

MODULE 05

SÉANCE SYSTÈME 07

TP D'INFORMATIQUE

Durée 2h30

ANDROID : AFFICHAGE DES DONNÉES MOTEUR

BLOC DE COMPÉTENCES

U5 - EXPLOITATION ET MAINTENANCE DE RÉSEAUX INFORMATIQUES

COMPÉTENCE(S)

C06 - VALIDER UN SYSTÈME INFORMATIQUE

OBJECTIFS PÉDAGOGIQUES

Affichage des données moteur : vitesse et régime. Intégration d'une classe JAVA permettant de tracer un cercle et une aiguille.

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

- | | |
|--|----------|
| • Maîtrise des environnements de développement, d'intégration, de déploiement logiciel et des versions logicielles associées | Niveau 2 |
| • Langages de développement, de description, de création d'API et les IDE associés | Niveau 3 |
| • Tests unitaires et d'intégration | Niveau 3 |

CONNAISSANCES OPÉRATIONNALISÉES

- | | |
|--|----------|
| • Concevoir une interface graphique sous Android : gestion des événements java et communication en C++ natif | Niveau 3 |
| • Versionner un code | Niveau 3 |

TP

Intégration d'une classe JAVA

Intégrer la classe `CircleView` :

```
package com.example.nomProjet;
import static java.lang.Math.cos;
import static java.lang.Math.sin;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.util.AttributeSet;
import android.util.Log;
import android.view.View;
public class CircleView extends View {
    private Paint circlePaint;
    private Paint linePaint;
    private int angle;
    public CircleView(Context context, AttributeSet attrs) {
        super(context, attrs);
        init();
    }
    private void init() {
        angle = 0;
        circlePaint = new Paint();
        circlePaint.setColor(Color.rgb(120,120,120));
        circlePaint.setStyle(Paint.Style.STROKE);
        circlePaint.setStrokeWidth(3);
        linePaint = new Paint();
        linePaint.setColor(Color.rgb(62,124,120));
        linePaint.setStrokeWidth(2);
    }
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        int centerX = getWidth() / 2;
        int centerY = getHeight() / 2;
        int radius = Math.min(centerX, centerY)-3;
        canvas.drawCircle(centerX, centerY, radius, circlePaint);
        float x= (float) (centerX+radius*cos((angle*3.14159)/180));
        float y= (float) (centerY+radius*sin((angle*3.14159)/180));
        canvas.drawLine(centerX, centerY, x, y, linePaint);
    }

    // Method to update the flag and trigger a redraw
    public void setDrawLine(int _angle){
        this.angle = _angle;
        this.invalidate(); // Trigger a redraw
    }
}
```

Créer dans le programme principal JAVA l'objet circleView : (l'objet binding doit être supprimé)

```
setContentView(R.layout.activity_main);
CircleView circleView = findViewById(R.id.circleView);

//binding = ActivityMainBinding.inflate(getLayoutInflater());
//setContentView(binding.getRoot());
circleView.setDrawLine(intV);
```

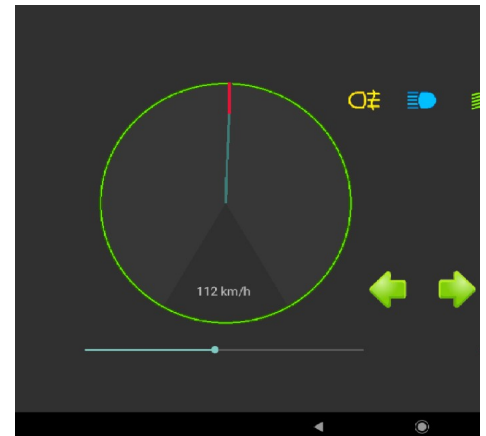
Dans le xml décrivant l'interface, ajouter :

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    tools:layout_editor_absoluteX="-66dp"
    tools:layout_editor_absoluteY="0dp">
    <com.example.nomProjet.CircleView
        android:id="@+id/circleView"
        android:layout_width="514dp"
        android:layout_height="473dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="74dp"
        android:layout_marginTop="184dp"
        android:layout_marginEnd="692dp"
        android:layout_marginBottom="143dp" />
</RelativeLayout>
```

Vérifier le code xml dans la vue de l'interface.

Ajouter une seekBar permettant de modifier l'angle de l'aiguille.

Afficher la vitesse du véhicule sur le cadran : (0km/h à 120° et 110km/h à 90°). Pour cela, une équation linéaire du type $y=ax+b$ doit être déterminée : $\text{angle}=a.\text{vitesse}+b$. Calculer a et b permettant de satisfaire : 0km/h à 120° et 110km/h à 90°.



Bonus : ajouter un son lorsque la vitesse dépasse les 130km/h

```
ToneGenerator toneG = new ToneGenerator(AudioManager.STREAM_ALARM, 100);
toneG.startTone(ToneGenerator.TONE_CDMA_INTERCEPT, 1000);
```

Versionner le code complet.