Projet Bataille Navale sur Android

28/03/2021

Compétences devant être maitrisées

DOCUMENTATION

Je sais concevoir un diagramme UML intégrant des notions de qualité et correspondant exactement à l'application que j'ai à développer.

Nous avons créé un diagramme de classe complet décrivant parfaitement l'application.

Je sais décrire un diagramme UML en mettant en valeur et en justifier les éléments essentiels.

Nous avons ajouté un fichier de description du diagramme de classe.

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

Nous avons créé un fichier contexte_bataille_navale.pdf permettant de comprendre comment marche l'application. De plus, nous avons ajouté un ReadMe.md .

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

Nous avons créé un diagramme de cas d'utilisation décrivant toutes les fonctionnalités de l'application.

CODE

Je sais utiliser les Intent pour faire communiquer deux activités.

Nous utilisons des Intent pour naviguer entre nos différentes activités.

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

Nous avons utilisé l'API 16 : Android 4.1.1 qui permet une compatibilité avec 99 % des téléphones android.

Je sais distinguer mes ressources en utilisant les qualifier

Nos ressources ont été séparées dans différents dossier :

-layout : pour les vues

-values : pour tous les styles, couleurs et textes ainsi que la gestion des thèmes

-drawable : pour toutes les images et compositions d'images

-raw : contenant nos 3 musiques

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

Nous avons créé différentes vues en utilisant des constraint layout, gridLayout et autres éléments xml. Elles se situent dans le package main/res/layout. J'ai utilisé des vues verticales et horizontales séparées pour que l'affichage se recentre.

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

Nous avons créé des activités permettant de gérer les actions des utilisateurs, cela permet à l'utilisateur de par exemple rentrer son pseudo, créer une photo et jouer à la bataille navale. Voir le dossier

Je sais coder une application en ayant un véritable métier

Nous avons créé un véritable métier pour notre application dans le package main/java/model. Nous avons par exemple des classes Bateau, Cellule, Plateau, Partie.

Nous avons utilisé un patron de conception Singleton pour notre classe GameManager.java

Nous avons séparé le métier dans différents packages comme manager/, persistance/, model/.

Je sais parfaitement séparer vue et modèle

Nous avons séparé la gestion des vues et du modèles. Les vues sont gérées dans les activités. Les interactions permettent de modifier le model à travers une classe GameManager.

Je maîtrise le cycle de vie de mon application

Nous gérons le cycle de vie de mon application à travers le OnDestroy d'une activité qui nous permet de sauvegarder les données de mon application lors de l'appel de cette méthode. De plus, nous avons un MusicManager qui est appelé dans le onResume et le onStop de notre application. Cela nous permet d'arrêter la musique quand l'application n'est plus affichée et de la remettre si il revient dessus. La musique continue bien aussi entre les différentes activités.

Je sais utiliser le findViewById à bon escient

Nous avons utilisé des boutons afin de naviguer, des ImagesView pour mettre l'image d'un joueur et des Edittext pour la gestion des pseudos. Nous avons récupéré ces éléments grâce au findViewById()

Je sais gérer les permissions dynamiques de mon application

Nous avons géré la permission de la caméra. Quand l'utilisateur veut prendre une photo, l'application lui demande la permission. La prise de photo est refusée si l'application n'a pas la permission. On retrouve cette gestion dans view/FormJoueur.java

Je sais gérer la persistance légère de mon application

Nous avons géré la persistance légère de notre application dans l'activité FormJoueur.java, l'activité n'est pas bloquée en mode portrait. Quand on tourne l'écran, l'image du joueur ou l'image de base s'il n'n a pas pris apparaît bien grâce à cette sauvegarde. Nous avons utilisé la méthode onSaveInstanceState() en sauvegardant dans le Bundle.

Je sais gérer la persistance profonde de mon application

Nous avons sauvegardé un historique des parties ayant eu lieu sur cet appareil. Quand une partie est terminée on ajoute cette partie à une liste de partie que l'on persiste dans un fichier local. Nous avons utilisé des objets qui implémente Serializable. Cette gestion a été réalisée dans le dossier persistance/ avec la classe SauverSer.java.

Je sais afficher une collection de données

Nous affichons une liste de parties. Le joueur peut choisir de supprimer une partie de l'historique dans le fichier historique_parties.xml et l'activité HistoriqueParties.java. Cela la supprimera définitivement. Nous avons aussi affiché une liste de Cellule pour la bataille navale dans placement_bateau.xml et plateau_jeu.xml, ces affichages sont gérés dans PlacementBateau.java et Plateaujeu.java

Je sais coder mon propre adaptateur

Nous avons codé 3 adaptateurs dans le dossier java/adapter. Un pour le placement des bateaux (GridAdapterPlacement.java) qui permet un affichage différent de celui pour le plateau de jeu (GridAdapterJeu.java). Ces deux adaptateurs utilisent la classe BaseAdapter.java. Enfin le troisième utilise les principes vus en cours car il hérite de la classe RecyclerView.Adapter. Nous avons utilisé cet adapter pour afficher la liste des parties de l'historique.

Nous avons créé deux ViewHolder pour la cellule d'une partie dans l'historique et pour la cellule du plateau de jeu. Cette dernière est modifiée quand elle est touchée ou coulée.

Je maîtrise l'usage des fragments

Nous n'avons pas utilisé les fragments pour notre projet. Après conversation avec notre professeur nous avons convenus que cela n'était pas indispensable compte tenu de tous les critères déjà implémentés.

Nous aurions pu créer des fragments quand nous cliquons sur une partie de l'historique pour voir le détail des plateaux de chaque joueur et d'autres précisions mais nous n'avons pas eu le temps de nous pencher sur la question.

Je maîtrise l'utilisation de Git

Nous avons totalement maitrisé Git durant ce projet. Nous avons créé deux branches pour travailler chacun de notre côté. Quand l'un des deux avait fini de travailler il poussait son travail sur le master du projet pour que l'autre puisse se tenir à jour des avancées.

APPLICATION

Je sais développer une application sans utiliser de librairies externes

Nous n'avons utilisé aucune librairie externe. Pas de Framework n'a été utilisé sur ce projet.

Je sais développer une application publiable sur le store.

Notre application devrait être publiable sur le score. Au moment où nous écrivons ce document nous n'avons pas pu tester la mise en ligne. Nous avons fait la demande à M. Bouhours pour la mettre en ligne. L'apk a bien été généré et testé sur différents téléphones.

Je sais utiliser la caméra

Nous avons utilisé la caméra dans l'activité FormJoueur.java, celui-ci peut se prendre en photo, en reprendre une si cela ne lui pas. Cette photo est directement affichée après la prise. Enfin lors de la victoire, le joueur gagnant se voit afficher sa photo sur l'écran de fin à coté de son pseudo dans affichage_fin_partie.xml géré par l'activité AffichageFinPartie.java