Mathilde ELOY
Elise JOFFRE

G4

RAPPORT D'ANDROID

Création du jeu mobile TETRIS

INSTITUT UNIVERSITAIRE DE TECHNOLOGIE
iut du Limousin

M4104C
Semestre 4
Département Informatique – 2ème année

Université de Limoges

Table des matières

I. Contexte

Dans le cadre du cours de développement d'applications mobiles du Semestre 4, nous avons décidé de réaliser l'application mobile Tetris.

Le déroulement de l'application s'effectuera en plusieurs étapes :

- Le joueur pourra donc s'inscrire en rentrant son pseudo et son email. Dans le cas où il est déjà enregistré il devra sélectionner son pseudo et lancer sa partie
- Le jeu sera lancé avec un plateau de Tetris classique.
- Le joueur aura la possibilité de consulter son score et le score des joueurs ayant joué sur son téléphone.
- A terme le joueur pourra partager son score sur les réseaux sociaux.

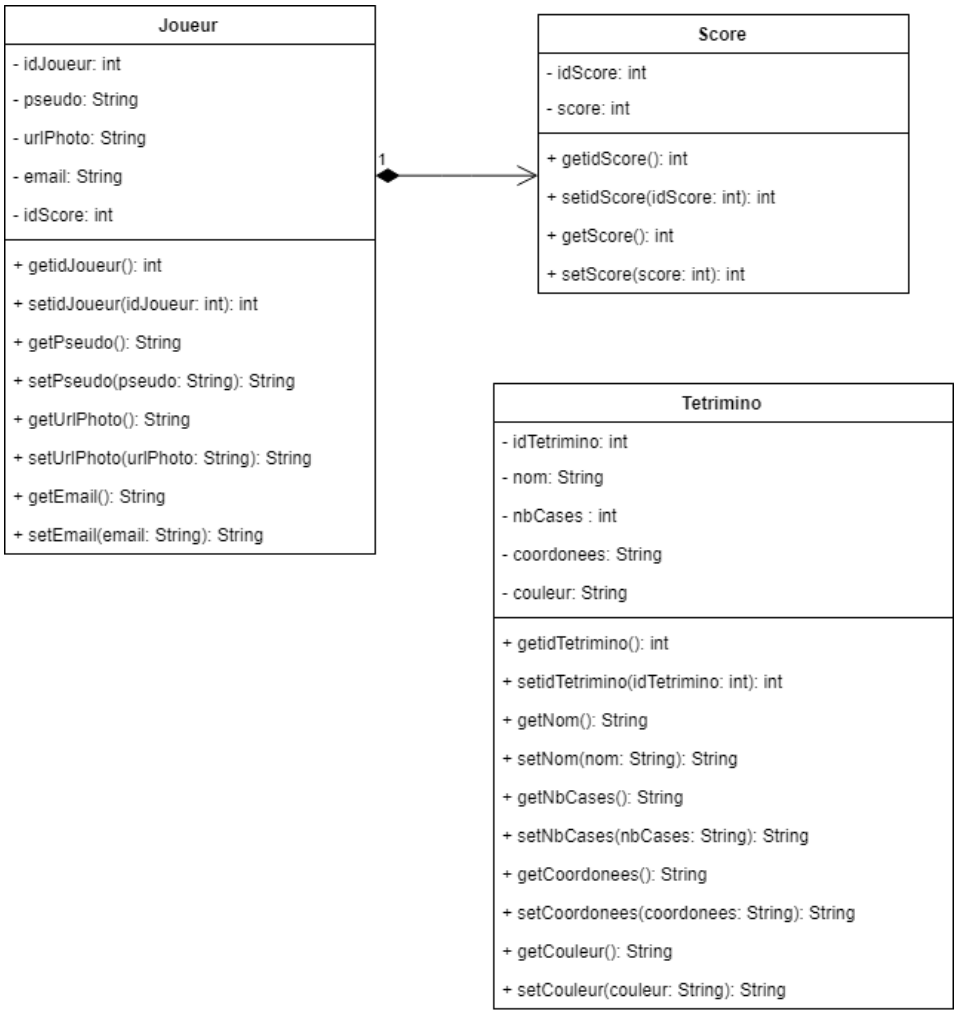
II. Choix techniques

1. Le langage et l'environnement de développement

Nous souhaitions développer l'intégralité de notre application sous Android Studio mais face au temps imparti et à la complexité de gérer les collisions entre les pièces nous nous sommes réorientés. En effet, nous avons préféré développer notre application sous Unity qui est un moteur de développement pour concevoir des expériences interactives et des jeux multiplateformes en 2D et en 3D. Le langage utilisé pour ce développement est le CSHARPE et le code ainsi programmé est alors portable sur de multiples plateformes comme Android. Ce logiciel était donc parfaitement adapté à notre problématique, c'est pour cela que nous l'avons choisi.

2. La base de données

Une fois la partie jeu terminée sous Unity, nous souhaitions exporter le Gradle du jeu sous Android studio puis associer une base de données SQLITE qui suivrait le modèle conceptuel de données suivant :



Mais nous avons préféré simplifier le modèle car nous n'avions pas besoin de la table « Tetrimino » sachant que tout est implémenté dans Unity. Nous avons donc une seule table « Joueur » contenant à l'id, le pseudo, l'email et le score du joueur.

3. Ajout d'une photo au joueur

L'une de nos idées au départ était de permettre au joueur d'associer à son profil une photo qui serait prise à son inscription. Il aurait la possibilité de prendre une photo en direct ou encore de prendre une photo dans sa galerie photos. Mais nous n'avons pas implémenté cette fonctionnalité pour l'instant car nous avons fait face à la difficulté de lier un code implémenté sous Android et sous Unity que nous détaillerons dans les difficultés. Nous avons donc mis de côté les fonctionnalités natives liées au téléphone.

III. Les fonctionnalités

1. S'inscrire

Lorsque l'utilisateur lance l'application, il a la possibilité soit de choisir un profil déjà enregistré soit de s'inscrire. Lorsqu'il s'inscrit l'utilisateur dois remplir son pseudo ainsi que son adresse mail, il est alors automatiquement ajouté à la base de données. Il aura alors la possibilité de se connecter à son profil.

2. Se connecter

Lorsque l'utilisateur a déjà enregistré un profil, il peut alors sélectionner son profil via une liste déroulante. S'il n'a jamais joué son score est de 0, sinon meilleur score est enregistré dans la base de données et il a donc pour objectif de battre son record et de se comparer aux autres joueurs !

3. Jouer

Une fois son profil sélectionné l'utilisateur commence la partie de Tetris. Les Tetriminos sont choisis aléatoirement par une fonction du script. Le joueur peut bouger les pièces durant leur descente via l'écran tactile du téléphone. En glissant à droite ou à gauche, le Tetrimino se déplacera vers la direction choisie. Si l'utilisateur glisse vers le bas, la vitesse de chute de la pièce est plus rapide. En tapant sur l'écran, on effectue une rotation de la pièce. Le score s'implémente au fur et à mesure. Le joueur gagnera 40 points pour une ligne pleine, 100 points pour deux lignes pleines en même temps 300 points pour trois lignes et 1200 points pour quatre lignes. Lorsqu'une ligne est remplie, elle disparaît et les lignes supérieures descendent du nombre de lignes détruites. Le joueur reçoit également des points supplémentaire par le nombre

de tours où il survit (un tour = une pièce générée). Lorsqu'une pièce dépasse le haut du plateau de jeu, c'est Game Over et le joueur est redirigé sur une autre page.

4. Game Over et gestion des scores

Une fois la partie perdue, on récupère le score du joueur sur cette partie et on le compare à son meilleur score qui est le score enregistré dans la base de données. Si le nouveau score est meilleur alors il remplace le score déjà existant et on va ensuite le comparer au meilleur score des autres joueurs pour savoir s'il apparaîtra sur le tableau des highscores. Dans le cas où le nouveau score est inférieur à l'ancien rien ne se passe.
Depuis le menu Game Over l'utilisateur peut rejouer ou changer de joueur. Il peut également partager son score par email (en sélectionnant un joueur) et via les réseaux sociaux. Pour cela on utilise le plugin Cross Platform Native Plugins qui détecte si le portable de l'utilisateur est en capacité de partager le score via la plateforme choisie.

IV. Les difficultés

1. Android Studio versus Unity

La première difficulté consistait au choix du langage et de l'IDE. Nous avions l'ambition de réaliser notre Tetris sous Android Studio, mais compte tenu du temps de réalisation accordé à ce projet et la difficulté que nous nous sommes imposée en choisissant ce sujet, nous avons opté pour utiliser Unity. La page de jeu devait être là seule page crée sous Unity, mais ne réussissant pas à passer les variables de Unity à Android Studio nous avons finalement décidé de réaliser entièrement le projet sous Unity.

2. La base de données

Au début du projet, nous voulions donc réaliser la base avec SQLITE en la reliant à Android. Nous avons dû trouver un nouveau moyen d'utiliser notre base de données avec Unity. Nous avons donc créé notre table utilisateur avec SQLITE Administrator et nous l'avons enregistré dans les Assets de notre projet Unity. Il s'agit donc d'une base interne qui n'est pas relié à Unity. La seconde difficulté par rapport à la base de données venait du fait que nous ne savions pas comment interagir avec les données dans nos scripts C#. Ce problème a été résolu par des tutoriels et de la pratique. Enfin, nous avons eu des difficultés à partager les variables d'une scène à une autre, mais une fois le fonctionnement compris cela n'a plus été compliqué.

3. Création de l'APK

Ayant quelques soucis avec la version du SDK pour Unity nous avons dû trouver un moyen de créer notre APK par un autre moyen. Pour cela, nous créons un gradle depuis Unity (à la place de build et run l'APK). Nous avons du utiliser des versions antérieures pour le SDK et le JDK, les nouvelles versions transformait nos méthodes en méthodes *depreciated*.

V. Notre code

Lien github : https://github.com/m-eloy/Tetris

Présent sur le github : dossier Unity (Contenant les fichiers et les scripts C# avec notre code), fichier APK.

Conclusion

Ce projet a été très intéressant pour nous deux, nous avons aimé le fait de pouvoir choisir un sujet dans le domaine de l'application mobile. La création d'un jeu ludique était quelque chose que nous voulions essayer, c'est pourquoi nous avons choisi un jeu classique à reproduire. Ce sujet nous a également été bénéfique afin de découvrir un peu plus l'IDE Unity et le langage C# bien que nous regrettons un peu le fait de ne pas avoir pu combiner Android Studio avec Unity. Une des améliorations que nous aurions aimé apporter est le fait de permettre au joueur d'avoir une photo et de la prendre depuis son portable.

Cependant, Unity est un logiciel pratique qui nous a permis de réaliser plus facilement une interface graphique qui nous aurait pris plus de temps en utilisant Android Studio. Enfin, cela nous a une fois de plus confirmé notre intérêt pour le développement mobile que nous continuerons à découvrir dans les années à venir.