l'Application MyCurrentLocation

Introduction

L'application MyCurrentLocation permet à l'utilisateur d'obtenir sa position géographique actuelle en utilisant les services de localisation d'Android. L'application utilise le LocationManager pour accéder aux coordonnées GPS de l'utilisateur, et le Geocoder pour convertir ces coordonnées en une adresse lisible.

Structure du Code

1. Classe MainActivity

La classe MainActivity étend AppCompatActivity et implémente LocationListener, permettant à l'application de recevoir des mises à jour de localisation.

Déclarations des Variables

```
public class MainActivity extends AppCompatActivity  implements LocationListener {
    Button button_location ;  2 usages
    TextView text_viewLocation;  2 usages
    LocationManager locationManager;  2 usages
```

- **button_location** : Bouton pour demander la localisation actuelle.
- **text_viewLocation** : TextView pour afficher l'adresse actuelle.
- **locationManager** : Gère les mises à jour de localisation.

Méthode onCreate

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

    text_viewLocation=findViewById(R.id.textView);
    button_location=findViewById(R.id.button);
```

- Récupère les références du TextView et du Button à partir du layout.

Vérification des Permissions

```
if (ContextCompat.checkSelfPermission( context: MainActivity.this,
        android.Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED)
            {
                ActivityCompat.requestPermissions( activity: MainActivity.this,
                        new String[]{android.Manifest.permission.ACCESS_FINE_LOCATION},
                        requestCode: 100);
            }
```

- Vérifie si l'application a les permissions nécessaires pour accéder à la localisation. Si ce n'est pas le cas, elle demande la permission à l'utilisateur.

Gestion de l'Événement de Clic

```
button_location.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        getLocation();
    }
});
```

- Configure un écouteur de clic pour le bouton, qui appelle la méthode getLocation lorsque l'utilisateur clique dessus.

2. Méthode getLocation

```
@SuppressLint("MissingPermission") 1 usage
private void getLocation() {

    try {
        locationManager = (LocationManager) getApplicationContext().getSystemService(LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
                minTimeMs: 5000,
                minDistanceM: 5,
                listener: MainActivity.this);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- **@SuppressLint("MissingPermission")** : Indique que la méthode utilise des permissions potentiellement manquantes.
- Récupère le service de localisation et demande des mises à jour de localisation via GPS toutes les 5 secondes ou lorsque la position change d'au moins 5 mètres.

3. Méthode onLocationChanged

```
/**
 *
 * @param location
 */
@Override  4 usages
public void onLocationChanged(@NonNull Location location) {
    Toast.makeText( context: this, text: ""+location.getLatitude()+" , "+location.getLongitude(),
            Toast.LENGTH_SHORT).show();
    try {
        Geocoder geocoder =new Geocoder( context: MainActivity.this, Locale.getDefault());
        List<Address> adresses=geocoder.getFromLocation(location.getLatitude(),
                location.getLongitude(),
                 maxResults: 1);
        String adress=adresses.get(0).getAddressLine( index: 0);
        text_viewLocation.setText(adress);

    } catch (Exception e) {
        e.printStackTrace();
    }
```

- Cette méthode est appelée chaque fois que la localisation change.
- Elle affiche un toast avec la latitude et la longitude actuelles.
- Utilise le Geocoder pour obtenir l'adresse à partir des coordonnées et met à jour le TextView avec l'adresse.

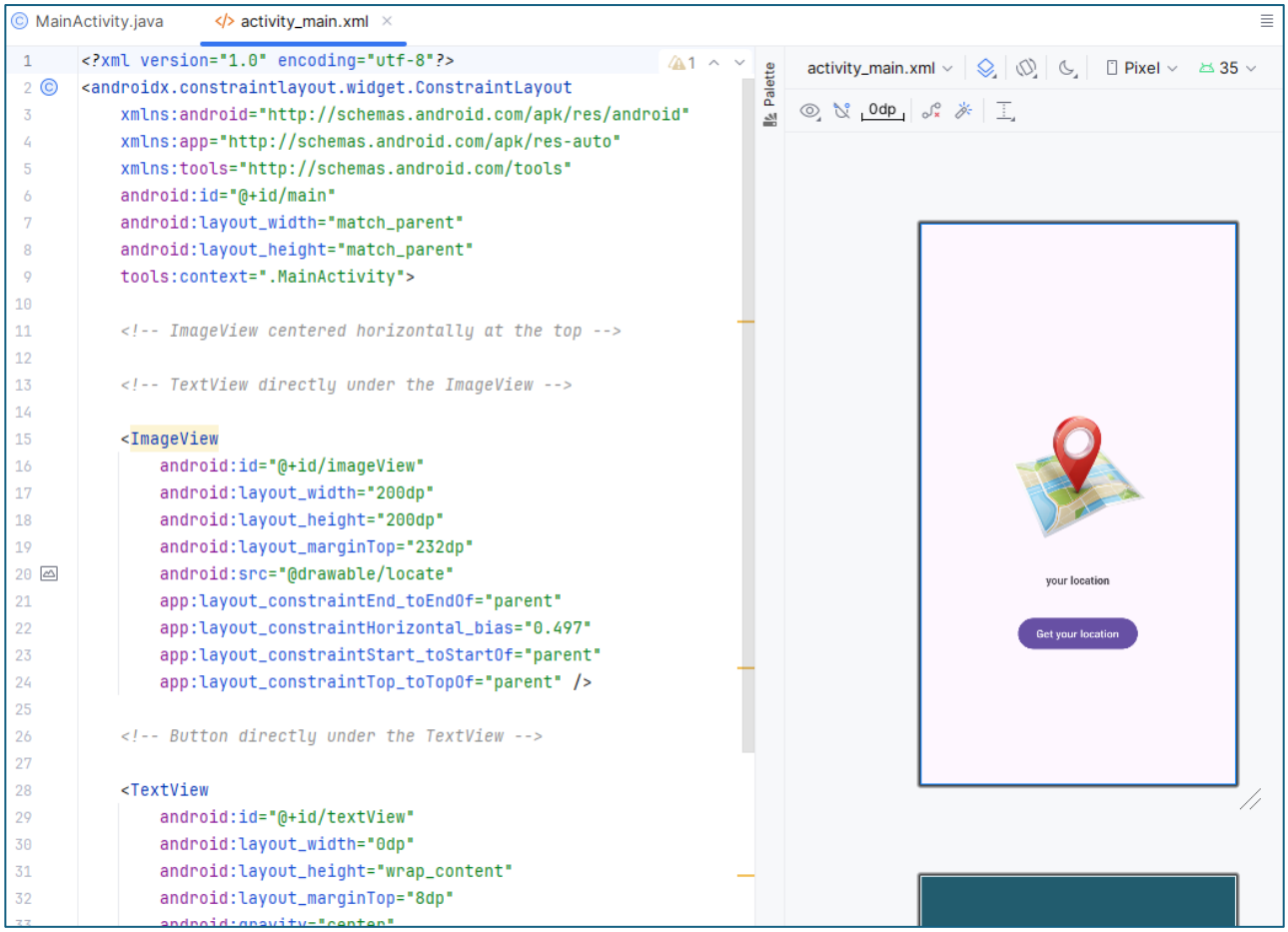
4. Méthodes supplémentaires de l'interface LocationListener

Ces méthodes sont implémentées mais ne sont pas utilisées dans cette application. Elles permettent de gérer des événements spécifiques liés à la localisation :

```java
@Override  3 usages
public void onLocationChanged(@NonNull List<Location> locations) {
    LocationListener.super.onLocationChanged(locations);
}

@Override  2 usages
public void onFlushComplete(int requestCode) {
    LocationListener.super.onFlushComplete(requestCode);
}

@Override  2 usages
public void onStatusChanged(String provider, int status, Bundle extras) {
    LocationListener.super.onStatusChanged(provider, status, extras);
}

@Override  2 usages
public void onProviderEnabled(@NonNull String provider) {
    LocationListener.super.onProviderEnabled(provider);
}

@Override  2 usages
public void onProviderDisabled(@NonNull String provider) {
    LocationListener.super.onProviderDisabled(provider);
}

@Override  2 usages
public void onPointerCaptureChanged(boolean hasCapture) {
    super.onPointerCaptureChanged(hasCapture);
}
```

Fichier XML - activity_main.xml

Le fichier XML définit l'interface utilisateur de l'application. Voici les éléments principaux :



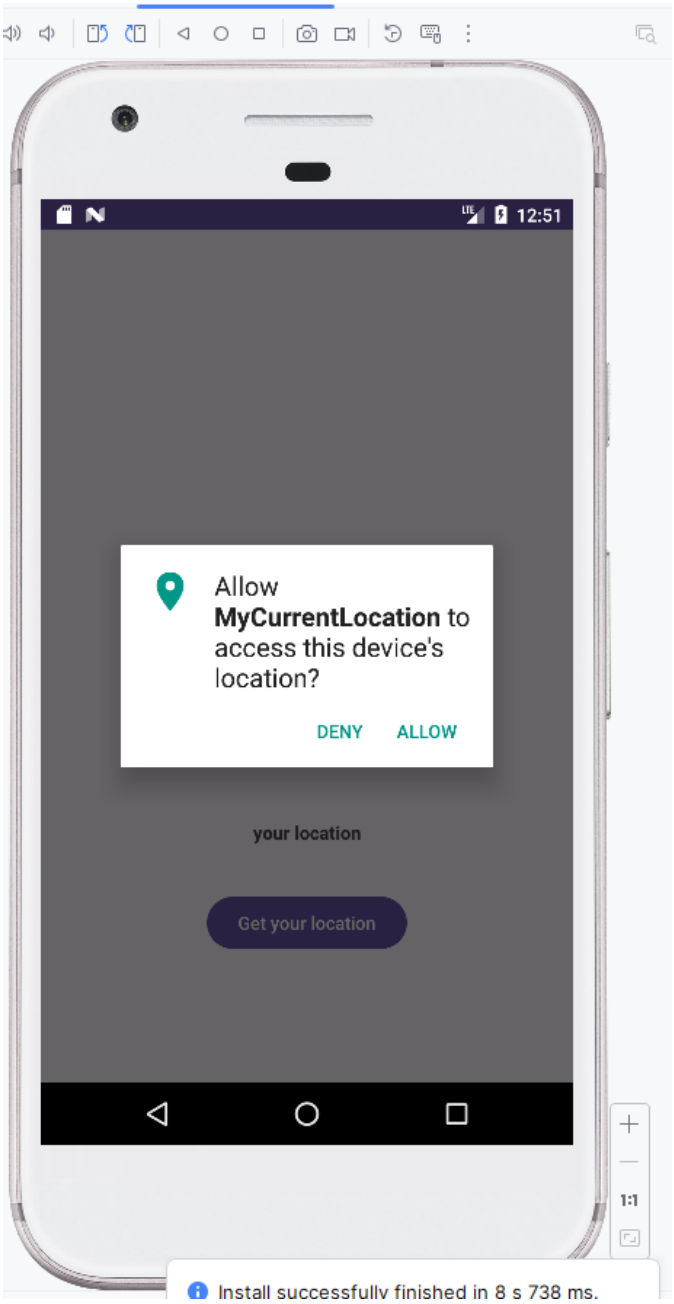
Permissions

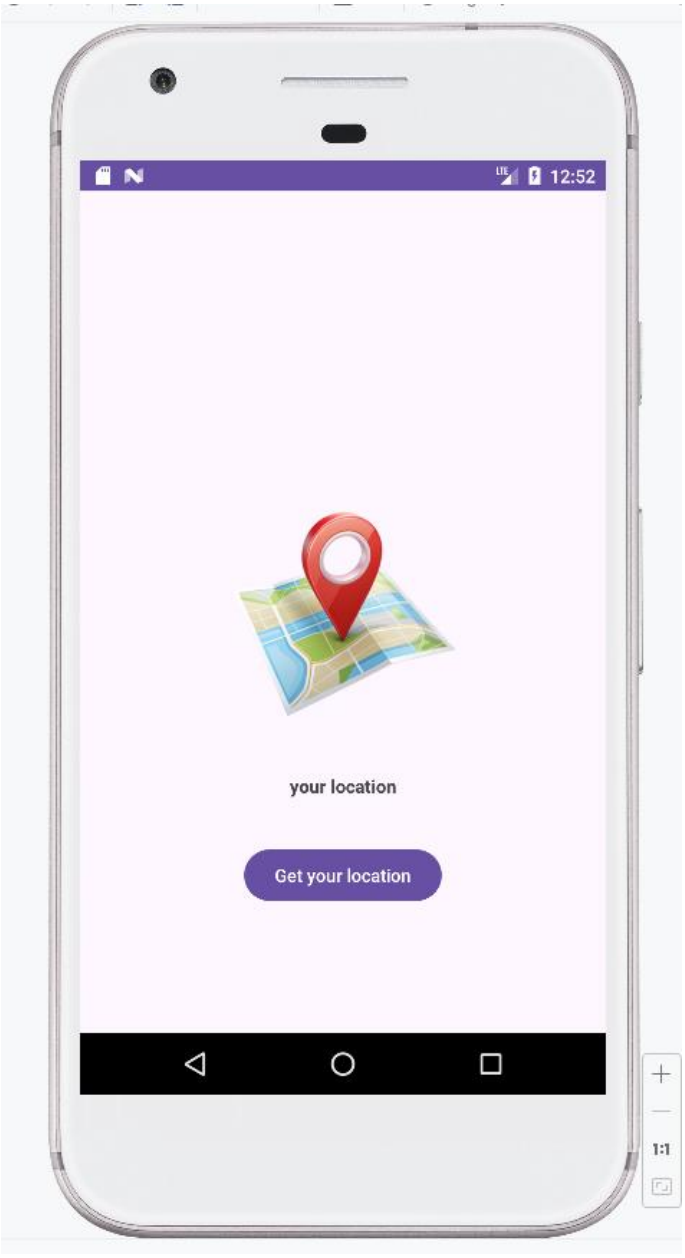
Le fichier AndroidManifest.xml contient les permissions nécessaires pour accéder à Internet et à la localisation :

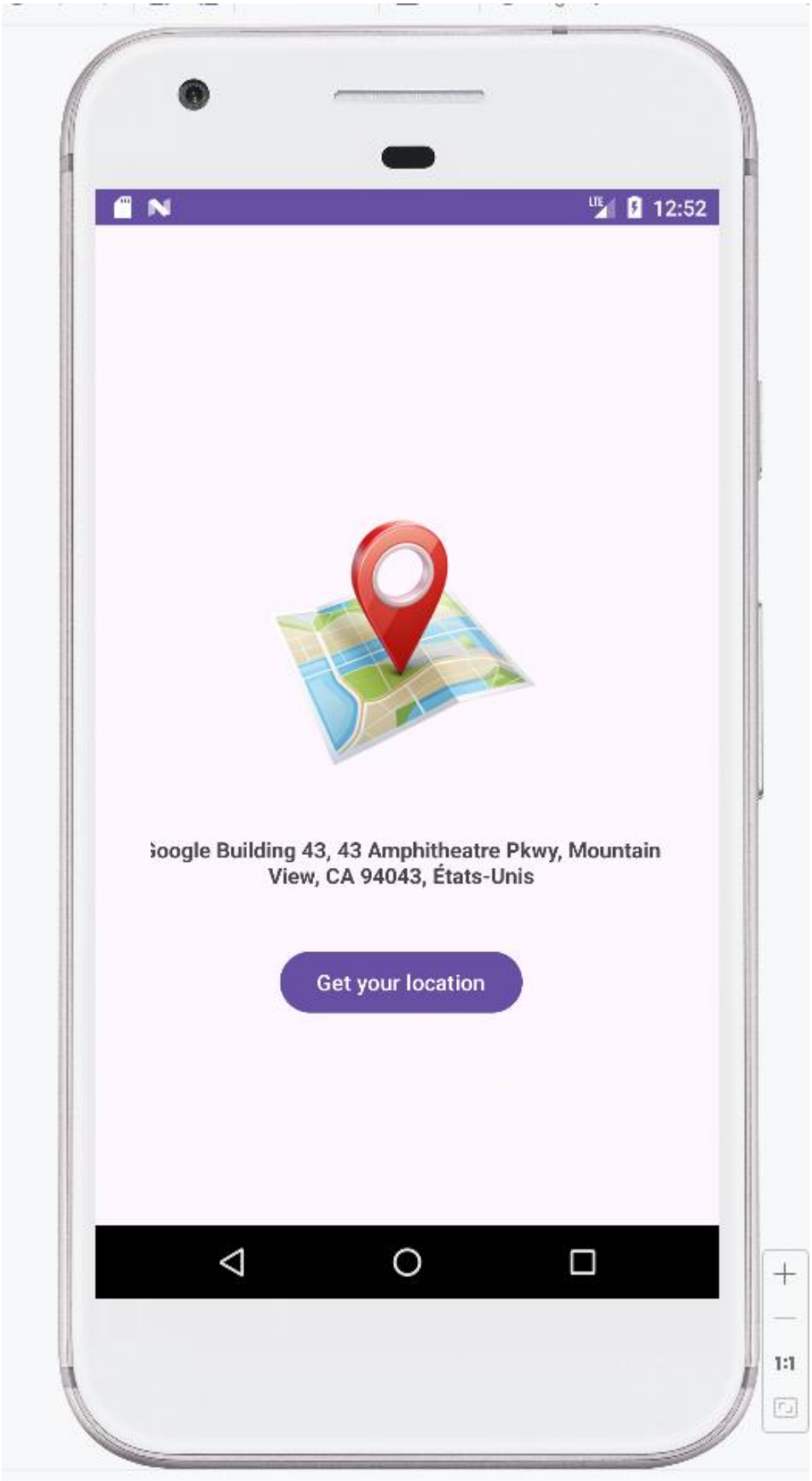
```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Conclusion

L'application MyCurrentLocation est un bon exemple d'utilisation des services de localisation d'Android pour obtenir et afficher la position actuelle de l'utilisateur. Elle gère correctement les permissions et utilise des composants UI pour afficher des informations à l'utilisateur. Des améliorations peuvent inclure une gestion plus robuste des erreurs et une interface utilisateur plus raffinée.


12:51
Allow
MyCurrentLocation to
access this device's
location?
DENY
ALLOW
your location
Get your location
1:1
Install successfully finished in 8 s 738 ms.


12:52
your location
Get your location
1:1


12:52
oogle Building 43, 43 Amphitheatre Pkwy, Mountain View, CA 94043, États-Unis
Get your location
1:1