CSI 201

*Introduction to Computer Science*

*Spring 2012*
Announcements
Reading

- You should have read chapters 1-4
- Read chapters 5 & 6

The midterm will cover everything from the class up through the end of Thursday. This includes:

- Chapters 1-5, 6-1 and 6-2 in the textbook and mentioned in Lectures or used in projects or labs.
- Labs, Assignments, & Exercises (tasks and material written on the assignments)
The Midterm will be this Thurs March 8th in class.

* Be prompt! Will begin at 4:15pm and MUST end at 5:35pm.

The midterm will be open book, open notes.

* No electronic devices are allowed
  * e.g. laptops, calculators, cell phones, iPods, headphones

* You must bring your Student ID to take the exam.

* You must remain in the exam room until halfway through the exam
  * Students who leave earlier than this will receive a zero on the exam

* You must arrive no later than halfway through the exam
  * Students arriving later than this will not be allowed to take the exam
Modifying Pictures Using Loops
The Generality of Programming

- Course is very focused on media programming (e.g. the digital images).
- But, it is important to remember that you can use the skills you learn here to write programs for almost any task.
- It’s limited mostly by your imagination.
  – And a bit by your skill.
The Generality of Programming

• A loop over Pixels is just an introduction to loops over ANY ARRAYS OF TICKETS with data.

• Calculation of color or x, y coordinates is just an introduction to computation of numbers for ALL KINSDS of purposes.

• It’s limited mostly by your imagination.
  – And a bit by your skill.
Faking a Sunset

• If you want to make an outdoor scene look like it happened during sunset
  – You might want to increase the red
    • But you can’t increase past 255
  – Another idea is to reduce the blue and green to emphasize the red.
    • Try to reduce the blue and green by 30%
    • IT WORKS!!
Algorithm to fake a sunset

Do this on EACH Pixel:

• Reduce the blue and green intensities by 30%

  1) **Change the blue intensity of the Pixel to 0.7 times its original value**

  2) **Change the green intensity of the Pixel to 0.7 times its original value**

This style of computing instructions is called **chromographic** PSEUDO-CODE.
This style of writing computing instructions is called PSEUDO-CODE.

- People who know how to compute can follow these instructions.
- People who know Java can translate them into programs.
- Good for early drafts of programs and writing tentative thoughts about what to code.
This style of writing computing instructions is called PSEUDO-CODE.

- PSEUDO: Doesn't conform to a programming language grammar so no compiler (like JDK) can process it.
- CODE: Exact enough for (educated) people to know what it tells them to compute.
- chromographic

THAT'S WORD IS A JOKE!
Algorithm to fake a sunset

Do this on EACH Pixel:

• Reduce the blue and green intensities by 30%

1) Set the blue intensity of the pixel to 0.7 times its original value

2) Set the green intensity of the pixel to 0.7 times its original value
where there is a Pixel

//Body does it on ONE PIXEL
int nVal;
nVal = p.getBlue() * 0.70;
p.setBlue(nVal);
nVal = p.getGreen() * 0.70;
p.setGreen(nVal);
Pixel[] pixA;
pixA =
    this.getPixels();
for( Pixel p : pixA )
{
    /*Changes ONE Pixel*/
}
Pixel[] pA;
pA = this.getPixels();
for(int i=0; i < pA.length; i = i + 1 )
{
    Pixel p = pA[i];
    int n = p.getBlue()*.7;
    //etc..
}
iClicker Question
int value = 1;
for( int i = 0 ;
    i < 5 ;
    i++)
{
    value = value*2;
}

System.out.println(value);
Using `System.out.println()` in a Loop

• To check what is happening in your program, add:
  – `System.out.println(expression);`

• You might add this to the body to check the value of a variable while the loop is executing.
  – And to verify that the increment happens after the last statement in the loop.
int value = 1;
for( int i = 0; i < 5; i++)
{
    value = value*2;
    System.out.println(“i holds “ + i + “, value holds “ + value);
}
System.out.println(value);
i holds 0, value holds 2
i holds 1, value holds 4
i holds 2, value holds 8
i holds 3, value holds 16
i holds 4, value holds 32
32
Algorithm to Fake a Sunset

Do this on EACH Pixel:

• Reduce the blue and green intensities by 30% of EACH Pixel in the column:

  1) **Change the blue intensity of the Pixel to 0.7 times the original value**

  2) **Change the green intensity of the Pixel to 0.7 times the original value**
for ( int x = 0; 
    x < this.getWidth(); 
    x++ ) 
{
    (Loop body)
    My job is to make the (ONE) whole column (the one that's the value of x steps to the right) look a bit redder.
iClicker Question
for ( int x = 0 ;
x < getWidth() ;
x++ )
{
    (Loop body)
}

How many times is this Loop body executed? (gets to control computer.)

(A) Picture's height
(B) Picture's width
(C) height - 1
(D) width - 1
(E) width * height
• ONE(1) body run is supposed to redden ONE(1) column.
  • The width is the number of columns.
  • The body should be run the number of times returned by this.getWidth( )
• The column (x) coordinate ranges from 0 to this.getWidth( ) – 1 inclusively.
for (int x = 0; x < this.getWidth(); x++)
{
    (Loop's BODY reddens ONE column) }

So this loop runs the BODY getWidth() times, with x ranging from 0 to getWidth() - 1. The whole loop reddens all getWidth() columns.
Now, let's make a body that will redden one column.

But which column??
The column whose x-coordinate is written on x

```java
for(int y = 0; y < getHeight(); y++)
{
    redden the Pixel at (x,y)
}
```
BODY to redden the column whose $x$-coord is written on $x$

```c
for(int y = 0; y < getHeight(); y++)
    redden the Pixel at ($x$, $y$)
```

The value of $x$ remains constant while the whole column is reddened.
Java for the pseudo-code:
“redden the one Pixel at (x,y)”

```java
{ //x from the MANY column loop
  //y from the ONE column loop
  //We process ONLY ONE PIXEL:
  Pixel p = this.getPixel(x,y);
  int temp = p.getBlue() * 0.7;
  p.setBlue(temp);
  temp = p.getGreen() * 0.7;
  p.setGreen(temp);
}
```
What is an Array?

- It is a sequence of tickets, each for the same type of data.
- Each ticket is called “an array element”
- The computer knows one entry from the others by its index number.
Pictures have large numbers of Pixels

• Each One Pixel IS a TICKET with 3 numbers.

• How can we refer to each Pixel?
  
  – As -
    pixel1, pixel2, pixel3, pixel4, pixel5, ...

• Do we really want to individually name each one?
  
  – A 640 x 480 image = 307,200 pixels.

• We DO want to refer to each Pixel by the Pixel's x and y (coordinate) LOCATION

• The 2-dimensional array is an excellent solution.
Arrays in Java

- Each array has its length FIXED when it was built.
- `myArray.length` is a field that has `myArray`'s length.
- Beware: length is not a method: avoid coding `length()` with no parens.
What Data does a Picture Object Have?

• A Picture **object** has an array of Pixel objects.
  – Color numbers copied from the JPEG or BMP file.

• It knows and can return the Picture width int by code
  
  ```java
  pictureObj.getWidth()
  ```

• It knows and can return the Picture height int by code
  
  ```java
  pictureObj.getHeight()
  ```

• It knows and can return the array of Pixels by
  
  ```java
  Pixel[] pixelArray = pictureObj.getPixels()
  ```
How to make the computer get at the array of Pixels in a G&E Picture.

• Given code like:

\[ \text{Picture } M = \text{new Picture("beach.jpg")}; \]

• To get at the Pixel array, you code:

\[ \text{Pixel[] pAr= M.getPixels();} \]

• So

\[ \text{pAr[0].setColor( java.awt.Color.black );} \]
\[ \text{M.repaint();} \]

makes the upper-leftmost Pixel black.
Both command same thing:
Blacken out the first Pixel.

```
pAr[0].
   setColor(java.awt.Color.black);
M.repaint();
```

```
M.getPixel(0,0).
   setColor(java.awt.Color.black);
M.repaint();
```
pAr[0].
setColor(java.awt.Color.black);
M.repaint();

(1) select Pixel object--DOT syntax--
M.getPixel(0,0).
setColor(java.awt.Color.black);
M.repaint();

(2) call [name( ) syntax]
setColor ON that Pixel
How to read Java programs smartly

When you SEE anything.....

methodName (......)

THINK: Method CALL ACTION!
Representing Pixel data in an array.

- We’re used to thinking of the images as a two-dimensional array/grid/matrix of pixels.
- Inside the computer memory, it is a one-dimensional array.

![Diagram of a 4x3 grid with labels for width and height]
Representing Pixel data in an array.

- The grid of pixels stored as a (one-dimensional) array.
Representing Pixel data in an array.

• The grid of pixels stored as a (one-dimensional) array.
• The top image row, elements [0], [1], [2], [3]
Representing Pixel data in an array.

- The grid of pixels stored as a (one-dimensional) array.
- The middle image row, elements [4], [5], [6], [7]
Representing Pixel data in an array.

- The grid of pixels stored as a (one-dimensional) array.
- The bottom row, elements [8], [9], [10] and [11]
So..to do something with every Pixel in a G&E Picture, code, for example:

```java
Picture p = new Picture
    (FileChooser.pickAFile());
Pixel[ ] ar = p.getPixels();
for(int i = 0;
    i < ar.length; i = i + 1)
{
    ar[i].setColor(Color.black);
}
```
OR..a different way, equally good:

```java
Picture p = new Picture
for(int x = 0; x < p.getWidth(); x++)
{
    for(int y = 0; y < p.getHeight(); y++)
    {
        p.getPixel(x,y).setColor(Color.black);
    }
}
```
OR.. a 3\textsuperscript{rd} way, equally good:

```java
Picture p = new Picture
for(int y = 0; y < p.getHeight(); y++)
{
    for(int x = 0; x < p.getWidth(); x++)
    {
        p.getPixel(x,y).setColor(Color.black);
    }
}
```
Nested Loop: Ways 2&3

for( x = 0; x < width; x++)
{
    for( y=0; y < height; y++)
    {
        /*process Pix at(x,y)*/
    }
}

for( y = 0; y < height; y++)
{
    for( x=0; x < width; x++)
    {
        /*process Pix at(x,y)*/
    }
}
iClicker Question
Nested Loop: Way 3

```c
for( y = 0; y<height; y++)
{
    for(x=0; x<width; x++)
    {
        /*process Pix at(x,y)*/
    }
}
```

The computer has 2 tickets, x and y. Which one does she erase and rewrite most rapidly and often?

(A) Ticket x  (B) Ticket y
Nested Loop

#3: for(..y..){
    for (..x..){process pix(x,y)}
}

is LEFT TO RIGHT MOST RAPIDLY
Nested Loop

#3: for(..y..){
    for (..x..){process pix(x,y)}
}

is LEFT TO RIGHT MOST RAPIDLY

Computer is in her FIRST run of the x (inner) for loop.
#3: \texttt{for(..y..)\{for (..x..)\{process pix(x,y)\}\}}

is LEFT TO RIGHT MOST RAPIDLY

Computer is in her FIRST run of the x (inner) for loop.
#3: for(..y..){
    for (..x..){process pix(x,y)}
}

is LEFT TO RIGHT MOST RAPIDLY

Computer is still in her FIRST run of the x (inner) for loop.
#3: for(...) {
    for (...) {process pix(x,y)}
}  
is LEFT TO RIGHT MOST RAPIDLY

Computer is still in her FIRST run of the x (inner) for loop, but is about to finish.
#3: \texttt{for(..y..)}{
    \texttt{for (..x..)}{\texttt{process} \texttt{pix(x,y)}}
} \\
\text{is LEFT TO RIGHT MOST RAPIDLY}

Computer is almost done with her 2nd run of the x (inner) for loop.
#3: for(..y..){
    for (..x..){process pix(x,y)}
}
is LEFT TO RIGHT MOST RAPIDLY

Computer is almost done with its 3nd run of the x (inner) for loop.
After running the x (inner) loop 9 times, he is almost finished with ONLY ONE run of the y (outer loop).
iClicker Question
iClicker Question

Which of the two nested loops correspond to the order of the arrows?

(A)
for (int y = 0; y < 4; y++) {
    for (int x = 0; x < 4; x++)
    {
        // Access array elt[x,y]
    }
}

(B)
for (int x = 0; x < 4; x++) {
    for (int y = 0; y < 4; y++)
    {
        // Access array elt[x,y]
    }
}
“DEBUGGING”

- Tracing through your code to determine what is wrong is known as “debugging”
- Figure out: Do the commands, as written, make the computer do what you want?
**CODE TRACING IN DETAIL**

- USE PAPER to keep track of values in memory!
- So before the loop we see:

```java
int index = 1; int count = 0;
while ( index < 4 )
    count = count + index;
    index = index + 1; }
System.out.println(count);
```
Changing to Grayscale

• Grayscale ranges from black to white
  – The red, green, and blue values are the same

• How can we change any color to gray?
  – What number can we use for all three values?
    • The intensity of the color
  – We can average the colors
    • \((\text{red} + \text{green} + \text{blue}) / 3\)
  – Example
    • \((15 + 25 + 230) / 3 = 90\)
Grayscale Result
Grayscale with Luminance Exercise

- Create a new method `grayscaleWithLuminance()`
- Use the new algorithm for calculating intensity.
  - `intensity = (int) (red * 0.299 + green * 0.587 + blue * 0.114);`
  - Notice that we’re not dividing by 3.
    - because.. \(0.299 + 0.587 + 0.114 = 1.0\)
Summary

• Pictures have pixels.
  – You can change the picture by changing the color of the pixels.
  – You will need to repaint the picture.

• Arrays let you store and retrieve values of the same type using an index.

• You will need to import classes that you wish to use that aren’t in java.lang.

• Loops allow you to execute a block of statements zero to many times.
Conditionally Modifying Pixels
If, Then, Else
Conditionally Modifying Pixels

This material corresponds to chapter six in the text.
Learning Goals

• Understand at a conceptual and practical level
  – How to use simple conditionals
  – How to use complex conditionals
  – How to remove red-eye
  – How to posterize a picture
  – How to do chroma-key or blue screen
Remove Red Eye

- Red eye is when the flash from the camera is reflected from the subject’s eyes.
- We want to change the red color in the eyes to another color.
  - But not change the red of her dress.
Red Eye Algorithm

• We can find the area around the eyes to limit where we change the colors
  – Using `pictureObj.explore()`
  – But we still just want to change the pixels that are “close to” red.
  – We can find the distance between the current color and our definition of red
    • And change the color of the current pixel only if the current color is within some distance to the desired color
Detailed Red Eye Algorithm

• Pass the x and y of the starting location, and the x and y value of the end location.

• Using a nested loop through x and y:
  – Get the pixel at this x and y
  – Get the distance between the pixel color and red
  – If the distance is less than some value (167) then change the color to some passed new color.
Detailed Red Eye Algorithm

• Pass the x and y of the starting location, and the x and y value of the end location.

• Using a nested loop through x and y:
  – Get the pixel at this x and y
  – Get the distance between the pixel color and red
  – If the distance is less than some value (167) then change the color to some passed new color.
Conditional Execution

- Sometimes we want a statement executed only if some expression is true
  - We can use the “if” statement in Java

```java
if (colorDistance < value)
    Statement or block to execute
next statement
```
Using if Exercise

- Open DrJava and try this in the interactions pane (or main method!)

```java
int x = 2;
if (x > 1)
    System.out.println("X is > 1");
System.out.println("X is " + x);
x = 0;
if (x > 1)
    System.out.println("X is > 1");
System.out.println("X is " + x);
```
Using if Exercise

• Open DrJava and try this in the interactions pane (or main method!)

```java
int x = 2;
if (x > 1)
    System.out.println("X is > 1");
    System.out.println("X is " + x);
x = 0;
if (x > 1)
    System.out.println("X is > 1");
    System.out.println("X is " + x);
```
A NOTE ABOUT SYNTAX..

• "if", "while", and "for" statements should **always** be indented, and require brackets if there is more than one line in the block.

```java
if (x > 1)
    System.out.println("I print conditionally");
    System.out.println("I will always print");
```

• So to print only if x is greater than one...

```java
if (x > 1) {
    System.out.println("I print conditionally");
    System.out.println("Me too!");
}
```
Color Distance

• The *distance* between two points is computed as

\[ \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \]

\((x_1, y_1)\)  \hspace{1cm} \((x_2, y_2)\)
Color Distance

• The distance between two colors can be computed
  – Square root of \(((\text{red1} - \text{red2})^2 + (\text{green1} - \text{green2})^2 + (\text{blue1} - \text{blue2})^2)\)
  – There is a method in the Pixel class to do this
    • \textit{double dist =}
    \begin{verbatim}
    pixelObj.colorDistance(color1);
    \end{verbatim}
public void removeRedEye(int startX, int startY, int endX, int endY, Color newColor) {

    // loop through the pixels in the rectangle
    // defined by values of
    // startX, startY, endX and endY
    for(int x = startX; x < endX; x++) {
        for(int y = startY; y < endY; y++) {

            // get the current pixel
            Pixel pixelObj = this.getPixel(x,y);

            // if the color is near red then change it
            if(pixelObj.colorDistance(Color.RED) < 167) {
                pixelObj.setColor(newColor);
            }
        }
    }
}
Testing removeRedEye

- `String file = FileChooser.getMediaPath("jenny-red.jpg");`
- `Picture p = new Picture(file);`
- `p.explore();`
- `Color colorObj = new Color(150, 150, 200);`
- `p.removeRedEye(123, 96, 135, 107, colorObj);`
- `p.explore();`
Edge Detection

• Find the areas of high contrast and turn pixels in this area black

  – Turn all other pixels white
Edge Detection Algorithm

• blacken areas of high contrast, whiten areas of low contrast.
– Loop from $y = 0$ to $y < \text{height} - 1$
  • Loop from $x = 0$ to $x < \text{width}$
    – Get the pixel at the $x$ and $y$ (top pixel)
    – Get the pixel at the $x$ and $(y + 1)$ (bottom pixel)
    – Get the average of the top pixel color values
    – Get the average of the bottom pixel color values
    – If the absolute value of the difference between the averages is over a passed limit
      » Turn the pixel black
      » Otherwise turn the pixel white
Use if and else for two possibilities

- Sometimes you want to do one thing if the expression is true
- and a different thing if it is false

```java
int x = 200;
if (x < 128)
    System.out.println("<128");
else
    System.out.println(">=128");
```
Edge Detection Exercise

• Write a method `edgeDetection` that takes an input limit
  – And turns all pixels black where the absolute value of the difference between that pixel and the pixel below is greater than the passed limit
  – And turns all pixels white where the absolute value of the difference between that pixel and the below pixel is less than or equal to the passed limit
public void edgeDetection()
{
    Pixel topPixel = null;
    Pixel bottomPixel = null;
    for (int x = 0; x < this.getWidth(); x++)
    {
        for (int y = 0; y < this.getHeight()-1; y++)
        {
            topPixel = this.getPixel(x,y);
            bottomPixel = this.getPixel(x,y+1);

            if (topPixel.colorDistance(bottomPixel.getColor()) > 25)
                topPixel.setColor(Color.BLACK);
            else
                topPixel.setColor(Color.WHITE);
        }
    }
}
Testing Edge Detection

- `String file = FileChooser.getMediaPath("butterfly1.jpg");`
- `Picture p = new Picture(file);`
- `p.explore();`
- `p.edgeDetection(10);`
- `p.explore();`
iClicker Question
What will be output by the following code?

```java
int x = 2; if (x > 0) System.out.println("X is positive."); System.out.println("X is \( + x + \) ");
```
What will be output by the following code?

```java
int x = 2;
if (x > 0) System.out.println("X is positive.");
System.out.println("X is " + x + ".");
```
What will be output by the following code?

```java
int x = 2;
if (x > 0) System.out.println("X is positive.");
System.out.println("X is " + x + ".");
```
What will be output by the following code?

```java
int x = 2;
if (2 > 0) System.out.println("X is positive.");
System.out.println("X is " + 2 + ".");
```
Modifying Pixels in a Matrix

Two Dimensional Arrays & Nested Loops
Modifying Pixels in a Matrix

This material corresponds to chapter five in the text.
Learning Goals

• Understand at a conceptual and practical level
  – What is a two-dimensional array?
  – How do you create two-dimensional arrays?
  – How do you access data in two-dimensional arrays?
  – How do you use nested loops?
For Loops

- Something else worth mentioning about for loops is that the change area can alter the loop variable in ways other than simply incrementing or decrementing.
For Loops

```java
public void turtleHop()
{
    for(int index = 0; index < 180; index += 10)
    {
        this.turn(45);
        this.forward(index);
    }
}
```
```java
public void drawSquare(int width){
    this.turnRight();
    this.forward(width);
    this.turnRight();
    this.forward(width);
    this.turnRight();
    this.forward(width);
    this.turnRight();
    this.forward(width);
}

public void turtleHop() {
    for(int index = 0; index < 180; index += 10) {
        this.turn(45);
        this.forward(index);
    }
}

public static void main(String[] args) {
    World earth = new World();
    Turtle t1 = new Turtle(earth);
}
```
For Loops - example (output)
For Loops

```java
public void dot50()
{
    Pixel[] pixelArray = this.getPixels();

    // loop through all the pixels
    for (int index = 0; index < pixelArray.length; index += 50)
    {
        // set the pixel color
        pixelArray[index].setColor(Color.black);
    }
}
```
iClicker Question
Given the following array and declaration,
which code fragment will print “9” to the console?

```
int x = 3;
```

- None of the above
What is a two-dimensional array?

- Each pixel has a value (vertical location)
- pictureObj.getPixel(x,y) returns the pixel at that location

![Diagram showing pixel values in a 2D array](image.png)
Example Two-Dimensional Arrays

- Maps
  - That city is in
Example Two-Dimensional Arrays

- Battleship game
  - Try I-5
  - Hit or mis
Example Two-Dimensional Array

- Chairs at a theater or game
  - Row C seat 20
Nested Loop

• How would you get all the pixels in a picture using their x and y values?
  – One possibility is - from left to right and top to bottom?
Nested Loop

- How would you get all the pixels in a picture using their x and y values?
  - One possibility is - from left to right and top to bottom?
• How would you get all the pixels in a picture using their x and y values?
  – One possibility is - from left to right and top to bottom?
Nested Loop

• How would you get all the pixels in a picture using their x and y values?
  – One possibility is - from left to right and top to bottom?
How would you get all the pixels in a picture using their x and y values?

- One possibility is - from left to right and top to bottom?
Nested Loop

• How would you get all the pixels in a picture using their x and y values?
  – One possibility is - from left to right and top to bottom?
Nested Loop

• How would you get all the pixels in a picture using their x and y values?
  – One possibility is - from **left to right** and **top to bottom**?
Nested Loop

• How would you get all the pixels in a picture using their x and y values?
  – One possibility is - from left to right and top to bottom?
Nested Loop

• How would you get all the pixels in a picture using their x and y values?
  – One possibility is - from left to right and top to bottom?
Nested Loop

- How would you get all the pixels in a picture using their x and y values?
  - From left to right and top to bottom?
  - $[0,0]$ (i.e. $x=0$ and $y=0$, $[1,0]$, $[2,0]$, ...)
  - $[0,1]$, $[1,1]$, $[2,1]$, ...
  - $[0,2]$, $[1,2]$, $[2,2]$, ...

- We need to have one loop inside another
  - The **outer loop** counts $y$ from 0 to $\text{height} - 1$
  - The **inner loop** counts $x$ from 0 to $\text{width} - 1$
Nested Loop

• To do this, we need to have two loops.
• Two nested loops: one loop inside another
  – The outer loop counts $y$ from 0 to $(\text{height} - 1)$
  – The inner loop counts $x$ from 0 to $(\text{width} - 1)$
Nested Loop

• The outer loop makes sure we do each row
  – \( y = 0, \) then 1, 2, 3, … (height - 1)

• The inner loops makes sure that for each row, we do each element
  – \( x = 0, \) then 1, 2, 3, … (width-1)
// loop through the columns
for(int x = 0; x < this.getWidth(); x++)
{
    // get the current pixel
    pixelObj = this.getPixel(x,y);
    // modify the color
    // put appropriate statements
    // set the new color
    pixelObj.setColor(colorObj);
} // Inner for loop

} // Outer for loop

Sets which row we’re working on.
For the current row, do the element in each column.
• We could also do it in the other order.
  – Do it column by column
  – Within each column, make sure we do all the rows.

• This is equivalent to switching the two loops.
  – Making the x loop the outer loop.
  – Making the y loop in inner loop.
Nested Loop - The Other Way

Sets which column we’re working on. For the current row, do the element in each row.

```java
// loop through the rows
for(int y = 0; y < this.getHeight(); y++) {
    // get the current pixel
    pixelObj = this.getPixel(x,y);

    // modify the color
    // put appropriate statements

    // set the new color
    pixelObj.setColor(colorObj);
} // Inner for loop

} // Outer for loop
```
Alternative Nested Loop

- How would you get all the pixels in a picture using their $x$ and $y$ values?
  - From top to bottom and left to right?
iClicker Question
Which of the following locations is not an element in the 2D array shown?

- [4,3]
- [2,1]
- [0,0]
- [3,3]
- All of the above are valid elements of the array
Using 2D Arrays for Pictures

• How can we use these 2D arrays to manipulate digital images?

• Let’s try an example where we lighten all of the pixels in an image...
Lighten the Color Algorithm

- Start $x$ at 0 and loop while $x < \text{the picture width}$ (add 1 to $x$ at the end of each loop)
  - Start $y$ at 0 and loop while $y < \text{the picture height}$ (add 1 to $y$ at the end of each loop)
    - Get the pixel at this location
    - Get the color at the pixel
    - Lighten (brighten) the color
    - Set the color for the pixel to the lighter color
public
void
lighten()
{
Pixel
pixel = null;
Color
color = null;

// loop through the columns (x direction)
for (int x = 0; x < this.getWidth(); x++)
{
// loop through the rows (y direction)
for (int y = 0; y < this.getHeight(); y++)
{
// get pixel at the x and y location
pixel = this.getPixel(x, y);

// get the current color
color = pixel.getColor();

// get a lighter color
color = color.brighter();

// set the pixel color to the lighter color
pixel.setColor(color);
}
// Inner for loop
}
// Outer for loop
} // method lighten()
Trying the Lighten Method

• In the interactions pane:

```java
String file = "c:/intro-prog-java/mediasources/caterpillar.jpg";

Picture p1 = new Picture(file);

p1.explore();
p1.lighten();
p1.explore();
```
Changing to Nested Loop Exercise

• Change the method `clearBlue()` to use a nested for loop to loop through all the pixels.

• Run the method again to check that it still works.

• Check that the blue values are all 0 using `pictureObj.explore()`.
iClicker Question
Which of the two sets of loops corresponds to the order shown by the arrows?

- for (int y = 0; y < 4; y++)
  
  for (int x = 0; x < 4; x++)
  
  {
    // Access [x,y]
  }

- for (int x = 0; x < 4; x++)
  
  for (int y = 0; y < 4; y++)
  
  {
    // Access [x,y]
  }

- Neither