Jérémy Liot

Gabriel Theuws

Android Projet Barman

Dossier Preuve de nos compétences

1. Je sais utiliser les Intent pour faire communiquer deux activités.

```java
/**
 * Demande l'activité victoire et la lance
 */
@Override
public void victoire(){
    Intent intent = new Intent(this, VictoireActivity.class);
    intent.putExtra("SCORE",votreScore.getText());
    startActivity(intent);
}
```

2. Je sais développer en utilisant le SDK le plus bas possible.

```
android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"

    defaultConfig {
        applicationId "com.example.projet_barman"
        minSdkVersion 16
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
```

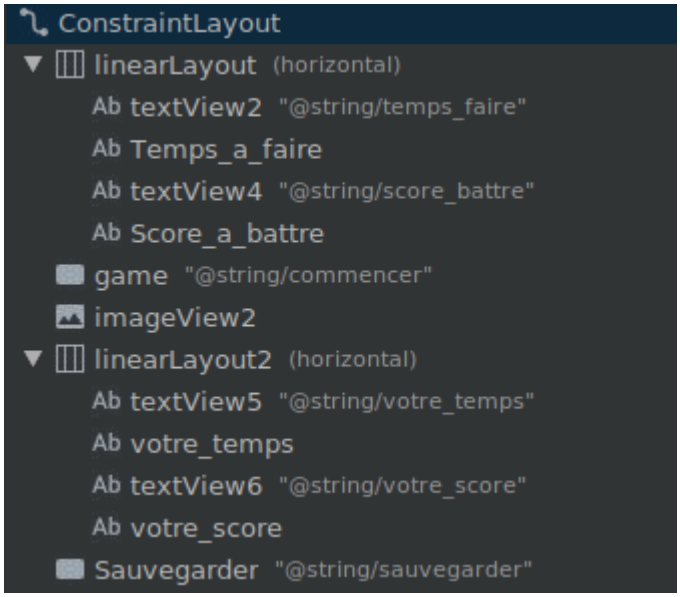
3. Je sais distinguer mes ressources en utilisant les qualifier

```
<!-- utilisé pour afficher le temps de secouage -->
<TextView
    android:id="@+id/votre_temps"
```

```
<!-- utilisé pour afficher le score du joueur -->
<TextView
    android:id="@+id/votre_score"
```

```
votreTemps = findViewById(R.id.votre_temps);
votreScore = findViewById(R.id.votre_score);
scoreABattre = findViewById(R.id.Score_a_battre);
tempsAFaire = findViewById(R.id.Temps_a_faire);
```

4. Je sais faire des vues xml en utilisant layouts et composants adéquats



5. Je sais coder proprement mes activités, en m'assurant qu'elles ne font que relayer les évènements

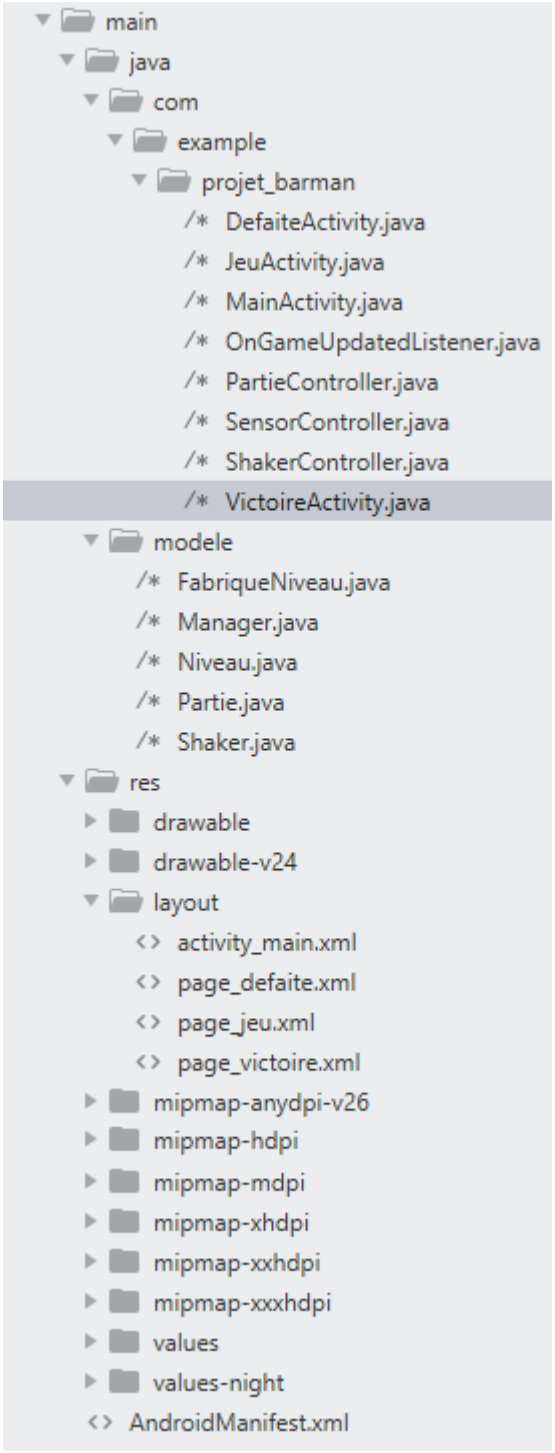
```java
/**
 * Lance le jeu
 * @param view inutilisée
 */
public void jouer(View view) {
    manager.lancerLeJeu();
}

/**
 * IMPORTANT si le joueur met en pause le jeu il ne faut pas prendre en compte le secouage
 * économie d'énergie et de passage dans l'event
 */
@Override
protected void onPause() {
    manager.seMettreEnpause();
    super.onPause();
}

/**
 * Utilise un nouveau thread qui permet de set le score dans la textview prévu à cet effet
 * @param score le score à écrire dans la textview
 */
@Override
public void updateScore(String score) {
    runOnUiThread(() -> votreScore.setText(score));
}

/**
 * Utilise un ouveau thread qui permet de set le temps dans la textview prévu à cet effet
 * @param temps le temps à écrire dans la textview
 */
@Override
public void updateTemps(String temps) {
    runOnUiThread(() -> votreTemps.setText(temps));
}
```

6. Je sais coder une application en ayant un véritable métier
7. Je sais parfaitement séparer vue et modèle



8. Je maîtrise le cycle de vie de mon application

```java
/**
 * IMPORTANT si le joueur met en pause le jeu il ne faut pas prendre en compte le secouage
 * économie d'énergie et de passage dans l'event
 */
@Override
protected void onPause() {
    manager.seMettreEnpause();
    super.onPause();
}
```

9. Je sais utiliser le findViewById à bon escient

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.page_jeu);
    manager.setSensorManager((SensorManager)getSystemService(Context.SENSOR_SERVICE));
    manager.setJeuActivity(this);
    votreTemps = findViewById(R.id.votre_temps);
    votreScore = findViewById(R.id.votre_score);
    scoreABattre = findViewById(R.id.Score_a_battre);
    tempsAFaire = findViewById(R.id.Temps_a_faire);
    scoreABattre.setText(String.valueOf(niveau.getNbMinPts()));
    System.out.println("Temps: " + shaker.getTpsshakeEnSecondes() + "s et points: "+ shaker.getMaxpts());
    tempsAFaire.setText(String.valueOf(shaker.getTpsshakeEnSecondes()));
}
```

10. Je sais gérer les permissions dynamiques de mon application

11. Je sais gérer la persistance légère de mon application

```java
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.page_defaite);
    if (savedInstanceState == null)
    {
        Bundle bundle = getIntent().getExtras();
        CharSequence sequence = bundle.getCharSequence("SCORE");
        TextView score = findViewById(R.id.textView3);
        score.setText(String.format("Vous avez perdu avec %s points", sequence));
        Log.d("DEBUG_BUNDLE","Voici la récupération du score via le Bundle : "+sequence);
    }
}
```

12. Je sais gérer la persistance profonde de mon application

```java
/**
 * sauvegarde la partie
 * @param view inutilisée
 * @throws IOException lance une exception si il y a une erreur d'entrée-sortie
 */
public void Sauvegarder(View view) throws IOException {
    File f = getFilesDir();
    File fichiersauvegarde= new File(f,"Partie.ser");
    if(fichiersauvegarde.exists()){
        fichiersauvegarde.delete();
    }

    FileOutputStream sauvegarde = new FileOutputStream(fichiersauvegarde);
    ObjectOutputStream oos = new ObjectOutputStream(sauvegarde);
    partie.serialiser(oos);
    oos.close();
    Intent intent = new Intent(this, MainActivity.class);
    startActivity(intent);
}
```

```java
/**
 * Charge une partie dans l'application
 * @param view inutilisée
 * @throws ClassNotFoundException lance une exception si la classe sérialisée n'est pas trouvée
 */
public void Charger(View view) throws ClassNotFoundException {
    try {
        File f= getFilesDir();
        File fichiersauvegarde = new File(f,"Partie.ser");
        FileInputStream sauvegarde = new FileInputStream(fichiersauvegarde);
        ObjectInputStream ois = new ObjectInputStream(sauvegarde);
        Partie p = manager.getPartieActuelle().deserialiser(ois);
        ois.close();
        manager.getPartieController().setPartieActuelle(p);
        Intent intent = new Intent(this, JeuActivity.class);
        startActivity(intent);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

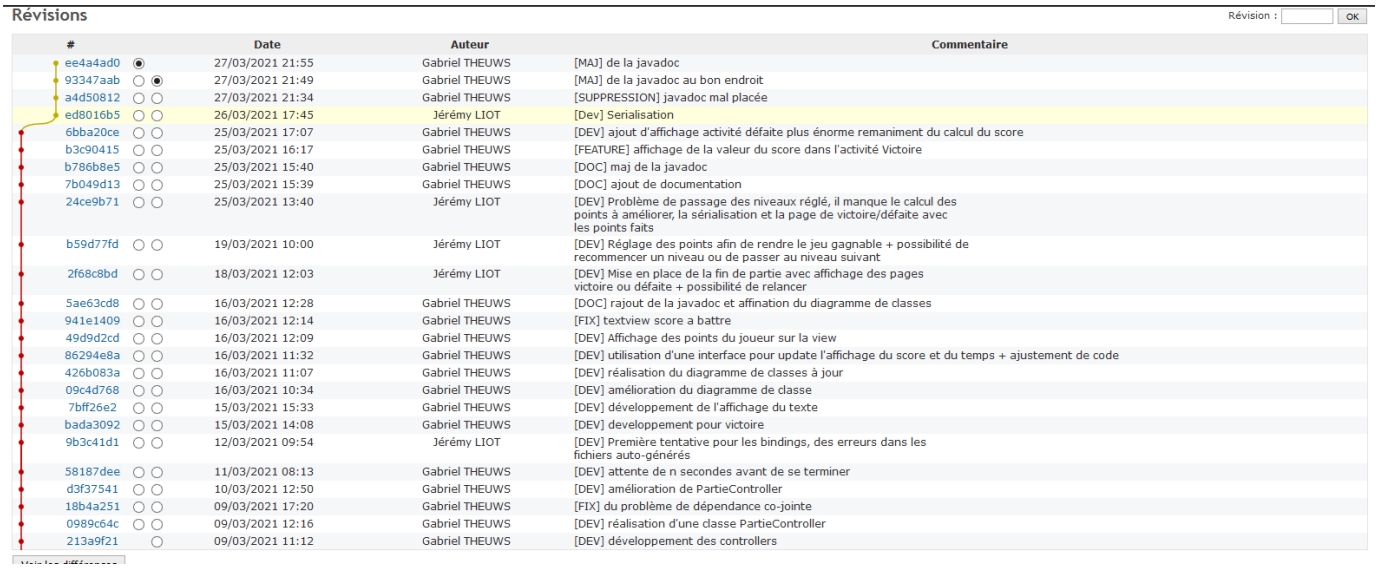
13. Je sais afficher une collection de données

14. Je sais coder mon propre adaptateur

15. Je maîtrise l'usage des fragments

16. Je maîtrise l'utilisation de Git

```
$ git log --oneline
ee4a4ad (HEAD -> master, origin/master, origin/HEAD) [MAJ] de la javadoc
93347aa [MAJ] de la javadoc au bon endroit
a4d5081 [SUPPRESSION] javadoc mal placée
ed8016b [Dev] Serialisation
6bba20c (origin/gabriel) [DEV] ajout d'affichage activité défaite plus énorm
b3c9041 [FEATURE] affichage de la valeur du score dans l'activité victoire
b786b8e [DOC] maj de la javadoc
7b049d1 [DOC] ajout de documentation
24ce9b7 (origin/jeremy) [DEV] Problème de passage des niveaux réglé, il manq
```



Révisions

Révision : OK

#	Date	Auteur	Commentaire
ee4a4ad0	27/03/2021 21:55	Gabriel THEUWS	[MAJ] de la javadoc
93347aab	27/03/2021 21:49	Gabriel THEUWS	[MAJ] de la javadoc au bon endroit
a4d50812	27/03/2021 21:34	Gabriel THEUWS	[SUPPRESSION] javadoc mal placée
ed8016b5	26/03/2021 17:45	Jérémy LIOT	[Dev] Serialisation
6bba20ce	25/03/2021 17:07	Gabriel THEUWS	[DEV] ajout d'affichage activité défaite plus énorme remaniment du calcul du score
b3c90415	25/03/2021 16:17	Gabriel THEUWS	[FEATURE] affichage de la valeur du score dans l'activité Victoire
b786b8e5	25/03/2021 15:40	Gabriel THEUWS	[DOC] maj de la javadoc
7b049d13	25/03/2021 15:39	Gabriel THEUWS	[DOC] ajout de documentation
24ce9b71	25/03/2021 13:40	Jérémy LIOT	[DEV] Problème de passage des niveaux réglé, il manque le calcul des points à améliorer, la sérialisation et la page de victoire/défaite avec les points faits
b59d77fd	19/03/2021 10:00	Jérémy LIOT	[DEV] Réglage des points afin de rendre le jeu gagnable + possibilité de recommencer un niveau ou de passer au niveau suivant
2f68c8bd	18/03/2021 12:03	Jérémy LIOT	[DEV] Mise en place de la fin de partie avec affichage des pages victoire ou défaite + possibilité de relancer
5ae63cd8	16/03/2021 12:28	Gabriel THEUWS	[DOC] rajout de la javadoc et affination du diagramme de classes
941e1409	16/03/2021 12:14	Gabriel THEUWS	[FIX] textview score a battre
49d9d2cd	16/03/2021 12:09	Gabriel THEUWS	[DEV] Affichage des points du joueur sur la view
86294e8a	16/03/2021 11:32	Gabriel THEUWS	[DEV] utilisation d'une interface pour update l'affichage du score et du temps + ajustement de code
426b083a	16/03/2021 11:07	Gabriel THEUWS	[DEV] réalisation du diagramme de classes à jour
09c4d768	16/03/2021 10:34	Gabriel THEUWS	[DEV] amélioration du diagramme de classe
7bff26e2	15/03/2021 15:33	Gabriel THEUWS	[DEV] développement de l'affichage du texte
bada3092	15/03/2021 14:08	Gabriel THEUWS	[DEV] developpement pour victoire
9b3c41d1	12/03/2021 09:54	Jérémy LIOT	[DEV] Première tentative pour les bindings, des erreurs dans les fichiers auto-générés
58187dee	11/03/2021 08:13	Gabriel THEUWS	[DEV] attente de n secondes avant de se terminer
d3f37541	10/03/2021 12:50	Gabriel THEUWS	[DEV] amélioration de PartieController
18b4a251	09/03/2021 17:20	Gabriel THEUWS	[FIX] du problème de dépendance co-jointe
0989c64c	09/03/2021 12:16	Gabriel THEUWS	[DEV] réalisation d'une classe PartieController
213a9f21	09/03/2021 11:12	Gabriel THEUWS	[DEV] développement des controllers

Application

- **Je sais développer une application sans utiliser de librairies externes**
  Nous n'avons pas utilisé de librairie externe
- **Je sais développer une application publiable sur le store**
  Notre application est jouable et fonctionne bien
- **Je sais utiliser l'accéléromètre**

```java
public class SensorController {
    private final SensorManager mSensorManager;
    private final Sensor mAccelerometer;
    private boolean registered=false;
    private long debutRegister;
    private long tempPasseARegister = 0;

    /**
     * Créer une instance de SensorController qui gère les listeners du sensor
     * @param manager le manager de l'accélèromètre
     */
    public SensorController(SensorManager manager)
    {
        mSensorManager = manager;
        int accelerometer = Sensor.TYPE_LINEAR_ACCELERATION; // meilleur que Sensor.TYPE_ACCELEROMETER car ne prend pas en compte la gravité
        if ((mAccelerometer = mSensorManager.getDefaultSensor(accelerometer)) != null){
            Log.d("DEBUG","Il y a bien un accéléromètre sur votre téléphone");
        } else {
            Log.d("ERROR","Il n'y a pas d'accéléromètre sur votre téléphone !");
        }
    }

    /**
     * Ajoute le sensorEventListener en tant que listener de l'accélèromètre via le manager de sensor
     * @param sensorEventListener ajoute le sensorEventListener au listener du sensor
     */
    public boolean register(SensorEventListener sensorEventListener) {
        debutRegister = SystemClock.elapsedRealtime();
        return registered = mSensorManager.registerListener(sensorEventListener, mAccelerometer, SensorManager.SENSOR_DELAY_NORMAL);
    }

    /**
     * Retire le listener sensorEventListener du manager de sensor
     * @param sensorEventListener le sensorEventListener qui sera rajouté
     */
    public void unregister(SensorEventListener sensorEventListener) {
            mSensorManager.unregisterListener(sensorEventListener);
            registered = false;
            tempPasseARegister = debutRegister - SystemClock.elapsedRealtime();
    }
```