Carlos Alejandro Buenrostro Ramírez

ANDREA MONSERRAT GUZMAN LOPEZ

Dalia Valeria Mejía Aguayo

Instituto Tecnologico de ciudad guzmán

Graficación y Manejo de recursos Multimedia

Desarrollo de Aplicaciones para Dispositivos Móviles

16 de mayo de 2018

# **Enunciado**

Desarrollar una app que permita realizar trazos en un canvas cuando el usuario toque y mueva su dedo sobre la pantalla.

# **Código y Layout de la aplicación.**

## **MainActivity**

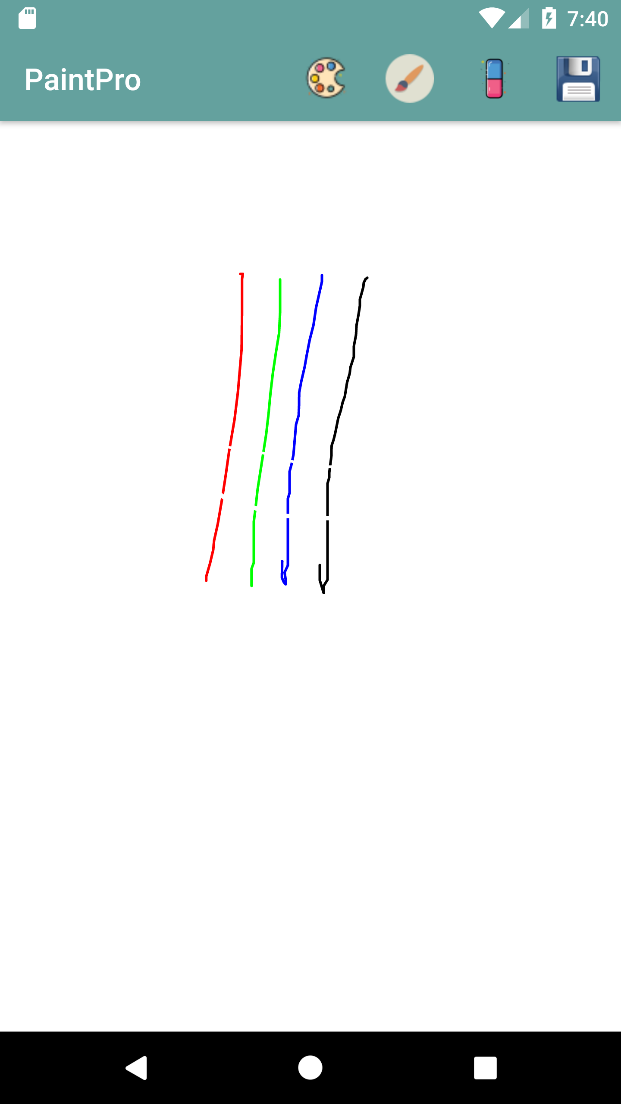
Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

## **Código**

**package** com.example.graficacion;  
**import** android.Manifest;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.os.Build;  
**import** android.os.Environment;  
**import** android.support.annotation.NonNull;  
**import** android.support.v4.app.ActivityCompat;  
**import** android.support.v4.content.ContextCompat;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.Menu;  
**import** android.view.MenuInflater;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.AdapterView;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.ListView;  
  
**import** java.io.File;  
**import** java.util.ArrayList;  
  
**public class** MainActivity **extends** AppCompatActivity  
{  
 **private** ListView **lvLista**;  
 **private** String **imagen** = **""**;  
 ArrayList **archivosLista** = **new** ArrayList();  
 **private static final int *MY\_PERMISSIONS\_REQUEST\_WRITE\_EXTERNAL\_STORAGE*** = 1;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **lvLista** = (ListView)findViewById(R.id.***lvLista***);  
  
 **if**(!checkPermission())  
 ActivityCompat.*requestPermissions*(**this**, **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***}, ***MY\_PERMISSIONS\_REQUEST\_WRITE\_EXTERNAL\_STORAGE***);  
  
 **lvLista**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> adapterView, View view, **int** i, **long** l) {  
 **imagen** = **lvLista**.getItemAtPosition(i) + **".png"**;  
 enviarImagenEditar(**imagen**);  
 }  
 });  
  
 }  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.***mi\_menu***,menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()){  
 **case** R.id.***nuevo***:  
 Intent intent = **new** Intent(**this**, segundaActivity.**class**);  
 startActivityForResult(intent, 1);  
 **return true**;  
 **case** R.id.***acercaDe***:  
 **final** AlertDialog.Builder dialogo = **new** AlertDialog.Builder(**this**);  
 dialogo.setTitle(**"PaintPRO"**);  
 dialogo.setMessage(**"Versión 1.0\n --------------INTEGRANTES--------------\n Andrea Monserrat Guzmán López\n Dalia Valeria Mejía Aguayo\n Carlos Alejandro Buenrostro Ramírez"**);  
 dialogo.setCancelable(**false**);  
 dialogo.setPositiveButton(**"OK"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogo.setCancelable(**true**);  
 }  
 });  
 dialogo.show();  
 **return true**;  
 **default**:  
 **return super**.onOptionsItemSelected(item);  
 }  
 }  
  
 *//CARGA EL LISTVIEW* **public void** cargarLista()  
 {  
 File ruta = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*); *//regresa la ruta* File archivo = **new** File(ruta.getAbsolutePath() +**"/PaintPro"**);  
 File[] nombres = archivo.listFiles();  
  
 **if** (nombres.**length** > 0)  
 {  
 **for** (**int** x=0; x< nombres.**length**; x++)  
 {  
 **if**(nombres[x].isFile())  
 {  
 String name = nombres[x].getName().replaceAll(**".png"**, **""**);  
 **archivosLista**.add(name);  
 }  
 }  
  
 ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_list\_item\_1***, **archivosLista**);  
 **lvLista**.setAdapter(adapter);  
 }  
 **else** {  
 ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_list\_item\_1***, **archivosLista**);  
 **lvLista**.setAdapter(adapter);  
 }  
 }  
  
 @Override  
 **protected void** onResume() {  
 **if**(checkPermission()) {  
 **archivosLista**.clear();  
 cargarLista();  
 }  
  
 **super**.onResume();  
 }  
  
 **private void** CrearDirectorio(String nombre)  
 {  
 File file = **new** File(Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*), nombre);  
 **if**(!file.exists())  
 file.mkdirs();*//la primera vez crea el directorio* }  
  
 **public void** enviarImagenEditar(String imagen){  
 Intent intent = **new** Intent(**this**,segundaActivity.**class**);  
 intent.putExtra(**"imagen"**, imagen);  
 startActivity(intent);  
 }  
  
 **public boolean** checkPermission(){  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***M***)  
 {  
 **if** (ContextCompat.*checkSelfPermission*(**this**, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***) != PackageManager.***PERMISSION\_GRANTED***)  
 **return false**;  
 }  
 **return true**;  
 }  
  
 @Override  
 **public void** onRequestPermissionsResult(**int** requestCode, @NonNull String[] permissions, @NonNull **int**[] grantResults) {  
  
 **switch** (requestCode)   
 {  
 **case *MY\_PERMISSIONS\_REQUEST\_WRITE\_EXTERNAL\_STORAGE***:  
  
 **if** (grantResults.**length** > 0 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***)  
 {  
 *//Aquí lo que se hace si aceptan el permiso* CrearDirectorio(**"PaintPro"**);  
 cargarLista();  
 }  
 **else** {  
 *//Aquí lo que se hace si no lo aceptan* }  
 **break**;  
 }  
 }  
}

## **SegundaActivity**



## **Código**

**package** com.example.graficacion;  
  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.graphics.Bitmap;  
**import** android.graphics.BitmapFactory;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.graphics.Path;  
**import** android.os.Environment;  
**import** android.support.constraint.ConstraintLayout;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.text.InputType;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuInflater;  
**import** android.view.MenuItem;  
**import** android.view.MotionEvent;  
**import** android.view.View;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**import** java.io.File;  
**import** java.io.FileOutputStream;  
  
**public class** segundaActivity **extends** AppCompatActivity {  
  
 **private** String **colores**[] = {**"Rojo"**,**"Verde"**,**"Azul"**,**"Negro"**,**"Blanco"**};  
 **private** String **pincel**[] = {**"1"**,**"2"**,**"4"**,**"6"**,**"8"**,**"10"**};  
 **private** String **borrador**[] = {**"Pequeño"**, **"Normal"**, **"Grande"**};  
 String **accion** = **"pintar"**;  
 **int tamañoPincel**, **colorPincel** = Color.***RED***, **tamañoBorrador** = 20;  
 Lienzo **fondo**;  
 **boolean grabarDibujo** = **false**, **esModificar** = **false**;  
 **private** Bundle **bundle**;  
 String **pathImagen** = **null**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_segunda***);  
 ConstraintLayout miLayout = (ConstraintLayout) findViewById(R.id.***miLayout***);  
 **fondo** = **new** Lienzo(**this**);  
  
  
 **bundle** = getIntent().getExtras();  
 **if** (**bundle** != **null**) {  
 **esModificar** = **true**;  
 *//trae el archivo* File ruta = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*);  
 File archivo = **new** File(ruta.getAbsolutePath() +**"/PaintPro"**);  
 String[] nombres = archivo.list();  
 **for** (**int** x=0;x<nombres.**length**;x++){  
 **if** (nombres[x].equals(**bundle**.getString(**"imagen"**))) {  
 **pathImagen** = archivo.getAbsolutePath() + **"/"** + nombres[x];  
 }  
 }  
 }  
 miLayout.addView(**fondo**);  
 }  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.***menu2***,menu);  
 **return true**;  
 }  
  
 **class** Lienzo **extends** View {  
 Path **path** = **new** Path();  
 Paint **pincel**, **pintarCanvas**;  
 Canvas **dibujarCanvas**;  
 Bitmap **canvasBitmap**;  
  
 **public** Lienzo(Context context) {  
 **super**(context);  
 **if**(**accion**.equals(**"pintar"**))  
 pintar();  
 **if**(**accion**.equals(**"borrar"**))  
 borrar();  
 }  
 **public void** pintar(){  
 **pincel** = **new** Paint();  
 **pincel**.setAntiAlias(**true**);  
 **pincel**.setColor(**colorPincel**);  
 **pincel**.setStrokeWidth(**tamañoPincel**);  
 **pincel**.setStyle(Paint.Style.***STROKE***);  
 **pincel**.setStrokeJoin(Paint.Join.***ROUND***);  
 **pincel**.setStrokeCap(Paint.Cap.***ROUND***);  
 **pintarCanvas** = **new** Paint(Paint.***DITHER\_FLAG***);  
 }  
 **public void** borrar(){  
 **pincel** = **new** Paint();  
 **pincel**.setAntiAlias(**true**);  
 **pincel**.setColor(Color.***WHITE***);  
 **pincel**.setStrokeWidth(**tamañoBorrador**);  
 **pincel**.setStyle(Paint.Style.***STROKE***);  
 **pincel**.setStrokeJoin(Paint.Join.***ROUND***);  
 **pincel**.setStrokeCap(Paint.Cap.***ROUND***);  
 **pintarCanvas** = **new** Paint(Paint.***DITHER\_FLAG***);  
 }  
 @Override  
 **protected void** onSizeChanged(**int** w, **int** h, **int** oldw, **int** oldh) {  
 **super**.onSizeChanged(w, h, oldw, oldh);  
 **if** (!**esModificar**){  
 **canvasBitmap** = Bitmap.*createBitmap*(w,h,Bitmap.Config.***ARGB\_8888***);  
 **dibujarCanvas** = **new** Canvas(**canvasBitmap**);  
 }**else**{  
 Bitmap workingBitmap = Bitmap.*createBitmap*(BitmapFactory.*decodeFile*(**pathImagen**));  
 **canvasBitmap** = workingBitmap.copy(Bitmap.Config.***ARGB\_8888***, **true**);  
 **dibujarCanvas** = **new** Canvas(**canvasBitmap**);  
 }*//Esto agregue para poder modificar la imagen, jeje saludos xD.* }  
 @Override  
 **protected void** onDraw(Canvas canvas) {  
 canvas.drawBitmap(**canvasBitmap**,0,0,**pintarCanvas**);  
 **if**(**accion**.equals(**"pintar"**)) {  
 **pincel**.setColor(**colorPincel**);  
 **pincel**.setStrokeWidth(**tamañoPincel**);  
 }  
 **else if**(**accion**.equals(**"borrar"**))  
 {  
 **pincel**.setColor(Color.***WHITE***);  
 **pincel**.setStrokeWidth(**tamañoBorrador**);  
 }  
 canvas.drawPath(**path**,**pincel**);  
  
 **super**.onDraw(canvas);  
  
 }  
 @Override  
 **public boolean** onTouchEvent(MotionEvent event) {  
 **float** x = event.getX();  
 **float** y = event.getY();  
 **switch** (event.getAction())  
 {  
 **case** MotionEvent.***ACTION\_DOWN***:  
 **path**.moveTo(x,y);  
 **break**;  
 **case** MotionEvent.***ACTION\_MOVE***:  
 **path**.lineTo(x,y);  
 **break**;  
 **case** MotionEvent.***ACTION\_UP***:  
 **path**.lineTo(x,y);  
 **dibujarCanvas**.drawPath(**path**,**pincel**);  
 **path**.reset();  
 **break**;  
 **default**:  
 **return false**;  
 }  
 invalidate();  
 **return true**;  
 }  
 }  
 @Override  
 **public boolean** onOptionsItemSelected(**final** MenuItem item) {  
 **switch** (item.getItemId())  
 {  
 **case** R.id.***paleta***:  
 **final** AlertDialog.Builder dialogo = **new** AlertDialog.Builder(**this**);  
 dialogo.setTitle(**"PALETA DE COLORES"**);  
 dialogo.setItems(**colores**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 **colorPincel** = color(i);  
 }  
 });  
 dialogo.setCancelable(**false**);  
 dialogo.show();  
 **accion** = **"pintar"**;  
 **return true**;  
 **case** R.id.***grosor***:  
 AlertDialog.Builder grosor = **new** AlertDialog.Builder(**this**);  
 grosor.setTitle(**"GROSOR DEL PINCEL"**);  
 grosor.setItems(**pincel**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 **tamañoPincel** = tamaño(i);  
 }  
 });  
 grosor.setCancelable(**false**);  
 grosor.show();  
 **accion** = **"pintar"**;  
 **return true**;  
 **case** R.id.***borrador***:  
 AlertDialog.Builder borrar = **new** AlertDialog.Builder(**this**);  
 borrar.setTitle(**"TAMAÑO DEL BORRADOR"**);  
 borrar.setItems(**borrador**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 **tamañoBorrador** = borrarTamaño(i);  
 }  
 });  
 borrar.setCancelable(**false**);  
 borrar.show();  
 **accion** = **"borrar"**;  
 **return true**;  
 **case** R.id.***grabar***:  
 **if**(**esModificar**)  
 reemplazar(**bundle**.getString(**"imagen"**));  
 **else** guardar();  
 **return true**;  
 **default**:  
 **return super**.onOptionsItemSelected(item);  
 }  
 }  
  
 **private void** reemplazar(String nombre)  
 {  
 String mensaje;  
 File ruta = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*); *//regresa la ruta* File archivo = **new** File(ruta.getAbsolutePath() +**"/PaintPro"**);  
 File[] numArchivos = archivo.listFiles();  
  
 **for** (**int** x=0; x< numArchivos.**length**; x++)  
 {  
 **if**(numArchivos[x].isFile())  
 {  
 **if**(numArchivos[x].getName().replaceAll(**".png"**, **""**).equals(nombre.replace(**".png"**, **""**)))  
 {  
 numArchivos[x].delete();  
 **fondo**.setDrawingCacheEnabled(**true**);  
 **try** {  
 **grabarDibujo** = **fondo**.getDrawingCache().compress(Bitmap.CompressFormat.***JPEG***, 100, **new** FileOutputStream(**new** File(Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*) + **"/PaintPro/"**, nombre)));  
 } **catch** (Exception e) {  
 Log.*e*(**"Error--------->"**, e.toString());  
 }  
  
 **if** (**grabarDibujo**)  
 mensaje = **"Dibujo guadardo"**;  
 **else** mensaje = **"No se ha podido grabar"**;  
  
 Toast.*makeText*(getApplicationContext(), mensaje, Toast.***LENGTH\_SHORT***).show();  
 **fondo**.destroyDrawingCache();  
 finish();  
 }  
 }  
 }  
 }  
  
 **private void** guardar()  
 {  
 **final** AlertDialog.Builder dialog = **new** AlertDialog.Builder(**this**);  
 dialog.setTitle(**"GUARDAR DIBUJO"**);  
 dialog.setMessage(**"Ingrese un nombre para el dibujo a guardar"**);  
 **final** EditText input = **new** EditText(**this**);  
 input.setInputType(InputType.***TYPE\_TEXT\_VARIATION\_FILTER***);  
 dialog.setView(input);  
 dialog.setCancelable(**false**);  
 dialog.setPositiveButton(**"Aceptar"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
  
 **fondo**.setBackgroundColor(Color.***WHITE***);  
 String nombre = String.*valueOf*(input.getText());  
 String mensaje;  
  
 **if**(validar(nombre))  
 {  
 **if**(!existeArchivo(nombre))  
 {  
 **fondo**.setDrawingCacheEnabled(**true**);  
 **try** {  
 **grabarDibujo** = **fondo**.getDrawingCache().compress(Bitmap.CompressFormat.***JPEG***, 100, **new** FileOutputStream(**new** File(Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*) + **"/PaintPro/"**, nombre + **".png"**)));  
 } **catch** (Exception e) {  
 Log.*e*(**"Error--------->"**, e.toString());  
 }  
  
 **if** (**grabarDibujo**)  
 mensaje = **"Dibujo guadardo"**;  
 **else** mensaje = **"No se ha podido grabar"**;  
  
 Toast.*makeText*(getApplicationContext(), mensaje, Toast.***LENGTH\_SHORT***).show();  
 **fondo**.destroyDrawingCache();  
 *//cierra la activity* finish();  
 }  
 **else** Toast.*makeText*(getApplicationContext(), **"Ya existe este archivo"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else** {  
 *//seria pro que no se cerrará aquí* Toast.*makeText*(getApplicationContext(), **"Solo se aceptan números, minusculas y guiones"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 });  
 dialog.setNegativeButton(**"Cancelar"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 }  
 });  
 dialog.show();  
 }  
  
 **private boolean** validar(String nombre)  
 {  
 **int** c =1;  
 *//minuscula* **if**(!Character.*isUpperCase*(nombre.charAt(0)))  
 {  
 **while** (c <= nombre.length()-1)  
 {  
 **if**(!Character.*isUpperCase*(nombre.charAt(c)) || nombre.charAt(c) == **'\_'** || Character.*isDigit*(nombre.charAt(c)) )  
 c++;  
 **else  
 return false**;  
 }  
 **return true**;  
 }  
 **return false**;  
 }  
  
 **public boolean** existeArchivo(String nombre)  
 {  
 File ruta = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*); *//regresa la ruta* File archivo = **new** File(ruta.getAbsolutePath() +**"/PaintPro"**);  
 File[] numArchivos = archivo.listFiles();  
  
 **if** (numArchivos.**length** > 0)  
 {  
 **for** (**int** x=0; x< numArchivos.**length**; x++)  
 {  
 **if**(numArchivos[x].isFile())  
 {  
 *//falta que lo encuentré* **if**(numArchivos[x].getName().replaceAll(**".png"**, **""**).equals(nombre))  
 **return true**;  
 }  
 }  
 }  
  
 **return false**;  
 }  
  
 **private int** borrarTamaño(**int** i) {  
 **switch** (i){  
 **case** 0: **tamañoBorrador** = 20;  
 **break**;  
 **case** 1: **tamañoBorrador** = 40;  
 **break**;  
 **case** 2: **tamañoBorrador** = 80;  
 }  
 **return tamañoBorrador**;  
 }  
  
 **private int** tamaño(**int** i) {  
 **switch** (i)  
 {  
 **case** 0: **tamañoPincel** = 1;  
 **break**;  
 **case** 1: **tamañoPincel** = 2;  
 **break**;  
 **case** 2: **tamañoPincel** = 4;  
 **break**;  
 **case** 3: **tamañoPincel** = 6;  
 **break**;  
 **case** 4: **tamañoPincel** = 8;  
 **break**;  
 **case** 5: **tamañoPincel** = 10;  
 }  
 **return tamañoPincel**;  
 }  
  
 **private int** color(**int** i) {  
 **int** color = Color.***RED***;  
 **switch** (i)  
 {  
 **case** 0: color = Color.***RED***;  
 **break**;  
 **case** 1: color = Color.***GREEN***;  
 **break**;  
 **case** 2: color = Color.***BLUE***;  
 **break**;  
 **case** 3: color = Color.***BLACK***;  
 **break**;  
 **case** 4: color = Color.***WHITE***;  
 }  
 **return** color;  
 }  
  
}