#### Roboting with iOS, OpenCV, BLE, and Lego Robot Robot Stick Ball Robot

But Why? Dömötör Gulyás' and Arnau Vázquez' Project for "Special Aspects of Autonomous Mobile Systems" @ TH Köln presented @ Cocoaheads AC, 31.3.2016

#### Overview





### iPhone

- Fancy CPU/GPU
- Fancy camera
- BLE
- Gyros! (not the nomnomnom kind)
- Brains of the operation
- Waaaay faster than rpi



# Lego Lego Lego Lego

- Lego Pieces
- Lego Motors
- Lego Gears
- Lego Tracks
- (Lego tracks don't like carpets)

- Everything Lego
- Except Electronics

...Lego

min(20, 209, 218) m

Stop CV

Sensor Settings.

Ball Info

# If it sparks, you haven't used enough tape

11.1V 3S 30

#### More Powwwr

- 4x 9V Lego motors, for a combined 0.05HP
- LiPoly Battery
- 1A motor drivers

#### Facts

- Top speed over 1.0 m/s when driving off tables
- Turn rate a dazzling 180°/s+
- Full speed too fast for 5 fps tunnel vision



#### RFduino

- Arduino compatible
- BLE Stack in Software
- Nordic 32bit ARM @ 16MHz
- tiny
- Program in Arduino C
- Drive motors

#### RFduino BLE

- CPU shared with binary BLE SoftDevice
- Custom BLE Service
   & Characteristics

# Xcode all the things

#### iOS natively, embedded with makefiles

🔴 🕘 🕒 🔳 🎯 r	fduino-lego-tank 👌 💻 My Mac	rfduino-lego-tank   Build rfduino-lego-tank: Succeeded   19.11.2014 a
	Edit	ble_gatts.h
┓ ヱ ♀ ♪ ♀ Ⅲ ┍ ₽	器   く >   💁 rfduino-lego-tank 〉 🗋 Make	efile > No Selection
rfduino-lego-tank 1 target, OS X SDK 10.10	<pre>1 # build scripts come from https://git 2 # modifiy Makefile.config.template fr</pre>	hub.com/sprhawk/nrf51822-macosx-build-scripts om build scripts to adapt your environment
Makefile Makefile.config	<pre>3 4 include Makefile.config 5</pre>	
<pre>src h ltank_board.h</pre>	6 TARGET_CHIP := NRF51822_QFAA_CA 7 BOARD := BOARD_RFDUINO	
h pstorage_platform.h s gcc_startup_nrf51.s c main.c	<pre>9 #Uncomment correct line if you have s 10 DEVICE_VARIANT := xxaa 11 #DEVICE_VARIANT := xxab</pre>	<pre>#Uncomment correct line if you have s110 programmed on the chip. DEVICE_VARIANT := xxaa #DEVICE_VARIANT := xxab</pre>
c Itank_ble.c h Itank_ble.h	12 13 #USE_SOFTDEVICE := S110 14 USE_SOFTDEVICE := S110-v7 15 #USE_SOFTDEVICE := S120	

v3 rotation = vcreate(deviceMotion.rotationRate.x, deviceMotion.rotationRate.y, deviceMotion.rotationRate.z); q4 orientation = qcreate(deviceMotion.attitude.quaternion.x, deviceMotion.attitude.quaternion.y, deviceMotion.attitude.quaternion.z, deviceMotion.attitude.quaternion.y, deviceMotion.attitude.quaternion.x, deviceMotion.attitude.quaternion.x, deviceMotion.attitude.quaternion.x, deviceMotion.attitude.quaternion.y, deviceMotion.attitude.quaternion.x, deviceMotion.atti

v3 globalRotation = qTransformVector(orientation, rotation); v3 globalForward = qTransformVector(orientation, localForward);

globalForward.v.z = 0.0f; globalForward = vmul(globalForward, 1.0f/vlength(globalTorward)) OOULES currentHeadingAngle = atan2f(globalForward.v.y, globalForward.v.x) OOULES yawRate = globalRotation.v.z; v3 targetForward = vcreate(cosf(targetHeadingAngle), sinf(targetHeadingAngle), 0.0f);

- iPhone app (Objective C, C++)
  - Image processing (OpenCV)
  - Reasoning (fetch the ball, atta boy!)
- Robot firmware (C)
  - Motor control

BOOL rollowEnabled = [defaults boolForKey: LTSteerToBallEnabledKey];
BOOL steeringEnabled = [defaults boolForKey: LTSteerToBallEnabledKey];

float leftSteering = gyroControlValue\*steeringEnabled;
float rightSteering = -gyroControlValue\*steeringEnabled;

Floot may and - 1 AF Emay E(lofk Chapping might Chapping);





- Library for image recognition and computer vision
- Can detect friggin balls

# OpenCV

- Currently at Version 3.x
- Called "OpenCV2.framework" when downloaded, haha
- C/C++, works on iOS out of the box
- Prototype in Python on OS X
- Image processing to hit the ground running

## Er, yeah, sometimes uggh.

• Badly parametrised tracker tracks badly.



#### There are lots of parameters



# HSV Conversion

- Yay colorspace conversions, always fun
- HSV more resilient to illumination changes than RGB







# Ball tracking (Mask)

- Mask is applied to selected range in the HSV image
- The output is a black and white image.







# Ball tracking (Blurring)

• for better edge detection, curiously





#### Ball tracking (Hough Circles)

- Objects shaped like circles are detected
- Also yellow sneakers



# Gyro-Assisted Steering



- Fast
- Can remember ball heading
- sounds fancy

## Indispensable Tools





#### **Open Bench Logic Sniffer**

#### Ball-on-a-Stick

## The End

Questions?