2D Array's and Graphics
by Mr. F
2D Arrays and Grids

Many games use 2D arrays to represent grids
Basics

Use a 2D array that contains values appropriate for the game

```java
public class Checkers {
    private int[][] board = new int[8][8];
}
```

0 - EMPTY
1 - RED
2 - BLACK
3 - RED KING
4 - BLACK KING
Drawing

* paint method draws based on what's in the 2D array

* Need to know the width/height of each box on the grid...

\[
\text{GETWIDTH()} = 500 \quad \text{BOARD.LENGTH} = 8 \text{ (ROWS)}
\]
\[
\text{BOARD[0].LENGTH} = 8 \text{ (COLS)}
\]
\[
\text{BOXWIDTH} = \frac{500}{\text{BOARD[0].LENGTH}}
\]
\[
\text{BOXHEIGHT} = \frac{600}{\text{BOARD.LENGTH}}
\]
The paint method analyzes each box

```java
public void paint(Graphics g)
{
    for(int row = 0; row < board.length; row++)
    {
        for(int col = 0; col < board[0].length; col++)
        {
            if(board[row][col] == 1) //draw a red piece
            {
                g.setColor(Color.RED);
                g.fillOval(..., ..., ..., ...);
            }
        }
    }
}
```
Row/Col Drawing

* Given the row and column, you can geometrically determine the parameters for drawing

- PANEL WIDTH = 500
- PANEL HEIGHT = 600
- ROW = 1
- COL = 3
- ULX = BOXWIDTH*COL
- ULY = BOXHEIGHT*ROW
- THE WIDTH AND HEIGHT OF YOUR BOXES ARE JUST THE BOXWIDTH AND BOXHEIGHT
public void paint(Graphics g)
{
    //calculate box width and box height
    for(int row = 0; row < board.length; row++) {
        for(int col = 0; col < board[0].length; col++) {
            //calculate ulx and uly
            if(board[row][col] == 1) //draw a red piece
            {
                g.setColor(Color.RED);
                g.fillOval(ulx, uly, boxWidth, boxHeight);
            }
        }
    }
}
Mouse Location

- `mousePressed()` is automatically called when the mouse is clicked.
- It tells you where in the panel the mouse clicked, but not the row and column in your array!

**PANEL WIDTH = 500**

**PANEL HEIGHT = 600**

How do we know that this is in row 1, column 3 mathematically?
Mouse Mathematics

- If you find how many box widths are in the x-coordinate (rounded down), that's the column!

- Similar for the row

PANEL WIDTH = 500

BOXWIDTH = 500/8 = 62
BOXHEIGHT = 600/8 = 75

COL = 190 / 62 = 3
ROW = 80 / 75 = 1

Fortunately, integer division rounds down automatically!
mousePressed vs paint

- You do NOT paint anything in mousePressed
- You do NOT do any game logic in paint

```java
public void paint(Graphics g) {
    //Let's see... what fun calculations can I do in here...
}
```

```java
public void mousePressed(MouseEvent e) {
    g.drawRect(0, 0, 100, 100);
}
```
The repaint command should ONLY be used at the end of an action event.

It's a signal that the paint method should be called.

```java
gpublic void mousePressed(MouseEvent e)
{
  //code...
  repaint();
}

gpublic void paint(Graphics g)
{
  //code..
  repaint();
}
```