**PREUVE MAITRISE DES COMPETENCES**

**Documentation : (dossier documentation)**

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



1. Je sais faire un diagramme de cas d’utilisation pour mettre en avant les différentes fonctionnalités de mon application.



1. Je sais concevoir un diagramme UML intégrant des notions de qualité et correspondant exactement à l’application que j’ai à développer.
2. Je sais décrire un diagramme UML en mettant en valeur et en justifier les éléments essentiels

**Code :**

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

*Dans MainActivity*

Une image contenant texte

Description générée automatiquement



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

Une image contenant texte

Description générée automatiquement

Le plus petit SDK est utilisé pour toucher 100% des appareils



1. Je sais distinguer mes ressources en utilisant les qualifier

Une image contenant texte

Description générée automatiquement

Nous utilisions les qualifiers concernant l’orientation du téléphone pour nos layouts.

Une image contenant texte

Description générée automatiquement

Nous utilisons également le qualifier ‘nodpi’ car il s'agit de ressources indépendantes de la densité. Le système ne met pas à l'échelle les ressources marquées avec ce qualificatif, quelle que soit la densité de l'écran actuel.



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

Nous avons principalement utilisé le ConstraintLayout, LinearLayout et TabLayout. Nous avons également utilisé les composants adéquats. Pour plus de détails voire le code de l’application



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

Nos activités ne font que relayer les évènements. Elles n’exercent pas de responsabilités supplémentaires.



1. Je sais coder une application en ayant un véritable métier

Notre métier est bien conceptualisé en ayant chaque classe avec une seule responsabilité



1. Je sais parfaitement séparer vue et modèle

Notre modèle et notre vue sont séparé. Notre modèle n’a pas besoin de connaitre la vue.



1. Je maîtrise le cycle de vie de mon application

Nous maitrisons le cycle de vie de notre application Nous n’avons eu besoin d’utiliser que onCreate et onSaveInstanceState (onSaveInstanceState se trouve dans MainActivity).



1. Je sais utiliser le findViewById à bon escient

Nous n’utilisons le findViewById que quand nous avons besoin de récupérer des éléments de notre vue. De plus, si nous avons besoin de réutiliser plus tard un élément récupéré via findViewById, nous stockons la référence dans une variable.



10- Je sais gérer la persistance légère de mon application

La persistance légère est gérée automatiquement grâce aux id données dans les fichiers xml. De plus, nous utilisons onSaveInstanceState.



1. Je sais gérer la persistance profonde de mon application

Pour faire de la persistance profonde, nous utilisons les préférences. La responsabilité de sauvegarde ainsi que de chargement des données est délégué au RankingManager. Pour ce faire nous sauvegardons les scores de notre tableau des scores dans des JSONObject pour ensuite placer ces JSONObject dans un JSONArray.



1. Je sais afficher une collection de données

Nous utilisons une RecyclerView pour afficher les scores et nous passons à l’Adaptater une List<ScoreRanking>. Nous affichons donc une collection de ScoreRanking.



1. Je sais coder mon propre adaptateur

*package modelandroid/view*





1. Je maîtrise l’usage des fragments

Premièrement, nous avons utilisé les fragments avec notre RecyclerView via un ItemClickListener. Vous pouvez afficher le détail d’un score du tableau des score en cliquant sur celui-ci. Le fragement se nomme RankingDetailFragment

Deuxièmement, notre menu de jeu possède un TabLayout. Pour le changement d’onglet nous avons utilisé un FragmentContainerView, nous permettant de changer le Fragment affiché en fonction de l’onglet sélectionné. MenuFragment correspond à l’affichage de la vue du menu et RankingFragment correspond à l’affichage de la vue du tableau des scores.



1. Je maîtrise l’utilisation de Git

Nous avons utilisé le gitlab de l’UCA pour versionner notre projet Android. Nous avons constamment poussé nos travaux sur gitlab tout en corrigeant d’éventuels conflits.

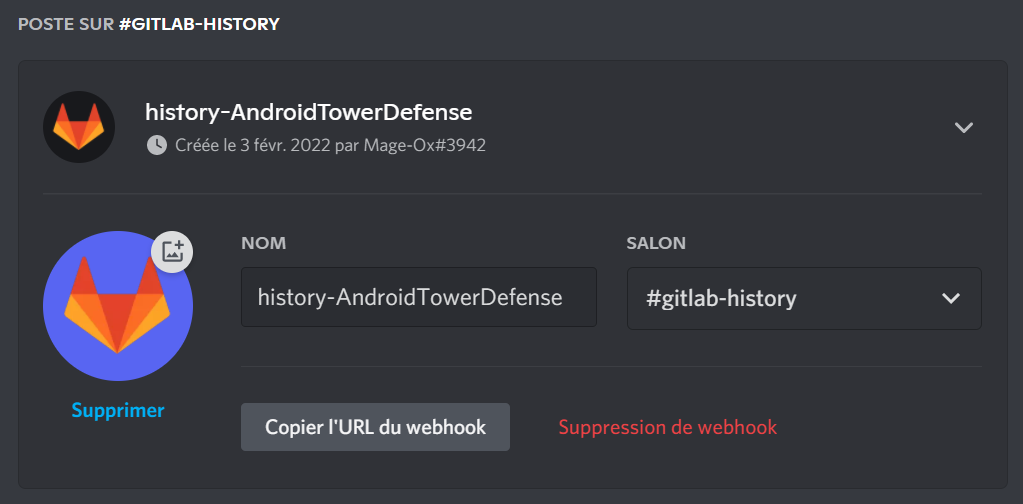
De plus, pour être informé de nos différents commits sur gitlab, nous avons utilisé une intégration ‘webhook’

Une image contenant texte

Description générée automatiquement

Nous l’avons intégré à un salon discord

Une image contenant texte

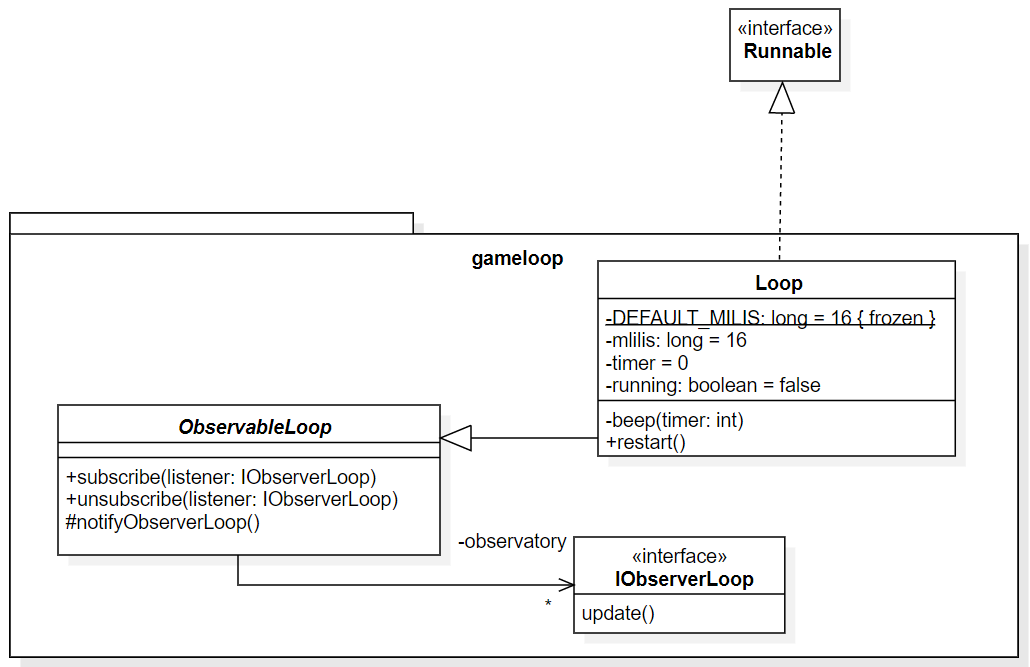
Description générée automatiquement

**Application :**



1. Je sais développer une application sans utiliser de librairies externes.
2. Je sais développer une application publiable sur le store.



1. Je sais développer un jeu intégrant une boucle de jeu threadée observable.

Pour démarrer la boucle de jeu :

*model/gamelogic/GameManager*

Une image contenant texte

Description générée automatiquement

Pour mettre en pause la boucle de jeu :

*model/gamelogic/gameloop*

Une image contenant texte

Description générée automatiquement

Pour relancer la boucle de jeu après la pause :

*model/gamelogic/gameloop*

Une image contenant texte

Description générée automatiquement



1. Je sais développer un jeu graphique sans utiliser de SurfaceView.

Nous n’avons pas utilisé de SurfaceView