CS 1124 Media Computation

16.2

December 10, 2008

Steve Harrison

Exercises

- Create a `DizzyTurtle` class
  - That turns a bit to the left and goes forward when asked to go forward
  - And turns a bit to the right and goes backward when asked to go backward
- Create a `SlowTurtle` class
  - That only goes forward and backward by 50 instead of 100 if you don't tell it how much to go forward or backward
- Create a `StubbornTurtle` class
  - Has a 50% chance of doing what you ask

Based on slides by Barb Ericson,
Georgia Institute of Technology

TODAY

- A brief review
  - Python
  - Java (compared with Python)
- Next Steps
- Survey
- Course Evaluation

TODAY

- A brief review
  - Python
  - Java (compared with Python)
- Next Steps
- Survey
- Course Evaluation

PYTHON

- Scaling pictures
- Sound
- Scaling sound is frequency shifting
- Recursion

PROBLEMS

- Scaling pictures
- Sound
- Scaling sound is frequency shifting
- Recursion

Scaling

- Scaling a picture (smaller or larger) has to do with *sampling* the source picture differently
  - **When we just copy, we sample every pixel**
  - **If we want a smaller copy, we skip some pixels**
    - We *sample* fewer pixels
  - **If we want a larger copy, we duplicate some pixels**
    - We *over-sample* some pixels

Scaling the picture down

```
def copyPictureHalfAsBig( file ):
  # Set up the source and target pictures
  pic = makePicture(file)
  canvasFile = getMediaPath("7inX95in.jpg")
  canvas = makePicture(canvasFile)
  # Now, do the actual copying
  sourceX = 45
  for targetX in range(100,100+((200-45)/2)):
    sourceY = 25
    for targetY in range(100,100+((200-25)/2)):
      color = getColor(getPixel(pic,sourceX,sourceY))
      setColor(getPixel(canvas,targetX,targetY), color)
      sourceY = sourceY + 2
    sourceX = sourceX + 2
  show(pic)
  show(canvas)
  return canvas
```

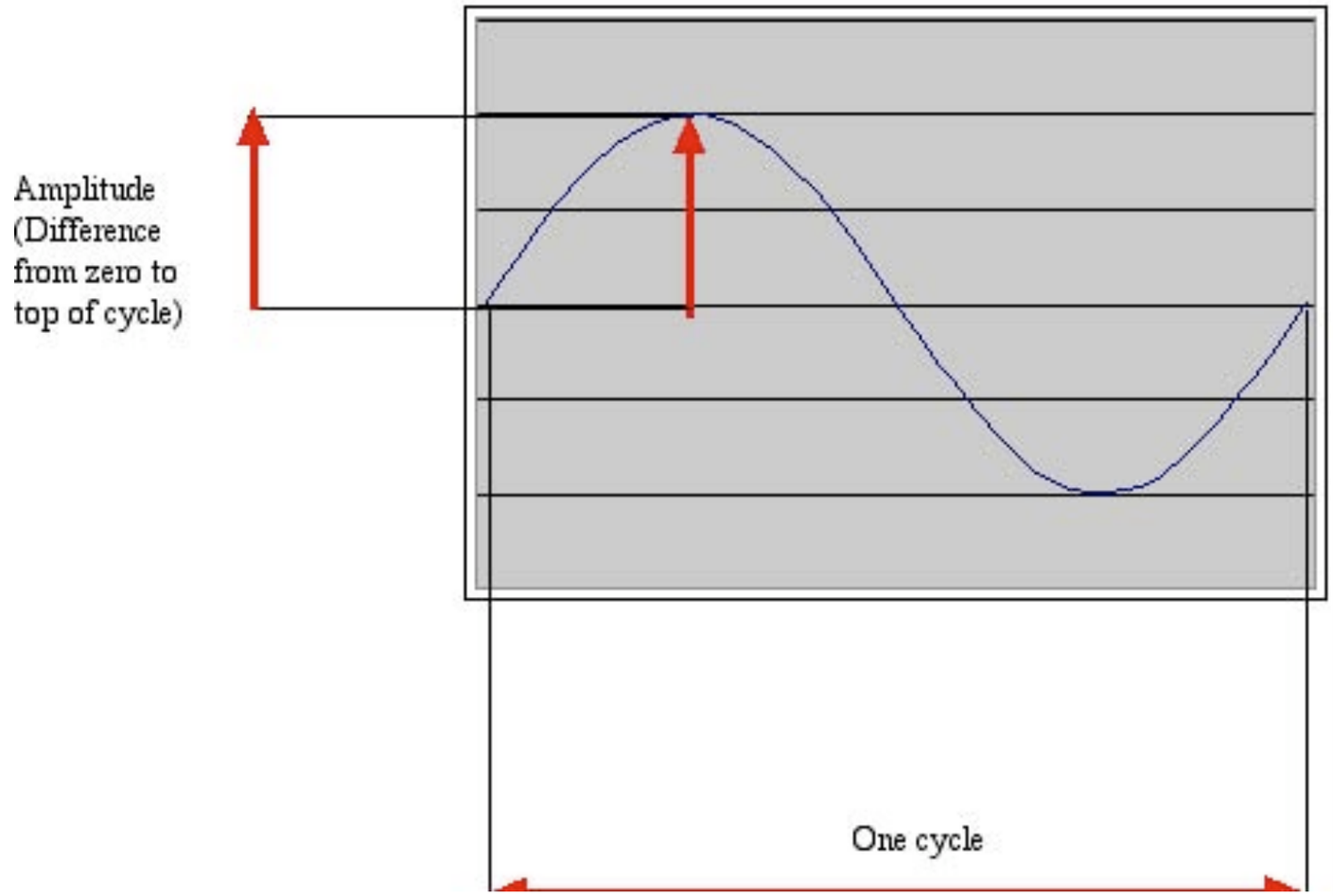
```
>>> barbFile = pickAFile()
>>> setMediaPath()
>>> smallPic = copyPictureHalfAsBig( barbFile )
```

PROBLEMS

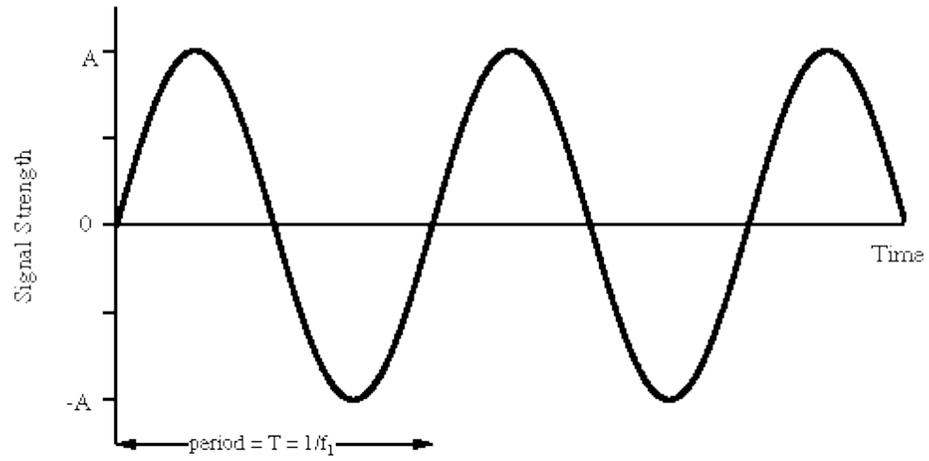
- Scaling pictures
- Sound
- Scaling sound is frequency shifting
- Recursion

How sound works: Acoustics, the physics of sound

- Sounds are waves of air pressure
  - **Sound comes in cycles**
  - **The frequency of a wave is the number of cycles per second (cps), or Hertz**
    - (Complex sounds have more than one frequency in them.)
  - **The amplitude is the maximum height of the wave**



Sine wave



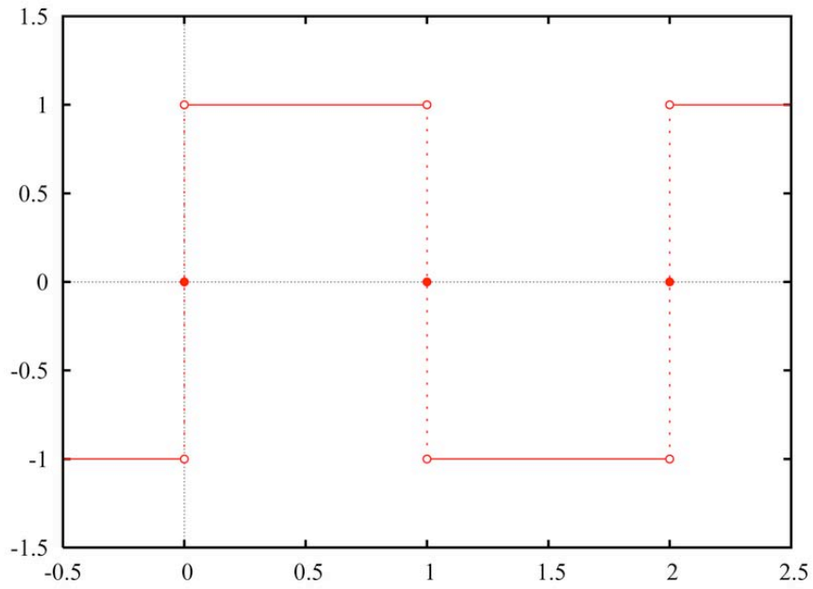
(a) Sine Wave

- recipe 70

```
def sineWave( freq, amplitude ) :
  mySound = getMediaPath("sec1silence.wav")
  buildSin = makeSound(mySound)
  sr = getSamplingRate(buildSin)  # sampling rate
  interval = 1.0 / freq                    # interval of sample
  samplesPerCycle = interval * sr    # samples / cycle
  maxCycle = 2 * pi
  for pos in range( 1, getLength( buildSin ) + 1 ) :
    rawSample = sin(( pos / samplesPerCycle) * maxCycle)
    sampleVal = int( amplitude * rawSample )
    setSampleValueAt( buildSin, pos, sampleVal )
  return buildSin
```

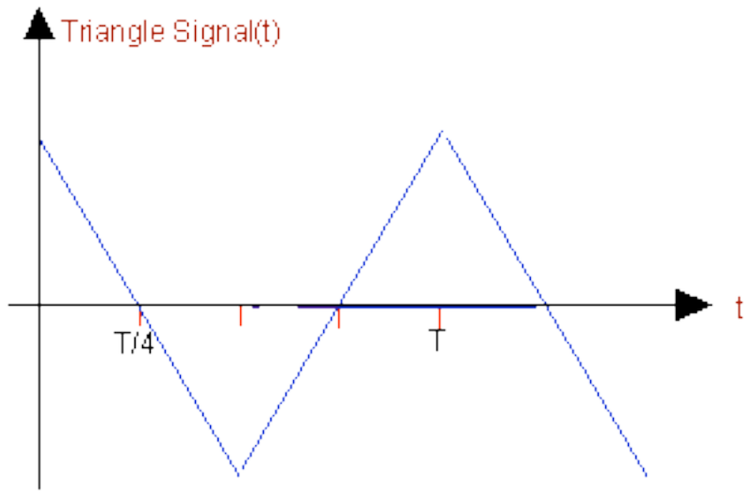
Square wave

- recipe 72




```
def squareWave( freq, amplitude ) :
  mySound = getMediaPath("sec1silence.wav")
  square = makeSound(mySound)
  samplingRate = getSamplingRate(square)   # sampling rate
  seconds = 1
interval = 1.0 * seconds / freq                    # interval of sample
samplesPerCycle = interval * samplingRate # samples / cycle
samplesPerHalfCycle = int(samplesPerCycle / 2)
sampleVal = amplitude
i = 1
for s in range( 1, getLength( square ) + 1 ) :
  if (i > samplesPerHalfCycle):
     sampleVal = sampleVal * -1
     i = 0
  setSampleValueAt( square,s, sampleVal )
  i = i + 1
return square
```

Triangluar wave



- recipe 73, modified

```
def triangleWave( freq ) :
  amplitude = 6000
  samplingRate = 22050                          # sampling rate
  seconds = 1
  triangle = makeEmptySound( seconds  )     # create a sound object (the book uses “sec1silence.wav”)
  interval = 1.0 * seconds / freq               # interval of sample
  samplesPerCycle = interval * samplingRate # samples / cycle
  samplesPerHalfCycle = int(samplesPerCycle / 2)
  increment = int( amplitude / samplesPerHalfCycle )
  sampleVal = -amplitude
  i = 1
  for s in range( 1, samplingRate + 1 ) :
    if (i > samplesPerCycle):
      increment = increment * -1
      i = 0
    sampleVal = sampleVal + increment
    setSampleValueAt( triangle, s, sampleVal )
    i = i + 1
  return triangle                               # return the sound (the book says play)
```

Adding

- recipe 71 (part 2)

```
def addSounds( sound1, sound2) :
  for index in range( 1, getLength(sound1) + 1 ) :
    s1Sample = getSampleValueAt( sound1, index )
    s2Sample = getSampleValueAt( sound2, index )
    setSampleValueAt( sound2, index, s1Sample + s2Sample )
  return sound2
```

Echo, echo, echo, echo

- recipe 64

```
def echoes( soundFile, delay, num ) :
  s1 = makeSound( soundFile )
  ends1 = getLength( s1 )
  ends2 = ends1 + (delay - num)
  s2 = makeEmptySound(1 + int( ends2 / getSamplingRate(s1)  )  )
  echoAmplitude = 1.0
  for echoCount in range( 1, num + 1 ) :
    echoAmplitude = echoAmplitude * 0.6        # each echo is 60% of previous
    for posn1 in range( 1, ends1 ) :
      posn2 = posn1 + (delay* echoCount )
      values1 = getSampleValueAt( s1, posn1) * echoAmplitude
      values2 = getSampleValueAt( s2, posn2)
      setSampleValueAt( s2, posn2, values1 + values2 )
  return s2
```

PROBLEMS

- Scaling pictures
- Sound
- Scaling sound is frequency shifting
- Recursion

Shifting the frequency

- recipe 68, modified
- how sampling keyboards work....

```
def shift( soundFile, factor ) :
  source = makeSound( soundFile )
  target = makeSound( soundFile )
  sourceIndex = 1
  sourceLength = getLength( source )

  for targetIndex in range( 1, sourceLength + 1 ) :
    setSampleValueAt( target, targetIndex, getSampleValueAt( source, int( sourceIndex )))
    sourceIndex = sourceIndex + factor
    if sourceIndex > sourceLength :
      sourceIndex = 1

  return target
```

PROBLEMS

- Scaling pictures
- Sound
- Scaling sound is frequency shifting
- Recursion

A very powerful idea: Recursion

- Recursion is writing functions that call *themselves*.
- When you write a recursive function, you write (at least) two pieces:
  - **What to do if the input is the smallest possible datum,**
  - **What to do if the input is larger so that you:**
    - (a) process one piece of the data (the “head”)
    - (b) call the function to deal with the rest. (“rest”)

Factorial -- the classic recursive function

```
def factorial( number ) :
  if number == 1 :
    return number
  else :
    return number * factorial( number - 1.0 )
```

TODAY

- A brief review
  - Python
  - Java (compared with Python)
- Next Steps
- Survey
- Course Evaluation

PRE-JAVA: JAVASCRIPT

- Hey, remember that stuff we used to make HTML do stuff?

JavaScript vs. Python

	Python	Javascript
write code	*JES*	*text editor*
interpret/ execute	*JES*	*browser (IE, Firefox, ...)*
context	*execute in command area*	*in HTML only*
variables	*just use*	*declare*
blocks	*“:” + indent*	*“{“ + “}”*
termination	*new line*	*“;”*

JavaScript vs. Python

- JavaScript's syntax is much like other programming languages.
- JavaScript can't do everything that Python can.
- Python is a more full-featured programming language.
- But Python can't be embedded inside of HTML.
  - **(Well, not cross-platform. It can on Windows with a Python plugin for your browser.)**

JAVA COMPARED

- Be sure to review the lecture sides!

Python vs. Javascript vs. Java

	Python	Javascript	Java
compile or interpret ?	interpret	interpret	compile
by ?	JES	browser	DrJava
develop & debug in?	JES	text editor - no debugger	DrJava
Command line test?	>>>		code view
performance	slow	not an issue	fast
	Note: there are other Python and Java environments. Later Java course will probably use “Eclipse”.		

Syntax and Appearance

	Python	Javascript	Java
end of line of code?	new line	“;”	“;”
blocks defined	indentation: <tab> or <spaces>	{ ... }	{ ... }
appearance	wide	like HTML: messy, wide, & tall	tall

Variable rules

	Python	Javascript	Java
declare?	no	yes	yes
declaration examples		var x; var x = 10;	int x; int x = 10;
scoping	by class or global	w/in enclosing block	w/in enclosing block
can contain “.” ?	all use “dot” notation to refer to a method in a class so “.” is NOT allowed in a name.		

Functions, Classes, Methods

	Python	Javascript	Java
Functions?	yes	yes	sort of
functional programing	yes	not exactly	no
Classes	allowed, but not required	required, mostly built-in (“document”)	required
Methods	allowed, but not required	required	required
Object-oriented	supported	yes	yes

More syntax

	Python	Javascript	Java
use of "="	varName =	flexible	flexible
		var varName = expression	type varName = expression
			Class varNam = expression
++ / --	allowed	often used	often used
Uniqueness of variable name	var, class, function(), method() overlay	var, class, method, method(params) are different	var, class, method, method(params) are different
		class = consMethod	class = consMethod

Even more objects & methods

	Python	Javascript	Java
defining a Class	class ClassNam(ofClass) :	public class ClassNam { ... }	public class ClassNam { ... }
constructor	def __INIT__.Class(self, arguments) :	public ClassNam() { ... }	public ClassNam() { ... }
polymorphism	all support inheritance and polymorphism so that special methods w/ same name can be created for each class		
bottom-up	pretty good	good	not so good
top-down	OK, but keep it simple	not so good	better

Loops Compared

	Java	Python

For each	for (Pixel pxlObj: pixlArray)
	{
	}
		for pxl in getPixel(pic):

For index in range	for (int i=0; i < pixelArray.length; i++)
	{
	}
		for i in range(0, getLength(pic)+1):

For index in range for source & target	for (int sX=0, tX = 0;
	sX < pixelArray.length;
	sX++, tX++)
	{
	}
		tX = 0
		for sX in range(0,getLength(pic)+1):
		....
		tX = tX+1

Python & Java similarities

swapBackground & chromakey	Java	Python
overall structure	name, init, loop(s)	
loop structure	nested nested	
algorithm	swapBackground: test each pixel to see if same as background only, replace with new background chromakey: test each pixel to see if green+red < blue, replace with new background	

Python & Java differences

swapBackground & chromakey	Java	Python
where	methods in class Picture	not defined as method
arguments	(background, newBackground)	(source, background, newBackground)
return result ?	operate on “this” object	return changed source
syntax	declare variables for loops blocks in curly braces { }	no type declaration for loops blocks indented

JAVA ON THE FINAL

- Take Home Portion
  - Java problem: LeaderTurtle & FollowerTurtle
    - due on Dec 17, @ 2:00 PM
- In 316 McBryde (our regular room) Dec 17, 4:25-6:25
  - Be sure JES works!
  - open book, open computer

TAKE HOME FINAL

- Java Turtle problem
- Create two new kinds of turtles, LeaderTurtles and FollowerTurtles
- FollowerTurtles move whenever Leaders move
- detailed instructions to be posted on website on Thursday

TODAY

- A brief review
  - Python
  - Java (compared with Python)
- Next Steps
- Survey
- Course Evaluation

ADVISING

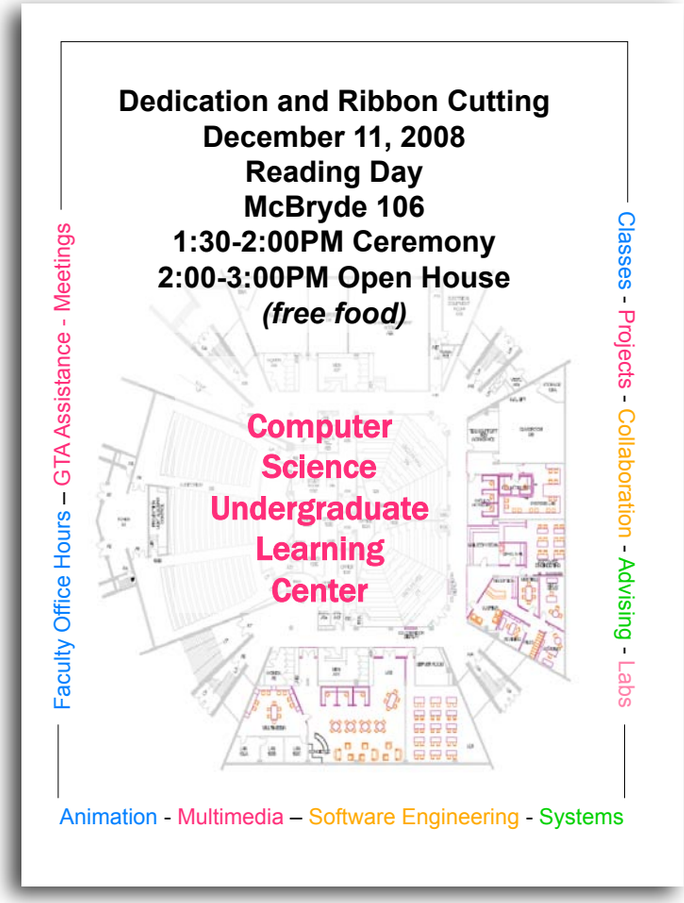
- So this course was kinda cool ...
  - in CS:
    - Soon: Make an appointment with CS advisors in McBryde
    - “Creative Computing Track”
  - out of CS:
    - Computer Music & Multimedia Design (MUS)
    - New Media Theory (A&AH)
    - “Cyberart Studio” (CS, A&AH, ???)

CREATIVE COMPUTING

- Basic CS BS program plus:
- CS Courses:
  - GUI Programming (new course)
  - Multimedia / Hypertext (CS 3624)
  - CyberArt or Game Design capstone (cs 4644)
- Electives:
  - CS 4204 Computer Graphics
  - CS 4634 Design of Information
  - CS 4804 Intro to AI
  - MUS 4??? Computer music & multimedia design
  - Art 4804 New Media Theory
  - Art 2704 Intro to 3D Animation *
  - Art 3704 Topics in 3D Animation*
  - Com 2054 Introduction to Film
  - Com 3194 Film Production
  - Com 4014 Media Effects
  - Com 4034 Functions of Popular Culture

Coming Attractions

- HW 10 - oche
  - –due on Thursday
  - –look at Python echo recipe
    - what is different?
- Thursday (2-3 PM) in 110 McB
  - –Open House
  - –Learn about game design, animation, multimedia, cyberart
  - –FOOD !
- Final, Wednesday 12.17 4:25-6:25 in 316 McBryde



TODAY

- A brief review
  - Python
  - Java (compared with Python)
- Next Steps
- Survey
- Course Evaluation

THANK YOU!

You've been a great class. Good luck on the final.

TODAY

- A brief review
  - Python
  - Java (compared with Python)
- Next Steps
- Survey
- Course Evaluation

COURSE OBJECTIVES

- Learn fundamental manipulations of digital media.
- Learn basic software engineering principles and programming skills.
- Simple data types, control structures, array and string data structures and algorithms, testing and debugging.