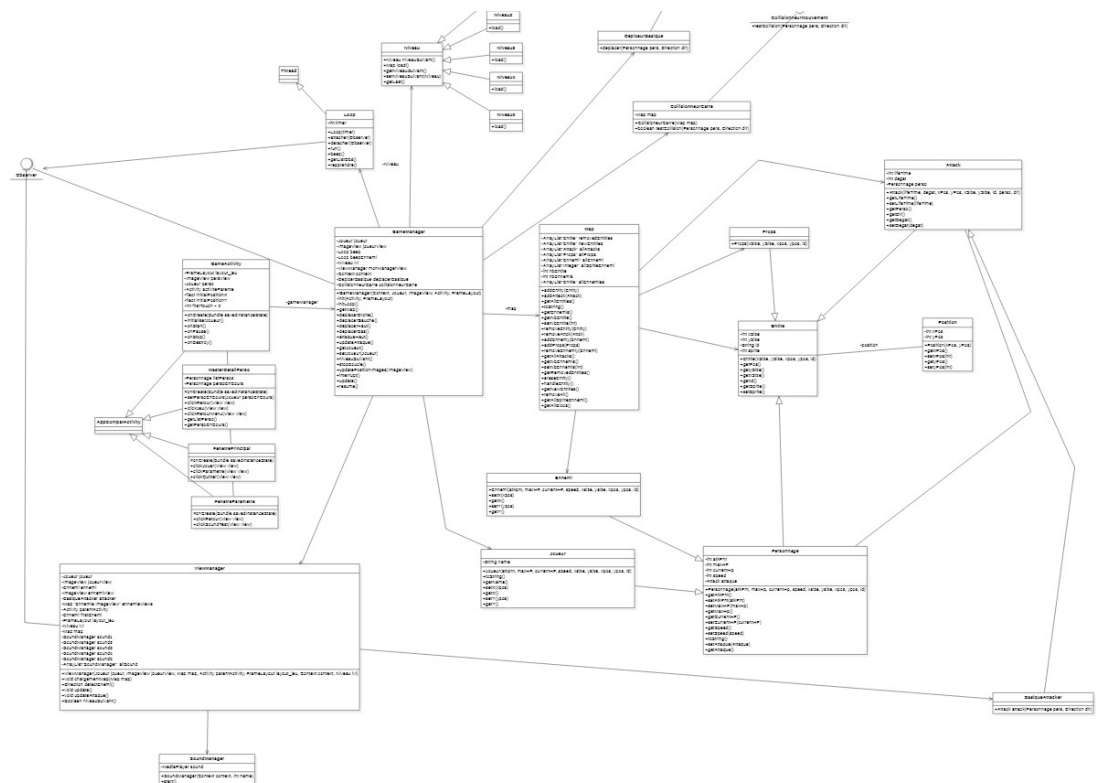


PREUVES

DOCUMENTATION:

- Je sais décrire le contexte de mon application, pour que n'importe qui soit capable de comprendre à quoi elle sert
- Je sais concevoir et décrire un diagramme de cas d'utilisation pour mettre en avant les différentes fonctionnalités de mon application
- Je sais concevoir un diagramme UML de qualité représentant mon application

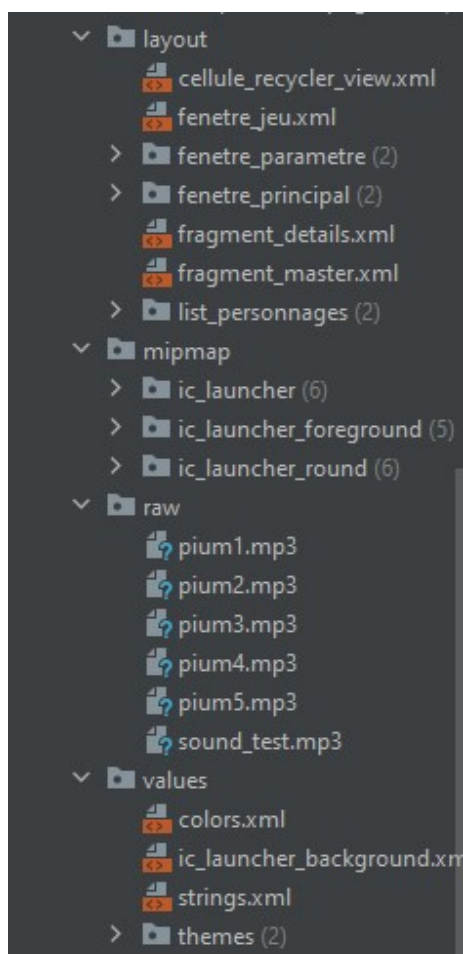


- Je sais décrire mon diagramme UML en mettant en valeur et en justifiant les éléments essentiels

Nous n'avons pas pu faire toutes les liaisons sur notre diagramme. Il faut retenir que nous avons au divisé la vue et le modèle au maximum et essayé de limiter la responsabilité des nos classe pour respecter les principes SOLID.

CODE:

- Je sais utiliser les intent pour faire communiquer deux activités
- Je sais développer en utilisant le SDK le plus bas possible
- Je sais distinguer mes ressources en utilisant les qualifier



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

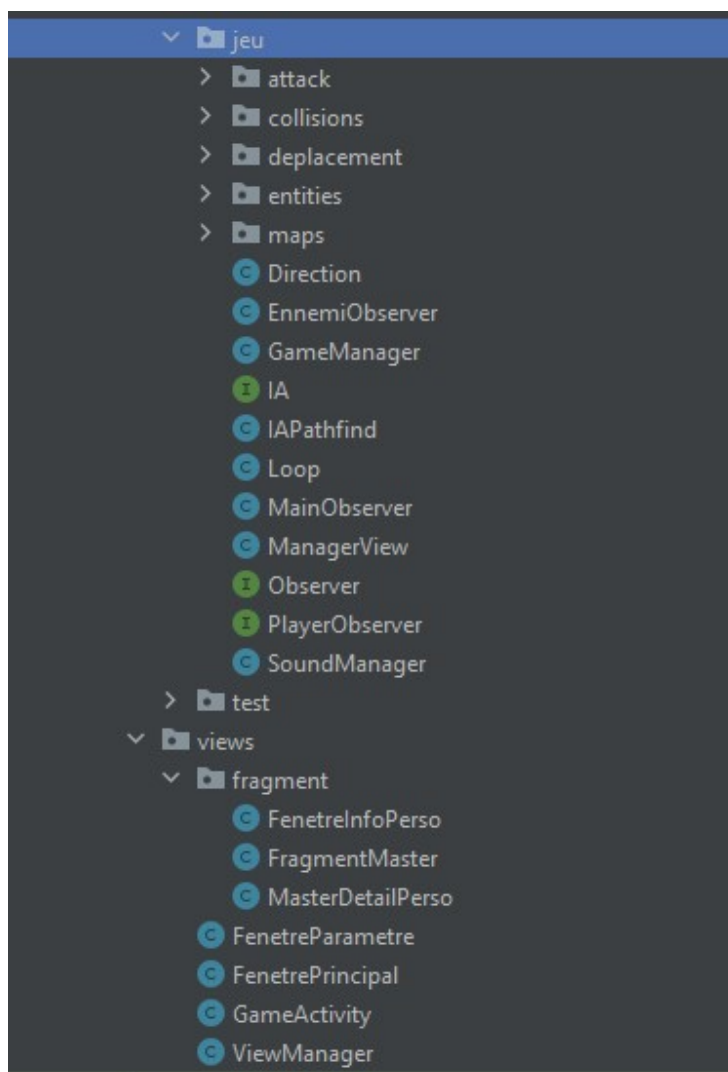
Nous avons utilisé les constraintLayout pour pouvoir positionné nos éléments de manière générique pour tout type de téléphone.

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

- Je sais coder une application en ayant un véritable métier

On peut le voir car nous avons séparé les classes métiers des vues et mis une couche d'abstraction pour de future modification.

- Je sais parfaitement séparer vue et modelé



- Je maitrise le cycle de vue de mon application

Nous utilisons les

- Je sais utiliser le findViewById à bon escient

GameActivity.java, ligne 38

```
37         this.activiteParente = getParent();
38         layout_jeu = (FrameLayout) findViewById(R.id.jeu);
39         initialiserJoueur();
```

GameActivity.java, ligne 46

```
45         persoView = new ImageView( context: this);
46         persoView = (ImageView) findViewById(R.id.imageView);
47         persoView.setImageResource(perso.getSprite());
```

- Je sais gérer les permissions dynamiques de mon application

AndroidManifest.xml, ligne 6-7

```
5
6         <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
7         <uses-permission android:name="android.permission.WAKE_LOCK" />
8     </application>
```

- Je sais gérer la persistance légère de mon application

Nous utilisons les id pour une persistance légère.

- Je sais gérer la persistance profonde de mon application

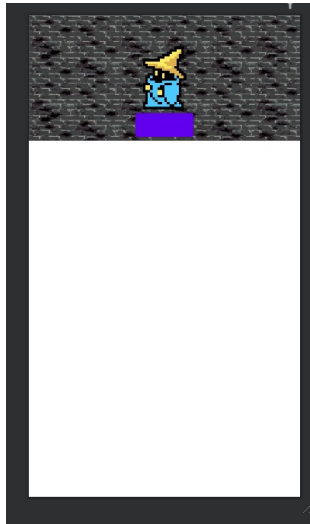
Nous n'avons pas trouvé d'utilisation correcte de persistance profonde pour notre application, nous n'en avons donc pas mis.

- Je sais afficher une collection de données

MonAdaptateur.java, ligne

42-47

```
@SuppressWarnings("UseCompatLoadingForDrawables")
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
@Override
public void onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, int position) {
    Joueur perso = list.get(position);
    ((ViewHolderPers)holder).setPersoEnCours(perso);
    ((ViewHolderPers)holder).getButton().setText(perso.getName());
    ((ViewHolderPers)holder).getImageView().setImageDrawable(activiteParent.getDrawable(R.drawable.template_character));
}
```



- Je sais coder mon propre adaptateur

```
public class MonAdaptateur extends RecyclerView.Adapter{

    private List<Joueur> list;
    private AppCompatActivity activiteParent;

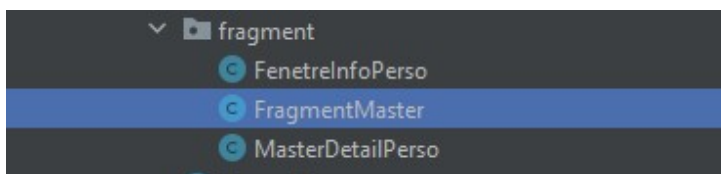
    public MonAdaptateur(List<Joueur> list, AppCompatActivity activiteParent) {
        this.list = list;
        this.activiteParent = activiteParent;
    }

    @NonNull
    @Override
    public ViewHolderPers onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        ConstraintLayout layout = (ConstraintLayout) LayoutInflater.from(parent.getContext()).inflate(R.layout.cellule_recycler_view, parent, false);
        return new ViewHolderPers(layout);
    }

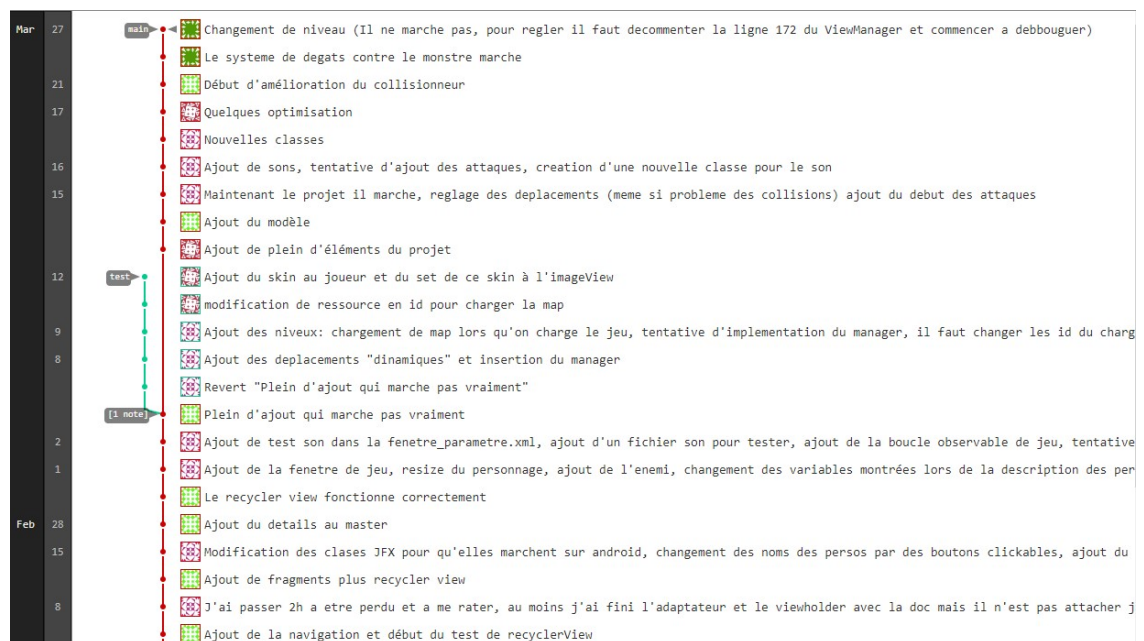
    @SuppressWarnings("UseCompatLoadingForDrawables")
    @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
    @Override
    public void onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, int position) {
        Joueur perso = list.get(position);
        ((ViewHolderPers)holder).setPersoEnCours(perso);
        ((ViewHolderPers)holder).getButton().setText(perso.getName());
        ((ViewHolderPers)holder).getImageView().setImageDrawable(activiteParent.getDrawable(R.drawable.template_character));
    }

    @Override
    public int getItemCount() { return list.size(); }
}
```

- Je maitrise l'usage de fragments



- Je maîtrise l'utilisation de Git



APPLICATION :

- Je sais développer une application sans utiliser des librairies externes

Nous n'en avons pas utilisé.

- Je sais développer une application publiable sur le store

- Je sais développer un jeu intégrant une boucle de jeu threadée observable

Nous avons mis en place une boucle de jeu avec un thread et des observateurs.

- Je sais développer un jeu graphique sans utiliser de SurfaceView