

Compétences Android : Bellec Fauchaux

- 1) Je sais décrire le contexte de mon application, pour que n'importe qui soit capable de comprendre à quoi elle sert :

Notre application est un jeu où le but est d'atteindre le plus de plateforme en sautant avec un petit personnage.

Pour que le personnage change de direction il faut utiliser le gyroscope de son téléphone.

- 2) Je sais faire un diagramme de cas d'utilisation pour mettre en avant les différentes fonctionnalités de mon application. :

Voir la documentation du projet.

- 3) Je sais concevoir un diagramme UML de qualité représentant mon application. :

Voir la documentation du projet

- 4) Je sais décrire un diagramme UML en mettant en valeur et en justifiant les éléments essentiels. :

Voir la documentation du projet

- 5) Je sais utiliser les Intent pour faire communiquer deux activités. :

Package activités (en particulier MainActivity)

```
Intent intent = new Intent(this, MainActivity.class);
startActivity(intent);
```

- 6) Je sais développer en utilisant le SDK le plus bas possible.

SDK avec 100% de coverage (API level 15)

- 7) Je sais distinguer mes ressources en utilisant les qualifier :

R.id / R.drawable

- 8) Je sais modifier le manifeste de l'application en fonction de mes besoins :

```
<activity android:name=".activities.ScoreActivity" />
<activity android:name=".activities.GameActivity"
android:configChanges="orientation" android:screenOrientation="portrait"/>
<activity android:name=".activities.OptionsActivity" />
```

- 9) Je sais faire des vues xml en utilisant layouts et composants adéquats :

Res/layout => Voir les vues du projet.

10) Je sais coder proprement mes activités, en m'assurant qu'elles ne font que relayer les évènements :

Package activities => Voir activités du projet

11) Je sais coder une application en ayant un véritable métier :

Package models => Voir classe Jumper et l'ensemble des classes métier

12) Je sais parfaitement séparer vue et modèle :

Architecture globale

models	Revert "Mise en forme du dossier + UseCase"	24 minutes ago
utils	Revert "Mise en forme du dossier + UseCase"	24 minutes ago
views	Revert "Mise en forme du dossier + UseCase"	24 minutes ago

13) Je maîtrise le cycle de vie de mon application :

Architecture globale

```
@Override
protected void onStart() {
    //this.gameService.start();
    super.onStart();
}

@Override
protected void onPause() {
    super.onPause();
    if (this.paused) {
        this.gameService.pause();
    }
}

@Override
protected void onResume() {
    if (!this.paused) {
        this.gameService.resume();
    }
    super.onResume();
}

@Override
protected void onStop() {
    this.gameService.stop();
    // SAVE GAMEMANAGER STATE
    super.onStop();
}

@Override
protected void onRestart() {
    // RESTORE GAMEMANAGER (instances + points etc)
    super.onRestart();
}
```

14) Je sais utiliser le findViewById à bon escient :

Package activities/GameActivity

```
GameView gameView = findViewById(R.id.gameView);  
this.gameService.startGameloop(gameView);
```

15) Je sais gérer les permissions dynamiques de mon application :

16) Je sais gérer la persistance légère de mon application :

Package activities/OptionsActivity

Je sais gérer la persistance profonde de mon application :

17) Je sais afficher une collection de données :

Package activities/ScoreActivities

```
lscores = sortList(lscores);  
ArrayAdapter<Score> arrayAdapter = new ArrayAdapter<Score>(this,  
android.R.layout.simple_list_item_1, lscores);  
scores.setAdapter(arrayAdapter);
```

18) Je sais coder mon propre adaptateur :

Package activities/ScoreActivities

```
ArrayAdapter<Score> arrayAdapter = new ArrayAdapter<Score>(this,  
android.R.layout.simple_list_item_1, lscores);
```

19) Je maîtrise l'usage des fragments :

Pas d'utilisation de fragment (pas vraiment l'utilité dans cette appli).

20) Je maîtrise l'utilisation de Git :

Voir depot Git

21) Je sais développer une application publiable sur le store. :

APK dans le projet + sons, images libre de droit

22) Je sais utiliser l'accéléromètre :

Package core/managers/SensorService

```
sensor = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
```

