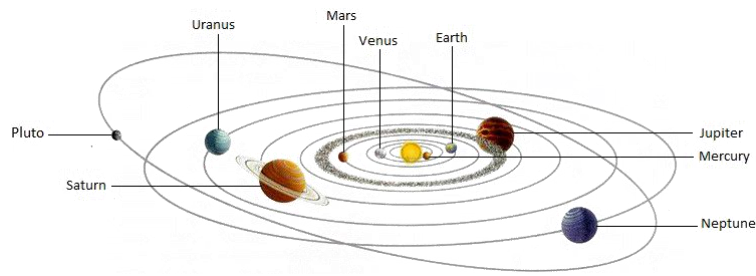


Rapport de Projet : Application Android pour le Système Solaire



Introduction

Ce projet vise à développer une application Android qui permet à l'utilisateur d'explorer les planètes du système solaire. L'application utilise une interface utilisateur intuitive pour afficher une liste de planètes avec des détails sur chacune d'elles, permettant à l'utilisateur de naviguer facilement entre les différentes informations.

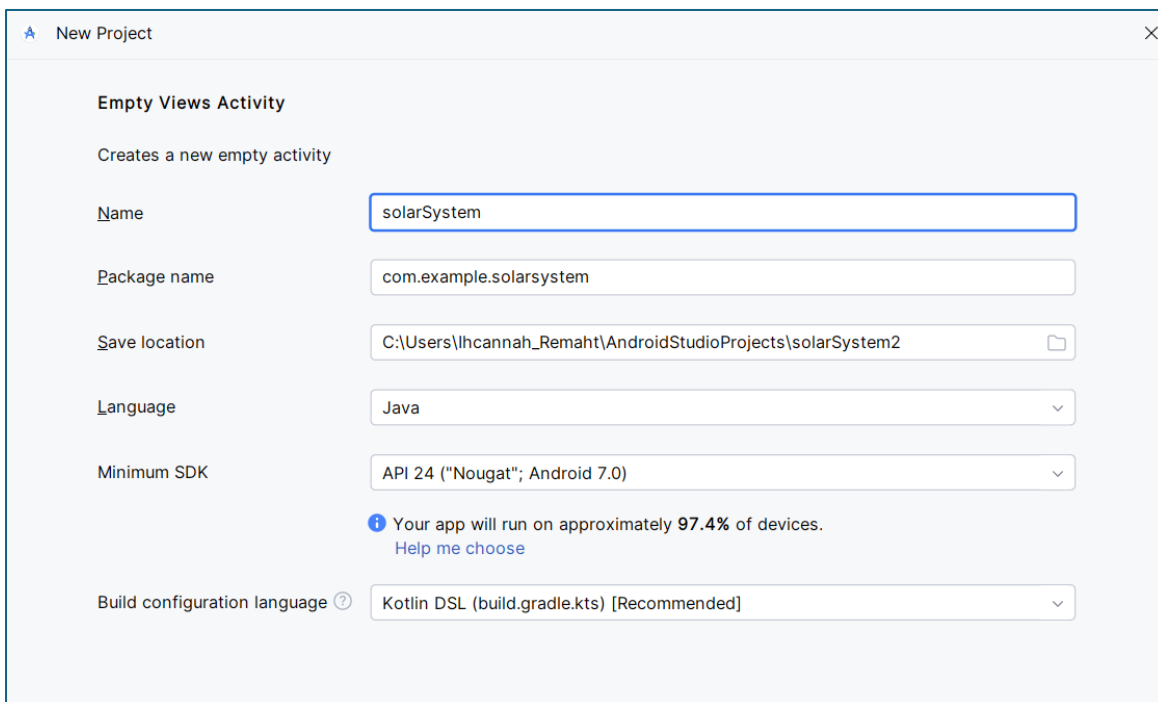
Objectifs

- Créer une application Android affichant une liste des planètes du système solaire.
- Implémenter un `RecyclerView` pour une présentation efficace des données.
- Ajouter une activité de détail pour afficher des informations complètes sur chaque planète.
- Permettre la navigation entre la liste des planètes et les détails de chaque planète.

Étapes de Développement

1. Création du Projet

Un nouveau projet Android a été créé dans Android Studio avec un nom de package `com.example.solarsystem`.



2. Configuration des Dépendances

Les dépendances nécessaires ont été ajoutées au fichier build.gradle :

Étape 1 : Ajouter RecyclerView à versions

Ajoutez la version de RecyclerView dans la section [versions] :

```
1  [versions]
2  agp = "8.6.0"
3  junit = "4.13.2"
4  junitVersion = "1.2.1"
5  espressoCore = "3.6.1"
6  appcompat = "1.7.0"
7  material = "1.12.0"
8  activity = "1.9.3"
9  constraintlayout = "2.2.0"
10 recyclerview = "1.3.1"
11
```

Étape 2 : Ajouter RecyclerView à libraries

Ajoutez RecyclerView à la section [libraries] :

```
12 [libraries]
13 junit = { group = "junit", name = "junit", version.ref = "junit" }
14 ext-junit = { group = "androidx.test.ext", name = "junit", version.ref = "junitVersion" }
15 espresso-core = { group = "androidx.test.espresso", name = "espresso-core", version.ref = "espressoCore" }
16 appcompat = { group = "androidx.appcompat", name = "appcompat", version.ref = "appcompat" }
17 material = { group = "com.google.android.material", name = "material", version.ref = "material" }
18 activity = { group = "androidx.activity", name = "activity", version.ref = "activity" }
19 constraintlayout = { group = "androidx.constraintlayout", name = "constraintlayout", version.ref = "constraintlayout" }
20 recyclerview = { group = "androidx.recyclerview", name = "recyclerview", version.ref = "recyclerview" }
21 [plugins]
22 android-application = { id = "com.android.application", version.ref = "agp" }
23
24
```

Étape 3 : Ajouter RecyclerView dans build.gradle de votre module (app)

Hnnachi Thamer L3 dsi ead

Si vous utilisez le fichier TOML pour gérer les dépendances, les dépendances dans le fichier build.gradle du module doivent inclure recyclerview de la manière suivante :

```
34     dependencies {
35
36
37         implementation(libs.appcompat)
38         implementation(libs.material)
39         implementation(libs.activity)
40         implementation(libs.constraintlayout)
41         testImplementation(libs.junit)
42         androidTestImplementation(libs.ext.junit)
43         androidTestImplementation(libs.espresso.core)
44         implementation(libs.recyclerview)
45     }
```

3. Création des Modèles de Données

Un modèle Planet a été créé pour représenter les planètes avec les attributs suivants :

```
public class Planet implements Serializable { 14 usages
    private String name; 2 usages
    private int imageResId; 2 usages
    private double distanceFromSun; // En millions de km 2 usages
    private double mass; // En kg 2 usages
    private double revolutionPeriod; // En jours terrestres 2 usages
    private int numberOfSatellites; 2 usages
    private double diameter; // En kilomètres 2 usages
    private double gravity; // En m/s² 2 usages
    private String atmosphere; // Composition de l'atmosphère 2 usages
    private double minTemperature; // Température minimale 2 usages
    private double maxTemperature; // Température maximale 2 usages

    public Planet(String name, int imageResId, double distanceFromSun, double mass, 8 usages
        double revolutionPeriod, int numberOfSatellites, double diameter,
        double gravity, String atmosphere, double minTemperature, double maxTemperature) {
        this.name = name;
        this.imageResId = imageResId;
        this.distanceFromSun = distanceFromSun;
        this.mass = mass;
        this.revolutionPeriod = revolutionPeriod;
        this.numberOfSatellites = numberOfSatellites;
        this.diameter = diameter;
        this.gravity = gravity;
        this.atmosphere = atmosphere;
        this.minTemperature = minTemperature;
        this.maxTemperature = maxTemperature;
    }
}
```

4 . Layout de la liste

1. Fichier res/layout/activity_main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android=
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

Inclure un RecyclerView :

Fichier res/layout/item_planet.xml :

- Layout pour un élément de liste

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="8dp">
    <ImageView
        android:id="@+id/planetImage"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:scaleType="centerCrop"
        android:layout_marginEnd="8dp" />
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="vertical">
        <TextView
            android:id="@+id/planetName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="18sp"
            android:textStyle="bold" />
        <TextView
            android:id="@+id/planetDistance"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="14sp" />
    </LinearLayout>
</LinearLayout>
```

Étape 4 : Adapter et RecyclerView

1. Classe PlanetAdapter.java

PlanetAdapter est responsable de la connexion entre les données (la liste des planètes) et l'interface utilisateur (l'affichage dans le RecyclerView), garantissant que chaque élément est correctement présenté et interactif.

Rôle de PlanetAdapter :

1. **Adaptation des Données** : PlanetAdapter prend une liste d'objets Planet (qui contiennent des informations comme le nom, l'image et la distance par rapport au soleil) et les adapte pour qu'ils soient affichés dans le RecyclerView.
2. **Création de Vues** : Elle crée des vues pour chaque élément de la liste lorsque le RecyclerView nécessite de nouveaux éléments à afficher. Cela se fait généralement dans la méthode onCreateViewHolder().
3. **Liaison des Données** : Elle lie les données de chaque objet Planet aux vues correspondantes dans onBindViewHolder(), en définissant les valeurs du texte, des images, etc.
4. **Gestion des Interactions** : PlanetAdapter peut également gérer les interactions des utilisateurs, comme les clics sur les éléments, pour naviguer vers une nouvelle activité affichant plus de détails sur la planète sélectionnée.

```
PlanetAdapter.java x MainActivity.java build.gradle.kts (:app) build.gradle.kts (SolarSystem)
14 public class PlanetAdapter extends RecyclerView.Adapter<PlanetAdapter.PlanetViewHolder>
15     private Context context; 4 usages
16     private List<Planet> planetList; 3 usages
17
18     public PlanetAdapter(Context context, List<Planet> planetList) { 1 usage
19         this.context = context;
20         this.planetList = planetList;
21     }
22     @NonNull
23     @Override
24     public PlanetViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)
25         View view = LayoutInflater.from(context).inflate(R.layout.item_planet, parent,
26             false);
27         return new PlanetViewHolder(view);
28     }
29     @Override
30     public void onBindViewHolder(@NonNull PlanetViewHolder holder, int position) {
31         Planet planet = planetList.get(position);
32         holder.planetName.setText(planet.getName());
33         holder.planetDistance.setText("Distance: " + planet.getDistanceFromSun() + " mi");
34         holder.planetImage.setImageResource(planet.getImageResId());
35
36         holder.itemView.setOnClickListener(v -> {
37             Intent intent = new Intent(context, PlanetDetailActivity.class);
38             intent.putExtra("name", planet.getName());
39             context.startActivity(intent);
40         });
41     }
42     @Override
43     public int getItemCount() {
44         return planetList.size();
45     }
```

Voici une brève description de chaque méthode de la classe PlanetAdapter :

1. PlanetAdapter(Context context, List<Planet> planetList)

- **Description** : Constructeur de la classe PlanetAdapter. Il initialise le contexte de l'application et la liste des planètes à afficher dans le RecyclerView.

2. onCreateViewHolder(@NonNull ViewGroup parent, int viewType)

- **Description** : Cette méthode est appelée pour créer de nouveaux ViewHolder. Elle utilise LayoutInflater pour gonfler le layout item_planet et retourner un nouvel objet PlanetViewHolder contenant la vue gonflée. C'est ici que vous définissez le type de vue qui sera utilisé pour chaque élément de la liste.

3. onBindViewHolder(@NonNull PlanetViewHolder holder, int position)

- **Description** : Cette méthode est appelée pour lier les données de la planète à la vue. Elle prend le ViewHolder correspondant à la position actuelle dans la liste et met à jour ses vues (nom, distance et image de la planète) en fonction des données de l'objet Planet à cette position. Elle configure également un écouteur de clic sur l'élément de la liste, qui démarre PlanetDetailActivity avec les informations de la planète sélectionnée.

4. getItemCount()

- **Description** : Cette méthode retourne le nombre total d'éléments dans la liste planetList. Elle est utilisée par le RecyclerView pour savoir combien d'éléments il doit afficher.

5. static class PlanetViewHolder extends RecyclerView.ViewHolder

- **Description** : Classe interne qui étend RecyclerView.ViewHolder. Elle définit les vues qui seront utilisées pour afficher chaque élément de la liste des planètes. Dans le constructeur, les éléments de la vue (TextView pour le nom et la distance de la planète, et ImageView pour l'image) sont initialisés en les recherchant dans le layout item_planet.

Étape 5 : Activité principale

1. Classe MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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

        RecyclerView recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(context, this));

        List<Planet> planets = new ArrayList<>();
        planets.add(new Planet( name: "Mercure", R.drawable.mercure, distanceFromSun: 57.9, mass: 3.285e23));
        planets.add(new Planet( name: "Vénus", R.drawable.venus, distanceFromSun: 108.2, mass: 4.867e24));
        planets.add(new Planet( name: "Terre", R.drawable.terre, distanceFromSun: 149.6, mass: 5.972e24));
        planets.add(new Planet( name: "Mars", R.drawable.mars, distanceFromSun: 227.9, mass: 6.39e23, revolutio));
        planets.add(new Planet( name: "Jupiter", R.drawable.jupiter, distanceFromSun: 778.5, mass: 1.898e27));
        planets.add(new Planet( name: "Saturne", R.drawable.saturne, distanceFromSun: 1427.0, mass: 5.683e26));
        planets.add(new Planet( name: "Uranus", R.drawable.uranus, distanceFromSun: 2871.0, mass: 8.681e25));
        planets.add(new Planet( name: "Neptune", R.drawable.neptune, distanceFromSun: 4495.1, mass: 1.024e26));

        PlanetAdapter adapter = new PlanetAdapter( context: this, planets);
        recyclerView.setAdapter(adapter);
    }
}
```

2. `List<Planet> planets = new ArrayList<>()` est rempli en se referant au document fournis

Planète	Mercure	Vénus	La Terre	Mars	Jupiter	Saturne	Uranus	Neptune
Distance moyenne au Soleil	57,91	108,2	149,6	227,9	778,3	1 429	2 875	4 504
1 : en millions de kilomètres 2 : en unité astronomique	0,3871	0,7233	1,000	1,524	5,203	9,537	19,23	30,10
Période de révolution	87 j 23 h	224 j 17 h	365 j 6 h	686 j 23 h	11 ans 315 j	29 ans 167 j	84 ans 7 j	164 ans 281 j
Excentricité de l'orbite	0,2056	0,0068	0,0167	0,0934	0,0485	0,0555	0,0464	0,0095
Inclinaison de l'orbite sur l'écliptique	7,00°	3,39°	0° (par définition)	1,85°	1,30°	2,49°	0,77°	1,77°
Inclinaison de l'axe de rotation	0,035°	177,3°	23,44°	25,19°	3,13°	26,73°	97,86°	28,31°
Période de rotation	58 j 16 h	243 j 0 h (rétrograde)	23 h 56 min	24 h 37 min	9 h 55 min	10 h 39 min	17 h 14 min (rétrograde)	16 h 7 min
Masse (Terre = 1)	0,0553	0,815	1,00	0,107	318	95,2	14,5	17,1
Diamètre équatorial (en kilomètres)	4 879	12 104	12 756	6 792	142 984	120 536	51 118	49 530
Densité (eau = 1)	5,43	5,20	5,52	3,93	1,33	0,687	1,27	1,64
Gravité à l'équateur (Terre = 1)	0,378	0,907	1,00	0,377	2,36	0,916	0,889	1,12
Constituants atmosphériques	Atmosphère quasi inexistante	CO ₂ N ₂	N ₂ O ₂ Ar	CO ₂ N ₂ Ar	H ₂ He	H ₂ He	H ₂ He CH ₄	H ₂ He CH ₄
Température	Min. : - 200 °C Max. : 430 °C	460 °C	Min. : - 90 °C Moy. : 15 °C Max. : 60 °C	Min. : - 140 °C Moy. : - 60 °C Max. : 20 °C	- 150 °C (au sommet des nuages)	- 180 °C (au sommet des nuages)	- 200 °C (au sommet des nuages)	- 200 °C (au sommet des nuages)
Nombre de satellites connus	0	0	1	2	67	62	27	14

Étape 6 : Activité pour les détails

1. Fichier res/layout/activity_planet_detail.xml :

- Layout pour afficher les détails d'une planète :


```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/detailImage"
        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:adjustViewBounds="true"
        android:scaleType="fitCenter"
        android:layout_marginBottom="16dp" />

    <TextView
        android:id="@+id/detailName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginTop="8dp" />

    <TextView
        android:id="@+id/detailMass"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:textSize="16sp" />
```

```
<TextView
    android:id="@+id/detailRevolution"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:textSize="16sp" />

<TextView
    android:id="@+id/detailSatellites"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:textSize="16sp" />

<TextView
    android:id="@+id/detailDiameter"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:textSize="16sp" />

<TextView
    android:id="@+id/detailGravity"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:textSize="16sp" />
```

Classe PlanetDetailActivity.java :

Récupération des Données de la Planète :

- La planète est récupérée à partir de l'intent qui a démarré cette activité. Elle est extraite en utilisant `getSerializableExtra("planet")`. Cela suppose que la classe Planet implémente l'interface `Serializable`, permettant ainsi de passer des objets entre les activités.

Affichage des Détails de la Planète :

- Les données de l'objet Planet (image, nom, masse, période de révolution, et nombre de satellites) sont ensuite affichées dans les éléments d'interface utilisateur respectifs. Cela permet à l'utilisateur de voir les informations détaillées de la planète sélectionnée.

```

10 public class PlanetDetailActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_planet_detail);
16         ImageView detailImage = findViewById(R.id.detailImage);
17         TextView detailName = findViewById(R.id.detailName);
18         TextView detailMass = findViewById(R.id.detailMass);
19         TextView detailRevolution = findViewById(R.id.detailRevolution);
20         TextView detailSatellites = findViewById(R.id.detailSatellites);
21         TextView detailDiameter = findViewById(R.id.detailDiameter);
22         TextView detailGravity = findViewById(R.id.detailGravity);
23         TextView detailAtmosphere = findViewById(R.id.detailAtmosphere);
24         TextView detailTempRange = findViewById(R.id.detailTempRange);
25         Planet planet = (Planet) getIntent().getSerializableExtra("planet");
26         if (planet == null) {
27             throw new IllegalArgumentException("Planet object is null!");
28         }
29
30         detailImage.setImageResource(planet.getImageResId());
31         detailName.setText(planet.getName());
32         detailMass.setText("Masse: " + planet.getMass() + " kg");
33         detailRevolution.setText("Période de révolution: " + planet.getRevolutionP
34         detailSatellites.setText("Nombre de satellites: " + planet.getNumberofSate
35         detailDiameter.setText("Diamètre: " + planet.getDiameter() + " km");
36         detailGravity.setText("Gravité: " + planet.getGravity() + " m/s²");
37         detailAtmosphere.setText("Atmosphère: " + planet.getAtmosphere());
38         detailTempRange.setText("Température: " + planet.getMinTemperature() + " °
39
40         Button backButton = findViewById(R.id.backButton);
41         backButton.setOnClickListener(v -> finish());

```

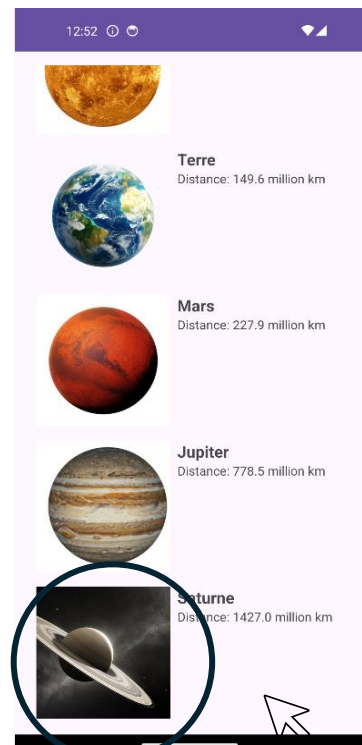
Affichage des données



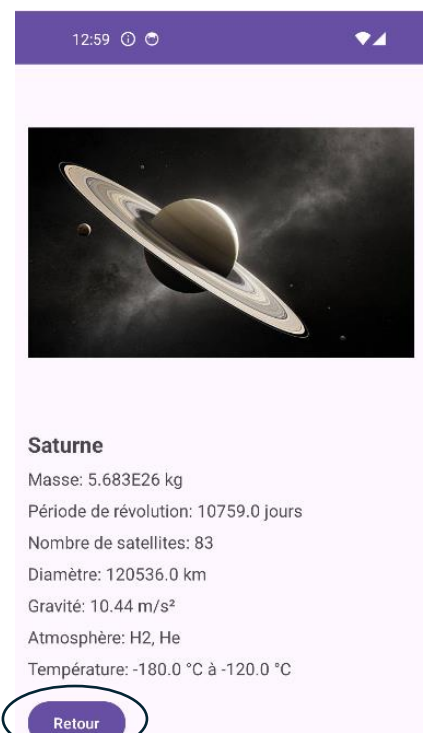
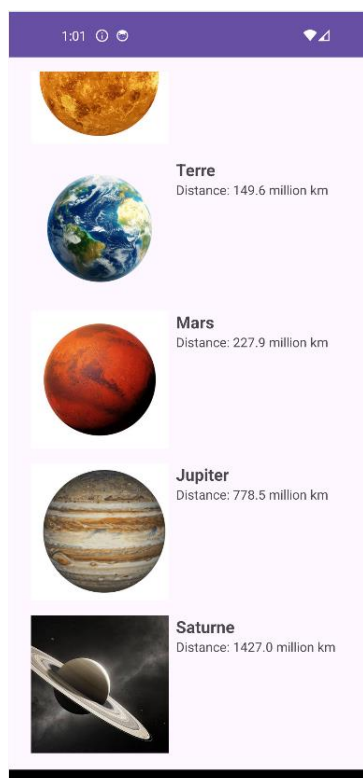
Passer les données depuis le RecyclerView :

```
@Override
public void onBindViewHolder(@NonNull PlanetViewHolder holder, int position) {
    Planet planet = planetList.get(position);
    holder.planetName.setText(planet.getName());
    holder.planetDistance.setText("Distance: " + planet.getDistanceFromSun() + " million km");
    holder.planetImage.setImageResource(planet.getImageResId());

    holder.itemView.setOnClickListener(v -> {
        Intent intent = new Intent(context, PlanetDetailActivity.class);
        intent.putExtra("name: planet", planet);
        context.startActivity(intent);
    });
}
```



```
holder.itemView.setOnClickListener(v -> {
    Intent intent = new Intent(context, PlanetDetailActivity.class);
    intent.putExtra("name", planet);
    context.startActivity(intent);
});
```



```
Button backButton = findViewById(R.id.backButton);
backButton.setOnClickListener(v -> finish());
```