INSTITUTO TECNOLÓGICO DE ORIZABA

8gFA – Tecnologías de Programación Emergentes para la Web - Manuel Panzi Utrera
21011009 - Muñoz Hernández Vania Lizeth

REPORTE DE PRÁCTICA 05: MANEJO DE PRIMITIVAS EN ANDROID

Durante las sesiones, se desarrolló una aplicación en Android con el objetivo de aprender a implementar diversos métodos que sirvan para dibujar gráficos en pantalla sobre un lienzo que estará asociado a una vista.

Una de las partes más complicadas fue entender cómo mostrar los botones definidos en el archivo XML (activity_main.xml), y al mismo tiempo utilizar la clase Vista anidada para manejar el dibujo en la pantalla. Al principio, no sabía cómo combinar ambos elementos, pero después logré entender que era posible y necesario que se le asignase al LinearLayout un ID, como es posible con el resto de elementos gráficos.

Una vez que el layout principal tenía un ID, pude crear una instancia de LinearLayout en el código y enlazarla con el XML usando findViewById. Luego, agregué la vista personalizada (Vista – clase anidada) al layout principal con el método addView.

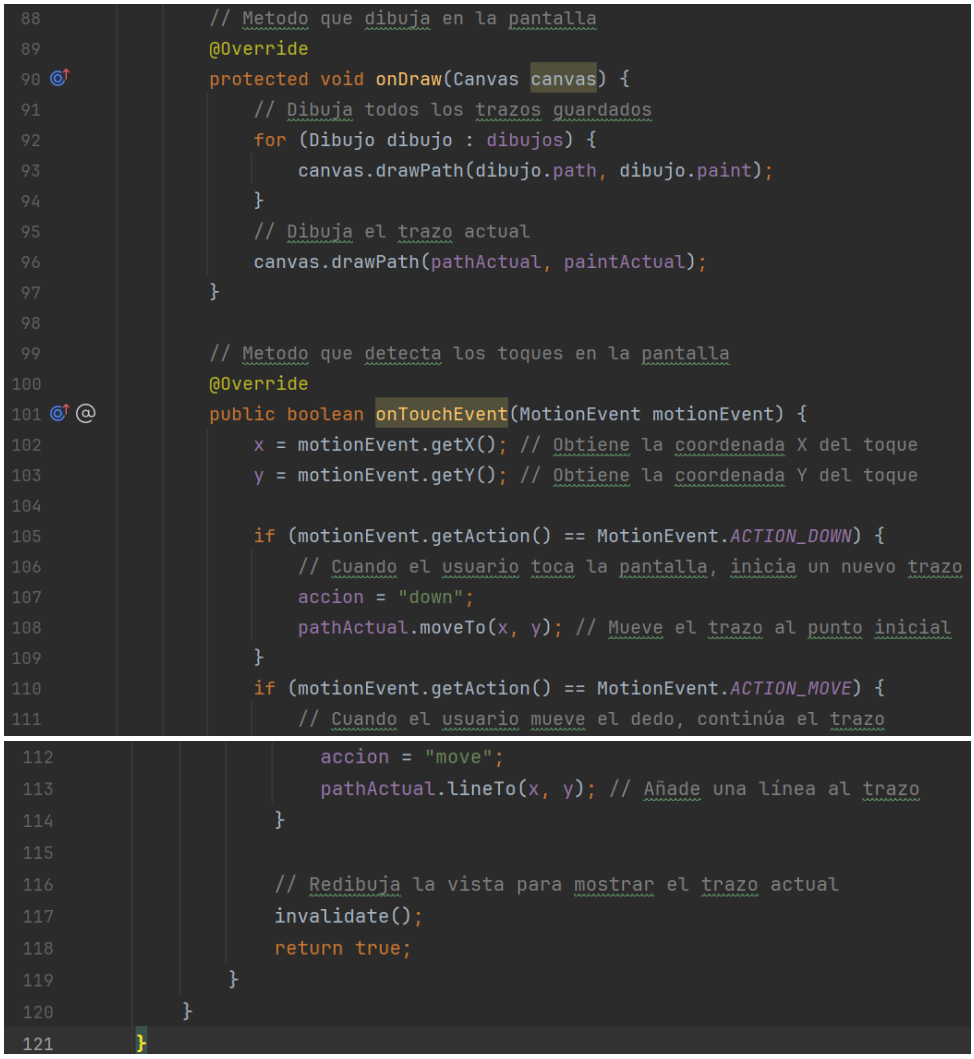
```
23      @Override
24      protected void onCreate(Bundle savedInstanceState) {
25          super.onCreate(savedInstanceState);
26          setContentView(R.layout.activity_main); // Carga el diseño del XML
27
28          //Se estableció un id para el Layout en el XML, por lo que puede crearse una instancia de LinearLayout
29          // y enlazarlo con la vista/XML, de esta forma es posible agregar al layout principal
30          // la vista que estamos creando, donde se dibuja
31          LinearLayout layout = findViewById(R.id.layoutPrincipal);
32          vista = new Vista( context: this);
33          layout.addView(vista);
```

También fue necesario utilizar más instancias, primero con el pathActual (indica el trazo que se está dibujando actualmente), cuando el usuario toca la pantalla, se inicia un nuevo Path, y al mover el dedo, se van añadiendo puntos al Path.

El paintActual, es el pincel actual y sus propiedades (fue necesario para que, en el cambio de colores, no se cambiase todo). Por último, el arreglo se utiliza para almacenar todos los trazos que el usuario ha dibujado, junto con su color correspondiente. Cuando el usuario cambia de color, el trazo actual (pathActual) y el pincel actual (paintActual) se guardan en la lista dibujos, luego, se crea un nuevo Path y un nuevo Paint para el siguiente trazo.

```
46          // Clase interna para la vista personalizada
47          class Vista extends View {
                3 usages
48              float x, y; // Coordenadas del toque
                2 usages
49              String accion = "accion"; // Acción actual (down o move)
                5 usages
50              Path pathActual = new Path(); // Trazo actual que se está dibujando
                8 usages
51              Paint paintActual = new Paint(); // Pincel actual con el que se dibuja
52
53              // Clase para guardar un trazo y su color
                3 usages
54              class Dibujo {
                    2 usages
55                  Path path; // El camino del trazo
                    2 usages
56                  Paint paint; // Pincel
57
                    1 usage
58                  Dibujo(Path path, Paint paint) {
59                      this.path = path;
60                      this.paint = new Paint(paint); // Crea una copia del pincel
61                  }
62              }
```

```
64              // Lista para guardar todos los trazos dibujados
                2 usages
65              ArrayList<Dibujo> dibujos = new ArrayList<>();
66              // Constructor de la vista
67              public Vista(Context context) {
68                  super(context);
69                  // Configura el pincel inicial
70                  paintActual.setStyle(Paint.Style.STROKE);
71                  paintActual.setStrokeWidth(4);
72                  paintActual.setColor(Color.BLUE);
73              }
74
75              // Metodo para cambiar el color del pincel
                3 usages
76              public void cambiarColor(int nuevoColor) {
77                  // Guarda el trazo actual en la lista de dibujos
78                  dibujos.add(new Dibujo(pathActual, paintActual));
79
80                  // Crea un nuevo trazo y un nuevo pincel con el color seleccionado
81                  pathActual = new Path();
82                  paintActual = new Paint(paintActual);
83                  paintActual.setColor(nuevoColor);
84                  // Redibuja la vista para mostrar los cambios
85                  invalidate();
86              }
```

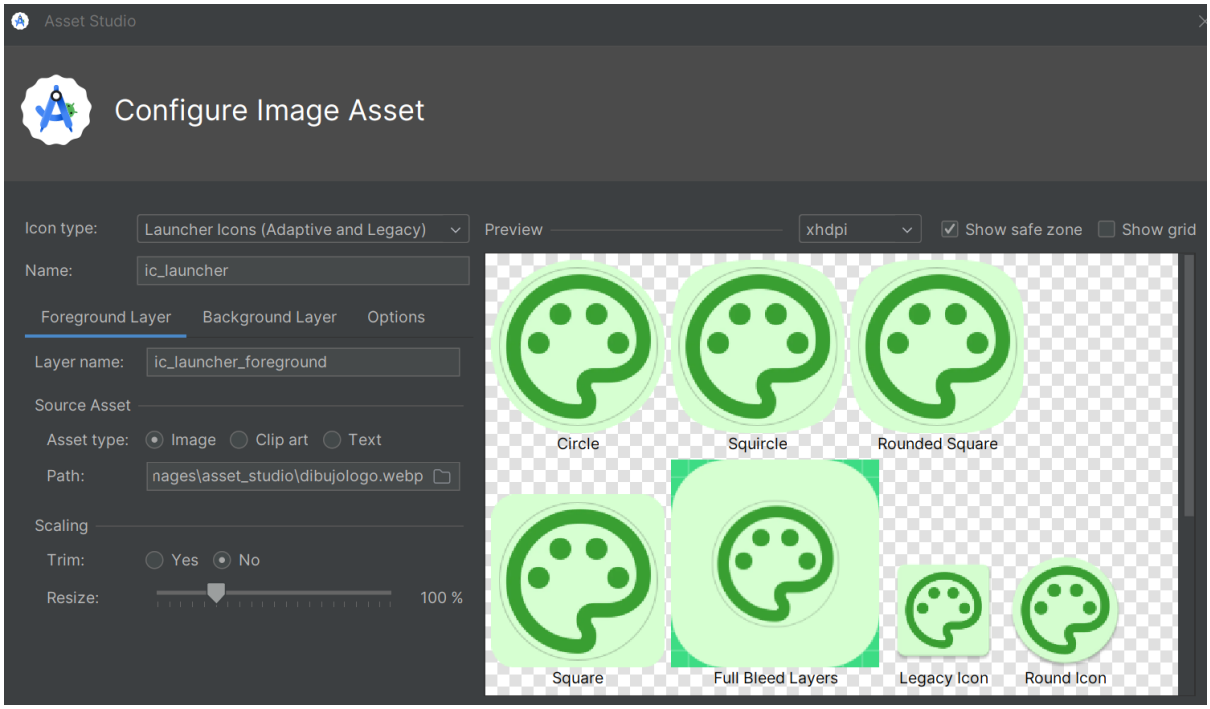


```
88          // Metodo que dibuja en la pantalla
89          @Override
90          protected void onDraw(Canvas canvas) {
91              // Dibuja todos los trazos guardados
92              for (Dibujo dibujo : dibujos) {
93                  canvas.drawPath(dibujo.path, dibujo.paint);
94              }
95              // Dibuja el trazo actual
96              canvas.drawPath(pathActual, paintActual);
97          }
98
99          // Metodo que detecta los toques en la pantalla
100         @Override
101         public boolean onTouchEvent(MotionEvent motionEvent) {
102             x = motionEvent.getX(); // Obtiene la coordenada X del toque
103             y = motionEvent.getY(); // Obtiene la coordenada Y del toque
104
105             if (motionEvent.getAction() == MotionEvent.ACTION_DOWN) {
106                 // Cuando el usuario toca la pantalla, inicia un nuevo trazo
107                 accion = "down";
108                 pathActual.moveTo(x, y); // Mueve el trazo al punto inicial
109             }
110             if (motionEvent.getAction() == MotionEvent.ACTION_MOVE) {
111                 // Cuando el usuario mueve el dedo, continúa el trazo
112                 accion = "move";
113                 pathActual.lineTo(x, y); // Añade una línea al trazo
114             }
115
116             // Redibuja la vista para mostrar el trazo actual
117             invalidate();
118             return true;
119         }
120     }
121 }
```

Además, cambie el nombre de la app, simplemente modificando el archivo string.xml:

```
<resources>
   <string name="app_name">Dibujo</string>
</resources>
```

Por último, se cambió también el logo de la aplicación, al siguiente:



EJECUCIÓN: