Android Intro

Robin Penea

Objectif

Vous donner les outils pour démarrer n'importe quelle app Android

Sommaire

1. Historique Android
2. Projet d'entrainement
3. Projet de la semaine

Historique d'Android

- OS Mobile racheté par Google en 2005
- 1ère release officielle : 2008
- Equipe aujourd'hui 87% des téléphones / tablettes

Robin Penea

Architecture de l'OS

Robin Penea

Pour les applications

- IDE utilisé : Android Studio
  - Un fork d'IntelliJ
- En Java ou Kotlin
  - Accès à tous l'écosystème de bibliothèques Java
- L'UI décrite en XML
  - Aujourd'hui un bon éditeur graphique

Anatomie d'un projet Android

- `AndroidManifest.xml` : Carte d'identité de l'App
  - Déclaration des Activity, point d'entrée du programme, etc
- `app/build.gradle` : Configuration du build Gradle
  - minSDK, targetSDK, bibliothèques

Anatomie d'un projet Android

- `src/java` : Code du projet
- `res/` : Ressources du projet
  - `drawable/` : images jpg / png / svg embarquées
  - `layout/` : fichiers XML de définition des écrans
  - `values/` : fichiers XML de traductions, themes, menus, etc.

⚠️ Une ressource doit être nommée `[a-z][0-9]_`

TP 1

- Création du projet d'entrainement **Money**
- Un convertisseur de monnaie

Objectif

- Afficher un message "Hello World" dans le Logcat

Robin Penea

Activity

- 1 écran = 1 Activity
- 1 Activity
  - une classe Java qui hérite de `AppCompatActivity` (logique)
  - un fichier XML qui décrit l'écran (UI)

Activity

- Le point d'entrée est la fonction `onCreate()`
- C'est l'Activity qui choisit son layout

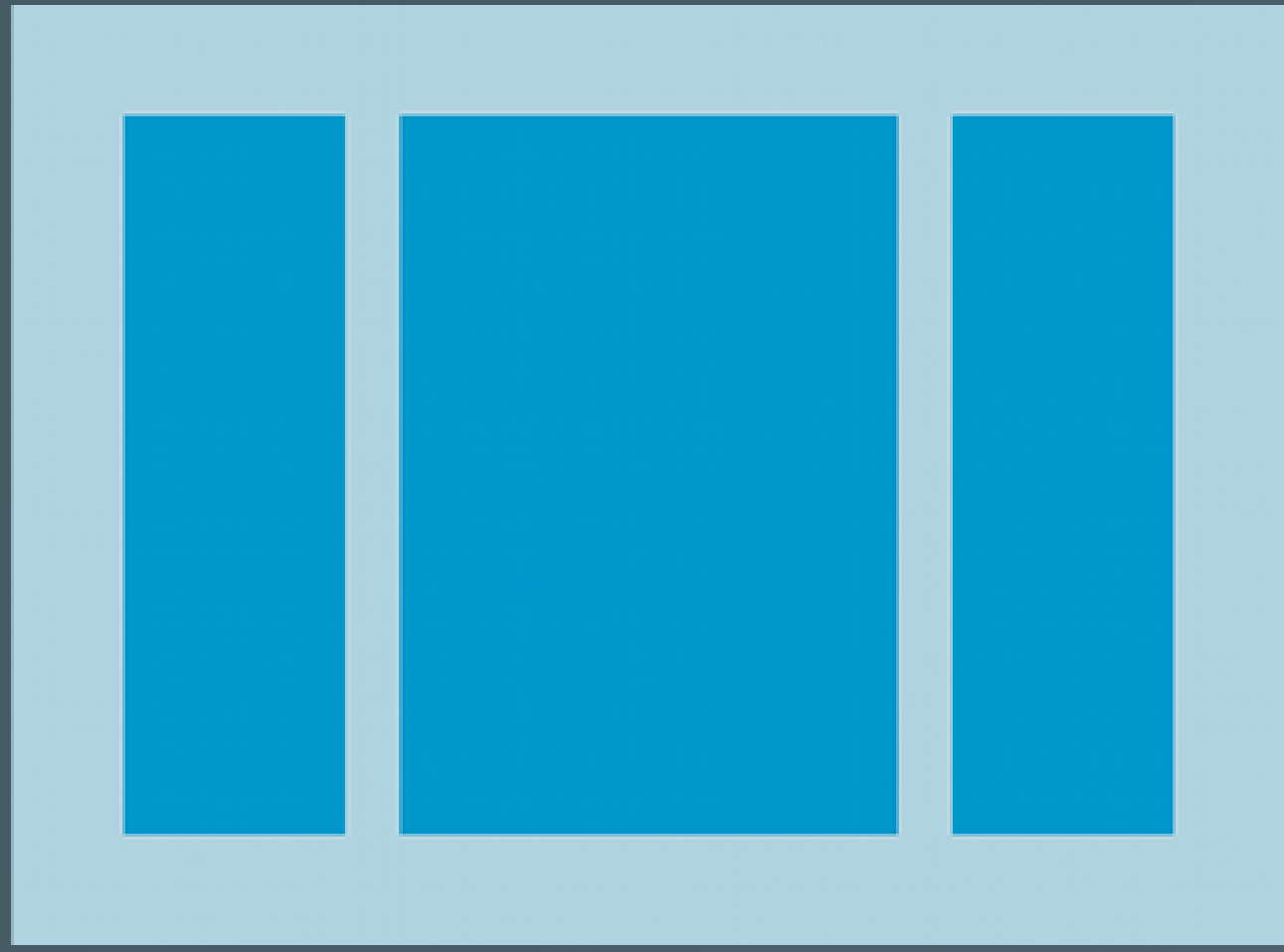
```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // réf vers le fichier XML
    }
}
```

Layouts

Robin Penea

`LinearLayout`

- Composants disposés en **lignes** ou en **colonnes**
- Donne la même taille à chaque composant
- On peut donner un `weight` a un componsant



`RelativeLayout`

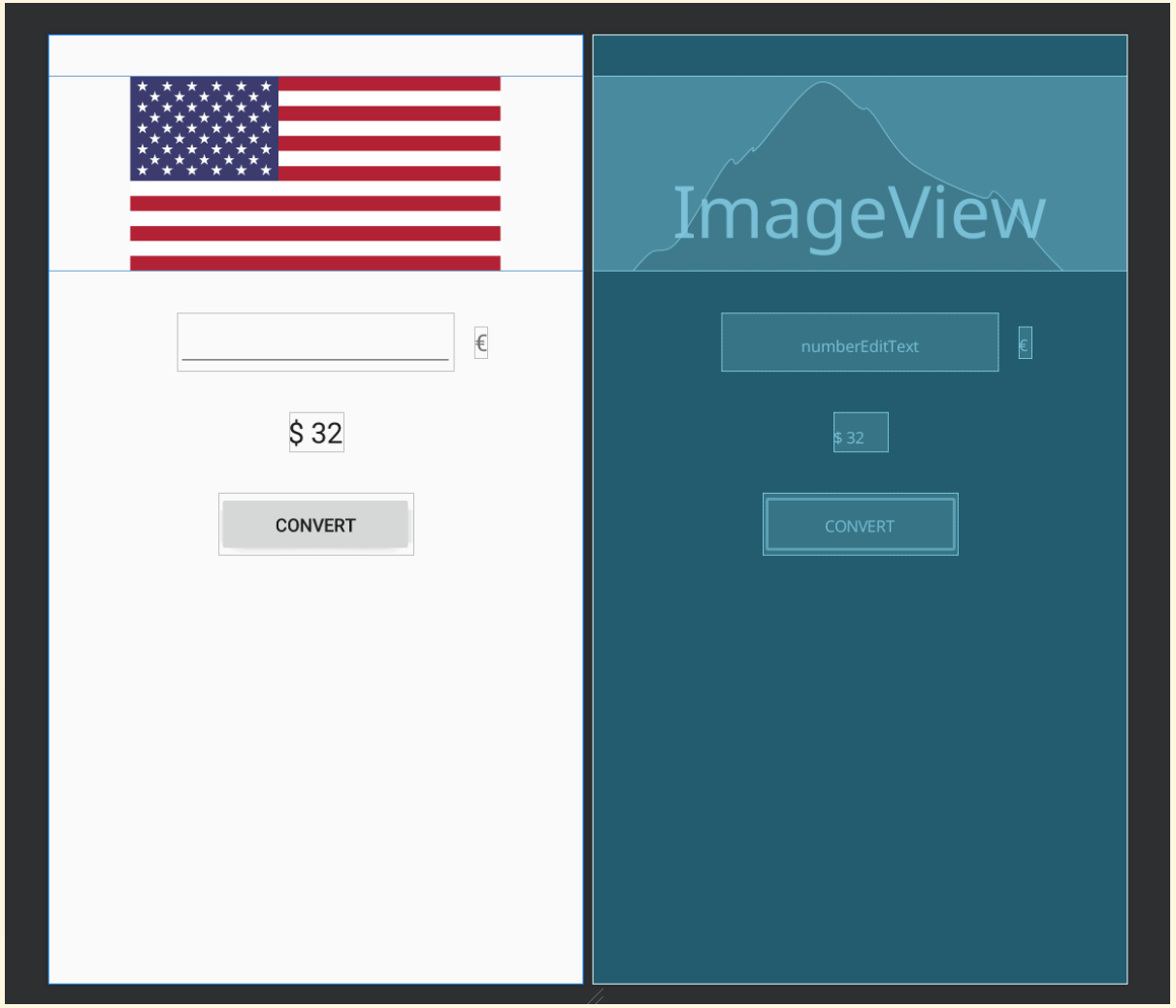
- L'ancêtre du `ConstraintLayout`
- Chaque élément est placé **relativement** aux autres
- **Préférez largement le `ConstraintLayout`**

`ConstraintLayout`

- `RelativeLayout` sous amphétamines
- On exprime des **contraintes** entre éléments
- `ConstraintLayout` essaie de respecter ces contraintes

TP 2

Contruire l'UI de conversion de monnaie



Robin Penea

Interactions UI

Pour rendre l'UI dynamique, il faut avoir une référence vers les éléments (TextView, Button, etc)

Robin Penea

Etapes interactions UI

1. Définir un id pour l'élément graphique (ex: `submitButton`, `resultTextView`, `flagImageView`)
2. Retrouver l'élément en code à partir de son id

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView resultTextView = findViewById(R.id.resultTextView);
    }
}
```

Etapes interactions UI

3. Modifier l'élément à partir de cette référence

```
TextView resultTextView = findViewById(R.id.resultTextView);
resultTextView.setText("Hey YOU! Look at ME!");
```

S'abonner à un évènement UI

Classe anonyme (plus couteux)

```
final Button myButton = findViewById(R.id.myButton);

myButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Log.i("MainActivity", "Click!");
    }
});
```

S'abonner à un évènement UI

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // ...
        findViewById(R.id.myButton).setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.myButton:
                Log.i("MainActivity", "Click!");
            break;
        }
    }
}
```

TP 3

Faire la conversion de monnaie

Objectif

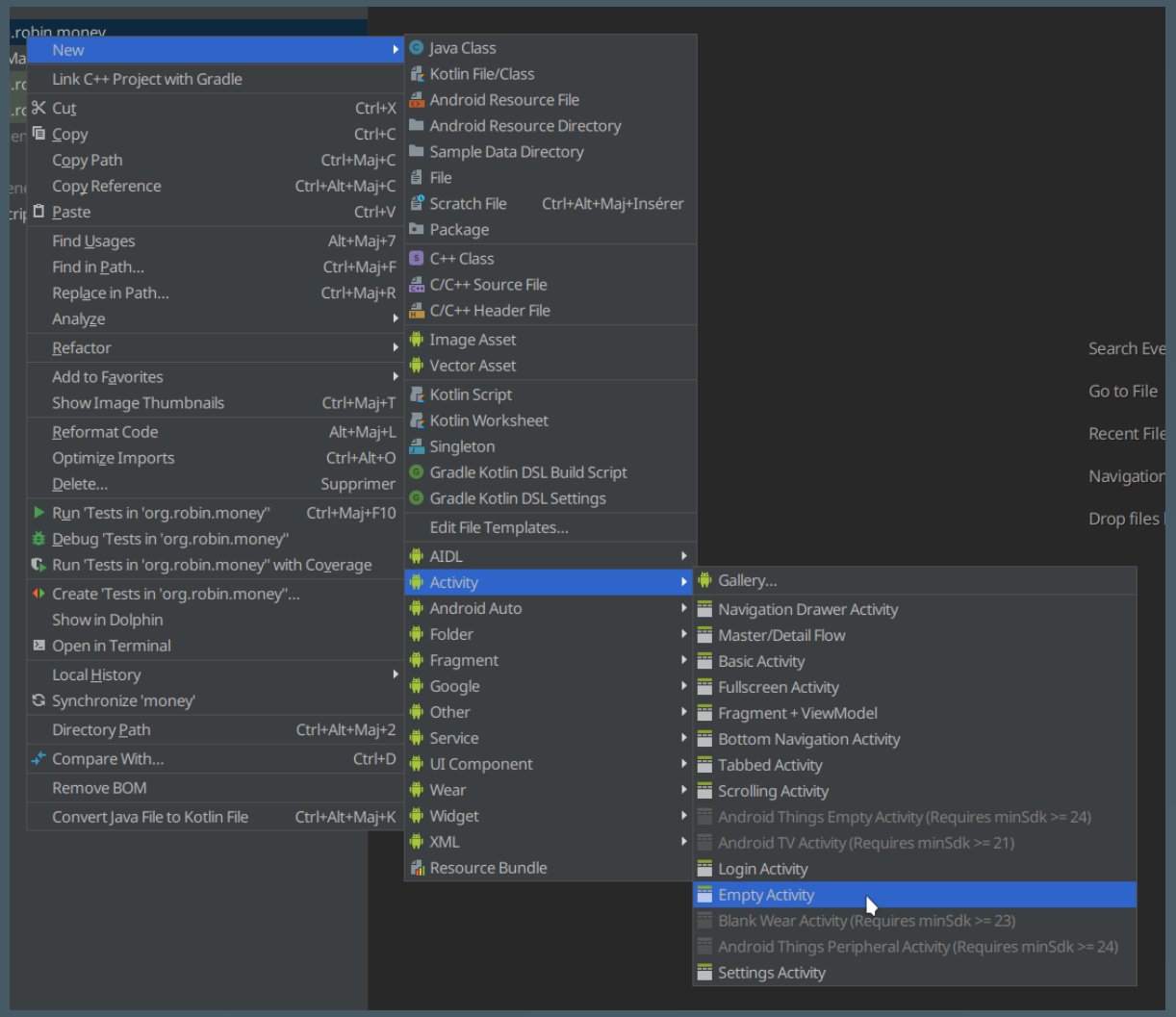
- Récupérer la valeur de l'`EditText`
- Faire le calcul euros => dollars
- Afficher le résultat dans un `TextView`

Robin Penea

Plusieurs Activity

- Pour naviguer entre plusieurs écrans, vous allez créer plusieurs Activity
- 1 Activity a toujours :
  - 1 classe Java
  - Un layout XML
  - Une définition dans l'`AndroidManifest.xml`

⚠️ Toujours créer une Activity comme ceci



Robin Penea

Navigation

- Vous ne faites **jamais** un `new MonActivity()`
  - C'est toujours Android qui le fera pour vous
- Pour démarrer une Activity, il faut décrire à l'OS son **intention**
- C'est Android qui se charge de créer et d'afficher l'Activity
- Il nous appelera dans la fonction `onCreate()` au bon moment

startActivity()

```
// Code dans une Activity

// this => Le contexte courant de l'Activity
// DestinationActivity.class => L'Activity que l'on veut lancer
Intent intent = new Intent(this, DestinationActivity.class);

startActivity(intent);
```

Activity Stack

- Une nouvelle Activity va **s'empiler** sur l'activity courante
- Un back va **dépiler** l'Activity courante
- On quitte l'application lors du back de la 1ère Activity

TP 4

Afficher un écran "A propos"

Objectif

- Ajouter un bouton "A propos"
- Lancer une nouvelle `AboutActivity` lors du clic sur le bouton
- `AboutActivity` affiche
  - la version de l'app
  - votre prénom / nom
  - Votre email

Robin Penea

Transmettre des informations

- Vous pouvez ajouter des données à un `Intent` en **extra** avec un système de clé / valeur
- Ces données pourront être lues par l'Activity lancée

Transmettre des informations

```java
// Ajout des information (Activity source)
Intent intent = new Intent(this, DestinationActivity.class);
intent.putExtra("aNumber", 42); // clé, valeur
intent.putExtra("aString", "Bob"); // clé, valeur

// Lecture des informations (Activity destination)
protected void onCreate(Bundle savedInstanceState) {
    // ...
    Intent srcIntent = getIntent();
    int n = srcIntent.getIntExtra("aNumber", 0); // clé, default
    String s = srcIntent.getStringExtra("aString"); // clé
}
```

Transmettre des objets

- Pour transférer un objet custom, la classe doit implémenter l'interface `Parcelable`
- Android ne connait pas votre classe, mais vous lui indiquez comment sérialiser / déserialiser la classe

Transmettre des objets

```
public class User implements Parcelable {
    String name;
    int age;

    protected User(Parcel in) {
        name = in.readString();
        age = in.readInt();
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(name);
        dest.writeInt(age);
    }

    @Override
    public int describeContents() { return 0; }

    public static final Creator<User> CREATOR = new Creator<User>() {
        @Override
        public User createFromParcel(Parcel in) { return new User(in); }
        @Override
        public User[] newArray(int size) { return new User[size]; }
    };
```

On peut (heureusement) générer tout ça...

Transmettre des objets

```
// serialiser le user
User user = new User("Bob", 20);

Intent intent = new Intent(this, DestinationActivity.class);
intent.putExtra("user", user); // user est Parcelable, c'est bon !

// désérialiser le user
User u = getIntent().getParcelableExtra("user");
```

TP 5

Choisir la monnaie de conversion

Objectif

- Créer une Activity `CurrencyChooserActivity` qui propose 3 monnaies : dollars, yen, pounds
- Définir `CurrencyChooserActivity` comme point d'entrée de l'app
- Lorsque l'utilisateur clique sur une monnaie, ouvrir `MainActivity` avec le bon drapeau ainsi que le bon taux de change pour la conversion

Robin Penea

Cycle de vie Activity

Android est multi-tâches et les développeurs doivent en tenir compte

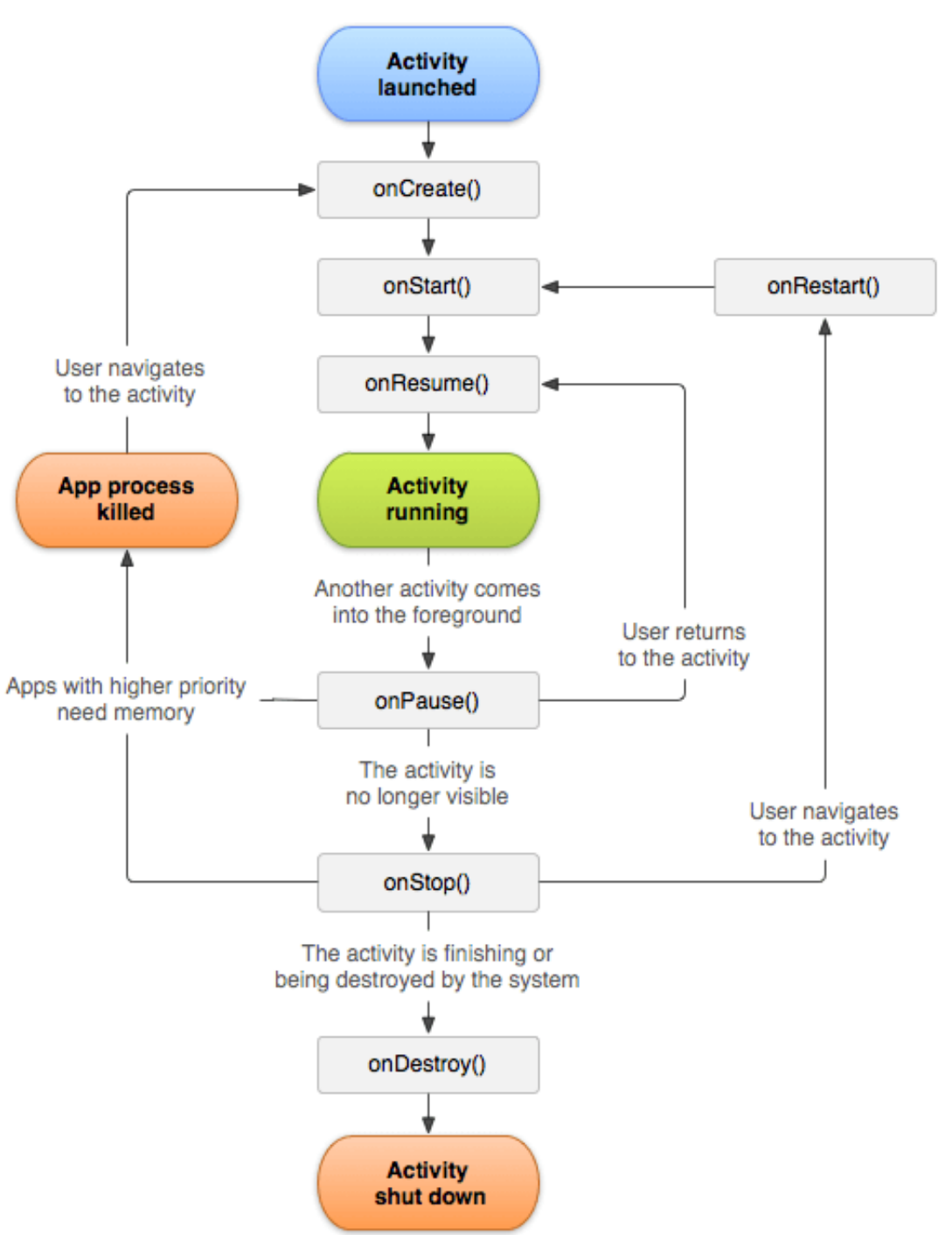
Robin Penea

Cycle de vie Activity

Use case classique

1. L'utilisateur est dans votre application
2. Un appel est reçu et l'application téléphone passe en foreground
3. L'appel se termine et votre application revient en foreground

Que s'est-il passé avec votre application ?


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Cycle de vie Activity

```
public class MainActivity extends AppCompatActivity {
    private static final String TAG = MainActivity.class.getSimpleName();

    @Override
    protected void onPause() {
        super.onPause();
        Log.i(TAG, "onPause");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.i(TAG, "onResume");
    }
}
```

TP 7

Afficher les états dans le logcat

Objectif

- Faire un log pour dans `CurrencyChooserActivity` et `MainActivity` pour les états suivants :
- `onCreate()`
- `onPause()`
- `onResume()`
- `onStop()`

Robin Penea

RecyclerView

Robin Penea

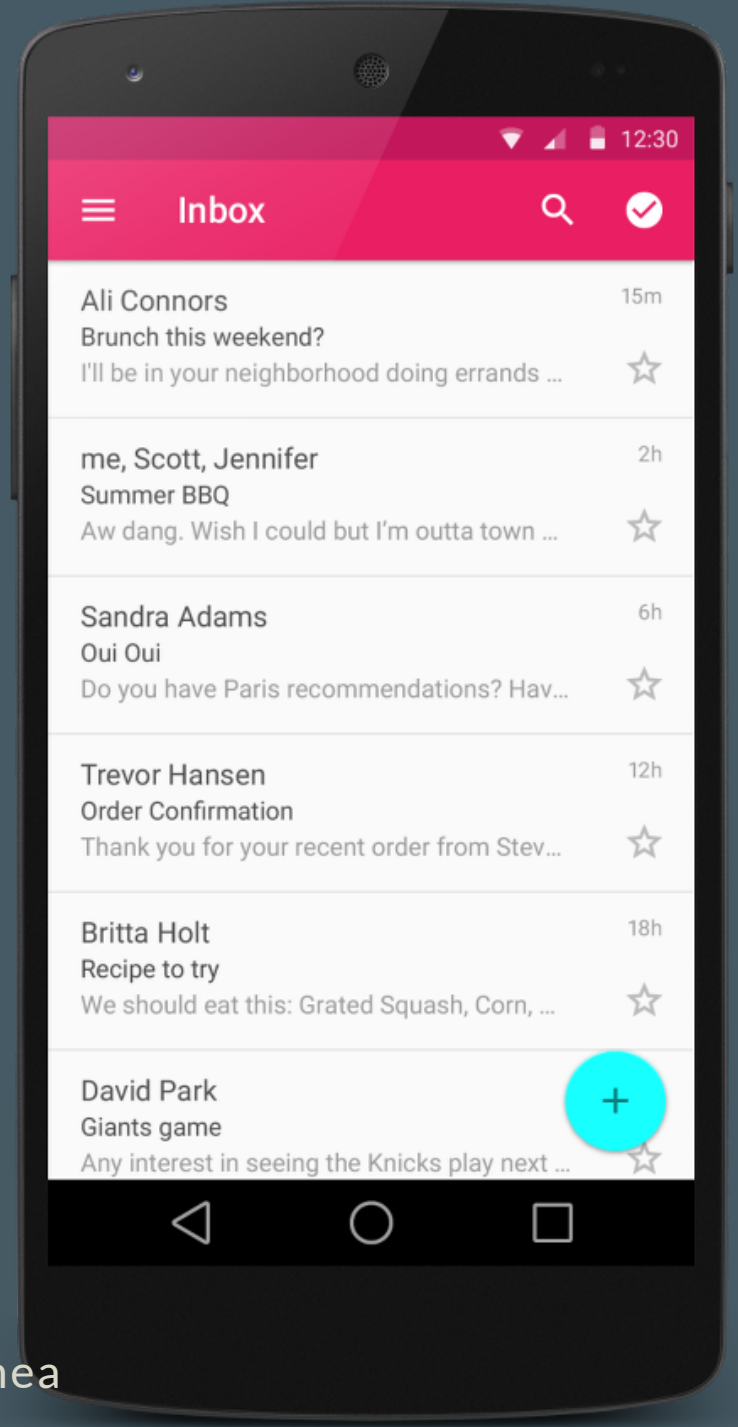
Le composant graphique le plus complexe d'Android

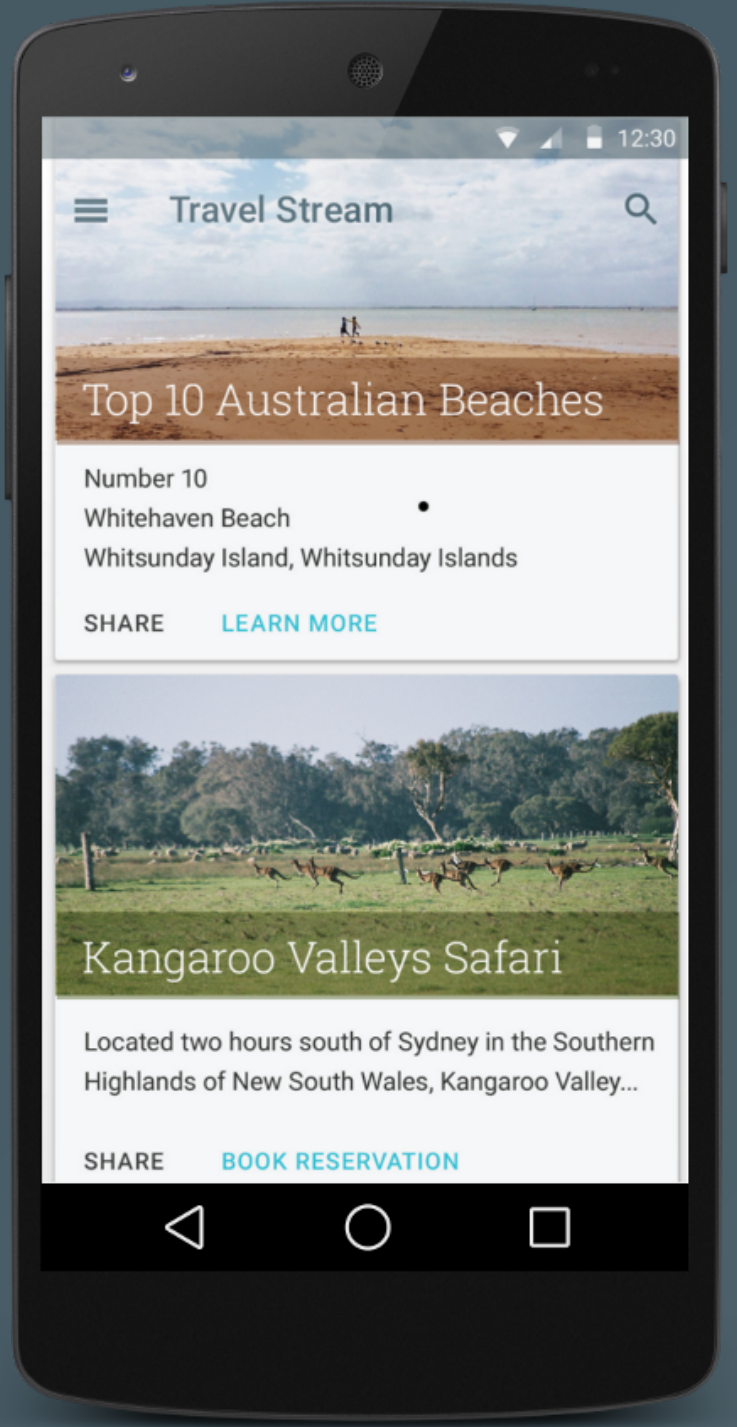
Robin Penea

(On oublie les fragments qui sont pénibles en plus d'être complexes)

Un layout pour afficher dynamiquement une liste d'éléments

Robin Penea


12:30
Inbox
Ali Connors
15m
Brunch this weekend?
I'll be in your neighborhood doing errands ...
me, Scott, Jennifer
2h
Summer BBQ
Aw dang. Wish I could but I'm outta town ...
Sandra Adams
6h
Oui Oui
Do you have Paris recommendations? Hav...
Trevor Hansen
12h
Order Confirmation
Thank you for your recent order from Stev...
Britta Holt
18h
Recipe to try
We should eat this: Grated Squash, Corn, ...
David Park
Giants game
Any interest in seeing the Knicks play next ...


12:30
Travel Stream
Top 10 Australian Beaches
Number 10
Whitehaven Beach
Whitsunday Island, Whitsunday Islands
SHARE
LEARN MORE
Kangaroo Valleys Safari
Located two hours south of Sydney in the Southern Highlands of New South Wales, Kangaroo Valley...
SHARE
BOOK RESERVATION

Robin Penea

Contraintes de l'embarqué

Robin Penea

Il faut minimiser les allocations mémoires

Pour conserver un défilement fluide

Le RecyclerView va donc

recycler les vues lors du défilement

RecyclerView

3 composants

- Une source de données : `List<String>`
- Un Adapter :
  `class CountryAdapter extends RecyclerView.Adapter { .. }`
- Une vue
  - RecyclerView : `<RecyclerView />`
  - Un layout par item de la list `item_country.xml`

Toute la logique se joue dans l'Adapter

Il est responsable de s'alimenter sur la source de données

et de fournir chaque élément d'item au RecyclerView

Structure d'un Adapter

```
class CountryAdapter : RecyclerView.Adapter<CountryAdapter.ViewHolder>() {

  private List<String> countries; // initialisé depuis le constructeur

  // class de "Cache" faisant le lien avec la View affichée
  class ViewHolder extends RecyclerView.ViewHolder { ... }

  // Crée et renvoie le ViewHolder
  public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) { ... }

  // Associe le ViewHolder à une donnée à l'index position
  public onBindViewHolder(ViewHolder holder, int position) { ... }

  // Renvoie la taille totale de la source de données
  public int getItemCount() { ... }
}
```

Exemple de layout d'un item

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="65dp">

    <ImageView android:id="@+id/icon" />
    <TextView android:id="@+id/name" />

</LinearLayout>
```

Détail de chaque étape

ViewHolder

```
public class ViewHolder extends RecyclerView.ViewHolder {
    final ImageView icon;
    final TextView name;
    public ViewHolder(@NonNull View itemView) {
        super(itemView);
        icon = itemView.findViewById(R.id.icon);
        name = itemView.findViewById(R.id.name);
    }
}
```

onCreateViewHolder

```
class CountryAdapter {
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        // Crée un View à partir du layout item_country et l'associe au ViewHolder
        View viewItem = LayoutInflater.from(parent.context)
                .inflate(R.layout.item_country, parent, false);
        return new ViewHolder(viewItem);
    }
}
```

onBindViewHolder

```
class CountryAdapter {
    void onBindViewHolder(Holder holder, int position) {
        String country = countries.get(position);
        holder.iconImageView.setImageResource(R.mipmap.ic_launcher_round);
        holder.nameTextView.setText(country);
    }
}
```

getItemCount

```
class CountryAdapter {
    public int getItemCount() {
        return countries.size();
    }
}
```

Le layout de la liste

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/countriesRecyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</RelativeLayout>
```

On fait la tuyauterie dans MainActivity

Robin Penea

```
class MainActivity extends AppCompatActivity implenets View.OnClickListener {

    List<String> countries = new ArrayList<String>(Arrays.listOf(
        "Afghanistan","Albania","Algeria","Andorra","Angola",
        "France","Georgia","Germany", "Vietnam","Zambia","Zimbabwe"
    ));
    private CountryAdapter adapter;

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

        // Création de l'adapter avec la source de données
        adapter = new CountryAdapter(countries);

        // Init du recycler view que l'on connecte à l'adatper
        RecyclerView recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        recyclerView.setAdapter(adapter);
    }
}
```

TP 8

- Créer une `CurrencyListActivity` affichant une liste de `Currency`
- Pour chaque `Currency`, afficher le drapeau, le symbole et le taux de change

Robin Penea