



CAPÍTULO

**5**

# Juegos



## 5.1 ASPECTOS GENERALES

### Canvas

*Canvas (lienzo) es una interfaz que permite pintar en un área rectangular, que refleja las acciones sobre un mapa de bits (Bitmap) asociado:*

```
Bitmap b = Bitmap.createBitmap  
    (100, 100, Bitmap.Config.ARGB_8888);  
Canvas c = new Canvas(b);
```

*Para escribir en la pantalla, no será necesario instanciar Canvas directamente.*

### Canvas-Métodos

*Principales métodos de Canvas:*

**drawBitmap**(Bitmap bitmap, float left, float top, Paint paint). Dibuja un Bitmap con la esquina superior izquierda en la posición indicada del lienzo.

**drawBitmap**(Bitmap bitmap, Rect src, Rect dst, Paint paint). Escala/ajusta un Bitmap al área (sic) de destino.

**drawCircle**(float cx, float cy, float radius, Paint paint). Dibuja un círculo.

**drawColor**(int color). Llena el lienzo con un color.

**drawLine**(float startX, float startY, float stopX, float stopY, Paint paint). Dibuja una línea de (startX, startY) a (stopX, stopY).

**drawRect**(float left, float top, float right, float bottom, Paint paint). Dibuja el rectángulo especificado.

**drawText**(String text, float x, float y, Paint paint). Dibuja texto en (x, y) con el Paint especificado.

**drawLine**(float startX, float startY, float stopX, float stopY, Paint paint). Dibuja una línea de (startX, startY) a (stopX, stopY).

**drawRect**(float left, float top, float right, float bottom, Paint paint). Dibuja el rectángulo especificado.

**drawText**(String text, float x, float y, Paint paint) Dibuja texto en (x, y) con el Paint especificado.

[**drawARGB/drawRGB/drawColor**. Llena el canvas con un solo color.]

Rafael Morón Abad

Fuente: recursos.crfptic.es:9080

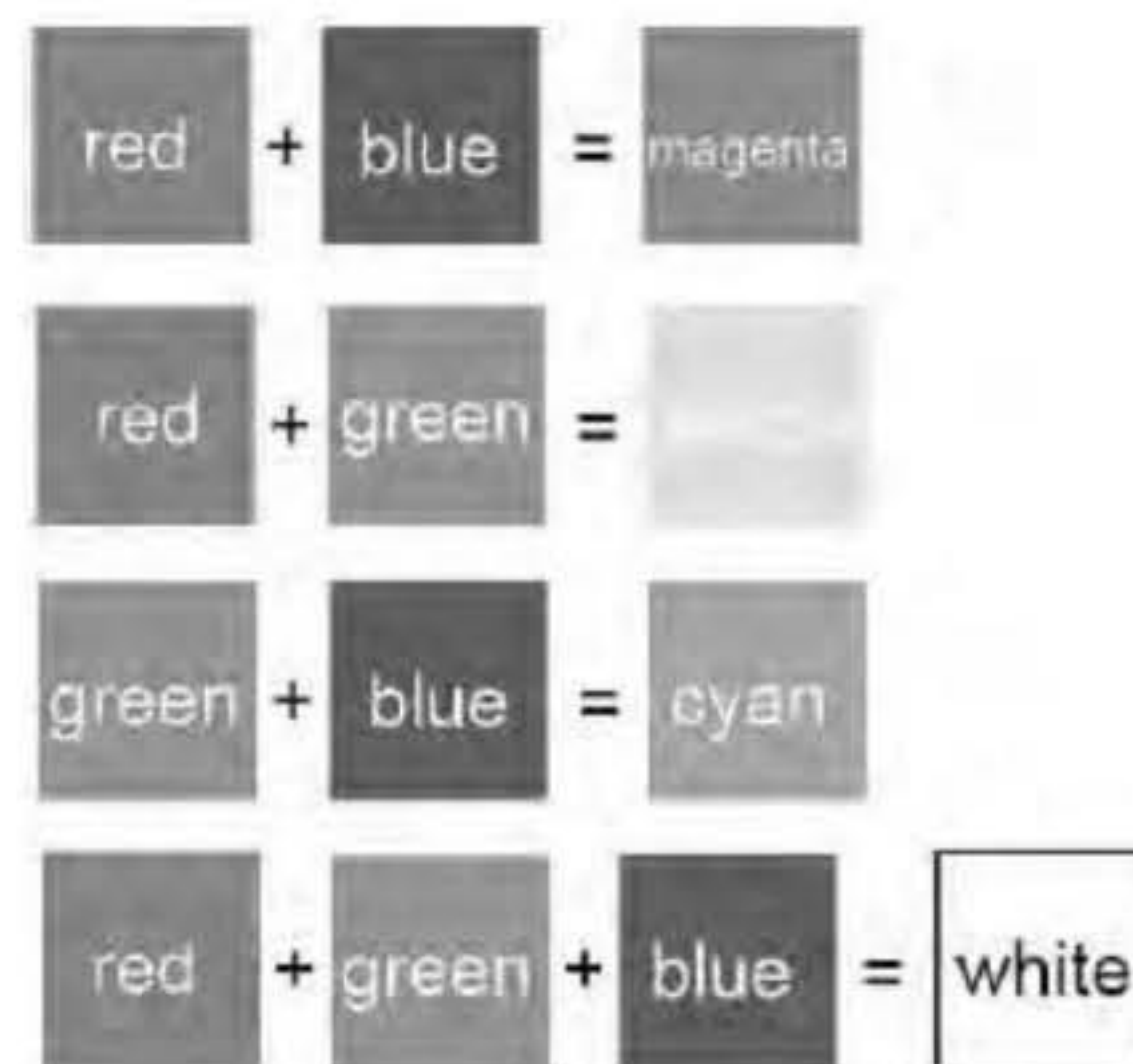
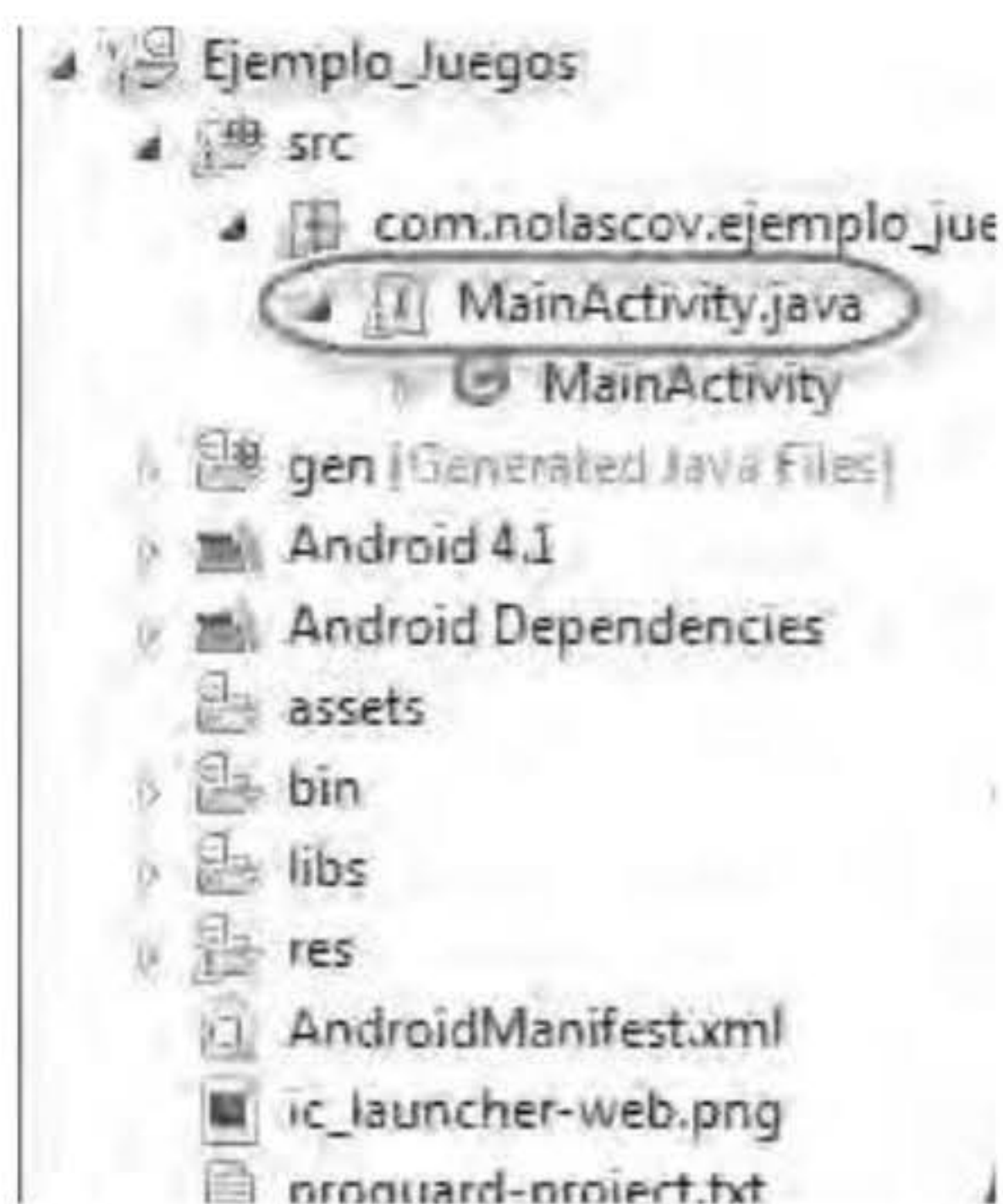
## 5.2 APLICACIÓN DE CANVAS: LÍNEAS, RECTÁNGULO Y CÍRCULOS

Cree una aplicación que permita demostrar el uso de canvas dibujando algunas figuras.

**Paso 1:** Después de haber creado la primera aplicación para Android, cree un nuevo proyecto seleccionando *Archivo > Nuevo > Android proyecto*.

**Paso 2:** Ingrese el nombre del proyecto: Ejemplo\_juegos (los pasos de creación de un proyecto ya se detallaron en ejemplos anteriores).

**Paso 3:** Modifique la actividad principal.



```

MainActivity.java El código de colores RGB, rojo - verde - azul
package com.nolascov.ejemplo_juegos;

import android.os.Bundle;

public class MainActivity extends Activity {

    class RenderView extends View {

        public RenderView(Context context) {
            super(context);
        }

        protected void onDraw(Canvas canvas) {
            canvas.drawRGB(0, 255, 0);
        }

    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new RenderView(this));
    }

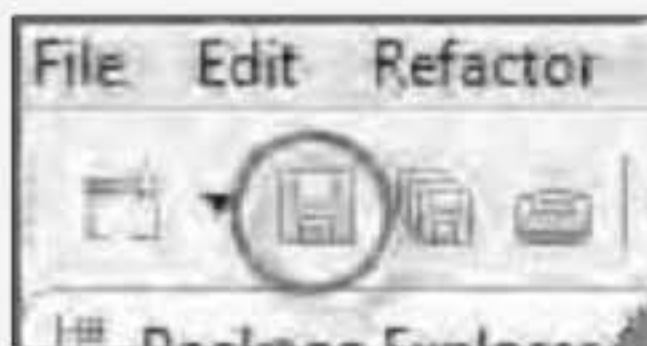
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
    
```

El color de fondo del canvas :  
 canvas.drawRGB(0, 255, 0);

Mostrar por pantalla lo  
 que se esta dibujando

**EJECUTANDO LA APLICACIÓN**

**Paso 1:** Guarde los cambios realizados, haciendo clic en el ícono



o presionando la combinación <Ctrl+S>.

**Paso 2:** Seleccione el nombre del proyecto en Eclipse. Haga clic derecho, seleccione la opción Run As > Android Application y observe que el emulador se pondrá en marcha:



**Paso 3:** Haga clic y mueva el ícono del candado en la zona superior para tocar un círculo que aparece al mover el ratón. Esto abre el emulador.

La siguiente figura muestra la ventana principal de la pantalla de Android 4.1.



**Paso 4:** Si el emulador se bloquea, deslice el botón de desbloqueo para contrarrestar su función.

La figura muestra la aplicación que se ejecuta en el emulador de Android.



### 5.2.1 DIBUJAR UNA LÍNEA VERTICAL

Para dibujar una línea vertical de color azul en medio de la pantalla, realice los siguientes pasos:

```

MainActivity.java
package com.nolascov.ejemplo_juegos;

import android.os.Bundle;

public class MainActivity extends Activity {

    class RenderView extends View {
        Paint paint;

        public RenderView(Context context) {
            super(context);
            paint = new Paint();
        }

        protected void onDraw(Canvas canvas) {
            canvas.drawRGB(0, 255, 0);
            paint.setColor(Color.BLUE);
            canvas.drawLine(canvas.getWidth()/2, 0, canvas.getWidth()/2, canvas.getHeight(), paint);
        }
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new RenderView(this));
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
    
```

La clase paint contiene información del color y estilo para dibujar las figuras

Línea de color azul

Dibuja una línea de (startX, startY) a (stopX, stopY).



## 5.2.2 DIBUJAR RECTÁNGULOS

Para dibujar algunos rectángulos, realice los siguientes pasos:

### Paso 1:

```

MainActivity.java
package com.nolasco.v.ejemplo_juegos;

import android.os.Bundle;

public class MainActivity extends Activity {

    class RenderView extends View {
        Paint paint;

        public RenderView(Context context) {
            super(context);
            paint = new Paint();
        }

        protected void onDraw(Canvas canvas) {
            canvas.drawRGB(0, 255, 0);
            int ancho=canvas.getWidth();
            paint.setColor(Color.BLUE);
            canvas.drawLine(canvas.getWidth()/2, 0,canvas.getWidth()/2,canvas.getHeight(), paint);

            paint.setARGB(255,255,0,0);
            canvas.drawRect(10,20,ancho-10,50,paint);
        }
    }
}

```

Rectángulo de color rojo

Calculamos el ancho de la pantalla

Dibujamos un rectángulo desde la coordenada columna:10 y fila 20 hasta la columna que coincide con el ancho de la pantalla menos 10 pixeles y la fila 50. Además le pasamos el pincel a utilizar



### Paso 2:

```

MainActivity.java
package com.nolasco.v.ejemplo_juegos;

import android.os.Bundle;

public class MainActivity extends Activity {

    class RenderView extends View {
        Paint paint;

        public RenderView(Context context) {
            super(context);
            paint = new Paint();
        }

        protected void onDraw(Canvas canvas) {
            canvas.drawRGB(0, 255, 0);
            int ancho=canvas.getWidth();
            paint.setColor(Color.BLUE);
            canvas.drawLine(canvas.getWidth()/2, 0,canvas.getWidth()/2,canvas.getHeight(), paint);

            paint.setARGB(255,255,0,0);
            canvas.drawRect(10,20,ancho-10,50,paint);

            paint.setStyle(Style.STROKE);
            canvas.drawRect(10,80,ancho-10,100,paint);
        }
    }
}

```

El siguiente rectángulo configuramos el pincel para que solo pinte el perímetro llamando al método `setStyle` y pasando la constante `STROKE`

Dibujamos un rectángulo desde la coordenada columna:10 y fila 80 hasta la columna que coincide con el ancho de la pantalla menos 10 pixeles y la fila 100. Además le



### 5.2.3 DIBUJAR UN CÍRCULO

Para dibujar un círculo de color blanco, realice los siguientes pasos:

```

MainActivity.java
class RenderView extends View {
    Paint paint;
    public RenderView(Context context) {
        super(context);
        paint = new Paint();
    }
    protected void onDraw(Canvas canvas) {
        canvas.drawRGB(0, 255, 0);
        int ancho=canvas.getWidth();

        paint.setColor(Color.BLUE);
        canvas.drawLine(canvas.getWidth()/2, 0,canvas.getWidth(), 0);

        paint.setARGB(255,255,0,0);
        canvas.drawRect(10,20,ancho-10,50,paint);

        paint.setStyle(Style.STROKE);
        canvas.drawRect(10,80,ancho-10,100,paint);

        paint.setColor(Color.WHITE);
        canvas.drawCircle(ancho/2, 220, 100,paint);
    }
}
    
```

Diagram annotations:

- A callout box labeled "círculo de color blanco" points to the `canvas.drawCircle(ancho/2, 220, 100,paint);` line in the code.
- Another callout box explains: "dibuja un círculo , los dos primeros argumentos especifican las coordenadas del punto central el segundo el radio". Arrows point from this text to the `ancho/2` and `220` arguments in the `drawCircle` method call.





**5.3 APLICACIÓN DE CANVAS: TEXTO**

```
package com.nolascov.ejemplo_juegos;
import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Paint.Style;
import android.graphics.Path;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.support.v4.app.NavUtils;

public class MainActivity extends Activity {

    class RenderView extends View {
        Paint paint;
        public RenderView(Context context) {
            super(context);
            paint = new Paint();
        }

        protected void onDraw(Canvas canvas) {
            canvas.drawRGB(0, 255, 0);
            int ancho=canvas.getWidth();

            paint.setColor(Color.BLUE);
            canvas.drawLine(canvas.getWidth()/2, 0,canvas.getWidth()/2,canvas.getHeight(),
            paint);

            paint.setARGB(255,255,0,0);
            canvas.drawRect(10,20,ancho-10,50,paint);

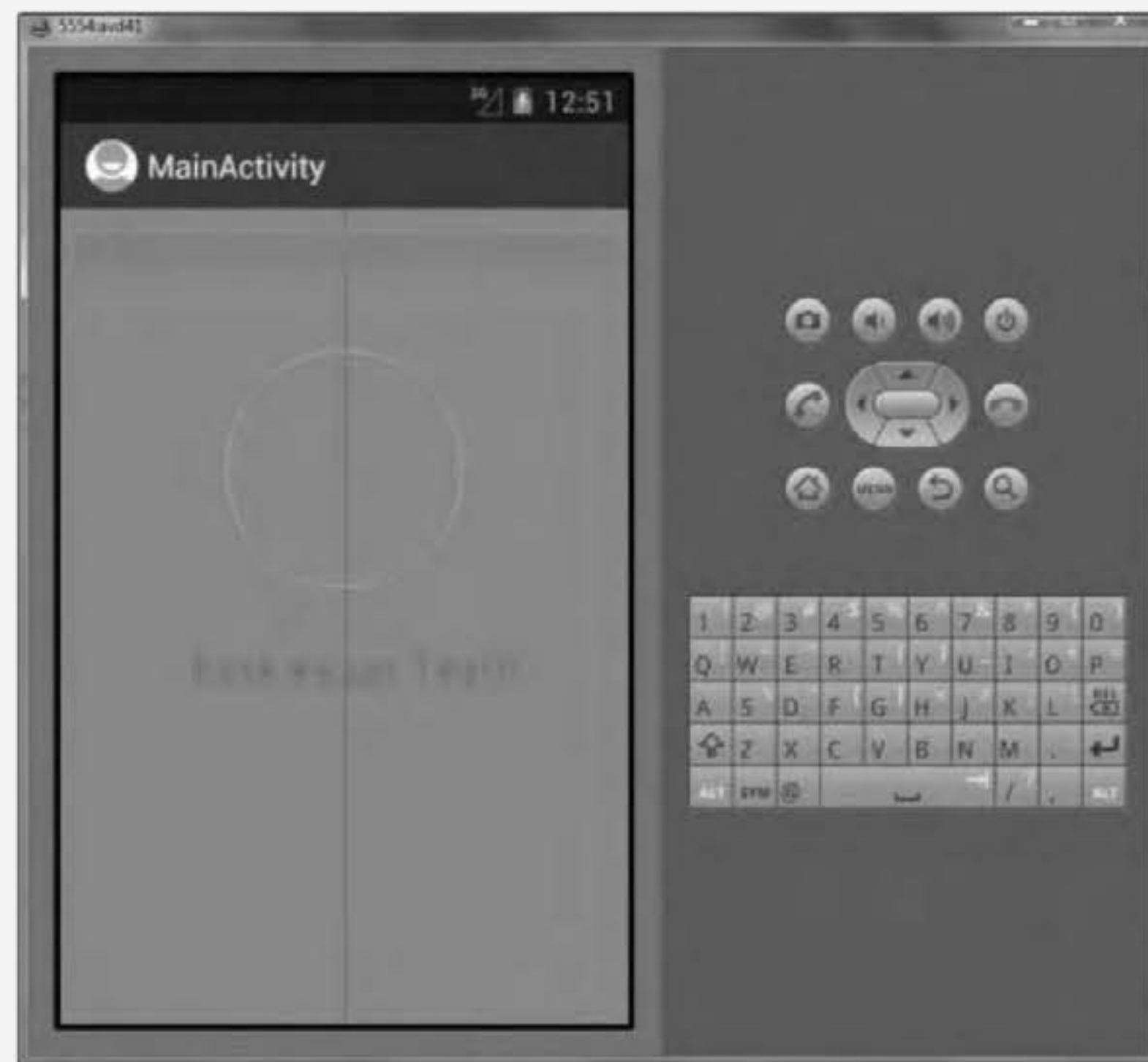
            paint.setStyle(Style.STROKE);
            canvas.drawRect(10,80,ancho-10,100,paint);

            paint.setColor(Color.WHITE);
            canvas.drawCircle(ancho/2, 220, 100,paint);

            paint.setColor(Color.RED);
            paint.setTextSize(38);
            paint.setShadowLayer(5, 0, 5, Color.BLACK);
            canvas.drawText("Este es un Texto", 60, 400, paint);
        }
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new RenderView(this));
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

**EJECUTANDO LA APLICACIÓN****5.4 APLICACIÓN DE FUENTES EXTERNAS: TEXTO**

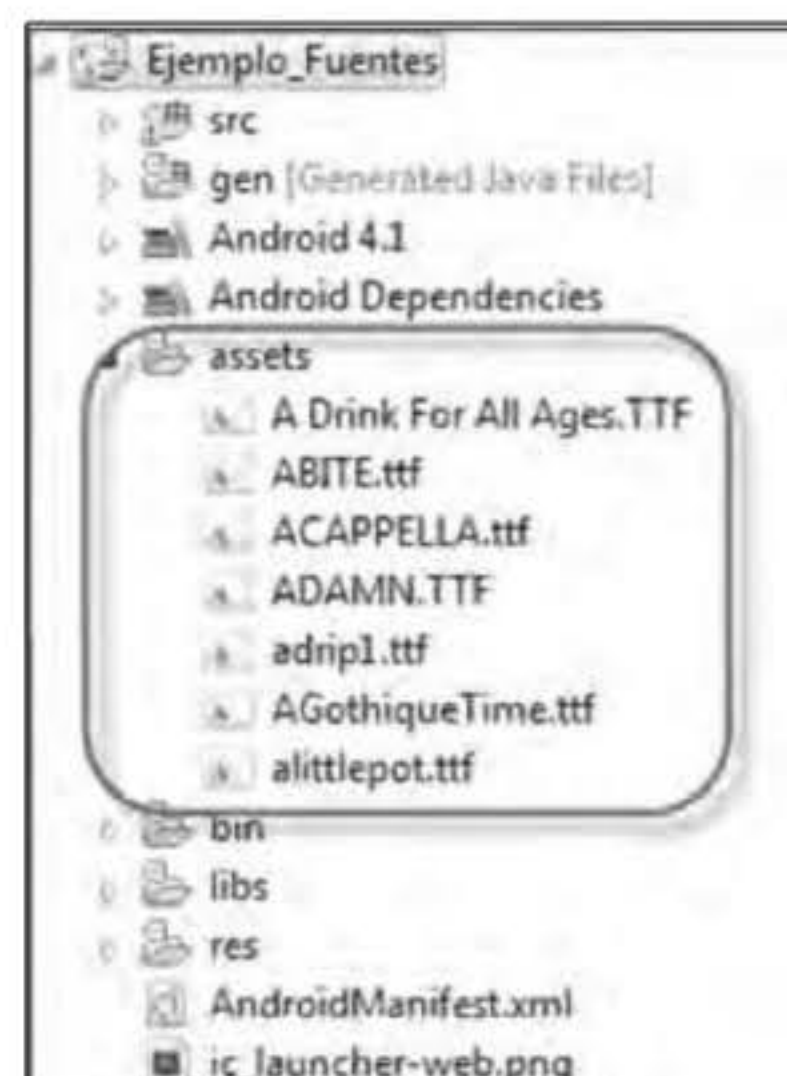
Para confeccionar aplicaciones e incorporar fuentes True Type externas, implemente una aplicación que muestre un texto con una fuente externa.

Para crear una aplicación que permita demostrar el uso de fuentes externas, realice los siguientes pasos:

**Paso 1:** Después de haber creado la primera aplicación para Android, cree un nuevo proyecto seleccionando *Archivo > Nuevo > Android proyecto*.

**Paso 2:** Ingrese el nombre del proyecto: *Ejemplo\_Fuentes* (los pasos de creación de un proyecto ya se detallaron en ejemplos anteriores).

**Paso 3:** Descargue las fuentes a utilizar y cópielas en la carpeta *assets* (ver fig.).



**Paso 4:** Modifique la actividad principal.

```
package com.nolasco.ejemplo_fuentes;

import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Rect;
import android.graphics.Typeface;
import android.graphics.Paint.Style;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.support.v4.app.NavUtils;

public class MainActivity extends Activity {

    class RenderView extends View {
        Paint paint;
        Typeface face;

        public RenderView(Context context) {
            super(context);
            paint = new Paint();
        }

        protected void onDraw(Canvas canvas) {
            canvas.drawRGB(255,255,255);
            paint.setARGB(255,0,0,0);
            paint.setTextSize(30);

            face = Typeface.createFromAsset(getAssets(),"A Drink For All Ages.TTF");
            paint.setTypeface(face);
            canvas.drawText("Tipo de Fuente : A Drink For All Ages", 0, 220, paint);

            face= Typeface.createFromAsset(getAssets(),"ABITE.ttf");
            paint.setTypeface(face);
            canvas.drawText("Tipo de Fuente : ABITE", 0, 320, paint);

            face= Typeface.createFromAsset(getAssets(),"ACAPPELLA.ttf");
            paint.setTypeface(face);
            canvas.drawText("Tipo de Fuente : ACAPPELLA", 0, 420, paint);

            face= Typeface.createFromAsset(getAssets(),"ADAMN.TTF");
            paint.setTypeface(face);
            canvas.drawText("Tipo de Fuente : ADAMN", 0, 520, paint);

            face= Typeface.createFromAsset(getAssets(),"adrip1.ttf");
            paint.setTypeface(face);
            canvas.drawText("Tipo de Fuente : adrip1", 0, 620, paint);

            face = Typeface.createFromAsset(getAssets(),"AGothiqueTime.ttf");
            paint.setTypeface(face);
            canvas.drawText("Tipo de Fuente : aGothiqueTime", 0, 720, paint);
        }
    }
}
```

```

        face= Typeface.createFromAsset(getAssets(),"alittlepot.ttf");
        paint.setTypeface(face);
        canvas.drawText("Tipo de Fuente : alittlepot", 0, 820, paint);
    }
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(new RenderView(this));
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.activity_main, menu);
    return true;
}
}

```

## 5.5 DATOS

Una manera muy sencilla y cómoda de guardar; por ejemplo, la información de un juego como es el usuario y la puntuación es por medio del uso de las preferencias.



### 5.5.1 PERSISTENCIA Y PREFERENCIA

Son almacenes clave valor que permite crear varios almacenes de preferencias; sin embargo, normalmente bastará con usar un almacén por defecto para cada aplicación.

### 5.5.2 APLICACION SOBRE EL USO DE PREFERENCIAS

Cree una aplicación que permita demostrar el uso de Preferencias.

**Paso 1:** Después de haber creado la primera aplicación para Android, cree un nuevo proyecto seleccionando *Archivo > Nuevo > Android proyecto*.

**Paso 2:** Ingrese el nombre del proyecto: *Ejemplo\_Preferencias* (los pasos de creación de un proyecto ya se detallaron en ejemplos anteriores).

**Paso 3:** Modifique el archivo: *activity\_main.xml*.



## activity\_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="43dp"
        android:layout_marginTop="22dp"
        android:text="Ingrese un Valor" />

    <EditText
        android:id="@+id/Datos"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
        android:layout_marginTop="39dp"
        android:ems="10" />

    <Spinner
        android:id="@+id/Tipo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/Datos"
        android:layout_below="@+id/Datos"
        android:layout_marginTop="24dp"
        android:entries="@array/etiquetas"
        android:entryValues="@array/valores"
        />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/Tipo"
        android:layout_below="@+id/Tipo"
        android:layout_marginTop="27dp"
        android:text="Acccion" />

    <TableLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView2"
        android:layout_centerVertical="true" >

        <TableRow
            android:id="@+id/tableRow1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
```

```
<Button
    android:id="@+id/btn_Leer"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Leer" />

<Button
    android:id="@+id/btn_Salvar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Salvar" />

</TableRow>
</TableLayout>

</RelativeLayout>
```

El Spinner utilizará los valores del archivo: strings.xml , desde un arreglo de cadenas.



```
strings.xml
<resources>
    <string name="app_name">Ejemplo_Preferencias </string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>
    <string name="title_activity_main">MainActivity</string>

    <string-array name="etiquetas">
        <item >String</item>
        <item >Boolean</item>
        <item >Integer</item>
        <item >Float</item>
    </string-array>

    <array name="valores">
        <item >0</item>
        <item >1</item>
        <item >2</item>
        <item >3</item>
    </array>
</resources>
```

### MainActivity.java

```
package com.nolascov.ejemplo_preferencias;

import android.os.Bundle;
import android.app.Activity;
import android.content.SharedPreferences;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
```

```
package com.nolascov.ejemplo_preferencias;

import android.os.Bundle;
import android.app.Activity;
import android.content.SharedPreferences;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

public class MainActivity extends Activity {

    private EditText Datos;
    private Spinner Tipo;
    private Button btn_Leer;
    private Button btn_Salvar;
    private TextView Salida;

    // variable para Las preferencias
    private static String PREFERENCIA="MisPreferencias";

    //modo de La preferencia privado solo acceso para La actividad
    private static int mode =Activity.MODE_PRIVATE;

    //variable de preferencia compartida
    private SharedPreferences MisPreferenciasCompartidas;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Datos = (EditText) findViewById(R.id.Datos);
        Tipo = (Spinner) findViewById(R.id.Tipo);
        btn_Leer = (Button) findViewById(R.id.btn_Leer);
        btn_Salvar = (Button) findViewById(R.id.btn_Salvar);
        Salida = (TextView) findViewById(R.id.Salida);

        //creando nuestra preferencia
        MisPreferenciasCompartidas= getSharedPreferences(PREFERENCIA, mode);

        this.btn_Leer.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                //datos leidos de La preferencia
                String cadena = "";
                //el tipo de La preferencia , el item seleccioando
                Integer i = Tipo.getSelectedItemPosition();
                switch(i)
                {
                    case 0:
                        cadena =MisPreferenciasCompartidas.getString("cadena", "Hola");
                        break;
                    case 1:
                        cadena =new Boolean(MisPreferenciasCompartidas.getBoolean("Logi-
                        co",true)).toString();
                        break;
                }
            }
        });
    }
}
```

```
        case 2:
            cadena =new Integer(MisPreferenciasCompartidas.getInt("Entero",11)).
                toString();
            break;
        case 3:
            cadena =new Float(MisPreferenciasCompartidas.getFloat("flotante",(-
                float)3.5)).toString();
            break;
        default:
            cadena ="";
            break;
    }

    Salida.setText(cadena);

}
});

this.btn_Salvar.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        //datos Leidos de La preferencia
        String Datos2 = Datos.getText().toString();
        //el tipo de la preferencia , el item seleccionando
        Integer i = Tipo.getSelectedItemPosition();

        SharedPreferences.Editor editor =MisPreferenciasCompartidas.edit();
        switch(i)
        {
            case 0:
                editor.putString("cadena", Datos2);
                break;
            case 1:
                editor.putBoolean("Logico", Boolean.parseBoolean(Datos2));
                break;
            case 2:
                editor.putInt("Entero", Integer.parseInt(Datos2));
                break;
            case 3:
                editor.putFloat("Flotante", Float.parseFloat(Datos2));
                break;
            default:
                break;
        }

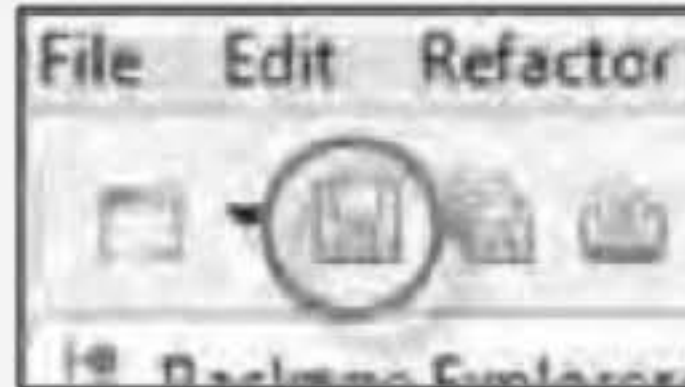
        editor.commit();
    }
});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.activity_main, menu);
    return true;
}
}
```



**EJECUTANDO LA APLICACIÓN**

**Paso 1:** Guarde los cambios realizados, haciendo clic en el ícono



o presionando la combinación <Ctrl+S>.

**Paso 2:** Seleccione el nombre del proyecto en Eclipse. Haga clic derecho, seleccione la opción Run As > Android Application y observe que el emulador se pondrá en marcha:

**5.6 DISPLAYMETRICS**

Devuelve una estructura que describe la información general sobre la pantalla del dispositivo móvil, tal como :

- tamaño
- densidad
- escala de la fuente

**Ejemplo:**

```
DisplayMetrics metrics = new DisplayMetrics();
getWindowManager().getDefaultDisplay().getMetrics(metrics);
```

**Constantes**

int	DENSITY_DEFAULT	La densidad de referencia utilizada en todo el sistema.
int	DENSITY_HIGH	Estándar cuantificado DPI para pantalla de alta resolución
int	DENSITY_LOW	Estándar cuantificado DPI para pantalla de baja resolución
int	DENSITY_MEDIUM	Estándar cuantificado DPI para pantalla de mediana resolución
int	DENSITY_TV	Densidad secundaria
int	DENSITY_XHIGH	Estándar cuantificado DPI para pantallas de extra alta densidad
int	DENSITY_XXHIGH	Estándar cuantificado DPI para pantallas de extra extra alta densidad

**Propiedades**

public float	density	La densidad de la pantalla.
public int	densityDpi	La densidad de la pantalla expresada en pulgadas
<b>public int</b>	<b>heightPixels</b>	<b>La altura de la pantalla en pixeles</b>
public float	scaledDensity	Un factor de escala para fuentes que se muestran en la pantalla.
<b>public int</b>	<b>widthPixels</b>	<b>El ancho de la pantalla en pixeles</b>
public float	xdpi	Los píxeles por pulgada de la pantalla en la dimensión X.
public float	ydpi	Los píxeles por pulgada de la pantalla en la dimensión Y.



Ahora diseñe una aplicación que permita realizar algunos trazos.

**Paso 1:** Después de haber creado la primera aplicación para Android, cree un nuevo proyecto seleccionando *Archivo > Nuevo > Android proyecto*.

**Paso 2:** Ingrese el nombre del proyecto: EjemploTrazo (los pasos de creación de un proyecto ya se detallaron en ejemplos anteriores).

### MainActivity.java

```
package com.nolascov.ejemplo pintado;

import android.app.Activity;
import android.content.Context;
import android.content.res.AssetManager;
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.Bundle;
import android.util.DisplayMetrics;
import android.view.Display;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.Window;
import android.view.WindowManager;
import java.io.InputStream;

public class MainActivity extends Activity {

    protected Display display;
    protected MainView view;
    protected Paint paint;
    protected Path path;
    protected Canvas scr_canvas;
    protected Canvas dib_canvas;
    protected Bitmap scr_bitmap;
    protected Bitmap dib_bitmap;
    protected Bitmap icon_accept;
    protected Bitmap icon_accept_sel;
    protected Bitmap icon_cancel;
    protected Bitmap icon_cancel_sel;

    protected int width;
    protected int height;
    protected int color;
    protected int size;

    protected float brush_radius;
    protected boolean anti_alias;
    protected int[] palette;

    protected MainScreen mainScreen;
    protected PaletteScreen paletteScreen;
    protected Screen currScreen;
```

```
@Override
public void onCreate(Bundle icle) {
    super.onCreate(icle);
    //pantalla sin titulo
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.
    LayoutParams.FLAG_FULLSCREEN);
    // dm objetos que posee ancho , alto , densidad de la pantalla
    DisplayMetrics dm = new DisplayMetrics();
    // rellena el objetos con los datos
    getWindowManager().getDefaultDisplay().getMetrics(dm);

    //obtiene las imagenes desde los recursos
    icon_accept = BitmapFactory.decodeResource(getResources(), R.drawable.button_
    accept);
    icon_accept_sel = BitmapFactory.decodeResource(getResources(), R.drawable.button_
    accept_sel);
    icon_cancel = BitmapFactory.decodeResource(getResources(), R.drawable.button_
    cancel);
    icon_cancel_sel = BitmapFactory.decodeResource(getResources(), R.drawable.button_
    cancel_sel);

    //obteniendo datos para el dm
    //ancho de la pantalla en pixeles
    width = dm.widthPixels;
    //alto de la pantalla en pixeles
    height = dm.heightPixels;
    //radio de la paleta
    brush_radius = 5;
    view = new MainView(this);
    //color negro
    color = Color.BLACK;
    //brocha para pintar
    paint = new Paint();
    //color a la brocha
    paint.setColor(color);
    //guardar puntos
    path = new Path();
    //crear una imagen en 32 bits del ancho y alto de la pantalla , pantalla de fondo
    scr_bitmap = Bitmap.createBitmap(width, height, Bitmap.Config.ARGB_8888);
    //es el objetos para pintar en el bitmap
    scr_canvas = new Canvas(scr_bitmap);
    //carga una paleta de 256 colores
    palette = loadPalette(this, "HEXEN.PAL");
    //instanciar la clase mainscreen
    mainScreen = new MainScreen(this);
    //no tiene uso
    paletteScreen = new PaletteScreen(this);
    //hace referencia a la pantalla actual
    currScreen = mainScreen;
    //muestra la pantalla
    mainScreen.show();

    setContentView(view);
}

@Override
public boolean onTouchEvent(MotionEvent event) {
    return currScreen.onTouchEvent(event);
}
```

```

@Override
    public final boolean onKeyDown(final int keyCode, final KeyEvent event) {
        return currScreen.onKeyDown(keyCode, event);
    }

    //manejo de La paleta de 256 colores
    public static int[] loadPalette(Context context, String name) {
        int palette[] = new int[256];
        //Lista de byte ( flujo de entrada )
        InputStream is = null;
        //tener acceso a la carpeta assets
        AssetManager am = context.getAssets();
        //Leer la paleta de colores
        try {
            //apertura
            is = am.open(name);
            for(int idx=0; idx<256; idx++) {
                //formato RGB 4 byte , alfa - Rojo - Verde - Azul
                palette[idx] = 0xFF000000 | (is.read() << 16) | (is.read() << 8) | is.read();
            }
        } catch(Exception ex) {
        } finally {
            if(is != null) {
                try {
                    is.close();
                } catch(Exception ex) {}
            }
        }
        return palette;
    }
}

```

### MainScreen.java

```

package com.nolascov.ejemplo pintado;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Color;
import android.net.Uri;
import android.provider.MediaStore;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.widget.Toast;

public class MainScreen extends Screen {

    private AlertDialog clearDialog;
    private AlertDialog brushDialog;
    private AlertDialog colorDialog;
    private AlertDialog exitDialog;

```

```
protected Button btClear;
protected Button btColor;
protected Button btBrush;
protected Button btShare;
protected Button btExit;

protected Bitmap icon_clear;
protected Bitmap icon_clear_sel;
protected Bitmap icon_brush;
protected Bitmap icon_brush_sel;
protected Bitmap icon_color;
protected Bitmap icon_color_sel;
protected Bitmap icon_share;
protected Bitmap icon_share_sel;
protected Bitmap icon_exit;
protected Bitmap icon_exit_sel;

protected int toolbar_height;
protected int icon_width;
protected int icon_height;
protected int icon_x;
protected int icon_y;

protected float lastX;
protected float lastY;
protected float currX;
protected float currY;

public MainScreen(Context context) {
    super(context);
    //carga Los iconos en La barra de herramientas
    loadBitmaps();
    //inicializa La barra herramientas
    initToolBox();
    //canvas donde se dibujara
    main.dib_bitmap = Bitmap.createBitmap(main.width, main.height-toolbar_height,
    Bitmap.Config.ARGB_8888);
    //objeto para dibujar
    main.dib_canvas = new Canvas(main.dib_bitmap);
    //borra el canvas con el color blanco
    main.dib_canvas.drawColor(Color.WHITE);
    //La brocha a color negro
    main.paint.setColor(0xFF000000);
}

@Override
protected void show() {
    //pinta color gris
    main.scr_canvas.drawColor(Color.GRAY);
    //dibuja La barra de herramientas
    drawToolBar();
    main.scr_canvas.drawBitmap(main.dib_bitmap, 0, toolbar_height, null);
    main.view.invalidate();
}

@Override
protected boolean onTouchEvent(MotionEvent event) {
    boolean res = false;
    int action = event.getActionMasked();
    if(action == MotionEvent.ACTION_DOWN || action == MotionEvent.ACTION_MOVE || action
    == MotionEvent.ACTION_UP) {
```

```

    res = true;
    float x = event.getX();
    float y = event.getY();
    if(y > toolbar_height ) {
        y -= toolbar_height;
        switch(action) {
            case MotionEvent.ACTION_DOWN:
                currX = x;
                currY = y;
                lastX = currX;
                lastY = currY;
                //dibuja un punto o circulo
                drawPoint(currX, currY);
                //actualizar la pantalla
                main.scr_canvas.drawBitmap(main.dib_bitmap, 0, toolbar_height, null);
                //para repinte la pantalla
                main.view.invalidate();
                break;
            case MotionEvent.ACTION_MOVE:
                currX = x;
                currY = y;
                //dibujo una línea desde el punto actual , al punto anterior
                drawLine();
                //actualizar la pantalla
                main.scr_canvas.drawBitmap(main.dib_bitmap, 0, toolbar_height, null);
                //para repinte la pantlla
                main.view.invalidate();
                lastX = currX;
                lastY = currY;
                break;
        }
    } else {
        // Los botones presionado
        if( btClear.isTouched(x,y) ) {
            btClear.onTouchEvent(event);
        } else
            if( btBrush.isTouched(x,y) ) {
                btBrush.onTouchEvent(event);
            } else
                if( btColor.isTouched(x,y) ) {
                    btColor.onTouchEvent(event);
                } else
                    if( btShare.isTouched(x,y) ) {
                        btShare.onTouchEvent(event);
                    } else
                        if( btExit.isTouched(x,y) ) {
                            btExit.onTouchEvent(event);
                        }
        main.view.invalidate();
    }
}
return res;
}

@Override
protected boolean onKeyDown(int keyCode, KeyEvent event) {
    boolean res = false;
    if(event.getAction() == KeyEvent.ACTION_DOWN) {
        if(keyCode == KeyEvent.KEYCODE_BACK) {
            res = true;
            getExitDialog();
        }
    }
}

```

```
        return res;
    }

    private void loadBitmaps() {
        icon_clear = BitmapFactory.decodeResource(main.getResources(), R.drawable.icon_menu);
        icon_clear_sel = BitmapFactory.decodeResource(main.getResources(), R.drawable.icon_
        menu_sel);
        icon_brush = BitmapFactory.decodeResource(main.getResources(), R.drawable.icon_brush);
        icon_brush_sel = BitmapFactory.decodeResource(main.getResources(), R.drawable.icon_
        brush_sel);
        icon_color = BitmapFactory.decodeResource(main.getResources(), R.drawable.icon_color);
        icon_color_sel = BitmapFactory.decodeResource(main.getResources(), R.drawable.icon_
        color_sel);
        icon_share = BitmapFactory.decodeResource(main.getResources(), R.drawable.icon_share);
        icon_share_sel = BitmapFactory.decodeResource(main.getResources(), R.drawable.icon_
        share_sel);
        icon_exit = BitmapFactory.decodeResource(main.getResources(), R.drawable.icon_tool);
        icon_exit_sel = BitmapFactory.decodeResource(main.getResources(), R.drawable.icon_
        tool_sel);
    }

    private void initToolBox() {
        icon_width = icon_clear.getWidth();
        icon_height = icon_clear.getHeight();
        main.size = icon_height;
        icon_y = 0;
        toolbar_height = icon_height + (icon_height >> 3);
        int sp = (main.width - (icon_width*5)) / 6;
        icon_x = sp;
        sp += icon_width;

        btClear = new Button(main, icon_x , icon_y, icon_clear, icon_clear_sel) {
            @Override
            protected void action() {
                getClearDialog().show();
            }
        };
        icon_x += sp;
        btBrush = new Button(main, icon_x , icon_y, icon_brush, icon_brush_sel) {
            @Override
            protected void action() {
                getBrushDialog().show();
            }
        };
        icon_x += sp;
        btColor = new Button(main, icon_x , icon_y, icon_color, icon_color_sel) {
            @Override
            protected void action() {
                getColorDialog().show();
            }
        };
        icon_x += sp;
        btShare = new Button(main, icon_x , icon_y, icon_share, icon_share_sel) {
            @Override
            protected void action() {
                shareBitmap();
            }
        };
        icon_x += sp;
    }
}
```



```

        btExit = new Button(main, icon_x , icon_y, icon_exit, icon_exit_sel) {
            @Override
            protected void action() {
                getExitDialog().show();
            }
        };
    }

    public AlertDialog getClearDialog() {
        if(!(clearDialog instanceof AlertDialog)) {
            clearDialog = new AlertDialog.Builder(main).create();
            clearDialog.setTitle("Clear Canvas?");
            clearDialog.setButton(AlertDialog.BUTTON_POSITIVE, "Yes", new DialogInterface.
                OnClickListener() {
                    public void onClick(DialogInterface paramDialogInterface, int paramInt) {
                        main.dib_canvas.drawColor(Color.WHITE);
                        main.scr_canvas.drawBitmap(main.dib_bitmap, 0, toolbar_height, null);
                        main.view.invalidate();
                    }
                });
            clearDialog.setButton(AlertDialog.BUTTON_NEGATIVE, "No", (DialogInterface.
                OnClickListener)null);
        }
        return clearDialog;
    }

    public AlertDialog getBrushDialog() {
        if(!(brushDialog instanceof AlertDialog)) {
            brushDialog = new BrushDialog(main);
        }
        return brushDialog;
    }

    public AlertDialog getColorDialog() {
        if(!(colorDialog instanceof AlertDialog)) {
            colorDialog = new ColorDialog(main);
        }
        return colorDialog;
    }

    public AlertDialog getExitDialog() {
        if(!(exitDialog instanceof AlertDialog)) {
            exitDialog = new AlertDialog.Builder(main).create();
            exitDialog.setTitle("Exit MicroPaint?");
            exitDialog.setButton(AlertDialog.BUTTON_POSITIVE, "Yes", new DialogInterface.
                OnClickListener() {
                    public void onClick(DialogInterface paramDialogInterface, int paramInt) {
                        SplashActivity.killed = true;
                        main.finish();
                    }
                });
            exitDialog.setButton(AlertDialog.BUTTON_NEGATIVE, "No", (DialogInterface.OnClickListener)null);
        }
        return exitDialog;
    }

    private void drawToolBar() {
        main.paint.setColor(0xFF202020);
        main.scr_canvas.drawRect(0, 0, main.width, main.size, main.paint);
        main.paint.setColor(main.color);
        btClear.paint(false);
        btColor.paint(false);
    }

```

```
        btBrush.paint(false);
        btShare.paint(false);
        btExit.paint(false);
    }

    private void shareBitmap() {
        try {
            String fpath = MediaStore.Images.Media.insertImage(main.getContentResolver(),
                main.dib_bitmap, "dib_bitmap", null);
            Uri screenshotUri = Uri.parse(fpath);
            final Intent emailIntent = new Intent(android.content.Intent.ACTION_SEND);
            emailIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            emailIntent.putExtra(Intent.EXTRA_STREAM, screenshotUri);
            emailIntent.setType("image/png");
            main.startActivity(Intent.createChooser(emailIntent, "Share image..."));
        } catch (Exception ex) {
            Toast.makeText(main, ex.getMessage(), Toast.LENGTH_LONG).show();
        }
    }

    private void drawLine() {
        float lenX = currX - lastX;
        float lenY = currY - lastY;
        float vx = 0;
        if(lenX > 0) {
            vx = 1;
        } else if(lenX < 0) {
            vx = -1;
        }
        float vy = 0;
        if(lenY > 0) {
            vy = 1;
        } else if(lenY < 0) {
            vy = -1;
        }
        float len;
        lenX = Math.abs(lenX);
        lenY = Math.abs(lenY);
        if(lenX > lenY) {
            vy = (lenY / lenX) * vy;
            len = lenX;
        } else {
            vx = (lenX / lenY) * vx;
            len = lenY;
        }
        float x = lastX;
        float y = lastY;
        for(int i=0; i < len; i++) {
            drawPoint((int)x, (int)y);
            x += vx;
            y += vy;
        }
    }

    private void drawPoint(final float x, final float y) {
        main.dib_canvas.drawCircle(x, y, main.brush_radius, main.paint);
    }
}
```

## BrushDialog.java

```
package com.nolascov.ejemplo pintado;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.view.View;
import android.widget.CheckBox;
import android.widget.LinearLayout;
import android.widget.LinearLayout.LayoutParams;
import android.widget.SeekBar;
import android.widget.TextView;

public class BrushDialog extends AlertDialog implements SeekBar.OnSeekBarChangeListener,
DialogInterface.OnClickListener, View.OnClickListener {

    private MainActivity main;
    private TextView text;
    private CheckBox check;
    private int brush_radius;

    public BrushDialog(Context context) {
        super(context);
        main = (MainActivity)context;
        brush_radius = (int)main.brush_radius;
        LayoutParams params = new LayoutParams(LayoutParams.MATCH_PARENT, LayoutParams.
MATCH_PARENT);
        LinearLayout layout = new LinearLayout(main);
        layout.setOrientation(LinearLayout.VERTICAL);
        layout.setPadding(8,8,8,8);

        text = new TextView(main);
        text.setText("size: "+(brush_radius << 1));
        layout.addView(text);
        SeekBar seekbar = new SeekBar(main);
        seekbar.setLayoutParams(params);
        seekbar.setOnSeekBarChangeListener(this);
        seekbar.setMax(100);
        seekbar.setProgress(brush_radius << 1);
        layout.addView(seekbar);

        check = new CheckBox(main);
        check.setText("Antialias");
        check.setChecked(main.anti_alias);
        check.setOnClickListener(this);
        layout.addView(check);

        setView(layout);
        setTitle("Brush Settings");
        setIcon(0);
        setButton(DialogInterface.BUTTON_POSITIVE, "Accept", this);
        setButton(DialogInterface.BUTTON_NEGATIVE, "Cancel", this);
    }

    @Override
    protected void onStart() {
        super.onStart();
        brush_radius = (int)main.brush_radius;
    }
}
```

```

        LayoutParams params = new LayoutParams(LayoutParams.MATCH_PARENT, LayoutParams.
        MATCH_PARENT);
        LinearLayout layout = new LinearLayout(main);
        layout.setOrientation(LinearLayout.VERTICAL);
        layout.setPadding(8,8,8,8);

        text = new TextView(main);
        text.setText("size: "+(brush_radius << 1));
        layout.addView(text);
        SeekBar seekbar = new SeekBar(main);
        seekbar.setLayoutParams(params);
        seekbar.setOnSeekBarChangeListener(this);
        seekbar.setMax(100);
        seekbar.setProgress(brush_radius << 1);
        layout.addView(seekbar);
    }

    public void onProgressChanged(SeekBar seekbar, int progress, boolean user) {
        brush_radius = progress >> 1;
        text.setText("size: "+progress);
    }

    public void onStartTrackingTouch(SeekBar seekbar) {}

    public void onStopTrackingTouch(SeekBar seekbar) {}

    public void onClick(DialogInterface dialog, int button) {
        if( button == DialogInterface.BUTTON_POSITIVE ) {
            main.brush_radius = brush_radius;
        }
    }

    public void onClick(View arg0) {
        if(main.anti_alias) {
            main.anti_alias = false;
        } else {
            main.anti_alias = true;
        }
        main.paint.setAntiAlias(main.anti_alias);
    }
}

```

### Button.java

```

package com.nolascov.ejemplo pintado;

import android.graphics.Bitmap;
import android.view.MotionEvent;

public class Button {

    protected float x;
    protected float y;
    protected float xx;
    protected float yy;
    protected int width;
    protected int height;
    protected Bitmap icon;
    protected Bitmap icon_sel;
}

```

```
protected float x1;
protected float y1;
protected float x2;
protected float y2;
private MainActivity main;

public Button(final MainActivity main, final float x, final float y, final Bitmap icon, final
Bitmap icon_sel) {
    this.main = main;
    this.x = x;
    this.y = y;
    width = icon.getWidth();
    height = icon.getHeight();
    xx = (x+width)-1;
    yy = (y+height)-1;
    this.icon = icon;
    this.icon_sel = icon_sel;
    x1 = x+2;
    y1 = y+2;
    x2 = xx-2;
    y2 = yy-2;
}

protected final boolean isTouched(final MotionEvent event) {
    return isTouched(event.getX(), event.getY());
}

protected final boolean isTouched(final float x, final float y) {
    return (x1 < x) && (x < x2) && (y1 < y) && (y < y2);
}

protected final boolean onTouchEvent(final MotionEvent event) {
    boolean res = false;
    int action = event.getActionMasked();
    if(action == MotionEvent.ACTION_DOWN) {
        paint(true);
        res = true;
    } else if (action == MotionEvent.ACTION_UP) {
        paint(false);
        res = true;
        action();
    }
    return res;
}

protected final void paint(final boolean hover) {
    if(hover) {
        main.scr_canvas.drawBitmap(icon_sel, x, y, null);
    } else {
        main.scr_canvas.drawBitmap(icon, x, y, null);
    }
    main.view.invalidate();
}

protected void action() {}
```

## ColorDialog.java

```
package com.nolascov.ejemplo pintado;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.graphics.Color;
import android.widget.LinearLayout;
import android.widget.LinearLayout.LayoutParams;
import android.widget.SeekBar;
import android.widget.TextView;

public class ColorDialog extends AlertDialog implements SeekBar.OnSeekBarChangeListener,
DialogInterface.OnClickListener {

    private MainActivity main;
    private TextView text_color;
    private TextView text_red;
    private TextView text_green;
    private TextView text_blue;
    private TextView text_alpha;
    private SeekBar bar_red;
    private SeekBar bar_green;
    private SeekBar bar_blue;
    private SeekBar bar_alpha;
    private int color;
    private int red;
    private int green;
    private int blue;
    private int alpha;

    public ColorDialog(Context context) {
        super(context);
        main = (MainActivity)context;
        LayoutParams params = new LayoutParams(LayoutParams.MATCH_PARENT, LayoutParams.
MATCH_PARENT);
        LinearLayout layout = new LinearLayout(main);
        layout.setOrientation(LinearLayout.VERTICAL);
        layout.setPadding(8,8,8,8);

        text_color = new TextView(main);
        text_color.setText(" ");
        layout.addView(text_color);

        text_red = new TextView(main);
        layout.addView(text_red);
        bar_red = new SeekBar(main);
        bar_red.setId(Color.RED);
        bar_red.setLayoutParams(params);
        bar_red.setOnSeekBarChangeListener(this);
        bar_red.setMax(255);
        layout.addView(bar_red);

        text_green = new TextView(main);
        layout.addView(text_green);
        bar_green = new SeekBar(main);
        bar_green.setId(Color.GREEN);
        bar_green.setLayoutParams(params);
        bar_green.setOnSeekBarChangeListener(this);
        bar_green.setMax(255);
```

```
        layout.addView(bar_green);

        text_blue = new TextView(main);
        layout.addView(text_blue);
        bar_blue = new SeekBar(main);
        bar_blue.setId(Color.BLUE);
        bar_blue.setLayoutParams(params);
        bar_blue.setOnSeekBarChangeListener(this);
        bar_blue.setMax(255);
        layout.addView(bar_blue);

        text_alpha = new TextView(main);
        layout.addView(text_alpha);
        bar_alpha = new SeekBar(main);
        bar_alpha.setId(Color.GRAY);
        bar_alpha.setLayoutParams(params);
        bar_alpha.setOnSeekBarChangeListener(this);
        bar_alpha.setMax(255);
        layout.addView(bar_alpha);

        setContentView(layout);
        setTitle("Color Settings");
        setIcon(0);
        setButton(DialogInterface.BUTTON_POSITIVE, "Accept", this);
        setButton(DialogInterface.BUTTON_NEGATIVE, "Cancel", this);
    }

    @Override
    protected void onStart() {
        super.onStart();
        color = main.color;
        red = Color.red(color);
        green = Color.green(color);
        blue = Color.blue(color);
        alpha = Color.alpha(color);

        text_color.setBackgroundColor(color);

        text_red.setText("red: "+red);
        bar_red.setBackgroundColor(Color.rgb(red,0,0));
        bar_red.setProgress(red);

        text_green.setText("green: "+green);
        bar_green.setBackgroundColor(Color.rgb(0,green,0));
        bar_green.setProgress(green);

        text_blue.setText("blue: "+blue);
        bar_blue.setBackgroundColor(Color.rgb(0,0,blue));
        bar_blue.setProgress(blue);

        text_alpha.setText("alpha: "+alpha);
        bar_alpha.setBackgroundColor(Color.rgb(alpha,alpha,alpha));
        bar_alpha.setProgress(alpha);
    }

    public void onProgressChanged(SeekBar seekbar, int progress, boolean user) {
        switch(seekbar.getId()) {
            case Color.RED:
                red = progress;
                bar_red.setBackgroundColor(Color.rgb(red,0,0));
                text_red.setText("red: "+red);
```

```

        break;
    case Color.GREEN:
        green = progress;
        bar_green.setBackgroundColor(Color.rgb(0,green,0));
        text_green.setText("green: "+green);
        break;
    case Color.BLUE:
        blue = progress;
        bar_blue.setBackgroundColor(Color.rgb(0,0,blue));
        text_blue.setText("blue: "+blue);
        break;
    case Color.GRAY:
        alpha = progress;
        bar_alpha.setBackgroundColor(Color.rgb(alpha,alpha,alpha));
        text_alpha.setText("alpha: "+alpha);
        break;
    }
    color = Color.argb(alpha, red, green, blue);
    text_color.setBackgroundColor(color);
}

public void onStartTrackingTouch(SeekBar seekbar) {
}

public void onStopTrackingTouch(SeekBar seekbar) {
}

public void onClick(DialogInterface dialog, int button) {
    if( button == DialogInterface.BUTTON_POSITIVE ) {
        main.color = color;
        main.paint.setColor(main.color);
    }
}
}
}

```

### MainView.java

```

package com.nolascov.ejemplo pintado;

import android.content.Context;
import android.graphics.Canvas;
import android.view.View;

public class MainView extends View {

    private MainActivity main;

    public MainView(Context context) {
        super(context);
        main = (MainActivity)context;
    }

    @Override
    protected void onDraw(Canvas canvas) {
        canvas.drawBitmap(main.scr_bitmap, 0, 0, null);
    }
}

```



**PaletteScreen.java**

```
package com.nolascov.ejemplo pintado;

import android.content.Context;
import android.graphics.Color;
import android.view.KeyEvent;
import android.view.MotionEvent;

public class PaletteScreen extends Screen {

    private Button btAccept;
    private Button btCancel;

    private int box;
    private int palX1;
    private int palY1;
    private int palX2;
    private int palY2;
    private int palWdt;
    private int palHgt;
    private int color;

    public PaletteScreen(Context context) {
        super(context);
        box = main.size >> 2;
        palWdt = box << 4;
        palHgt = palWdt;
        palX1 = (main.width - palWdt) >> 1;
        palY1 = main.size+box;
        palX2 = palX1+palWdt;
        palY2 = palY1+palHgt;
        color = main.color;

        btAccept = new Button(main, 0 , main.height-main.icon_accept.getHeight(), main.
            icon_accept, main.icon_accept_sel) {
            @Override
            protected void action() {
                main.color = color;
                goback();
            }
        };
        btCancel = new Button(main, main.width-main.icon_cancel.getWidth() , main.height-
            main.icon_cancel.getHeight(), main.icon_cancel, main.icon_cancel_sel) {
            @Override
            protected void action() {
                goback();
            }
        };
    }

    @Override
    protected void show() {
        main.scr_canvas.drawColor(Color.DKGRAY);
        drawColorBox();
        drawPalette();
        btAccept.paint(false);
        btCancel.paint(false);
        main.view.invalidate();
    }
}
```

```

private void drawColorBox() {
    main.paint.setColor(color);
    main.scr_canvas.drawRect(0, 0, main.width, main.size, main.paint);
}

private void drawPalette() {
    int col = 0;
    for(int i=0; i<16; i++) {
        for(int j=0; j<16; j++) {
            main.paint.setColor(main.palette[col]);
            col++;
            main.scr_canvas.drawRect((j*box)+palX1, (i*box)+palY1, (j*box)+box+palX1,
                (i*box)+box+palY1, main.paint);
        }
    }
}

@Override
protected final boolean onTouchEvent(final MotionEvent event) {
    boolean res = false;
    int action = event.getActionMasked();
    if(action == MotionEvent.ACTION_DOWN || action == MotionEvent.ACTION_MOVE ||
        action == MotionEvent.ACTION_UP) {
        res = true;
        float x = event.getX();
        float y = event.getY();
        if((x > palX1) && (y > palY1) && (x < palX2) && (y < palY2)) {
            int px = (int)((x-palX1)/box);
            int py = (int)((y-palY1)/box);
            color = main.palette[(py << 4) | px];
            drawColorBox();
            main.view.invalidate();
        } else {
            if( btAccept.isTouched(x,y) ) {
                btAccept.onTouchEvent(event);
            } else
                if( btCancel.isTouched(x,y) ) {
                    btCancel.onTouchEvent(event);
                }
        }
    }
    return res;
}

@Override
protected boolean onKeyDown(int keyCode, KeyEvent event) {
    boolean res = false;
    if(event.getAction() == KeyEvent.ACTION_DOWN) {
        if(keyCode == KeyEvent.KEYCODE_BACK) {
            res = true;
            goback();
        }
    }
    return res;
}

private void goback() {
    main.paint.setColor(main.color);
    main.currScreen = main.mainScreen;
    main.mainScreen.show();
}
}

```

## Screen.java

```
package com.nolascov.ejemplo pintado;

import android.content.Context;
import android.view.KeyEvent;
import android.view.MotionEvent;

public class Screen {

    protected MainActivity main;

    public Screen(Context context) {
        main = (MainActivity)context;
    }

    protected boolean onTouchEvent(final MotionEvent event) {
        return false;
    }

    protected boolean onKeyDown(final int keyCode, final KeyEvent event) {
        return false;
    }

    protected void show() {}

}
```

## SplashActivity.java

```
package com.nolascov.ejemplo pintado;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Window;
import android.view.WindowManager;
import android.widget.ImageView;
import java.util.Timer;
import java.util.TimerTask;

public class SplashActivity extends Activity {

    protected static boolean killed;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //desactivar título
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        //mostrar en toda la pantalla
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.
        LayoutParams.FLAG_FULLSCREEN);
        //la imagen a mostrar inicialmente
        ImageView image = new ImageView(this);
        //código imagen
    }
}
```

```

        image.setImageResource(R.drawable.logo);
        //colocar la imagen por defecto
        setContentView(image);
    }

    @Override
    protected void onStart() {
        super.onStart();
        //preguntar salida completa de la app
        if(killed) {
            killed = false;
            //cerrar la actividad
            finish();
        } else {
            //timer de 2 segundos
            new Timer().schedule(new TimerTask() {
                @Override
                public void run() {
                    Intent intent = new Intent(SplashActivity.this, MainActivity.class);
                    // no borrar la actividad anterior
                    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(intent);
                }
            }, 4000);
        }
    }
}

```

### EJECUTANDO LA APLICACIÓN

**Paso 1:** Seleccione el nombre del proyecto en Eclipse. Haga clic derecho, seleccione la opción Run As > Android Application y observe que el emulador se pondrá en marcha:



Puede descargar la aplicación desde el siguiente link:

<https://play.google.com/store/search?q=lnkadroid&c=apps>