

Notice Jeu de UNO

Projet Java de 2A – Novembre 2019

Le projet

Ce programme permet de jouer au UNO selon les règles de base, avec 2 à 9 joueurs.

Une partie se déroule en plusieurs manches : une manche est gagnée quand un joueur n'a plus de cartes. Celui-ci gagne alors les points des cartes qui restent dans les mains de ses adversaires. Le critère de fin de partie diffère selon le type de partie que l'on joue : pour plus de détails, lire la partie « Comment jouer ».

Les fichiers sont divisés en trois paquetages : *classes_partagees*, *client* et *serveur*.

Dans *classes_partagees*, sont codées les cartes, les paquets de cartes, les couleurs. On peut ajouter d'autres types de cartes spéciales, par sous-typage : il faut dans ce cas penser à modifier la méthode *nouvelleCarte* dans la classe *Carte*.

Dans *client*, sont codées les classes *Joueur* et *JoueurMain*. Elles permettent d'attendre des messages du serveur, de les afficher à l'utilisateur et de renvoyer son choix. Elles importent le paquetage *classes_partagees*.

Dans *serveur*, on trouve les classes qui modélisent une partie, une manche : ce sont les classes de ce paquetage qui font fonctionner le jeu. La classe *LienAvecClient* permet d'envoyer des messages à chacun des joueurs connectés : elle est construite à partir de la socket du client correspondant. Ces classes importent le paquetage *classes_partagees*, mais sont complètement indépendantes du paquetage *client*.

Comment jouer

Lancer le serveur

Pour lancer une partie, il faut d'abord allumer le serveur de jeu : exécutez la classe *ServeurMain*. Sur la console, il faudra entrer le port sur lequel on veut allumer le serveur (par défaut 6789), puis le nombre de joueurs (entre 2 et 9). Ensuite il faut choisir le type de partie que l'on veut jouer : soit la partie officielle où le gagnant est le joueur qui arrive le premier à 500 points (valeur configurable dans ce programme), soit la partie communément jouée où le nombre de manches est déterminé à l'avance. Le serveur est alors prêt à accepter des connexions de joueurs.

Connecter les clients

Chaque joueur doit exécuter la classe *JoueurMain*. Sur la console, il entre alors son nom, puis l'adresse IP et le port du serveur. Par défaut, l'adresse IP est 127.0.0.1 (en local) et le port est 6789. Ensuite, le joueur choisit s'il souhaite que l'ordinateur trie automatiquement sa main. Quand il y a autant de clients connectés que le nombre de joueurs défini au lancement du serveur, la partie démarre.

Si des clients continuent à se connecter après le début de la partie, ils reçoivent un message d'erreur puis sont déconnectés.

Jouer la partie

Les utilisateurs suivent les instructions affichées sur la console : pour piocher une carte, il faut entrer « piocher ». Pour jouer une carte, il faut entrer par exemple « 5-rouge », « Inverser-bleu » ou encore « Joker ». Si l'utilisateur entre une carte qui n'est pas dans sa main, la console lui redemande d'entrer une carte ; de même si son entrée ne correspondant pas à une carte existante. En revanche l'utilisateur

peut essayer de tricher en posant un « 4-bleu » sur un « 5-rouge », par exemple. Mais mon serveur repère cette triche et lui fera prendre deux cartes de pénalité.

Si l'utilisateur demande à piocher, le serveur lui fait piocher une carte. La console lui demande alors s'il veut la poser (cas de bonne pioche), ici encore l'utilisateur peut essayer de tricher.

Si l'utilisateur demande à jouer un Joker ou un +4, la console lui demande en réponse quelle couleur il choisit (noir n'est pas considéré comme une couleur bien sûr). À noter que cette information s'affichera dans les consoles de tous les joueurs, mais une seule fois : comme dans la vraie vie, il faut être attentif.

À la fin d'une manche, les scores sont affichés sur la console et une autre manche recommence après trois secondes de pause. Si le critère de fin de partie est atteint, la partie s'arrête et le classement final est affiché. Dans ce cas tous les clients sont déconnectés, et le serveur s'éteint.

Extensions réalisées

LANCEMENT DU JEU

On peut jouer **deux types de parties différentes** : la partie officielle et la partie que l'on joue dans la vraie vie.

Si l'utilisateur **fait des erreurs** dans l'adresse IP, le port, ou encore met un espace dans son nom, **le programme le repère** et lui demande de modifier son entrée.

Si l'utilisateur le demande, ses cartes sont automatiquement **triées**.

Les clients retardataires **reçoivent un message d'erreur** et sont ensuite déconnectés.

PARTIE

Une partie se joue en **plusieurs manches**.

On peut jouer avec **toutes les cartes**, y compris le Joker et le +4.

La pioche **se reforme à partir du talon** quand elle est vide. Si **le jeu est bloqué** (toutes les cartes sont dans les mains des joueurs, il n'y a plus de cartes dans la pioche ni dans le talon) et qu'un joueur demande à piocher, le serveur lui fait passer son tour sans lui donner de carte.

Après la pose d'un +4, le joueur suivant peut, pour éviter de piocher 4 cartes, **contester le +4**. En effet, on n'a le droit de poser un +4 que si on n'a pas de cartes de la couleur en cours dans sa main. Si le contestataire a raison, le fraudeur pioche 4 cartes ; si le contestataire a tort, il pioche 4 cartes plus 2 de pénalité.

À chaque fin de tour, les joueurs **connaissent le nombre de cartes** de leurs adversaires. Chaque joueur est tenu au courant de **ce que jouent ses adversaires**, ainsi que des effets des cartes posées.

FIN DE JEU

A la fin de la partie le **classement** est affiché sur la console. Puis les clients sont déconnectés et **le serveur s'éteint**.

Si **le serveur s'éteint** pendant la partie, les clients se déconnectent et affichent un message d'erreur. Si **un client se déconnecte** en cours de jeu, la partie continue sans lui. Il ne peut pas revenir dans la partie. Si **tous les clients sont déconnectés**, le serveur s'éteint.