



ANDROID PROGRAMMING

# 메뉴와 대화상자

1871384 유수미

# INDEX

1/ 문제 설명

---

2/ 해결 방법

---

3/ 실행 결과

---

4/ 결론

---

# 1 문제 설명



# 문제 설명

## [문제상황]

- 그려지는 도형의 색과 모양을 메뉴와 대화상자로 선택할 수 있도록 메뉴/대화상자를 추가
- 선택 가능한 색은 5가지 이상, 모양은 사각형, 원, 세모 등 3가지 이상이 가능하도록 함.

문제 1. 메뉴에 [색 변경], [붓 모양 변경]을 추가하고 해당 메뉴를 선택하면 색 변경 또는 붓 모양 변경 대화상자를 표시한다.

문제 2. 대화상자에서 선택한 색이나 붓 모양에 따라 그려지는 도형의 색과 모양을 변경한다. 이때 대화상자는 AlertDialog의 목록 선택 가능한 대화상자를 이용한다.



# 2 해결 방법



## 해결 방법

### 파일 구성

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

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/color"
    android:title="@string/color" /> //title을 [색 변경]으로 설정
  <item
    android:id="@+id/shape"
    android:title="@string/shape" /> //title을 [붓 모양 변경]으로 설정
</menu>
```

메뉴에 [색 변경], [붓 모양 변경] 항목 추가



## 해결 방법

### 파일 구성

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

```
public static Context mContext;
```

```
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 *mContext*에 넣어주었다.



## 해결 방법

### 파일 구성

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

```
public boolean onCreateOptionsMenu(Menu menu) {  
  
    MenuInflater inflater=getMenuInflater();  
  
    inflater.inflate(R.menu.menu_main,menu);  
  
    return super.onCreateOptionsMenu(menu);  
}
```

XML 메뉴 리소스(memu\_main.xml)를 읽어들여  
옵션 메뉴 초기화





# 해결 방법

## 파일 구성

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

```
private final String colorItems[]={"Yellow","Cyan","Magenta","Gray","Black"};  
private final String shapeItems[]={"Rect","Circle","Triangle"};
```

colorItems[]에 색 변경 대화상자의 목록을 넣어준다.  
shapeItems[]에 붓 모양 변경 대화상자의 목록을 넣어준다.

```
public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
  
    AlertDialog.Builder builder;  
    switch (item.getItemId()){  
        case R.id.color: // 메뉴의 [색 변경]을 선택했다면  
            builder=new AlertDialog.Builder(this);  
            builder.setTitle(R.string.color); // 제목을 “색 변경”으로 설정  
            builder.setItems(colorItems, new DialogInterface.OnClickListener() {
```

colorItems[]으로 대화상자의 목록을 표시해주고,  
목록을 선택하면 onClick()이 호출된다.



## 해결 방법

```
public void onClick(DialogInterface dialog, int which) {  
    switch (which){  
        case 0: // Yellow 선택되면  
            color=Color.YELLOW;        // color 변수에 Yellow 색을 넣는다.  
            break;  
        case 1: // Cyan 선택되면  
            color=Color.CYAN;          // color 변수에 Cyan 색을 넣는다.  
            break;  
        case 2: // Magenta 선택되면  
            color=Color.MAGENTA;       // color 변수에 Magenta 색을 넣는다.  
            break;  
        case 3: // Gray 선택되면  
            color=Color.GRAY;          // color 변수에 Gray 색을 넣는다.  
            break;  
        case 4: // Black 선택되면  
            color=Color.BLACK;         // color 변수에 Black 색을 넣는다.  
            break;  
    }  
    binding.radioGroup.check(-1); // 메뉴에서 색상이 선택되면 라디오버튼의 체크상태를 모두 해제한다.  
}  
});  
builder.create().show(); // 대화상자 표시  
break;
```



## 해결 방법

```
case R.id.shape: // 메뉴의 [붓 모양 변경]을 선택했다면
    builder=new AlertDialog.Builder(this);
    builder.setTitle(R.string.shape); // 제목을 “붓 모양 변경”으로 설정
    builder.setItems(shapeItems, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            switch (which){
                case 0: // Rect가 선택되면
                    shape=0; // shape 변수에 0을 넣어준다.
                    break;
                case 1: // Circle이 선택되면
                    shape=1; // shape 변수에 1을 넣어준다.
                    break;
                case 2: // Triangle이 선택되면
                    shape=2; // shape 변수에 2를 넣어준다.
                    break;
            }
        }
    });
    builder.create().show(); // 대화상자 표시
    break;
}
return super.onOptionsItemSelected(item);
}
```



## 해결 방법

### 파일 구성

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

```
private int color, shape=0; // 선택된 색과 모양을 넣어주기 위한 변수
```

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



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

```
public int shapeValue(){  
    return shape;  
}
```



**MyView 클래스**에서 이 함수를 통해 현재 선택된 모양을 받을 수 있다.



## 해결 방법

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

### 파일 구성

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

(x좌표, y좌표, 색, 모양)을  
한쌍으로 묶어서 저장



# 해결 방법

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

public boolean onTouchEvent(MotionEvent event) {
    mCurColor=((MainActivity)MainActivity.mContext).colorValue();
```

**MainActivity** 클래스에 있는 함수 **shapeValue()**를 호출하여 값(모양)을 받아온 후 **mshape**에 값을 넣어주어 앞으로 그려질 **모양**을 설정해주었다.

```
mshape=((MainActivity)MainActivity.mContext).shapeValue();
```

```
switch (event.getAction()) {
```

```
    case MotionEvent.ACTION_DOWN:
```

```
    case MotionEvent.ACTION_MOVE: mPoints.add(new
MyPoint(event.getX(), event.getY(), mCurColor, mshape));
```

```
        invalidate();
```

```
        return true;
```

```
    case MotionEvent.ACTION_UP:
```

```
        return true;
```

```
}
```

```
return super.onTouchEvent(event);
```

```
}
```

## 파일 구성

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

눌러지거나 움직일 경우 그때의 x, y좌표,  
선택된 색과 모양을 mPoints에 넣는다.



# 해결 방법

## 파일 구성

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

```
protected void onDraw(Canvas canvas) {  
    for (int i = 0; i < mPoints.size(); i += 2) {  
        MyPoint myPoint = mPoints.get(i);  
        mPaint.setColor(myPoint.color);  
  
        if(myPoint.shape==0) // Rect 모양이 선택되었다면  
            canvas.drawRect(myPoint.x - R, myPoint.y - R, myPoint.x + R,  
myPoint.y + R, mPaint);    // 해당 좌표와 선택된 색으로 사각형을 그려준다
```

```
        else if(myPoint.shape==1) // Circle 모양이 선택되었다면  
            canvas.drawCircle(myPoint.x,myPoint.y,R,mPaint);  
            // 해당 좌표와 선택된 색으로 원을 그려준다
```

```
        else if(myPoint.shape==2){// Triangle 모양이 선택되었다면  
            Path path=new Path();  
            path.moveTo(myPoint.x,myPoint.y);  
            path.lineTo(myPoint.x-(2*R),myPoint.y+(2*R));  
            path.lineTo(myPoint.x+(2*R),myPoint.y+(2*R));  
            canvas.drawPath(path,mPaint);  
        }  
    }  
}
```

해당 좌표와 선택된  
색으로 삼각형을  
그려준다.

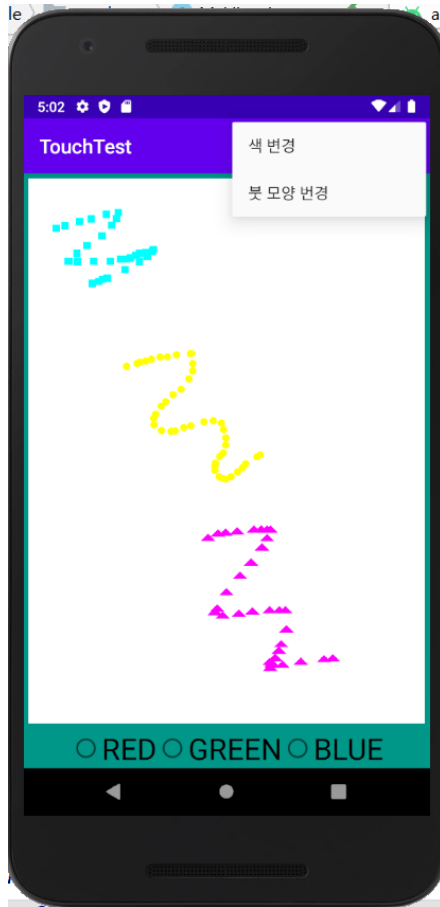


# 3 실행 결과

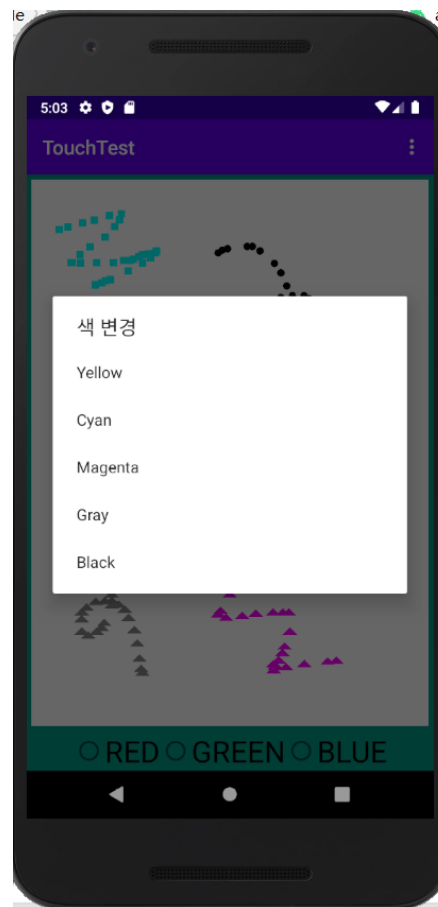




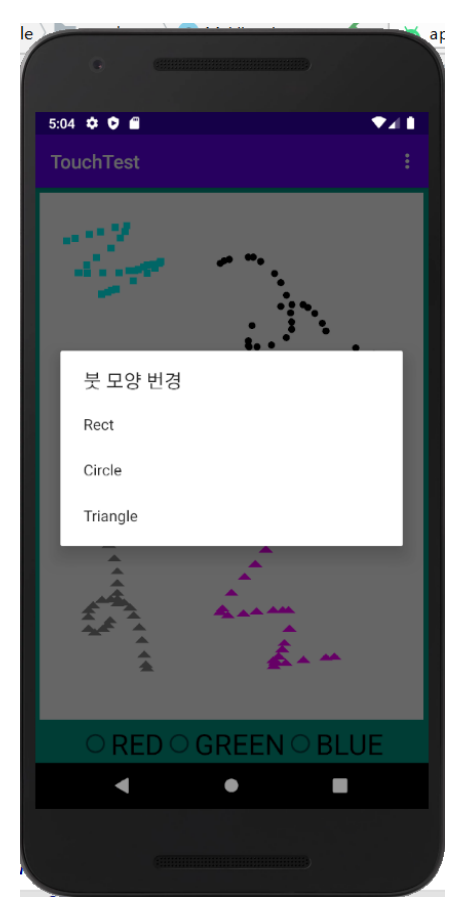
# 실행 결과



메뉴



색 변경 대화상자



붓 모양 변경 대화상자

# 4 결론



**문제1:** 메뉴의 각 [색 변경], [봇 모양 변경] 항목이 선택되면 불러지는 함수를 이용해 그 항목에 해당되는 대화상자가 나오게 한다.

**문제2:** 대화상자의 항목을 선택하면 변수(color, shape)에 해당되는 값을 넣어주고, (x좌표, y좌표, 색, 모양)을 한쌍으로 묶어주는 자료구조를 이용해 터치가 될때 마다 값들을 ArrayList에 넣고, 그릴 때 값을 꺼내서 선택된 모양으로 그린다.





**감사합니다**