

Travail final : Le Jeu « 97 Cartes »

Dans ce travail final, vous devrez développer une application Android qui sera un jeu de cartes virtuelles. Le fonctionnement du jeu est relativement simple et fortement inspiré du jeu *98 Cards* que vous pouvez télécharger sur Google Play.

Vous pouvez également voir le jeu en action dans le vidéo suivant :

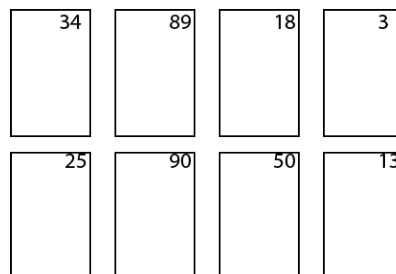
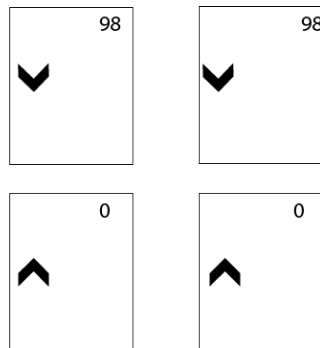
<https://www.youtube.com/watch?v=KzLhZz3XHt0>

Vous serez évalué sur :

- Le modèle objet sur lequel repose l'application
- Les activités (2) permettant de jouer au jeu
- Le bon fonctionnement de celui-ci
- La persistance du pointage à l'aide d'un table SQLite

1. Règles du jeu

A- Le jeu est composé de 4 suites (2 croissantes commençant à 0, 2 descendantes commençant à 98) et de 8 cartes initiales formant notre jeu :



- B- Les 8 cartes initiales sont numérotées aléatoirement entre 1 – 97
- C- Le but est d'écouler le maximum des 97 cartes (dont 8 sont présentes actuellement) sur les piles en respectant l'ascendance ou la descendance.

EX. : On peut placer la carte 3 sur la pile croissante où c'est indiqué 0; elle vaut maintenant 3

On pourrait ensuite placer la carte 25 sur la pile où est la carte 3 car $25 > 3$.

- D- Quand deux cartes sont libres, 2 autres apparaissent toujours avec un nombre aléatoire (autres que ceux déjà sorties)
- E- Une règle spéciale existe : la règle de 10. Si une carte vaut 10 de plus que la carte sur une suite descendante, on peut la placer. De la même façon, on peut placer une carte de 10 unités inférieure à la carte sur une pile ascendante.

EX : On place la carte 13 sur une pile initiale ascendante $13 > 0$ OK !

On peut ensuite placer la carte 3 sur la carte 13 même si la pile est ascendante car 13 est exactement de 10 plus grand que 3.

Autre ex : à 2m01 sur le vidéo ci-haut

- F- Si on joue une carte sur une pile et qu'elle ne fonctionne pas, elle est retournée à sa place de départ.
- G- La partie se termine si on a réussi à placer les 97 cartes (Bonne chance !) ou si aucun déplacement n'est possible.
- H- On doit afficher en tout temps le nombre de cartes restantes à placer et un pointage. Le pointage doit être fonction :
 - La proximité de la carte jouée de la carte sur la suite
 - La vitesse avec laquelle on a joué la carte
 - Le nombre de cartes restantes (plus de points à mesure qu'on avance dans le jeu)
- I- Finalement, un chronomètre doit être présent indiquant le nombre de temps en secondes depuis le début du jeu.

2. Modèle objet

Vous devez coder des classes représentant les objets qui seront utilisés lors du jeu; ces classes doivent contenir des attributs et des méthodes représentant les différentes actions à accomplir durant le jeu.

Toute la logique du jeu doit être représentée dans ces classes (incluant la fin du jeu). Vous utiliserez des objets provenant de ces classes dans vos 2 Activités

3. La vue, les 2 Activités :

2 Activités porteront le projet :

A) L'activité de départ sera relativement simple à déployer. Elle doit contenir :

- Le nom du jeu
- Un bouton permettant d'aller à l'activité contenant le jeu
- Le plus haut pointage que vous avez réussi (en vous référant à votre table SQLite)

B) L'activité de jeu

Dessinez l'interface avec soin; vous pouvez utiliser des `GridLayout` et des `LinearLayout`. Pour les cartes, j'ai utilisé de simples `TextView` mais vous avez le champ libre.

Présentez également des espaces pour le nombre de cartes restantes, le chronomètre et les points.

La gestion des événements est principalement axée sur le « Drag & Drop » qu'on a pratiqué à l'annexe 3.

4. La base de données simple

Lorsque la partie est terminée (lorsque plus aucun mouvement de cartes n'est possible), vous devez écrire le pointage dans une table d'une base de données locale au téléphone. Vous pouvez y entrer d'autres renseignements pertinents à votre guise. Le plus haut pointage à ce jour doit être affiché dans l'activité de départ.

5. BONUS (ce que je n'ai pas fait)

- 1. Utiliser le cadre de l'annexe 5 afin de pouvoir continuer le jeu (et le chronomètre) quand on ferme l'app ou lorsque qu'elle perd l'évidence.
- 2. Faire un bouton « annuler dernier mouvement » qui retourne la dernière carte jouée dans le jeu de l'utilisateur

6. Grille d'évaluation sommaire

Modèle objet	<ul style="list-style-type: none">• Génération des cartes à valeur aléatoire• Attributs• Méthodes représentant les actions à effectuer dans le jeu	25%
Activités	<ul style="list-style-type: none">• Présentation• Fonctionnalités• Chronomètre• Pointage	60%
Accès à la base de données		10%
Commentaires / indentation / standards de programmation		5%
Bonus 1		+5%
Bonus 2		+5%