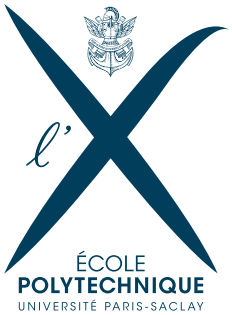
SET MULTIJOUEURS

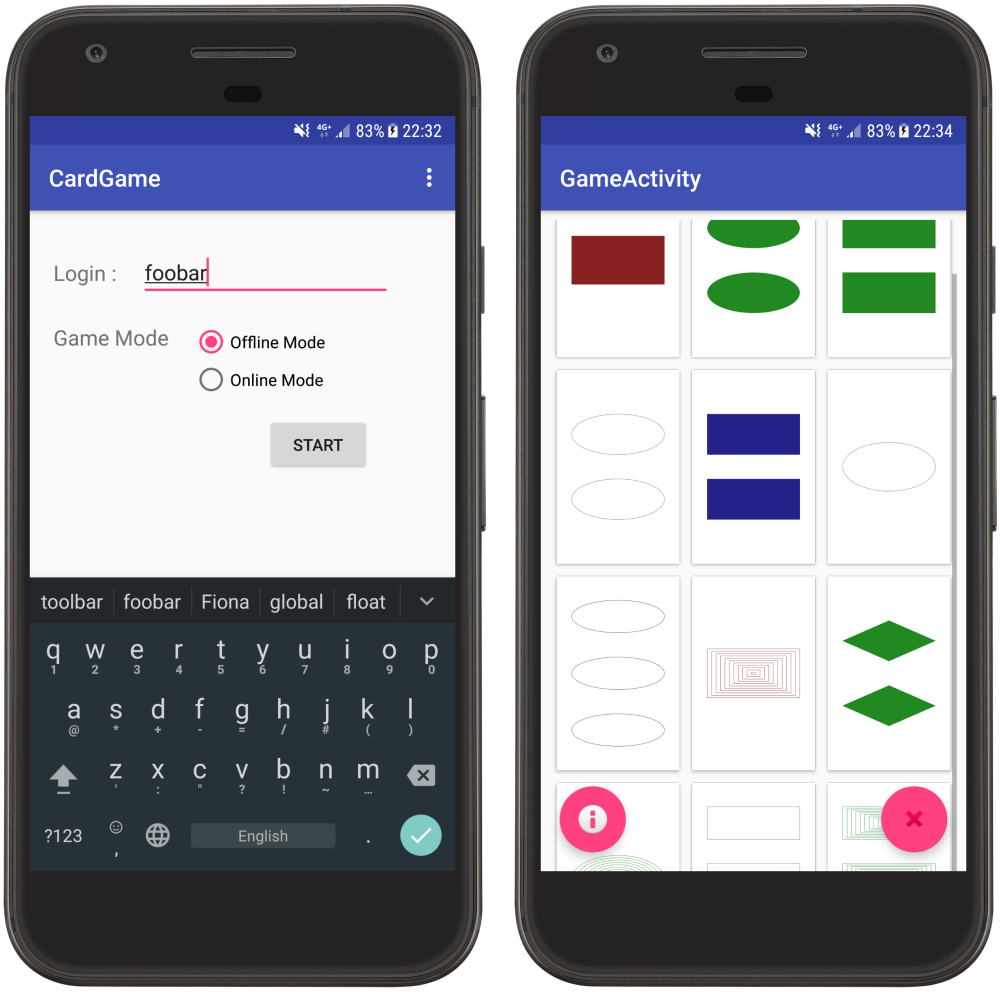
06/04/2017

Zuli HUANG
Yunxiang ZHANG



1 Application à un joueur

Dans un premier temps, nous avons développé une application pour un joueur seul sous Android. Le joueur peut choisir son propre compte pour accéder au jeu. Dans la graphe 1a, nous proposons au joueur deux genres de jeu. C'est-à-dire en ligne et off ligne. Par défaut, le choix est mis à `Offline Mode`. Nous commençons le jeu par cliquer `START`.



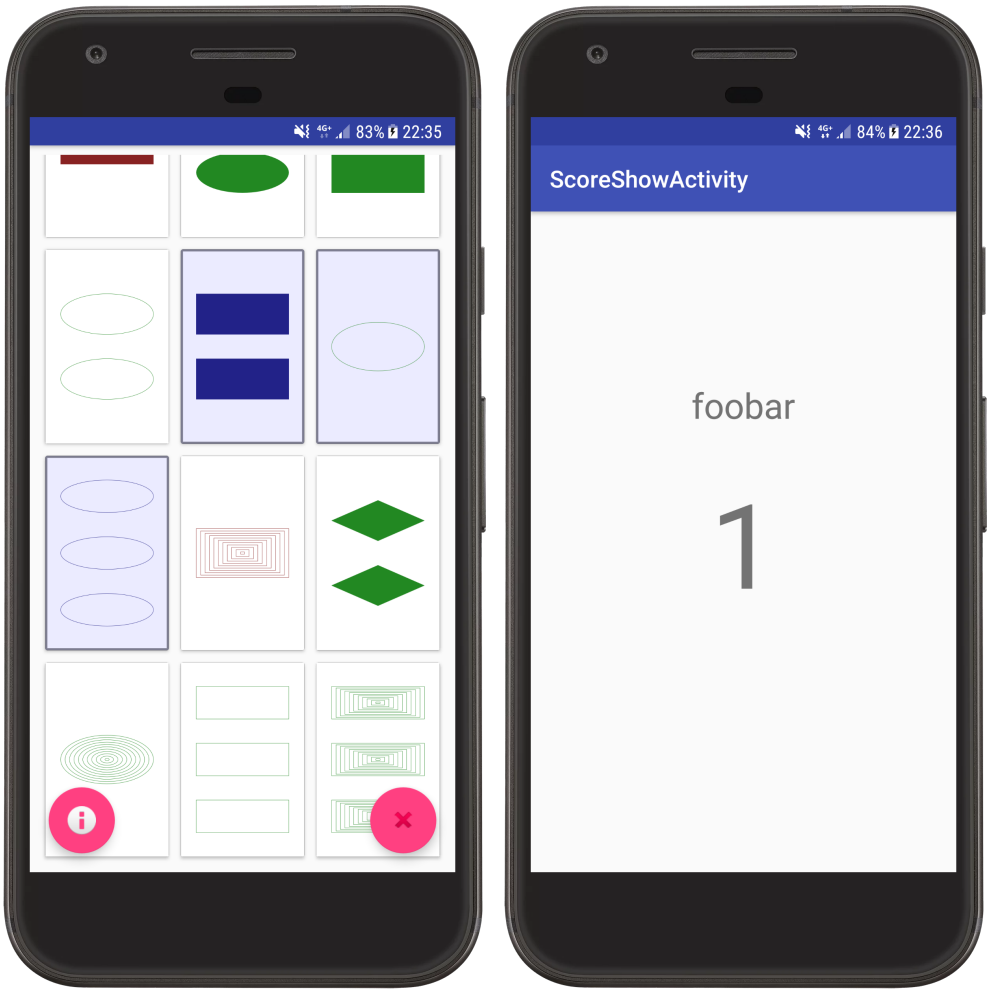
(a) Login Screen (b) Main Game Screen

Dans cette interface graphique, nous avons utilisé un `OnClickListener` pour le bouton `START` pour que l'application réagisse à l'appuie de l'utilisateur.

Puis nous entrons au jeu, initialement, il y a 12 cartes. Nous avons conçu également un algorithme simple pour vérifier s'il existe un set dans toutes les cartes. Sinon, le système ajoutera automatiques 3 cartes. Par hypothèse, il existe toujours un set parmi 15 cartes. Le joueur peut choisir ses cartes par cliquer celles qui sont affichées. Il peut choisir 3 cartes au maximum. Au-delà de trois cartes, le programme l'empêchera à choisir les cartes.

Notammant, nous avons associé un `OnClickListener` pour chaque carte affiché sur l'écran. Suite à tous les changement d'état d'une carte, nous changeons sa couleur du fond. Puis, nous notifions à

l'adapter des cartes affichables les changements. Enfin, l'application raffraîchit l'affichage.



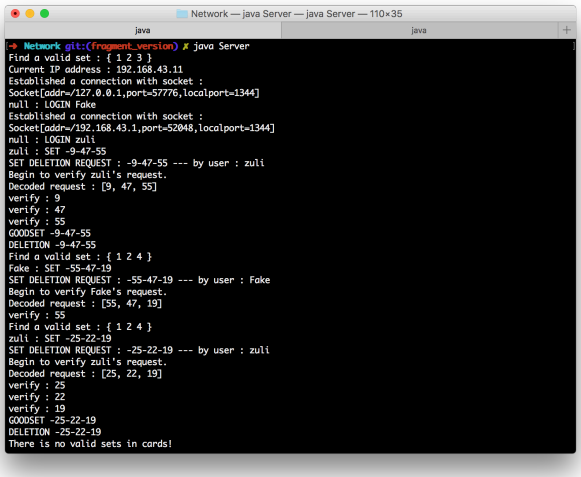
(a) Choisir les cartes (b) Points gagnés

En cliquant le bouton rouge avec une croix, le programme vérifie si les cartes choisies par le joueur sont un set valid. Par taper le bouton rouge avec un point d'exlamation au milieu, nous afficherons les points que ce jour gagne. C'est-à-dire, nous l'associons aussi un `OnClickListener`. Ce listener evoquera l'affichage d'activité qui nous montre les points. Aprés chaque élimination, nous vérifions également s'il existe encore des sets valids.

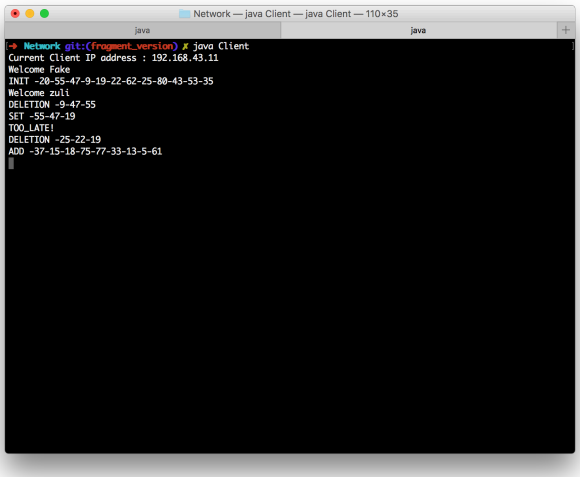
2 Implémentation Multijoueurs

Nous avons choisit de mettre le serveur dans un ordinateur. Le serveur s'occupe d'abord d'initialiser les cartes qui seront distribuées aux joueurs. Puis, nous ouvrons un thread qui accepte la demande envoyée par les terminaux de joueurs. Une fois que nous recevons une demande, nous l'acceptons, et initialisons un thread pour chaque joueur afin de recevoir les messages. Nous avons deux sections critiques dans lesquelles nous ajoutons et retirons les cartes. Nous utilisons `lock()` et `unlock()` pour synchroniser les demandes envoyées par les joueurs.

Le serveur reçoit des commandes de clients. Un client qui reçoit uniquement des messages de serveur.

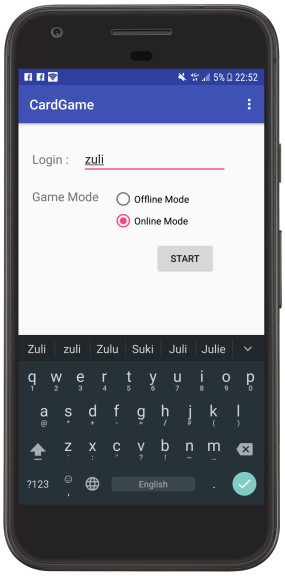


(a) Serveur

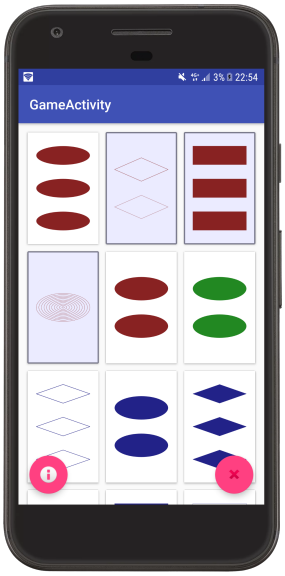


(b) Fake Client

Et les comportements au portable.



(a) Online login



(b) Online click