CSE 8A Lecture 7

• Reading for next class: 5.2-5.3

• If you don’t see your RQ and P grades, please make sure your clicker is registered.

• PSA3 – interview: due Saturday 5pm.
  – Remember your (complete) comments including partner history
  – Style in the assignments (we’ll talk today about it, start counting more credit in PSA4)

• PSA 4: discussion sections to get started

• Exam 2: Coming up next Thursday

• 1 week to ask for grading “corrections” (check piazza post and syllabus for details).

CLICKERS OUT!
What picture most accurately describes what this code does?

```java
Pixel[] pixelArray = this.getPixels();
Pixel p;
Pixel q;
for(int index = 0; index < pixelArray.length-1; index++)
{
    p = pixelArray[index];
    q = pixelArray[index+1];
    p.setRed(q.getRed());
    p.setBlue(q.getBlue());
    p.setGreen(q.getGreen());
}
```

What if we change to ...
```java
p = pixelArray[index+1];
q = pixelArray[index];
```

A.

B.

C.

D. None of these
Why does this code have an error?

A. It tries to access pixelArray[-1]
B. It tries to access pixelArray[0]
C. It tries to access pixelArray[pixelArray.length]
D. It tries to access pixelArray[pixelArray.length+1]
E. None of the above

Pixel[] pixelArray = this.getPixels();
int value = 0;
Pixel p = null;
for(int index = 0; index < pixelArray.length; index++)
{
    p = pixelArray[index];
    q = pixelArray[index+1];
    p.setRed(q.getRed());
    p.setBlue(q.getRed());
    p.setGreen(q.getGreen());
}
Fill in the for(......) to loop over pixels bottom right to top left

• Like this:

```java
Pixel[] pixArr = this.getPixels();

for (                     )
{
    //Some code doing set on pixArr[i]
}
```
• True or False: The following code could be written using a for-loop.

```java
Pixel[] pixelArray = this.getPixels();
int index = 0;
while ( index < pixelArray.length )
{
    Pixel pix = pixelArray[index];
    pix.setGreen(255);
    index = index + 1;
}
```

A. True
B. False
while

Pixel[] pixelArray = this.getPixels();
int index = 0;
while ( index < pixelArray.length )
{
    Pixel pix = pixelArray[index];
    pix.setGreen(255);
    index = index + 1;
}

do...while

Pixel[] pixelArray = this.getPixels();
int index = 0;
do {
    Pixel pix = pixelArray[index];
    pix.setGreen(255);
    index = index + 1;
} while (index < pixelArray.length);

for

Pixel[] pixelArray = this.getPixels();
for ( int index = 0; index < pixelArray.length; index++ )
{
    Pixel pix = pixelArray[index];
    pix.setGreen(255);
}

for each

Pixel[] pixelArray = this.getPixels();
for ( Pixel pix: pixelArray )
{
    pix.setGreen(255);
}

Which do you prefer?
Why?
while vs. for vs. for each

- So when to use each? Sometimes it’s a matter of style, sometimes it’s ease of functionality

Choose the best loop to use in each of these situations
A. For
B. For each
C. While

1. You want to loop through a picture until you find a pixel that is all black. Then you want your loop to stop.

2. You want to loop through all the pixels in a picture and set each pixel red value equal to its green value

3. You want loop through the pixels in the first half of the picture and make them all black.
while vs. for vs. for each: Summary

- **for each**
  - Use when you know you need to access and modify each pixel directly

- **for**
  - Use when you need to loop through a known number of pixels and need access to their index value

- **while**
  - Use when you are not sure how many pixels to loop through

Often, though, you can just choose the loop you like best
Nested Loops: How do they work? What order are pixels changed?

- This is called a “nested loop” because it is a loop inside another loop. WHAT?!! Can you do that?!

```java
Pixel p;
for (int foo = 0; foo < getWidth(); foo++)
{
    for (int bar = 0; bar < getHeight(); bar++)
    {
        System.out.println( foo + " " + bar );
    }
}
```
Nested Loops: How do they work? What order are pixels changed?

• Nested loops unroll the same way!

```java
Pixel p;
for (int foo = 0; foo < getWidth(); foo++)
{
    System.out.println( foo );
}
```

If the width of the Picture is 2, this will unroll to:

```java
System.out.println( 0 );
System.out.println( 1 );
```
Nested Loops: How do they work? What order are pixels changed?

• Consider a simple loop

```java
Pixel p;
for (int foo = 0; foo < getWidth(); foo++)
{
    for (int bar = 0; bar < getHeight(); bar++) {
        System.out.println( foo + " " + bar );
    }
}
```

If the width of the Picture is 2, this will unroll to:

```java
for (int bar = 0; bar < getHeight(); bar++) {
    System.out.println( 0 + " " + bar );
}
for (int bar = 0; bar < getHeight(); bar++) {
    System.out.println( 1 + " " + bar );
}
```

Then these loops unroll too…
Nested Loops: How do they work?
What order are pixels changed?

- A method in Picture.java... what does it print if width is 2 and height is 3?

```java
Pixel p;
for (int foo = 0; foo < getWidth(); foo++)
{
    for (int bar = 0; bar < getHeight(); bar++)
    {
        System.out.println( foo + " " + bar );
    }
}
```

1) Solo: (30 sec)
2) Discuss: (2min)
3) Group: (30 sec)

Will be converted automatically to a string for printing

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>10</td>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>20</td>
<td>02</td>
<td>22</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>01</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>11</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>21</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>
Nested Loops: Tracing code

- A method in Picture.java... what does it print if width is 2 and height is 3?

```java
Pixel p;
for (int foo = 0; foo < getWidth(); foo++)
{
    for (int bar = 0; bar < getHeight(); bar++)
    {
        System.out.println(foo + " " + bar);
    }
}
```

foo  

bar
What do these Picture methods do? What are their return types?

- `getPixel(int x, int y)`
- `getHeight()`
- `getWidth()`
Nested Loops: How do they work? In what order are pixels changed?

- A method in Picture.java...

```java
Pixel p;
for (int bar = 0; bar < getHeight(); bar++)
{
    for (int foo = 0; foo < getWidth(); foo++)
    {
        p = getPixel(foo, bar);
        p.setColor(Color.BLACK);
    }
}
```

1) Solo: (30 sec)
2) Discuss: (2min)
3) Group: (30 sec)
Why does this have an error?

- In a method in Picture.java... assume height=50, width=100

    Pixel p;
    for (int bar = 0; bar < getWidth();  bar++)
    {
        for (int foo = 0; foo < getHeight(); foo++)
        {
            p = getPixel(foo, bar);
            p.setColor(Color.BLACK);
        }
    }

A. It doesn’t, this loops across rows, top to bottom
B. It doesn’t, this loops down columns, left to right
C. It tries to index a pixel off the end of a row (x value too big)
D. It tries to index a pixel off the end of a column (y value too big)
Why did that have an error?

- The method `getPixel` in `Picture.java` with two parameters interprets the first one as an ‘x’ coordinate, and the second one as a ‘y’ coordinate of the Pixel to get.

- When you call that method to get a Pixel from a Picture, it doesn’t matter what the names of the variables are that you pass in!

- `getPixel(foo,bar)` or `getPixel(bar,foo)` or `getPixel(x,y)` or `getPixel(y,x)`…

- The first parameter is always interpreted as the ‘x’ coordinate, and the second one as the ‘y’ coordinate, of the pixel you want.
How to fix that error

• Since `bar` takes values 0 to `getWidth()`, it is acting like an ‘x’ coordinate

• Since `foo` takes values 0 to `getHeight()`, it is acting like a ‘y’ coordinate

• So pass `bar` as first argument, and `foo` as second argument, to `getPixel`:

```
p = getPixel(bar, foo);
```

• (Better yet: write `x` instead of `bar` and `y` instead of `foo`; the computer doesn’t care, but it makes the code clearer to a human reader!)
What’s with foo and bar anyway?

The use of foo in hacker and eventually in programming context may have begun in MIT's Tech Model Railroad Club (TMRC)

*Foobar* may have derived from the military acronym **FUBAR** and gained popularity because it is pronounced the same.

–Wikipedia foobar page

Despite their popularity, **foo** and **bar** are NOT good choices for variable names

As the name of a bar, it’s pretty good, though
Some comments on style

Pixel p; for (int bar = 0; bar < getWidth(); bar++)
{
    for (int foo = 0; foo < getHeight(); foo++)
        {p = getPixel(foo, bar);
         p.setColor(Color.BLACK);}
}

What’s wrong with this code?
Some comments on style

Meaningful variable names (generally more than 1 character)

Pixel pix;
for (int xpos = 0; xpos < getWidth(); xpos++)
{
    for (int ypos = 0; ypos < getHeight(); ypos++)
    {
        pix = getPixel(xpos, ypos);
        pix.setColor(Color.BLACK);
    }
}

Proper indentation (Dr. Java will help with this)

One statement per line

Lines not longer than 80 characters
But sometimes it’s fun to break the rules

www.ioccc.org/
Fill in the code below to make the right (approximately) half of the picture pure blue.

```java
Pixel p;
for (int x = ___________; x < ___________; x = ______) {
    for (int y = _____; y < ____________; y = _________ ) {
        p = getPixel( x, y )
        p.setBlue( ________ );
        p.setRed( _______ );
        p.setGreen( ________ );
    }
}
```
Mirroring Around Vertical Axis

Mirror left to right

Vertical axis
Mirroring Around Vertical Axis

Mirror right to left

Vertical axis
Mirroring Around Vertical Axis: Left to Right

- What are the parameter values we use to index `leftPixel` and `rightPixel` for the first three iterations of the inner loop? (assume picture has a height = 50 and width = 100)

```java
int mirrorPt = getWidth()/2;
Pixel leftP, rightP;
for (int y = 0; y < getHeight(); y++)
{
    for (int x = 0; x < mirrorPt; x++)
    {
        leftP = getPixel(x,y);
        rightP = getPixel(getWidth()-1-x,y);
        rightP.setColor(leftP.getColor());
    }
}
```

- A) 0.99 99.0
- B) 0.98 98.0
- C) 0.97 97.0
- D) 0.96 96.0
- E) None of the above
How do you figure these kinds of questions out?

- **Answer:** Draw a diagram
  - imagine “beginning” and “answer”
  - Draw arrows to show how to get from beginning to answer
  - Then fill in numbers in order, write loops to create those numbers
Copying pixels

The key to (almost) all of the image manipulation problems in lab is to copy the color value across pixels in an image. The key is figuring out which pixels to copy and where to copy them to.

Here is the generic template that you will use for almost all of these problems:

```java
Pixel sourcePixel, targetPixel;
for ( int sourceX = [start of source region (horiz)];
    sourceX < [end of source region]; [move sourceX]) {
    for ( int sourceY = [start of source region (vert)];
        sourceY < [end of source region]; [move sourceY]) {
        sourcePixel = getPixel( sourceX, sourceY )
        targetPixel = getPixel( _________, __________ )
        targetPixel.setColor( sourcePixel.getColor() );
    }
}
```
The key to (almost) all of the image manipulation problems in lab is to copy the color value across pixels in an image. The key is figuring out which pixels to copy and where to copy them to.

Here is the generic template that you will use for almost all of these problems:

```java
Pixel sourcePixel, targetPixel;
for ( int sourceX = [start of source region (horiz)]; sourceX < [end of source region]; [move sourceX] ) {
    for ( int sourceY = [start of source region (vert)]; sourceY < [end of source region]; [move sourceY] ) {
        sourcePixel = getPixel( sourceX, sourceY )
        targetPixel = getPixel( __________, __________ )
        targetPixel.setColor( sourcePixel.getColor() );
    }
}
```

WARNING! This is difficult! It takes lots of practice and careful reasoning, so don't be alarmed if you don't get it at first. You will, so stick with it.
Mirroring Even Width versus Odd Width

```java
int mirrorPt = getWidth() / 2;
...
for (int x = 0; x < mirrorPt; x++)
```
Mirroring Odd-width Pictures

• What happens when this code attempts to mirror a Picture around the vertical axis when the Picture’s width is odd (e.g. 101)?

```java
int mirrorPt = getWidth()/2;
Pixel leftP, rightP;
for (int y = 0; y < getHeight();  y++)
{
    for (int x = 0; x < mirrorPt; x++)
    {
        leftP = getPixel(x,y);
        rightP = getPixel(getWidth()-1-x,y);
        rightP.setColor(leftP.getColor());
    }
}
```

A. It will work fine

B. It will run, but it won’t mirror correctly

C. I won’t run, there will be an index out of bounds exception

D. It won’t even compile if getWidth() is odd
Mirror versus “flip” (PSA?) (around vertical axis)
What are the first \((x,y)\) coords for `topP` and `bottomP` to mirror around horizontal axis?

*topP*      *bottomP*

A. \((0,0)\)      \((0,3)\)
   \((0,1)\)      \((0,2)\)
   \((1,0)\)      \((1,3)\)

B. \((0,0)\)      \((0,3)\)
   \((1,0)\)      \((1,3)\)
   \((2,0)\)      \((2,3)\)

C. either A or B will work

D. none of the above
Challenge: Complete the code that mirrors in the order specified by answer B

```java
int height = getHeight();
int width = getWidth();
int mid = height/2;
Pixel topP, botP;
for (                                    ){
    for(                                    ) {
        topP = getPixel(                        );
        botP = getPixel(                        );
        botP.setColor(topP.getColor());
    }
}
```
TODO

• Keep working on PSA4 and show your cool images on Piazza!

• Reading and video for the next class.