

DOCUMENTATION

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

Voir rapport.pdf

Je sais concevoir et décrire un diagramme de cas d'utilisation pour mettre en avant les différentes fonctionnalités de mon application.

Voir rapport.pdf

Je sais concevoir un diagramme UML de qualité représentant mon application.

Voir Diagram Class.pdf

Je sais décrire mon diagramme UML en mettant en valeur et en justifiant les éléments essentiels.

Voir Diagram Class.pdf

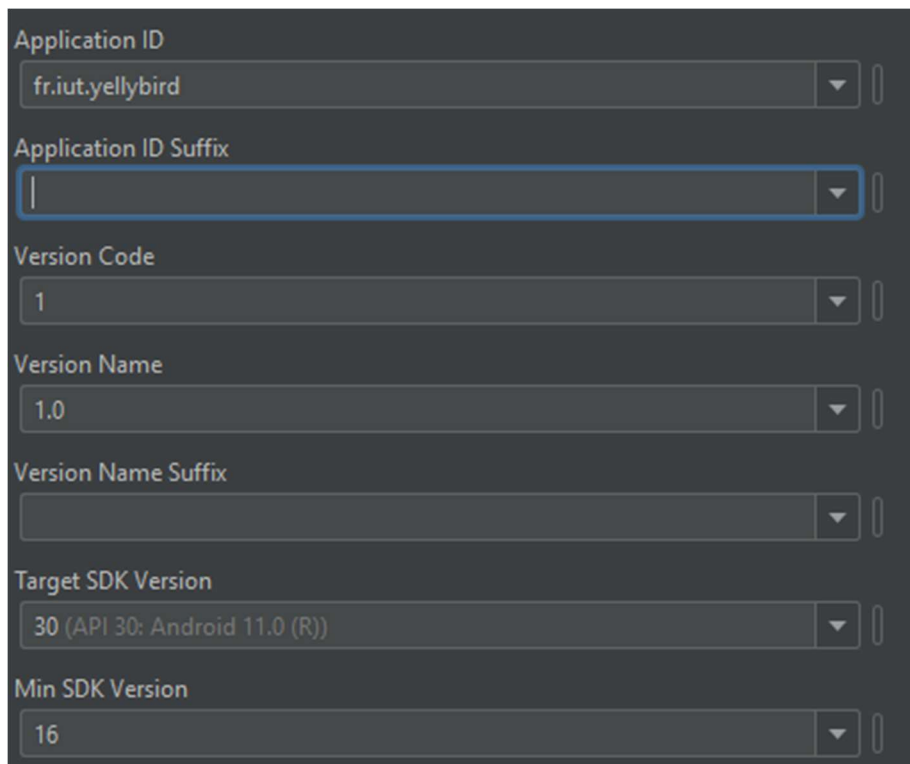
CODE

Je sais utiliser les Intent pour faire communiquer deux activités.

```
skins.setOnClickListener(view -> {
    Intent activity = new Intent(getApplicationContext(),SkinActivity.class);
    startActivity(activity);
    finish();
});

restart.setOnClickListener(view -> {
    Intent activity = new Intent(getApplicationContext(),GameActivity.class);
    startActivity(activity);
    finish();
});
```

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



Application ID

fr.iut.yellybird

Application ID Suffix

|

Version Code

1

Version Name

1.0

Version Name Suffix

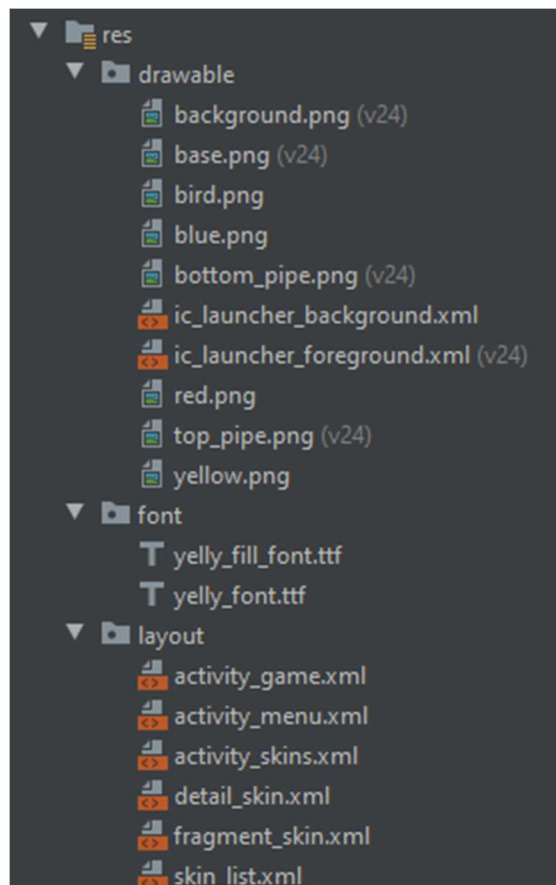
Target SDK Version

30 (API 30: Android 11.0 (R))

Min SDK Version

16

Je sais distinguer mes ressources en utilisant les qualifier Je sais faire des vues xml en utilisant layouts et composants adéquats



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

```
public class MainActivity extends AppCompatActivity {

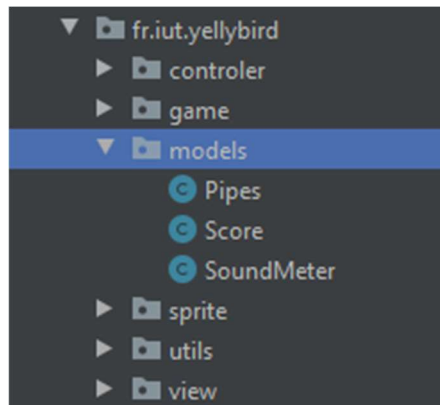
    private GameView gameView;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        gameView = new GameView(context: this);
        setContentView(gameView);
    }

    @Override
    protected void onResume() {
        super.onResume();
        // gameView.resume();
    }

    @Override
    protected void onPause() {
        super.onPause();
        // sprite.pause();
    }
}
```

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



Je maîtrise le cycle de vie de mon application Je sais utiliser le findViewById à bon escient

```
public class SkinList extends Fragment {
    private SkinActivity myActivity;

    public SkinList() { super(R.layout.skin_list); }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        myActivity = ((SkinActivity)getContext());

        RecyclerView listView = view.findViewById(R.id.skin_list);
        listView.setLayoutManager(new LinearLayoutManager(myActivity));
        listView.setAdapter(new SkinListAdapter(myActivity.getSkins()));
    }
}
```

Je sais gérer les permissions dynamiques de mon application

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="fr.iut.yellybird">
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
```

```

private boolean checkPermission() {
    int result = ActivityCompat.checkSelfPermission( context: this, Manifest.permission.RECORD_AUDIO);
    return result == PackageManager.PERMISSION_GRANTED;
}

private void requestPermission() {
    ActivityCompat.requestPermissions( activity: this, permissions, PERMISSION_REQUEST_CODE);
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    switch (requestCode) {
        case PERMISSION_REQUEST_CODE:
            if (grantResults.length > 0) {

                boolean microAccepted = grantResults[0] == PackageManager.PERMISSION_GRANTED;

            }
            break;
    }
}

```

Je sais gérer la persistance légère de mon application Je sais gérer la persistance profonde de mon application

Légère uniquement

```

public class Serializer {
    /**
     * Serialization of an object
     *
     * @param filename the filename
     * @param object the object
     * @param context the context
     */
    public static void serialize(String filename, Object object, Context context)
    {
        try{
            FileOutputStream stream = context.openFileOutput(filename, Context.MODE_PRIVATE);
            ObjectOutputStream objectStream = new ObjectOutputStream(stream);
            objectStream.writeObject(object);
            objectStream.flush();
            objectStream.close();
        } catch (FileNotFoundException e) {
            //File not found
            e.printStackTrace();
        } catch (IOException e) {
            //Error serialisation
            e.printStackTrace();
        }
    }

    /**
     * Deserialization of an object
     *
     * @param filename the filename
     * @param context the context
     * @return object
     */
    public static Object deserialize(String filename, Context context)
    {
        try {
            FileInputStream stream = context.openFileInput(filename);
            ObjectInputStream objectStream = new ObjectInputStream(stream);
            Object object = objectStream.readObject();
            objectStream.close();
            return object;
        } catch (FileNotFoundException e) {
            //File not found
            e.printStackTrace();
        } catch (IOException e) {
            //Error serialisation
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            //Error read object
            e.printStackTrace();
        }
        return null;
    }
}

```

Je sais afficher une collection de données Je sais coder mon propre adaptateur

Voir view.adapter.*

Je maîtrise l'usage des fragments

Voir view.fragments.*

Je maîtrise l'utilisation de Git

Oui. Utilisation de 2 branches pour repartir le travail entre binôme.

APPLICATION

Je sais développer une application sans utiliser de librairies externe

Oui aucune.

Je sais développer une application publiable sur le store.

Non.

Je sais utiliser toute autre contrainte validée par l'enseignant

Utilisation de l'amplitude micro

Voir class SoundMeter