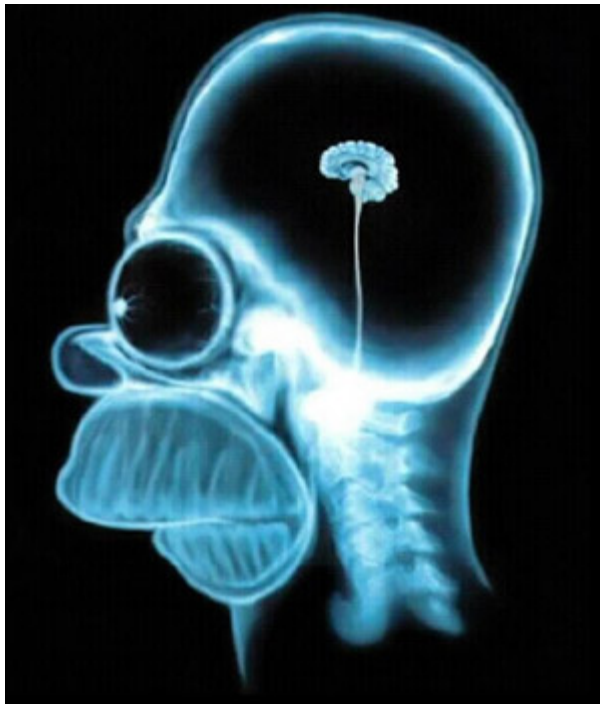
Building Robot Brains

Week #3

Prof. Ryan Kastner

Robot Brains

- When you write a program, you are actually building a brain for your Robot
- In the computing world, this brain is replaceable
  - Firefox as a Browser
  - iTunes as a Media Player
  - VLC as a Movie Player



- By learning to write Robot programs you are also learning to write computer programs

Structure of a Robot Brain

- The basic structure of a Program (Brain) is:

```
def main():
    <do something>
    <do something>
    ...
```

- Every Robot program will begin with

```
from myro import *
init()
```

- Then you import other files/modules or you ask the robot to do something using the modules

Robot Dance

- Make the Robot do some random movements using function *moves()*

```
# The main dance program
def main():
    print "Running the dance routine..."
    yoyo(0.5, 0.5)
    wiggle(0.5, 0.5)
    yoyo(1, 1)
    wiggle(1, 1)
    print "...Done"
main()
```

The function 'print' will print out what you have entered in the double quotes " "
When the main() function is called, the functions in main() are called in order so you get the output:

```
Running the dance routine…
…Done
```

Pythonese

- Try this:

speak("Dude! Pardon me, would you have any Grey Poupon?")

- Python comes with several other useful libraries or modules
- Libraries are made up of sets of functions
- You can import the commands provided in a library
- Every programming language has a set of predefined functions and a mechanism for defining additional functions

Pythonese – Rules

- A name in Python must begin with either an alphabetic letter (a-z or A-Z) or the underscore (i.e. _) and can be followed by any sequence of letters, digits, or underscore letters

  jitterBug2

  my_2_cents

- By giving functions a name you have a way of defining new functions

Pythonese – Rules

- Functions can take parameters that help customize what they do. In the above example, you can issue the following two commands:

>>> yoyo(0.8, 2.5)

>>> yoyo(0.3, 1.5)

- It is better you choose names which makes sense

For example:

A function *turnRight()* should turn the robot right and NOT make it go in circles or dance

PS: This is not mandatory though! ☺

Pythonese – Rules

- Values
  - Designating values by names is an important feature of Programming
  - We can create names for speed, temperature etc.
  - Designating Values

    aveHighTemp = 37

    DowIndex = 12548.30

    myFavoriteRobot = "C3PO"
- *Syntax: <variable name> = <expression>*
  - *Strings are given within double quotes*

Pythonese – Rules

- What you type at the Python prompt >>> is an Expression

```
>>> 5
5
>>> 5 + 3
8
>>> 3 * 4
12
>>> 3.2 + 4.7
7.9
>>> 10 / 2
5
```

- Addition (+), subtraction (-), multiplication (*), and division (/) can be used on numbers to form expressions that involve numbers
- Whole numbers are called *integers* and those with decimal points in them are called *floating point* numbers
- Python handles both (*try 10.0/3.0*)

Pythonese – Rules

- Strings
- Python requires that strings be written enclosed in quotes: which could be single ('I am a string'), double ("Me too!"), or even triple quotes ('''I'm string as well!''')
- Python also provides some operations on strings using which you can write some useful string expressions

```
>>> mySchool = "Bryn Mawr College"
>>> yourSchool = "Georgia Institute of Technology“
>>> yourSchool+mySchool
'Georgia Institute of TechnologyBryn Mawr College'
```

Pythonese – Computation

- Estimate the world population growth in a year and also per day. Given that on January 1, 2008 the world's population was estimated at 6,650,000,000 and the estimated growth is at the rate of +1.14%

Pythonese – Computation

- In a large program if you want to change some value, you can search for the value line by line and modify your program to reflect the new value
- Instead you can use the *input* facility of Python
- Python has a simple input command
  - *Syntax: <variable name> = input(<some prompt string>)*
  - *Population = input("Enter current world population: ")*

Pythonese – Computation

This is how your GUI will look when you use the *input* function

```
>>>
This program computes population growth figures.
Enter current world population: 6650000000
Enter the growth rate: 1.14
World population today is 6650000000
In one year, it will grow by 75810000.0
An average daily increase of 207698.630137
>>> main()
This program computes population growth figures.
Enter current world population: 6725810000
Enter the growth rate: 2.2
World population today is 6725810000
In one year, it will grow by 147967820.0
An average daily increase of 405391.287671
>>>
```

Pythonese – Repetitions

- If you want the dance behavior 10 times

```
for i in range(10):
        dance()
```

- *Syntax for repetition:*

```
for <variable> in <sequence>:
    <do something>
    <do something>
    ...
```

- *range()* function is used to specify a sequence

```
range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Pythonese – Repetition

- What if you want to make the robot dance forever??
- How do you stop it?
- In addition to repeating by counting, you can also specify repetition using time

```
while timeRemaining(10):
    <do something>
    <do something>
    ...
```

Summary

- Values in Python can be numbers (integers or floating point numbers) or strings
- Operations can be performed on Values
- *<variable name> = <expression>*

  This is how Python assigns values to variables
- *range(10)*

  Generates a sequence of numbers from *0..9*
- range(n1, n2)

  Generates a list of numbers starting from n1…(n2-1)

  range(5, 10) will generate the list of numbers *[5, 6, 7, 8, 9]*
- range(n1, n2, step)

  Generates a list of numbers from n1…(n2-1) in steps of step

  range(5, 10, 2) will generate the list of numbers *[5, 7, 9]*

Summary

Repetition

for <variable> in <sequence>:

    <do something>

    <do something>

        ...

while timeRemaining(<seconds>):

    <do something>

    <do something>

    ...

while True:

<do something>

<do something>