



# CMSC 150

# INTRODUCTION TO COMPUTING

LAB – WEEK 4

- DRAWING WITH JAVA

# INPUT AND OUTPUT

- Input devices



Keyboard



Mouse



Hard drive



Network



Digital camera



Microphone

- Output devices.



Display



Speakers



Hard drive



Network



Printer



MP3 Player

- Goal. Java programs that interact with the outside world.

- Java Libraries support these interactions
- We use the Operating System (OS) to connect our program to them

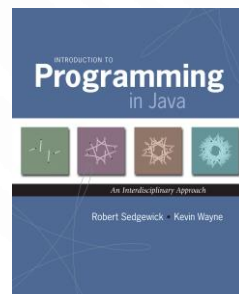
The background features a series of concentric, light blue circles centered in the middle. In the four corners, there are stylized circuit board traces in a light blue color, consisting of straight lines and small circles representing components or vias.

# STANDARD DRAWING

# STANDARD DRAWING

- Standard drawing (StdDraw) is library for producing graphical output which the authors of your book made.

library developed  
for this course  
(not for broad usage!)



```
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 filledCircle(double x, double y, double r)
    void square(double x, double y, double r)
    void filledSquare(double x, double y, double r)
    void polygon(double[] x, double[] y)
    void filledPolygon(double[] x, double[] y)

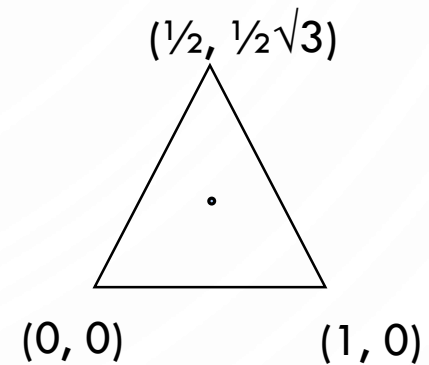
    void setXscale(double x0, double x1)      reset x range to (x0, x1)
    void setYscale(double y0, double y1)      reset y range to (y0, y1)
    void setPenRadius(double r)               set pen radius to r
    void setPenColor(Color c)                 set pen color to c
    void setFont(Font f)                      set text font to f
    void setCanvasSize(int w, int h)          set canvas to w-by-h window
    void clear(Color c)                        clear the canvas; color it c
    void show(int dt)                          show all; pause dt milliseconds
    void save(String filename)                 save to a .jpg or w.png file
}
```

*Note: Methods with the same names but no arguments reset to default values.*

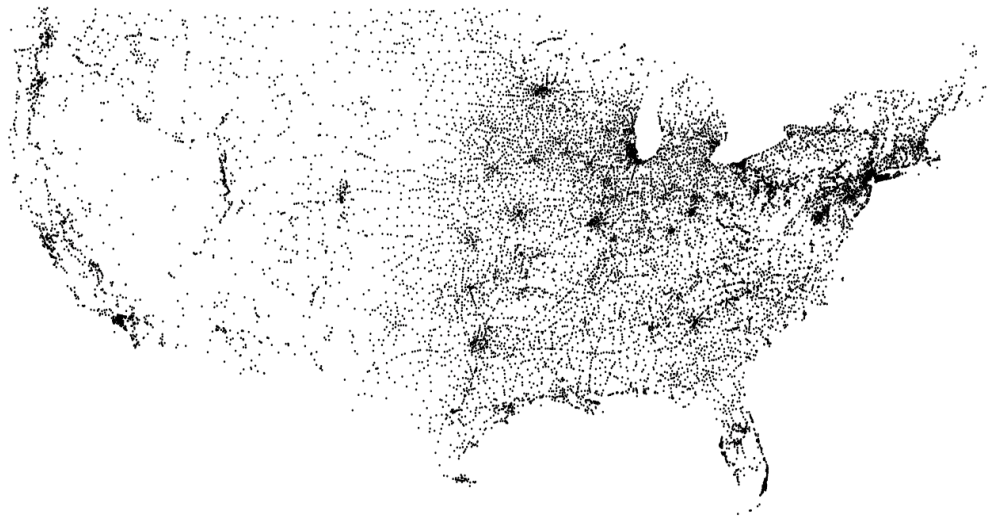
# STANDARD DRAW

- Practice with StdDraw. To use: download StdDraw.java and put in working directory.

```
1. public class Triangle {  
2.     public static void main(String[] args) {  
3.         double t = Math.sqrt(3.0) / 2.0;  
4.         StdDraw.line(0.0, 0.0, 1.0, 0.0);  
5.         StdDraw.line(1.0, 0.0, 0.5, t);  
6.         StdDraw.line(0.5, t, 0.0, 0.0);  
7.         StdDraw.point(0.5, t/3.0);  
8.     }  
9. }
```



# DATA VISUALIZATION



```
% more < USA.txt
669905.0 247205.0 1244962.0 490000.0
1097038.8890 245552.7780
1103961.1110 247133.3330
1104677.7780 247205.5560
...

% java PlotFilter < USA.txt
```

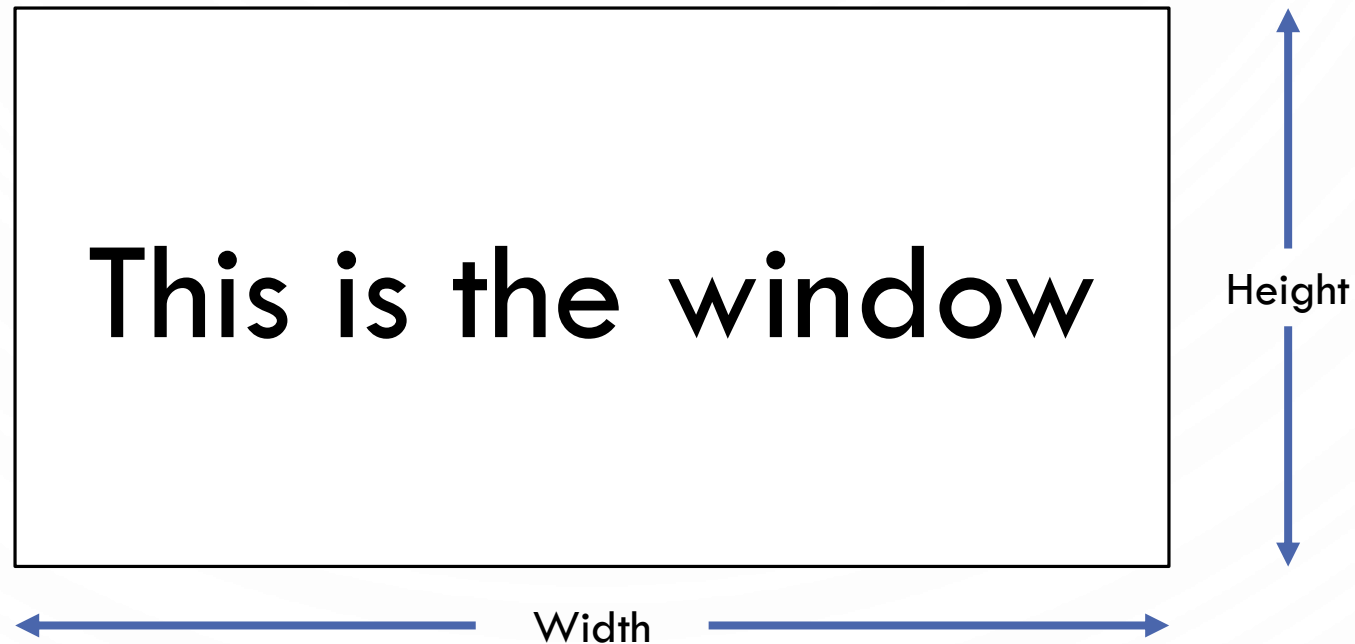
bounding box

coordinates of  
13,509 US cities

```
1. public class PlotFilter {
2.     public static void main(String[] args) {
3.         double xmin = StdIn.readDouble();
4.         double ymin = StdIn.readDouble();
5.         double xmax = StdIn.readDouble();
6.         double ymax = StdIn.readDouble();
7.         StdDraw.setXscale(xmin, xmax); //Set up coordinate
8.         StdDraw.setYscale(ymin, ymax); //system.
9.
10.        while (!StdIn.isEmpty()) {
11.            double x = StdIn.readDouble();
12.            double y = StdIn.readDouble();
13.            StdDraw.point(x, y);
14.        }
15.    }
```

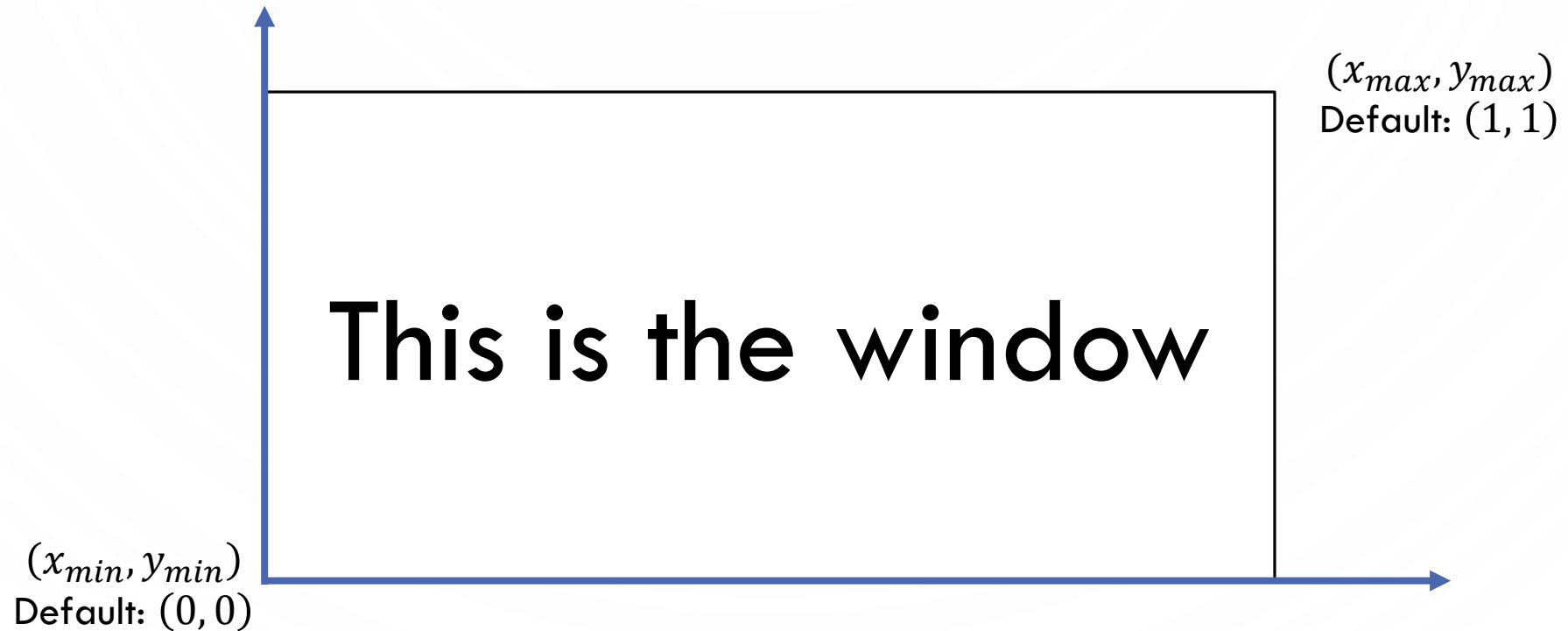
# SET SIZE OF WINDOW

- Use `StdDraw.setCanvasSize(width, height)`
  - Width and height are integers representing pixels



# COORDINATE SYSTEM WITH STDDRAW

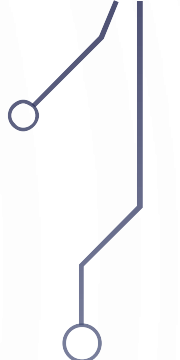
- Use `StdDraw.setXscale(xmin, xmax)` and `StdDraw.setYscale(ymin, ymax)`
  - `xmin`, `xmax`, `ymin`, and `ymax` are doubles. Note the difference between pixels!







# COLORS

- Change color with `StdDraw.setPenColor(Color)`
    - Use `StdDraw.BLACK`, `StdDraw.WHITE`, `StdDraw.BLUE`, `StdDraw.RED`, etc
    - Can define own colors with Java color library (uses RGB)
- 

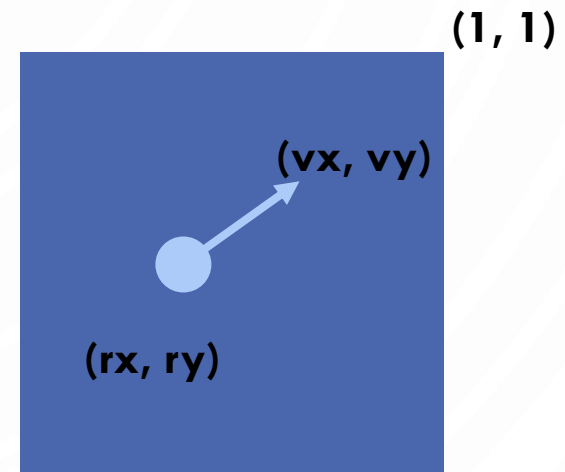
# ANIMATION

- Animation loop. Repeat the following:

- Clear the screen.
- Move the object.
- Draw the object.
- Display and pause for a short while.

- Ex. Bouncing ball.

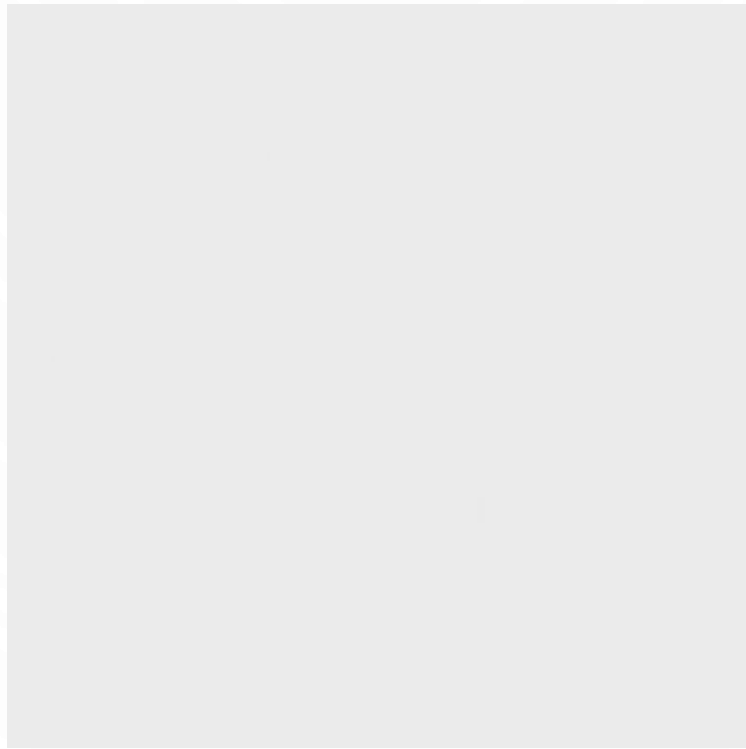
- Ball has position  $(rx, ry)$  and constant velocity  $(vx, vy)$ .
- Detect collision with wall and reverse velocity.  $(-1, -1)$



# BOUNCING BALL

```
1. public class BouncingBall {
2.     public static void main(String[] args) {
3.         double rx = .480, ry = .860;           //x, y position
4.         double vx = .015, vy = .023;         //x, y velocity
5.         double radius = .05;                 //radius of ball
6.         StdDraw.setXscale(-1.0, 1.0); StdDraw.setYscale(-1.0, 1.0); //Set coordinate system
7.
8.         while(true) {                         //Simulation loop
9.             if (Math.abs(rx + vx) + radius > 1.0) vx = -vx; //Update ball velocity if at the boundary
10.            if (Math.abs(ry + vy) + radius > 1.0) vy = -vy;
11.
12.            rx = rx + vx; ry = ry + vy;        //Update position (add velocity)
13.
14.            StdDraw.setPenColor(StdDraw.GRAY); //Clear screen
15.            StdDraw.filledSquare(0.0, 0.0, 1.0);
16.            StdDraw.setPenColor(StdDraw.BLACK); //Render ball
17.            StdDraw.filledCircle(rx, ry, radius);
18.            StdDraw.show();
19.            StdDraw.pause(20);                 //Pause for 20ms so that we can see it nicely
20.        }
21.    }
22. }
```

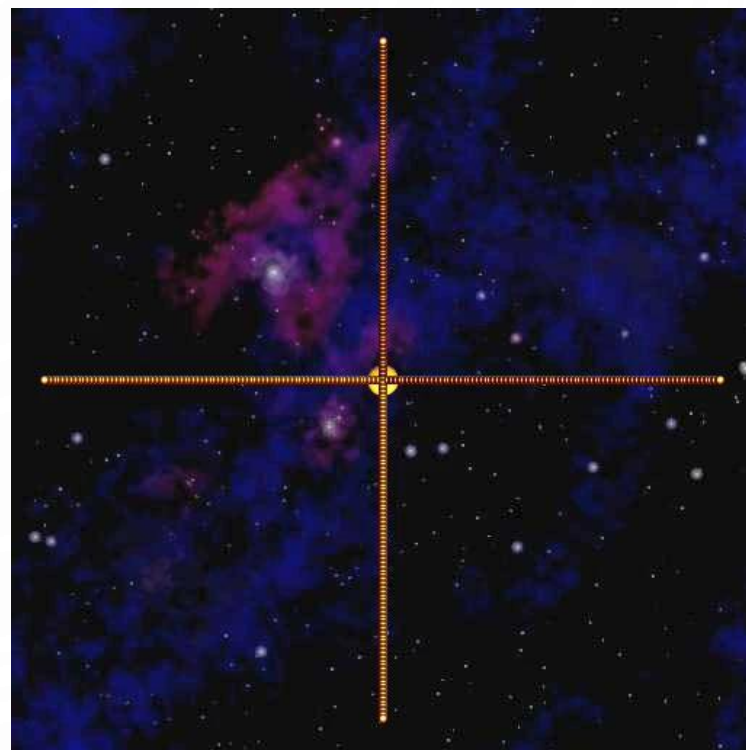
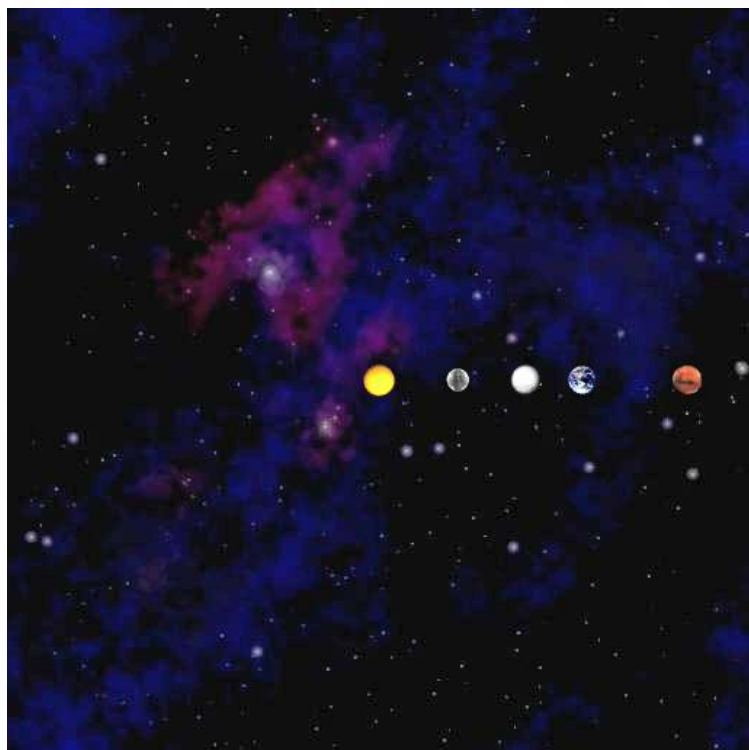
# BOUNCING BALL DEMO



# SPECIAL EFFECTS

- Images. Put .gif, .png, or .jpg file in the working directory and use `StdDraw.picture()` to draw it.
- Sound effects. Put .wav, .mid, or .au file in the working directory and use `StdAudio.play()` to play it. Will need the `StdAudio.java` file from book website.

# EXAMPLES

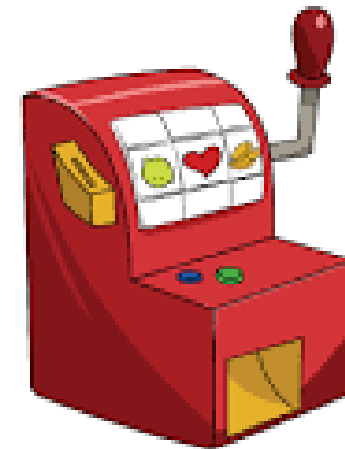


The image features a light blue background with a subtle pattern of concentric circles. In the four corners, there are decorative circuit-like line art elements consisting of thin lines and small circles, resembling a stylized PCB or network diagram.

# VISUAL DISPLAY FOR SLOT MACHINE

# EXERCISE – IN TRIPLETS

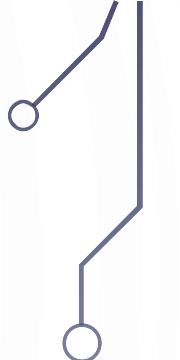
- Yes you have to be with someone!
- Starters: You work for JLDiablo Consultants Inc., which specializes in making software for Casino games (Cha-ching! \$\$\$\$). A new casino in Reno needs a slot game called Binary Slots 101010.
  - How it works:
    - A player enters a bet of their choice
    - Three Boolean values are randomly generated
    - If they are all true, then the player earns twice their money back!







# PROGRAM IS TOO LARGE FOR SLIDES!

- Each piece of code fits piecewise. It is not difficult to put them together.
- 

## EXERCISE – INITIALIZATION STEP

1. `//Initialize`
2. `StdDraw.setCanvasSize(600, 600); // Resize the window`
3. `StdDraw.setXscale(-100, 100); //Redefine coordinate system`
4. `StdDraw.setYscale(-100, 100);`
5. `StdDraw.enableDoubleBuffering(); //Allows for smooth animation`

# EXERCISE – GAME LOOP

```
1. //Play game until quit 'p' is pressed
2. while(true) {
3.     //Wait for key press
4.     while(!StdDraw.hasNextKeyTyped()) {
5.         StdDraw.show();
6.         StdDraw.pause(33);
7.     }
8.     char key = StdDraw.nextKeyTyped();
9.     if(key == 'q')
10.        break;
11. }
12. System.exit(0);
```

# EXERCISE – BEFORE THE KEY PRESSED. SHOW A MESSAGE FOR USER TO START THE GAME

1. `//Draw slot machine.`
2. `StdDraw.textRight(90, -70, "Binary Slots 101!");`
3. `StdDraw.textRight(90, -80, "Match 3 to win");`
4. `StdDraw.textRight(90, -90, "Press a key to play. Q to quit.");`
5. `StdDraw.circle(-55, -5, 10);`
6. `StdDraw.circle(-5, -5, 10);`
7. `StdDraw.circle(45, -5, 10);`

# EXERCISE – NOW WE CAN DO OUR MAIN GAME LOOP

```
1. //Simulate spin. Randomly switch color of circles
2. boolean a = false, b = false, c = false;
3. for(int i = 1; i <= 30; ++i) {
4.     StdDraw.clear(); //Clear
5.
6.     StdDraw.textRight(90, -70, "Binary Slots 101!"); //Redraw text
7.     StdDraw.textRight(90, -80, "Match 3 to win");
8.
9.     a = Math.random() < 0.5; b = Math.random() < 0.5; c = Math.random() < 0.5; //Simulate game
10.
11.    if(a) StdDraw.setPenColor(StdDraw.BLUE); //Draw all the circles
12.    else StdDraw.setPenColor(StdDraw.RED);
13.    StdDraw.filledCircle(-55, -5, 10);
14.    if(b) StdDraw.setPenColor(StdDraw.BLUE);
15.    else StdDraw.setPenColor(StdDraw.RED);
16.    StdDraw.filledCircle(-5, -5, 10);
17.    if(c) StdDraw.setPenColor(StdDraw.BLUE);
18.    else StdDraw.setPenColor(StdDraw.RED);
19.    StdDraw.filledCircle(45, -5, 10);
20.    StdDraw.setPenColor(StdDraw.BLACK);
21.    StdDraw.circle(-55, -5, 10); StdDraw.circle(-5, -5, 10); StdDraw.circle(45, -5, 10);
22.
23.    StdDraw.show(); //Render
24.    StdDraw.pause(i*10); //Pause
25. }
```

## EXERCISE – FINISH OFF WITH A FINAL MESSAGE

1. `//Draw final slot machine`
2. `StdDraw.pause(1000);`
3. **if**(`a && b && c`)
4.     `StdDraw.text(-20, 50, "You win!");`
5. **else**
6.     `StdDraw.text(-20, 50, "You lose :(");`
7. `StdDraw.show();`

# EXERCISE

- Until the end of lab work with your team to improve the slots game.

## Possibilities:

- Track user's money
- Allow a varying bet based on key presses
- Make it visually appealing
- Add pictures for the background/etc

