

1 NavigationDrawer

Beim Nachforschen und beim Erstellen des NavigationDrawers wurden die Blogs [*How to Add a Navigation Drawer in Android* [1]], [*Android Sliding Menu using Navigation Drawer* [2]] und die Anleitung zu NavigationDrawers [*Creating a Navigation Drawer* [3]] verwendet.

Der NavigationDrawer ist ein Menü, das die wichtigsten Navigationsoptionen am linken Bildschirmrand darstellt. Diese Leiste ist meistens versteckt und wird angezeigt sobald der Benutzer mit dem Finger vom linken Rand des Bildschirms in Richtung Bildschirmmitte streicht oder auf das NavigationDrawerIcon klickt. Erneutes Klicken auf dieses Icon oder Streichen nach links versteckt den NavigationDrawer.

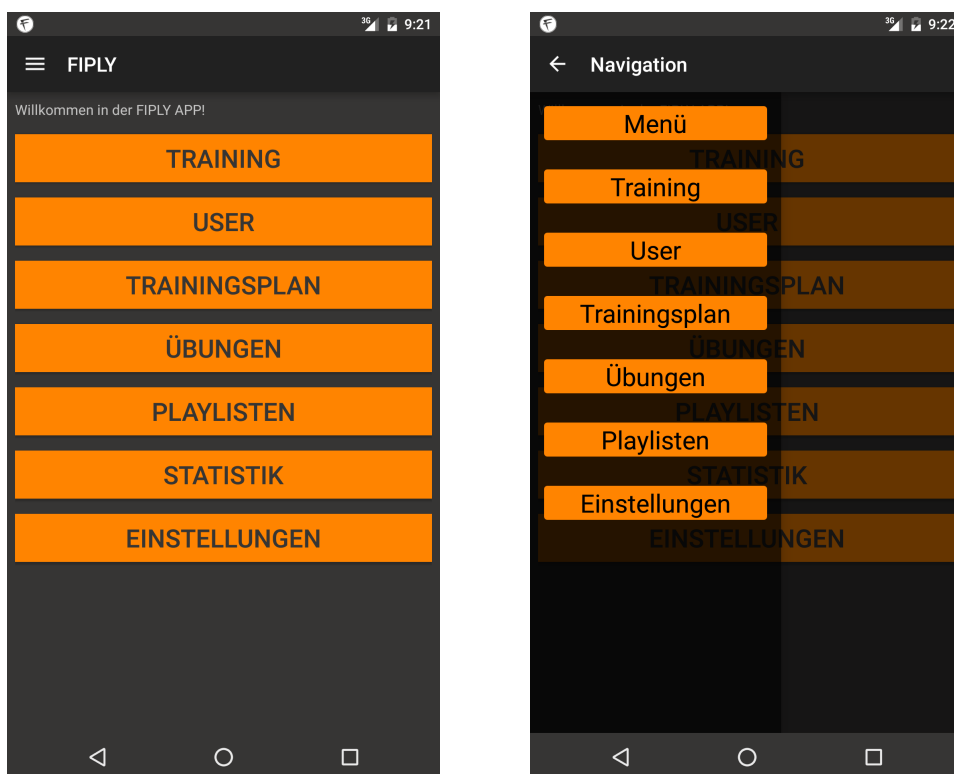


Abbildung 1: Links wird der NavigationDrawer versteckt. Rechts ist der NavigationDrawer geöffnet.

```

1 <android.support.v4.widget.DrawerLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/drawer_layout"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context="htl_leonding.fiplyteam.fiply.menu.MainActivity">
8
9   <FrameLayout
10    android:id="@+id/fraPlace"
11    android:layout_width="match_parent"
12    android:layout_height="match_parent" />
13
14   <ListView
15    android:id="@+id/navlist"
16    android:layout_width="250dp"
17    android:layout_height="match_parent"
18    android:layout_gravity="start"
19    android:background="@color/darkNavigationBackground"
20    android:divider="@color/darkNavigationDivider"
21    android:dividerHeight="1dp" />
22
23 </android.support.v4.widget.DrawerLayout>

```

Um einen NavigationDrawer zu einer App hinzuzufügen wird ein DrawerLayout als Root-View in dem layout XML file einer Activity deklariert. Eine Root-View beinhaltet alle anderen Views der App. In diesem DrawerLayout wird anschließend eine View angelegt, die dazu verwendet wird den Hauptinhalt der App darzustellen. Dabei ist zu Beachten, dass das die View die alle Elemente in sich trägt als erstes Kind des DrawerLayouts deklariert wird. Da diese View den ganzen Bildschirm befüllen soll müssen die Höhe und die Breite auf "match_parent" gestellt werden.

Zusätzlich wird eine weitere View hinzugefügt, die die Elemente des NavigationDrawers in sich trägt. Diese muss das Attribut layout_gravity überschreiben. Um right-to-left Sprachen zu unterstützen, soll "start" anstatt von "left" dafür spezifiziert werden. Die Höhe soll die Länge des ganzen Bildschirms umfassen, die Breite wird in Density-independent Pixels definiert und sollte 320dp nicht überschreiten da der Benutzer immer Teile des Hauptinhalts sehen soll.

In dieser Arbeit wird für den Hauptinhalt das FrameLayout fraPlace verwendet. Dieses Layout dient als Container für Fragments, die die einzelnen Ansichten der App darstellen. Die Elemente des NavigationDrawers werden in der ListView navlist definiert.

```

1 public class MainActivity extends AppCompatActivity {
2     ListView mDrawerList;
3     ArrayAdapter<String> mAdapter;
4     DrawerLayout mDrawerLayout;
5     String[] navArray = new String[7];
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11
12        mDrawerList = (ListView) findViewById(R.id.navlist);
13        mDrawerLayout = (DrawerLayout) findViewById(
14            R.id.drawer_layout);
15        navArray = res.getStringArray(R.array.navigationArray);
16
17        mDrawerList.setOnItemClickListener(new AdapterView
18            .OnItemClickListener() {
19
20            @Override
21            public void onItemClick(AdapterView<?> parent, View view,
22                int position, long id) {
23                displayView(position);
24            }
25        });
26
27        mAdapter = new ArrayAdapter<>(this, R.layout.navigation_list,
28            R.id.navlist_content, navArray);
29        mDrawerList.setAdapter(mAdapter)
30    }
31 }

```

Das ist der minimale Code, um einen funktionierenden NavigationDrawer zu erstellen. Will man das Design und Verhalten des Icons, zum Öffnen des NavigationDrawers, ändern, ist noch etwas Code nötig. Dieser Code wird ebenfalls im onCreate der Activity, die einen NavigationDrawer erhalten soll, ausgeführt.

```

1 mDrawerToggle = new ActionBarDrawerToggle(this, mDrawerLayout,
2     R.string.drawer_open, R.string.drawer_close) {
3     ...
4 };
5 mDrawerToggle.setDrawerIndicatorEnabled(true);
6 mDrawerLayout.addDrawerListener(mDrawerToggle);
7 getSupportActionBar().setDisplayHomeAsUpEnabled(true);
8 getSupportActionBar().setHomeButtonEnabled(true);

```