

CHEVASSUS Noé

PATOUILLARD Guillaume

Preuves Projet Android

Documentation

- **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_CutMonster permettant de comprendre le jeu.

- **Je sais concevoir et décrire 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

DiagrammeDeCasDutilisation_CutMonster.pdf permettant de décrire les fonctionnalités de l'application.

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

Nous avons créé un diagramme de classe DiagrammeClasse_CutMonster.pdf représentant l'application.

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

Nous avons ajouté à côté du diagramme de classe dans le fichier un texte permettant de le décrire.

CODE

- **Je sais utiliser les Intent pour faire communiquer deux activités.**

Nous avons utilisé des Intent pour la navigation entre nos différentes activités (AccueilActivity, ScoreActivity, JeuActivity).

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

Nous avons utilisé l'API 16 permettant une grande compatibilité entre tous les téléphones Android.

- **Je sais distinguer mes ressources en utilisant les qualifier**

Nos ressources sont séparées dans des dossiers différents :

- le layout pour les vues
- values pour les styles, couleurs, gestion des thèmes et textes.
- drawable pour toutes les images
- raw pour la musique du jeu

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

Nous avons plusieurs vues, 2 utilisent un LinearLayout et la vue de jeu utilise un constraintLayout situées dans le package main/res/layout.

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

Notre activité d'accueil permet à l'utilisateur de rentrer son pseudo afin qu'il puisse jouer. L'activité score affiche la liste de joueurs avec leurs nombres de points.

- **Je sais coder une application en ayant un véritable métier**

Nous avons séparé dans des packages nos classes dans le modele situé : main/java/modele. Par exemple un package entite, boucleur, deplaceur avec leurs classes qui leurs sont propres etc.

Un singleton a été utilisé pour la classe PersistenceManager.

- **Je sais parfaitement séparer vue et modèle**

Nous avons séparé la vue et le modèle. En effet, les vues sont gérées dans les activités qui appellent des méthodes situées dans le GameManager principalement.

- **Je maîtrise le cycle de vie de mon application**

Nous maîtrisons le cycle de vie de notre application à travers le onDestroy dans l'activité jeu qui permet de sauvegarder nos données ainsi que stopper le boucleur. La musique elle est appelée dans le onResume et arrêté dans le onPause.

- **Je sais utiliser le findViewById à bon escient**

Nous avons utilisé le findViewById pour récupérer l'élément rentré par exemple par un utilisateur dans l'editText pour son pseudo mais aussi pour récupérer l'imageView du défenseur.

- **Je sais gérer les permissions dynamiques de mon application**

Nous n'avons pas eu à gérer les permissions dynamiques de notre application puisque l'accéléromètre n'en a pas besoin.

- **Je sais gérer la persistance légère de mon application**

Nous n'avons pas eu besoin de gérer la persistance légère puisque la vue d'accueil, l'orientation n'est pas bloquée mais les strings avec leurs ID l'induisent indirectement. De plus, nous bloquons l'orientation en mode portrait pour la vue du jeu.

- **Je sais gérer la persistance profonde de mon application**

Nous sauvegardons la liste des joueurs du jeu comprenant leurs pseudos et leurs nombres de points correspondants. Quand le joueur a perdu ou qu'il appuie sur le

bouton retour du jeu, nous ajoutons son pseudo et son nombre de points dans une liste (situé dans la classe PersistenceManager) que l'on persiste dans un fichier local. La classe Joueur implémente de ce fait Serializable. Cette persistance est visible dans la classe PersistenceManager situé dans le package manager, ainsi que dans JeuActivity.

- **Je sais afficher une collection de données**

Nous avons affiché une liste de joueurs comprenant leurs pseudos et leurs nombres de points correspondants situé dans la vue page_score qui implique son activité : ScoreActivity .

- **Je sais coder mon propre adaptateur**

Nous avons utilisé l'adaptateur pour afficher la liste de joueurs dans la page_score. Cet adaptateur est situé dans le package adaptateur situé dans le package view de l'application.

- **Je maîtrise l'usage des fragments**

Nous n'avons pas eu le temps d'utiliser des fragments. On aurait pu les utiliser pour l'affichage de notre liste de joueurs.

- **Je maîtrise l'utilisation de Git**

Nous avons utilisé GIT pour ce projet. Nous avons créé 2 branches pour travailler avec par exemple une branche tir pour implémenter une fonctionnalité de tir que nous n'avons finalement pas gardé.

Dès qu'une fonctionnalité était fini et totalement fonctionnelle, nous poussons notre travail dans le master pour que l'autre puisse se tenir à jour.

APPLICATION

- **Je sais développer une application sans utiliser de librairies externes**

Nous n'avons pas utilisé de librairies externes pour ce projet.

- **Je sais développer une application publiable sur le store.**

Nous avons fait la demande pour publier notre application sur le store. Elle devrait être publiable.

- **Je sais utiliser l'accéléromètre et ou le gyroscope**

Nous avons utilisé l'accéléromètre dans l'activité JeuActivity afin de déplacer le défenseur. En effet, pour couper les monstres qui tombent, l'utilisateur doit incliner son téléphone à gauche ou à droite pour déplacer le défenseur.