Graphics Intro

By: Megan Avery
Updated February 2019

Before we begin

- Make sure you have Java & Java JDK downloaded on your computer
  - can run **`java -version`** in terminal to check
- Make sure you have IntelliJ downloaded on your computer

- *Suggested:* Watch previous Java tutorials

Objects Intro: Vocab Alert!

object: An entity that contains data and behavior.

- data: variables inside the object
- behavior: methods called on object

Graphical Objects

DrawingPanel: A window on the screen.

Graphics: A "pen" to draw shapes and lines on a window.

Color: Colors in which to draw shapes.

DrawingPanel Object

"Canvas" objects that represents windows/drawing surfaces

Note: Will need to be downloaded from the internet and put in the same folder as the program you are using it with

Example:

```
DrawingPanel name = new DrawingPanel(width, height);
```

Graphics Object

"Pen" or "paint brush" objects to draw lines and shapes

Creation:

```
Graphics g = panel.getGraphics();
```

Usage:

```
g.fillRect(10, 30, 60, 35);
g.fillOval(80, 40, 50, 70);
```

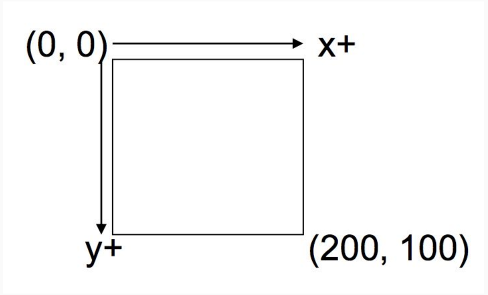
Note: Will need to import for graphics

Coordinate System

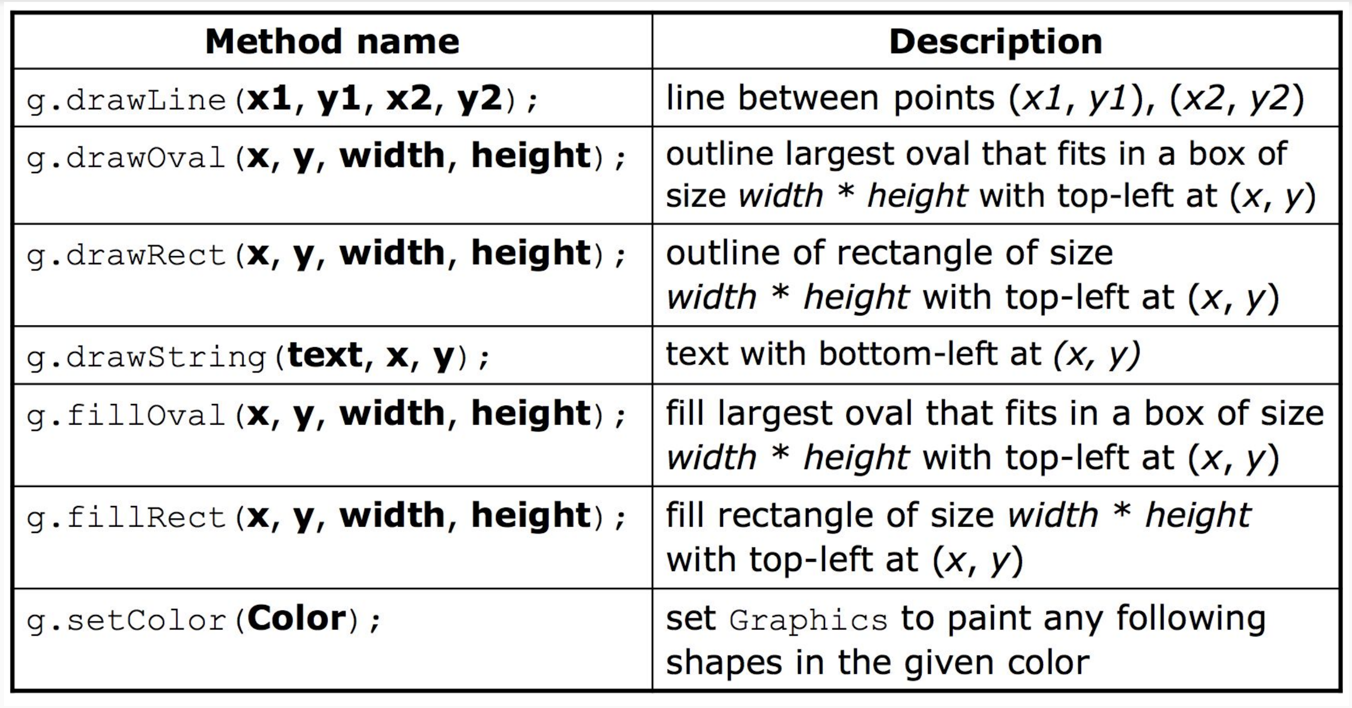
Each (x, y) position is a pixel ("picture element"). Position (0, 0) is at the window's top-left corner.

- x increases rightward and the y increases downward.

The rectangle from (0, 0) to (200, 100) looks like this:



Graphics Methods



Method name	Description
g.drawLine(**x1, y1, x2, y2**);	line between points (*x1*, *y1*), (*x2*, *y2*)
g.drawOval(**x, y, width, height**);	outline largest oval that fits in a box of size *width* * *height* with top-left at (*x*, *y*)
g.drawRect(**x, y, width, height**);	outline of rectangle of size *width* * *height* with top-left at (*x*, *y*)
g.drawString(**text, x, y**);	text with bottom-left at *(x, y)*
g.fillOval(**x, y, width, height**);	fill largest oval that fits in a box of size *width* * *height* with top-left at (*x*, *y*)
g.fillRect(**x, y, width, height**);	fill rectangle of size *width* * *height* with top-left at (*x*, *y*)
g.setColor(**Color**);	set `Graphics` to paint any following shapes in the given color

Color Object

Specified as predefined Color class constants:
Color.CONSTANT_NAME

Or create one using Red-Green-Blue (RGB) values of 0-255.

```
Color name = new Color(red, green, blue);
```

Example:

```
Color brown = new Color(192, 128, 64);
Color burntOrange = new Color(191, 87, 0);
```

Constant color options:

BLACK, BLUE, CYAN, DARK_GRAY, GRAY, GREEN, LIGHT_GRAY, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW

Using Colors

Pass a Color to Graphics object's setColor method

- Subsequent shapes will be drawn in the new color.

Example:

```
g.setColor(Color.BLACK);
g.fillRect(10, 30, 100, 50);
g.drawLine(20, 0, 10, 30);
g.setColor(Color.RED);
g.fillOval(60, 40, 40, 70);
```

Pass a color to DrawingPanel's setBackground method

- The overall window background color will change.

Example:

```
Color brown = new Color(192, 128, 64);
panel.setBackground(brown);
```

Shapes on top of shapes

When ≥ 2 shapes occupy the same pixels, the last drawn "wins."

Play With Graphics!

Practice for Later

Just keep playing! Maybe draw a simple house? A cat? The sky is the limit (ooo maybe a cloud?)

The End