Graphics

Java, Windows, Rectangles, Circles and Mouse Clicks

A first look

Import the graphics library

```
import graphics.*;

public class DrawSquare
{
   public static void main(String [] args)
   {
      Window w = new Window(400, 200);

      final int SIZE = 60;
      Rectangle square = new Rectangle(0, 0, SIZE, SIZE);

      w.draw(square);
   }
}
```

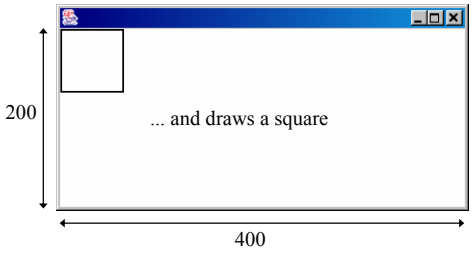
Create a Window

Create a Square

Draw the Square

DrawSquare.java

This creates a window



Pixels

- Pixels (picture elements) are the basic unit on a computer screen. Computer images are formed from pixels.
- A modern computer screen has millions of pixels, each of which can be coloured in thousands of different shades.

Objects to Draw

- There are a number of inbuilt graphics objects:
  - Point
  - Line
  - Rectangle
  - Circle
  - Ellipse
  - Arc

Objects

- Basic
  - Point
- Shape
  - Line
  - Rectangle
  - Circle
  - Ellipse
  - Arc

Constructing Objects: Examples

- A point needs an x and y coordinate:
  ```
  new Point(5, 10)
  ```
- A line needs a start and end point:
  ```
  new Line(10, 0, 10, 60)
  ```
- A Rectangle gets the x and y coordinates followed by width and height
  ```
  new Rectangle(200, 10, 50, 3)
  ```
- A Circle gets x and y (the centre) and a radius:
  ```
  new Circle(100, 30, 15)
  ```

How it appears

Drawing Objects

- Each is drawn by calling the draw() method of Window

```
Window w = new Window(400, 200);
Circle c = new Circle(100, 30, 15);
w.draw(c);
```

alternately

```
w.draw(new Circle(100, 30, 15));
```

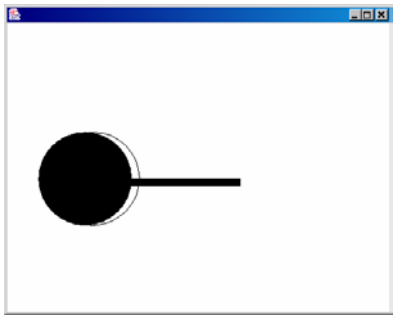
Other Methods

- The window has other methods
  - fill: fill the area enclosed by the shape
  - erase: erase an object (inverse of draw)
  - clear: clear an object (inverse of fill)
- These methods can be applied to all the shapes (e.g. Line, Rectangle, Circle, Ellipse and Arc)

Shape Methods

- Each Shape can do the following
  - move: change the current location
  - expand: increase the size of the object
  - shrink: decrease the size of the object

```
import graphics.*;

public class OtherMethods
{
   public static void main(String [] args)
   {
      Window w = new Window(500, 400);

      Rectangle rect = new Rectangle(100, 200, 200, 10);
      Circle circ = new Circle(100, 200, 60);
      w.draw(circ);
      w.erase(circ);

      w.fill(rect);
      w.fill(circ);

      // Move along 10 pixels in the x direction
      circ.move(10, 0);

      // and draw the moved circle
      w.draw(circ);
   }
}
```

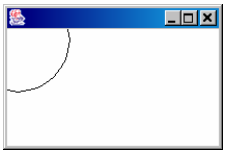


Note: Off site drawing

The following code places a circle on a window:

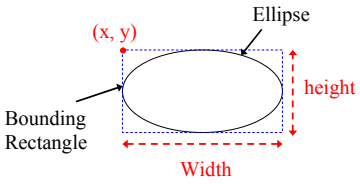
```
Window win = new Window(250, 150);
Circle c = new Circle(10, 10, 50);
win.draw(c);
```

The circle doesn't fit on the window; note that it is not an error, only the visible part of the circle is displayed.



Ellipse

To create an Ellipse, you need to enter the coordinates of the bounding rectangle

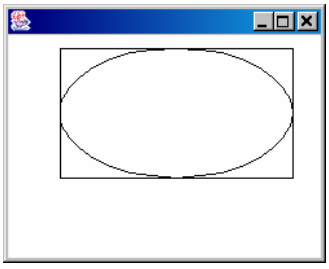



```
new Ellipse(x, y, width, height)
```

```
import graphics.*;

public class DrawEllipse
{
   public static void main(String [] args)
   {
      Window win = new Window(250, 200);

      int x = 40;
      int y = 10;
      int width = 180;
      int height = 100;
      Ellipse e = new Ellipse(x, y, width, height);
      Rectangle r = new Rectangle(x, y, width, height);

      win.draw(e);
      win.draw(r);
   }
}
```

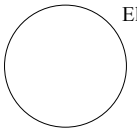
DrawEllipse.java

Creates the following shape

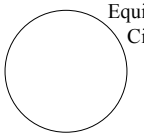


Ellipse Vs Circle

An Ellipse can also represent a Circle if the width equals the height



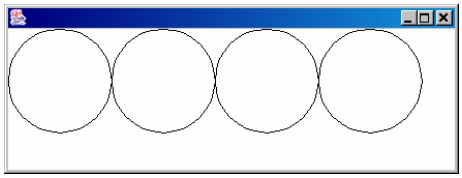

```
new Ellipse(0, 0, 10, 10)
```




```
new Circle(5, 5, 5)
```

Loops and Objects

- We want to create the following drawing:



Solution 1

- Just create and draw four circles

```
Circle c1 = new Circle(50, 50, 50);
win.draw(c1);

Circle c2 = new Circle(150, 50, 50);
win.draw(c2);

Circle c3 = new Circle(250, 50, 50);
win.draw(c3);

Circle c4 = new Circle(350, 50, 50);
win.draw(c4);
```

Four circles, we need to increase the x-cordinate each time.

Solution 1b

- We could use a single reference

```
Circle c = new Circle(50, 50, 50);
win.draw(c);

c = new Circle(150, 50, 50);
win.draw(c);

c = new Circle(250, 50, 50);
win.draw(c);

c = new Circle(350, 50, 50);
win.draw(c);
```

This is a minor modification

Solution 2

- Use a for loop

```
for(int x = 50; x <= 350; x += 100)
{
   Circle c = new Circle(x, 50, 50);
   win.draw(c);
}
```

This is much neater

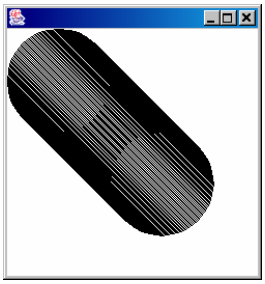
Solution 2b

- It is neater just to move the circle and draw it again. Note that moving the circle doesn't change the previously drawn circles.

```
Circle c = new Circle(50, 50, 50);
for(int i = 0; i < 4; i++)
{
   win.draw(c);
   c.move(100, 0);
}
```

Exercise

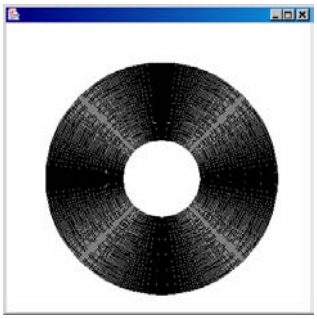
- See if you can create the following image (formed by drawing many circles). Each circle is just one pixel apart.



Exercise 2

This picture is formed by drawing many circles about one centre but with gradually increasing radius.

Hint: you can use the expand method.



Colours

- Colours are formed by mixing red, green and blue.
- Create a colour by adding appropriate amounts of each:

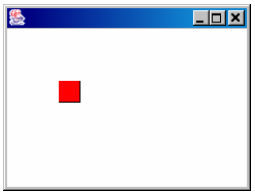
```
Color fire = new Color(r, g, b);
```

- Each colour should be in the range 0-255, so bright red would have r = 255 and b and g would be 0.

Red Square

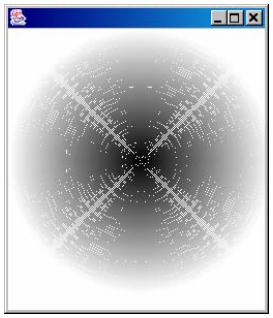
```
Color red = new Color(255, 0, 0);
win.setForeground(red);
win.fill(new Rectangle(50, 50, 20, 20));
```

You also have to call the setForeground method of the Window.

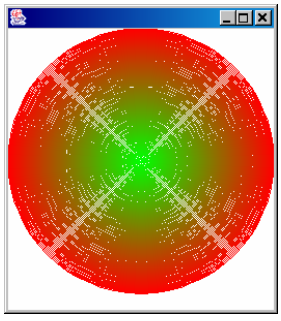


```
// Create a circle with zero radius
Circle roundy = new Circle(128, 128, 0);

// This loop varies the colour.
// When i is zero, the colour is black.
// As i increases the colour gets brighter
for(int i = 0; i <= 255; i++)
{
   // Create a new colour
   Color shade = new Color(i, i, i);
   // Make the window use this colour
   win.setForeground(shade);

   // Expand the circle by one unit.
   roundy.expand(1, 1);

   // Finally draw it.
   win.draw(roundy);
}
```

Ghostly Disc



Exercise



Summary

- Create a window
- Create a shape (e.g. Circle, Rectangle)
- Draw it (or fill, erase, clear it)
- Loops can help create interesting images
- Colours are formed by combining red, green and blue.