Haute école arc

Développement mobile

BallJump

Auteurs :
Da Cruz Costa Pedro Emmanuel
Piquerez Thibaut
Pedretti Maël

Responsables :
Rizzoti Aïcha
Jupil Dany

28 janvier 2018

Le projet BallJump est réalisé dans le cadre du cours de développement mobile, une branche de 3ème année de Bachelor au sein de la Haute-École Arc — Ingénierie, section Développement Logiciel et Multimédia. Le présent document décrit le déroulement du projet et présente les fonctionnalités implémentées.

haute école
neuchâtel berne jura
arc
ingénierie
saint-imier le locle delémont

Table des matières

1 Résumé

Le projet Ball Jump a été réalisé dans le cadre du cours de Développement mobile, un cours de 3ème année de Bachelor au sein de la Haute-École Arc - Ingénierie, section Développement Logiciel et Multimédia. Ce document décrit le déroulement du projet et présente les fonctionnalités implémentées. Ball Jump est un jeu mobile qui se déroulere sur le principe du jeu "Doodle Jump".

2 Principe du jeu

Le jeu se déroule de la manière suivante. Le joueur doit diriger un personnage en inclinant le téléphone de gauche à droite. Le terrain de jeu est composé d'une multitude de plateformes qui font rebondir le personnage lorsqu'il atterit dessus.
Le but du jeu est d'arriver le plus haut possible. Les plateformes se génèrent au fur et mesure que le personnage monte. Des items qui permettront de débloquer des bonus se trouveront également dans le décore.

3 Structure

3.1 MenuActivity

Première activitée à être lancé lors de l'ouverture de l'application. Elle permet juste d'afficher le menu principal qui contient les boutons suivant.

- PLAY : lancer le jeu
- STATISTICS : affiche des statistiques sur les scores
- SHARE : partage de scores en bluetooth
- CREDIT : crédit du jeu

Ensuite en fonction des boutons pressés l'activité va lancer les activités correspondante.

3.2 Statistics

3.3 SensorAccelerationActivity

Cette classe s'occupe d'instancier le panel ainsi que de gèrer les capteurs. Elle utilise l'accéléromètre et le magnétomètre pour pouvoir trouver l'angle de l'appareil. Pour cela la classe implémente SensorEvent-Listener et doit surcharger la fonction onSensorChanged() qui est appelé a chaque changement de valeur des capteurs.
Dans cette fonction on récupère la valeur des deux capteurs et on utilise la classe SensorManager pour calculer l'angle de la manière suivante.
On passe les résultats des capteurs dans la fonction getRotationMatrix() de la classe SensorManager qui prends comme paramètre 4 tableaux, 2 qui viennent des capteurs et 2 tableaux dans lesquels seront stockés les résultats. Un de ces tableaux correspond a une matrice de rotation qui est passée dans une autre fonction de SensorManager qui se nomme getOrientation() qui elle va retourner un tableau d'angles selon tous les axes.
Donc on passe l'angle qui nons intéresse au panel pour qu'il fasse bouger le personnage du jeu.

3.4 GamePanel

3.5 plateform

3.6 Player

3.7 GameOver

4 Fonctionnalités

Cette section développe les fonctionnalités du téléphone qui sont utilisés par l'application.

4.1 Capteurs

Dans le cadre de ce projet, les capteur utilisé sont l'accéléromètre et le magnétomètre.
Selon Wikipédia,

> *Un accéléromètre est un capteur qui, fixé à un mobile ou tout autre objet, permet de mesurer l'accélération linéaire de ce dernier. On parle d'accéléromètre même lorsqu'il s'agit en fait de 3 accéléromètres qui calculent les accélérations linéaires selon 3 axes orthogonaux.*

Il s'agit plus simplement d'un capteur qui permet de détecter les mouvements du téléphone. Dans ce projet, il est utilisé pour détecter les changements d'orientation afin de déplacer le joueur sur l'axe horizontal du jeu.
Le magnétomètre lui est un capteur qui détecte les changements dans le champ magnétiques avoisinant. Dans une utilisation combinée, ces deux capteurs permettent de détecter l'inclinaison du téléphone et donc de déplacer le joueur.

4.2 Plateformes

4.3 Rebonds

Pour les rebonds c'est assez simple, quand une colision est détecté entre le joueur et une plateforme on assigne une vitesse au joueur positive. Cela a pour éffet de faire monter le personnage. A chaque déplacement du joueur la vitesse est décrémentée. Donc il va monter de moins en moins vite et ensuite la vitesse va devenir négative par conséquent le joueur va redéscendre jusqu'à atteindre une vitesse maximum. Ensuite si il entre a nouveu en collision il va remonter à nouveau.
Le joueur monte seulement jusqu'à le moitié de l'écran et ensuite se sont les plateformes qui descandent ce qui donne l'impression que le joueur monte. Cette manière de faire évite que le joueur sorte de l'écran vers le haut si il saute plusieurs fois de suite.

4.4 Statistiques

4.5 Persistance

Dans le but de pouvoir créer des statistiques, les scores sont sauvegardés dans un fichier texte. Afin de simplifier le stockage et n'ayant qu'un champ à sauvegarder, cette solution est avantageuse comparée à une utilisation d'une base de données.

5 Problèmes rencontrés

Cette section détaille les problèmes rencontrés, qu'il s'agisse d'un point de vue matériel ou de développement.

5.1 Liés aux appareils

5.1.1 Capteurs

Tous les téléphones ne disposant pas des mêmes capteurs et l'emulateur ne permettant pas de simuler des orientations, il a été difficile de pouvoir tester l'application. En effet, les tablettes prêtées par l'école ne disposent pas d'accéléromètre.
Ce problème a induit un codage à "l'aveugle" pour les étudiants ne disposant pas d'autres appareils Androïd. De plus, il n'a été possible de réaliser au préalable les tests uniquement sur un téléphone.

5.1.2 Puissance de calcul

Tous les appareils ne disposant pas du même processeur et de la même quantité de RAM, le jeu ne se déroulait pas à la même vitesses sur tous les périphériques.
Il a été nécessaire de trouver un moyen de ralentir les appareils trop rapides afin que l'expérience utilisateur soit toujours autant agréable. Les appareils plus lents quant à eux ne sont pas ralentis et affichent le jeu au maximum de leur capacités.
Il est donc possible que le jeu se déroule plus lentement sur certains appareils, mais jamais "trop" vite.

5.2 Liés au développement

5.2.1 Collisions pas détectées

Il y avait un problème quand le joueur retombait à la vitesse maximale car le joueur pouvait passé de dessus a en dessous d'une plateforme en une frame. Donc quand on vérifie les collisions il n'y en a pas. Pour régler le problème nous avons dû augmenter l'épasseur des plateformes virtuellement. C'est a dire que visuellement elles ne chagent pas mais dans la détection des collisions elles sont plus éapaisse. De cette manière il n'est plus possible de traveré une plateforme en une seul frame.

5.2.2 Panel qui ne se redessine pas

Le problème était que lorsqu'on pressait sur PLAY dans le menu principal une page blanche s'affichait parfois pendant plusieurs dizaine de secondes. Après quelques recherches on a trouvé que le problème venait du fait qu'on lockait le canvas pour pouvoir déssiner dessus mais on unlockais pas au bon endroit qui avait pour effet de ne pas redessiner les modifications.
Après cette modification on a remarqué que l'affichage mettait un petit temps de chargement donc pour règler le problème le jeu ne démarre pas tout pendant que le joueur n'a pas touché l'écran comme cela le joueur démarre quand il est prêt-

6 Bugs connus

7 Auto-critique

8 Conclusion