CSE 8A Lecture 8

• Check all your grades (class, psa, lab …) in grade-source. Contact grader ASAP if you find a mistake.

• PSA4: deadline tonight. Interview Friday noon.
  – Careful! No more “submission” mistakes.

• Start early PSA5.
  – Tuesday no interviews (usually).
  – Tutors can’t give you solutions. Don’t ask for that.

• Tomorrow: LAB 4

• Reading for next class: None. Prepare for In-term 2

CLICKERS OUT!
Exam topics

Up to Thursday last week (so up to “style”).

• All topics from Exam 1 (though these will be built-on, rather than the sole focus)
• Loops:
  – all types (for, for-each, while) including nested loops
  – writing code involving loops
  – tracing code involving loops
  – identifying errors in code involving loops
  – ... all in the context of both the Turtle and Pictures, as well as simple loops that just involve print statements
• Methods:
  – Parameters
  – Return values
  – Writing methods
  – Calling methods, including how to refer to the calling object inside the method
• Pictures:
  – Pixels and Pixel manipulation
  – Accessing Pixels in the image, using both an array of Pixels as well as getting the Pixel "directly" by its x, y coordinate
  – Picture representation (i.e. how the x, y coordinates relate to the picture, as well as how the 2D array is unrolled into 1D)
  – Color representation/modification
• Random walks, Random number generation
• Style
1. How do I blend two picture objects with 50% of one picture and 50% of another picture?

A. By multiplying their colors together and by 0.5

B. By adding them together

C. By multiplying their colors by 0.5 and then adding them
2. When copying an entire picture (source) to another picture (target) do they need to be the same size?

A. Yes, if they are not an error will occur

B. No, if the source is smaller than the target then they do not need to be the same size

C. No, if the source is bigger than the target then they do not need to be the same size
2. To scale a picture smaller by 2 (to make a copy of the picture that is half the size of the source) what should go in the blanks?

```java
public void copySmaller()
{
    Picture pictObj = new Picture(FileChooser.pickAFile());
    Pixel sourcePixel = null;
    Pixel targetPixel = null;

    //loop through the columns
    for(int sourceX=0, targetX=0; sourceX < pictObj.getWidth(); _1_, targetX++)
    {
        //loop through the rows
        for(int sourceY=0; targetY=0; sourceY < pictObj.getHeight(); _2_, targetY++)
        {
            sourcePixel = pictObj.getPixel(sourceX, sourceY);
            targetPixel = this.getPixel(targetX, targetY);
            targetPixel.setColor(sourcePixel.getColor());
        }
    }
}
```

A. 1. sourceX++ 2. sourceY+=2  
B. 1. sourceX+=2 2. sourceY+=2  
C. 1. sourceX+=2 2. sourceY++  
D. 1. sourceX++ 2. sourceY++
Some comments on style

Meaningful variable names (generally more than 1 character)

```java
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
Writing nested for loops

Do we have to cast \(\text{getwidth()}/2\) to an \text{int}? A. Yes \(\text{B. No}\)

Fill in the code below to make the right (approximately) half of the picture \textit{pure blue}

Assume this code is in the \texttt{Picture} class

```java
Pixel p;
for (int x = \underline{\text{getwidth()}/2}; x < \underline{\text{getwidth()}}; x = x + 1) {
    for (int y = \underline{0}; y < \underline{\text{getHeight()}}; y = y + 1) {
        p = \underline{\text{getPixel}(x, y)};
        p.setBlue(\underline{255});
        p.setRed(\underline{0});
        p.setGreen(\underline{0});
    }
}
```
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 (coordinates) 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());
    }
}
```

<table>
<thead>
<tr>
<th>Left Pixel</th>
<th>Right Pixel</th>
</tr>
</thead>
<tbody>
<tr>
<td>A) 0, 99</td>
<td>99, 0</td>
</tr>
<tr>
<td>0, 98</td>
<td>98, 0</td>
</tr>
<tr>
<td>0, 97</td>
<td>97, 0</td>
</tr>
<tr>
<td>B) 0, 0</td>
<td>99, 0</td>
</tr>
<tr>
<td>1, 0</td>
<td>98, 0</td>
</tr>
<tr>
<td>2, 0</td>
<td>97, 0</td>
</tr>
<tr>
<td>C) 0, 49</td>
<td>49, 0</td>
</tr>
<tr>
<td>0, 48</td>
<td>48, 0</td>
</tr>
<tr>
<td>0, 47</td>
<td>47, 0</td>
</tr>
<tr>
<td>D) 0, 0</td>
<td>49, 0</td>
</tr>
<tr>
<td>1, 0</td>
<td>48, 0</td>
</tr>
<tr>
<td>2, 0</td>
<td>47, 0</td>
</tr>
</tbody>
</table>

E) None of the previous options
How do you figure these kinds of questions out?

• Answer: Draw a diagram
  – imagine “beginning” and “answer” (goal)
  – 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 (or something similar to this):

```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 tricky! 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

```
int mirrorPt = getWidth() / 2;
...
for (int x = 0; x < mirrorPt; x++)
```

```
int mirrorPt = getWidth() / 2;
...  
for (int x = 0; x < mirrorPt; x++)
```
• What happens when this code attempts to mirror a Picture around the vertical axis when the Picture’s width is odd (e.g. 101, or 3)?

```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

It won’t do exactly the same thing as even, because it won’t touch the middle column, but it will produce a correct mirror.
Mirror versus “flip” (PSA!) (around vertical axis)
What are the first \((x,y)\) coords for \(\text{topP}\) and \(\text{bottomP}\) to mirror around horizontal axis?

- A. \((0,0)\) \((0,3)\)
- B. \((0,0)\) \((1,0)\) \((1,3)\)
- 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 (int y=0; y<mid; y++) {
    for (int x=0; x<getWidth(); x++) {
        topP = getPixel(x, y);
        botP = getPixel(x, getHeight()-1-y);
        botP.setColor(topP.getColor());
    }
}
```
Order of copying pixels

• When mirroring, we need to copy certain pixels to certain other pixels

• It doesn’t matter what order we copy in, as long as when we are done, pixels have been copied correctly

• Two most common orders:
  • **Row major order**: copy all the pixels in one row, then go on to the next row
  • **Column major order**: copy all the pixels in one column, then go on to the next column
Mirroring around horizontal axis

column-major order

```java
int height = getHeight();
int width = getWidth();
int mid = height/2;
Pixel topP, botP;
for(int x=0; x<width; x++) {
    for(int y=0; y<mid; y++) {
        topP = getPixel(x,y);
        botP = getPixel(x, height-1-y);
        // copy one to the other...
    }
}
```

topP bottP
A. (0,0) (0,3)
   (0,1) (0,2)
   (1,0) (1,3)
   ...
   ...

row-major order

```java
int height = getHeight();
int width = getWidth();
int mid = height/2;
Pixel topP, botP;
for(int y=0; y<mid; y++) {
    for(int x=0; x<width; x++) {
        topP = getPixel(x,y);
        botP = getPixel(x, height-1-y);
        // copy one to the other...
    }
}
```

topP bottP
B. (0,0) (0,3)
   (1,0) (1,3)
   (2,0) (2,3)
   ...
   ...

TODO

• Study for In-term Exam 2
• Finish PSA4
• Lab tomorrow!