

INF8405 – Informatique mobile

Travail Pratique N°1 – Application de jeu pour Android

1. Informations générales

Session	Hiver 2017
Auteur	Fabien BERQUEZ
Chargé de laboratoire	Fabien BERQUEZ (fabien.berquez@polymtl.ca)
Date de début	19 janvier 2017
Date de remise	15 février 2017 à 23h 55min au plus tard
Laboratoire	L-4708
Pondération	15%
Directives particulières	<ol style="list-style-type: none"> 1. Tout projet rendu sera pénalisé de 5 points s'il est soumis par une équipe dont la taille est différente de celle définie (3), sans l'approbation préalable du chargé de laboratoire. 2. Rapport à rendre en format PDF ou Word. 3. Le code source + le rapport sont à rendre dans une archive ZIP, avec l'APK de l'application 4. Soumission du rapport et des travaux réalisés par moodle uniquement (http://moodle.polymtl.ca). 5. Tout retard de soumission du rapport et/ou des travaux réalisés sera pénalisé de 5 points par jour de retard.

2. Connaissances préalables

- Langage Java (programmation orientée objet, développement d'interface graphique, programmation événementielle).
- Langage XML.

3. Environnement et outils nécessaires

- Environnement de développement Intégré (IDE) : Android studio ou Eclipse + SDK Android + Plug-in Eclipse d'Android (ADT).
- Tablette Android.

4. Objectifs du laboratoire

L'objectif principal de ce laboratoire est d'initier l'étudiant au développement de services pour des terminaux mobiles. De manière spécifique, au terme de ce laboratoire, il s'agira pour l'étudiant de :

- développer une application pour terminaux mobiles ;
- se familiariser avec les caractéristiques des applications mobiles ;
- se familiariser avec un système d'exploitation pour terminaux mobiles.

5. Présentation de l'application

Ce TP consiste à réaliser une application de jeu mono-joueur. Il s'agira de développer une version simplifiée d'un jeu de type **match-3** (des exemples bien connus sont *Candy Crush* et *Bejeweled*). Le support du match-3 est généralement une grille (rectangulaire ou avec une forme plus complexe).

Déroulement habituel d'une partie :

A l'initialisation d'une partie, la grille de jeu est remplie des différents items du jeu. Le joueur peut alors faire glisser un élément vers un autre (nécessairement adjacent, soit verticalement soit horizontalement), de manière à échanger leurs places. L'objectif de chaque coup est d'aligner au moins 3 éléments similaires, que ce soit verticalement ou horizontalement. Si plus de 3 éléments identiques sont alignés à la suite du coup, les éléments supplémentaires disparaissent également (dans les versions plus évoluées, ils génèrent des items rapportant plus de points ou permettant des bonus). Les éléments qui étaient au-dessus des éléments qui ont disparu descendent alors dans la grille, et sont remplacés par des éléments choisis aléatoirement. Il n'est pas possible d'échanger la place de deux éléments si l'échange ne permet pas de former un groupe de 3 minimum.

Chaque type de combinaisons est généralement associé à des points, et une partie commence avec un objectif de points à atteindre et un nombre de coups et/ou un temps limité pour l'atteindre.

Déroulement d'une partie pour le jeu simplifié :

Le jeu simplifié comprendra une grille rectangulaire (de taille potentiellement variable). Les items présents dans la grille seront des cercles pleins, de couleurs **rouge, bleu, vert, orange, jaune et violet**.

Les échanges de place d'items se font par un glissement du doigt sur l'écran (généralement appelé *swipe*). Pour rappel, un échange n'est possible que verticalement ou horizontalement, et si l'échange permet de former un groupe de 3 items identiques alignés verticalement ou horizontalement. Dans le cas contraire, l'échange est interdit ou doit être annulé (voir *Candy Crush* pour un exemple de comportement possible).

Lorsqu'un groupe de 3 est formé, celui-ci doit disparaître. Pour la version simplifiée, si un groupe fait plus de 3 items, il est supprimé de la même manière (il ne forme pas un nouvel item).

Les éléments qui se trouvaient au-dessus du groupe supprimé doivent descendre et de nouveaux éléments choisis aléatoirement parmi les couleurs possibles vont prendre la place qu'ils libèrent au haut de la grille.

Il n'est pas nécessaire de vérifier qu'une solution peut être trouvée dans la grille.

Il faut par contre compter le score et le nombre de coups restants. A chaque échange effectué par le joueur, son nombre de coups restants est diminué de 1.

Pour les scores :

- Un groupe de 3 : + 100 points
- Un groupe de 4 : + 200 points
- Un groupe de 5 : + 300 points

Si deux groupes (ou plus) sont supprimés en même temps, le score rapporté par le coup est l'addition des scores rapportés par chacun des groupes (exemple : suppression de 2 groupes de 3 : + 200 points). Lorsque des combos sont réalisés (après la disparition d'un groupe, un nouveau groupe est formé et peut être supprimé sans action du joueur), le score du nouveau groupe qui disparaît est multiplié par 2. Si le combo se poursuit (par exemple la disparition de ce groupe permet la formation d'un nouveau groupe), le score est multiplié par 3. Et ainsi de suite jusqu'à ce que le combo soit brisé, c'est-à-dire jusqu'à ce que le joueur ait à nouveau effectué une action.

Lorsque le nombre de coups restants atteint 0, le score final du niveau est affiché, et si le score est supérieur à l'objectif du niveau, il est possible de passer au niveau suivant.

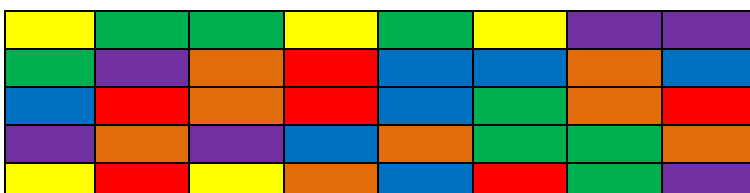
6. Requis

6.1. Requis fonctionnels

Dans le cadre du TP, le jeu à programmer fonctionne globalement comme suit : au lancement de l'application, un menu propose trois choix : **Démarrer**, **Règles du jeu** et **Quitter**. Lorsque l'utilisateur appuie sur **Démarrer**, un menu s'affiche et propose le choix du niveau. L'utilisateur ne peut par défaut sélectionner que le niveau 1. La victoire à un niveau (i.e : le nombre de points requis est atteint en le nombre de coups possibles) débloquent l'accès au niveau suivant. Si l'utilisateur revient sur la liste des niveaux, il doit pouvoir sélectionner les niveaux qu'il a débloquent. Il n'est en revanche pas demandé de conserver la liste des niveaux disponibles lorsque l'utilisateur quitte l'application. Le choix **Règles du jeu** affiche une page explicative des règles du jeu. Le bouton **Quitter** ferme toutes les activités de l'application (et donc l'application elle-même).

De manière spécifique, les fonctionnalités attendues de l'application sont :

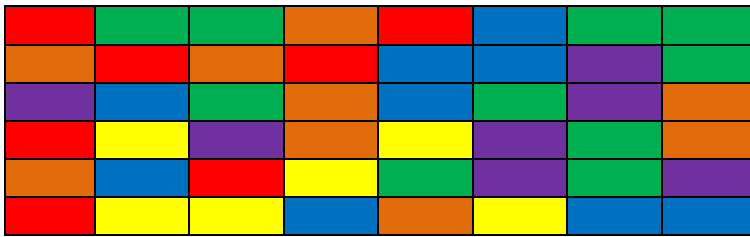
- ✓ Choix du niveau au lancement de l'application. Le joueur peut également abandonner à cette phase et quitter le jeu.
- ✓ Initialisation d'une partie de jeu en fonction du niveau de jeu. Des exemples de grilles initiales sont fournies à la **Figure 1**. Seuls ces exemples devront être pris en compte dans la réalisation de l'application.
- ✓ Jouer une partie (de manière tactile évidemment).
- ✓ Proposition de passage au niveau suivant lorsqu'un niveau est remporté.
- ✓ Possibilité de naviguer et de revenir aux niveaux déjà joués. Seul l'état initial de la grille est rétabli.
- ✓ Lorsque l'utilisateur appuie sur **Démarrer**, il doit avoir accès aux niveaux débloquent.
- ✓ Possibilité de mettre fin au jeu à n'importe quel moment. Pour prévenir les sorties en cas de mauvaises manipulations, le joueur doit confirmer la fermeture de l'application.
- ✓ Blocage de l'accès à un niveau si la partie de niveau immédiatement inférieur n'a pas déjà été gagnée. Par exemple, le joueur n'a pas accès à la partie de jeu de niveau 3 s'il n'a pas gagné celle de niveau 2.
- ✓ Possibilité de réinitialiser une partie de jeu en cours (recommencer la partie). Dans ce cas le score est remis à 0 et le nombre de coups réinitialisé.
- ✓ Pour faciliter vos tests ainsi que la vérification du passage de niveaux, vous pouvez rajouter un bouton *Joker*, qui permet de remettre la grille à l'état initial tout en conservant le score et le nombre de coups restants (ce sera utile au cas où aucun échange n'est possible).
- ✓ Affichage en permanence à l'écran du score et du nombre de coups restants



Niveau 1

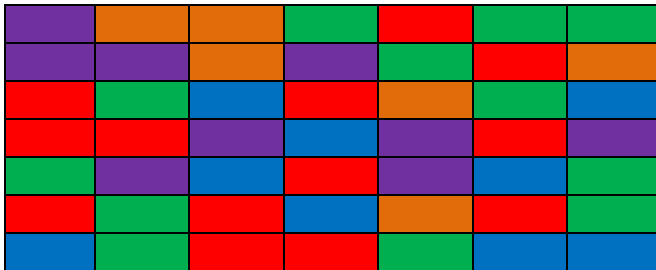
Nombre de coups : 6

Objectif de points : 800



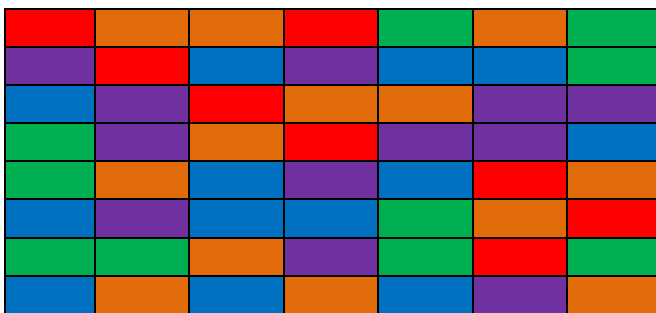
Niveau 2

Nombre de coups : 10
Objectif de points : 1 200



Niveau 3

Nombre de coups : 10
Objectif de points : 1 400



Niveau 4

Nombre de coups : 10
Objectif de points : 1 800

Figure 1 : Niveaux de départ du jeu

6.2.Requis non fonctionnels

Les exigences non fonctionnelles attendues sont les suivantes :

- ✓ Le jeu sera développé pour une plate-forme Android. À des fins de compatibilité, la version de la plate-forme à considérer est **Android 4.3.1 (API 18)** et version supérieure. Les versions antérieures à **Android 4.3.1 (API 18)** ne seront donc pas tolérées.
- ✓ Le jeu doit être accessible aux terminaux mobiles notamment les tablettes. Dans le cadre de l'évaluation, la présentation de l'application se fera uniquement sur des tablettes répondant aux requis de plate-forme indiqués ci-haut.
- ✓ L'application doit être facile d'utilisation. Un soin doit être accordé au système de navigation et à l'interface en général.
- ✓ La production d'un code lisible et suffisamment documenté (commentaires du code) est vivement attendue. Le développement doit se faire en Java avec Android Studio (recommandé) ou Eclipse

7. Conseils complémentaires

- Les fichiers XML peuvent être des bons moyens de stocker les caractéristiques des niveaux. L'API Android contient un certain nombre de méthodes pour les traiter facilement. Elles peuvent vous permettre de gagner en efficacité.
- N'oubliez pas de tirer parti des fichiers de ressources comme strings.xml et colors.xml pour votre interface ainsi que l'aspect graphique.
- Pour les graphismes, faites simple. Des cercles basiques pour représenter les éléments (l'API Android fournit des méthodes pour les créer) sont un bon moyen de le faire. Des jeux comme Candy Crush incluent beaucoup d'animations. Ne vous en préoccupez pas.
- N'hésitez pas à poser des questions !

8. Livrables

Les livrables à fournir à terme seront regroupés dans une archive (ZIP) dont le nom est formé des numéros de matricules des membres de l'équipe, séparés par le caractère « trait de soulignement » c'est-à-dire ('_'). L'archive contiendra un rapport de TP au format PDF ou Word et le code source de l'application, ainsi que l'APK généré. Le non-respect d'une de ces consignes peut être sujet à pénalité.

8.1.Rapport de TP

Le rapport doit comporter une introduction, une section consacrée à la présentation technique du travail réalisé, une section consacrée aux difficultés particulières qui ont été rencontrées et une section consacrée aux critiques et suggestions. En outre, une page de couverture est indispensable et doit porter au minimum : le nom ou le logo de l'école ; le numéro et le titre du cours ; la session ; le libellé du TP ; les noms, prénoms et matricules des membres de l'équipe; la mention « Soumis à : **nom et prénom du chargé de laboratoire** » ; la date de soumission.

8.2.Code source de l'application

Afin de faciliter la correction, vous devrez remettre un fichier .zip contenant votre code source, ainsi que l'APK compilé de votre application, et le rapport. Pour être sûrs de ne rien oublier, vous pouvez remettre l'intégralité du dossier de votre projet Android Studio (y compris les fichiers Gradle).

9. Evaluation

La grille d'évaluation est la suivante :

Rubriques	Points
Évaluation de l'exécutable : (fonctionnalité, ergonomie, portabilité, performance)	7
Évaluation du code source (implémentation de l'architecture, logique de programmation, documentation du code)	7
Rapport de TP (fond, forme)	6
Total	20