

## Correction TP05

Matière : ATELIER DEVELOPPEMENT MOBILE 1

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 = findViewById(R.id.edNom);
        edPrenom = findViewById(R.id.edPrenom);
        edAdresse = findViewById(R.id.edAdresse);
        btnAjouter = findViewById(R.id.btnAjouter);
        initCouleurs();
        actualiser();
        AjouterEcouteur();
    }
    private void initCouleurs() {
        edNom.setBackgroundColor(Color.rgb(255, 200, 200, 200));
        edPrenom.setBackgroundColor(Color.rgb(255, 150, 150, 150));
        edAdresse.setBackgroundColor(Color.rgb(255, 100, 100, 100));
        btnAjouter.setBackgroundColor(Color.rgb(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 view, boolean hasFocus) {
        if (hasFocus == true) {
            couleurInitiale = ((ColorDrawable) view.getBackground()).getColor();
            view.setBackgroundColor(Color.CYAN);
        } else
            view.setBackgroundColor(couleurInitiale);
    }
}
```

## Exercise 2

```
<ImageView
    android:id="@+id/ivFruit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:scaleType="matrix"
    android:src="@drawable/animaux" />

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);
    }
}
```

## Exercise 3

```
<menu xmlns:android="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;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        init();
    }
    private void init() {
        imgSaison = findViewById(R.id.imgSaison);
        tvSaison = 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("Été");
                break;
        }
    }
    private void ajouterEcouteur() {
        imgSaison.setOnTouchListener(new OnTouchListener() {
            @Override
            public boolean onTouch(View v, MotionEvent event) {
                float xDebut, xFin;
                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)
                            suivant();
                        else
                            precedent();
                }
                break;
            }
        });
        return true;
    }
}

```

```

private void suivant() {
    saisonCourante = (saisonCourante + 1) % NB_SAISON;
    selectionnerSaison(saisonCourante);
}
private void precedent() {
    if(saisonCourante==0)
        saisonCourante= NB_SAISON - 1 ;
    else
        saisonCourante-- ;
    selectionnerSaison(saisonCourante);
}
@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();
}
}

```