

Mathis RIBEMONT

Martin MAULUS

Documentation : Preuve

DOCUMENTATION

Je sais décrire le contexte de mon application, pour que n'importe qui soit capable de comprendre à quoi elle sert : voir [contexte.pdf](#)

Je sais concevoir et décrire un diagramme de cas d'utilisation pour mettre en avant les différentes fonctionnalités de mon application : voir [Explication-diagramme-cas-utilisation.pdf](#)

Je sais concevoir un diagramme UML de qualité représentant mon application : voir [diagrammeClasse.pdf](#)

Je sais décrire mon diagramme UML en mettant en valeur et en justifiant les éléments essentiels : voir explication dans le fichier [Explication-diagramme-classe.pdf](#)

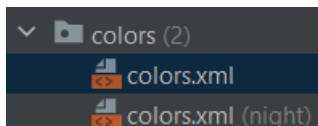
CODE

Je sais utiliser les Intent pour faire communiquer deux activités :

```
public void clickButtonSetting(View view) {  
    Intent monIntent = new Intent( packageContext: this, PageSetting.class);  
    startActivity(monIntent);  
}
```

Je sais développer en utilisant le SDK le plus bas possible : Le projet est conçu pour Android 4

Je sais distinguer mes ressources en utilisant les qualifier :



Je sais faire des vues xml en utilisant layouts et composants adéquats : voir les vues [mainwindow.xml](#) / [pagesetting.xml](#) / [fenetredejeu.xml](#)

Je sais coder proprement mes activités, en m'assurant qu'elles ne font que relayer les évènements : voir l'activité [PageSetting.java](#) et la CustomView [CellsGrid](#) (ce sont les plus pertinents)

Je sais coder une application en ayant un véritable métier : voir le package [modele](#)

Je sais parfaitement séparer vue et modèle : voir les packages [modele](#) et [vue](#)

Je maîtrise le cycle de vie de mon application : oui voir les activités (utilisation des méthodes [OnCreate](#), [OnResume](#), ...)

Je sais utiliser le findViewById à bon escient :

```
public int getValueSeekBar(View view){  
    SeekBar seek = (SeekBar) findViewById(R.id.seekBar);  
    int seekValue = seek.getProgress();  
    return seekValue;  
}
```

Je sais gérer les permissions dynamiques de mon application : Nous n'en avons pas eu besoin.

Je sais gérer la persistance légère de mon application : voir activité PageSetting.java (l'activité va chercher les données dans le modèle via le singleton du dieu).

Je sais gérer la persistance profonde de mon application : voir les classes Saver.java et Loader.java dans le modèle

Je sais afficher une collection de données : voir LoaderDialogFragment.java

Je sais coder mon propre adaptateur : Nous n'en avons pas.

Je maîtrise l'usage des fragments :

```
private void instantiateFragment(@Nullable Bundle savedInstanceState) {  
    if(savedInstanceState == null){  
        Log.d( tag: "PageSetting", msg: "Debug > Création fragment");  
  
        //argument pour fragmentBornRules  
        Bundle bundleBorn = new Bundle();  
        bundleBorn.putString("text_to_show", "Nombre de voisins pour naître".toString());  
        bundleBorn.putBooleanArray("arrayToBind", Dieu.getDieu().getRules().getBornRules());  
  
        //argument pour fragmentSurvivesRules  
        Bundle bundleSurvive = new Bundle();  
        bundleSurvive.putString("text_to_show", "Nombre de voisins pour survivre".toString());  
        bundleSurvive.putBooleanArray("arrayToBind", Dieu.getDieu().getRules().getSurviveRules());  
  
        getSupportFragmentManager().beginTransaction()  
            .setReorderingAllowed(true)  
            .replace(R.id.fragmentBornRules, FragmentRules.class, bundleBorn)  
            .replace(R.id.fragmentSurvivesRules, FragmentRules.class, bundleSurvive)  
            .commit();  
    }  
}
```

Je maîtrise l'utilisation de Git : voir notre dépôt

APPLICATION

Je sais développer une application sans utiliser de bibliothèques externes : voir code

Je sais développer une application publiable sur le store : Nous ne l'avons pas fait

Je sais développer un jeu intégrant une boucle de jeu threadée observable : voir BoucleDeJeu.java dans le modèle

Je sais développer un jeu graphique sans utiliser de SurfaceView : voir fenetredejeu.xml