ROS Scratch: Enabling Block-Based Robotics

Brian Thomas

Brown University
Department of Computer Science

Robots for Education, May 2, 2011
What did I do?

- New blocks
- ROS <-> Scratch Interface
- Demos for New Blocks
- Wiki instructions
• New motor command Scratch blocks

- motors forward at 80% power
- motors left -100, right 100
- motors forward at 80% power for 1.0 secs
- motors left -100, right 100 for 1.0 secs
- motors stop
New Blocks

- New motor command Scratch blocks

```plaintext
(broadcastMotorPower 0) 
(broadcastMotorPowerSlow 25) 
(broadcastMotorPowerMedium 50) 
(broadcastMotorPowerFast 75) 
(broadcastMotorPowerFullPower 100)
```

```plaintext
robotMotorPowerMenu

<table>
<thead>
<tr>
<th>menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>menu + CustomMenu new</td>
</tr>
</tbody>
</table>

```plaintext
 paused 0
 slow 25
 medium 50
 fast 75
 full power 100
```

```plaintext
 do: [ pair |
 menu
 add '(' + UTM, pair second printString, ')', pair first localized action: pair second). |
 menu
```

```plaintext
$0 = code [ True + ChooseArgMorph new getOptionsSelector: #verbsMenu andOutput: ''].
$1 = code [ True + ChooseArgMorph new getOptionsSelector: #robotMotorDirection andOutput: ''].
$2 = code [ True + ChooseArgMorph new getOptionsSelector: #motorPower andOutput: ''].
$3 = code [ True + ChooseArgMorph new getOptionsSelector: #motorName andOutput: ''].
$4 = code [ True + ChooseArgMorph new getOptionsSelector: #motorDirection andOutput: ''].
$5 = code [ True + ChooseArgMorph new getOptionsSelector: #motorPower andOutput: ''].
$6 = code [ True + ChooseArgMorph new getOptionsSelector: #robotMotorPowerMenu andOutput: ''].
$7 = code [ True + ChooseArgMorph new getOptionsSelector: #robotMotorPowerMenu andOutput: ''].
```

```plaintext
broadcastMotorsLeftRight:
 leftRight: right
 self broadcast: left, left string: 'left', right string: 'right', right string: 'right', string with Argument 0.
```
New Blocks

- New robot sensor Scratch blocks

```plaintext
('bump sensor'  b  bumpSensor)
('light sensor left'  r  lightSensorLeft)
('light sensor right'  r  lightSensorRight)

bumpSensor
"true if pressed, false else."

+ ((self sensor: 'rosScratchBump') = 1).

lightSensorLeft
"Range is ~ 0 to 1024."

+ self sensor: 'rosScratchLightLeft'.
```
Making New Scratch Functionality

- Entire filesystem is an image
- Code in system browser

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CameraMedia</td>
<td>FilterPack</td>
<td>ImageMedia</td>
<td>MovieMedia</td>
<td>ScratchMedia</td>
<td>ScratchSpriteMorph</td>
<td>ScratchStageMorph</td>
<td>SpriteableScratchMorph</td>
<td>SoundMedia</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System Browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- all --</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

broadcastMotorsLeft: left
Right: right
Secs: secs elapsed: elapsed from: ignored
elapsed := ((secs/1000) as integer) - 100
ifTrue: [if self broadcastMotorsStop, self broadcastMotorsLeft: left Right: right]
ROS Scratch
Brian Thomas

ROS<->Scratch Interface

```python
#!/usr/bin/env python
import ...

def parseData(str):
    # Parse text received from Scratch broadcasts
    ...
    if e: # Successfully parsed
        ...
        tank(left,right)

def tank(left, right):
    # Call the tank service for l,r
    ...

def sendScratchSensor(variable, value, scratchSock):
    sendScratchCommand('sensor-update "'+variable+'" '+value+' ', scratchSock)

def sendScratchCommand(cmd, scratchSock):
    ...
    scratchSock.send(a.tostring() + cmd)

def cb_sensorPacket(sp):
    ...
    sendScratchSensor("rosScratchBump", "1" if (sp.bumpLeft or sp.bumpRight) else "0", scratchSock)
    sendScratchSensor("rosScratchLightLeft", str(sp.cliffFrontLeftSignal), scratchSock)
    sendScratchSensor("rosScratchLightRight", str(sp.cliffFrontRightSignal), scratchSock)

def makeConnection():
    global scratchSock
    ...
    scratchSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    ...

def main():
    # Connect to ROS
    ...
    # Receive Scratch commands:
    ...
    makeConnection()
    ...
    data = scratchSock.recv(1024)
    ...
    parseData(data)
```
ROS<->Scratch Interface

- **Main release**: iRobot Create (Movement and sensing)
- **Experimental release**: iRobot Create + Camera
- **Experimental release**: AR.Drone (Movement only)
Made ROS part more failure-robust

- Automatic node restarts

```xml
<launch>
  <!-- iCreate -->
  <param name="/brown/irobot_create_2_1/port" value="/dev/rfcomm0" />
  <node name="irobot_create_2_1" pkg="irobot_create_2_1" type="driver.py" respawn="true" />

  <!-- ROS<>Scratch interface -->
  <node name="ros_scratch_icreate" pkg="ros_scratch" type="ros_scratch_icreate.py" respawn="true" output="screen" />
</launch>
```
Demos for New Blocks

• AR tag following
Demos for New Blocks

- Enclosure escape

- [Diagram of Scratch code for enclosure escape]
Demos for New Blocks

- Line following

```plaintext
when space key pressed
set leftRightThres to 800
set lightBrightThres to 500
forever
  if bump sensor
    motors stop
  else
    if light sensor left > leftRightThres
      motors left -20 right 20
    else
      if light sensor right > rightBrightThres
        motors left 20 right -20
      else
        motors forward at 20 to power

when ▼ key pressed
  motors stop
  stop all
```
Demos for New Blocks

- Basic motor control
Demos for New Blocks

- Teleoperation
Installation

There are two major ways to install this software. The simpler of the two is to simply download a copy of our virtual machine², which has all of the software pre-installed, and run the machine inside of VirtualBox.

A more advanced installation below requires the user to download and install packages in Ubuntu or another ROS and Scratch compatible operating system.

Modified Scratch

Install Scratch (sudo aptitude install scratch). Obtain the ros-scratch.zip package and unzip it on your $PATH. Modify the path within the executable wrapper ros-scratch to reflect the path to the file ScratchSourceCode12.4 image.

ROS and ros_scratch

Install ROS, then install brown-ros-pkg by checking it out of SVN and placing it on $ROS_PACKAGE_PATH.

Changes from original Scratch

For developers looking to incorporate the changes made in our Scratch image into their own Scratch images, the following files in the virtual filesystem have been modified:

- Scratch-Objects/ScratchSpriteMorph2/class/block\ specs/blockSpecs
- Scratch-Objects/ScriptableScratchMorph2/class/block\ specs/blockSpecs
- Scratch-Objects/ScriptableScratchMorph2/instance/sensing\ ops\ robot/ (entire folder)
- Scratch-Objects/ScriptableScratchMorph2/instance/motor\ ops\ robot/ (entire folder)
- Scratch-Blocks/CommandBlockMorph2/instance/accessing/uncoloredArgMorphFor:
• Questions?
• Comments?
This slide intentionally left blank.