MEDIA COMPUTATION

MANIPULATING A PICTURE

- To manipulate a picture we need to manipulate the pixels that make it up.
  - Change the red, green, or blue values at the pixel
WHAT DATA DOES A PICTURE OBJECT HAVE?

• A picture object has an array of pixel objects.
  – That it read from the JPEG file.
• It knows the picture width
  `pictureObj.getWidth()`
• It knows the picture height
  `pictureObj.getHeight()`
• It knows how to return an array of pixels
  `Pixel[] pixelArray = pictureObj.getPixels()`

REPRESENTING PIXEL DATA IN AN ARRAY.

• We’re used to thinking of the images as a two-dimensional array/grid/matrix of pixels.
• We can also represent it as a one-dimensional array.

![Diagram](image)
REPRESENTING PIXEL DATA IN AN ARRAY.

- The grid of pixels represented as a (one-dimensional) array.

```
String fileName = FileChooser.pickAFile();
Picture pictureObj = new Picture(fileName);
pictureObj.show();
int width = pictureObj.getWidth();
System.out.println("The picture width is " + width);
int height = pictureObj.getHeight();
System.out.println("The picture height is " + height);
Pixel[] pixelArray = pictureObj.getPixels();
System.out.println(pixelArray.length + " pixels");
```

PICTURE EXERCISE

- Create a picture in DrJava
  - get the picture’s width, height, and pixels

```
String fileName = FileChooser.pickAFile();
Picture pictureObj = new Picture(fileName);
pictureObj.show();
int width = pictureObj.getWidth();
System.out.println("The picture width is " + width);
int height = pictureObj.getHeight();
System.out.println("The picture height is " + height);
Pixel[] pixelArray = pictureObj.getPixels();
System.out.println(pixelArray.length + " pixels");
```

PIXEL OBJECTS

- Each pixel has a red, green, and blue value, which we can get or set:

  ```java
  - valueR = pixelObj.getRed()
  - valueG = pixelObj.getGreen()
  - valueB = pixelObj.getBlue()
  - pixelObj.setRed(valueR)
  - pixelObj.setGreen(valueG)
  - pixelObj.setBlue(valueB)
  ```
PIXEL OBJECTS

- Each pixel knows the location where it belongs in the picture object:
  - `valueX = pixelObj.getX()`
  - `valueY = pixelObj.getY()`

PIXEL EXERCISE

- In DrJava
  - Pick a file and create a picture object.
  - Get the array of pixels from the picture object.
  - Get the 1st pixel from the array of pixels
    - `Pixel pixelObj = pixelArray[0]; // 0 is first one`
  - Get the red, green, and blue value for this pixel.
  - Get the x and y location of this pixel.
  - Get the color of this pixel.
    - Get the red, green, and blue values of the color.

COLOR OBJECTS

- There is a class defined in Java that represents color:
  - The Color class in the package java.awt
  - To use the class you must either
    - `import java.awt.Color;`
    - or
    - Use the full name - `java.awt.Color;`
CREATING COLORS

• You can create a color object by giving the red, green, and blue values:
  ```java
  Color colorObj = new Color(255, 10, 125);
  ```

PREDEFINED COLORS

• The Color class has defined class constants for many colors
  ```java
  Color.red, Color.green, Color.blue, Color.black, Color.white, Color.yellow, Color.gray, Color.orange, Color.pink, Color.cyan, Color.magenta
  ```
  – Or you can use all uppercase names
  ```java
  Color.RED, Color.BLUE, Color.BLACK, ...
  ```

GETTING AND SETTING PIXEL COLORS

• To get a pixel's color as a color object
  ```java
  import java.awt.Color;
  Color color1 = pixelObj.getColor();
  int red = color1.getRed();
  int green = color1.getGreen();
  int blue = color1.getBlue();
  ```

• To set a pixel's color using a new color object
  ```java
  import java.awt.Color;
  Color color2 = new Color(red, green, blue);
  pixelObj.setColor(color2);
  ```

USING CLASSES IN PACKAGES

• All classes in the Java language are grouped into packages.
  – You can already use any class in java.lang:
    • System
    • Math
    • Object
      because they are always available in the Java language.
  
• That is, they are so frequently used, the decision was made to have them automatically available.
USING CLASSES IN PACKAGES

• For classes in other packages you need to import them:
  – import java.awt.Color;
  or
  – import java.awt.*; //import all classes in java.awt package to use the short name: Color

• Or use the fully qualified name:
  – packageName.ClassName
  – java.awt.Color

UNDEFINED CLASS ERROR

• If you forget to import a class
  – but use the short name for the class
  – it won’t compile
    • Undefined class error

• Undefined class errors mean:
  – You need to import the class
  – Or you misspelled the class
  – Or used the wrong case

CHANGING PIXEL COLORS

• There are two ways to change the color of a pixel in a picture.
  – Set the red, green, and blue values individually
    • pixelObj.setRed(value);
    • pixelObj.setGreen(value);
    • pixelObj.setBlue(value);
  – Or set the color
    • pixelObj.setColor(colorObj);

• But, you won’t see any change in the picture until you “redraw” it:
  – Until you ask it to repaint:
    pictureObj.repaint();
  – Or show it again:
    pictureObj.show();
CHANGING A COLOR

- The Color class has methods for making a color object...
  - Lighter
    - `colorObj.brighter();`
  - Darker
    - `colorObj.darker();`

```
import java.awt.Color;
Color testColor = new Color(168,131,105);
System.out.println(testColor);
java.awt.Color[r=168,g=131,b=105]
testColor = testColor.darker();
System.out.println(testColor);
java.awt.Color[r=117,g=91,b=73]
testColor = testColor.brighter();
System.out.println(testColor);
java.awt.Color[r=167,g=130,b=104]
```

ROUNDING ERRORS

- Notice that when you make the color darker and then lighter the resulting color is slightly different than the original.
  - The change is calculated in floating point.
  - The result is stored in integer form.
  - The decimal part is lost.

- Rounding errors also occur because of the limited storage for floating point numbers.
  - We can't store all the digits necessary to represent some numbers.

PICTURES ARE 2-DIMENSIONAL ARRAYS

- They have columns and rows (x and y).
- You can get a pixel at a particular x and y location

```
Pixel pixelObj = pictureObj.getPixel(x,y);
```

- The columns and rows
  - start with index 0
  - end with `num-1`, where num is the size of the row or column.

CHANGING A PICTURE EXERCISE

```
import java.awt.Color;
String fileName = "C:/intro-prog-java/mediasources/caterpillar.jpg";
Picture pictureObj = new Picture(fileName);
pictureObj.show();
pictureObj.repaint();
pictureObj.getPixel(10,100).setColor(Color.black);
pictureObj.getPixel(11,100).setColor(Color.black);
pictureObj.getPixel(12,100).setColor(Color.black);
pictureObj.getPixel(13,100).setColor(Color.black);
pictureObj.getPixel(14,100).setColor(Color.black);
pictureObj.getPixel(15,100).setColor(Color.black);
pictureObj.getPixel(16,100).setColor(Color.black);
pictureObj.getPixel(17,100).setColor(Color.black);
pictureObj.getPixel(18,100).setColor(Color.black);
pictureObj.getPixel(19,100).setColor(Color.black);
pictureObj.repaint();
```
WE NEED A LOOP (ITERATION)

• The loop will help us execute a repeated series of statements without having to write each one out.

WE NEED A LOOP (ITERATION)

• A loop is a way to execute a series of statements.
• Two key parts of the loop:
  – Something changing each time the statements are executed.
    • For instance, a different index for the pixel to change.
  – And some way to tell when we are done with the repetition.
    • Some test to see if the loop should stop.

LOOP EXERCISE

• Ask a person to clap 12 times
  – How does she know when to stop?
  – What changes each time she claps?
• If you are following a recipe that asks you to stir the ingredients 50 times how would you do this?
• What if you were trying to break a sit-up record
  – How would you know if you did break it?

LOOPS OFTEN NEED COUNTERS

• If you want to do something \( x \) times you often need a counter
  • The counter is also called an index.
  – It starts at 0
  – And you add 1 to it each time you do the task you are repeating.
  – When the counter reaches the number you are trying to do you stop the loop.
  • What is the value of the counter the last time the statements of the loop are executed?