

Application d'Agenda avec gestion d'événements

Introduction

Le projet a été fait sous Android Studio en utilisant mon Honor 9 comme appareil Android pour exécuter les projets.

Description du projet

Ce projet est constitué de deux activités (MainActivity et AddEvent) et d'une classe (Evenement). Ces deux activités constituent les deux pages différentes de l'application. Une sert à afficher le calendrier ainsi que les événements prévus associés à chaque journée. L'autre sert à ajouter ces événements. Ce qui fait le lien entre les deux sont deux "Button". Dans un sens pour ajouter et dans l'autre pour valider l'ajout. Une fois qu'un "Button" a été cliqué, l'activité appelle l'autre par l'intermédiaire d'un intent, avec lequel on envoie les données nécessaires à l'autre activité.

MainActivity

Il s'agit de l'activité principale de notre application, c'est à dire celle qui affiche le calendrier. Pour cela, elle utilise une classe CalendarView qui est un widget permettant d'afficher un calendrier déjà rempli et fonctionnel mais qui n'implémente pas le principe d'événements associés.

```
1 public class MainActivity extends AppCompatActivity {
2     CalendarView calendar;
3     TextView dateView;
4     @Override
5     protected void onCreate(Bundle savedInstanceState) {
6         super.onCreate(savedInstanceState);
7         setContentView(R.layout.activity_main);
8         calendar = findViewById(R.id.calender);
9         dateView = findViewById(R.id.dateView);
10        calendar.setOnDateChangeListener(new CalendarView.
11        OnDateChangeListener() {
12            @Override
13            public void onSelectedDayChange(@NonNull CalendarView view, int
14            year, int month, int dayOfMonth) {
15                String Date = dayOfMonth + "—" + (month + 1) + "—" + year;
16                dateView.setText(Date);
17            }
18        })
19    }
20 }
```

```

16     }
17 }

```

C'est pour cela que l'on a la classe "Evenement". Elle nous permet de pouvoir créer un objet de type Evenement qui contient une date composée du jour, du mois et de l'année, de l'heure, du titre de l'événement et de sa description. Ainsi, on peut grâce à cela, créer une liste d'événements static qui contiendra tous les événements créés et ensuite sélectionner ceux que l'on veut afficher en fonction du jour sur lequel on a cliqué.

```

1  for (Evenement e : listEvenements){
2      if ((dayOfMonth == e.getDay()) && (month+1 == e.getMonth()) && (year ==
        e.getYear())) {
3          TextView txtEvent = new TextView(MainActivity.this);
4          txtEvent.setText("Date: " + e.getDay() + "/" + e.getMonth() +
5                          "/" + e.getYear() + " Time: " + e.getHour() +
6                          "h \nTitle: " + e.getTitle() +
7                          " Description: " + e.getDescr());
8          ll.addView(txtEvent);
9      }
10 }

```

AddEvent

Comme dit plus haut, cette activité sert à ajouter un événement à notre agenda. Comme nous avons à notre disposition la classe Evenement, cela se fait très facilement. Il suffit de demander à l'utilisateur de rentrer la date, l'heure, le titre de l'événement et sa description, le tout dans des "EditText". Une fois que l'utilisateur à fini, il peut appuyer sur le button Submit et donc envoyer avec un intent les informations saisies à MainActivity pour les ajouter dans la liste d'événements et ensuite les afficher.

```

1  public class AddEvent extends AppCompatActivity {
2
3      EditText date, time, title, description;
4
5      @Override
6      protected void onCreate(Bundle savedInstanceState) {
7          super.onCreate(savedInstanceState);
8          setContentView(R.layout.activity_add_event);

```

```

9
10     date = (EditText)findViewById(R.id.editTextDate);
11     time = (EditText)findViewById(R.id.editTextTime);
12     title = (EditText)findViewById(R.id.editText2);
13     description = (EditText)findViewById(R.id.editText);
14
15
16 }
17
18 public void submitEvent(View view){
19     Intent intent = new Intent(AddEvent.this , MainActivity.class);
20     intent.putExtra("date", date.getText().toString());
21     intent.putExtra("time", time.getText().toString());
22     intent.putExtra("title", title.getText().toString());
23     intent.putExtra("description", description.getText().toString());
24     startActivity(intent);
25 }
26 }

```

A Noter

Pour le bon fonctionnement de l'application, on ne peut rentrer que l'heure de l'événement et pas les minutes. Donc quelque chose de la forme 11h mais pas 11h04 par exemple. Aussi, si l'application est fermée une fois, on ne peut plus la réouvrir suite à un bug inconnu.