



UTILISATION DES MENUS ET SOUS-MENUS

PROGRAMMATION MOBILE(FAGROUD)

166

Options Menu

OptionsMenu : Il s'agit d'un menu qui apparaît dans la barre d'application (action bar) ou en haut de l'écran.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/action_settings"
    android:title="Paramètres" />
  <item
    android:id="@+id/action_about"
    android:title="À propos" />
</menu>
```

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.action_settings) {
        // Code pour l'action des paramètres
        return true;
    }
    else if (id == R.id.action_about) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
```

2024-2025

PROGRAMMATION MOBILE(FAGROUD)

167

ContextMenu

ContextMenu : Ce menu apparaît lorsque l'utilisateur effectue une pression longue sur un élément.

```
registerForContextMenu(listView);
```

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    menu.add("Option 1");
    menu.add("Option 2");
}

no usages
@Override
public boolean onContextItemSelected(MenuItem item) {
    // Gère les actions pour les éléments du menu
    return super.onContextItemSelected(item);
}
```

PopupMenu

PopupMenu : Menu contextuel apparaissant comme une fenêtre contextuelle sur l'écran, lié à une action spécifique.

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

    // Initialisation du bouton
    Button button = findViewById(R.id.popupButton);

    // Configuration du menu popup
    button.setOnClickListener(v -> {
        PopupMenu popupMenu = new PopupMenu(context, MainActivity.this, v);

        // Ajout des éléments au menu
        popupMenu.getMenuInflater().inflate(R.menu.popup_menu, popupMenu.getMenu());

        // Affiche le menu
        popupMenu.show();
    });
}
```

Sous-menus

- ❖ Les sous-menus permettent de regrouper des actions similaires ou connexes.
- ❖ Ils peuvent être définis dans le fichier XML en utilisant l'élément <menu> imbriqué.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/action_settings"
    android:title="Paramètres" >
    <menu>
      <item
        android:id="@+id/sub_menu_item1"
        android:title="Sous-menu 1" />
      <item
        android:id="@+id/sub_menu_item2"
        android:title="Sous-menu 2" />
    </menu>
  </item>
  <item
    android:id="@+id/action_about"
    android:title="À propos" />
</menu>
```



TP 9

Énoncé

Objectif :

- Créer une activité qui utilise une Toolbar personnalisée pour afficher un menu d'options

Menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/tests" android:title="Tests" />
  <item
    android:id="@+id/ressources"
    android:title="Ressources" >
    <menu>
      <item
        android:id="@+id/coursera"
        android:title="Coursera" />
      <item
        android:id="@+id/ressource2"
        android:title="Ressource 2" />
    </menu>
  </item>
</menu>
```

toolbar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/black"
        app:title="LEARNING APP"
        app:titleTextColor="@android:color/white"
    />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Intent implicites

- ❖ Un Intent implicite permet de demander une action sans indiquer l'application exacte à utiliser. Android recherche alors les applications capables de gérer cette action.

```
Intent intent = new Intent(Intent.ACTION_VIEW);
intent.setData(Uri.parse("https://www.google.com"));
startActivity(intent);
```

toolbar.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.toolbar);
    // Initialisation de la Toolbar
    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
}

5 usages
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu, menu);
    return true;
}

4 usages
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.tests) {
        Intent intent1 = new Intent( packageContext toolbar.this, homepage.class);
        startActivity(intent1);
        return true;
    }
}
```

```
if (id == R.id.coursersa) {
    // Action pour "à propos"
    Intent intent = new Intent(Intent.ACTION_VIEW);
    intent.setData(Uri.parse( UriString "https://www.coursersa.org"));
    startActivity(intent);
    return true;
}

return super.onOptionsItemSelected(item);
}
```



PERSISTANCE DES DONNÉES

Firestore

❖ Firestore est une plateforme développée par Google, qui offre divers outils et services pour aider les développeurs à créer, améliorer et développer leurs applications.

❖ Principales fonctionnalités :

- **Base de données en temps réel** : Synchronisation instantanée des données entre utilisateurs via une base NoSQL.
- **Authentification** : Gestion simplifiée des connexions (e-mail, téléphone, réseaux sociaux).
- **Cloud Firestore** : Base NoSQL évolutive pour des applications complexes.
- **Fonctions cloud** : Exécution de code backend sans gérer des serveurs.
- **Hébergement** : Hébergement rapide et sécurisé pour applications web.
- **Analyse** : Suivi du comportement des utilisateurs et performance des applications.
- **Notifications Push** : Envoi de notifications ciblées et personnalisées.
- **Crashlytics** : Surveillance et correction des plantages d'applications.

Firestore

❖ **Configuration de Firestore**

- ✓ Créer un nouveau projet dans Firestore Console <https://console.firebase.google>.
- ✓ Suivez les instructions pour ajouter une application Android à votre projet Firestore \
- ✓ Télécharger le fichier de configuration google-services.json



TP 10

2024-2025

PROGRAMMATION MOBILE(FAGROUD)

180

Résultat attendue

Adresse e-mail
Mot de passe
Se connecter

Email
Mot de passe
Se connecter
Mot de passe oublié ?
S'inscrire

11:09
LEARNING APP
Tests
Ressources
Se deconnecter
Bienvenue : f.fagroud@ensi.ma

2024-2025

PROGRAMMATION MOBILE(FAGROUD)

181

login.java

```
public class logout extends AppCompatActivity {
    3 usages
    private FirebaseAuth auth;
    3 usages
    private EditText emailField, passwordField;
    2 usages
    private Button loginBtn, logoutBtn;
    2 usages
    private TextView register;

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

        // Initialisation Firebase
        auth = FirebaseAuth.getInstance();

        // Initialisation des vues
        emailField = findViewById(R.id.emailField);
        passwordField = findViewById(R.id.passwordField);
        register = findViewById(R.id.registerText);
        loginBtn = findViewById(R.id.loginBtn);
        // logoutBtn = findViewById(R.id.logoutBtn);

        // Boutons de clic
        register.setOnClickListener(v -> registerUser());
        loginBtn.setOnClickListener(v -> loginUser());
        // logoutBtn.setOnClickListener(v -> logoutUser());

        checkUserSession();
    }
}
```

```
// ♦ Vérifier si un utilisateur est déjà connecté
1 usage
private void checkUserSession() {
    FirebaseUser user = auth.getCurrentUser();
    if (user != null) {
        Toast.makeText(context, this, "Utilisateur connecté : " + user.getEmail(), Toast.LENGTH_SHORT).show();
    }
}

// ♦ Inscription avec email et mot de passe
1 usage
private void registerUser() {
    Intent intent1 = new Intent(packageContext, inscription.class);
    startActivity(intent1);
}
}
```

2024-2025

PROGRAMMATION MOBILE(FAGROUD)

182

login.java

```
// ♦ Connexion avec email et mot de passe
1 usage
private void loginUser() {
    String email = emailField.getText().toString();
    String password = passwordField.getText().toString();

    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(task -> {
            if (task.isSuccessful()) {
                //Toast.makeText(logout.this, "Connexion réussie !", Toast.LENGTH_SHORT).show();
                Intent intent1 = new Intent(packageContext, toolbar.class);
                startActivity(intent1);
            } else {
                Toast.makeText(context, logout.this, "Erreur : " + task.getException().getMessage(), Toast.LENGTH_SHORT).show();
                emailField.setError("Vérifiez votre email");
                passwordField.setError("Vérifiez votre mot de passe");
            }
        });
}
}
```

2024-2025

PROGRAMMATION MOBILE(FAGROUD)

183

register.java

```
public class Inscription extends AppCompatActivity {
    2 usages
    private EditText emailEditText, passwordEditText;
    2 usages
    private Button registerButton;
    3 usages
    private FirebaseAuth auth;

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

        emailEditText = findViewById(R.id.emailEditText);
        passwordEditText = findViewById(R.id.passwordEditText);
        registerButton = findViewById(R.id.registerButton);

        auth = FirebaseAuth.getInstance();

        registerButton.setOnClickListener(v -> {
            String email = emailEditText.getText().toString().trim();
            String password = passwordEditText.getText().toString().trim();

            if (!email.isEmpty() && !password.isEmpty()) {
                registerUser(email, password);
            } else {
                Toast.makeText(this, "Veuillez remplir tous les champs", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

```
usage
private void registerUser(String email, String password) {
    auth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(task -> {
            if (task.isSuccessful()) {
                FirebaseAuth user = auth.getCurrentUser();
                Toast.makeText(this, "Inscription réussie : " + user.getEmail(), Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "Erreur : " + task.getException().getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
}
```

2024-2025

PROGRAMMATION MOBILE(FAGROUD)

184

Menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/tests" android:title="Tests" />
    <item
        android:id="@+id/ressources"
        android:title="Ressources" >
        <menu>
            <item
                android:id="@+id/coursera"
                android:title="Coursera" />
            <item
                android:id="@+id/ressource2"
                android:title="Ressource 2" />
        </menu>
    </item>
    <item
        android:id="@+id/con" android:title="Se deconnecter" />
</menu>
```

2024-2025

PROGRAMMATION MOBILE(FAGROUD)

185

toolbar.java

Modification 1

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.toolbar);
    // Initialisation de la Toolbar
    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    TextView Text=findViewById(R.id.text);
    FirebaseAuth user = FirebaseAuth.getInstance().getCurrentUser();
    if (user != null) {
        String email = user.getEmail();
        //Toast.makeText(this, "Email de l'utilisateur : " + email, Toast.LENGTH_SHORT).show();
        Text.setText("Bienvenue : " + email);
    } else {
        Toast.makeText(context, this, text: "Aucun utilisateur connecté", Toast.LENGTH_SHORT).show();
    }
}
```

Modification 2

```
if (id == R.id.con) {
    FirebaseAuth auth = FirebaseAuth.getInstance();
    auth.signOut();
    Toast.makeText(context, this, text: "Déconnexion réussie", Toast.LENGTH_SHORT).show();

    // Rediriger l'utilisateur vers la page de connexion ou d'accueil
    Intent intent = new Intent(packageContext, this, logout.class);
    startActivity(intent);
    finish();
}
```

Menus de navigation

Navigation Drawer :

- ❖ Utilisé pour afficher un menu latéral coulissant avec des options principales de navigation. Voici quelques caractéristiques clés :
 - Accessible via un bouton (souvent appelé hamburger menu) ou un swipe.
 - Idéal pour organiser des options nombreuses ou secondaires.
- ❖ Implémenté avec un DrawerLayout et un NavigationView.

Bottom Navigation :

- ❖ Un menu de navigation placé au bas de l'écran
- ❖ Accessible en permanence à l'utilisateur
- ❖ utilise le composant BottomNavigationView

Points à considérer pour choisir entre les deux :

- ❖ Si votre application a moins de 5 sections importantes, la Bottom Navigation est idéale.
- ❖ Si elle a de nombreuses sections ou des fonctionnalités secondaires, le Navigation Drawer peut être un meilleur choix.

Menus de navigation

main_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- Contenu principal -->
    <FrameLayout
        android:id="@+id/content_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <!-- Navigation Drawer -->
    <com.google.android.material.navigation.NavigationView
        android:id="@+id/navigation_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        app:menu="@menu/nav_menu" />

</androidx.drawerlayout.widget.DrawerLayout>
```

nav_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/nav_home"
        android:title="Accueil" />

    <item
        android:id="@+id/nav_settings"
        android:title="Paramètres" />

</menu>
```

main_activity.java

```
public class MainActivity extends AppCompatActivity {
    1 usage;
    private DrawerLayout drawerLayout;
    2 usages;
    private NavigationView navigationView;

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

        drawerLayout = findViewById(R.id.drawer_layout);
        navigationView = findViewById(R.id.navigation_view);

        navigationView.setNavigationItemSelectedListener(menuItem -> {
            return true;
        });
    }
}
```

2024-2025

PROGRAMMATION MOBILE (FAGROUD)

188

Menus de navigation

main_activity.xml

```
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottom_navigation"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    app:menu="@menu/bottom_nav_menu" />
```

bottom_nav_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/nav_home"
        android:title="Accueil" />

    <item
        android:id="@+id/nav_settings"
        android:title="Paramètres" />

    <item
        android:id="@+id/nav_profile"
        android:title="Profil" />

</menu>
```

main_activity.java

```
BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);
bottomNavigationView.setOnNavigationItemSelectedListener(menuItem -> {
    switch (menuItem.getItemId()) {
        case R.id.nav_home:
            // Logique pour Accueil
            break;
        case R.id.nav_settings:
            // Logique pour Paramètres
            break;
        case R.id.nav_profile:
            // Logique pour Profil
            break;
    }
    return true;
});
```

2024-2025

PROGRAMMATION MOBILE (FAGROUD)

189