wiiRemote
useR! 2009 Focus Multimedia

Landon Jensen (lsjensen@micron.com)

Vatsal Shah (vshah@purdue.edu)

07/09/09

Micron®

Nintendo Wii?

http://www.nintendo.com/wii/what/meetwii

Wii Remote from Nintendo

A main feature of the Wii Remote is its motion sensing capability, which allows the user to interact with and manipulate items on screen via movement and pointing through the use of accelerometer and optical sensor technology.

Bluetooth connection

Extensions… homebRew

http://en.wikipedia.org/wiki/Wii_Remote

Previous Work: Wii+Matlab = WiiLAB

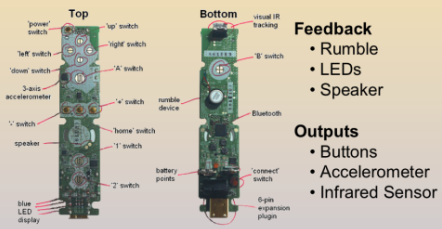
Wiimote Interactions for Freshmen Engineering Education

Jordan Brindza and Jessica Szweda
Computer Science and Engineering - University of Notre Dame
This work was supported by National Science Foundation Grant CNS-0754933

Introduction

The focus of this project was to develop hands-on laboratory modules and demonstrations involving the Nintendo Wiimote to augment the programming module for the freshmen Introduction to Engineering course. By creating a robust set of MATLAB functions we hope to make the Wiimote accessible to students of all programming levels.

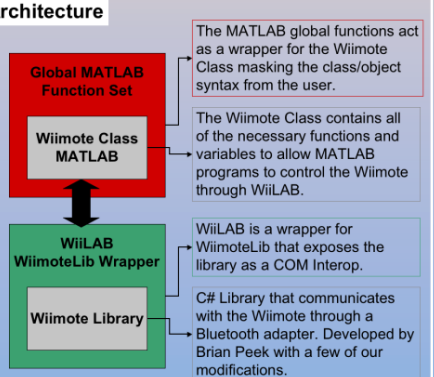
Wiimote



Feedback
- Rumble
- LEDs
- Speaker

Outputs
- Buttons
- Accelerometer
- Infrared Sensor

Architecture



The MATLAB global functions act as a wrapper for the Wiimote Class masking the class/object syntax from the user.

The Wiimote Class contains all of the necessary functions and variables to allow MATLAB programs to control the Wiimote through WiiLAB.

WiiLAB is a wrapper for WiimoteLib that exposes the library as a COM Interop.

C# Library that communicates with the Wiimote through a Bluetooth adapter. Developed by Brian Peek with a few of our modifications.

WiiLAB

Functions

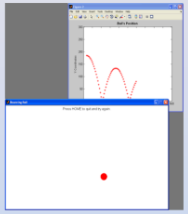
Connection	initializeWiimote( ) isWiimoteConnected( ) disconnectWiimote( )
Wiimote State	isButtonPressed(button) getWiimoteAccel( ) getAccelData(seconds) getWiimoteIR( )
Feedback	setWiimoteLEDs(led1, led2, led3, led4) setWiimoteRumble(on)
Custom Functions	initializeAccelGraph( ) initializeAccelIndicator(object)

Demos

We developed a number of demonstration applications that show how to use the Wiimote's global function set to create fun and educational programs.

Inverted Pendulum:
- Using the Wiimote's acceleration the player attempts to keep the ball balanced.

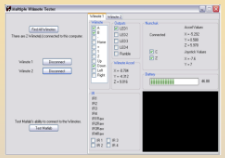
Bouncing Ball:
- Using the Wiimote, the user can throw a ball and study its movement from a position graph that is created.



Pong:
- Two players are able to use the Wiimotes to control their respective paddles in a classic tennis style game.

WiiLABTest

WiiLABTest is a stand alone C# application that allows the user to:
- Check that the Wiimotes are able to connect
  - Up to four Wiimotes
  - Validate operability of Wiimote features
- Verify that WiiLAB was installed correctly without having to go through MATLAB



Wiki

Our work is documented on the NetScale Laboratory's TWiki: http://netscale.cse.nd.edu/twiki/bin/view/Edu/WiiMote.

- Tutorials
  - Installation
  - Connecting the Wiimote
  - Getting started with WiiLAB
- Wiimote-Bluetooth pairing
- Functions
  - Description and usage
  - Quick reference guide
- Demo walkthroughs
- WiiLABTest
- Open-source

References

Peek, Brain. (2007, March). Managed Library for Nintendo's Wiimote. Coding4Fun. Retrieved June 2008, from http://blogs.msdn.com/coding4fun/archive/2007/03/14/1879033.aspx.
#widev at EFnet. (2008, July). Wiimote. WiiBrew. Retrieved July 2008, from http://wiibrew.org/wiki/Wiimote.
WiiLi Organization. (2008, July). Wiimote. WiiLi. Retrieved July 2008, from http://www.wiili.org/index.php/Wiimote
Nintendo. (2008). Wii Controllers. Nintendo. Retrieved June 2008, from http://www.nintendo.com/wii/what/controllers\#remote

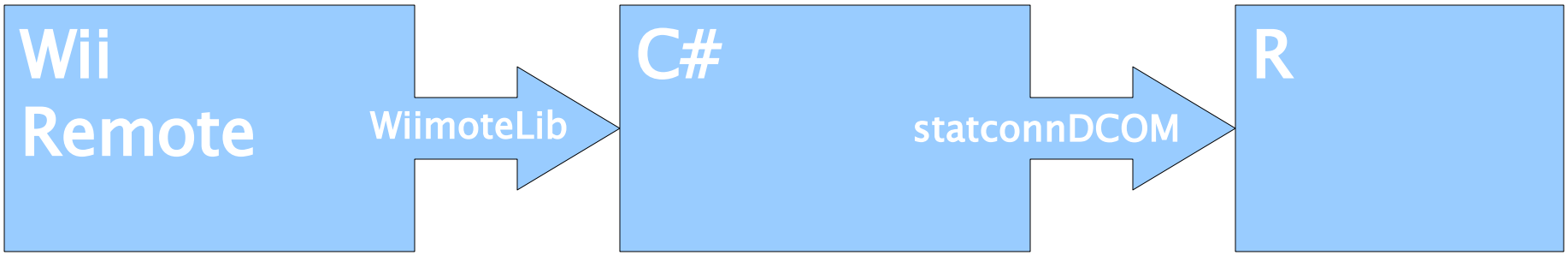
http://netscale.cse.nd.edu/twiki/bin/view/Edu/WiiMote

Wii + R: Technical Details

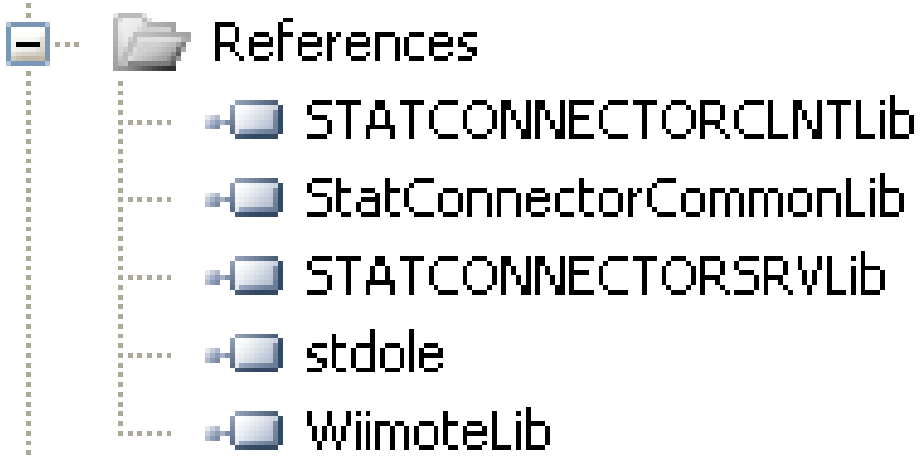
Combine

- Visual C# 2008 Studio Express
- R (via statconnDCOM)
- Wii Remote (via WiimoteLib)

Wii Remote sensor data captured by event/polling and sent to R for graphics



Technical Details (code snippets)




```
// Initiate R
StatConnector test1 = new STATCONNECTORSRVLib.StatConnectorClass();
test1.Init("R");

// create a new instance of the Wiimote
Wiimote wm = new Wiimote();

// connect to the Wiimote
wm.Connect();

// set the report type to return the IR sensor and accelerometer data (buttons always come back)
wm.SetReportType(WiimoteLib.InputReport.IRAccel, true);
```

Technical Details (code snippets, cont)

```
for (int count = 1; count < 1000; count++)
{
    // pause for 100ms
    System.Threading.Thread.Sleep(100);

    // get IR status
    state[0, 0] = (double)(wm.WiimoteState.IRState.IRSensors[0].Found ? 1 : 0);
    state[0, 1] = (double)(wm.WiimoteState.IRState.IRSensors[0].Position.X);
    state[0, 2] = (double)(wm.WiimoteState.IRState.IRSensors[0].Position.Y);

    // send IR data to R
    test1.EvaluateNoReturn(string.Concat("xpnt = 3*", 1 - state[0, 1], "- 1.5"));

    // redraw plot in R
    .....

}
```

Let's have some fun

Use WiiRemote to create/interact with data (1D, 2D, 3D, and beyond)

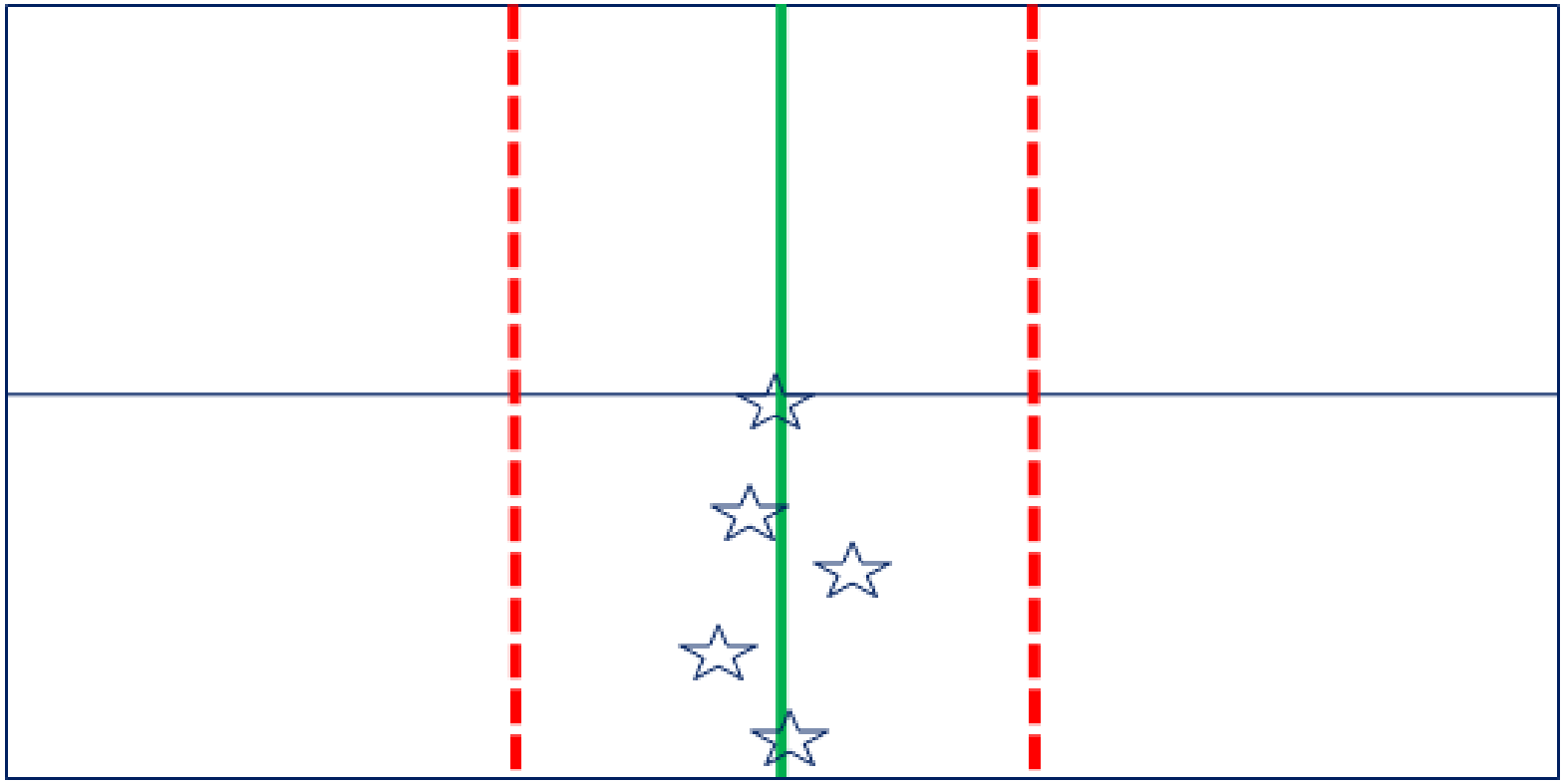
Possible applications?

- Enhance learning experience
- Check out these simple games…

Live Demo #1

1D SPC/R2R game

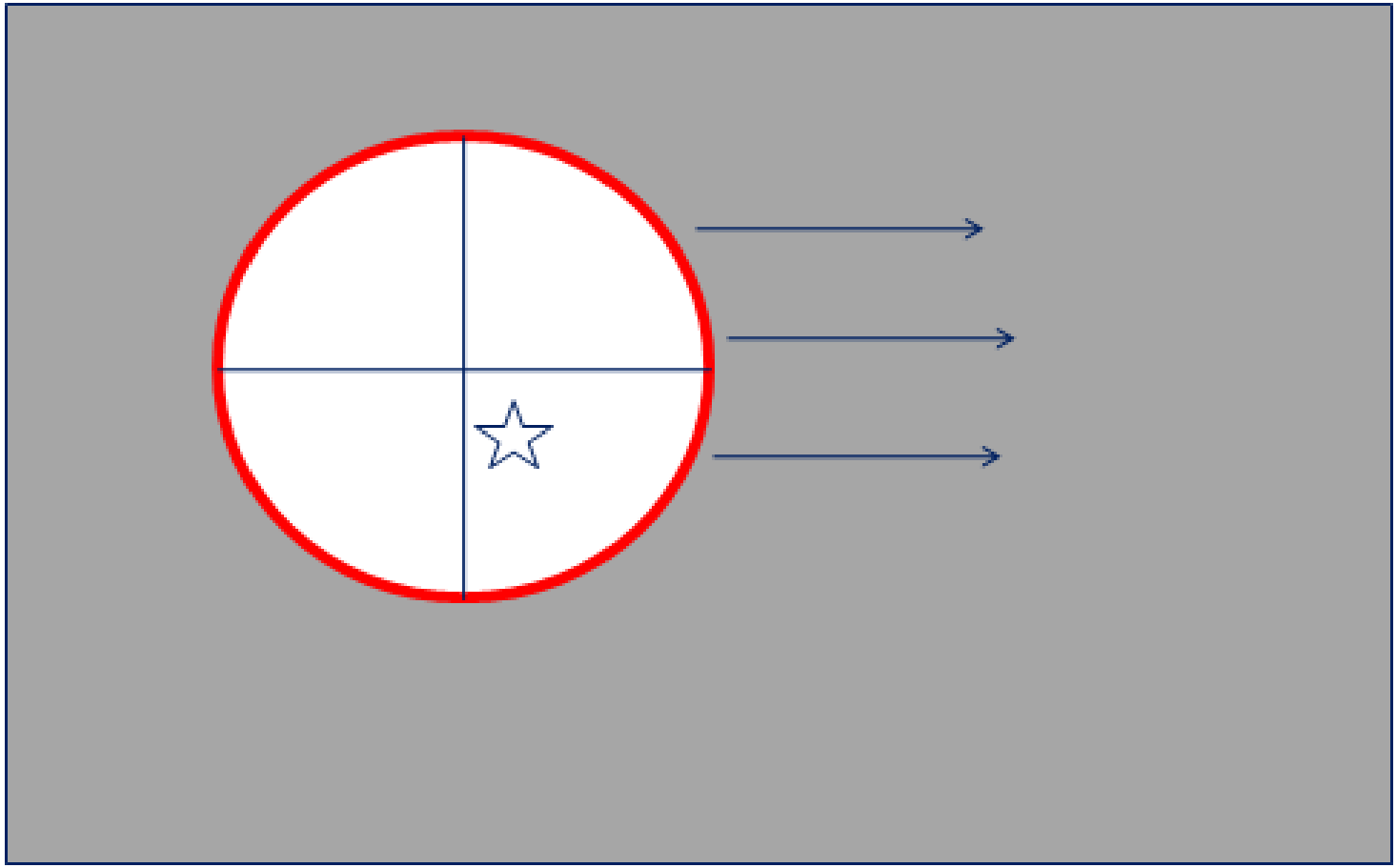
DEMO1



Live Demo #2

2D SPC/R2R game

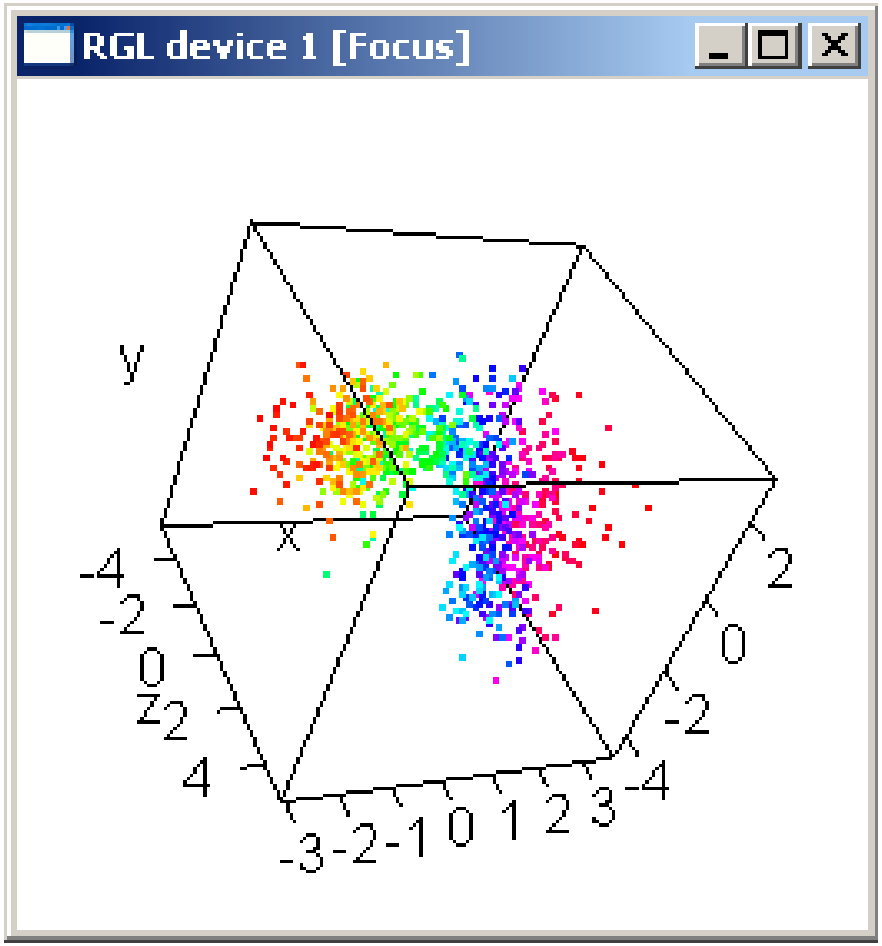
DEMO2

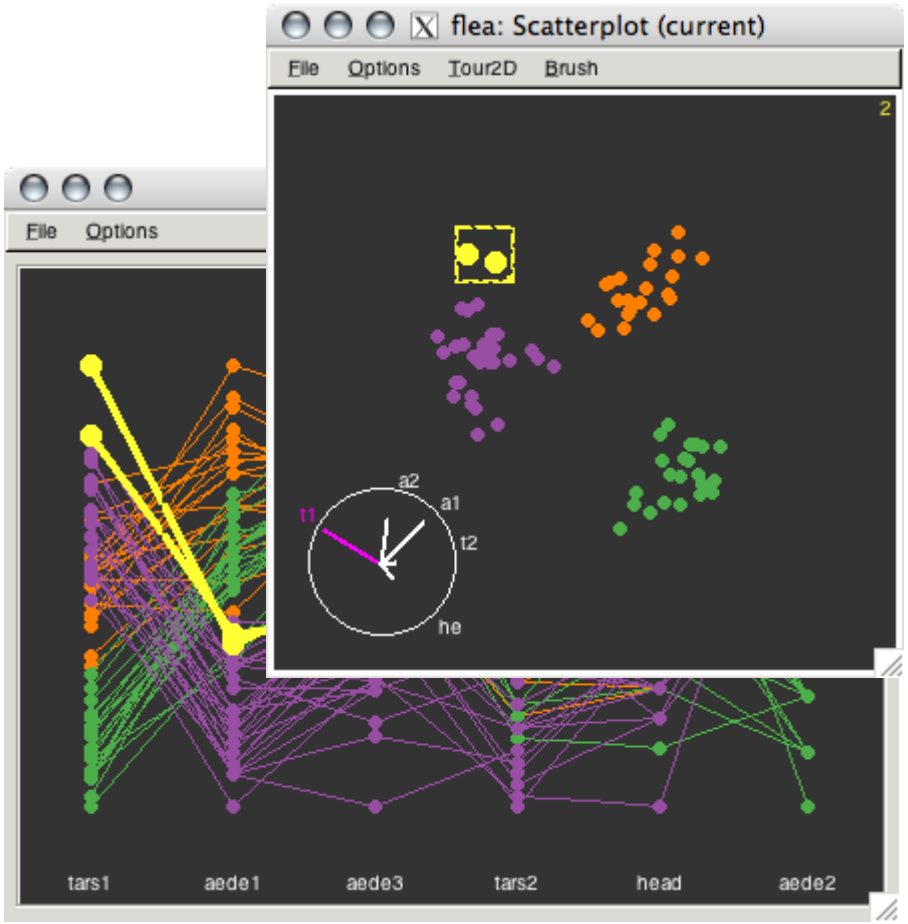


Live Demo #3

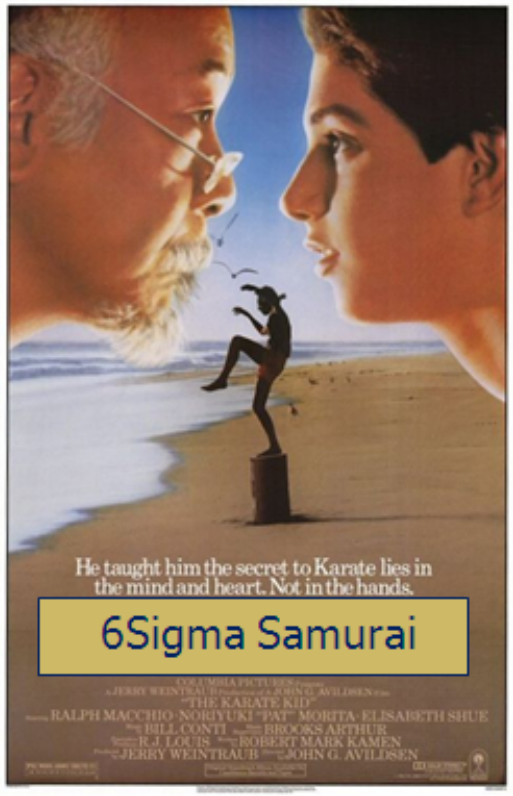
3D scatterplot via rgl or with ggobi

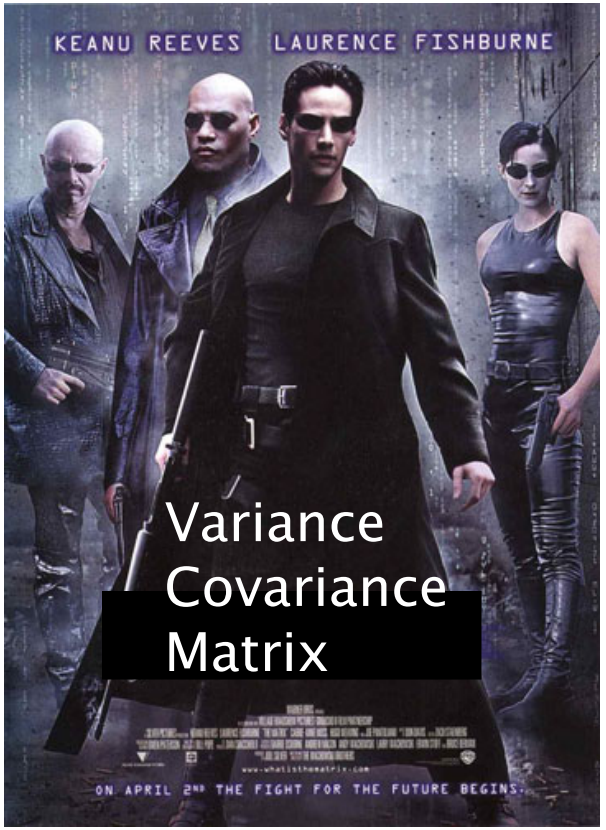
DEMO3





Other games…





Special thanks...

Johnny Chung Lee for his cool Wii Project ideas

Brian Peek for his WiimoteLib libraries

Jason Smith for Wiimote Presentor

Great packages: rgl, iplots

Micron Technology, Inc and Purdue University for sponsoring our travels