

Lab 9. Canvas and MultiThreading (4 tiết)

I. Yêu cầu

- Sinh viên sử dụng công cụ Android Studio để làm bài. Kết quả bài làm cần được chụp lại và giữ lại toàn bộ dự án để sử dụng khi cần thiết.
- Mỗi người làm trên một dự án khác nhau.
- Khi có yêu cầu, sinh viên nộp qua email giáo viên hoặc một kênh khác.

II. Luyện tập

1. Canvas

Canvas là một bề mặt mà chúng ta có thể vẽ bất cứ thứ gì lên đó. Ví dụ như vẽ một điểm, đường thẳng, hình chữ nhật, đường tròn, elip, văn bản, hay thậm chí là một hình ảnh và các hình ảnh phức tạp khác.

Để vẽ được Canvas, ta tạo một lớp kế thừa từ lớp View và vẽ các đối tượng sử dụng hàm `onDraw(canvas)`:

```
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    //Draw component in here
}
```

Một số phương thức cơ bản phục vụ cho vẽ các đối tượng:

- **`setColor(int color)`**: Thiết lập màu cho nét vẽ.
- **`setStrokeWidth(float width)`**: Thiết lập giá trị độ rộng của nét vẽ.
- **`setStyle(Style style)`**: thiết lập kiểu cho nét vẽ, các giá trị như sau: `Paint.Style.FILL`: Tô đối tượng (hình tròn, elip, oval); `Paint.Style.STROKE`: Vẽ đường; `Paint.Style.FILL_AND_STROKE`: Kiểu vừa vẽ vừa tô.
- **`setStrokeCap(Cap cap)`**: Thiết lập style vẽ ở những điểm kết thúc của hai đường thẳng và có những giá trị sau: `Cap.ROUND` là bo tròn nét vẽ ở hai đầu mút của đoạn thẳng; `Cap.SQUARE` là vẽ nét vẽ bình thường.
- **`setTypeface(Typeface typeface)`**: Thiết lập kiểu phong cho nét vẽ, sử dụng trong trường hợp chúng ta muốn vẽ văn bản.
- **`setTextSize(float textSize)`**: Thiết lập font size cho nét vẽ, sử dụng trong trường hợp vẽ văn bản.

Sinh viên theo các bước như sau để thực hiện Canvas:

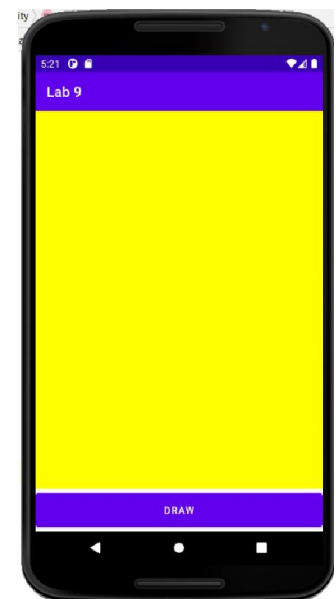
Bước 1. Tạo một lớp có tên MyCanvas, kế thừa từ lớp View như sau:

```
public class MyCanvas extends View {
    public static int color = Color.YELLOW;
    public MyCanvas(Context context) {
        super(context, attrs: null);
    }
    public MyCanvas(Context context, AttributeSet attrs) {
        super(context, attrs, defStyleAttr: 0);
    }
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        // Vẽ nền vàng
        canvas.drawColor(color);
    }
}
```

Bước 2. Trong activity_main.xml, thiết kế MyCanVas và Button trong LinearLayout như sau:

```
<LinearLayout android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:weightSum="10"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <com.example.lab9.MyCanvas
        android:id="@+id/myCanvas"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="9"/>
    <Button
        android:id="@+id/btnDraw"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:text="Draw"
        android:layout_weight="1"/>
</LinearLayout>
```

Chạy để xem kết quả:



Bước 3. Để thêm phần sinh động, trong phần MainActivity.Java, chúng ta thêm các đối tượng như sau trên đầu lớp:

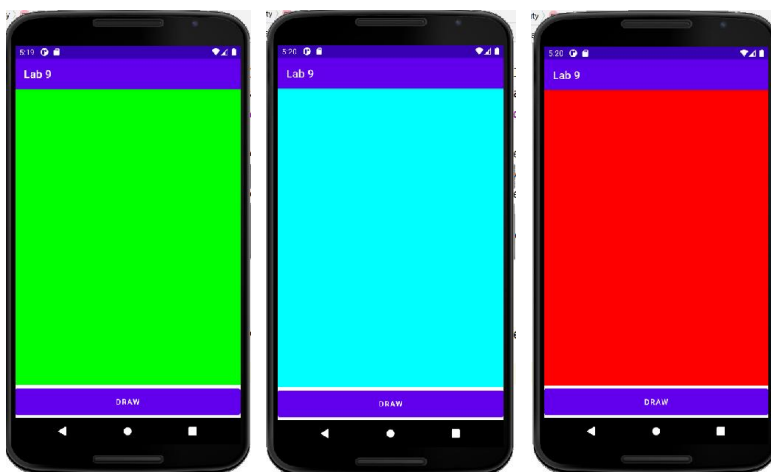
```
MyCanvas canvas;
Button button;
int[] colors = new int[]{Color.RED, Color.GREEN, Color.GRAY,
                        Color.YELLOW, Color.CYAN, Color.WHITE};
Random rd = new Random();
```

Trong hàm onCreate, viết đoạn mã sau đây:

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

    MyCanvas canvas = findViewById(R.id.myCanvas);
    Button btnDraw = findViewById(R.id.btnDraw);
    btnDraw.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // Random chỉ số màu
            int colorIndex = rd.nextInt(colors.length);
            // Lấy màu trong mảng và gán cho giá trị màu trong Lớp MyCanvas
            MyCanvas.color = colors[colorIndex];
            canvas.invalidate(); // Vẽ Lại
        }
    });
}
```

Bước 4. Chạy và bấm nút Draw để xem các màu được hiển thị Random:



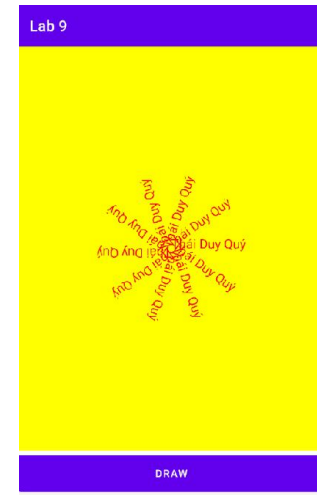
Một số lệnh cơ bản:

- Vẽ chữ

```
private void drawText(Canvas canvas, String mMessage){
    int viewWidth = getWidth();
    int viewHeight = getHeight();
    canvas.translate( dx: viewWidth/2, dy: viewHeight/2);

    Paint mPaint = new Paint();
    mPaint.setColor(Color.RED);
    mPaint.setTextSize(60f);

    for(int i=0; i<10; i++) {
        canvas.drawText(mMessage, x: 0, y: 0, mPaint);
        canvas.rotate( degrees: 36);
    }
}
```



Gọi hàm trong onDraw:

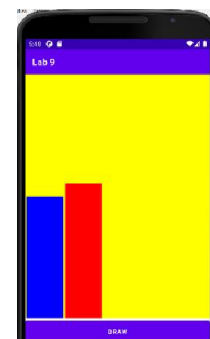
```
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    canvas.drawColor(color);
    // Vẽ chữ
    drawText(canvas, mMessage: "Thái Duy Quý");
}
```

- Vẽ hình chữ nhật thẳng đứng

```
private void drawRect(Canvas canvas){
    int pos = 10;
    int viewWidth = getWidth();
    int viewHeight = getHeight();
    Paint mPaint = new Paint();
    mPaint.setColor(Color.BLUE);
    canvas.drawRect(pos, top: viewHeight/2, right: pos+viewWidth/5,viewHeight,mPaint);
    mPaint.setColor(Color.RED);
    canvas.drawRect( left: pos*2+viewWidth/5, top: viewHeight/2-100,
        right: (pos+viewWidth/5)*2,viewHeight,mPaint);
}
```

Kết quả gọi và chạy trên màn hình:

```
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    canvas.drawColor(color);
    // Vẽ hình chữ nhật
    drawRect(canvas);
}
```



2. Multithreading

Multithreading là kỹ thuật lập trình đa luồng (đa tiến trình trong Android). Đây là một kỹ thuật giúp cho các chương trình chạy song song, truyền các tham số và hiển thị lên giao diện.

Chi tiết, sinh viên tham khảo ở đây:

<https://duythanhcse.wordpress.com/2013/06/14/bai-tap-34-da-tien-trinh-trong-android-multi-threading/>

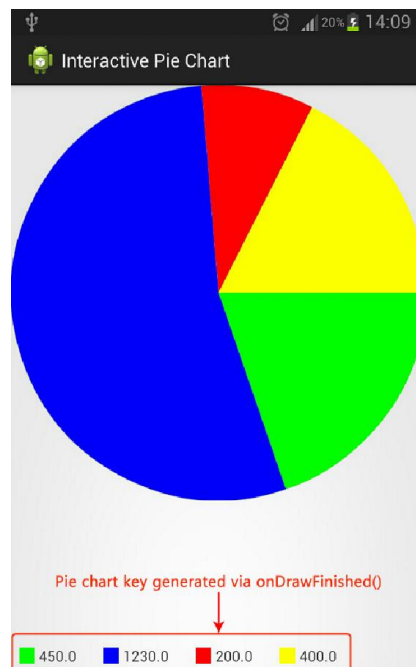
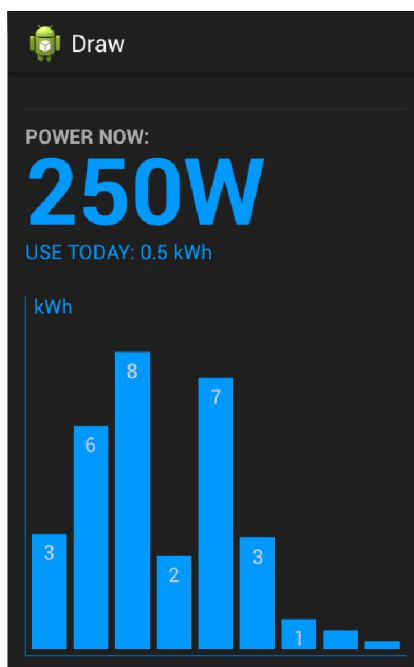
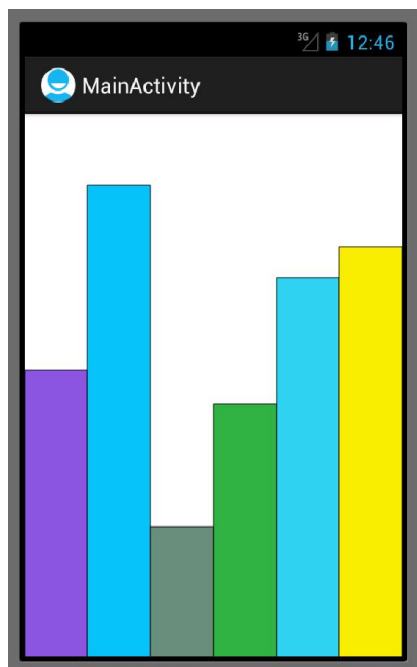
<https://viblo.asia/p/multi-thread-va-service-trong-android-oaKYMNeBR83E>

<https://www.youtube.com/watch?v=sZ1AYapviOE>

III. Bài tập

Bài tập 1:

Vẽ các biểu đồ sau sử dụng Canvas trong Android:



Lưu ý: Nên cho người dùng thay đổi các tham số và có nút Draw để vẽ.

Bài tập 2:

Xem và thực hiện lại Game Cuộc đua kỳ thú theo hướng dẫn tại link sau:

<https://www.youtube.com/watch?v=7MWDiCStLPQ&index=34&list=PLzrVYRaiOriSRJ3M3bifVWWRq5eJMu6tv>

**Một số bài tập làm thêm (Bonus)*:**

1. Tìm hiểu lập trình Animation Sprite thông qua ví dụ cho ở liên kết phía dưới:

<http://gamecodeschool.com/android/coding-android-sprite-sheet-animations/>

2. Flappy bird: <https://www.youtube.com/watch?v=3BLESx44GL4>

3. Android 2D game:

<https://o7planning.org/en/10521/android-2d-game-tutorial-for-beginners>

4. Space Fighter:

<https://www.simplifiedcoding.net/android-game-development-tutorial-1/>

--Hết--