2-D arrays and images

Learning Objectives

Understand and use a 2D array.

Understand the coordinate system for pixels and how to retrieve one.

Pictures and 2-D Arrays

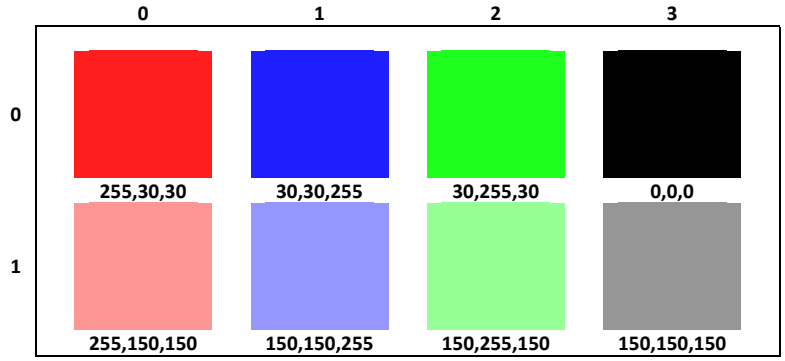
As we saw previously, an array is like a line of boxes, each storing one particular item. This line only extends in one dimension (length), so it is sometimes referred to as a 1-D array. A 2-D array on the other hand, has both a width and a height. A chessboard is a good example of a 2-D array: it is an 8 by 8 grid of squares, and each square can hold a chess piece.

two-dimensional array (definition)

A way of representing a two-dimensional structure such as an image using an array of arrays. Two dimensional arrays use two indices (column and row) instead of one to access their data elements.

Accessing individual boxes in a 2-D array is similar to that of a 1-D array, except that now, you need to provide 2 indexes. One index is for the row, and the other index is for the column. They are in the form of *arrayname[column][row]*. Thus, on the chessboard, to access one the top left corner, we could use chessboard[0][0], and to access the diagonally opposite corner, we could use chessboard[7][7].

learn by doing



The image above represents a 2D array named colourMatrix.

1. How would you access the colour indicated at (30,255,30)? _____
2. How would you access the colour indicated as (150,150,255)? _____
3. What is the red value of the colour at colourMatrix[3][0]? _____

Our solution

1. The correct answer is colourMatrix[2][0]. You can index into a 2-D array by using column and row coordinates. In this case, you can do something like colourMatrix[columnindex][rowindex].
2. The correct answer is colourMatrix[1][1].
3. You can index into a 2-D array by using column and row coordinates. The first number (3) is the column, and the second (0) is the row. The correct answer is the black square. The red value of the black square is the R in RGB (the first number, i.e. 0).

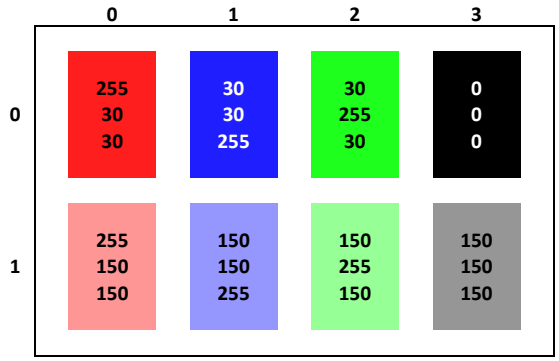
Pictures are like 2-D arrays. They are filled with pixels in two directions (like the squares on a chessboard). Individual pixels can be addressed using their x and y coordinates. To do this with the picture object, we can use the following code:

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

This gets the pixel information at point (x,y), and stores it an object of type Pixel called pixelObj.

Example: A 2-D Array of Pixels

This 2D array represents a picture that is very tiny. It is 4 pixels wide and 2 pixels high. getPixel(0,0) would return the red coloured pixel, whose RGB values are (255, 30, 30). getPixel(3,1) would return the grey pixel whose RGB values are (150,150,150)



This 2D array represents a picture that is very tiny. It is 4 pixels wide and 2 pixels high. getPixel(0,0) would return the red coloured pixel, whose RGB values are (255, 30, 30). getPixel(3,1) would return the grey pixel whose RGB values are (150,150,150)

Did I get this?

If you start to look, you will see two dimensional arrays in other places besides images. For example, when you get theatre tickets, they may provide a row number and a seat number. Can you think of any other real world examples of two dimensional arrays?

Our Solution

Another good example is a paper street map. You can look up a street on a street map, and it will have something like "D5" written next to it. This tells you to look for the area of the map labeled D (4th column) and 5 (5th row). Many board games also use a two dimensional grid (such as checkers and chess).

We can define a two-dimensional array in the same way we define a one-dimensional array, but with an extra set of brackets. To define a 5-by-4 int array, for example, we would use int[][] arr = new int[5][4]; To get the number at column 2, row 3, we would use arr[1][2].

Changing a picture

Now that we know how to get to pixels in the middle of an image, let's try to make a visible change in an image. Load any image you would like, (or use the image in the netbeans package folder - MediaFourEx), and let's draw a horizontal line. Remember that the first coordinate of a picture controls the x (horizontal) position.

```
String filename = FileChooser.pickAFile();
Picture mypic = new Picture(filename);
```

```
mypic.getPixel(10, 100).setColor(Color.YELLOW);
mypic.getPixel(11, 100).setColor(Color.YELLOW);
mypic.getPixel(12, 100).setColor(Color.YELLOW);
mypic.getPixel(13, 100).setColor(Color.YELLOW);
mypic.getPixel(14, 100).setColor(Color.YELLOW);
mypic.getPixel(15, 100).setColor(Color.YELLOW);
mypic.getPixel(16, 100).setColor(Color.YELLOW);
mypic.getPixel(17, 100).setColor(Color.YELLOW);
mypic.getPixel(18, 100).setColor(Color.YELLOW);
mypic.getPixel(19, 100).setColor(Color.YELLOW);
mypic.getPixel(20, 100).setColor(Color.YELLOW);
mypic.repaint();
```

This draws a horizontal YELLOW line, 10 pixels long, and 100 pixels from the top of the image. Can you see it in the image below?

Summary

A 2D array stores items in multiple rows. Those items are indexed using arrayName[column][row]

A picture can be thought of as a 2D array of pixels. To retrieve a pixel at a specific location, we use getPixel(*x*,*y*), which specifies the horizontal (x) coordinate and the vertical (y) coordinate of the pixel to be retrieved.