MOBI3002: A3 – Bouncing

Assignment Submission

By: Connor Goodwin

W#: W0488245

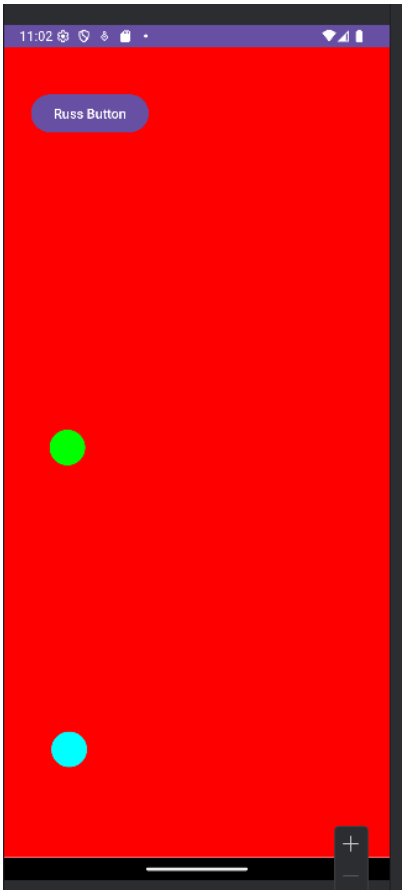
Date: 2025-09-22

1. **[10%]** Change the box color… i.e. the background colour. What did you do?

   I went to the BoucingBallView class and changed the object instantiation's constructor paremetor from BLACK to RED.

   Class: BouncingBallView

```
31      // create the box
32      box = new Box(Color.RED);  // ARGB
```

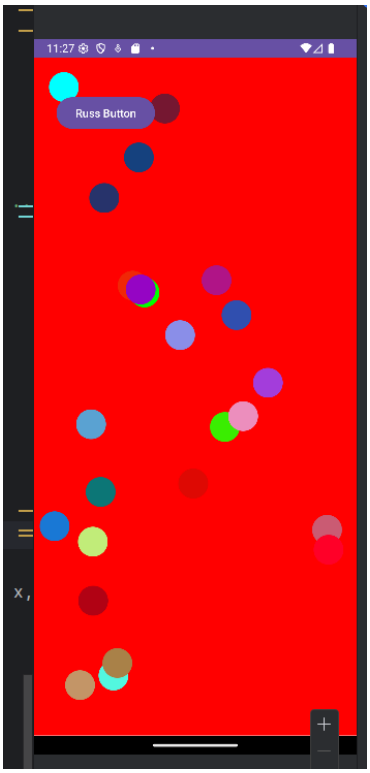


2. **[10%]** Change the color of newly made ball to a new random color for each new ball. What did you do?

   I generated 3 random numbers, one for red, one for green, and one for blue. I used the Color.rgb method, and passed in the generated colors for the red green and blue arguments for the rgb method.

   Class: BouncingBallView

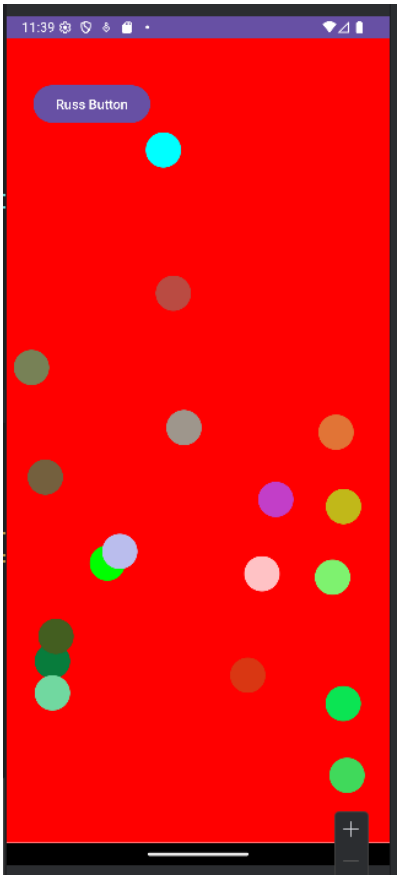
```
162         int randomRed = rand.nextInt( bound: 256);
163         int randomGreen = rand.nextInt( bound: 256);
164         int randomBlue = rand.nextInt( bound: 256);
165         balls.add(new Ball((Color.rgb(randomRed, randomGreen, randomBlue)), x, y, dx, dy));  // ad
```



3. **[10%]** Make the newly made balls go super-fast and super-slow (with a code change). What did you do?

   I made a variable called: “fastOrSlow”, that will generate either a 1 or 2. If 1 is generated then the ball will be fast, otherwise the ball will be slow.

```
163
164         int fastOrSlow = rand.nextInt( bound: 3);
165
166         if (fastOrSlow == 1) {
167             dx = 150;
168             dy = 150;
169         } else {
170             dx = 10;
171             dy = 10;
172         }
```



4. **[10%]** Try different approaches for invalidate() (different code locations, methods, ...):

```
104    public void onSizeChanged(int w, int h, int oldW, int oldH) {
105        this.invalidate();
106        // Set the movement bounds for the ball
107        box.set( x: 0,  y: 0, w, h);
108        Log.w( tag: "BouncingBallLog",  msg: "onSizeChanged w=" + w + " h=" + h);
109    }
```

```
112        @Override
113        public boolean onTouchEvent(MotionEvent event) {
114            this.invalidate();
```

```
152    public void RussButtonPressed() {
153        this.invalidate();
154        Log.d( tag: "BouncingBallView  BUTTON",  msg: "User tapped the  button ... VIEW");
```

1. Does the program still work each time?

   Each time it doesn't seem to change anything.

2. What does invalidate() do? What happens when it isn't called at all?

   When it is not called within the ondraw method:

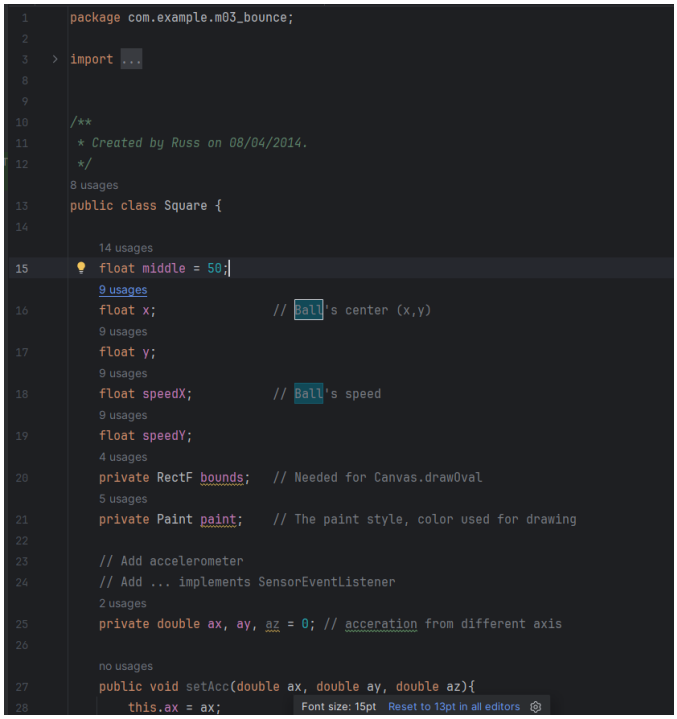
   ```
   99        //        this.invalidate();
   ```

   It seems to freeze all the balls until another ball is spawned. Invalidate will say that everything on the screen should be updated the next time onDraw is called.

3. What are the times that onDraw() is called?

   onDraw is never explicitly called by any line in the code, it is called within the android framework itself. Invalidate tells the framework that onDraw should be called again in the next frame.

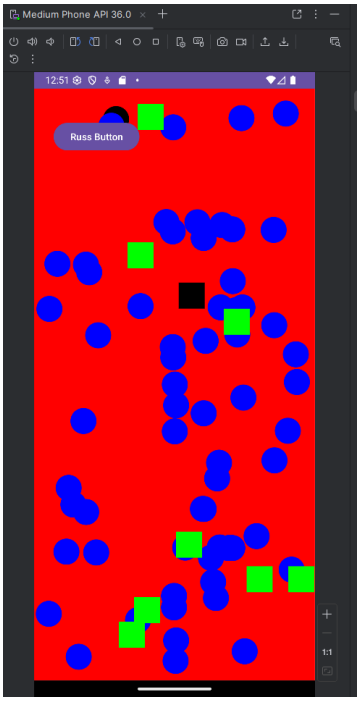
5. **[10%]** Add a square shape class…fast swipes makes the square shape, slow swipes make circles.

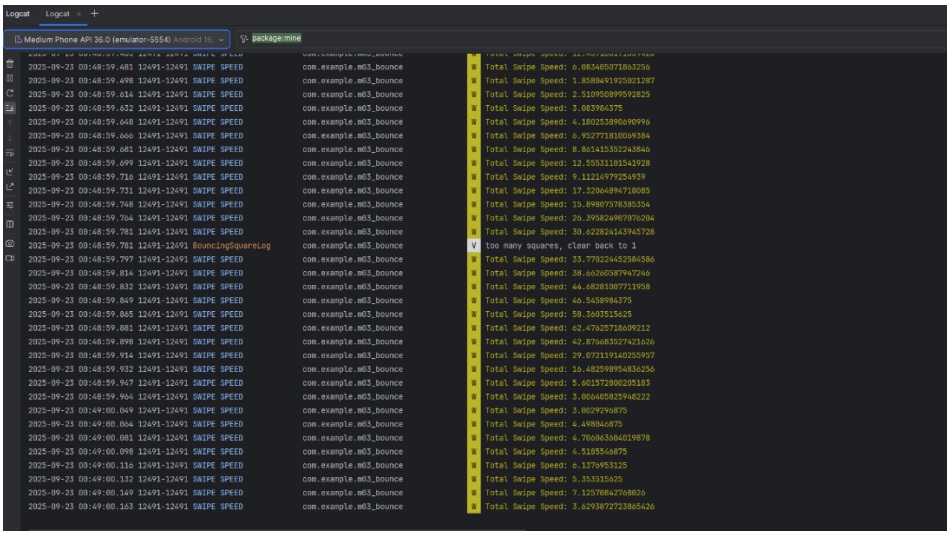


```
package com.example.m03_bounce;

import ...


/**
 * Created by Russ on 08/04/2014.
 */
8 usages
public class Square {

    14 usages
    float middle = 50;
    9 usages
    float x;                // Ball's center (x,y)
    9 usages
    float y;
    9 usages
    float speedX;           // Ball's speed
    9 usages
    float speedY;
    4 usages
    private RectF bounds;   // Needed for Canvas.drawOval
    5 usages
    private Paint paint;    // The paint style, color used for drawing

    // Add accelerometer
    // Add ... implements SensorEventListener
    2 usages
    private double ax, ay, az = 0; // acceration from different axis

    no usages
    public void setAcc(double ax, double ay, double az){
        this.ax = ax;
```

Font size: 15pt Reset to 13pt in all editors

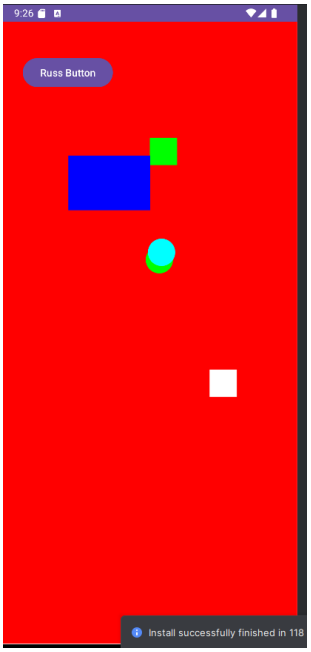
```
                double xSpeed = Math.abs(deltaX);
                double ySpeed = Math.abs(deltaY);
                double totalSpeed = Math.sqrt((xSpeed * xSpeed) + (ySpeed * ySpeed));
                Log.w( tag: "SWIPE SPEED", msg: "Total Swipe Speed: " + totalSpeed);

                if (totalSpeed >= 25) {
                    square_1.speedX += deltaX * scalingFactor;
                    square_1.speedY += deltaY * scalingFactor;
                    squares.add(new Square(Color.GREEN, previousX, previousY, deltaX, deltaY));  // a
                } else {
                    ball_1.speedX += deltaX * scalingFactor;
                    ball_1.speedY += deltaY * scalingFactor;
                    balls.add(new Ball(Color.BLUE, previousX, previousY, deltaX, deltaY));  // add ba
                }

                // A way to clear list when too many balls
                if (balls.size() > 100) {
                    // leave first ball, remove the rest
                    Log.v( tag: "BouncingBallLog", msg: "too many balls, clear back to 1");
                    balls.clear();
                    balls.add(new Ball(Color.BLACK));
                    ball_1 = balls.get(0);  // points ball_1 to the first (zero-ith) element of list
                }

                if (squares.size() > 100) {
                    // leave first ball, remove the rest
                    Log.v( tag: "BouncingSquareLog", msg: "too many squares, clear back to 1");
                    squares.clear();
                    squares.add(new Square(Color.BLACK));
                    square_1 = squares.get(0);
                }
```

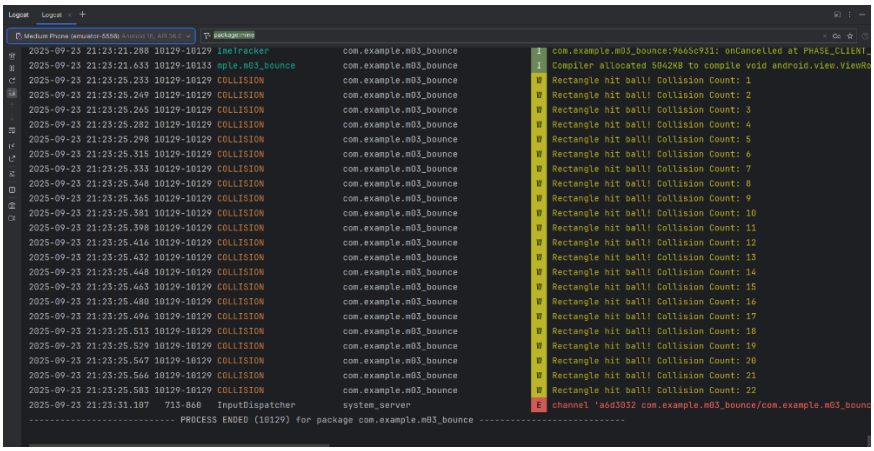

Medium Phone API 36.0
12:51
Russ Button
1:1


Logcat
Logcat
Medium Phone API 36.0 (emulator-5554) Android 16,
package:mine
2025-09-23 09:48:59.481 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 6.083460371863256
2025-09-23 09:48:59.498 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 1.8588491925021287
2025-09-23 09:48:59.614 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 2.5109508995928825
2025-09-23 09:48:59.632 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 3.063984375
2025-09-23 09:48:59.648 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 4.180253090690996
2025-09-23 09:48:59.666 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 6.9327718108D69384
2025-09-23 09:48:59.681 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 8.861415352243846
2025-09-23 09:48:59.699 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 12.555311015419928
2025-09-23 09:48:59.716 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 9.112149779254939
2025-09-23 09:48:59.731 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 17.320648947180085
2025-09-23 09:48:59.748 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 15.898079578385354
2025-09-23 09:48:59.764 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 26.395824907976284
2025-09-23 09:48:59.781 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 30.622824143945728
2025-09-23 09:48:59.781 12491-12491 BouncingSquareLog com.example.m03_bounce V too many squares, clear back to 1
2025-09-23 09:48:59.797 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 33.770224452584586
2025-09-23 09:48:59.814 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 38.66260587947246
2025-09-23 09:48:59.832 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 44.68281087711958
2025-09-23 09:48:59.849 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 46.5458984375
2025-09-23 09:48:59.865 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 58.3603515625
2025-09-23 09:48:59.881 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 62.470257160992I2
2025-09-23 09:48:59.898 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 42.87668327421026
2025-09-23 09:48:59.914 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 29.072119140255957
2025-09-23 09:48:59.932 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 16.482598954836256
2025-09-23 09:48:59.947 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 5.601572800205183
2025-09-23 09:48:59.964 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 3.0604658259482222
2025-09-23 09:49:00.049 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 3.0029296875
2025-09-23 09:49:00.064 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 4.498046875
2025-09-23 09:49:00.081 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 4.7668636040019878
2025-09-23 09:49:00.098 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 4.5185546875
2025-09-23 09:49:00.116 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 6.1779953125
2025-09-23 09:49:00.132 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 5.353515625
2025-09-23 09:49:00.149 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 7.125768427068826
2025-09-23 09:49:00.163 12491-12491 SWIPE SPEED com.example.m03_bounce D Total Swipe Speed: 3.629387272365426

6. **[10/10%]** Add a rectangle shape…any time a shape collides with that rectangle you increment a score count (show score on logcat).

```
public void moveWithCollisionDetection(Box box, ArrayList<Ball> balls, ArrayList<Square> squares) {
    // Get new (x,y) position
    x += speedX;
    y += speedY;

    // Add acceleration to speed
    speedX += ax;
    speedY += ay;

    for (Ball ball : balls) {
        float closestX = Math.max(x - this.width, Math.min(ball.x, x + this.width));
        float closestY = Math.max(y - this.height, Math.min(ball.y, y + this.height));

        float dx = ball.x - closestX;
        float dy = ball.y - closestY;

        if (dx*dx + dy*dy < ball.radius * ball.radius) {
            this.collisionCount++;
            Log.w( tag: "COLLISION",  msg: "Rectangle hit ball!" + " Collision Count: " + this.collisionCount);
        }
    }
```





7. **[20%]** Think of another change (…and do that change) yourself, …What did you do? Show this in your MP4.

   I added an Image Rectangle class, that is just a rectangle but with an image on it.

```
public ImageRectangle(Context context, int imageResId) {
    bounds = new RectF();

    this.x = width;
    this.y = height;


    this.speedX = 10;
    this.speedY = 25;


    bitmap = BitmapFactory.decodeResource(context.getResources(), imageResId);
    bitmap = Bitmap.createScaledBitmap(bitmap, (int)(width*2), (int)(height*2), filter: true);
}
```