

# **Compte rendu :**

## **Lab07 - RecyclerView - Systeme Solaire**

**Enseignant :** Maddouri Faouzi

**Réalisé par :**

Farhaoui Eya

Guesmi Ikram

**Année universitaire :** 2024/2025

**Etablissement:** ISET Rades

# Sommaire

<b>1</b>	<b>Utilisation de recyclerview:</b>	<b>2</b>
1.1	Dans MainActivity : . . . . .	2
1.2	DansPlaneteAdapter : . . . . .	3
<b>2</b>	<b>Implémentation d'un Listener:</b>	<b>4</b>
2.1	Dans PlaneteAdapter : . . . . .	4
2.2	Dans DetailsActivity : . . . . .	5
<b>3</b>	<b>les interfaces:</b>	<b>5</b>

Liste des Figures

1	l'activité d'accueil . . . . .	6
2	l'activité de la planète mars . . . . .	6
3	l'activité de la planète mercure . . . . .	7
4	l'activité de la planète terre . . . . .	7
5	l'activité de venus . . . . .	8

## Objectif:

création d'une application qui affiche la liste des planetes du systeme solaire et leurs détails par exemple; masse, periode de revolution en jours terrestres, Nombre de staelites orbitantes, etc.Et cela, doit être réalisé en utilisant un recyclerview et en implementant un listener .

# 1 Utilisation de recyclerview:

## 1.1 Dans MainActivity :

a) code: le RecyclerView à été utilisé dans l'activité MainActivity pour afficher une liste d'objets Planete dans une interface graphique

```
1 package tn.rnu.isetr.app_solaire;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.recyclerview.widget.LinearLayoutManager;
5 import androidx.recyclerview.widget.RecyclerView;
6
7 import android.os.Bundle;
8
9 import java.util.ArrayList;
10 import java.util.List;
11
12 public class MainActivity extends AppCompatActivity {
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         RecyclerView recyclerView = findViewById(R.id.recyclerViewPlanetes);
20         recyclerView.setLayoutManager(new LinearLayoutManager(this));
21
22         List<Planete> planetes = new ArrayList<>();
23         planetes.add(new Planete("Mercure", R.drawable.mercure, "57.91_million_km",
24             "3.3011 10 ^23_kg", "88_jours"));
25         planetes.add(new Planete("Venus", R.drawable.venus, "108.2_million_km", "
26             4.8675 10 ^24_kg", "225_jours"));
27         planetes.add(new Planete("La_Terre", R.drawable.terre, "149.6_million_km",
28             "5.972 10 ^24_kg", "365.25_jours"));
29         planetes.add(new Planete("Mars", R.drawable.mars, "227.9_million_km", "
30             6.4171 10 ^23_kg", "687_jours"));
31
32         PlaneteAdapter adapter = new PlaneteAdapter(planetes, this);
33         recyclerView.setAdapter(adapter);
34     }
35 }
```

b) Explication:

- **Déclaration et initialisation** : RecyclerView est initialisé via findViewById : `RecyclerView recyclerView = findViewById(R.id.recyclerViewPlanetes);`
- **Configuration du LayoutManager** : pour définir la disposition des éléments de ce recyclerView (liste linéaire), on a utilisé LinearLayoutManager dans cette ligne de code `recyclerView.setLayoutManager(new LinearLayoutManager(this));`

## 1.2 Dans PlaneteAdapter :

a) code: Dans ce code, le RecyclerView est utilisé pour afficher une liste de planètes (les objets Planete). Chaque élément de la liste contient un nom de planète et une image associée, et chaque élément est cliquable, ce qui permet de naviguer vers une activité de détails pour la planète sélectionnée.

```
1 package tn.rnu.isetr.app_solaire;
2
3 import android.content.Context;
4 import android.content.Intent;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.ImageView;
9 import android.widget.TextView;
10
11 import androidx.annotation.NonNull;
12 import androidx.recyclerview.widget.RecyclerView;
13
14 import java.util.List;
15
16 public class PlaneteAdapter extends RecyclerView.Adapter<PlaneteAdapter.
17     PlaneteViewHolder> {
18     private List<Planete> planetes;
19     private Context context; // D clarez la variable context
20
21     // Modifiez le constructeur pour accepter le context
22     public PlaneteAdapter(List<Planete> planetes, Context context) {
23         this.planetes = planetes;
24         this.context = context; // Initialisez le contexte ici
25     }
26
27     @NonNull
28     @Override
29     public PlaneteViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
30         viewType) {
31         View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.
32             item_planete, parent, false);
33         return new PlaneteViewHolder(view);
34     }
35
36     @Override
37     public void onBindViewHolder(@NonNull PlaneteViewHolder holder, int
38         position) {
39         Planete planete = planetes.get(position);
40         holder.textViewNom.setText(planete.getNom());
41         holder.imageViewPhoto.setImageResource(planete.getPhotoResId());
42
43         holder.itemView.setOnClickListener(v -> {
44             Intent intent = new Intent(context, DetailsActivity.class);
45             intent.putExtra("nom", planete.getNom());
46             intent.putExtra("photoResId", planete.getPhotoResId());
47             intent.putExtra("distanceSoleil", planete.getDistanceSoleil());
```

```

45         intent.putExtra("masse", planete.getMasse());
46         intent.putExtra("periodeRevolution", planete.getPeriodeRevolution
47             ());
48         context.startActivity(intent);
49     });
50 }
51 @Override
52 public int getItemCount() {
53     return planetes.size();
54 }
55
56 public static class PlaneteViewHolder extends RecyclerView.ViewHolder {
57     TextView textViewNom;
58     ImageView imageViewPhoto;
59
60     public PlaneteViewHolder(@NonNull View itemView) {
61         super(itemView);
62         textViewNom = itemView.findViewById(R.id.textViewNom);
63         imageViewPhoto = itemView.findViewById(R.id.imageViewPhoto);
64     }
65 }
66 }

```

b) Explication:

- **Constructeur de PlaneteAdapter** : Le constructeur est appelé par le RecyclerView pour initialiser PlaneteAdapter avec la liste des données (les planètes) et le contexte de l'application. Le RecyclerView utilise PlaneteAdapter pour obtenir les données à afficher.
- **onBindViewHolder** : Le RecyclerView appelle cette méthode chaque fois qu'une vue recyclée doit être liée à des données spécifiques. Elle associe les données d'une planète (nom et image) à la vue correspondante (holder). Cette méthode garantit que chaque élément de la liste est affiché correctement, même lors du défilement où les vues sont réutilisées.
- **getItemCount** : Cette méthode informe le RecyclerView du nombre total d'éléments dans la liste. Le RecyclerView utilise cette information pour déterminer combien d'éléments il doit gérer et afficher.

## 2 Implémentation d'un Listener:

### 2.1 Dans PlaneteAdapter :

a) code:

```

1 holder.itemView.setOnClickListener(v -> {
2     Intent intent = new Intent(context, DetailsActivity.class);
3     intent.putExtra("nom", planete.getNom());
4     intent.putExtra("photoResId", planete.getPhotoResId());
5     intent.putExtra("distanceSoleil", planete.getDistanceSoleil());
6     intent.putExtra("masse", planete.getMasse());
7     intent.putExtra("periodeRevolution", planete.getPeriodeRevolution());
8     context.startActivity(intent);
9 });

```

b) explication; Dans ce code, Un listener de type OnClickListener est implémenté pour gérer les clics sur chaque élément de la liste. Lorsqu'un utilisateur clique sur un élément, un Intent est créé pour démarrer l'activité DetailsActivity. Les données de la planète sélectionnée (nom, image, distance au Soleil, masse, période de révolution) sont transmises à l'activité via les putExtra. L'activité DetailsActivity est démarrée en appelant context.startActivity(intent).

## 2.2 Dans DetailsActivity :

a) code:

```

1 buttonRetour.setOnClickListener(v -> {
2     // Revenir la MainActivity
3     Intent retourIntent = new Intent(DetailsActivity.this, MainActivity.class)
4     ;
5     startActivity(retourIntent);
6     finish(); pour fermer la page de d tails
7 });
8 });

```

b) explication: Le listener utilisé ici est un OnClickListener. Ce listener est déclenché lorsqu'un utilisateur clique sur le bouton. En fait, Lorsqu'on clique sur le bouton "Retour", l'application fait la création d'un Intent pour naviguer vers MainActivity. En effet, elle Lance cette activité en appelant startActivity(retourIntent). Finalement, elle Appelle finish() pour fermer la DetailsActivity et empêcher l'utilisateur de revenir en arrière à cette activité via le bouton de retour système.

## 3 les interfaces:



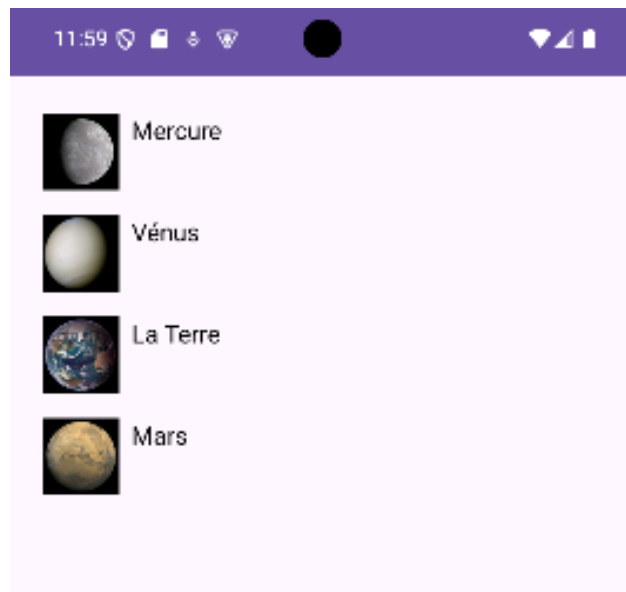


Figure 1: l'activité d'accueil



Figure 2: l'activité de la planète mars

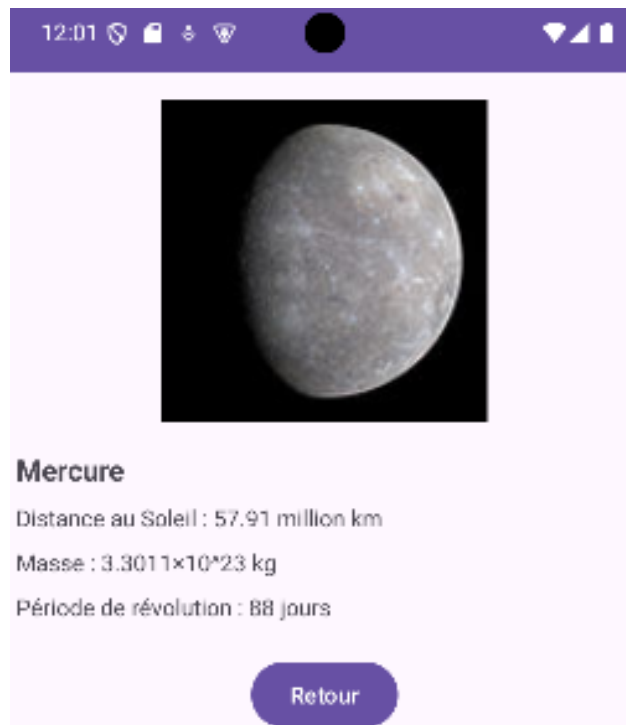


Figure 3: l'activité de la planète mercure

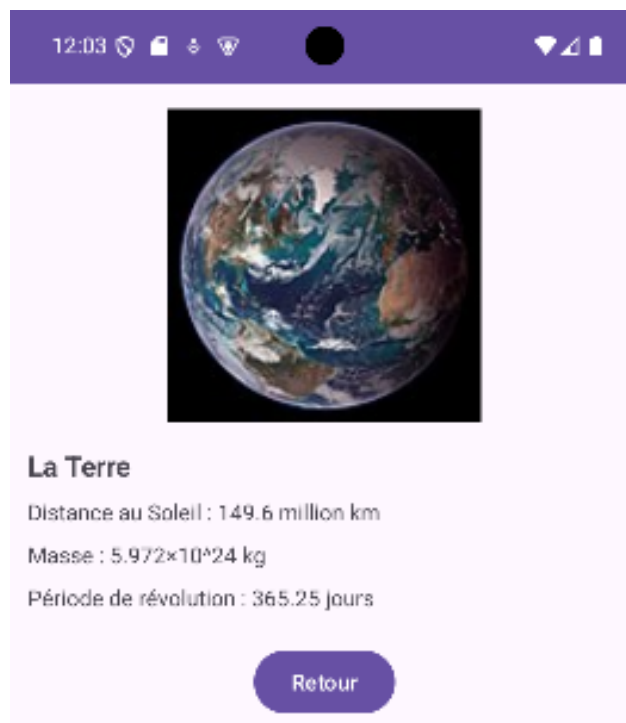


Figure 4: l'activité de la planète terre

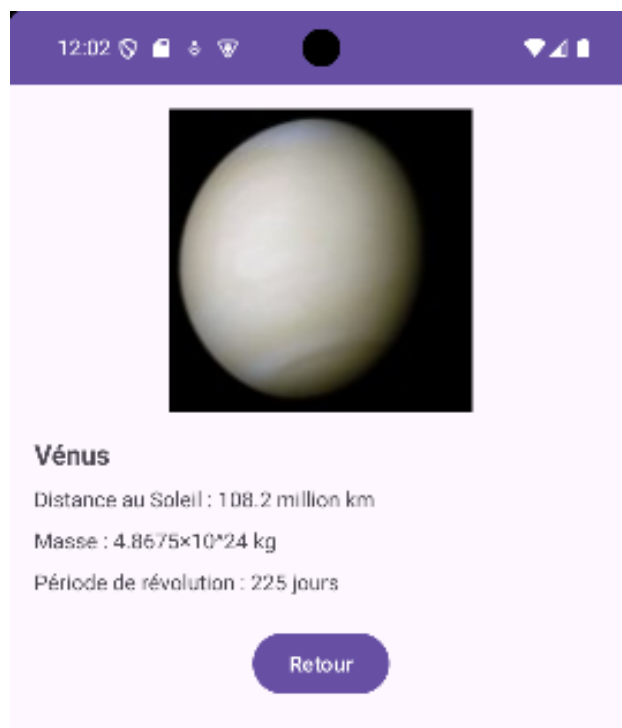


Figure 5: l'activité de venus