Robotics in the Classroom: Making Computer Science and Math Come Alive

GaETC 2010
BYOL Session 46
Christopher Michaud, Nebo Elementary
Paulding County Schools
Dallas, Georgia

www.nebomusic.net
Three Activities of Children

- Games
- Storytelling
- Simulations

All activities work well with Robotics!
Why Programming?

- Programming allows students to encode and reflect on sequential and logical thought in a dynamic system.
- Real applications of math concepts
  - Coordinate plain
  - Directions
  - Grids/Arrays
- Game making is a form of storytelling
- Develops Technological Fluency
Hierarchy of Technology Skills

1. **Passive Reception:** From creator to receiver with no action on receiver to alter flow of content. (watching a video)

2. **Active Research:** From creator to receiver with receiver selecting path and specific points of content. (Surfing the web)

3. **Interactive 1 way:** From creator to receiver through game model. Receiver must master a skill to progress through activity. (Skill and Drill)

4. **Communication / Expression:** User creates content and art with technology. Verbal / Text Images Plot Web

5. **Interactive 2 way:** creator and users interact. (Email, Blogs, Multi-User Virtual Environments)

6. **Data manipulation and Analysis:** User uses computer to interact and manipulate the content and data.

7. **Software creation / programming:** Users encodes logical thought and algorithms into computer. User tells computer what to do. Provides platforms and engines for above skill sets.

8. **Hardware creation:** User designs and assembles hardware to run, input, display software.
7 Essentials Elementary Programming:

Objects

Methods

Properties

Loops

Conditionals

Events

Variables
7 Essentials Elementary Programming:

Who

Objects
Methods
Properties

What

Loops
Conditionals
Variables

When

Events
GPS Standards

- **M5P3**: Students will communicate mathematically.

- **M5P1**: Students will solve problems (using appropriate technology).
Today’s Goals

- Overview of Training Robot
- Using Bricx IDE
- Basic Robot commands in Bricx
- Write 3 simple programs for Robot
Training Robot

Right: Motor B

Left: Motor C

Touch Sensor
Getting Started

NXT Tools

Programming Area
Installing Software

- Insert CD and Copy Folder to Desktop
- Double Click "bricxcc_setup_33719.exe"
- Run Installation
- Install Driver: Double Click "LegoMindstormsNXTdriver32.msi"
Connecting the Robot

- Start Bricx Command Center
- Select “usb” and “NXT”
- Turn On Robot
- Plug Robot Into Computer
- Wait short time for computer to recognize Robot
- Click “OK”
Test Drive

- Click “Joystick” Tool
- Set Motors to C Left and B Right
- Try the Arrows! (Make sure the robot does not drive off the table!)
Basic Bricx / NXC Commands:

**Motor (Output) Commands:**
- `OnFwd("ports", "pwr");`
- `OnRev("ports", "pwr");`
- `Off("ports");`
- `OnFwdSync("ports", "pwr", "turnpct");`
- `RotateMotor("ports", "pwr", "angle");`

**Flow Commands:**
- `Wait(4000);` (milliseconds)
- `until ("condition");`
- `repeat ("value")`
  `{`
  `  "body"`
  `}`

**Task and Sub Commands:**
- `Sub methodName()`
  `{`
  `  "body"`
  `}`
- `task main()`
  `{`
  `  "body"`
  `}`
First Robot Program: Out and Back

```c
1 task main()
2 {
3     OnFwd(OUT_BC, 50);
4     Wait(4000);
5     OnRev(OUT_BC, 50);
6     Wait(4000);
7     Off(OUT_BC);
8 }
9
```
Compile and Download

- Save Your Program (File -> Save)
- Select “Compile->Compile” from menu bar. (Or “F5”)
- Turn On Robot
- Select Compile -> Download
- Wait for Beep
- Unplug Robot and Test! (Use Orange Button)
  - Robot: Software -> My Files -> Your Program
  - Select Program with Orange Button
Simple Square with Synchronized Motors

```c
// Simple Square Program
// Using Wait Blocks
// Mr. Michaud
// www.nebomusic.net

sub simpleSquare()
{
    repeat(4)
    {
        OnFwdSync(OUT_BC, 50, 0); // Forward
        Wait(2000);
        Off(OUT_BC);
        RotateMotor(OUT_B, 50, 410); // 90 Degree Turn Left
    }
}

// main()

main()
{
    simpleSquare();
}
```
Rectangle using Rotation Sensors

```c
// Declare Variable for Starting Rotation
int startingRotation;

// Subroutine for Rectangle
sub driveRectangle()
{
    ResetRotationCount(OUT_ABC); // Reset Rotation Sensors
    repeat(2)
    {
        // Long Leg of Rectangle
        startingRotation = MotorRotationCount(OUT_B); // Set Baseline Rotation
        OnFwdSync(OUT_BC, 50, 0);
        until(MotorRotationCount(OUT_B) > startingRotation + 900); // Wait Until Rotate 900 Degrees.
        OffEx(OUT_BC, RESET_ALL);
        Wait(250);
        RotateMotor(OUT_B, 50, 410); // Swing Turn 90 Degrees Left
        Wait(250);

        // Short Leg of Rectangle
        startingRotation = MotorRotationCount(OUT_B);
        OnFwdSync(OUT_BC, 50, 0);
        until(MotorRotationCount(OUT_B) > startingRotation + 720);
        OffEx(OUT_BC, RESET_ALL);
        Wait(250);
        RotateMotor(OUT_B, 50, 410);
        Wait(250);
    }
}

// Main Method
task main()
{
    driveRectangle();
}
```
Apply: Maze #2 with “Wait”

```c
1 // Simple Maze #2 Program
2 // Using Wait Commands
3 // Mr. Michaud
4 // www.nebomusic.net
5
6 task main ();
7 {
8    OnFwdSync(OUT_BC, 50, 0); // Forward
9    Wait(2000);
10   Off(OUT_BC);
11   RotateMotor(OUT_B, 50, 410); // 90 Degree Turn Left
12   OnFwdSync(OUT_BC, 50, 0); // Forward
13    Wait(1000);
14   Off(OUT_BC);
15   RotateMotor(OUT_C, 50, 410); // 90 Degree Turn Right
16   OnFwdSync(OUT_BC, 50, 0); // Forward
17    Wait(2000);
18   Off(OUT_BC);
19   RotateMotor(OUT_B, 50, 410); // 90 Degree Turn Left
20   OnFwdSync(OUT_BC, 50, 0); // Forward
21    Wait(2000);
22   Off(OUT_BC);
23 }
24
```
Maze #2
Maze #1
Touch Stop (Using Touch Sensor)

```c
1 sub touchStop()
2 {
3     SetSensor(IN_1, SENSOR_TOUCH);
4     OnFwdReg(OUT_BC, 50, OUT_REGMODE_SYNC);
5     until (SENSOR_1 == 1);
6     Off(OUT_BC);
7 }
8
9 task main()
10 {
11     touchStop();
12 }
13```
Simple Line Follow with Light Sensor

```cpp
#define LOWERLIGHT 450
#define UPPERLIGHT 500

string light;
int SV;

sub lineFollow();
{
    // Set Sensor type and mode
    SV = Sensor(IN_3);
    if (SV < LOWERLIGHT)
    {
        OnFwd(OUT_C, 65);
        OnFwd(OUT_B, 25);
    }
    else
    {
        if ((SV > LOWERLIGHT) && (SV < UPPERLIGHT))
        {
            OnFwd(OUT_BC, 40);
        }
        else
        {
            OnFwd(OUT_B, 65);
            OnFwd(OUT_C, 25);
        }
    }
}

task main() {
    SetSensor(IN_3, IN_TYPE_LIGHT_ACTIVE);
    // SetSensorType(IN_3, SENSOR_TYPE_LIGHT_ACTIVE);
    while (true)
    {
        light = NumToStr(Sensor(IN_3));
        TextOut(0, 0, light, true);
        lineFollow();
    }
}
```
Curvy Path (Cap the Well)
Resources Online

- http://bricxcc.sourceforge.net/nbc/
  - NXC_Guide.pdf (On CD)
  - NXC_Tutorial.pdf (On CD)
- http://www.legoeducation.us/store/
Have a Great Nebo Day!

Questions: Please contact Mr. Michaud at:
nebomusic@gmail.com
-Or-
www.nebomusic.net