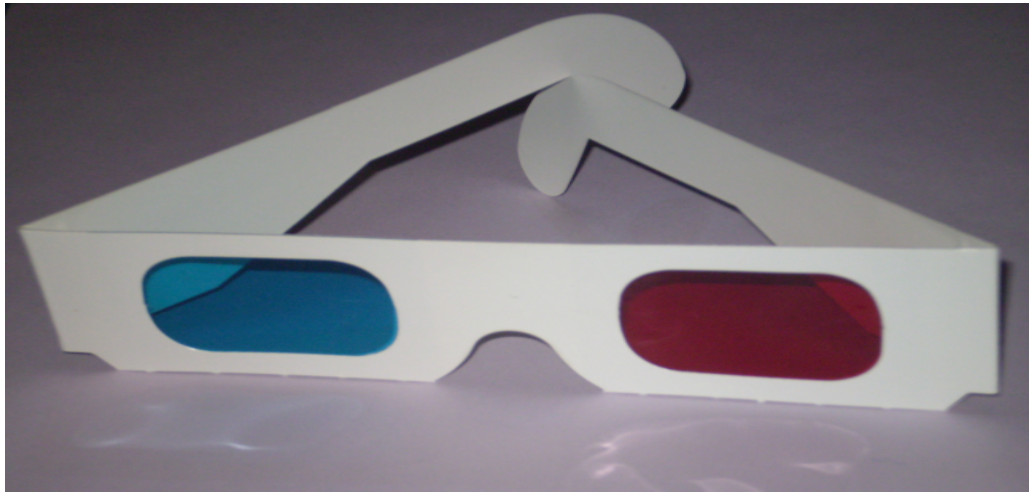
JavaOne™ ORACLE

# Demos
# Let’s have a look!

MOON MONSTERS LAUNCH ATTACK AGAINST EARTH!
HOW CAN SCIENCE MEET THE MENACE of ASTRAL ASSASSINS? New SCIENCE-FICTION THRILLS!
3D
ADVENTURES INTO THE FUTURE in New TRU-3 DIMENSION!
ROBOT MONSTER
RELEASED THRU ASTOR PICTURES CORP.
with GEORGE NADER CLAUDIA BARRETT Produced by AL ZIMBALIST

## Welcoming our Robot Overlords…

# IN 3D!!! *



* *Skull Faced Alien Space Helmeted Gorilla Suit not included.*

# Know your Overlord!!

## Loading a Robot Overlord 3D Model

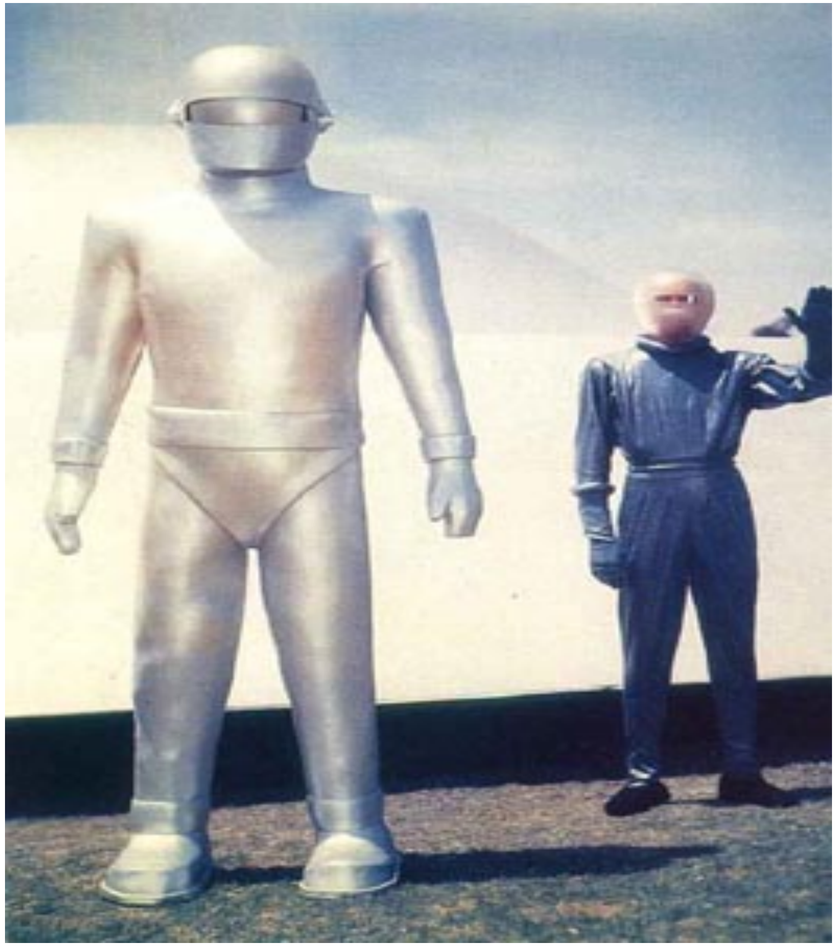
- Overlord class is a Group
  - Load model assests (.obj) within constructor
  - Leverages Interactive
    - http://www.interactivemesh.org/models/jfx3dimporter.html
  - Importer adds object MeshViews as children
- All transforms made on the entire group
- Overlord has no control smarts only API
- Exceptions handled at next years JavaOne

# Klaatu barada nikto

## Acquiring and Processing Commands

- Drag and Drop your .afr Flight Command Log onto the 3D scene.
- Function to parse into FlightCommand Object
- Add your Overlord like any other object

```
Task <Void> task = new Task<Void>() {
  @Override
  protected Void call() throws Exception {
  try {
    //Read in the playback file:
    List<FlightCommand> playback = Files.lines(file.toPath())
      .filter(line -> !line.isEmpty())
      .map(mapToFlightCommand)
      .collect(toList());

Platform.runLater(() -> {
    sceneRoot.getChildren().add(overlord);
});
```



# This is the Voice of World Control.

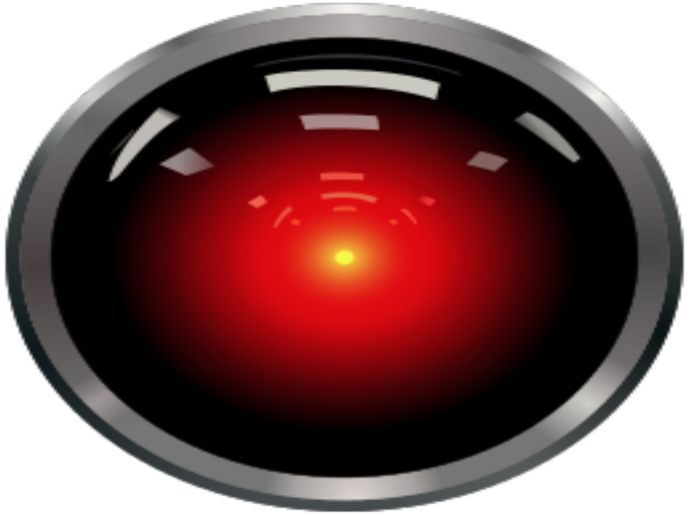
- Command & Control

```
playback.stream().forEach(flightCommand ->{
  switch(flightCommand.commandType) {
    case TAKEOFF : {
      overlord.takeoff(flightCommand.power, flightCommand.duration);
      break; }
    case FORWARD : {
      overlord.forward(flightCommand.power, flightCommand.duration);
      break; }
}
try {
    Thread.sleep(flightCommand.duration);
} catch (InterruptedException ex) {

  //The choice is yours: Obey me and live, or disobey and die." — Colossus

}
```

# Everybody… Do the Robot!

## Overlord Animation

```
public void takeoff(Integer power, Integer duration) {

//Can't really use a Path here because we are in 3D
   final Timeline t = new Timeline();
   t.getKeyFrames().addAll(new KeyFrame[]{
      new KeyFrame(
         Duration.millis(duration),
         new KeyValue[]{
            new KeyValue(yTranslateProperty(),
            getyTranslate() + (-power*sensitivity),
    Interpolator.EASE_BOTH),
         })
      });
   t.playFromStart();
}
private final DoubleProperty yTranslate
        = new SimpleDoubleProperty(0) {
                @Override
                protected void invalidated() {
                        updateGroup();
        }
};
```



Demo.  Pay attention.

# Next Step: Self-aware Drone Implementing a Control Loop

- "General Autonomy" vs. "Advanced Autonomy"
- Creating an external control loop
- Self-adjustment to achieve/maintain goal
- Higher-level goals vs. commands
- May require change of platform

# In Conclusion…

- Thank you for attending
- Keep the information flowing
  - Sean Phillips (@SeanMiPhillips)
  - James Weaver (@JavaFXpert)
  - Mark Heckler (@MkHeck)
- Any questions?

JavaOne™ ORACLE

# Appendix A
# Parts list: Positronic Brain

- Raspberry Pi Model B
- Pi case (ex: http://www.amazon.com/gp/product/B008TD1FSQ/)
- SD card (ex: http://www.amazon.com/gp/product/B00DX5D9I4/)
- Edimax wifi adapters ( http://www.amazon.com/Edimax-EW-7811Un-Adapter-Raspberry-Supports/dp/B003MTTJOY/)
- Power adapter (ex: http://www.amazon.com/gp/product/B00GWDLJGS/)
- USB charger (ex: http://www.amazon.com/gp/product/B00H7TR9WY/)
- Ethernet cable

# Safe Harbor Statement

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.

# CREATE THE FUTURE

JavaOne™

ORACLE®