Turtle Graphics In Python
Part I: An Introduction

May 31, 2015
Brian A. Malloy
1. Python Graphics

- Many modules provide graphics:
  - turtle
  - IDLE
  - tkinter – GUI library
  - PyGame – Gaming API
  - PyQt – Nice GUI library
  - PIL – Python Imaging Library
  - PyOpenGL – Bindings to OpenGL

- Simply import them and start using
- Some might need installation
- turtle & IDLE come with Python
2. Turtle Graphics

- Powerful, easy to use package.
- Uses Tkinter for underlying graphics
- A turtle (pen) walks around on a canvas.
- If `pendown()`, turtle draws as he walks
- if `penup()`, turtle moves but doesn’t draw.
- Turtle begins facing right
- Turtles have state:
  - Current position: (x, y)
  - Current facing: left right, up, down
  - Color
  - width
- Documentation:
  [https://docs.python.org/2/library/turtle.html](https://docs.python.org/2/library/turtle.html)
3. A Single Line

```python
1 import turtle
2 window = turtle.Screen()
3 print turtle.screensize()
4 window.setup(300, 300)
5 window.bgcolor("pink")
6 window.title("Draw a Line!")
7
8 ninja = turtle.Turtle()
9 ninja.color("green")
10 ninja.pensize(3)
11 ninja.forward(100)
12 turtle.mainloop()
```
3.1. Line: line-by-line

- Line #1 loads the Turtle graphics package
- Line #2 creates a screen that’s 400x300
- Line #3 prints (400, 300) at the terminal
- Line #4 sets the screen to 300x300 pixels
- Line #5 sets the background color to pink
- Line #6 sets the title of the screen (see top)
- Line #8 creates a turtle, assigns it to Ninja
- Line #9 sets the pen color to green
- Line #10 sets the pen width to 3 pixels
- Drawing begins in the middle of the screen, with the pen facing to the right. Line #11 draws a 100 pixel length horizontal line
- Line #12 makes the program loop
4. Square

```python
import turtle
window = turtle.Screen()
window.bgcolor("white")
window.title("Hello Turtle!")
ninja = turtle.Turtle()
ninja.color("blue")
ninja.pensize(3)
ninja.forward(50)
ninja.left(90)
ninja.forward(50)
ninja.left(90)
ninja.forward(50)
ninja.left(90)
ninja.forward(50)
ninja.left(90)
ninja.forward(50)
ninja.left(90)
ninja.forward(50)
turtle.mainloop()
```
4.1. Square: line-by-line

• Line #2 creates a screen that’s 400x300 pixels; you can print this with:
  
  \texttt{print turtle.screensize()}

• Line #3 paints the screen \texttt{white}

• Line #4 sets the title of the screen (see top)

• Line #5 creates a \texttt{turtle}, assigns it to \texttt{ninja}

• Line #6 sets the color of the pen

• Line #7 sets the pen width to 3 pixels

• Line #8 draws a 50 pixel horizontal line

• Line #9 turns the pen facing up

• Line #15 makes the program loop so we can see something
4.2. A Better Square

```python
1 import turtle
2 window = turtle.Screen()
3 window.setup(300, 300)
4 window.title("Hello Turtle!")
5 ninja = turtle.Turtle()
6 
7 def goLeft():
8     ninja.forward(50)
9     ninja.left(90)
10  
11 goLeft()
12 goLeft()
13 goLeft()
14 goLeft()
15 turtle.mainloop()
```

drawLeft() makes code cleaner & easier to read.
Python Graphics
Turtle Graphics
A Single Line
Square
Almost Centered . . .
Nested Circles
Squares & Star
5. Almost Centered Triangle

```python
import turtle
window = turtle.Screen()
window.title("A Triangle!")
ninja = turtle.Turtle()

def drawLeft():
    ninja.forward(100)
    ninja.left(120)

ninja.penup()
ninja.setpos(-50, -50)
ninja.pendown()
drawLeft()
drawLeft()
drawLeft()
turtle.mainloop()
```
5.1. Almost Centered: Explanation

• We want center the triangle on the screen
• Turtle starts drawing in middle of screen
• We want to move the turtle left and down, so the triangle is more toward the center.
• Line #10 – #12, move w/out drawing:
  – Line #10 – picks the pen up
  – Line #11 – move pen without drawing
  – Line #12 – put pen down so we can draw the triangle
• Why almost:
6. Nested Circles

```python
1 import turtle
2 window = turtle.Screen()
3 ninja = turtle.Turtle()
4
5 def changePosition(x, y):
6    ninja.penup()
7    ninja.setpos(x, y)
8    ninja.pendown()
9
10 def drawCircle(radius, ps):
11    ninja.pensize(ps)
12    ninja.circle(radius)
13
14 changePosition(0, -30)
15 drawCircle(50, 3)
16 changePosition(0, -10)
17 drawCircle(30, 2)
18 turtle.mainloop()
```
7. Squares & Star