Sights & Sounds

Week #8
Prof. Ryan Kastner
More than Numerical Calculations?

- Notion of computation extends far beyond simple numerical calculations

- Using basic computational techniques you have learned so far, you can do computations on shapes and sounds
Have you ever used drawing applications on your computer?

To draw something using Python, you first need a place to draw it:

```python
myCanvas = GraphWin()
```
Sights: Drawing

- To make the window go away, type
  
  ```
  myCanvas.close()
  ```

- To create a graphics window of any size and a name that you specify, type
  
  ```
  myCanvas = GraphWin("My Masterpiece", 200, 300)
  ```
Sights: Drawing

- To change the background color

  ```java
  myCanvas.setBackground("white")
  ```

- You can even try “red”, “blue”, “yellow” or more exotic colors ranging from “AntiqueWhite” to “LavenderBlush” to “WhiteSmoke”
Sights: Drawing

- You know how to create a canvas, and now it is time to draw in it 😊

- You can create and draw all kinds of geometrical objects: points, lines, circles, rectangle, even text and images

- To draw things:
  - You should first create it and then draw it
  - You should also know the coordinate system of the window
Coordinates

- Graphics window with width, $W$, and height, $H$, has $W \times H$ pixels
- The pixel (0,0) is at the top left corner
- The pixel (199, 299) is at the bottom right corner
Then Create and Draw it

- The simplest object is a point

\[ p = \text{Point}(100, 50) \quad \Rightarrow \text{CREATE WITH COORDINATES} \]
\[ p.\text{draw}(\text{myCanvas}) \quad \Rightarrow \text{DRAW IT} \]

- The general form of commands issued on objects

\[ \langle \text{object} \rangle.\langle \text{function} \rangle(\langle \text{parameters} \rangle) \]

- A line requires the two end points specified

\[ L = \text{Line}(\text{Point}(0, 0), \text{Point}(100, 200)) \quad \Rightarrow \text{CREATE WITH COORDINATES} \]
\[ L.\text{draw}(\text{myCanvas}) \quad \Rightarrow \text{DRAW IT} \]
Sights: Drawing

- Can you draw this?

```python
for n in range(0, 200, 5):
    L=Line(Point(n,25),Point(100,100))
    L.draw(myCanvas)
```
Sights: Drawing

- Try these

```python
for n in range(0, 200, 5):
    L = Line(Point(n, 25), Point(100, 100))
    L.draw(myCanvas)
    wait(0.3)
    L.undraw()

C = Circle(centerPoint, radius)
C.draw(myCanvas)

C = Circle(Point(100, 150), 30)
C.draw(myCanvas)
```
Sights: Drawing

- Also try these

You can get the center point of a circle:

```python
centerPoint = C.getCenter()
```

You can specify color for objects:

```python
C = Circle(Point(100, 150), 30)
C.draw(myCanvas)
C.setOutline("red")
C.setFill("yellow")
```
Exercise

- Can you draw this?
# Program to draw a bunch of random colored circles
from myro import *
from random import *

def makeCircle(x, y, r):
    # creates a Circle centered at point (x, y) of radius r
    return Circle(Point(x, y), r)

def makeColor():
    # creates a new color using random RGB values
    red = randrange(0, 256)
    green = randrange(0, 256)
    blue = randrange(0, 256)
    return color_rgb(red, green, blue)

def main():
    # Create and display a graphics window
    width = 500
    height = 500
    myCanvas = GraphWin('Circles', width, height)
    myCanvas.setBackground("white")

    # draw a bunch of random circles with random colors.
    N = 500
    for i in range(N):
        # pick random center point and radius in the window
        x = randrange(0, width)
        y = randrange(0, height)
        r = randrange(5, 25)
        c = makeCircle(x, y, r)
        # select a random color
        c.setFill(makeColor())
        c.draw(myCanvas)

    main()
Drawing Text and Images

- To place text in a graphics window

  \[\text{myText} = \text{Text (<anchor point>, <string>)}\]

- To place an image in a graphics window

  \[\text{myPhoto} = \text{Image(<center point>, <filename>)}\]
Sound

- Try
  
  \[ \text{beep}(1, 440) \]

- This command instructs the robot to play a tone at 440Hz for 1 second

- The letters Hz are an abbreviation for Hertz

- We use Hertz as a unit for specifying frequencies
  
  \[ 1 \text{ Hertz} = 1 \text{ cycle/second} \]
Sound

- The most common use of frequencies is in specifying the clock speeds of computer CPU’s.

\[1 \text{ GigaHertz} = 10^9 \text{ cycles/second}\]

- The human ear is capable of distinguishing sounds that differ only by a few Hertz, as little as 1 Hz. *Try these, can you feel the difference?*

  - `beep(1, 440)`
  - `beep(1, 450)`
Musical Scales

- In western music, a scale is divided into 12 notes (from 7 major notes: ABCDEFG). Further there are octaves.

- An octave in C comprises of the 12 notes:
  C C#/Db D D#/Eb E F F#/Gb G G#/Ab A A#/Bb B

- Frequencies corresponding to a specific note, say C, are multiplier (or divided) by 2 to achieve the same note in a higher (or lower) octave
What is the relationship between these two tones?

\[
\text{beep}(1, 440) \\
\text{beep}(1, 880)
\]

The second tone is exactly one octave the first. To raise a tone by an octave, you simply multiply the frequency by 2.

To make a tone an octave lower, you divide by 2
Musical Scales

- In common tuning the 12 notes are equidistant. Thus if the frequency doubles every octave, each successive note is $2^{1/2}$ apart.

- If C4 is 261.63 Hz, what will be C# (or Db)?

\[
\text{C#}_4/\text{Db}_4 = 261.63 \text{ Hz} \times 2^{1/2} = 277.18 \text{ Hz}
\]
Making Music

- Making songs by frequency is a lot of work

- Luckily, Myro contains a set of functions

- A Myro song is a set of characters composed like so:

  \[ \text{NOTE1} [\text{NOTE2}] \text{ WHOLEPART} \]

  NOTE1 is either a frequency or a NOTENAME, 
  NOTE2 is the same, and optional. Use for Chords. 
  WHOLEPART is a number representing how much of a whole note to play.
Making Music

- NOTENAMES are case-insensitive strings. Here is an entire scale of NOTENAMES:

  C   C#/Db   D   D#/Eb   E   F   F#/Gb   G   G#/Ab   A   A#/Bb   B   C

- 5th octave version can be written as

  C5   C#5/Db5   D5   D#5/Eb5   E5   F5   F#5/Gb5   G5   G#5/Ab5   A5   A#5/Bb5   B5   C6
Making Music

- Try this:

  - c 1
  - c .5
  - c .5
  - c 1
  - c .5
  - c .5
  - e 1
  - c .5
  - c .5
  - e 2
  - e 1
  - e .5
  - e .5
  - e 1
  - e .5
  - e .5
  - g 1
  - e .5
  - e .5
  - e 2
Using a Song

- You need to initialize the robot in a different way
  
  \[ \text{Robot} = \text{scribbler}() \]

- If your song is in a file, you can read it
  
  \[ s = \text{readSong(filename)} \]

- The you can play it on the robot
  
  \[ \text{robot.playSong}(s) \]
  
  Or on the computer
  
  \[ \text{computer.playSong}(s) \]

- You can also \textit{makesong}(text) to make a song
  
  \[ s = \text{makeSong}("c \ 1\; d \ 1\; e \ 1\; f \ 1\; g \ 1\; a \ 1\; b \ 1\; c7 \ 1;") \]