



ANDROID PROGRAMMING

# 그래픽 출력과 UI 이벤트 처리

1871384 유수미

# INDEX

1/ 문제 설명

---

2/ 해결 방법

---

3/ 실행 결과

---

4/ 결론

---

# 1 문제 설명



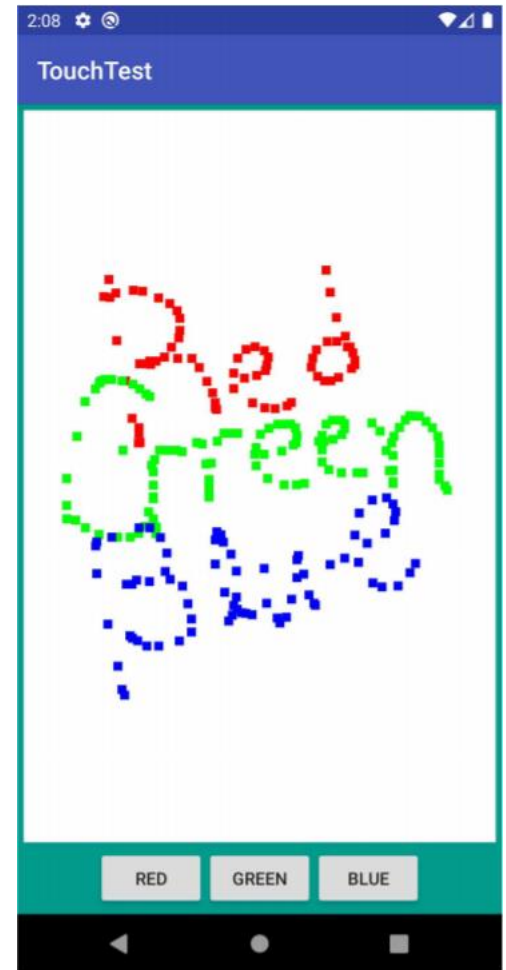
## 문제 설명

**문제상황:** 안드로이드 **벡터 그래픽 편집기** 개발을 계획.  
색깔을 변경하고 터치하여 그릴 수 있는지 확인을 하려고 함.

**문제 1.** 색 선택이 가능하도록 **레이아웃**을 수정하기

**문제 2.** 선택한 색깔로 그려지도록 코드와 **mPoints**의  
**자료구조** 수정하기

**문제 3.** 버튼보다 **라디오 버튼**이 더 좋은가?



# 2 해결 방법



# 해결 방법

## 문제 1. 색 선택이 가능하도록 레이아웃을 수정하기

### RadioGroup

```
android:id="@+id/radioGroup2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:orientation="horizontal"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent">
```

### <RadioButton

```
android:id="@+id/button_red"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:onClick="mOnClick"  
android:text="@string/red"  
android:textSize="30sp" />
```

### <RadioButton

```
android:id="@+id/button_green"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:onClick="mOnClick"  
android:text="@string/green"  
android:textSize="30sp" />
```

### <RadioButton

```
android:id="@+id/button_blue"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:onClick="mOnClick"  
android:text="@string/blue"  
android:textSize="30sp" />
```

```
</RadioGroup>
```

## 파일 구성

- activity\_main.xml
- MainActivity.java
- MyView.java

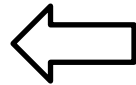
라디오 그룹으로 라디오 버튼들을 묶어서 하나의 항목만 선택할 수 있게 수정했다.



## 해결 방법

문제2. 선택한 색깔로 그려지도록 **mPoints**의 자료구조 수정하기

```
class MyPoint {  
    float x,y;  
    int color;  
    MyPoint(float x, float y, int c) {  
        this.x = x;  
        this.y = y;  
        this.color = c;  
    }  
}
```



(x좌표, y좌표, 색)을 한쌍으로 묶어서 저장하면 자료를 쉽게 관리할 수 있고, 보다 편하게 배열에 넣고 뺄 수 있다.

### 파일 구성

- activity\_main.xml
- MainActivity.java
- MyView.java

### <MyPoint 사용 예시>

```
private ArrayList<MyPoint> mPoints = new ArrayList<>();  
mPoints.add(new MyPoint(event.getX(), event.getY(), Color.RED));  
MyPoint myPoint = mPoints.get(0);
```



# 해결 방법

## 문제2. 선택한 색깔로 그려지도록 코드 수정하기

```
public class MainActivity extends AppCompatActivity {
    private ActivityMainBinding binding;
    private int color;
    public static Context mContext;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding=ActivityMainBinding.inflate(getLayoutInflater());
        View view=binding.getRoot();
        setContentView(view);

        mContext=this;
    }
    public void mOnClick(View v){
        if(binding.buttonRed.isChecked()){
            color= Color.RED;
        }
        else if(binding.buttonGreen.isChecked()){
            color=Color.GREEN;
        }
        else if(binding.buttonBlue.isChecked()){
            color=Color.BLUE;
        }
    }
    public int colorValue(){
        return color;
    }
}
```

### 파일 구성

- activity\_main.xml
- MainActivity.java
- MyView.java

선택된 색을 넣어주기 위한 변수

현재 context를 넣어주기 위한 변수





# 해결 방법

문제2. 선택한 색깔로 그려지도록 코드 수정하기

## 파일 구성

- activity\_main.xml
- MainActivity.java
- MyView.java

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    binding=ActivityMainBinding.inflate(getLayoutInflater());  
    View view=binding.getRoot();  
    setContentView(view);  
  
    mContext=this;  
}
```

MyView 클래스에서 해당 클래스의 함수를 사용할 수 있도록 MainActivity 객체를 mContext에 넣어주었다.



# 해결 방법

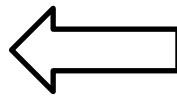
문제2. 선택한 색깔로 그려지도록 코드 수정하기

## 파일 구성

- activity\_main.xml
- MainActivity.java
- MyView.java

```
public void mOnClick(View v){  
    if(binding.buttonRed.isChecked()){ // RED 라디오 버튼이 체크 상태면  
        color= Color.RED;              // color 변수에 빨간색을 넣는다.  
    }  
    else if(binding.buttonGreen.isChecked()){ // Green 라디오 버튼이 체크 상태면  
        color=Color.GREEN;              // color 변수에 초록색을 넣는다.  
    }  
    else if(binding.buttonBlue.isChecked()){ // Blue 라디오 버튼이 체크 상태면  
        color=Color.BLUE;              // color 변수에 파란색을 넣는다.  
    }  
}
```

```
public int colorValue(){  
    return color;  
}
```



**MyView 클래스에서** 이 함수를 통해 현재 선택된 색깔을 받을 수 있다.



# 해결 방법

문제2. 선택한 색깔로 그려지도록 코드 수정하기

```
private int mCurColor;

private ArrayList<MyPoint> mPoints = new ArrayList<>();

public boolean onTouchEvent(MotionEvent event) {

    mCurColor=((MainActivity)MainActivity.mContext).colorValue();

    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
        case MotionEvent.ACTION_MOVE: mPoints.add(new MyPoint(event.getX(), event.getY(), mCurColor));
            invalidate();
            return true;
        case MotionEvent.ACTION_UP:
            return true;
    }
    return super.onTouchEvent(event);
}
```

## 파일 구성

- activity\_main.xml
- MainActivity.java
- MyView.java

**MainActivity** 클래스에 있는 함수 **colorValue()**를 호출하여 값(색깔)을 받아온 후 mCurColor에 색을 넣어주어 앞으로 그려질 색을 설정해주었다.



# 해결 방법

문제2. 선택한 색깔로 그려지도록 코드 수정하기

```
private int mCurColor;
private ArrayList<MyPoint> mPoints = new ArrayList<>();

public boolean onTouchEvent(MotionEvent event) {

    mCurColor=((MainActivity)MainActivity.mContext).colorValue();

    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
        case MotionEvent.ACTION_MOVE: mPoints.add(new MyPoint(event.getX(),
event.getY(), mCurColor));
        invalidate();
        return true;
        case MotionEvent.ACTION_UP:
        return true;
    }
    return super.onTouchEvent(event);
}
```

onDraw() 함수가 호출된다.

## 파일 구성

- activity\_main.xml
- MainActivity.java
- MyView.java

눌러지거나 움직일 경우 그때의 x, y좌표와  
선택된 색을 mPoints에 넣는다.



# 해결 방법

문제2. 선택한 색깔로 그려지도록 코드 수정하기

## 파일 구성

- activity\_main.xml
- MainActivity.java
- MyView.java

```
private static final int R = 10;
private Paint mPaint = new Paint();

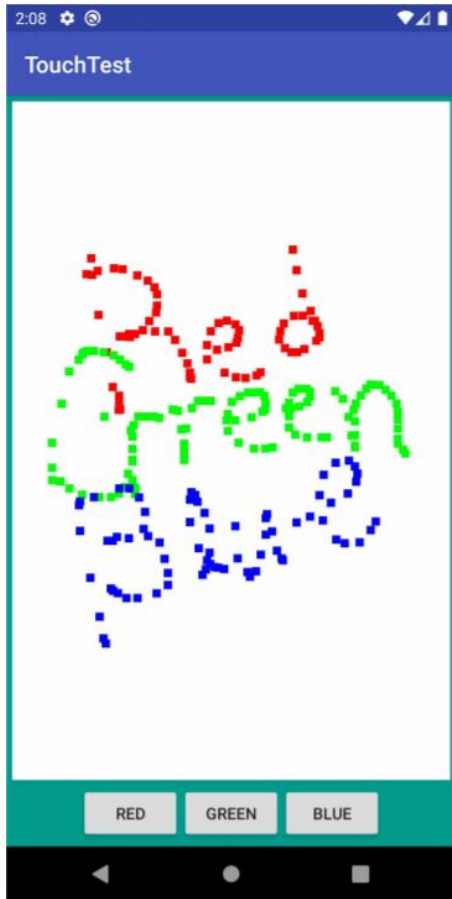
protected void onDraw(Canvas canvas) {
    for (int i = 0; i < mPoints.size(); i += 2) {
        MyPoint myPoint = mPoints.get(i);
        mPaint.setColor(myPoint.color); // 색 설정
        canvas.drawRect(myPoint.x - R, myPoint.y - R, myPoint.x
+ R, myPoint.y + R, mPaint); // x, y좌표를 이용해서 위치를 정해주고 사각형을 그린다.
    }
}
```

mPoints에 있는 값들을 전부 꺼내 저장되어있던  
x, y 좌표(위치) 정보나 색상의 정보를 이용해서  
사각형을 그려준다.



# 해결 방법

## 문제3. 버튼보다 라디오 버튼이 더 좋은가?



BUTTON



RADIO BUTTON

버튼은 현재 자신이 선택한 색이 무슨 색인지 알 수 없어, 이를 구분할 수 있는 라디오 버튼이 더 좋다.

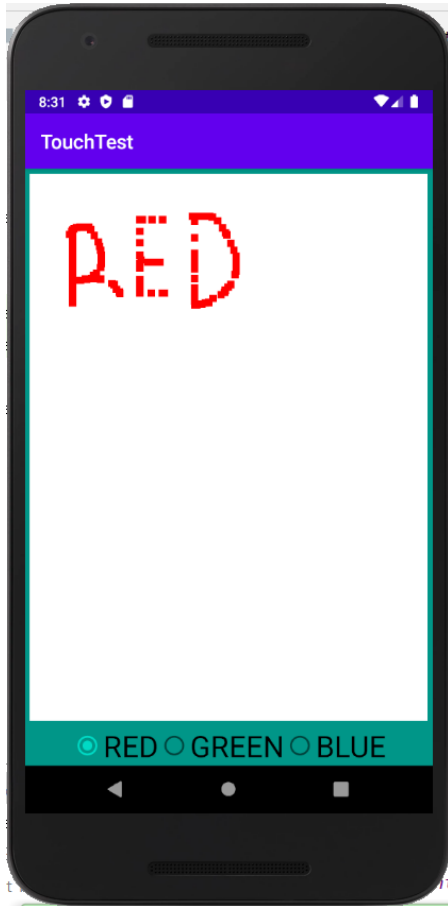
또, 이 경우에는 한번에 한가지 색만 선택하기 때문에 버튼이나 체크박스 보다는 라디오 버튼이 더 적합하다.



# 3 실행 결과



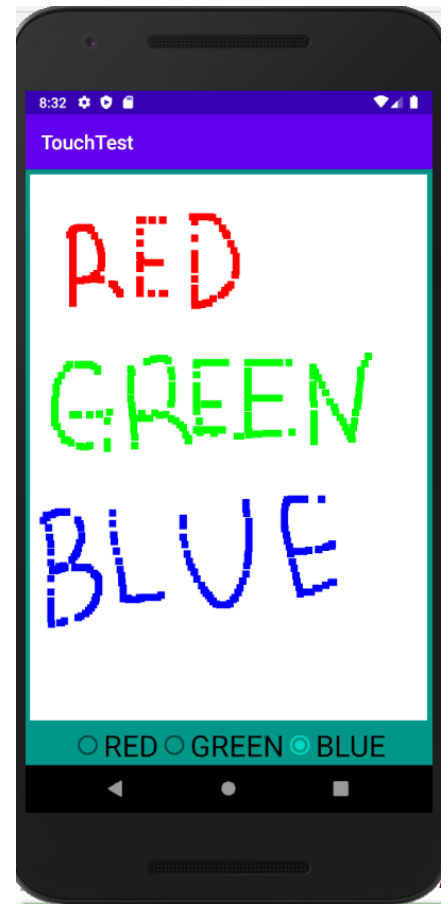
## 실행 결과



RED 선택



GREEN 선택



BLUE 선택



# 4 결론



# 결론

**문제1:** 라디오 그룹으로 라디오 버튼들을 묶어서 하나의 항목만 선택할 수 있는 레이아웃으로 만들었다.

**문제2:** (x좌표, y좌표, 색)을 한쌍으로 묶어주는 자료구조를 이용해 터치가 될때마다 값들을 ArrayList에 넣고, 그릴 때 꺼내쓴다.

**문제3:** 이 경우에는 선택된게 무엇인지 눈으로 확인할 수 있는 라디오 버튼이 구현에 더 적합한다.





**감사합니다**