CS 1124
Media Computation

Steve Harrison

Lecture 1.2 (August 27, 2008)

Today

- Computer science
- Look at Jython
- Look at some cool image things
- Pixels

Why study CS?

- What is computer science about?
- What do computers *really* understand?
- Media Computation: Why digitize media?
- It is all about communications and process

What is CS about?

- Computer science is the study of "***recipes***"
- Computer scientists study...
  - *How the recipes are written* (***algorithms, software engineering***)
  - *The units used in the recipes* (***data structures, databases***)
  - *What can recipes be written for* (***systems, intelligent systems, theory***)
  - *How well the recipes work* (***human-computer interfaces***)

“Recipes” = Programs

- Key concept: The COMPUTER does the recipe!
- Make them as hard, tedious, complex as you want!
- Crank through a million genomes? No problem!
- Find one person in a 30,000 person campus? Sure.
- Process a million dots in an image or a bazillion sound samples?
    - That’s media computation

"Recipes" for all occasions (er, purposes)

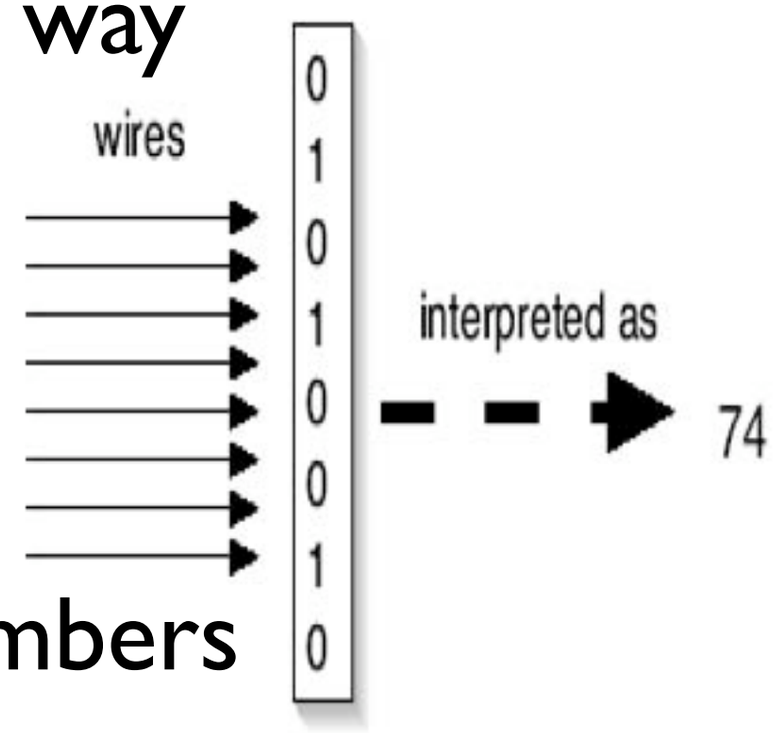
- Some people specialize in crepes or barbeque
- Computer scientists can also specialize on special kinds of recipes:
  - create pictures, sounds, movies, animations (graphics, computer music)
  - still others look at emergent properties of computer "recipes"
  - what happens when lots of recipes talk to one another (networking, non-linear systems)

What do computers understand?

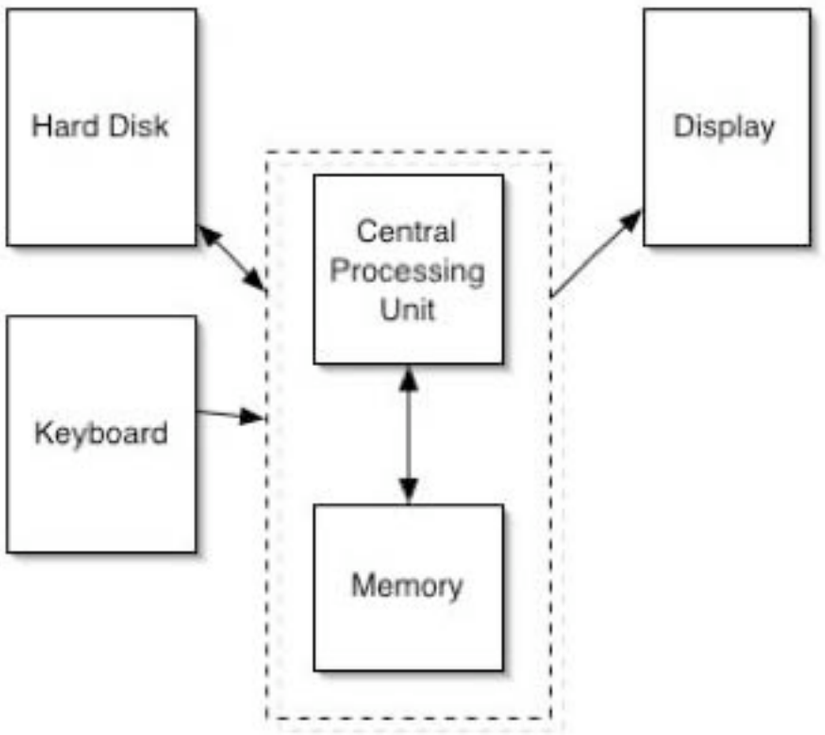
- It's not really multimedia at all.
  - It's unimedia (Nicholas Negroponte)
  - Everything is 0's and 1's
- Computers are not intelligent at all
- The only data they understand is 0's & 1's
- They can only do the most simple things with those 0's & 1's
  - Move this value here
  - Add, multiply, subtract, divide these values
  - Compare these values, and if one is less than the other, go follow this step rather than that one.

Key Concept: ENCODINGS

- interpret these numbers any way we want.
- encode information in those numbers
- Even the notion that the computer understands numbers is an interpretation
- encode the voltages on wires as 0's and 1's, eight of these defining a ***byte***
- can interpret as a decimal number



How do they do it? (Computers, that is)



- adding and comparing happens in the ***Central Processing Unit (CPU)***.
- The CPU talks to the ***memory***
  - Think of memory as a sequence of millions of mailboxes, each one byte in size, each of which has a numeric address
- The ***hard disk*** provides 10 times or more storage than in memory (60 billion bytes versus 512 million bytes), but is millions of times slower
- The ***display*** is the monitor or LCD (or whatever)

Layers of encodings...

- One encoding, ***ASCII***, defines an "A" as 65
- If there's a byte with a 65 in it, and we decide that it's a ***string***, POOF! It's an "A"!
- We can *string* together lots of these numbers together to make usable text
- "77, 97, 114, 107" is "Mark"
- "60, 97, 32, 104, 114, 101, 102, 61" is "<a href=" (HTML)

Hey, what about pictures, video, and MP3?

Layered?

"60, 97, 32, 104, 114, 101, 102, 61" is "<a href="

- A number is just a number is just a number but don't forget the context...
- If you have to treat it as a letter, there's a piece of software that does it
- For example, ASCII associates 65 with the graphical representation for "A"
- If you have to treat it as part of an HTML document, there's a piece of software that does it
- That understands that "<A HREF=" is the beginning of a link
- That part that knows HTML communicates with the part that knows that 65 is an "A"

“Multi”media is “Uni”media

- But that same byte with a 65 in it might be interpreted as...
- A very small piece of sound (e.g., 1/44100-th of a second)
- The amount of redness in a single dot in a picture
- The amount of redness in a single dot in a picture which is a single frame in a full-length motion picture

Software (aka "recipes")

- Computer programs manage all these layers
- How do you decide what a number should mean, and how you should organize your numbers to represent all the data you want?
- That's ***data structures***
- If that sounds like a lot of data, it is
- To represent all the dots on your screen probably takes more than 3,145,728 bytes
- Each second of sound on a CD takes 44,100 bytes!!

Lets hear it for Moore's Law

- Gordon Moore, one of the founders of Intel, made the claim that (essentially) computer power doubles for the same dollar every 18 months.
- This has held true for over 30 years.
- Go ahead! Make your computer do the same thing to every one of 3 million dots on your screen. It won't take much time!

Why *digital* media?

- Digitizing media is encoding media into numbers
- Real media is analogue (continuous). Except for movies and TV ...
- To digitize it, we break it into parts where we can't perceive the parts.
- By converting them, we can more easily manipulate them, store them, transmit them without error, etc.

How can this be?

- Why don't we perceive the breaks?
- We can only do it because human perception is limited.
- We don't see the dots in the pictures, or the gaps in the sounds.
- We can make this happen because we know about
  - physics (science of the physical world)
  - psychophysics (psychology of how we perceive the physical world)

Why should you study recipes?

- To understand better the recipe-way of thinking (called "***algorithmic***" thinking)
- It's influencing everything, from computational science to art (yup, there is something called "algorithmic art")
- It is becoming part of everyone's notion of a liberal education
- That's the process argument
- BTW, to work with and manage computer scientists
- AND...to communicate!
- Writers, marketers, producers communicate through computation

We'll take these in opposite order

Computation for communication

- All media are going digital
- Digital media are manipulated with software
- You are limited in your communication by what your software allows
- Thinking algorithmically will make you a better user of software.

What if you want to do something that Microsoft or Adobe or Apple doesn't let you do?

Programming as communication skill

- want something your tools don't allow, program it yourself
- want to understand what your tools can or cannot do, need to understand what the programs are doing
- If you care about preparing media for the Web, for marketing, for print, for broadcast... then it's worth your while to understand media and manipulation of media
- Knowledge is Power, Knowing how media work is powerful and freeing

Knowing about programming
is knowing about process

- ***Alan Perlis***
- One of the founders of computer science
- Argued in 1961 that Computer Science should be part of a liberal education: Everyone should learn to program.
- Perhaps computing is more critical to a liberal education than Calculus
- Calculus is about rates, and that's important to many.
- Computer science is about process, and that's important to everyone.

A Recipe is a statement of process

- A recipe defines how something is done
- In a programming language that defines how the recipe is written
- When you learn the recipe that implements a Photoshop filter, you learn how Photoshop does what it does.
- And that is powerful.
- **Programming is about Communicating Process**
  - A program is the most concise statement possible to communicate a process
  - That's why it's important to scientists and others who want to specify how to do something understandably in the most precise words as possible

Python

Our (first) programming language

- The programming language we will be using is called Python
  - Python— was invented by Guido van Rossum & researchers across the Internet
  - http://www.python.org
  - It's used by companies like Google, Industrial Light & Magic, Nextel, Disney Animation, and others
- The kind of Python we're using is called Jython
  - It's Java-based Python
  - http://www.jython.org
- We'll be using a specific tool to make Python programming easier, called JES.
  - Yeah, we did invent that one

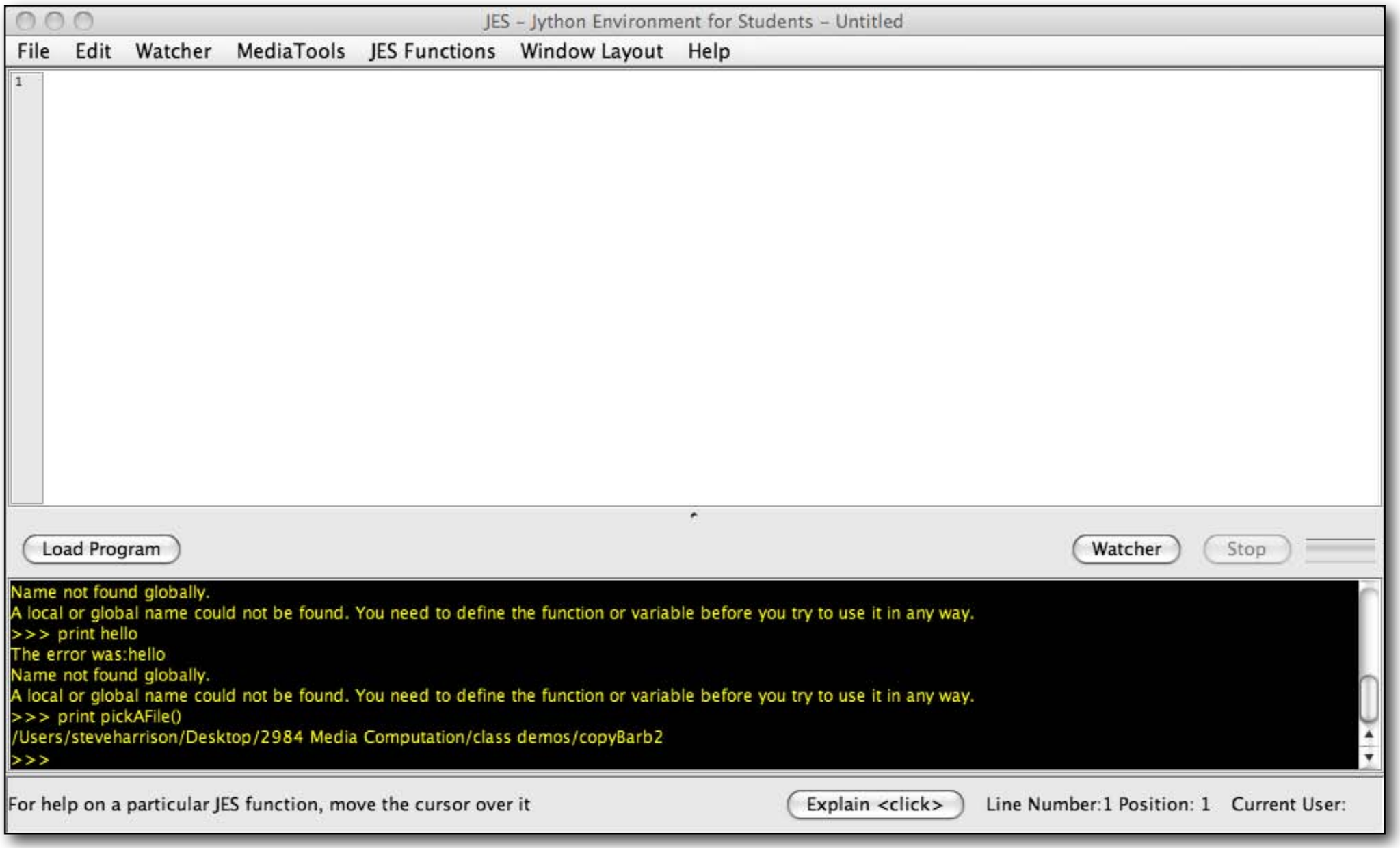
A note to those with previous experience...

- We start with functional programming
- Move to object-oriented programming in 10 weeks
- Jython (our version of Python) is built on Java so uses Java classes and methods
- Jython is more foregiving than Java
  - implicit rather than explicit typing
  - variable scoping not an issue
  - global name-space; names of functions, classes, objects can only be re-used; parameters are NOT part of name

If you did not understand this, don't worry. It was not meant for you.

- We start with functional programming
- Move to object-oriented programming in 10 weeks
- Jython (our version of Python) is built on Java so uses Java classes and methods
- Jython is more foregiving than Java
  - implicit rather than explicit typing
  - variable scoping not an issue
  - global name-space; names of functions, classes, objects can only be re-used; parameters are NOT part of name

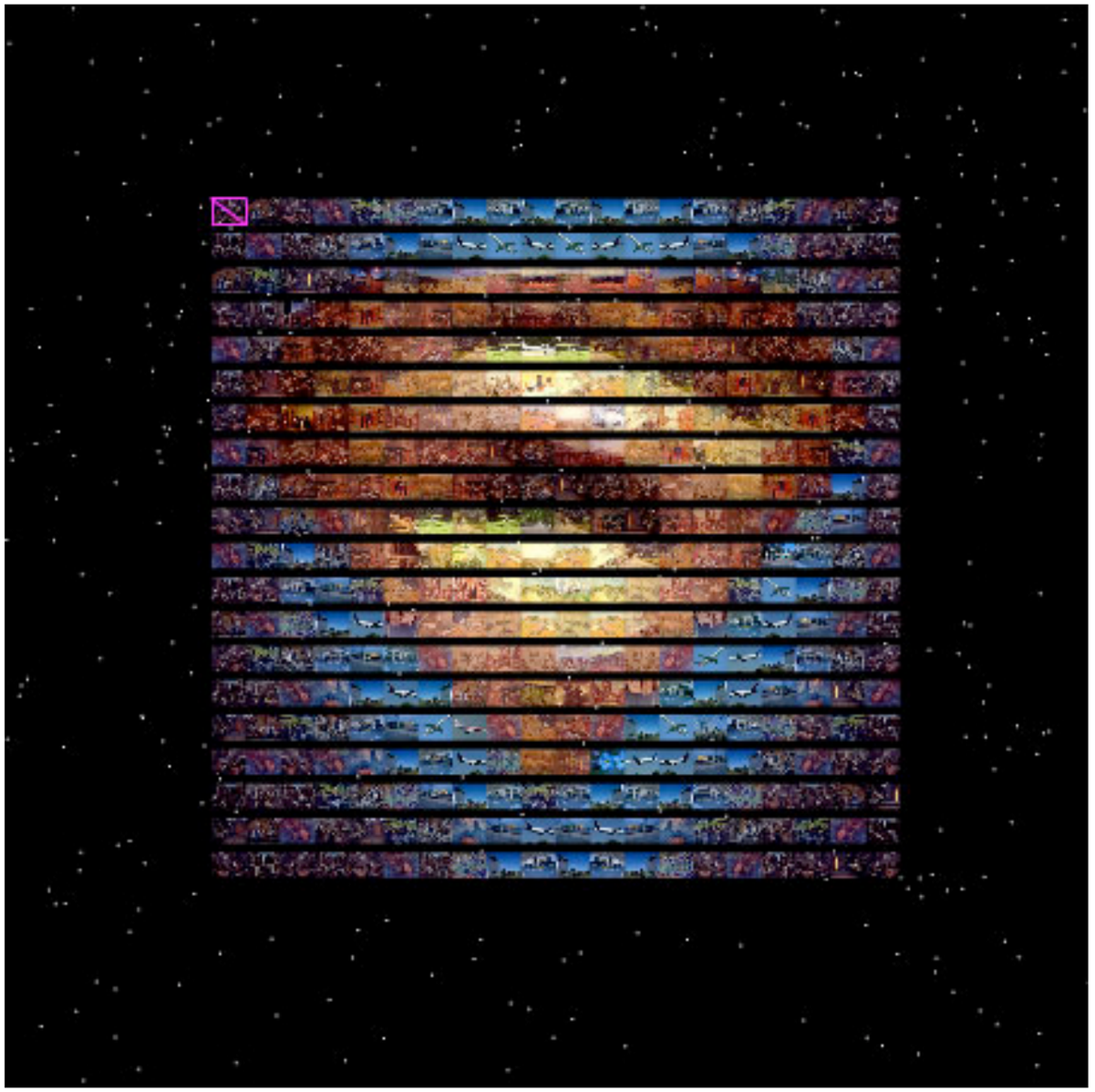
Demo of Jython



The Wooden Mirror

- Video
- How does it work?
- Color ?
- *Look up "DLP"*

How about a picture of pictures? (mosaics)



Digitizing pictures as bunches of little dots

- We digitize pictures into lots of little dots
- Enough dots and it looks like a continuous whole to our eye
  - **Our eye has limited resolution**
  - **Our background/depth acuity is particularly low**
- Each picture element is referred to as a *pixel*

Pixels

- Both the Wooden Mirror and the mosaic images are made of pixels
  - *Wooden Mirror* pixels are squares of wood flipping back and forth
  - mosaic images are small pictures, each the same size
- How many pixels are needed to make picture you understand? See www.guimp.com (its 16 x 16 pixels)

Pixels

- Pixels are *picture elements*
  - Each pixel object knows its **color**
  - It also knows where it is in its **picture**

Coming attractions

- Read chapters 1 & 2
- Online Quiz 1 due on Friday at 2:00 PM
  (the quiz is on the reading)
- Friday is in 1080 Torgersen! NOT HERE.
- For Monday:
  - read Chapter 3
  - Quiz 2 due Monday @ 10:00 AM