Correction

- If \( \text{instr}[pc] == 'r' \):
  - \( y \):
    - \( \text{temp} = \text{ydir} \)
    - \( \text{ydir} = \text{xdir} \)
    - \( \text{xdir} = -1*\text{temp} \)
  - \( n \)

- If \( \text{instr}[pc] == 'l' \):
  - \( y \):
    - \( \text{temp} = \text{xdir} \)
    - \( \text{xdir} = \text{ydir} \)
    - \( \text{ydir} = -1*\text{temp} \)
  - \( n \)
We’ve ripped right along, and learned a lot!

What We’ve Learned So Far

Lawrence Snyder
University of Washington, Seattle
Announcements

- The Midterm is Monday –
  - One sheet (8.5” x 11”) of notes, handwritten OK
  - No other materials except pencil & eraser – no phone, calculator, computer, books, etc.
  - The test is on paper ... this means that program text – int, for, mouseX – will not be highlighted
Information is physically represented by the presence or absence of some phenomenon at a specific place or time –

- **Properties:** discrete, be able to set it / detect it
  - Give examples of these properties

- **Direct use of properties:** winking; dog barking;
  - Other examples ... abstract all as binary

- **Numbers represented in binary**
  - Count, add, covert to/from decimal, hex

- **Letters represented in ASCII, UTF-8**
  - Difference between 0011 0111 and 0000 0111?
Abstraction

- Functional abstraction: find a sequence of operations that perform a “meaningful” operation: package them w/name, parameters and precise specification
  - Ex 1: Lightbot 2.0; std name, no params, 8 insts.
  - Ex 2: Symbolic Lightbot: name, no params, $n$ insts.
  - Ex 3: Processing: name, params, $n$ instructions
- When functionally abstracting you create a concept – a new idea (w/ name & meaning) to use without “worrying about the details”
To Abstract ...  

- To abstract (in CS) is to “extract the rule or pattern” from a process or situation:
  - To draw three squares in a row, we write
    - \( \text{rect}(x, y, s, s) \);
    - \( \text{rect}(x + s, y, s, s) \);
    - \( \text{rect}(x + 2s, y, s, s) \);
  - We saw Pacman change
    - \( \text{arc}(\text{mv}, 100, 80, 80, \text{radians}(50 - 2 \times (1 + \text{mv} \times 25), ...)) \);
- You have been learning to abstract in this sense, too

But when we **abstract**, we see that this is really \( \text{rect}(x + j \times s, y, s, s) \) for \( j=0,1,2 \)
Privacy & Social Use of I’net

- Privacy – right of people to decide the extent they reveal information about themselves
  - Discussion: Revealing other people’s information
  - Guidelines for public discourse on I’net: Offensen...
  - Five guidelines for posting on social network
Programs that can approximate an artist’s work through random numbers give a different perspective on classic questions: “What is the artist’s contribution to Art?”
Properties of Computation

- Composed of commands or instructions
- Presented in sequence; executed in sequence
- Commands direct an agent
- Keeping track of the “current instruction” in the sequence: yellow box in Lightbot; program counter for computers
The execution sequence can be interrupted to execute a function – suspend current sequence, go to function definition, initialize parameters, run – so that when complete, return is to point of suspension.

Conditional commands (If-statements) skip instructions; looping repeats instructions.
Programming ... the ideas

- Give examples of these ...
  - Declarations
  - Data types
    - Also, examples of specific data types like float
  - Expressions
  - Assignment statements
  - If-statements
  - For-statements
  - Function definitions and calls
    - Also, distinction between value-returning and void

These ideas are in ALL programming languages; we give our examples using Processing.
We have learned how to read code

What do these do …?

```plaintext
x = (x + 1) % 5;
float y = 12;
if (age > 18) {
    vote = "yes";
} else {
    vote = "no";
}
for (int j = 0; j < 10; j = j + 1) {
    rect(10 + 20 * j, 50, 10, 10);
}
ellipse(mouseX, mouseY, 20, 20);
```

```plaintext
float x = 100;
float y = x % 10;
if (x == 10) {
    y = y + 1
}
What are x, y after this code?
```

```plaintext
for (int j = 0; j < 10; j = j + 1) {
    w = j;
}
What values does w get assigned?
```
Computer Structure

- Computers have five main parts...
- Computers are “instruction execution engines”
- The “engine” is the fetch/execute cycle with its infinite loop of five operations
- What does instruction ADDB 30,40,50 mean?
- How often must a computer reference its memory to execute an instruction?
Many technological advances have come since WWII, but only a few are ‘game changers’ ... what are examples?

The invention of Integrated Circuits was a game changer because to properties, which were?

http is a language computers speak to each other that enables what?
The number 77 can be written in binary ... what is it?
The binary number 0010 0010 0100 is what decimal number?
The hexadecimal number B4 1E is what in binary
Given binary 0101, 0101 what is their sum
UTF-8 is a standard character encoding that uses 1-4 bytes to encode characters ... why do we need it when we have ASCII?
State the Bias-free Universal Medium Principle

- Give three examples that illustrate this for
  0000 0000 0000 0000 1111 1111

- Functions can be built from primitive instructions or other functions ... which allows problems to be solved using ...?
  - Illustrate the point by describing a computation developed in class or an assignment
I have taught a LOT of capabilities and information – and you have worked hard to learn it! Good going!!

What questions remain?