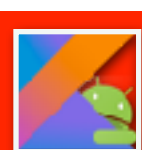




# Homework

@somkiat



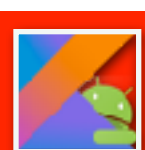
# Topics

แสดง Dot มากกว่า 1 Dot

ทำการลบ Dot ทั้งหมด

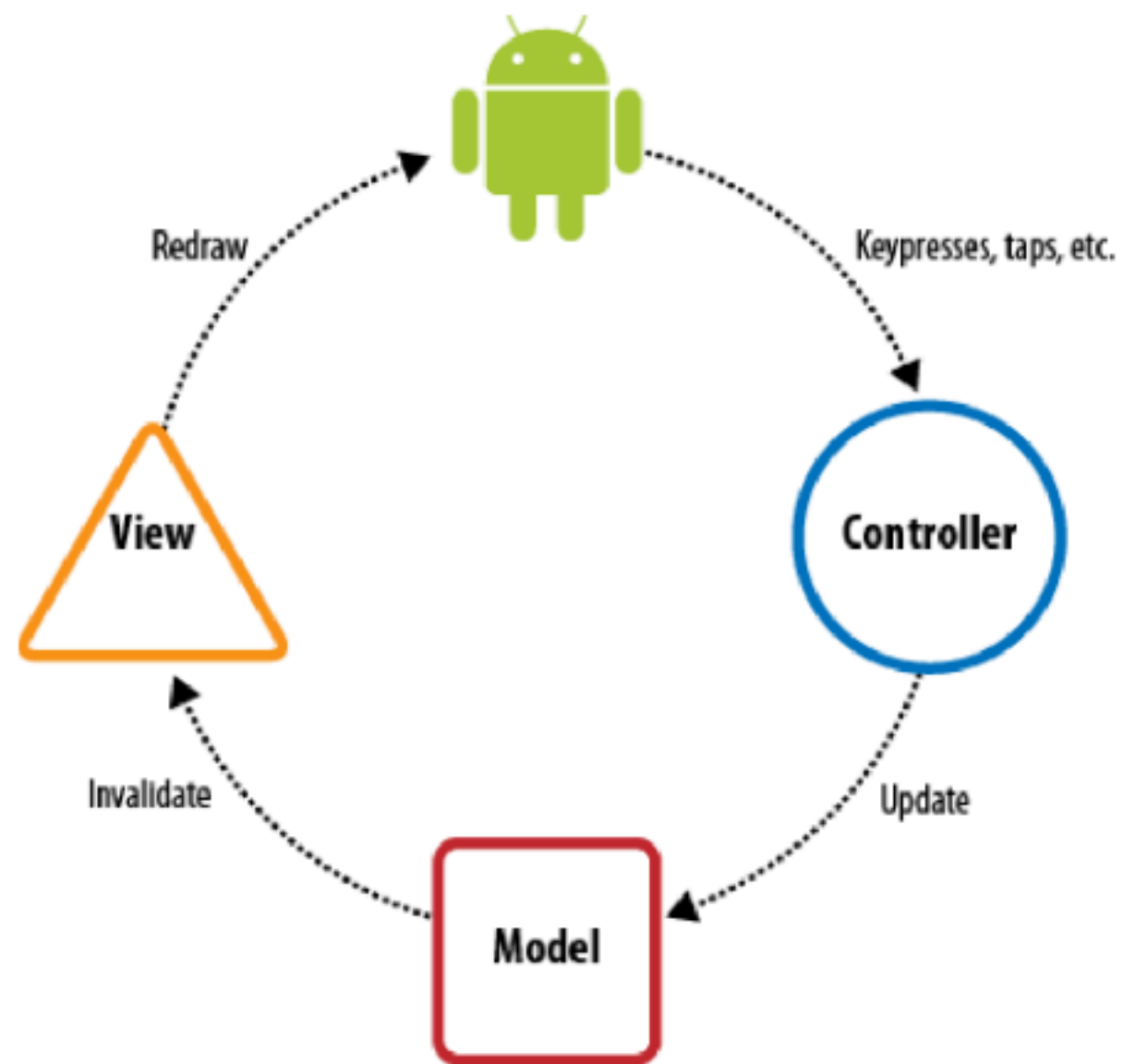
ทำการสร้าง Dot จากการ press/click

ทำการลบ Dot จากการ press/click/long click

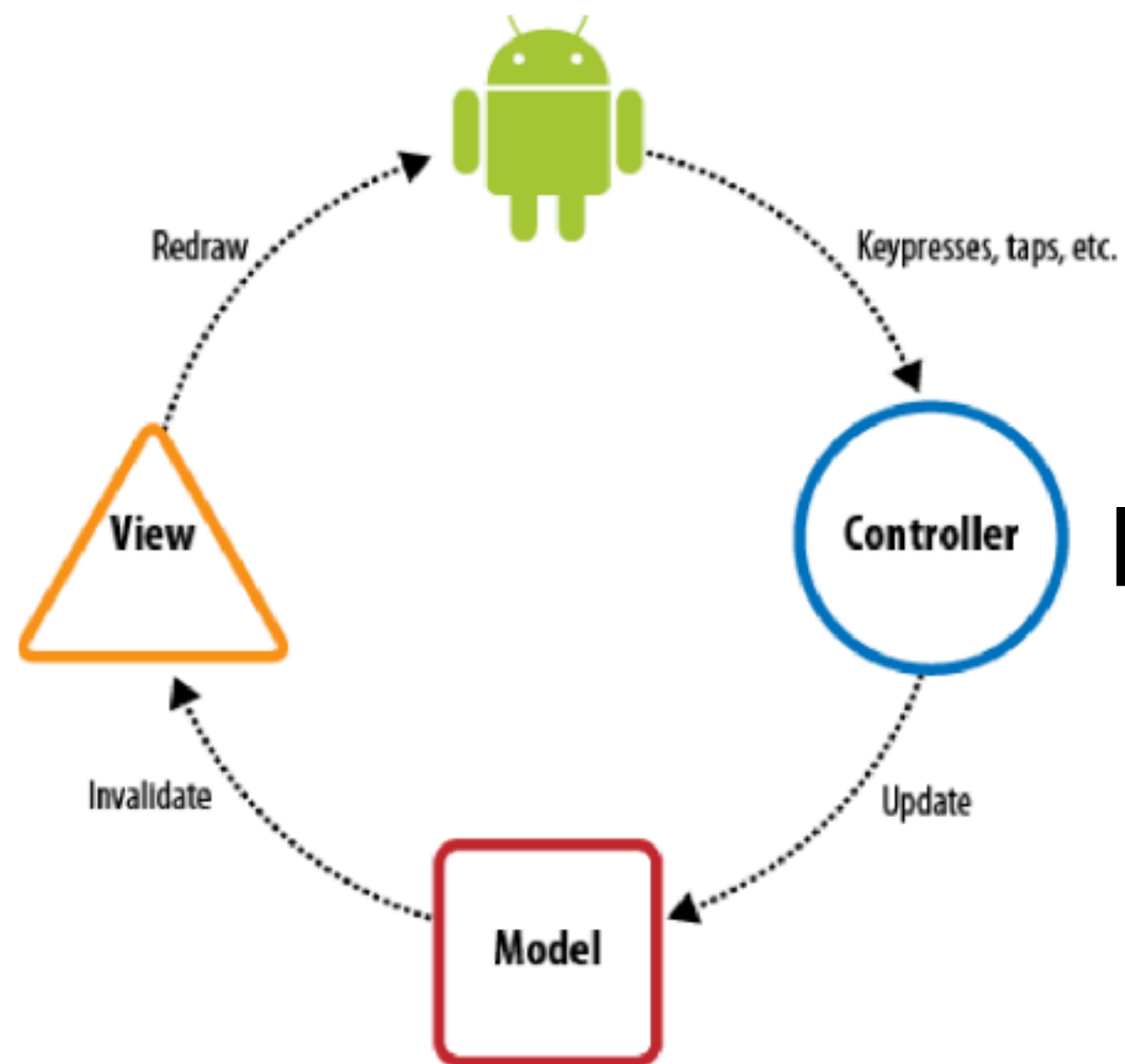


# 1. แสดง Dot มากกว่า 1 Dot



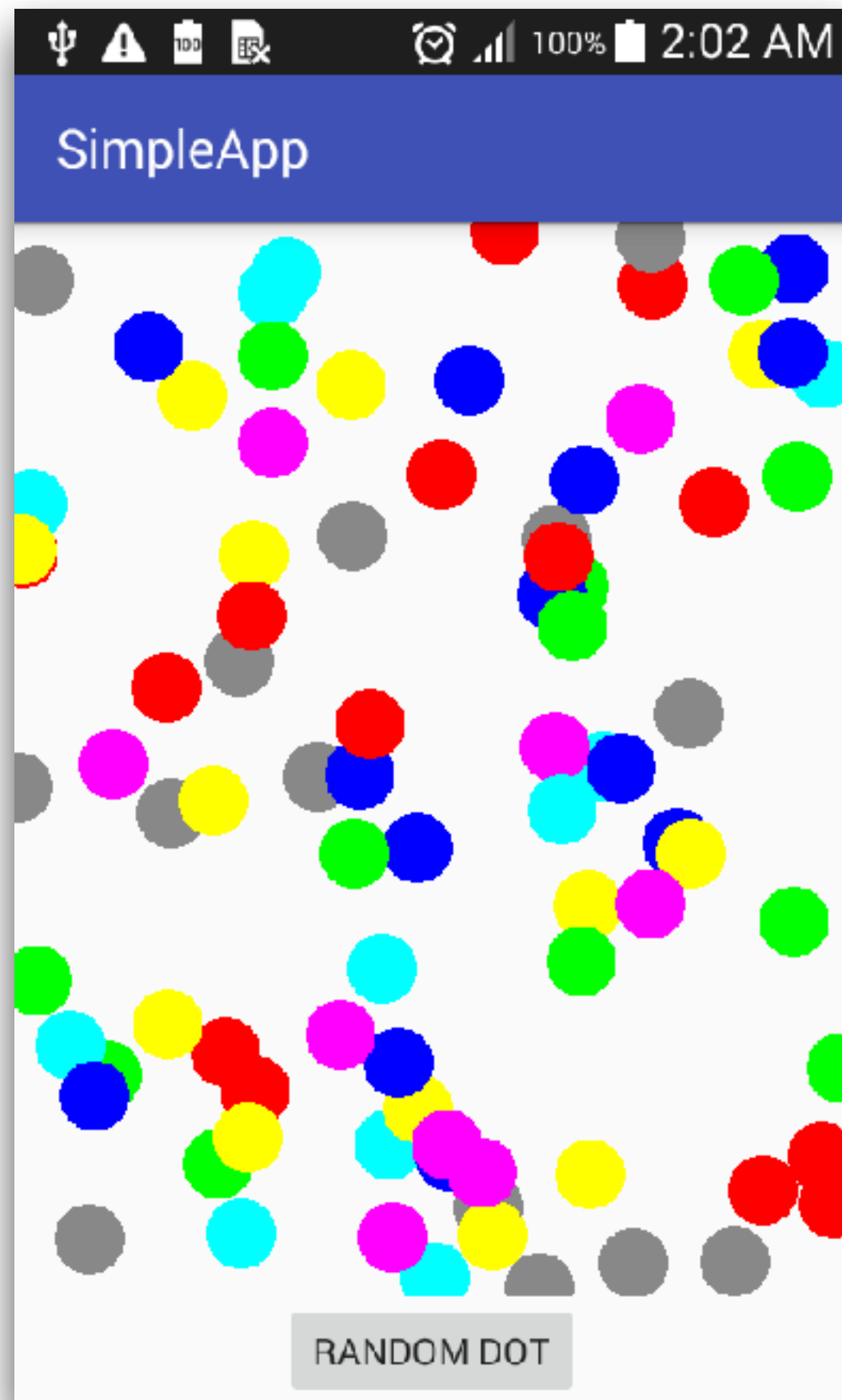


# DotView



# MainActivity

# Dot



# Model ?



# Dots

สำหรับเก็บ Dot ทั้งหมดที่สร้างขึ้นมา  
โครงสร้างข้อมูลเป็น List



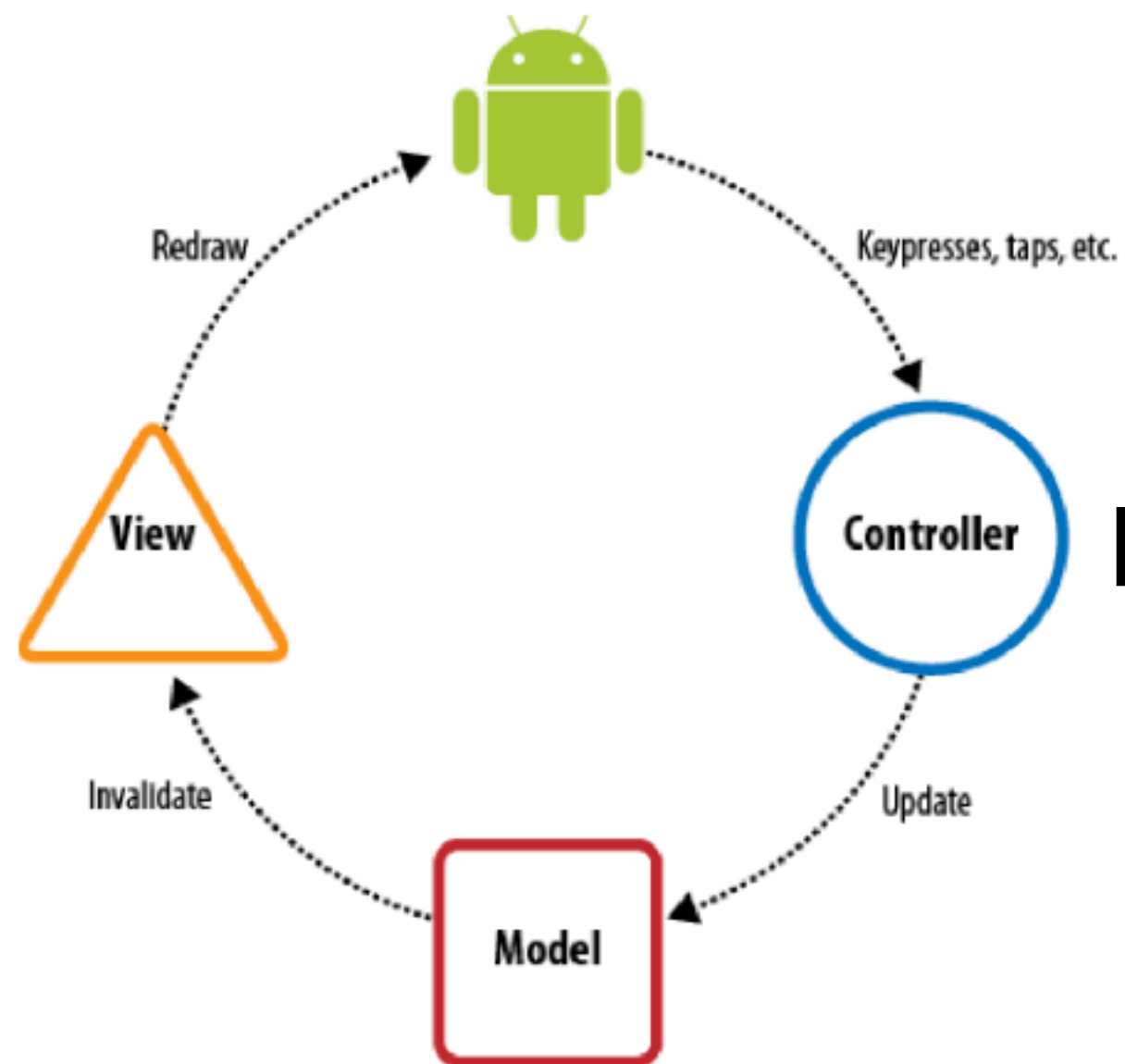


# Dots.java

```
public class Dots {  
    private List<Dot> allDot = new ArrayList<>();  
    public List<Dot> getAllDot() { return allDot; }  
    public void addDot(Dot dot) {  
        this.allDot.add(dot);  
    }  
}
```



# DotView



# MainActivity

## Dots

# ทำการวาด Dot ทั้งหมดเมื่อใด ?



# Dot

จะทำการวาดเมื่อ Dots เปลี่ยนเท่านั้น  
ลบ Listener หรือ Interface ออกไป

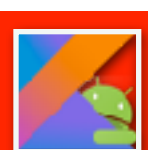


# Dot.java

```
public class Dot {  
    private int centerX;  
    private int centerY;  
    private int radius;  
  
    public Dot(int centerX, int centerY, int radius) {  
        this.centerX = centerX;  
        this.centerY = centerY;  
        this.radius = radius;  
    }  
}
```



# ทำการสร้าง Listener/Interface ใน Dots



# Dots.java

```
public class Dots {  
  
    public interface OnDotsChangeListener {  
        void onDotsChanged(Dots dots);  
    }  
  
    private OnDotsChangeListener listener;  
  
    public void setListener(OnDotsChangeListener listener) {  
        this.listener = listener;  
    }  
}
```



# Dots.java

```
private List<Dot> allDot = new ArrayList<>();

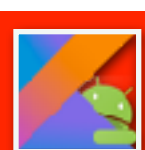
public List<Dot> getAllDot() { return allDot; }

public void addDot(Dot dot) {
    this.allDot.add(dot);
    this.listener.onDotsChanged(this);
}
```





# ทำการแก้ไข MainActivity



# MainActivity.java

เปลี่ยนการ implement ด้วย interface จาก Dots  
ทำการเพิ่ม Dot เมื่อกดปุ่ม Random ไปยัง Dots  
ทำवाद Dot ทั้งหมดไปที่ DotView



# MainActivity.java

เปลี่ยนการ implement ด้วย interface จาก Dots

```
public class MainActivity extends AppCompatActivity  
implements Dots.OnDotsChangeListener{  
  
    private DotView dotView;  
    private Dots dots;
```



# MainActivity.java

ทำการเพิ่ม Dot เมื่อกดปุ่ม Random ไปยัง Dots

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);
```

```
    dotView = (DotView) findViewById(R.id.dotView);  
    dots = new Dots();  
    dots.setListener(this);  
}
```

```
public void onRandomDot(View view) {  
    Random random = new Random();  
    int centerX = random.nextInt(dotView.getWidth());  
    int centerY = random.nextInt(dotView.getHeight());  
    Dot newDot = new Dot(centerX, centerY, 30);  
    dots.addDot(newDot);  
}
```

# MainActivity.java

ทำवाद Dot ทั้งหมดไปที่ DotView

```
@Override  
public void onDotsChanged(Dots dots) {  
    dotView.setDots(dots);  
    dotView.invalidate();  
}
```



# ต้องทำการแก้ไข DotView ด้วย



# DotView.java

เปลี่ยนจาก Dot มาเป็น Dots

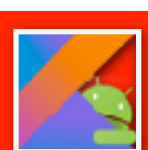
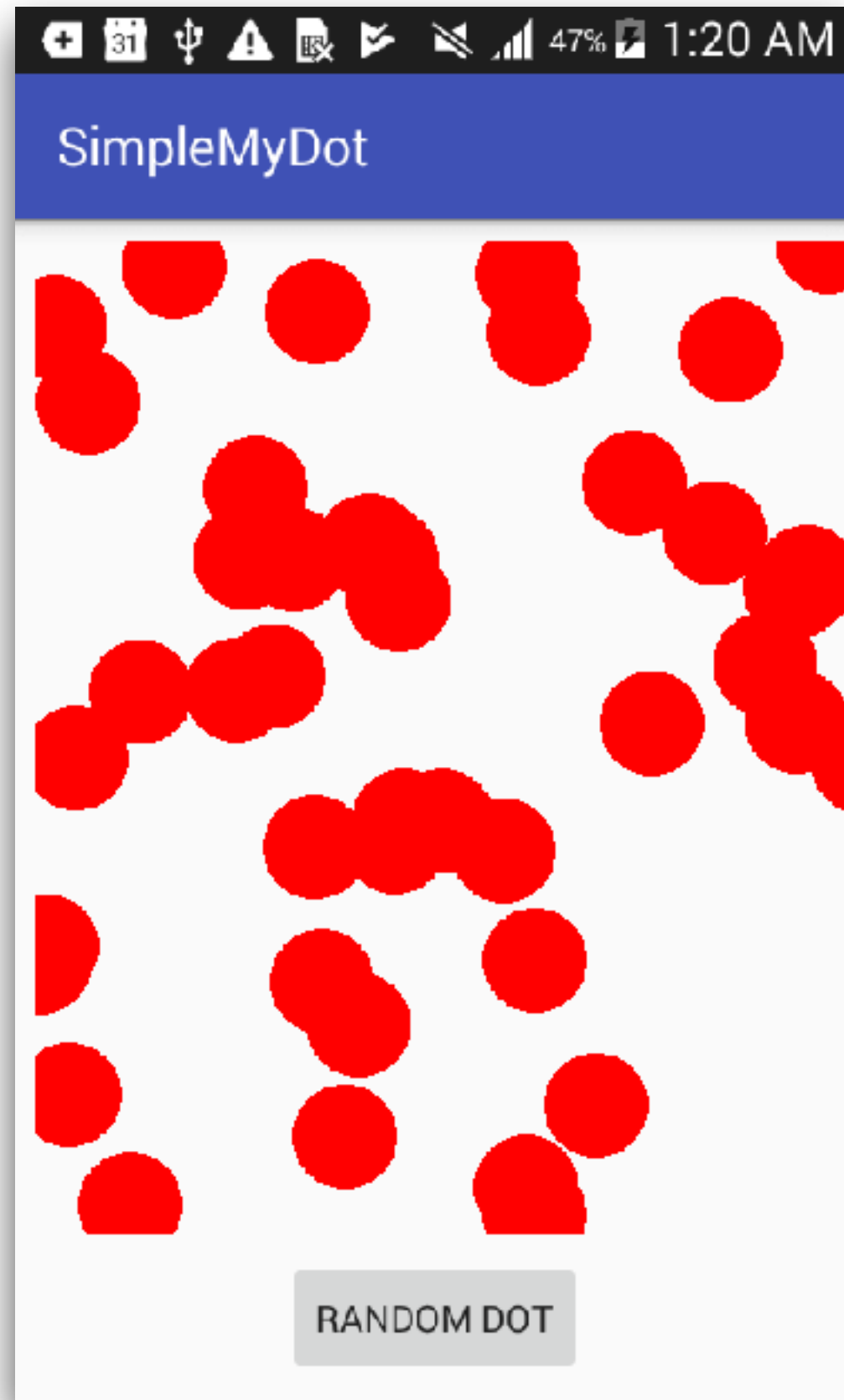
```
public class DotView extends View {  
  
    private Paint paint;  
    private Dots allDot;  
  
    @Override  
    protected void onDraw(Canvas canvas) {  
        super.onDraw(canvas);  
        paint.setColor(Color.RED);  
        if(this.allDot != null) {  
            for (Dot dot: allDot.getAllDot()) {  
                canvas.drawCircle(  
                    dot.getCenterX(),  
                    dot.getCenterY(), 30, paint);  
            }  
        }  
    }  
}
```



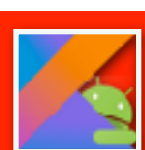
# ผลการดำเนินงาน



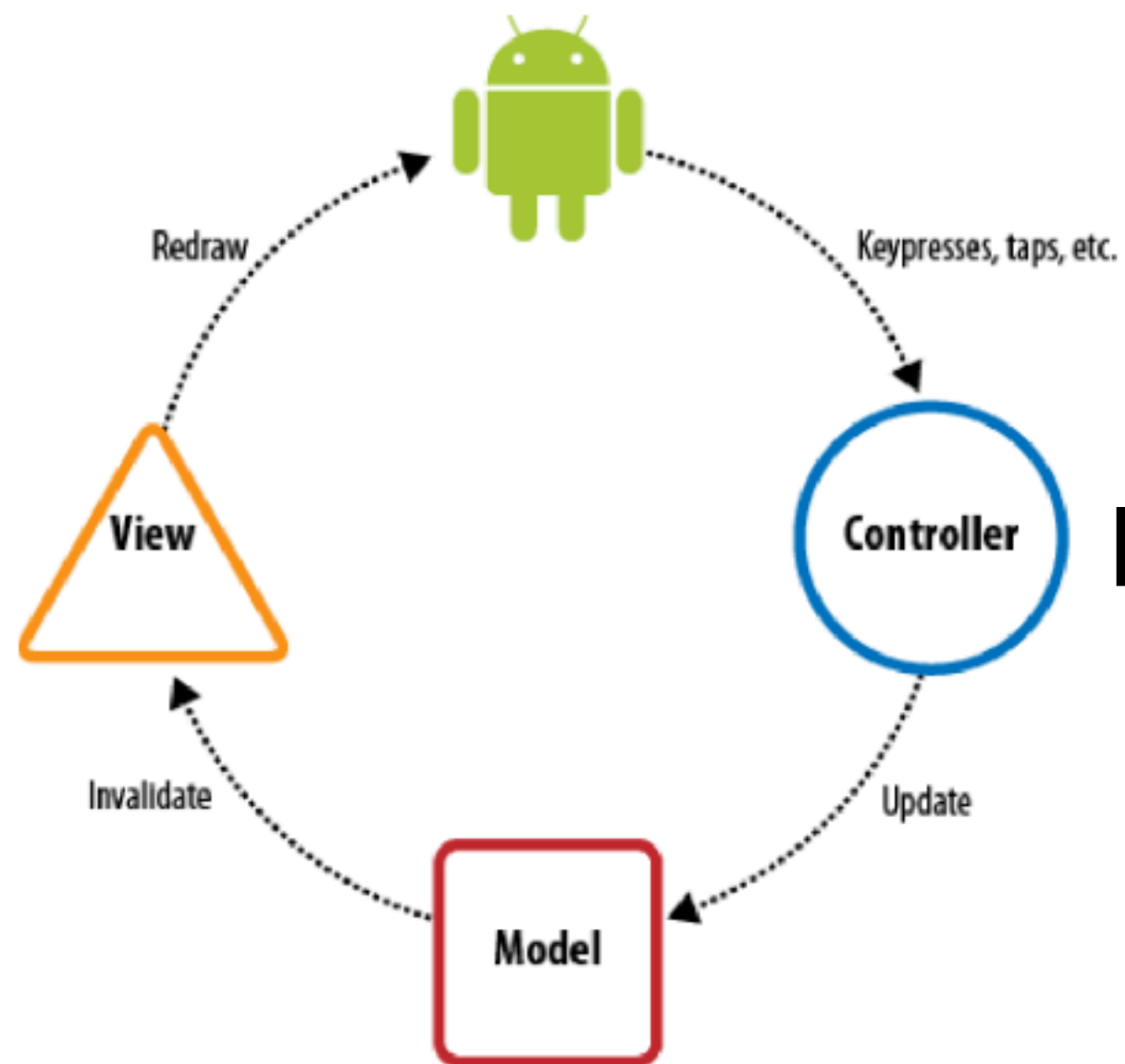




# ถ้าต้องการให้ Random ลี ?



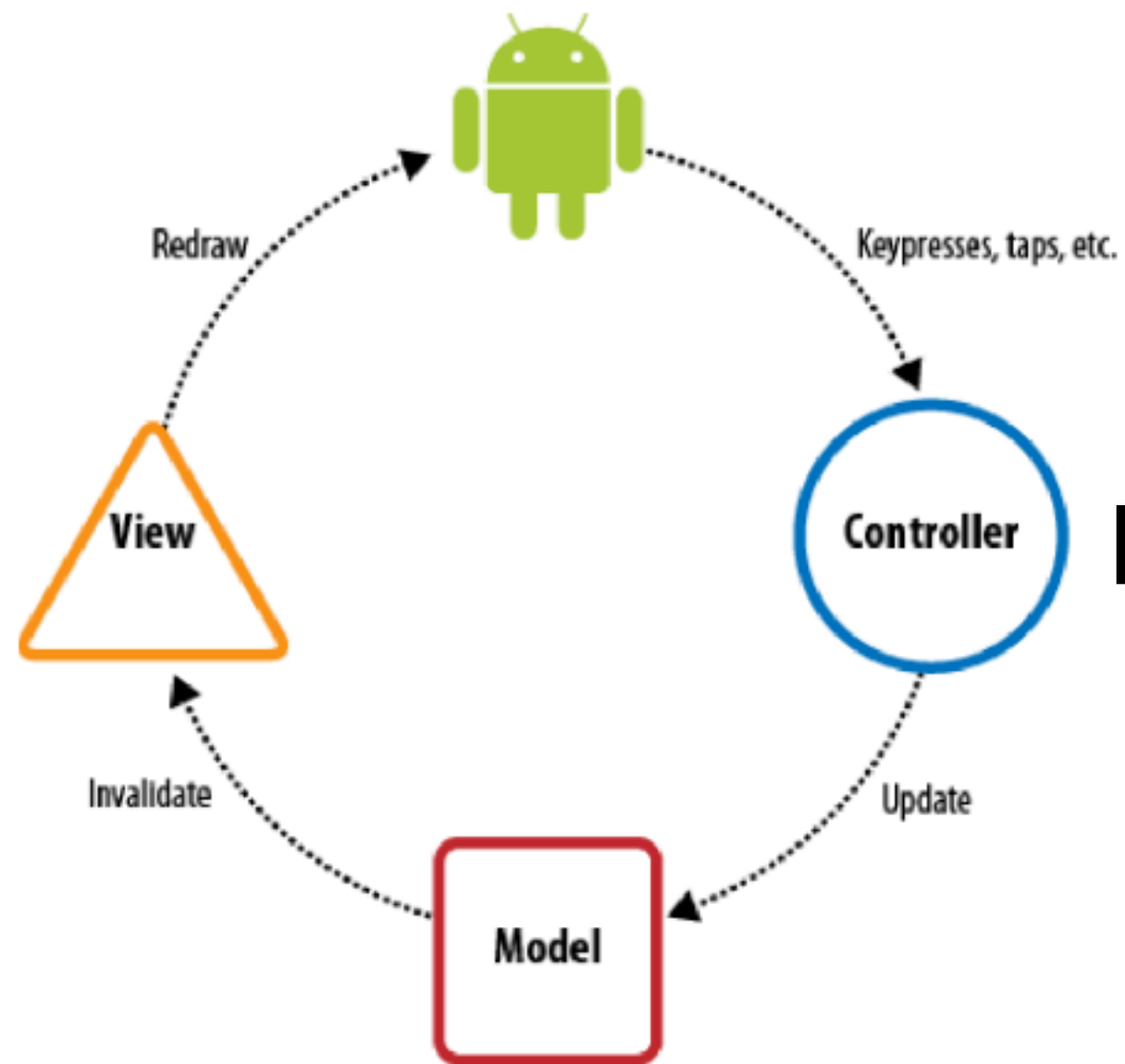
# DotView



# MainActivity

## Dots

# DotView



# MainActivity

## Dots Colors

# Colors.java

ทำการสร้าง model สำหรับจัดการเรื่องของสีของ Dot

```
public class Colors {  
    private List<Integer> colorList = new ArrayList<>();  
  
    public Colors() {  
        colorList.add(Color.RED);  
        colorList.add(Color.GREEN);  
        colorList.add(Color.BLUE);  
        colorList.add(Color.CYAN);  
        colorList.add(Color.MAGENTA);  
        colorList.add(Color.YELLOW);  
        colorList.add(Color.GRAY);  
    }  
    public int getColor() {  
        return colorList.get(  
            new Random().nextInt(colorList.size()));  
    }  
}
```



# Dot.java

เพิ่ม color เข้ามายัง Dot

```
public class Dot {  
    private int centerX;  
    private int centerY;  
    private int radius;  
    private int color;  
  
    public Dot(int centerX, int centerY, int radius) {  
        this(centerX, centerY, radius, Color.RED);  
    }  
  
    public Dot(int centerX, int centerY, int radius, int color) {  
        this.centerX = centerX;  
        this.centerY = centerY;  
        this.radius = radius;  
        this.color = color;  
    }  
}
```

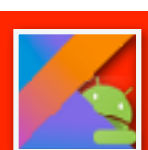
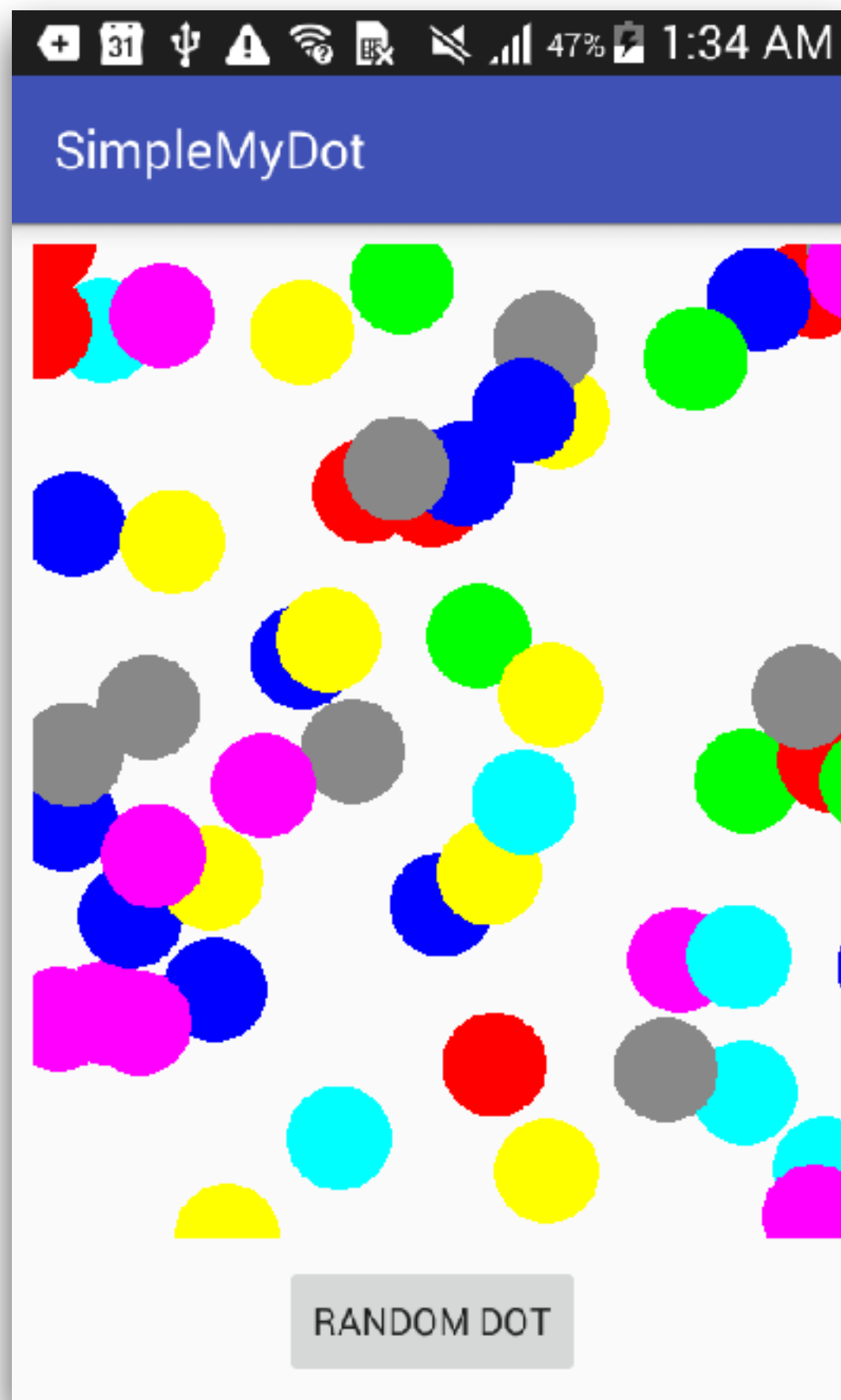


# DotView.java

ทำการวาด Dot ด้วยสีที่ Random มาได้

```
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    if(this.allDot != null) {
        for (Dot dot: allDot.getAllDot()) {
            paint.setColor(dot.getColor());
            canvas.drawCircle(
                dot.getCenterX(),
                dot.getCenterY(), 30, paint);
        }
    }
}
```



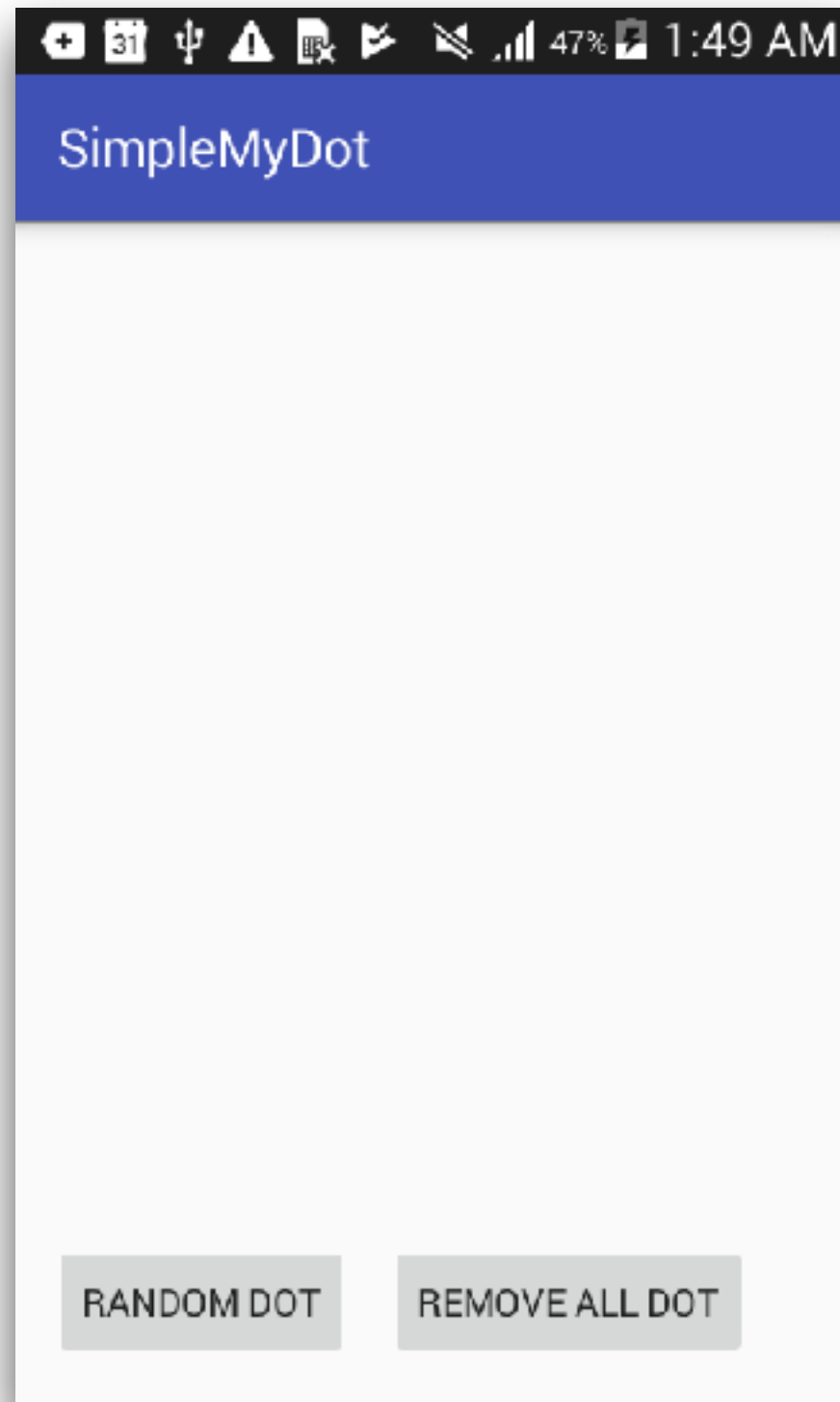




## 2. ทำการลบ Dot ทั้งหมด



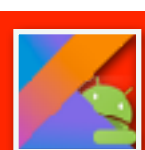
# เพิ่มปุ่ม Remove All



# เพิ่มปุ่ม Remove All

ดัดการกดปุ่มที่ MainActivity

```
public void onRemoveAll(View view) {  
    dots.clearAll();  
}
```

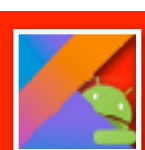
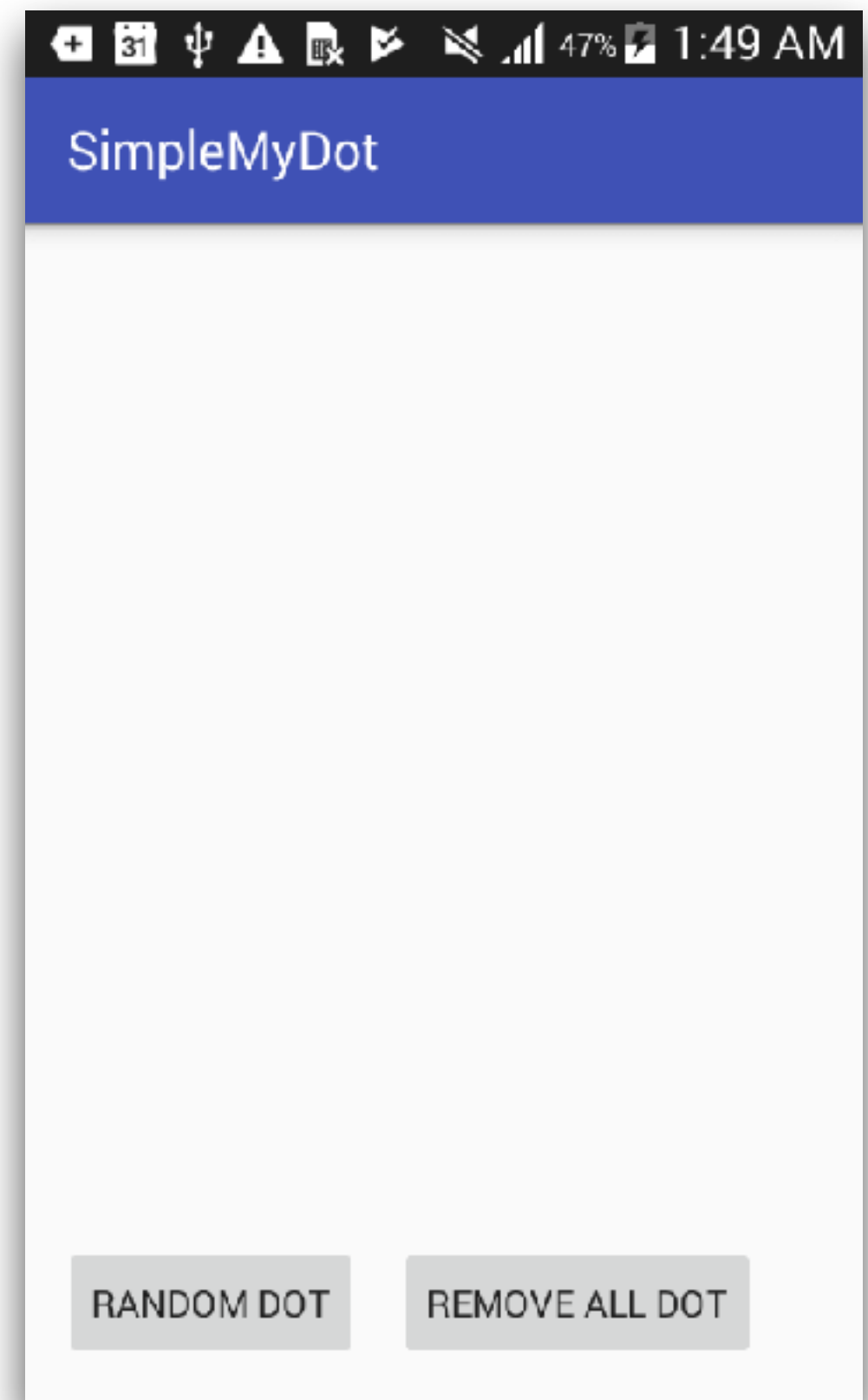
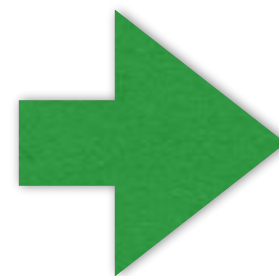


# แก้ไข Dots.java

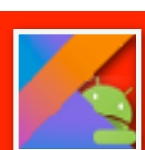
เพิ่ม method clearAll()

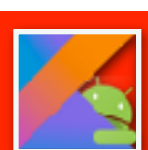
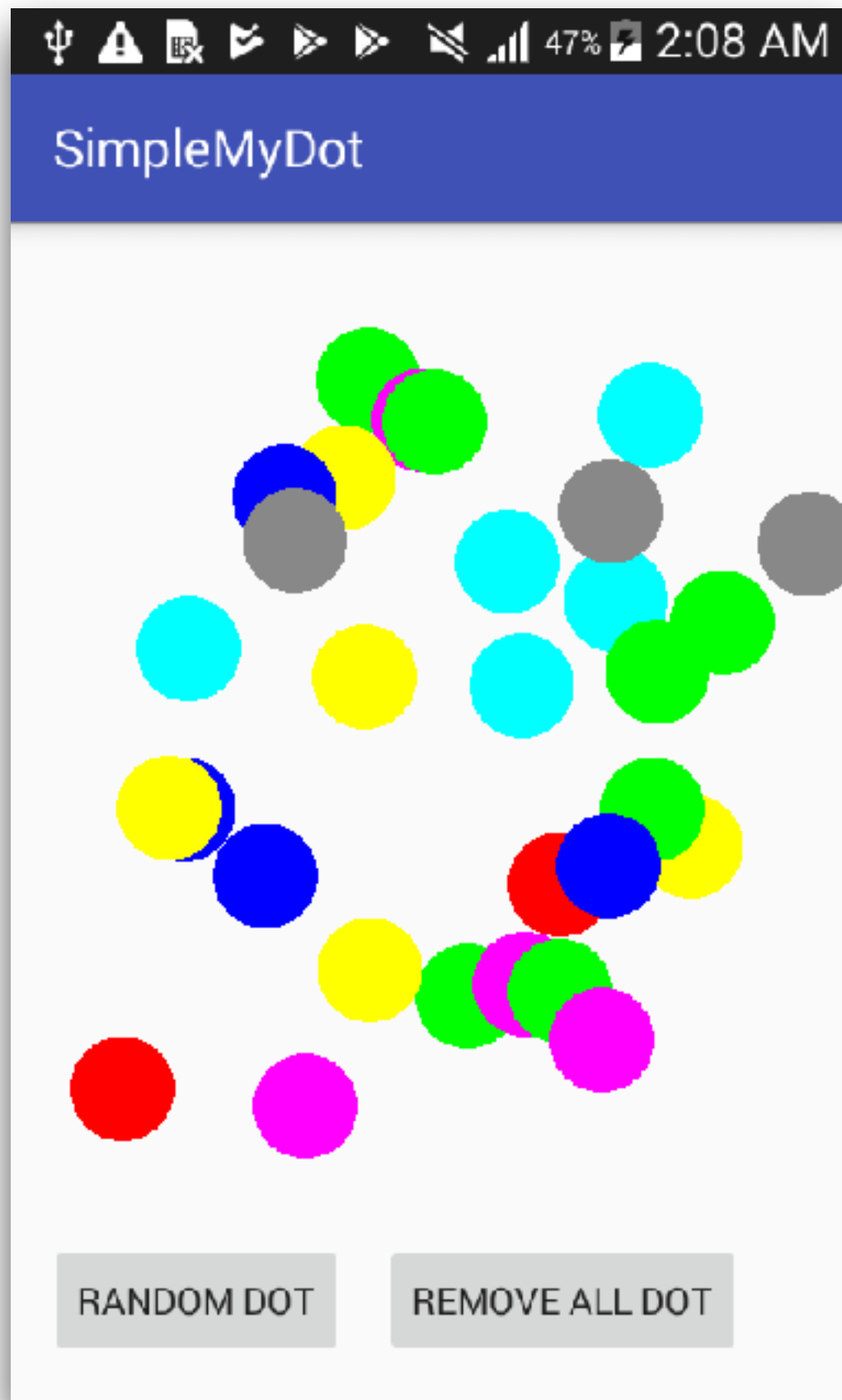
```
public void clearAll() {  
    allDot.clear();  
    this.listener.onDotsChanged(this);  
}
```





### 3. ทำการสร้าง Dot จากการ press/click





# DotView.java

ทำการดักการ press/click

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            this.onDotViewPressListener
                .onDotViewPressed(
                    (int)event.getX(),
                    (int)event.getY());
            return true;
        }
    return false;
}
```



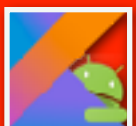


# DotView.java

สร้าง interface/listener ใหม่

```
public interface OnDotViewPressListener{  
    void onDotViewPressed(int x, int y);  
}
```

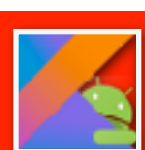
```
private OnDotViewPressListener onDotViewPressListener;  
public void setOnDotViewPressListener(  
    OnDotViewPressListener onDotViewPressListener) {  
    this.onDotViewPressListener = onDotViewPressListener;  
}
```



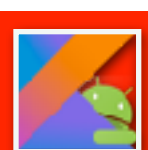
# MainActivity.java

ทำการเพิ่ม Dot เข้าไปยัง Dots

```
@Override  
public void onDotViewPressed(int x, int y) {  
    Dot newDot = new Dot(x, y, 30, new Colors().getColor());  
    dots.addDot(newDot);  
}
```



# 4. ทำการลบ Dot จากการ press/click



# แนวคิด

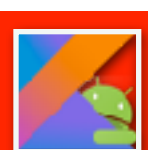
ถ้าทำการ press ตรงช่องว่างจะสร้าง Dot ใหม่  
ถ้าทำการ press ตรง Dot ที่ีมีจะทำการลบ



# สิ่งแรกที่ต้องทำคืออะไร ?



**ทำการตรวจสอบว่าจุดที่ press มี Dot ?**

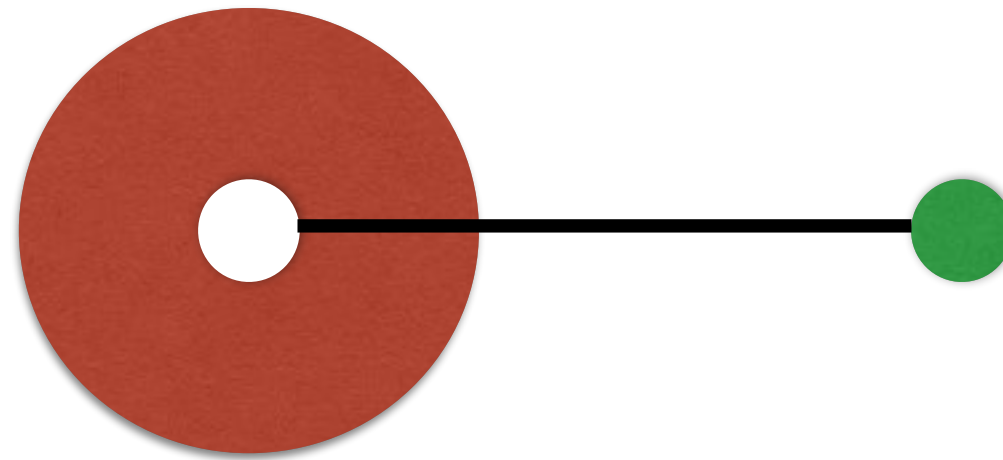


# Dots.java

ทำการเพิ่ม method findDot()

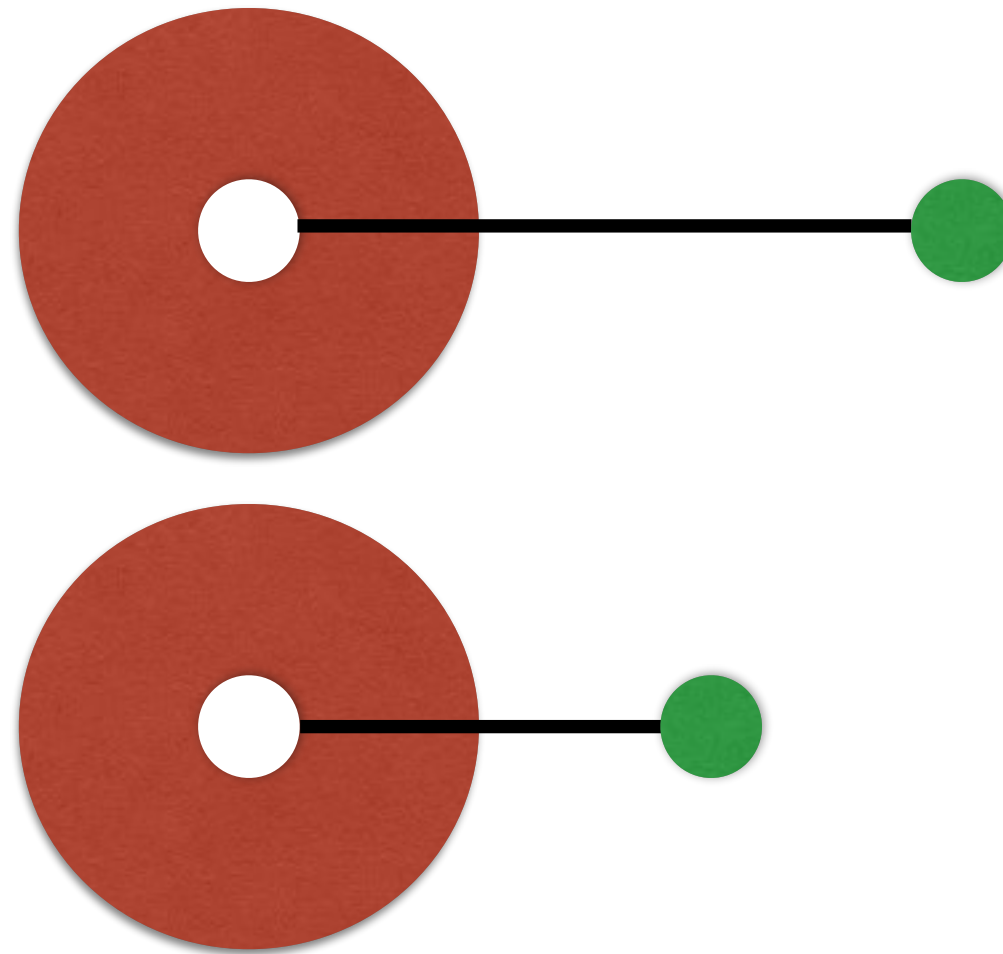
```
public int findDot(int x, int y) {  
    for (int i = 0; i < allDot.size(); i++) {  
        int centerX = allDot.get(i).getCenterX();  
        int centerY = allDot.get(i).getCenterY();  
        double distance = Math.sqrt(Math.pow(centerX - x, 2)) +  
            Math.sqrt(Math.pow(centerY - y, 2));  
        if (distance <= 30) {  
            return i;  
        }  
    }  
    return -1;  
}
```

??

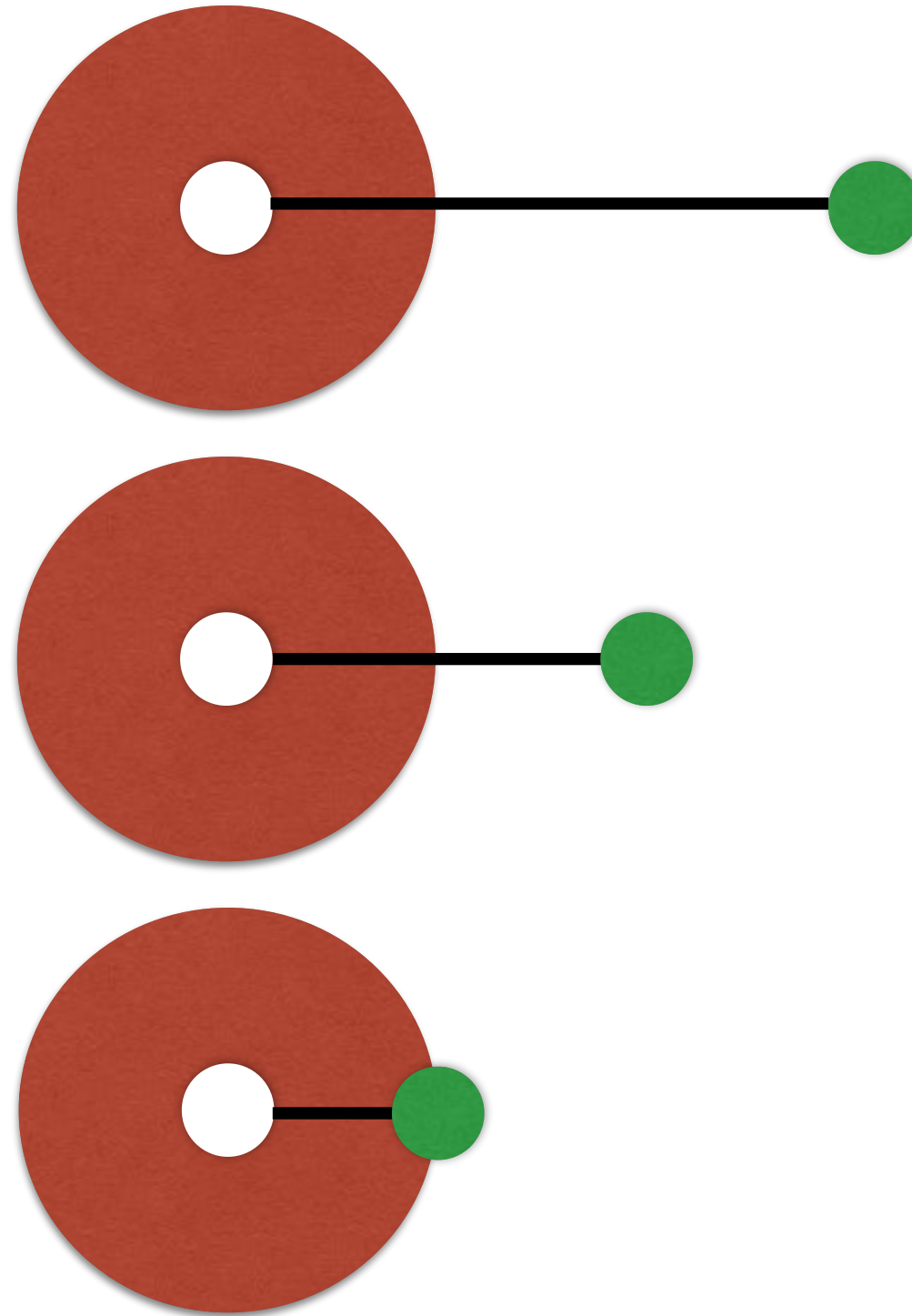




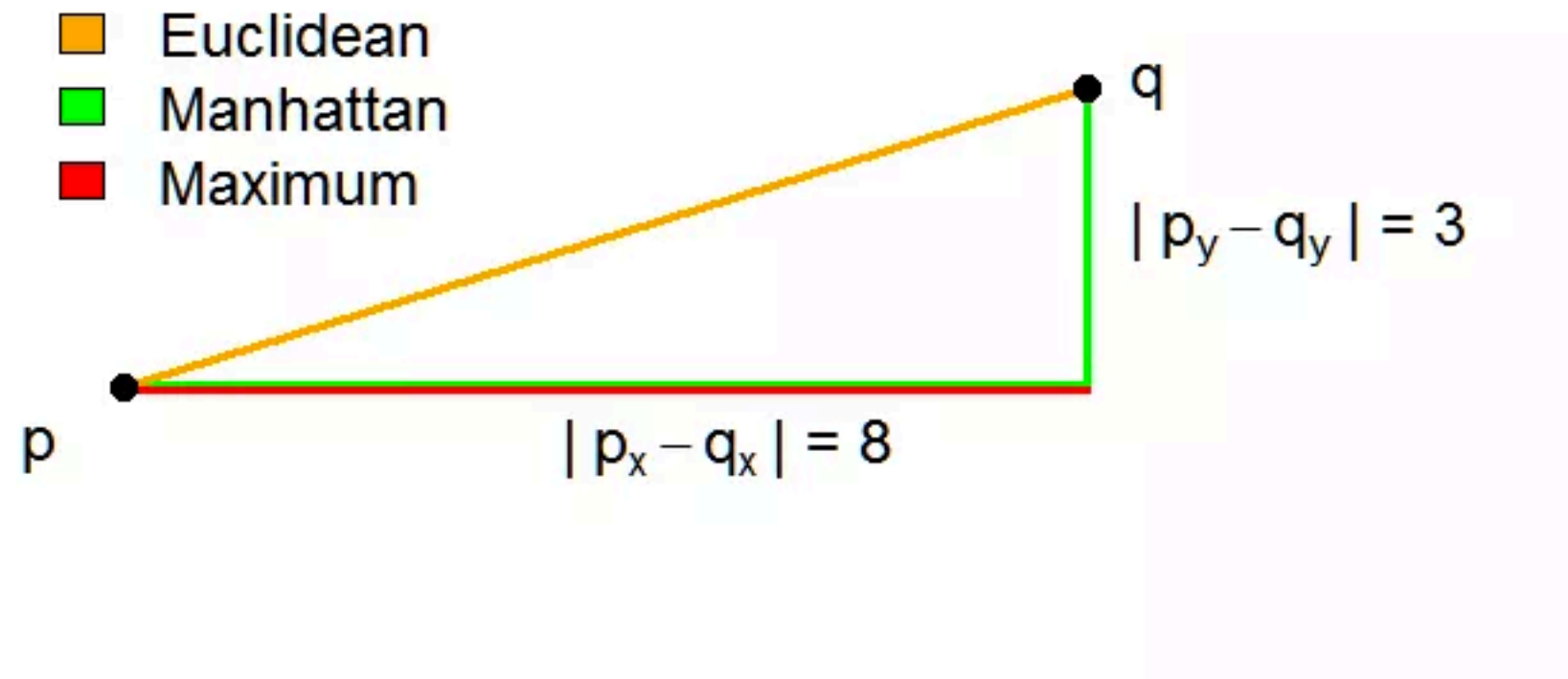
??



??

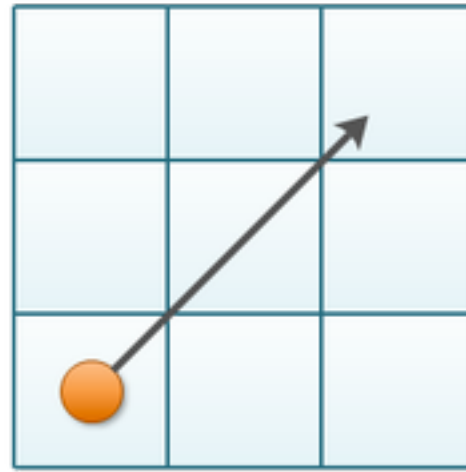


# Euclidean distance



# Euclidean distance

**Euclidean Distance**



$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

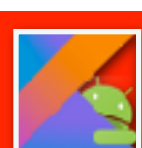


# MainActivity.java

ทำการเรียกใช้งาน findDot()

@Override

```
public void onDotViewPressed(int x, int y) {  
    int dotPosition = dots.findDot(x, y);  
    if(dotPosition == -1) {  
        Dot newDot = new Dot(x, y, 30, new Colors().getColor());  
        dots.addDot(newDot);  
    } else {  
        dots.removeBy(dotPosition);  
    }  
}
```



# Dots.java

ทำการลบ dot ออกจาก List

```
public void removeBy(int position) {  
    allDot.remove(position);  
    this.listener.onDotsChanged(this);  
}
```



# Let's practice !!

