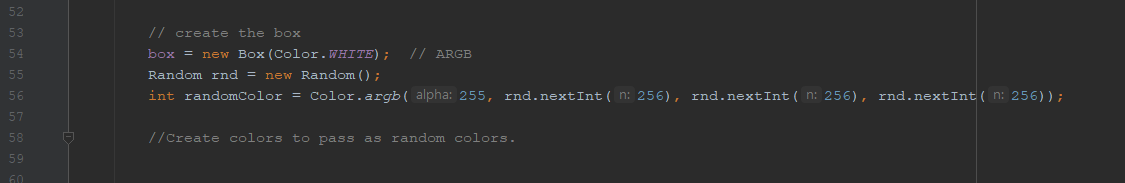
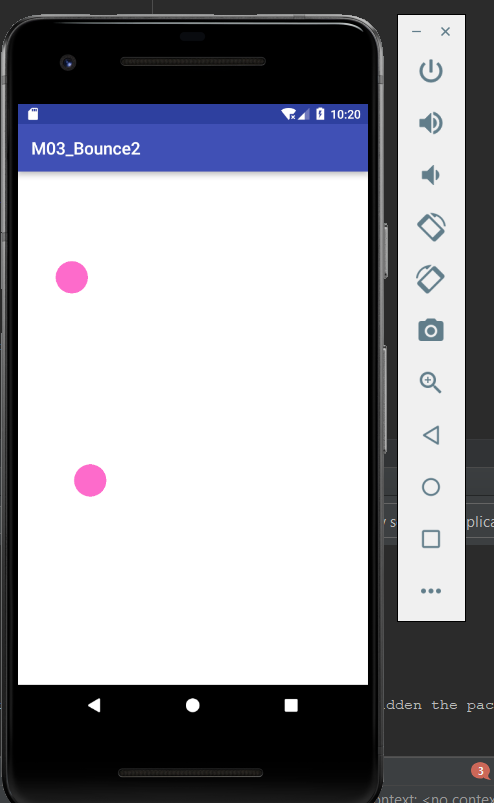
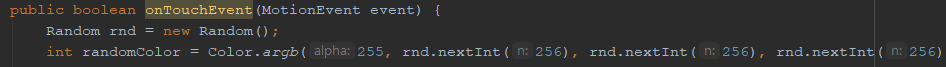
1. Change the box color...  i.e. the background colour. What did you do?

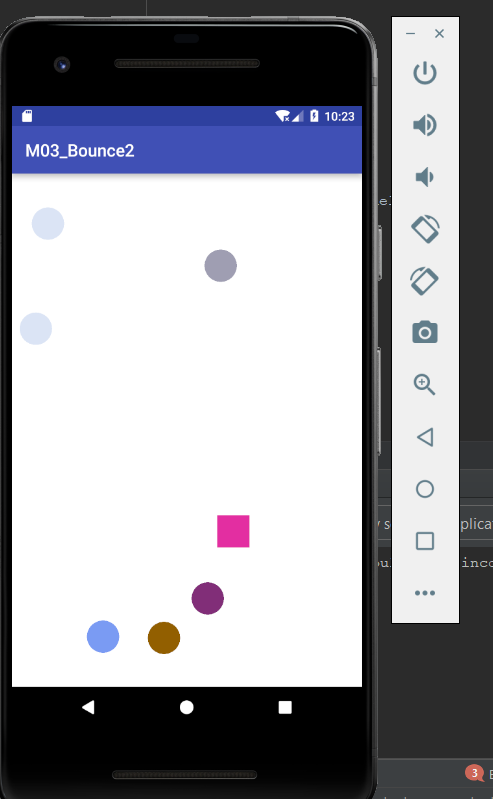




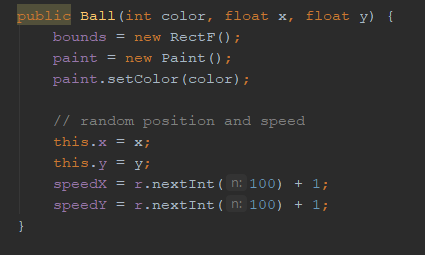
1. Change the color of newly made balls to a new random color for each new ball. What did you do?







1. Make the newly made balls go super-fast and super-slow.  What did you do?



Above is new constructor that takes r (previously defined random() ) and sets speed x & y to random speeds

1. Try different approaches for invalidate() (different code locations, methods, ...):
   1. Does the program still work each time?

From where I tried, it needs to be precisely where it was located in order to work.

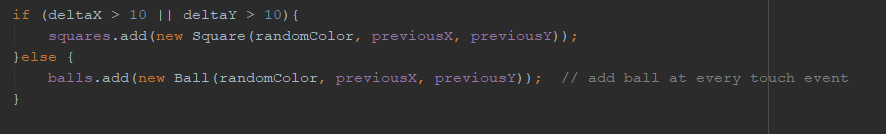
* 1. What does invalidate() do?  What happens when it isn't called at all?

I did a bit of reading about this, and from what I can tell it invalidates the previous draw on frame, and requires the program to redraw. When I removed it, it only displayed one frame and then stopped

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

onDraw is called every time invalidate is called. I did a small amount of reading about androids library, and from what I understand invalidate is what refreshes ondraw.

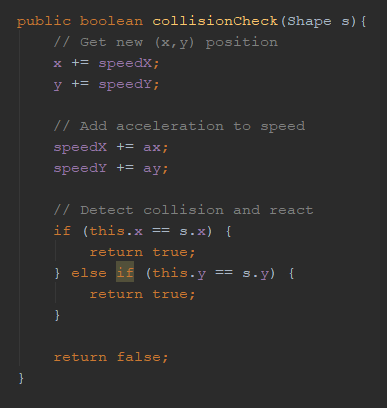
1. Add another shape class...fast swipes makes the new shape, slow swipes make circles.



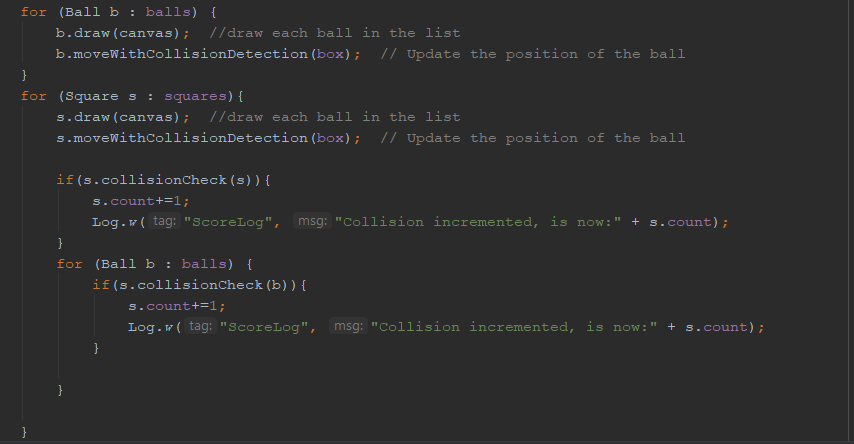
Above code snippet is from onTouchEvent, which takes deltaX and y as the speed of a touch event. If either X or Y surpasses ten, a square is made, otherwise a ball is made.

1. Add a rectangle shape...any time a shape collides with that rectangle you increment a score count (show score on logcat).

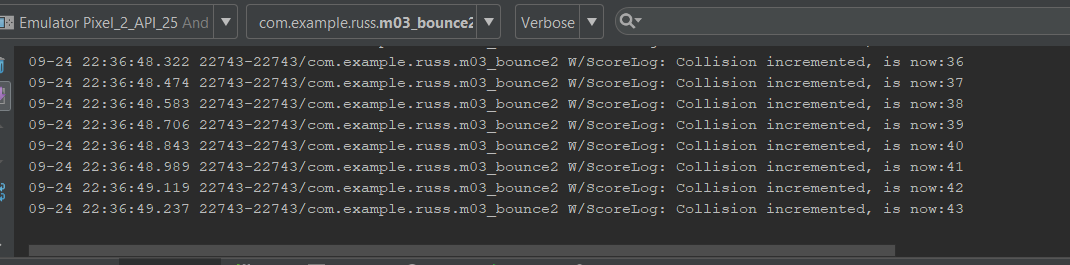
My square shape is essentially the same as the ball class, just with drawrect. I added this method to check for collisions:



I also created a Shape parent class for ball and square, so that it is easy to compare both in collision check.



Above is my complex bit of code. This is in onDraw() in BouncingBallView.java. For each square, it checks if it collides with other squares in the array, then checks if it collides with any balls in the ball array. It then increments the square classes count, and logs it. Below is catlog:



1. Think of another change (...and do that change) yourself, ...What did you do?

I changed the amount of balls possible on screen at once. It was quite the fun sight to see.

