ROS Scratch: Enabling Block-Based Robotics

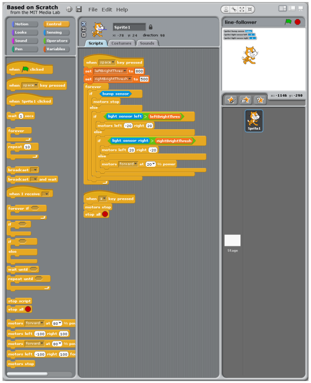
Brian Thomas

Brown University
Department of Computer Science

Robots for Education, May 2, 2011

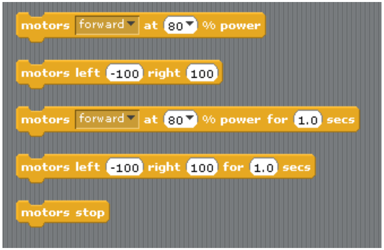
What did I do?

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


```
when [space] key pressed
set [leftBrightThres] to 800
set [rightBrightThresh] to 500
forever
    if <bump sensor>
        motors stop
    else
        if <(light sensor left) > (leftBrightThres)>
            motors left -20 right 20
        else
            if <(light sensor right) > (rightBrightThresh)>
                motors left 20 right -20
            else
                motors [forward] at 20 % power

when [a] key pressed
motors stop
stop all
```

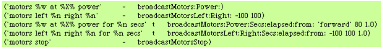
New Blocks

- New motor command Scratch blocks

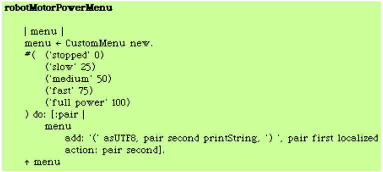

motors forward at 80 % power
motors left -100 right 100
motors forward at 80 % power for 1.0 secs
motors left -100 right 100 for 1.0 secs
motors stop

New Blocks

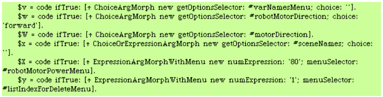
- New motor command Scratch blocks



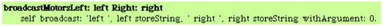
```
('motors %w at %X% power'          -    broadcastMotors:Power:)
('motors left %n right %n'         -    broadcastMotorsLeft:Right: -100 100)
('motors %w at %X% power for %n secs'  t    broadcastMotors:Power:Secs:elapsed:from: 'forward' 80 1.0)
('motors left %n right %n for %n secs'  t    broadcastMotorsLeft:Right:Secs:elapsed:from: -100 100 1.0)
('motors stop'                     -    broadcastMotorsStop)
```



```
robotMotorPowerMenu

    | menu |
    menu ← CustomMenu new.
    #(  ('stopped' 0)
        ('slow' 25)
        ('medium' 50)
        ('fast' 75)
        ('full power' 100)
    ) do: [:pair |
        menu
            add: '(' asUTF8, pair second printString, ') ', pair first localized
            action: pair second].
    ↑ menu
```



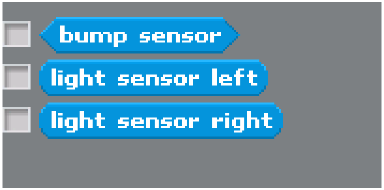
```
    $v = code ifTrue: [↑ ChoiceArgMorph new getOptionsSelector: #varNamesMenu; choice: ''].
    $w = code ifTrue: [↑ ChoiceArgMorph new getOptionsSelector: #robotMotorDirection; choice:
'forward'].
    $W = code ifTrue: [↑ ChoiceArgMorph new getOptionsSelector: #motorDirection].
    $x = code ifTrue: [↑ ChoiceOrExpressionArgMorph new getOptionsSelector: #sceneNames; choice:
''].
    $X = code ifTrue: [↑ ExpressionArgMorphWithMenu new numExpression: '80'; menuSelector:
#robotMotorPowerMenu].
    $y = code ifTrue: [↑ ExpressionArgMorphWithMenu new numExpression: '1'; menuSelector:
#listIndexForDeleteMenu].
```

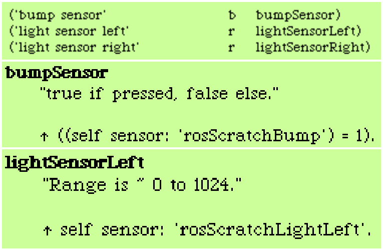


```
broadcastMotorsLeft: left Right: right
    self broadcast: 'left ', left storeString, ' right ', right storeString withArgument: 0.
```

New Blocks

- New robot sensor Scratch blocks





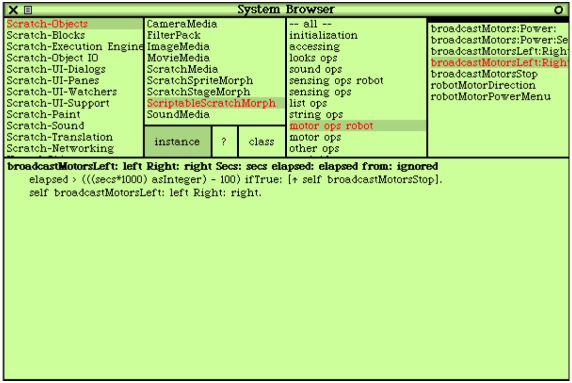
```
('bump sensor'                 b    bumpSensor)
('light sensor left'           r    lightSensorLeft)
('light sensor right'          r    lightSensorRight)
```

```
bumpSensor
    "true if pressed, false else."

    ↑ ((self sensor: 'rosScratchBump') = 1).
```

```
lightSensorLeft
    "Range is ~ 0 to 1024."

    ↑ self sensor: 'rosScratchLightLeft'.
```

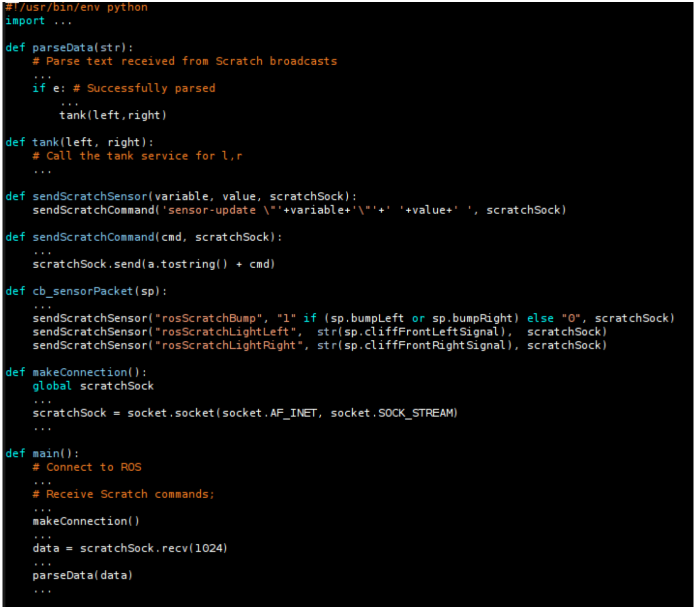
Making New Scratch Functionality

- Entire filesystem is an image
- Code in system browser



```
broadcastMotorsLeft: left Right: right Secs: secs elapsed: elapsed from: ignored
    elapsed > (((secs*1000) asInteger) - 100) ifTrue: [↑ self broadcastMotorsStop].
    self broadcastMotorsLeft: left Right: right.
```

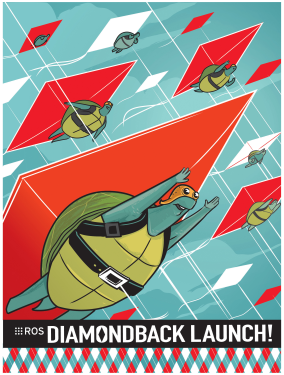
ROS<->Scratch Interface



```
#!/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)
    ...
```

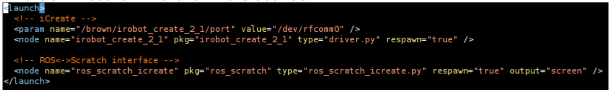
ROS<->Scratch Interface

- **Main release**: iRobot Create (Movement and sensing)
- **Experimental release**: iRobot Create + Camera
- **Experimental release**: AR.Drone (Movement only)



Made ROS part more failure-robust

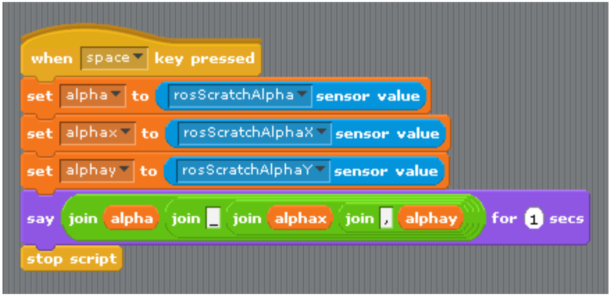
- Automatic node restarts



```
<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

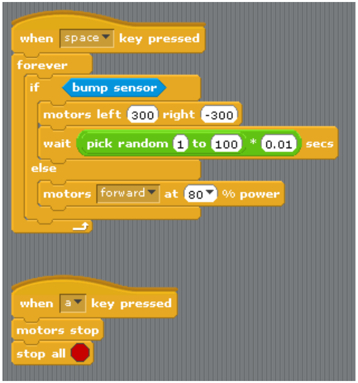
- AR tag following



```
when space key pressed
set alpha to rosScratchAlpha sensor value
set alphax to rosScratchAlphaX sensor value
set alphay to rosScratchAlphaY sensor value
say join alpha join _ join alphax join , alphay for 1 secs
stop script
```

Demos for New Blocks

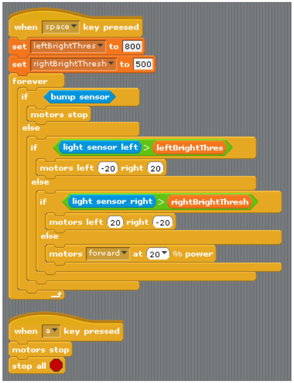
- Enclosure escape



```
when [space] key pressed
forever
    if <bump sensor>
        motors left (300) right (-300)
        wait ((pick random (1) to (100)) * (0.01)) secs
    else
        motors [forward] at (80) % power
    

when [a] key pressed
motors stop
stop all
```

Demos for New Blocks

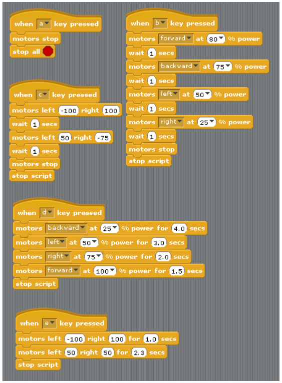
- Line following



```
when space key pressed
set leftBrightThres to 800
set rightBrightThresh to 500
forever
    if bump sensor
        motors stop
    else
        if light sensor left > leftBrightThres
            motors left -20 right 20
        else
            if light sensor right > rightBrightThresh
                motors left 20 right -20
            else
                motors forward at 20 % power

when a key pressed
motors stop
stop all
```

Demos for New Blocks

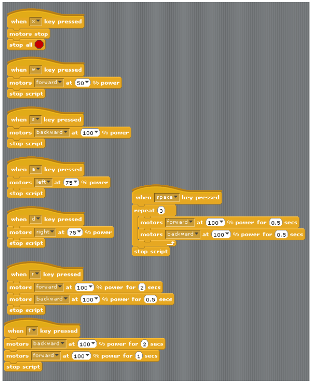
- Basic motor control



```
when a key pressed
motors stop
stop all

when b key pressed
motors forward at 80 % power
wait 1 secs
motors backward at 75 % power
wait 1 secs
motors left at 50 % power
wait 1 secs
motors right at 25 % power
wait 1 secs
motors stop
stop script

when c key pressed
motors left -100 right 100
wait 1 secs
motors left 50 right -75
wait 1 secs
motors stop
stop script

when d key pressed
motors backward at 25 % power for 4.0 secs
motors left at 50 % power for 3.0 secs
motors right at 75 % power for 2.0 secs
motors forward at 100 % power for 1.5 secs
stop script

when e key pressed
motors left -100 right 100 for 1.0 secs
motors left 50 right 50 for 2.3 secs
stop script
```

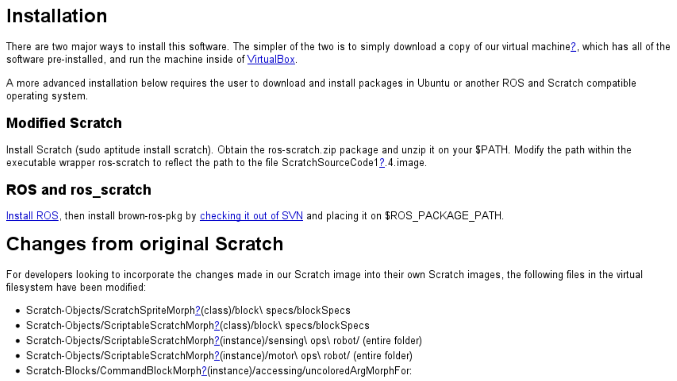
Demos for New Blocks

- Teleoperation



```
when x key pressed
motors stop
stop all

when w key pressed
motors forward at 50 % power
stop script

when s key pressed
motors backward at 100 % power
stop script

when a key pressed
motors left at 75 % power
stop script

when d key pressed
motors right at 75 % power
stop script

when r key pressed
motors forward at 100 % power for 2 secs
motors backward at 100 % power for 0.5 secs
stop script

when f key pressed
motors backward at 100 % power for 2 secs
motors forward at 100 % power for 1 secs
stop script

when space key pressed
repeat 3
    motors forward at 100 % power for 0.5 secs
    motors backward at 100 % power for 0.5 secs
stop script
```

Wiki instructions



Installation

There are two major ways to install this software. The simpler of the two is to simply download a copy of our virtual machine?, 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 ScratchSourceCode1?.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.

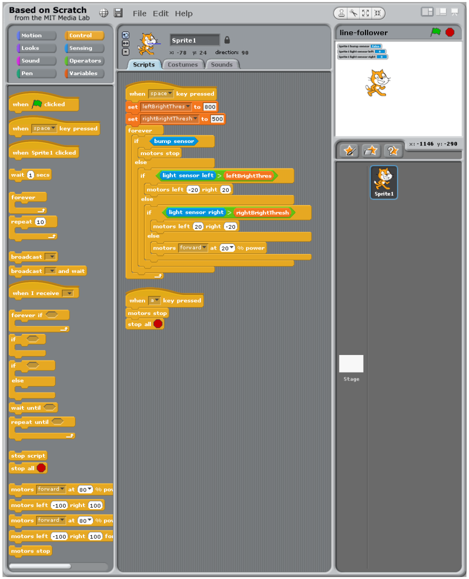
Changes from original Scratch

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/ScratchSpriteMorph?(class)/block\ specs/blockSpecs
- Scratch-Objects/ScriptableScratchMorph?(class)/block\ specs/blockSpecs
- Scratch-Objects/ScriptableScratchMorph?(instance)/sensing\ ops\ robot/ (entire folder)
- Scratch-Objects/ScriptableScratchMorph?(instance)/motor\ ops\ robot/ (entire folder)
- Scratch-Blocks/CommandBlockMorph?(instance)/accessing/uncoloredArgMorphFor:

The End

- Questions?
- Comments?



```
when [space] key pressed
set [leftBrightThres] to 800
set [rightBrightThresh] to 500
forever
  if <bump sensor>
    motors stop
  else
    if <(light sensor left) > (leftBrightThres)>
      motors left -20 right 20
    else
      if <(light sensor right) > (rightBrightThresh)>
        motors left 20 right -20
      else
        motors [forward] at 20 % power

when [a] key pressed
motors stop
stop all
```

Blank

This slide intentionally left blank.