INTRODUCTION TO
COMPUTER SCIENCE

Rutgers University

6. Input / Output- continued



http://introcs.cs.rutgers.edu



DOWNLOAD and ADD to your program folder

- StdDraw.java

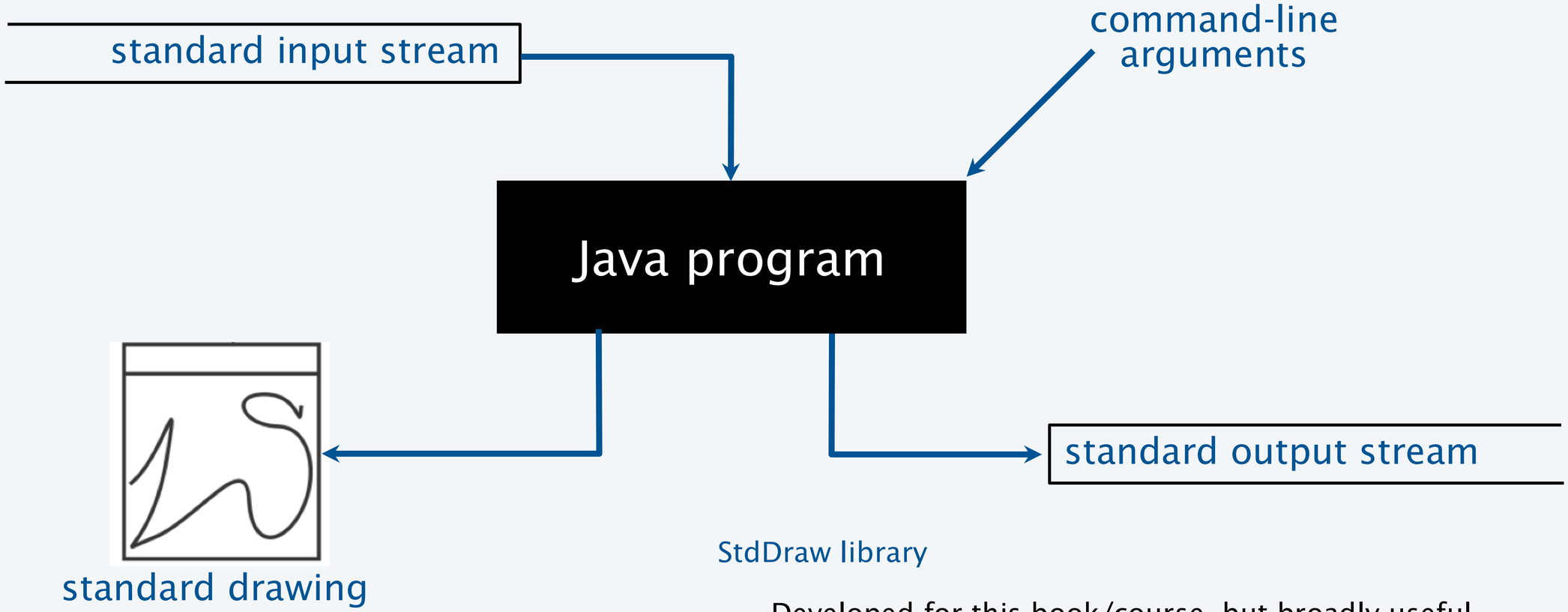


4. Input and Output

- Standard input and output
- Standard drawing
- Fractal drawings
- Animation

Further improvements to our I/O abstraction

Add the ability to create a *drawing*.



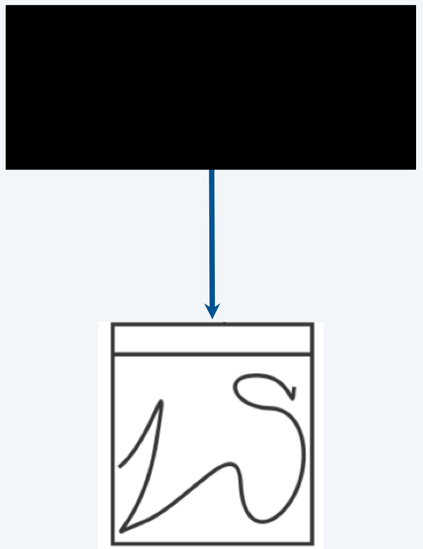
StdDraw library

- Developed for this book/course, but broadly useful.
- Available for download at booksite.
- Included in introcs software.

StdDraw library

public class StdDraw	
void line(double x0, double y0, double x1, double y1)	
void point(double x, double y)	
void text(double x, double y, String s)	
void circle(double x, double y, double r)	
void square(double x, double y, double r)	
void polygon(double x, double y, double r)	
void picture(double x, double y, String filename)	*place* .gif, .jpg *or* .png *file*
void setPenRadius(double r)	
void setPenColor(Color c)	
void setXscale(double x0, double x1)	*reset x range to* [x_0, x_1)
void setYscale(double y0, double y1)	*reset y range to* [y_0, y_1)
void show(int dt)	*show all; pause* dt *millisecs*

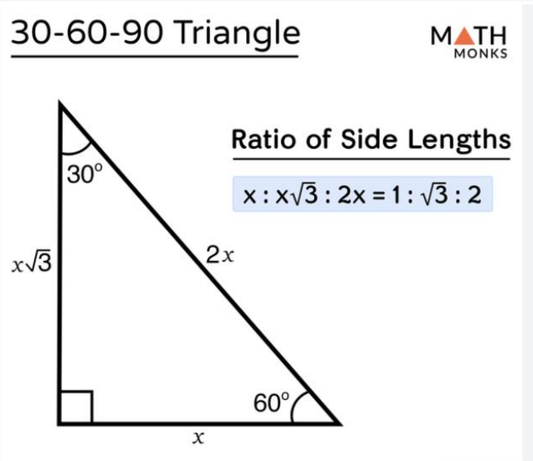
also filledCircle(), filledSquare(), and filledPolygon()



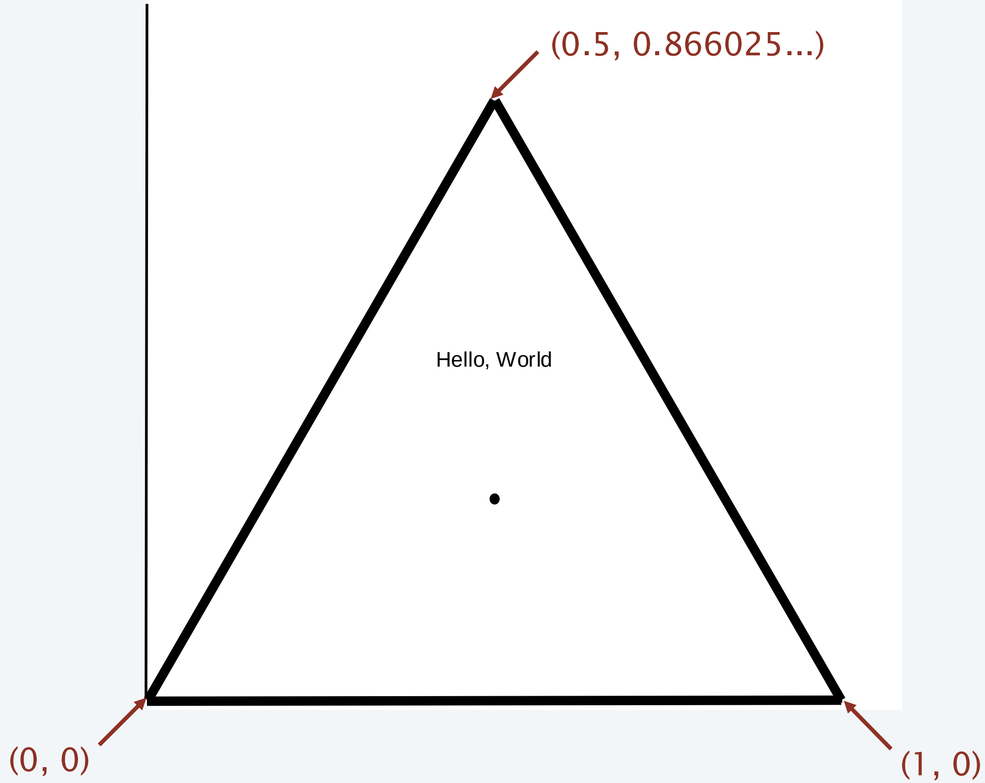
standard drawing

"Hello, World" for StdDraw

```
public class Triangle
{
   public static void main(String[] args)
   {
     double c = Math.sqrt(3.0) / 2.0;
     StdDraw.setPenRadius(0.01);
     StdDraw.line(0.0, 0.0, 1.0, 0.0);
     StdDraw.line(1.0, 0.0, 0.5, c);
     StdDraw.line(0.5, c, 0.0, 0.0);
     StdDraw.point(0.5, c/3.0);
     StdDraw.text(0.5, 0.5, "Hello, World");
   }
}
```



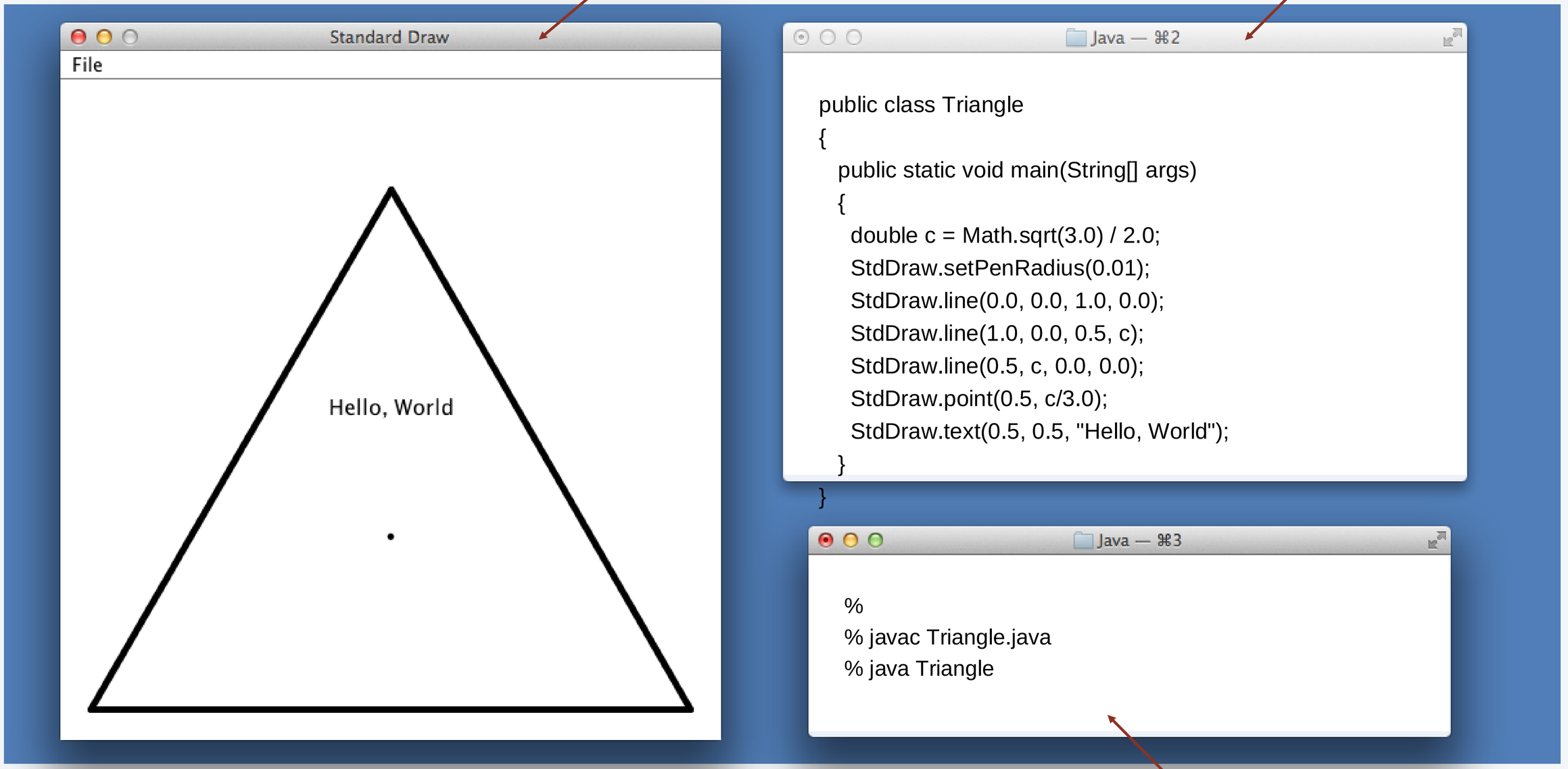
Trace of drawing



"Hello, World" for StdDraw

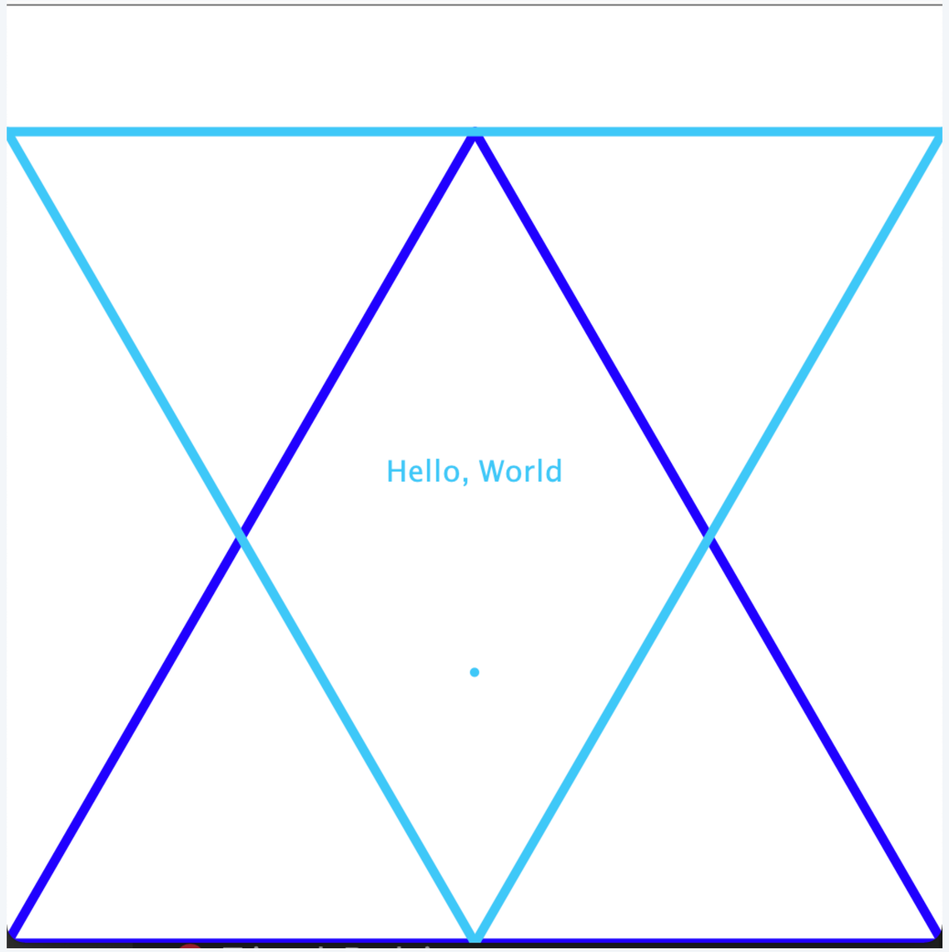
window for standard drawing

virtual terminal for editor



virtual terminal for OS commands

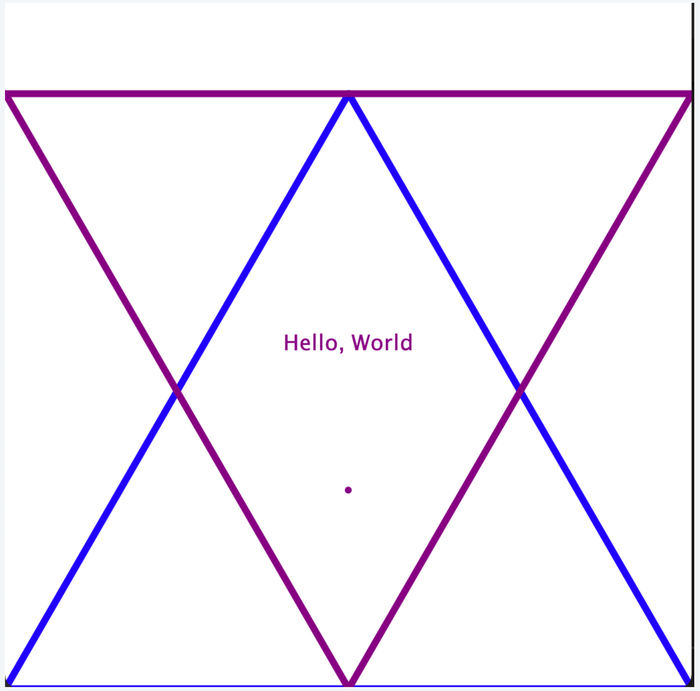
Can you FLIP it?



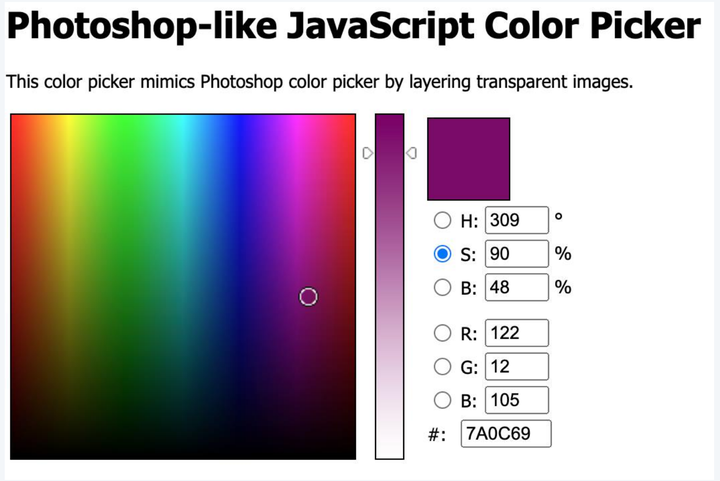
Line 7 of Triangle.java:
StdDraw.setPenColor.

Auto complete will list colors.

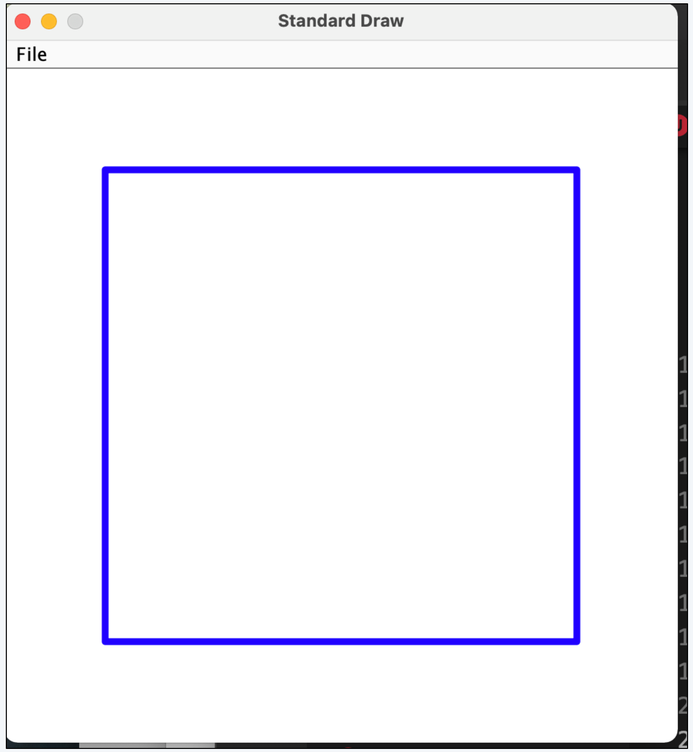
Can you FLIP it?

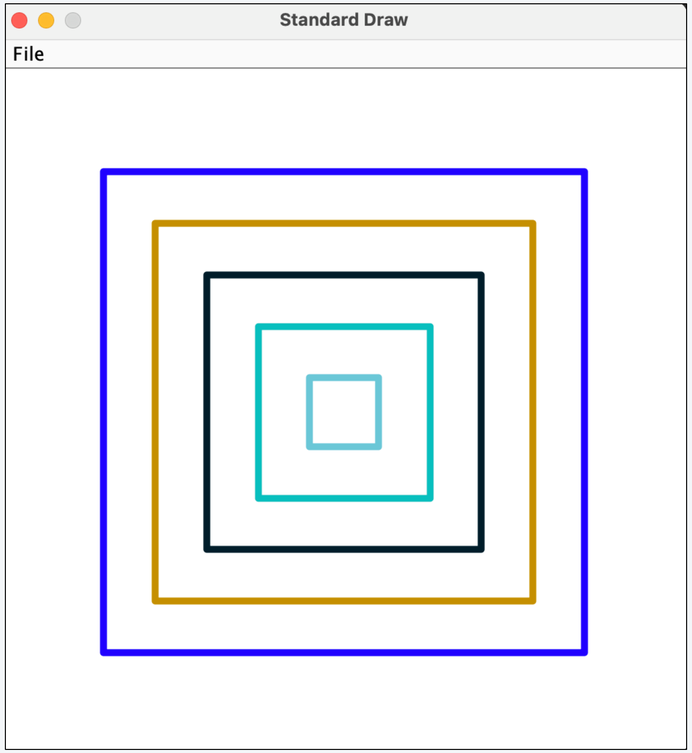


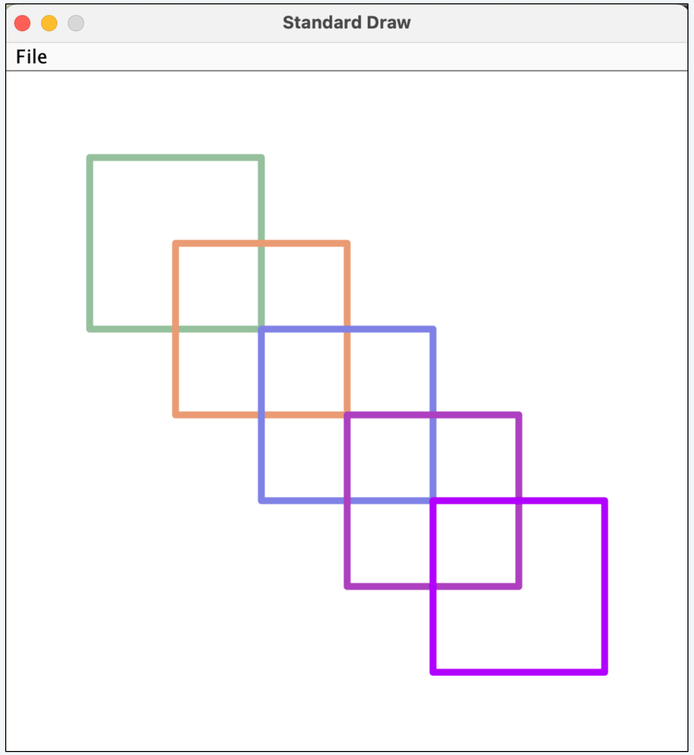
Line 943 of StdDraw and https://j.hn/lab/colorpicker/ :
StdDraw.setPenColor(122,12,125);



Can you code and display the SQUARES below?

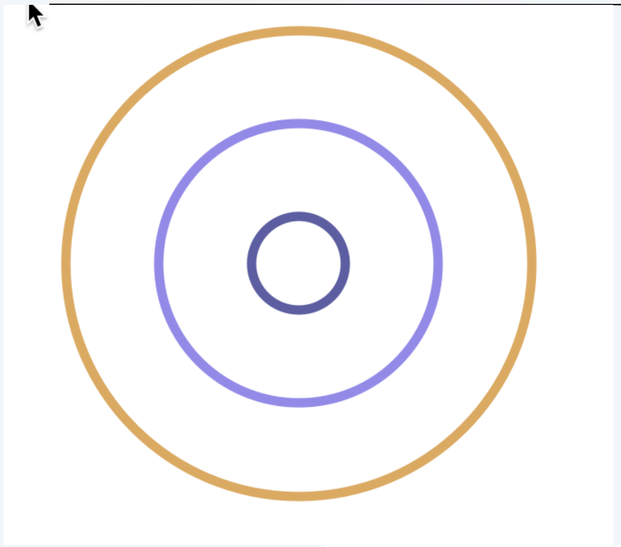


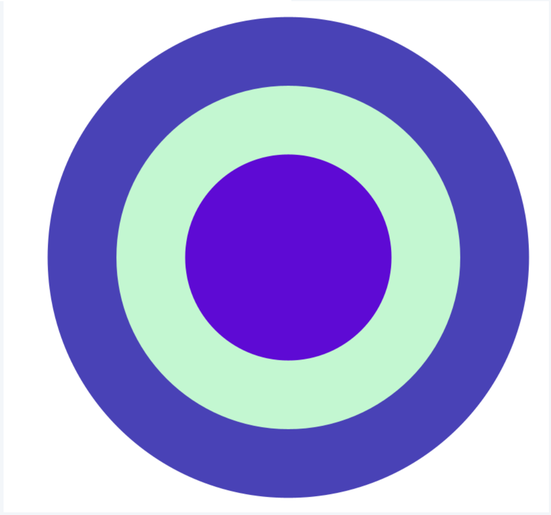





```
StdDraw.setScale(0,2); //canvas
double halfLength = 0.7;   //half the length of a side
StdDraw.square(1,1,halfLength); // x-coordinate of center, y-coordinate of center, halfLength
```

Circles





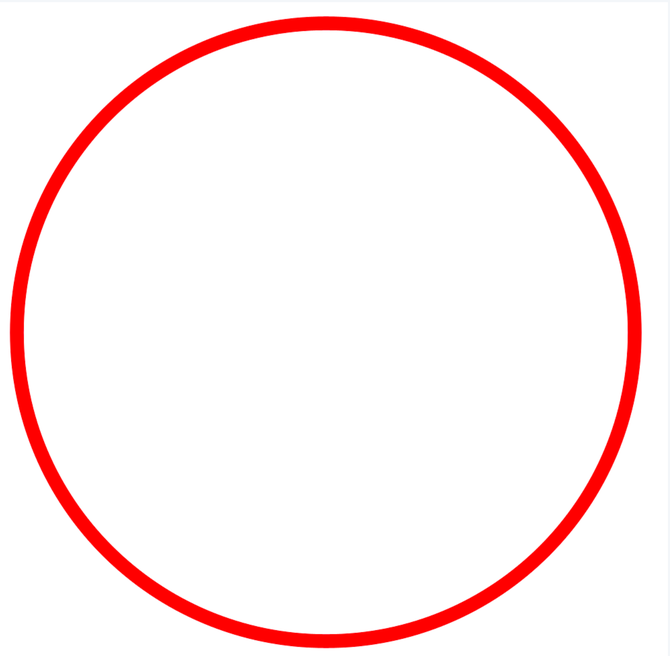
```java
public class Circle{
    public static void main(String[] args){
        StdDraw.setScale(0,2);
        StdDraw.setPenRadius(0.02);
        double radius = 0.7;
        for(int i = 0; i < 3; i++){
            int r = (int)(Math.random()* 255);
            int g = (int)(Math.random()* 255);
            int b = (int)(Math.random()* 255);
            StdDraw.setPenColor(r,g,b);
            StdDraw.circle(1,1,radius);
            radius = radius - 0.2;
        }
    }
}
```

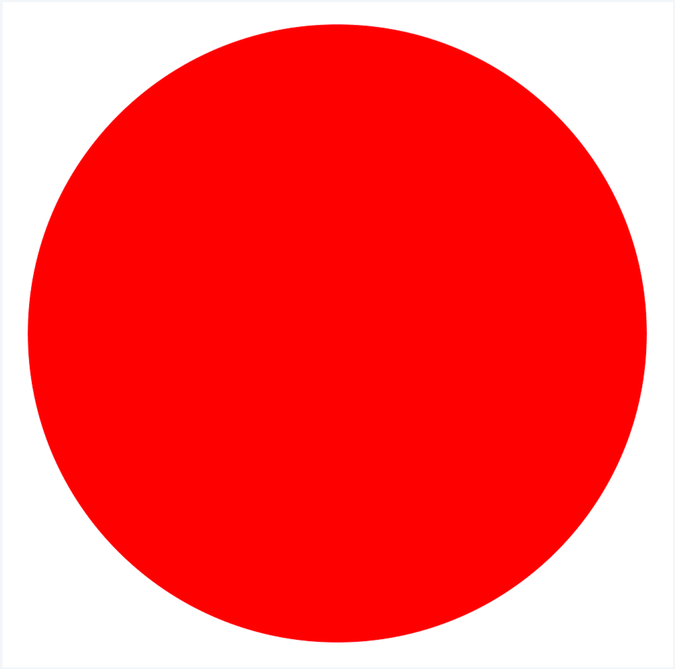
StdDraw.circle(1,1,r); //center-x, center-y, radius
StdDraw.**filledCircle**(1,1,r); //center-x, center-y, radius

Circles

```
StdDraw.setPenRadius(0.02);
StdDraw.setPenColor(StdDraw.RED);
StdDraw.setScale(0,2);
double r = 0.9;
StdDraw.circle(1,1,r);
//  StdDraw.filledCircle(1,1,radius);
```



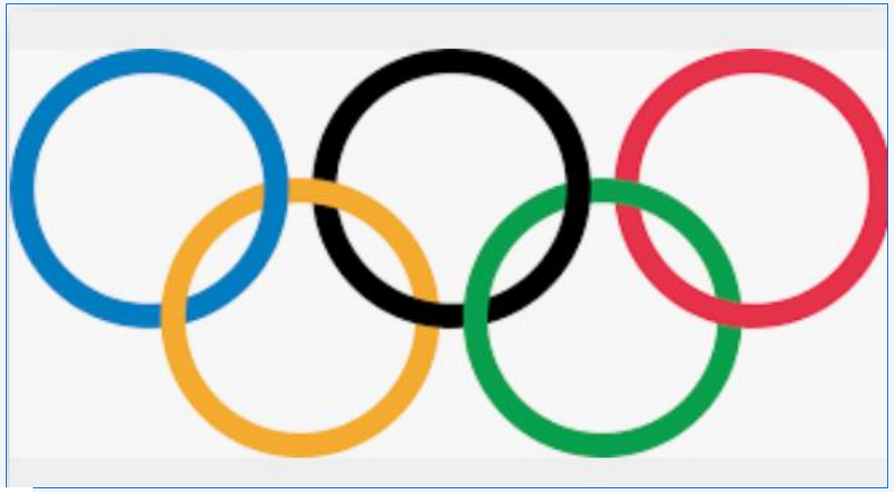






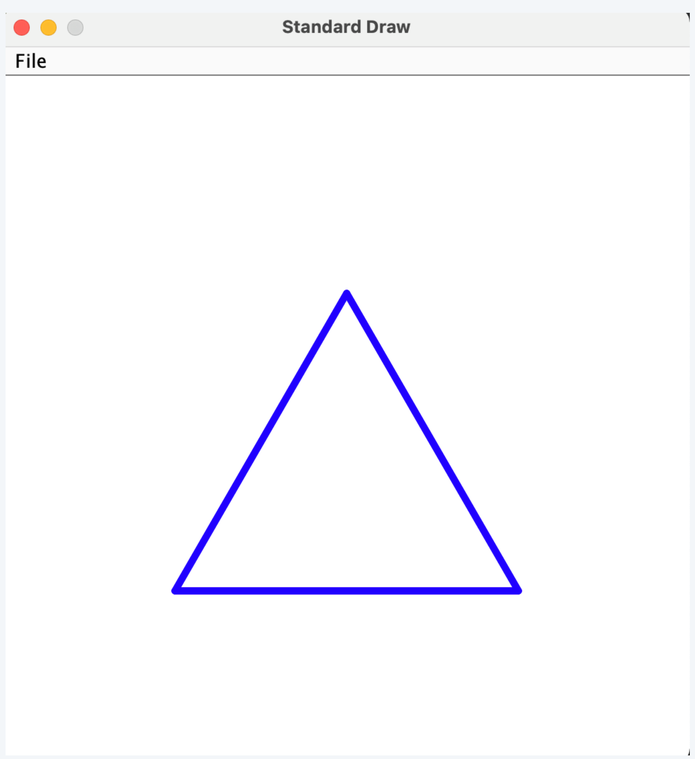
What about????

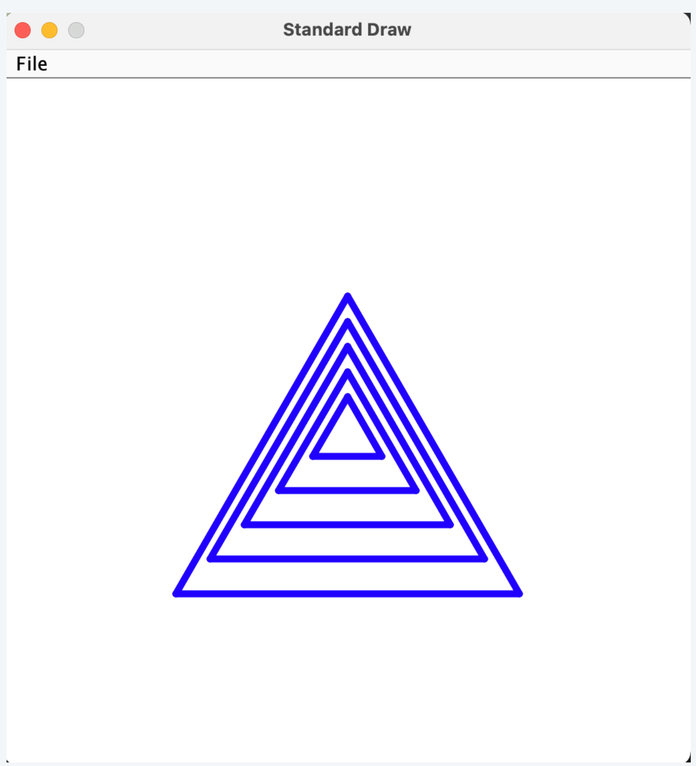


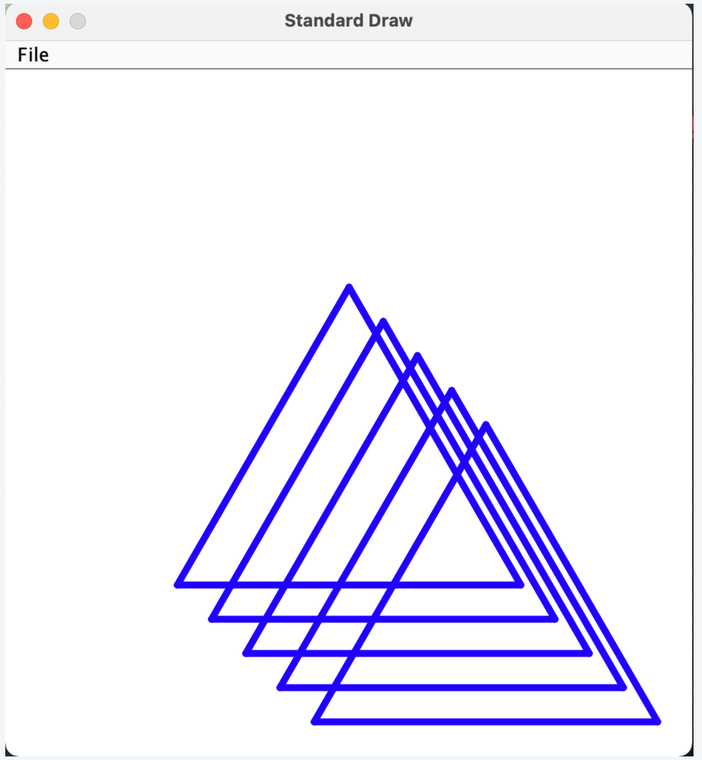




Can you modify your original TRIANGLE program to display the ones below?







StdDraw.setScale(-0.5, 1.5);



4. Input and Output

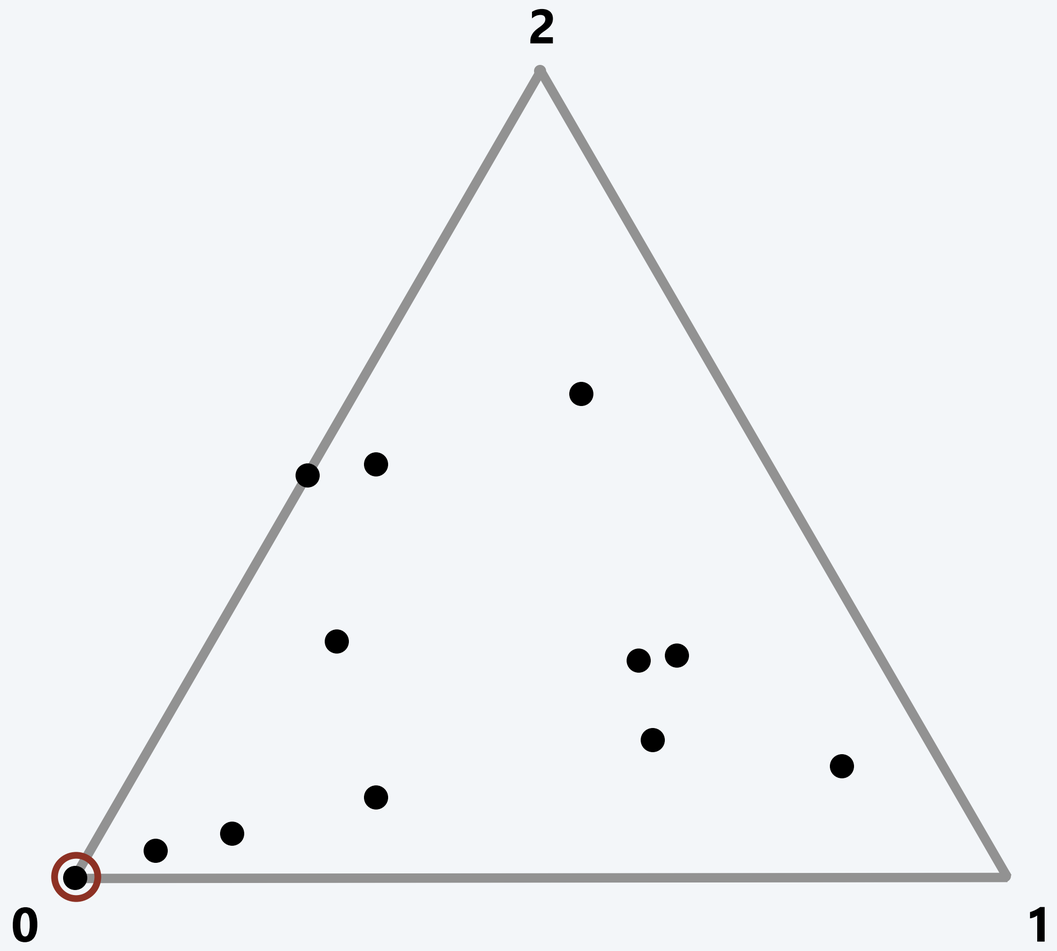
- Standard input and output
- Standard drawing
- Fractal drawings (optional)
- Animation

StdDraw application: a random game

Draw an equilateral triangle, number the vertices 0, 1, 2 and make 0 the *current point.*

- Pick a vertex at random.
- Draw a point halfway between that vertex and the current point.
- Repeat.

0	1	2	3	4	5	6	7	8	9	10
2	**1**	**2**	**0**	**1**	**0**	**0**	**0**	**2**	**1**	**1**



vertex	*ID*	*probability*	*new x*	*new y*
(0, 0)	0	1/3	.5*x*	.5*y*
(1, 0)	1	1/3	.5*x* + .5	.5*y*
(.5, √3/2)	2	1/3	.5*x* + .25	.5*y* + .433

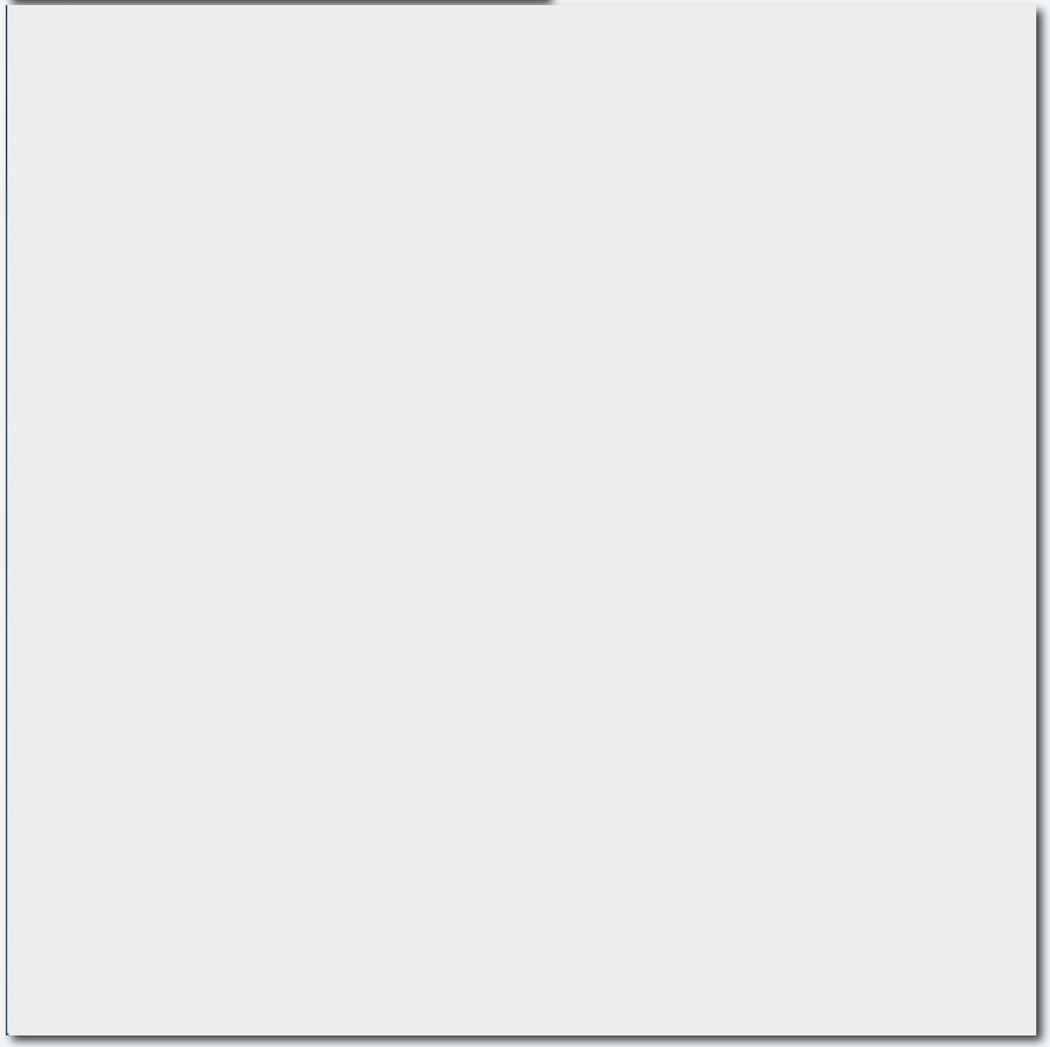
StdDraw application: a random game


```
public class Chaos
{
   public static void main(String[] args)
   {
      int trials = Integer.parseInt(args[0]);

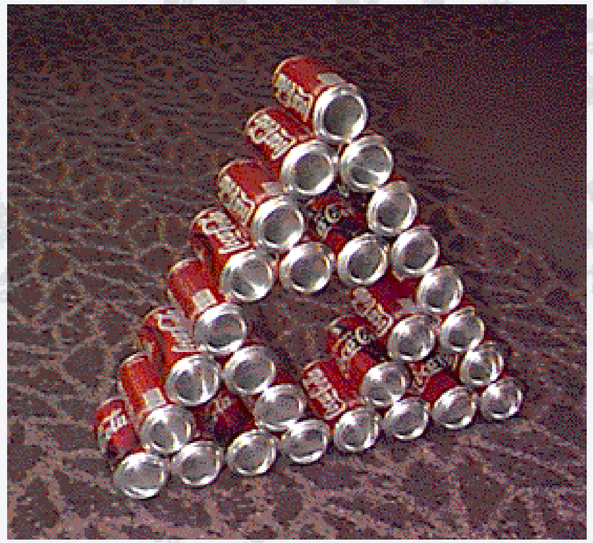
      double c = Math.sqrt(3.0) / 2.0;
      double[] cx = { 0.000, 1.000, 0.500 };
      double[] cy = { 0.000, 0.000, c };

      StdDraw.setPenRadius(0.01);
      double x = 0.0, y = 0.0;
      for (int t = 0; t < trials; t++)
      {
         int r = (int) (Math.random() * 3);
         x = (x + cx[r]) / 2.0;
         y = (y + cy[r]) / 2.0;
         StdDraw.point(x, y);
      }
   }
}
```

```
% java Chaos 10000
```

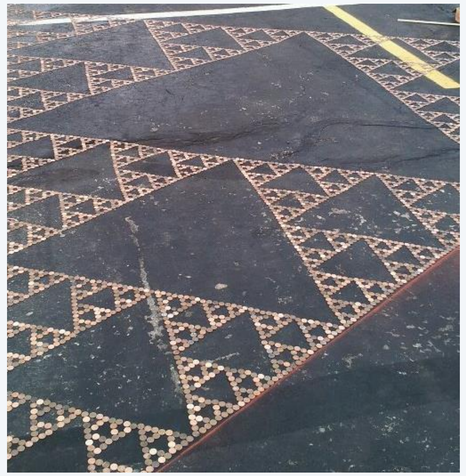


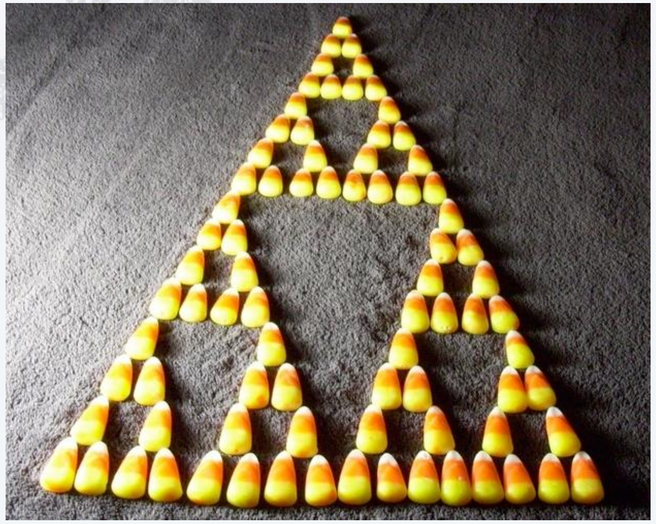
Sierpinski triangles in the wild

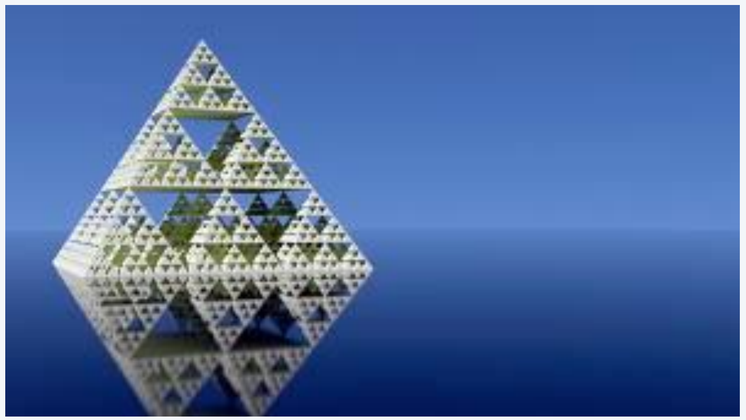


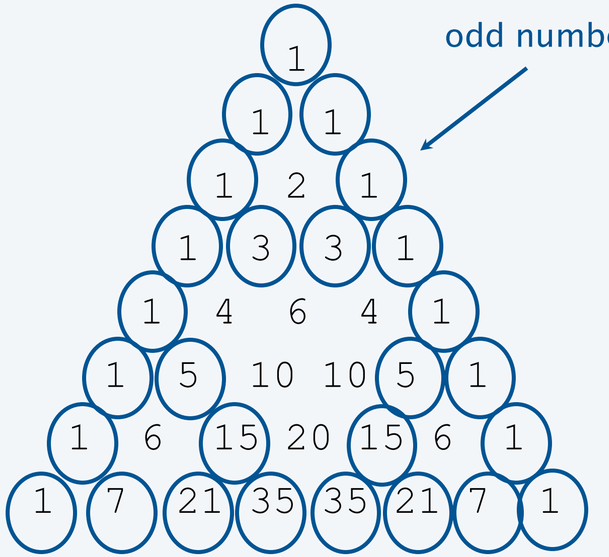












Pascal's triangle

Iterated function systems

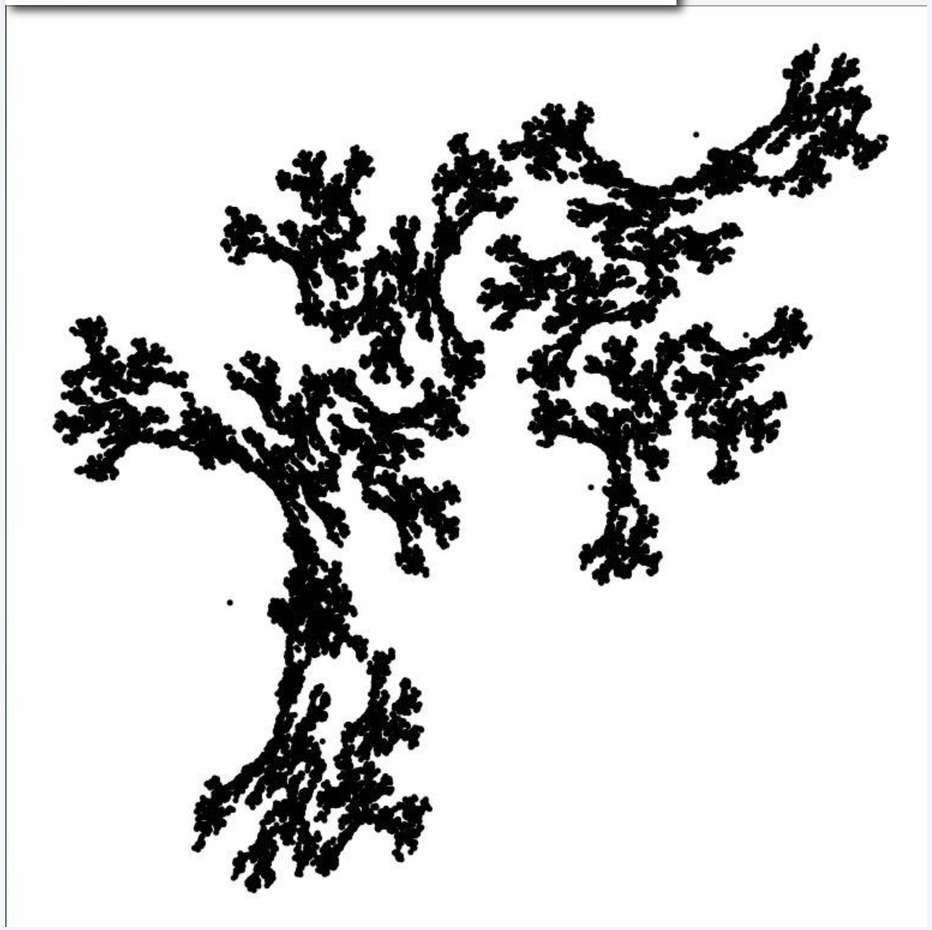
- What happens when we change the rules?

probability	*new x*	*new y*
40%	.31*x* − .53*y* + .89	−.46*x* − .29y + 1.10
15%	.31*x* − .08*y* + .22	.15*x* − .45*y* + .34
45%	.55*y* + .01	.69*x* − .20*y* + .38

IFS.java (Program 2.2.3) is a *data-driven* program that takes the coefficients from *standard input.*

```
% more coral.txt
3
   0.40  0.15  0.45
3 3
   0.307692 -0.531469  0.8863493
   0.307692 -0.076923  0.2166292
   0.000000  0.545455  0.0106363
3 3
  -0.461538 -0.293706  1.0962865
   0.153846 -0.447552  0.3383760
   0.692308 -0.195804  0.3808254
```

```
% java IFS 10000 < coral.txt
```



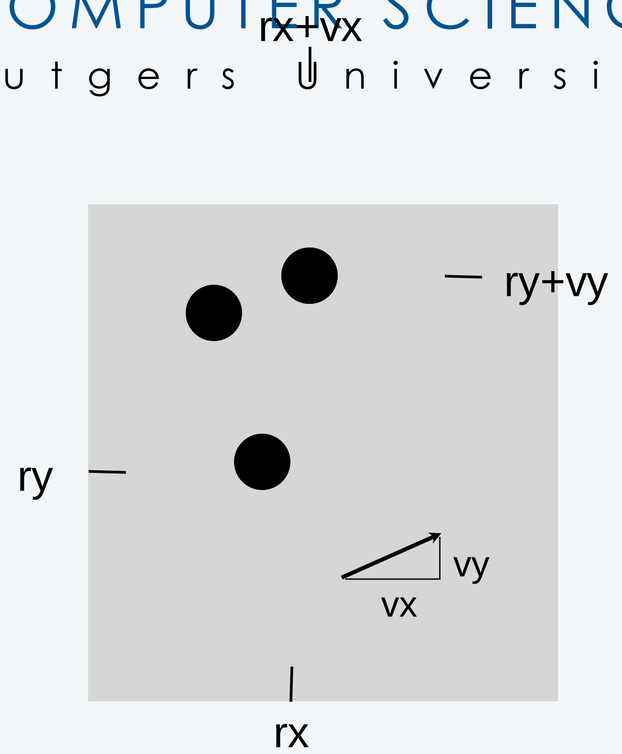


4. Input and Output

- Standard input and output
- Standard drawing
- Fractal drawings
- Animation (optional)

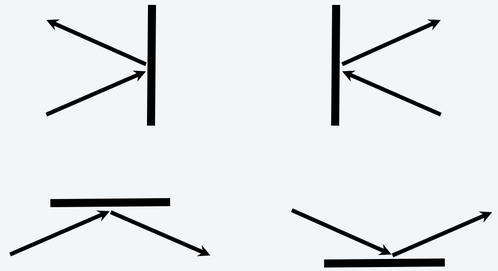
Animation

- To create animation with StdDraw.
- Repeat the following:
- Clear the screen.
- Move the object.
- Draw the object.
- Display and pause briefly.
- When display time is much greater than the screen-clear time, we have the illusion of motion.



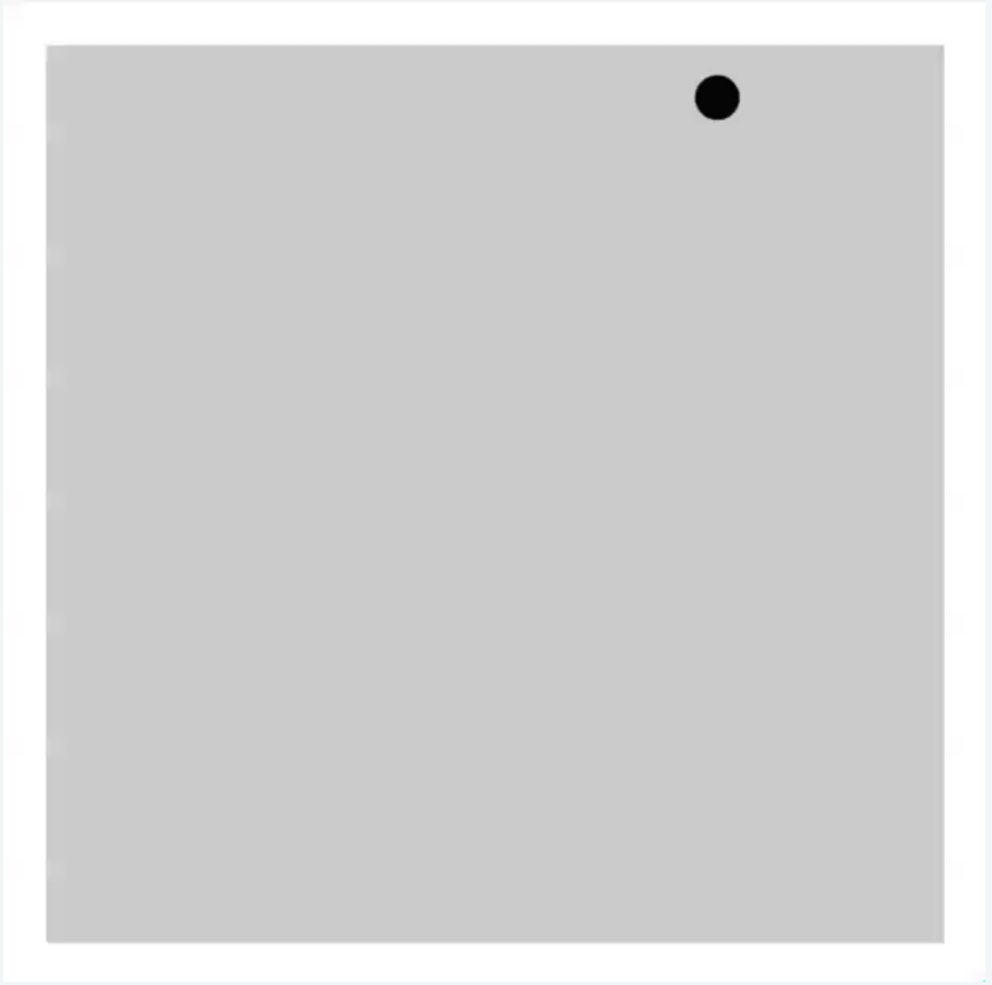
Bouncing ball.

- Ball has position (rx, ry) and constant velocity (vx, vy).
- To *move* the ball, update position to (rx+vx, ry+vy).
- If the ball hits a *vertical* wall, set vx to -vx.
- If the ball hits a *horizontal* wall, set vy to -vy.



```
public class BouncingBall
{
   public static void main(String[] args)
   {
      double rx = .480, ry = .860;
      double vx = .015, vy = .023;
      double radius = .05;
      StdDraw.setXscale(-1.0, +1.0);
      StdDraw.setYscale(-1.0, +1.0);
      while(true)
      {
        StdDraw.setPenColor(StdDraw.LIGHT_GRAY);
        StdDraw.filledSquare(0.0, 0.0, 1.0);
        if (Math.abs(rx + vx) + radius > 1.0) vx = -vx;
        if (Math.abs(ry + vy) + radius > 1.0) vy = -vy;
        rx = rx + vx;
        ry = ry + vy;
        StdDraw.setPenColor(StdDraw.BLACK);
        StdDraw.filledCircle(rx, ry, radius);
        StdDraw.show(20);
      }
   }
}
```

```
% java BouncingBall
```



Pop quiz on animation

Q. What happens if we move *clear the screen* out of the loop?

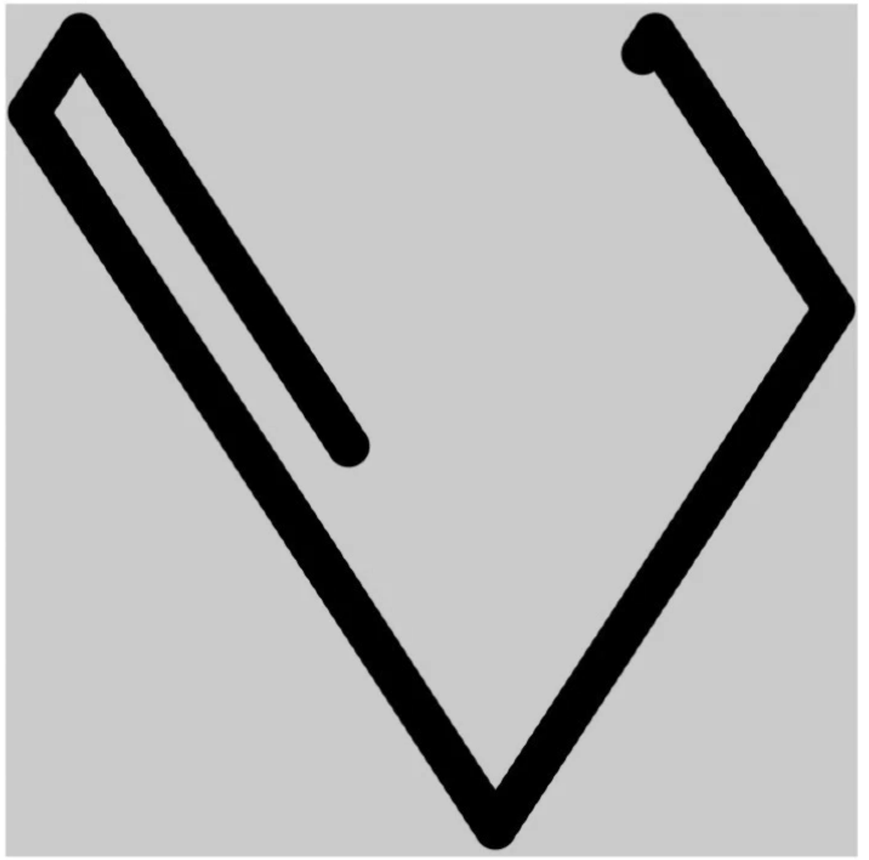
```
public class BouncingBall
{
   public static void main(String[] args)
   {
      double rx = .480, ry = .860;
      double vx = .015, vy = .023;
      double radius = .05;
      StdDraw.setXscale(-1.0, +1.0);
      StdDraw.setYscale(-1.0, +1.0);
     while(true)
      {  StdDraw.setPenColor(StdDraw. LIGHT_GRAY);
         StdDraw.filledSquare(0.0, 0.0, 1.0);

        if (Math.abs(rx + vx) + radius > 1.0) vx = -vx;
        if (Math.abs(ry + vy) + radius > 1.0) vy = -vy;
        rx = rx + vx;
        ry = ry + vy;
        StdDraw.setPenColor(StdDraw.BLACK);
        StdDraw.filledCircle(rx, ry, sz);
        StdDraw.show(20);
      }
   }
}
```

Pop quiz on animation

Q. What happens if we move *clear the screen* out of the loop?

```
public class BouncingBall
{
   public static void main(String[] args)
   {
      double rx = .480, ry = .860;
      double vx = .015, vy = .023;
      double radius = .05;
      StdDraw.setXscale(-1.0, +1.0);
      StdDraw.setYscale(-1.0, +1.0);
      StdDraw.setPenColor(StdDraw. LIGHT_GRAY);
      StdDraw.filledSquare(0.0, 0.0, 1.0);
      while(true)
      {
        if (Math.abs(rx + vx) + radius > 1.0) vx = -vx;
        if (Math.abs(ry + vy) + radius > 1.0) vy = -vy;
        rx = rx + vx;
        ry = ry + vy;
        StdDraw.setPenColor(StdDraw.BLACK);
        StdDraw.filledCircle(rx, ry, radius);
        StdDraw.show(20);
      }
   }
}
```

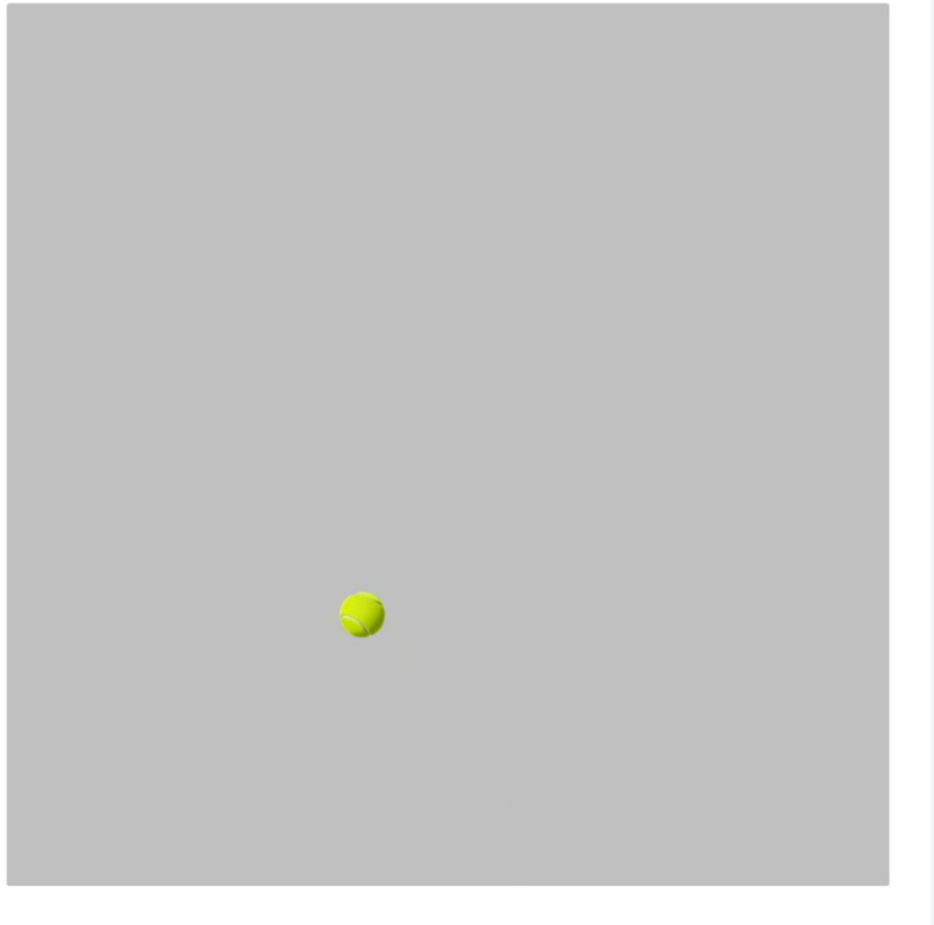


A. We see the ball's entire path.

Deluxe bouncing ball


```
public class BouncingBallDeluxe
{
   public static void main(String[] args)
   {
      double rx = .480, ry = .860;
      double vx = .015, vy = .023;
      double radius = .05;
      StdDraw.setXscale(-1.0, +1.0);
      StdDraw.setYscale(-1.0, +1.0);
      while(true)
      {
        StdDraw.setPenColor(StdDraw. LIGHT_GRAY);
        StdDraw.filledSquare(0.0, 0.0, 1.0);
        if (Math.abs(rx + vx) + radius > 1.0)
        {  StdAudio.play("pipebang.wav"); vx = -vx;  }
        if (Math.abs(ry + vy) + radius > 1.0)
        {  StdAudio.play("pipebang.wav"); vy = -vy;  }
        rx = rx + vx;
        ry = ry + vy;
        StdDraw.picture(rx, ry, "TennisBall.png");
        StdDraw.show(20);
      }
   }
```

```
% java BouncingBallDeluxe
```



Stay tuned to next lecture for full description of StdAudio.

Image sources

http://paulbourke.net/fractals/gasket/cokegasket.gif

http://www.buzzfeed.com/atmccann/11-awesome-math-foods#39wokfk

http://sheilakh.deviantart.com/art/The-Legend-of-Sierpinski-308953447

http://commons.wikimedia.org/wiki/File:Lady_Fern_frond_-_normal_appearance.jpg

http://img3.wikia.nocookie.net/__cb20100707172110/jamescameronsavatar/images/e/e1/Avatar_concept_art-3.jpg

Image sources

http://www.digitalreins.com/wp-content/uploads/2013/05/Binary-code.jpg

http://en.wikipedia.org/wiki/Punched_tape#mediaviewer/File:Harwell-dekatron-witch-10.jpg



1.5: Input and Output

- Standard input and output
- Standard drawing
- Fractal drawings
- Animation



INTRODUCTION TO COMPUTER SCIENCE

Rutgers University

6. Input / Output

http://introcs.cs.rutgers.edu