

Institute for Personal Robots in Education (IPRE)

Class Notes for Computer Science 1 with Robots

Title: Introduction to Robot & Myro

Contents:

- A first introduction to:
 - The Myro module
 - Using the Robot
 - Functions
 - Variables

IPRE Class Notes are designed to act as a reference, reminder, and prompt to an instructor lecturing. You may use PowerPoint slides for some or all of your lecture, but ideally you will have a projected display from a computer with Python/Myro installed and an example robot to demonstrate the following concepts live in class. We believe that programming in front of your students is the best way to teach programming. Corrections and additions to these class notes and other educator resource material is appreciated. Please contact Jay Summet (contact info on the www.roboteducation.org website) with corrections or additions.

Start of Class:

- Your class specific reminders, which may include any of the following:
 - Details on where to purchase/obtain the robot/textbook.
 - Scheduling of office hours, introduction of TA's.
 - Your school's policies, honor code, etc...
 - The class syllabus, grading policies, etc...
- What students will learn in this class:
 - How a computer works, and how to control the computer to do work for you.
 - Write small programs.
 - How to debug programs.
 - How to make the robot do what they want.
- We are using Python!
 - An interpreted language
 - No compile/run cycle.
 - Allows for execution of single python statements in the development environment (IDLE).
 - Cleaner syntax than C, more lightweight than Java.
 - Supports Object Oriented Programming, but does not require it.
 - Used in the "real-world" by companies such as:
 - Industrial Light and Magic
 - Walt Disney Feature Animation
 - Google!
 - Yahoo
 - Lawrence Livermore Labs
 - Red Hat
 - NASA
 - National Weather Service
 - Blender 3D modeling program

Optional Motivating Activity: Download the free movie "Elephants Dream" (<http://www.elephantsdream.org>) and show the first two minutes of the movie. Explain that the movie was created (rendered) completely inside of computers (the voice acting was recorded) by volunteers working together on the internet. The primary tool they used was Blender 3D, and open source modeling and rendering tool. Blender uses Python as its internal scripting language. For large scenes with lots of moving parts (such as the scene at the beginning where all of the cables are un-plugging and plugging themselves) all of the motion was scripted using Python. This movie makes the students aware that computers can be used for impressive creative activities and that Python is used in the "real world".

- What is computer science?
 - The study of computation.
 - Theoretical foundations of information and computation and their implementation and application in computer systems.
 - This class will NOT make you into a Computer Scientist! It will however give you a taste for the issues involved in computer science and make you into a beginning programmer.

This class uses Robots:

- More information about the robots, online textbook, etc can be found at <http://wiki.roboteducation.org>.
- Demonstrate the robot:
 - Open the IDLE window with the StartPython icon.
 - Load the myro module with

```
>>> from myro import *
```
 - Initialize the robot with

```
>>>init("COM5")
```


(replacing COM5 with the appropriate outgoing COM port for your computer/robot)
 - Show the students the joystick control

```
>>>joyStick()
```
 - Demonstrate taking a picture and showing it (students really love this one!)

```
>>> myPicture = takePicture()
>>> show(myPicture)
```
 - If you have a USB gamepad plugged into your computer, the **gamepad()** function allows you to control the robot using the gamepad, which is a bit easier than using the mouse with the joyStick interface. The buttons on the gamepad are also mapped to various functions such as beeps and taking/displaying a picture.

You've learned some "programming":

- Importing functions from a module (from the module named myro, import everything)
- Calling a function (init, joyStick, takePicture, show).
 - Functions are names that point to pieces of code (that implement behaviors).
 - When you type the name, followed by and open/close parenthesis () you call or invoke (or run) the bit of code/function.
 - Some functions have no parameters.
 - Other functions require parameters (such as show(aPicture)).
 - Functions can return values (takePicture, getName)
- Variables (myPicture) are names that point to values.
 - You can assign them values using the single equal sign (assignment)
 - (the print command will show the value a variable points to)
 - ```
>>> myVariable = 5
>>> print myVariable
>>> myVariable = 10
>>> print myVariable
>>> robotName = getName()
>>> print robotName
```
- More Robot Examples:
  - forward(speed, duration) - speed is a number from zero to one (0 is no motion, 1 is full speed ahead). Duration is measured in seconds. Floating point values (0.5) are allowed for both speed and duration, so forward(0.5, 2.5) is 1/2 speed for 2 and a half seconds.

- *Note that technically, speed can range from -1 to 1, all negative values make the motor turn in reverse. However, we recommend using only speeds of 0...1 for the first few weeks of class. Tell the students about the backward function to go backwards...*
- backward(speed,duration)
- turnLeft(speed, duration)
- turnRight(speed, duration)
- *Note also that all of these functions can be called with only one parameter (omitting the duration limit), but we strongly recommend that you do NOT show this to students until at least halfway through the course. If you do not specify a duration, the motion will continue until the stop() function is called or some other motion function changes the current direction/speed. Later in the class this can be used to illustrate the difference between blocking and non-blocking calls, for for now, it's much better for the students to enter a speed AND a duration at the same time, making each function call atomic.*
- The beep command can be used to make "music".
- beep(duration, frequency) - *We suggest you try a frequency between 300-5000 to ensure that it is audible. 440 is an A note, 880 is also an A one octave higher. Also, try not to confuse the order of the parameters. A function call of beep(1, 880) provides a 1 second A note, while a call to beep(880,1) produces no audible output for 880 seconds!*
- speak("Any String") - will speak a string (from the laptops speakers) if your speech synthesis software is correctly installed.

## Putting it all together into a program

- So far we have been typing individual commands (mostly function calls) into the IDLE prompt. You can put a sequence of python statements together into a text file (saved with a .py extension so the python interpreter knows that it is a python program) and run them all at once.
- Use IDLE to open a new window (File->New).
- Type a small program into the window:
 

```
forward(1,1)
turnRight(0.5,0.5)
forward(1,2)
turnRight(0.5,0.5)
forward(1,1)
beep(1,800)
```
- Save the program (File->Save) and give it a name such as **Dance.py**.
- Run the program with the (Run->Run Module) command.
  - Show the students that pressing F5 does the same thing.
- If your StartPython icon is set up correctly, your program does not need to have the initialization code at the beginning (assuming that you have already imported the myro module and initialized the robot). However, if the system does a complete re-start when you run a program, you may need to include the "magic" lines at the beginning of your program:
 

```
from myro import *
init()
```
- The students now know how to create, save, and run a simple program.