Final Review Topics Spring 2012 ICSI 201
Chromakey—Lab 07

• Textbook section: HOW DOES IT WORK??

• Loop to get at Pixels and their values in one Picture, get at their color numbers
  – And get at Pixels in another Picture

• If statement to decide whether to change the color of a Pixel
  – Lecture and Textbook material on if, and if/else statements.

• GE methods to get at Pixels, get a Pixel's color, and change a Pixels color.
Method and Loop to rastorize lines..Lab 08

- COMBINE loop to rastorize a line (actually put a line in a Picture, Pixel by Pixel) with making a parametrized instance method to add to the Picture class.

- The bonus part included mathematical calculations of the coordinates of which Pixels to color. Everyone saw that when we made a ball in live coding!
Distinctions expressed by nesting

- One computation:

  ```
  for( .. ; .. ; .. )
  {
    DO SOMETHING before the inner for loop
    for( ... ; ... ; ... )
    {
      DO SOMETHING like blacken with ONE Pixel
    }
  }
  ```
Distinctions expressed by nesting

- A different computation:
  
  ```
  for( .. ; .. ; .. )
  {
    DO SOMETHING like blacken with ONE Pixel
  }
  (I just moved one little } UP 5 lines!)
  ```

Boolean and other values

• Boolean value means the set consisting of `true` and `false` (only two values in the set!)

• Only expressions with Boolean value are allowed in the ( )'s of `if` and `while` statements.

• Other primitive Java types: `int` and `double`
  – What they are good for: For example, ints but not doubles for array index values
  – Examples of the many elements in their sets:
    • Typical numbers without decimal points
    • Typical numbers with decimal points.
Boolean values

• Expressions for computing Boolean values
  – Example:
    ```java
    boolean X;
    X = ( sc.nextInt() > 100 );
    ```

• Boolean variable to control a loop
  – Example:
    ```java
    boolean keepShopping = true;
    while( keepShopping )
    {
        /* Get input */
        if( <a thing to buy> )
        {
            ... } else { keepShopping = false; }
    }
    ```
Loops, calculations, if statements

• Making a method to draw a ball on a Picture
• Parameters
• Loops
• Calculations (is the pixel in the ball? What should it's color be? Compute the color as a weighted average)
• If statements
• Developed in lecture LIVE!
Chapters 4, 5, 6

- The heart of the subject in the context of digital images (Pictures)
- Locating data in
  - Pictures by x,y location (2 indices) of Pixels
  - Arrays by one index.
- Calculating from and/or changing/storing data in an Picture or an array
- Loops
- Calculations
- Conditionals
- COMBINATIONS of Loops, calculations, conditionals
Blur and edge detection are separate examples from Chapter 6 (modifying Pixels)

Edge detection ALONE:
- is TWO STEPS for each pixel
  - Compute absolute difference between brightness of Pixel and the Pixel above it
  - Detect line if the difference is big, don't detect if small
- Albany idea: SEE the result of EACH STEP SEPARATELY

COMPARE:
- Edge detection in original picture
- Edge detection in blurred original
Intro to Image Processing: Proj 03

- Blur and edge detection are separate examples from Chapter 6 (modifing Pixels)

- HOW DOES EDGE DETECTION WORK?
  - Explain in English and/or example of the calculation, NOT by regurgitating code.

- HOW DOES BLURRING WORK?
  - Explain in English and/or example of the calculation, NOT by regurgitating code.
Java Programming Lesson of Proj 3

Java static (also called class) methods are a good choice when the method does not have an obvious OBJECT to be CALLED ON.

Static methods cannot use “this” within their bodies...that makes no sense because “this” refers to the OBJECT the method is CALLED ON.

In Proj 03: You made the P3 class to contain a static method, like the Math class.
Methods returning values (or not)

Whether a method be static or instance, you might design it to either return a value or NOT return a value.

Topic: HOW to code (1) using the return value (easy: USAGE ( methodCall(...) )
(2) what value to return (easy: put it in the return statement in the method body)
Voting Machine Labs—practice coding solutions to those problems!

• User interface loops
  – Prompting for input
  – Reading input into a variable
  – if's to decide what to do

• Counting different things, like votes for different people, using simple int variables and, BETTER, arrays of int variables.

• Loops to print, process, search information stored in an array.
House, Primary, Student, Album, Bill classes

• Introduction to making your own classes:
  Textbook chapter 11
• The Mad Ph.D. Video:
  – A class as a BLUEPRINT for making objects, NOT an object itself!
  – What the new operation does.
  – What the new operation returns, and how to use what it returns
• Making your own class continues with Album of Project 04.
iClicker question
What does the Java new operation return?

(A) News that a new object was built.
(B) An actual new object.
(C) An address, location or reference to a new object.
(D) An address, location or reference to an old object that was built before the new operation was started.

See "Factory Pattern" for an alternative to new
What can your program the computer to do with the address, location or reference that a new operation just returned after it made an object?

(A) Print the address.
(B) Call a method ON the object.
(C) Save or copy the address into a reference ticket or variable.
(D) Call a method with the address as a parameter.
(E) B, C, and D but NOT (A)
Project 04

- Practice making and using **fields**, which are **variables inside objects**.
- Constructor methods
  - Study Ch. 11
  - Different polymorphic forms (See Ch. 11)
  - relationship with new
  - What they are good for: Force field values to start out right.
- Method bodies coded with if/else to do **DIFFERENT THINGS** in **DIFFERENT USE CASES**. They depend on field values!
Project 04

- Method bodies coded with if/else to do DIFFERENT THINGS in DIFFERENT USE CASES. Very realistic application of if statements, alongside chromakey, circle-making, detecting a new min or max, etc. KNOW the difference between conditionals and loops.

  - Please STUDY, trace examples from book(s), etc. EXACTLY what these mean:

    ```
    if ( TEST ) { BODY }
    if ( TEST ) { if-PART } else { else-PART }
    while( TEST ) { BODY }
    for( INIT ; TEST ; INCR ) { BODY }
    ```

  Hint: Write the 4 flowcharts on your note sheet!
Project 04

- Accumulating data from step after step after step:
  - Track where to start copying the next Picture (nextX), after copying the previous.
  - Count of Pictures given and where to save next one in an array (nPictures)
  - Sum of the widths
  - Maximum of the heights

- Also illustrated in the Bill class (live coded)
Project 04

- Coding to an interface SOMEBODY ELSE (the Prof!) gives you
  - Class MUST be named Album
  - It MUST have Album(int,int) and Album(int) constructors. (Each one earns you 50%)
  - It MUST have addPicture(Picture) and getResult() methods, that work as specified.
  - An example (AlbumTester.java) of user conversation code that RELIES on your coding the Album class properly.

- Classes you write that way are unit testable
Unit Testing

- Classes you write that way are **unit testable**
- That means they can be tested with unit tests.
- Java and other software development support systems have “frameworks” or infrastructure for defining and then running unit tests automatically
  - [http://www.junit.org/](http://www.junit.org/)
Revision Control, Version Histories

- It's modern practice for all serious software developers and large web site writers.
- Especially important when a possibly global team collaborates.
- Web links:
  - GIT, Mercurial (distributed)
  - CVS, Subversion (client-server)
- You can confidently DELETE instead of commenting out code you'll probably not use.
Project 4 and Bill class demos

- Main 201 goal: When given a sufficiently detailed description, or her/his own conception, of what a computer can do and a strategy for doing it, the student shall write a program that makes the computer do it by implementing the given strategy.

- Dependencies: What data must be input or computed FIRST, before other data it depends on can be computed or output.
  - Help you figure out a program that makes the computer compute in a correct ORDER.
Project 4 and Bill class demos

- Learning goals for data structures course readiness:
  - variables, arrays, control statements, methods, classes, basic problem solving
  - Solve program reading and writing problems in which some int variables are used for ARRAY INDEXING (and locating Pixels by coordinates). Examples: Finding index of largest and smallest elements, sorting the array, finding the candidate's NUMBER given the candidate's NAME, etc.

  Goal: You become ready to deal with pointer or reference and/or array index DATA whose purpose in to locate OTHER DATA.
Java arrays

- A Java array is a Java object, (like a Picture, Turtle, Pixel, Album, Bill, etc.)
- SO... it MUST be made (instantiated) by `new`
  - (code like `= { 2, 4, 6, 8 }` is just shorthand.)
- AND a reference variable must be used to store its address (otherwise, it's useless).
- The length of the array must be “known” (stored in the computer) FIRST, before the actual array can be made; and the array length CANNOT BE CHANGED.
How to make and use a Java array

• Declare a reference variable ready for the computer to write the address:
  – `int refToArray[];
    //NEVER PUT THE LENGTH IN!
• Having ALWAYS decided or computed the length, actually MAKE (instantiate) the array:
  – `refToArray = new int[366];`
• Having already decided or computed an index value:
  – Store data in an element: `refToArray[3] = 7;`
  – Retrieve data from an element:
    `System.out.println(refToArray[3]);`
Similarity to making & using objects

• Declare reference variables:
  - Bill myBill; Bill yourBill;

• ACTUALLY MAKE THE OBJECTS:
  - myBill = new Bill(); yourBill = new Bill();

• Use objects by calling methods:
  - myBill.addThing(60000,"Sports Car");
  - yourBill.addThing(50,"Junky used bike");
Java arrays

• Given the array and index value, you can get at the element FAST; NO looping, searching, etc.

• The length of the array must be “known” (stored in the computer) FIRST, before the actual array can be made; and the array length CANNOT BE CHANGED.

• Why is this important?
  – Other languages, add-ons (ArrayList), and data structure techniques remove this limitation.