

**Correction TP07****Matière : ATELIER DEVELOPPEMENT MOBILE1****Classes : SEM21****Exercice 1**

```
package com.focus;
//imports
public class MainActivity extends Activity implements OnFocusChangeListener {
    private EditText edNom;
    private EditText edPrenom;
    private EditText edAdresse;
    private Button btnAjouter;
    private int couleurInitiale;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        init();
    }
    public void init() {
        edNom = (EditText) findViewById(R.id.edNom);
        edPrenom = (EditText) findViewById(R.id.edPrenom);
        edAdresse = (EditText) findViewById(R.id.edAdresse);
        btnAjouter = (Button) findViewById(R.id.btnAjouter);
        initCouleurs();
        actualiser();
        AjouterEcouteur();
    }
    private void initCouleurs() {
        edNom.setBackgroundColor(Color.argb(255, 200, 200, 200));
        edPrenom.setBackgroundColor(Color.argb(255, 150, 150, 150));
        edAdresse.setBackgroundColor(Color.argb(255, 100, 100, 100));
        btnAjouter.setBackgroundColor(Color.argb(255, 50, 50, 50));
    }
    private void actualiser() {
        couleurInitiale = ((ColorDrawable)edNom.getBackground()).getColor();
        if(edNom.hasFocus())
            edNom.setBackgroundColor(Color.CYAN);
    }
    public void AjouterEcouteur() {
        edNom.setOnFocusChangeListener(this);
        edPrenom.setOnFocusChangeListener(this);
        edAdresse.setOnFocusChangeListener(this);
        btnAjouter.setOnFocusChangeListener(this);
    }
    @Override
    public void onFocusChange(View arg0, boolean arg1) {
        if (arg1 == true) {
            couleurInitiale =((ColorDrawable)arg0.getBackground()).getColor();
            arg0.setBackgroundColor(Color.CYAN);
        } else
            arg0.setBackgroundColor(couleurInitiale);
    }
}
```

Exercice 2

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="top"
    android:orientation="vertical" >
    <ScrollView    android:id="@+id/scrollView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >
        <HorizontalScrollView      android:id="@+id/horizontalScrollView1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >
            <LinearLayout
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:gravity="top" >
                <ImageView
                    android:id="@+id/ivFruit"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:scaleType="matrix"
                    android:src="@drawable/animaux" />
            </LinearLayout>
        </HorizontalScrollView>
    </ScrollView>
</LinearLayout>
package com.zoom;
//imports
public class MainActivity extends Activity {
    public static final float PAS=0.1f;
    private ImageView ivFruit;
    private Matrix matrice;
    private float echelle;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        init();
    }
    private void init() {
        ivFruit = (ImageView) findViewById(R.id.ivFruit);
        matrice = new Matrix();
        echelle = PAS;
        agrandir();
        ajouterEcouteurs();
    }
    private void ajouterEcouteurs() {
        ivFruit.setOnLongClickListener(new OnLongClickListener() {
            @Override
            public boolean onLongClick(View v) {
                agrandir();
                return false;
            }
        });
    }
    protected void agrandir() {
        if (echelle < 1)
            echelle+=PAS;
        else
            echelle = PAS;
        matrice.setScale(echelle, echelle);
        ivFruit.setImageMatrix(matrice);
```

```
}
```

Exercice 3

```
<menu xmlns:androclass="http://schemas.android.com/apk/res/android" >
<item android:id="@+id/automne"
      android:title="Automne"/>
<item android:id="@+id/hivers"
      android:title="Hivers"/>
<item android:id="@+id/printemps"
      android:title="Printemps"/>
<item android:id="@+id/ete"
      android:title="Eté"/>
<item android:id="@+id/quitter"
      android:title="Quitter"/>
</menu>
public class MainActivity extends Activity {
    public static final int NB_SAISON = 4;
    private ImageView imgSaison;
    private TextView tvSaison;
    private int saisonCourante;
    private float xDebut, xFin;
    String saison;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        init();
    }
    private void init() {
        imgSaison = (ImageView) findViewById(R.id.imgSaison);
        tvSaison = (TextView) findViewById(R.id.tvSaison);
        saisonCourante = 0;
        selectionnerSaison(saisonCourante);
        ajouterEcouteur();
    }
    private void selectionnerSaison(int saisonCourante) {
        switch (saisonCourante) {
            case 0:
                imgSaison.setImageResource(R.drawable.automne);
                tvSaison.setText("Automne");
                break;
            case 1:
                imgSaison.setImageResource(R.drawable.hivers);
                tvSaison.setText("Hivers");
                break;
            case 2:
                imgSaison.setImageResource(R.drawable.printemps);
                tvSaison.setText("Printemps");
                break;
            case 3:
                imgSaison.setImageResource(R.drawable.ete);
                tvSaison.setText("Eté");
                break;
        }
    }
    private void ajouterEcouteur() {
        imgSaison.setOnTouchListener(new OnTouchListener() {
            @Override
            public boolean onTouch(View v, MotionEvent event) {
                switch (event.getAction()) {
                    case MotionEvent.ACTION_DOWN: // début du Touch on enregistre xDebut
                        xDebut = event.getX();
                }
            }
        });
    }
}
```

```

        break;
    case MotionEvent.ACTION_UP: // fin du Touch on enregistre xFin et on change
        // la saison
        xFin = event.getX();
        if (xDebut > xFin)
            saisonCourante = (saisonCourante + 1) % NB_SAISON; // suivant
        else
            saisonCourante = (saisonCourante - 1 + NB_SAISON) % NB_SAISON; // precedent
        selectionnerSaison(saisonCourante);
        break;
    }
    return true;
});
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return super.onCreateOptionsMenu(menu);
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    switch (id) {
        case R.id.automne:
            saisonCourante=0;
            selectionnerSaison(saisonCourante);
            break;
        case R.id.hivers:
            saisonCourante=1;
            selectionnerSaison(saisonCourante);
            break;
        case R.id.printemps:
            saisonCourante=2;
            selectionnerSaison(saisonCourante);
            break;
        case R.id.ete:
            saisonCourante=3;
            selectionnerSaison(saisonCourante);
            break;
        case R.id.quitter:
            confirmerQuitter();
    }
    return super.onOptionsItemSelected(item);
}
private void confirmerQuitter() {
    AlertDialog.Builder ab = new AlertDialog.Builder(MainActivity.this);
    ab.setTitle(this.getTitle() + " Quitter");
    ab.setMessage("Etes-vous sûr de vouloir quitter?");
    ab.setPositiveButton("Oui", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            MainActivity.this.finish();
        }
    });
    ab.setNegativeButton("Non", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            dialog.cancel();
        }
    });
    AlertDialog a = ab.create();
    a.show();
}
}

```

Exercice 4

```
public class Point {  
    public float x;  
    public float y;  
  
    public Point(float x, float y) {  
        this.x = x;  
        this.y = y;  
    }  
}  
public class DrawLL extends LinearLayout {  
    private Canvas canvas;  
    private SparseArray<Point> tabPoint;  
    public DrawLL(Context context) {  
        super(context);  
        tabPoint=new SparseArray<Point>();  
    }  
  
    @Override  
    protected void onDraw(Canvas canvas) {  
        super.onDraw(canvas);  
        this.canvas=canvas;  
        for(int i=0;i<tabPoint.size();i++) {  
            Point p=tabPoint.get(i);  
            dessinerCercle(p.x, p.y, 5);  
        }  
    }  
  
    private void dessinerCercle(float cx, float cy, float r){  
        Paint paint=new Paint();  
        paint.setStyle(Paint.Style.FILL);  
        paint.setStrokeWidth(r);  
        paint.setColor(Color.BLUE);  
        if(canvas!=null) {  
            canvas.drawCircle(cx, cy, r, paint);  
        }  
    }  
    public void ajouterPoint(Point p){  
        tabPoint.put(tabPoint.size(),p);  
        invalidate();  
    }  
}  
public class MainActivity extends AppCompatActivity {  
    private LinearLayout mainLL;  
    private DrawLL drawLL;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        initGraphique();  
    }  
  
    private void initGraphique() {  
        mainLL=findViewById(R.id.mainLL);  
        drawLL=new DrawLL(this);  
        drawLL.setBackgroundColor(Color.CYAN);  
        drawLL.setMinimumWidth(200);  
        drawLL.setMinimumHeight(1000);  
        drawLL.setVisibility(View.VISIBLE);  
        mainLL.addView(drawLL);  
    }  
}
```

```
drawLL.postInvalidate();
drawLL.invalidate();
ajouterEcouteurs();
}

private void ajouterEcouteurs() {
    drawLL.setOnTouchListener(new View.OnTouchListener() {
        @Override
        public boolean onTouch(View view, MotionEvent motionEvent) {
            //drawLL.ajouterPoint(new Point(motionEvent.getX(),motionEvent.getY()));
            for(int i=0;i<motionEvent.getPointerCount();i++)
                drawLL.ajouterPoint(new Point(motionEvent.getX(i),motionEvent.getY(i)));
            return true;
        }
    });
}

}
```