

MEMORY

BEKKA Jazil 12317132

TRAN Alice 12311202

Table des matières

Introduction	3
Description du projet et règles implémentées	4
Modélisation de l'application	5
Problèmes rencontré et solution apporté	7
Organisation du travail	8
Conclusion.....	9
Annexes	10

Introduction

Afin de valider l'U.E plusieurs projets nous ont été proposés, nous avons choisi le jeu Memory car c'est un jeu connu par tous. De plus, lorsque nous avons lu le descriptif du jeu, nous avons déjà une idée de la façon de procéder afin de développer cette application.

Nous commencerons donc par vous présenter le projet, puis sa modélisation, les problèmes que nous avons rencontrés et enfin comment nous nous sommes organisés afin de réaliser cette application.

Description du projet et règles implémentées

Le projet que nous avons choisi fut le jeu de Memory, lors du lancement de l'application nous retrouvons le menu principal qui est composé de quatre boutons, le premier bouton « jouer » permet de lancer le jeu, le deuxième « score » quant à lui affiche le meilleur score obtenu, le troisième bouton « about » permet d'afficher les noms du binôme, quant au dernier bouton « système », il permet de gérer la musique à l'aide d'un switch et de quitter l'application avec le bouton « quitter ».

Lorsque le jeu se lance, vingt cartes faces cachées apparaissent sur l'écran, le but du jeu est de retrouver les dix paires dans le temps imparti et avec un nombre de coups limité. Pour cela il suffit de toucher les cartes faces cachées. Lorsqu'on touche une carte, celle-ci s'efface afin de faire apparaître le symbole chinois qui se trouve derrière. Lorsque deux cartes sont affichées, il y a deux possibilités, soit les symboles sont identiques et cela compte alors comme une paire trouvée, celle-ci reste alors affichée sur l'écran. Où au contraire les symboles sont différents, dans ce cas les deux cartes se colorient donc et redeviennent des cartes face cachée.

Concernant la limite de temps elle est fixée à une minute, le chronomètre démarre dès que la première carte se dévoile. Quant aux nombre de coups maximum celui-ci est de trente coups, à chaque fois qu'une paire est dévoilée, on soustrait un au nombre de coup maximum restant. À la fin du jeu, si le joueur gagne son score est affiché, celui-ci est calculé grâce à la formule suivante : nombre de coups restants multipliés par le temps restant. Si ce score est le meilleur alors celui-ci est enregistré dans les préférences afin de le mémoriser de manière persistante.

Ces paramètres ont été choisi afin de rendre le jeu plus difficile, nous avons également choisi d'utiliser des sinogrammes chinois afin de duper le joueur, car ceux-ci sont assez similaires et durs à différencier ainsi que libre de droits.

Modélisation de l'application

Notre programme contient huit fichiers JAVA :

MainActivity est l'activité principale qui se lance dès le lancement du jeu, elle contient 4 boutons qui dans l'ordre permettent de lancer le jeu, afficher le meilleur score obtenu, afficher le nom du binôme et gérer la musique ou mettre fin à l'application. C'est aussi dans cette activité qu'est déclarée la musique. Cf. *annexe p10*.

LaunchGame est l'activité permettant de lancer le jeu on y affiche la Progress bar qui permet de gérer le temps qu'a le joueur afin de finir le jeu, on y affiche également un bouton qui permet de rejouer durant la partie ou à la fin de celle-ci. Et pour finir on y affiche la view JeuView. Cf. *annexe p11*.

JeuView, c'est dans cette view que contient le code du jeu. Il s'agit d'une surface view, car celle-ci nous permet de jouer des animations, elles nous donnent plus de liberté. Dans cette View on y dessine notre terrain de jeu qui est un plateau de 4x5 cartes. Les cartes sont initialisées dans un tableau face caché et de manière aléatoire ce qui fait qu'elles changent de position entre chaque partie. Afin de retourner les cartes nous utilisons l'événement du doigt sur l'écran. Lorsque l'on clique sur une carte celle-ci se retourne, une fois la deuxième carte retournée, nous analysons les deux cartes effet nous attribuons des entiers à chaque carte donc si les deux cartes possèdent le même entier alors elles sont identiques. Dans ce cas nous les laissons retournés et incrémentons le compteur qui permet de gagner, si celui-ci arrive à 10 la partie est gagnée, on efface donc le plateau de jeu et faisons apparaître le score ainsi qu'une image de victoire. Dans tous les cas nous décrétons le compteur de coups restant à jouer de un pour chaque paire retournée. Si les cartes ne sont pas similaires dans ce cas on les dissimule. Le fait de découvrir les cartes ou les dissimuler est fait à l'aide d'animation que l'on gère à partir d'un thread . Nous avons fait en sorte qu'aucune action ne peut être effectuée tant que les cartes ne sont pas cachées comme cela le joueur a le temps de mémoriser les cartes et cela évite au tricheur de retourner les cartes rapidement pour battre le temps et ne pas les mémoriser, avec ce système ils n'ont pas le choix que d'attendre et respecter le principe du jeu. Si le compteur du nombre de coups maximum arrive à zéro ou que le temps est écoulé dans ce cas la partie est perdue et on empêche le joueur de retourner d'autre carte tout en affichant une image montrant que celui-ci a perdu. Cf. *annexe p12*.

Initialisation est une class comportant une méthode qui permet d'attribuer des chiffres de 0 à 9 deux fois dans une ArrayList et de les mélanger, elle nous permet de mettre les cartes aléatoirement dans le plateau de jeu.

Timer est une class qui permet de gérer la Progress bar sur un thread secondaire, en effet elle nous permet de le démarrer ainsi que de le stopper.

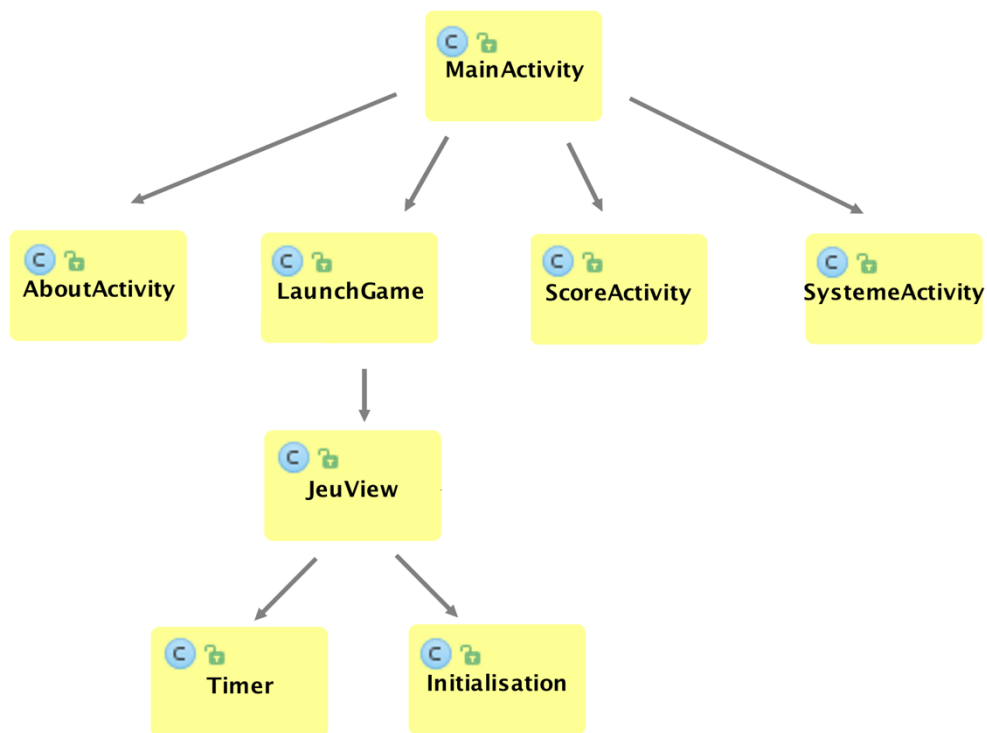
SystemeActivity est l'activité permettant de gérer la musique, elle contient un switch qui permet d'activer ou de désactiver la musique du jeu et de sauvegarder l'état de manière

persistante grâce aux sharedPreference. Elle possède aussi un bouton « quitter » qui permet de quitter l'application. Cf. annexe p13.

AboutActivity est l'activité qui ne contient qu'une image en background affichant les noms du binôme ayant participé à la réalisation de l'application et la personne ayant participé à la réalisation de la musique. Cf. annexe p14.

ScoreActivity est une activité qui affiche le meilleur score obtenu. Le score est affiché dans une TextView et sauvegarder grâce aux sharedPreference.

Diagramme des classes



Problèmes rencontré et solution apporté

Lors de la réalisation de ce projet nous avons rencontré certains problèmes. Le premier problème fut la résolution de nos téléphones, en effet ayant des téléphones différents, la résolution n'était pas la même, nous avons des téléphones en full HD et en 2K, il fallait donc gérer la taille des cartes, leur disposition sur l'écran, faire en sorte que le background s'adapte à chaque écran de téléphone.

Le deuxième problème apparut lorsqu'on a essayé de rajouter des animations. Nous avons commencé à coder l'application grâce à des simples layout en disposant des boutons comme des cartes cependant les animations ne fonctionnaient pas et on ne pouvait donc pas afficher les cartes correctement. Nous avons donc du tout refaire en surface view ce qui nous a prit environ 2 jours.

Un autre des problèmes que nous avons rencontrés est le fait que nous ne pouvions pas rejouer directement. Nous avons donc créé un bouton réinitialisant le jeu ainsi que la Progress bar et le nombre de coups restant à jouer.

Organisation du travail

Nous avons sauvegardé nos données sur github. Ce fut l'occasion pour nous de nous familiariser avec une plate-forme communautaire très utilisée dans le monde du développement.

La communication se faisait majoritairement par Skype ou encore TeamViewer. Nous nous sommes d'abord concentré sur le jeu en lui-même en affichant le terrain avec les cartes, les aléatoirement, les retourner en cliquant dessus, valider lorsqu'une paire est trouvée. Faire en sorte que l'on gagne et perde. Puis sur le plus telle que la musique, les animations ou encore le score. Le jeu en lui-même a été réalisé rapidement, nous avons donc pu profiter du temps qui nous restait afin d'ajouter des plus et d'optimiser l'application.

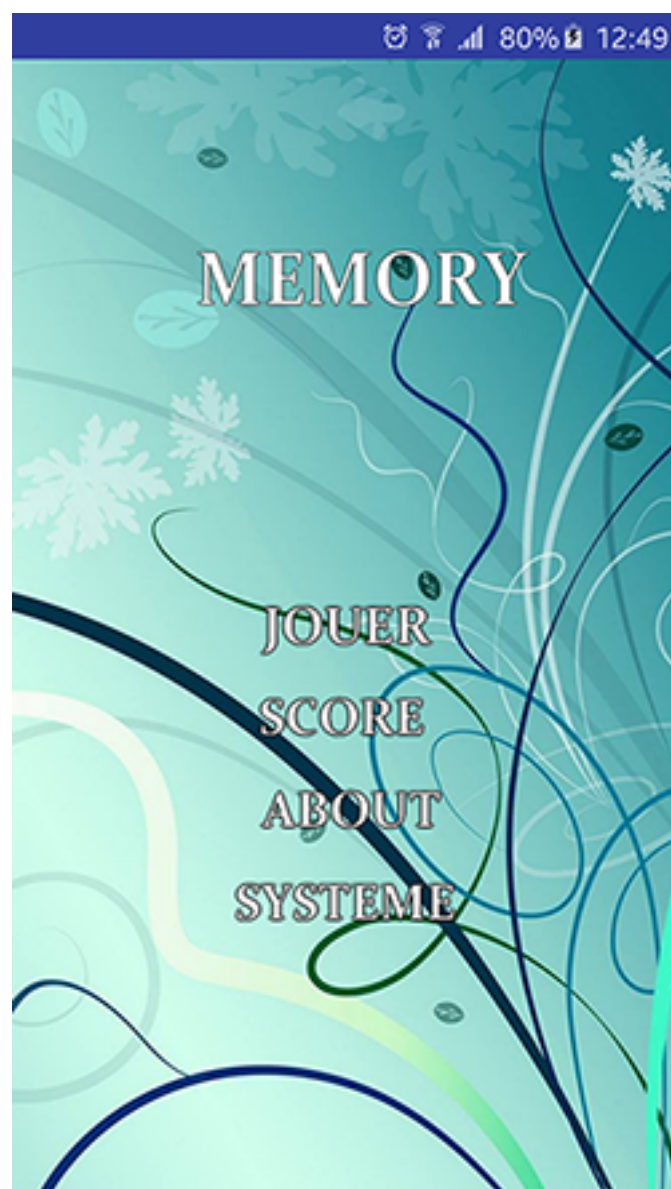
Conclusion

Créer cette application a été bénéfique, en plus de nous avoir permis d'avoir pour l'une une première expérience avec Android Studio, pour l'autre ce fut l'occasion de se familiariser avec celui-ci. Avoir une application sous Android est un grand plus car en effet le domaine des applications mobiles est en pleine expansion. De plus celui-ci est un domaine qui nous intéresse particulièrement et dans lequel on souhaiterait peut-être continuer.

Sortir une application peut aussi nous aider lors de notre recherche d'emploi ou de stage puisque nous pouvons présenter directement notre application lors de l'entretien avec notre employé afin de lui montrer le travail réalisé.

Pour ce qui est des améliorations possibles on pourrait éventuellement dans une version 2.0 ajouter des niveaux de difficulté changeant le temps et le nombre de coups restant à jouer, éventuellement changer aléatoirement la taille du terrain.

Annexes



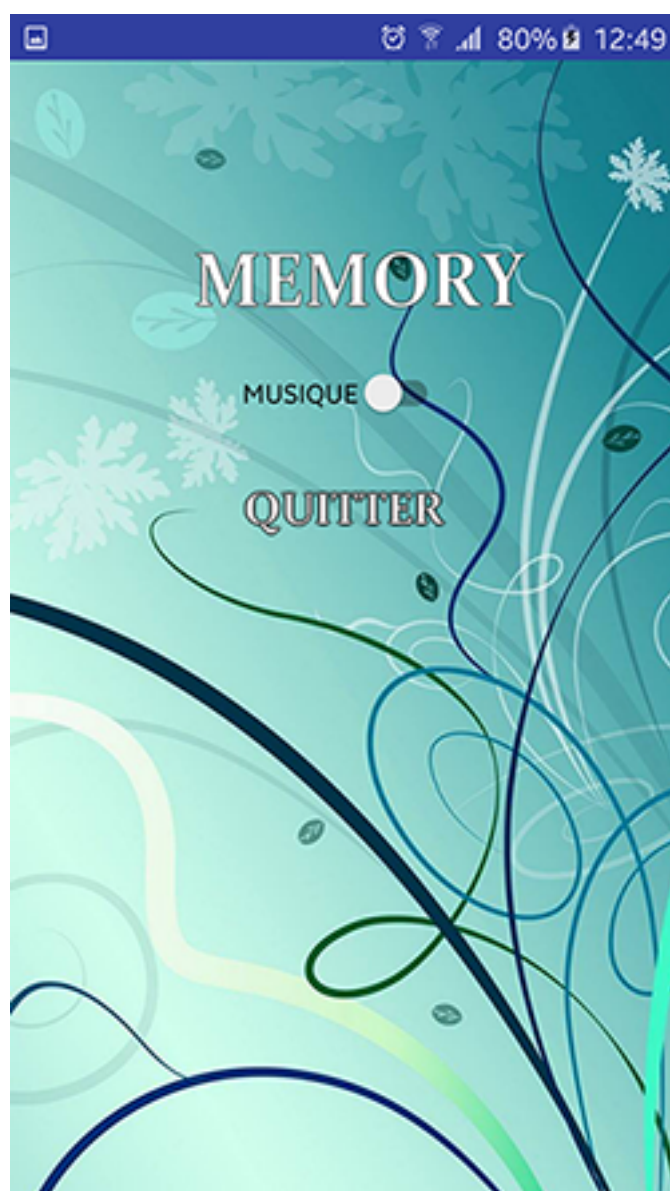
Accueil de l'application



La vue de l'activité LaunchGame



Exemple de jeu au cours d'une partie



La vue de l'activité « systeme »



La vue de l'activité « about »