**Announcements**

- All your clickers should work now. Let’s sync.
- Midterm moved to Tues Feb 12th to allow homework on truth tables, Huffman coding etc (due Feb 5th) to be graded and returned before the midterm
- Sections: you may need to go twice some weeks!

**More Announcements: Syllabus**

More than one homework a week. Broken into smaller pieces.

**Key ideas from last time**

- By 1943, it had been shown that computers could be used for which of the following tasks …???
  - A. Counting, Arithmetic
  - B. Translation of language
  - C. Logic.
  - D. All of the above.
  - E. None of the above

**More Key Ideas**

- Representation of data on computers:
  - Presence or Absence of signal: 0 and 1
  - Signal can be mechanical, electrical, color, light, blinking an eye, moving a finger, yes, no, true, false, 0 1
  - Hollerith tabulating machine: mechanical => electrical

**Abstraction ...**

- Formulating blocks of computation as a “concept” is functional abstraction
- What we did is important and we do it all the time in everyday life (otherwise we couldn’t deal with life’s complexity)...
  - We break a task down into (one or more) subtasks
  - We solve a subtask using a sequence of instructions
  - We put the solution into a function “package”, gave it a name, “process a riser,” and thus created a new thing, a concept, something we can talk about & use
  - Then we used it to solve something more complicated ... and probably repeat this approach at the next higher level
- This lets us do more complicated things
Recursion is when

- A. You call the same function as many times as you want.
- B. A function calls itself
- C. You use functional abstraction.
- D. None of the above.

Sierpinski: function call when mouse pressed

Processing Basics: Recursive Circles

Why does recursion work?
Size of ‘problem’ gets smaller on each call.
Stopping condition.
Will go back to recursion for creativity assignment.

Hollerith’s Tabulating Machine

To Process Data

- A mechanical machine can “read” a card with ... a “metal brush”
Sensing Punch Allows Some Action

- When the brush touches the source of current, the circuit closes,
- The electrical impulse can cause a mechanical action to happen that gives an instruction or records data

A General Idea

- Digital Information: Detecting the presence or absence of a phenomenon at a specific place and time: PandA
- Phenomena: light, magnetism, charge, mass, color, current, ...
- Detecting depends on phenomenon – but the result must be discrete: it was detected or not; there is no option for “sorta there”

Making Data Digital

- “Digitizing” makes information discrete, it’s either there (1) or not (0), (not sorta there)
- A machine can determine that fact using mechanical or electronic means.
- Once data is digital, it can be processed in many different ways
- Processing power can grow and grow

McCulloch & Pitts Neuron

\[ \text{Output} = \begin{cases} 1 & \text{if Input} > \text{Threshold} \\ 0 & \text{otherwise} \end{cases} \]

Truth Table for And (using True and False)

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P and Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

Truth Table for And (using 0 and 1)

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P and Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
These are the fundamental units of computers:

- **AND**
- **OR**
- **NOT**
- **NAND**
- **XOR**

Just like the neurons, but built out of transistors:

- **AND**
- **OR**
- **NOT**
- **NAND**
- **XOR**

---

So computers can do logic, using simple gates (like neurons) for each logic function. But there are more key ideas.

---

**Bytes: Standard encodings of meaning**

- A byte is eight bits treated as a unit
  - Adopted by IBM in 1960s
  - A standard measure ever since
- Bytes encode the Latin alphabet using ASCII -- the American Standard Code for Information Interchange

---

**Bits & Bytes**

- P and A is a binary representation because it uses 2 patterns
- Bit -- it’s a contraction for “binary digit”
  - a position in space/time capable of being set and detected in 2 patterns

---

ASCII chart:

<table>
<thead>
<tr>
<th>ASCII</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100 0010</td>
<td>0</td>
</tr>
<tr>
<td>0100 0011</td>
<td>A</td>
</tr>
<tr>
<td>0101 0001</td>
<td>C</td>
</tr>
<tr>
<td>0101 0010</td>
<td>T</td>
</tr>
</tbody>
</table>

---

1/15/13
UTF-8: All the alphabets in the world

- Uniform Transformation Format: a variable-width encoding that can represent every character in the Unicode Character set
- 1,112,064 of them!!!
- UTF-8 is the dominant character encoding for the World-Wide Web, accounting for more than half of all Web pages.
- The Internet Engineering Task Force (IETF) requires all Internet protocols to identify the encoding used for character data
- The supported character encodings must include UTF-8.

Encoding Information: There's more!

- Bits and bytes encode the information, but that’s not all
- Tags encode format and some structure in word processors
- Tags encode format and some structure in HTML
- Tags are one form of meta-data
- Meta-data is information about information

Meta information

Can tell the web browser about the meaning or type of the information

```html
<font face="calibri">

<a href= "downloads.html"> DATA and SOFTWARE DOWNLOADS </a>

</font>
```

href tells the web browser that downloads.html is a file (another web page)
More kinds of meta data all the time

- “The semantic web”
- An “ontology” i.e. “taxonomy” of the kinds of things there are in the world
  - People, place, thing, animal, organization, country
- New tags within web pages
- Could be put there by people
- Could be programatically identified by text processing algorithms (like what Watson Jeopardy uses)

Representing Information: Summary

- Bits encode numbers using binary representations
  - 11 01 10 00
- Bits encode letters using ASCII for North American and Western European languages
  - 1110 0111
- Bits can be combined with other bits to do logic
  - This suggests an principle we will soon argue:
    - All information can be represented with bits

First: How did Symbolic Lightbot go?

- HW2. Lightbot. Was due today
- First Processing homework. Due next Tuesday
- Download processing from www.processing.org. Do the tutorials

Recursion is when

- A. You call the same function as many times as you want.
- B. A function calls itself
- C. You use functional abstraction.
- D. None of the above.

Key idea: Functional Abstraction
The Function Becomes A Concept

- Because F1() “processes a riser,” I think of the programming task as
  - With F1() as a concept, I simplify the programming to just 9 steps rather than 21 OR
  - It also suggests another concept:
    - Move_to_next_riser()
Now we will use these ideas in Processing ...

- Processing is a language for programming graphical and image-based computations
- Fun!
- Easier to do because we “see” what’s happening
- Immediate feedback => bottom up programming style

Get It: http://processing.org/download/

- If you have a personal computer, then grab a copy of the Processing system and put it on your machine
- You will want “Windows” or “Mac” versions
- Following installation instructions ... it takes less than 5 minutes and then you can work on your own computer!

What You See

- When you start up the Processing system...

Add Some Code

- Type in instructions that you will learn shortly
- Then run your program

What is a color in processing?

- A sequence of 0’s and 1’s....
- Red, Green, Blue
- Each color element is represented by one BYTE (8 bits).
- http://processing.org/learning/color/
- So how many shades of Red (Green, Blue) are there?
  A. 8
  B. 16
  C. 256
  D. 512
The Color Purple

- Colors in most Web programming are given as three values: RGB, for red, green, blue
- The Color Purple, for example, is: \(128,0,128\)
- These positions are the intensity of the little lights that make up a pixel on the screen
  - The least intensity is 0, that is, off
  - The greatest intensity is 255, maximum brightness
- Amazingly, the three max RGB colors make white
- So, purple is \(\frac{1}{2}\) intensity of red, no green, and \(\frac{1}{2}\) intensity of blue — makes sense

RGB Color: http://processing.org/learning/color/

Questions about “Iron Rich Snow”

- The angel is on a rust-colored background specified by: `background(192, 64, 0);` ...
- which means?
- Stroke sets line color: `stroke(255,255,255);`
- Suppose the angel is “iron rich” and the snow white
- Fill sets color of object: `fill(128, 0, 128);`

Simple Shapes Make Robots

- Reas and Fry, in their book, show us a cute robot they programmed using simple shapes
- They give their code and we can see how they built it
- To make the point that all code must "make sense" – it’s not gibberish – lets look at it even though we don’t know Processing yet
Looking At Simpler Code

- Drawing a snow angel is straightforward ...

```java
void setup ()
{
  size(400,400);
  background(152,152,152);
}

void draw ()
{
  if (keyPressed)
  {
    stroke(255,255,255);
    fill(255,0,0);
    ellipseMode(RIGHT)
  }
  else
  {
    stroke(120,0,0);
    fill(255,0,0);
    ellipse(150,55,45);
    fill(200,200,200);
    rect(225,205,235);
  }
}
```

Red or White?

This program draws a
A. red rectangle
B. white rectangle

```java
rect(10,10,30,40);
fill(255,0,0);
```

The Color Purple

- Colors in most Web programming are given as three values: RGB, for red, green, blue
- The Color Purple, for example, is: 128,0,128
- These positions are the intensity of the little lights that make up a pixel on the screen
  - The least intensity is 0, that is, off
  - The greatest intensity is 255, maximum brightness
  - Amazingly, the three max RGB colors make white
- So, purple is ½ intensity of Red, no Green, and ½ intensity of Blue ... makes sense
Questions about “Iron Rich Snow”

- The angel is on a rust-colored background specified by: `background(192, 64, 0);` ...
  which means?
- Stroke sets line color: `stroke(255, 255, 255);`
- Suppose the angel is “iron rich” and the snow white
- Fill sets color of object: `fill(128, 0, 128);`

It’s All The Same

- When the values for RGB are all the same, it’s some color of gray, or white, or black
- Since writing `background(255, 255, 255)` is kind of a drag, Processing allows us to give just one argument; so `background(255)` is equivalent to giving all three 255s
- What colors are these backgrounds?
  - `background(255, 0, 0);`
  - `background(64);`
  - `background(0, 0, 64);`

Simple Shapes Make Robots

- Reas and Fry, in their book, show us a cute robot they programmed using simple shapes
- They give their code and we can see how they built it
- To make the point that all code must “make sense” – it’s not gibberish – lets look at it even though we don’t know Processing yet

Robot Code, 1

```java
// Reas and Fry, in their book, show us a cute robot
// they programmed using simple shapes
// They give their code and we can see how they built it
// To make the point that all code must “make sense” – it’s not gibberish –
// lets look at it even though we don’t know Processing yet
```

Robot Code, 2

```java
// Reas and Fry, in their book, show us a cute robot
// they programmed using simple shapes
// They give their code and we can see how they built it
// To make the point that all code must “make sense” – it’s not gibberish –
// lets look at it even though we don’t know Processing yet
```

Knowing Only About Color …

- We “improve” the robot by adding some color
- Then we make it move!
- Detailed instructions in homework PDF

Just Do It!
A Quick Comment on Processing

- We have written two kinds of Processing programs –
  - static, which only draw a picture
  - dynamic, which keep drawing a picture

If there is time, start on Privacy lecture.