Algorithmique et Structures de Données
Projet : BLACKJACK

Licence Informatique 2ème année
Université de Nice-Sophia Antipolis

Le but de ce projet est de développer le jeu de carte du Blackjack en Java. Le projet est individuel et doit être rendu avant le 18 décembre 23h59. Tout projet non rendu dans les temps occasionnera une note de 0. Vous devrez créer une archive qui contiendra votre code source ainsi qu'un rapport (au format pdf de préférence) d'au plus 4 pages et l'envoyer par mail à Jean-Charles Régin (`jcregin@gmail.com`) ainsi qu'à votre chargé de TP. Le rapport contiendra une notice d'utilisation de votre jeu et expliquera les choix de développement que vous aurez été amené à faire : représentation des données, fonctions implémentées, etc.

1 Règles

Il existe de nombreuses variantes du Blackjack. Les règles retenues dans le cadre du projet sont exposées ici.

Le jeu se joue avec 6 jeu de 52 cartes et de 1 à 7 joueurs (utilisateurs) contre le croupier (ordinateur). Après avoir reçu deux cartes, le joueur tire des cartes pour s'approcher de la valeur 21 sans la dépasser. Le but du joueur est de battre le croupier en obtenant un total de points supérieur à celui-ci ou en voyant ce dernier dépasser 21. Chaque joueur joue contre le croupier, qui représente la banque. Les cartes de 2 à 10 ont leur valeur nominale, les figures (valet, dame et roi) valent 10 et l'as vaut 1 ou 11 à la convenance du joueur.

Au début de la partie, le croupier mélange les cartes puis défausse face caché les 5 premières cartes du paquet, les cartes restantes forment le sabot. La partie se déroule ensuite en tours de jeu successifs tant que les joueurs décident de continuer. Si à la fin d'un tour le sabot contient moins de 52 cartes, le croupier reprend les cartes du sabot et de la défausse pour procéder à une nouvelle mise en place.

Au début du tour de jeu, chaque joueur mise une certaine somme puis le croupier distribue face visible deux cartes à chaque joueur et une carte pour lui-même. On appelle blackjack un jeu totalisant 21 point sur ces deux premières cartes, ce sera aussi vrai lorsque le croupier tirera sa deuxième carte. Chaque joueur a ensuite, chacun son tour, le choix entre plusieurs options :

- **Tirer une carte (Hit)** Le joueur reçoit une carte face visible du croupier. Si son total dépasse alors 21 il est éliminé du tour et perd sa mise. Il peut renouveler sa demande autant de fois qu'il le désire tant que son score ne dépasse pas 21.
- **Rester (Stand)** Le joueur est satisfait de sa main et en reste là. On passe au joueur suivant.
- **Doubler la mise (Double)** Après avoir reçu ces deux cartes (et seulement à ce moment), un joueur peut doubler sa mise. Le croupier lui distribue une dernière carte face visible et passe au joueur suivant.
- **Partager (Split)** Si les deux cartes qu'un joueur vient de recevoir ont la même valeur, il peut décider de partager ses cartes afin de jouer avec deux jeux. Pour cela, il doit ajouter la même mise que celle qu'il a joué au départ pour son deuxième jeu. Le croupier distribue une nouvelle carte pour compléter chacun des deux jeux qui sont alors traités de façon indépendante. On peut partager un jeu issu d'un partage si la nouvelle carte le permet. On peut doubler sur un jeu partagé. Avoir un jeu de valeur 21 après un split ne compte pas comme un blackjack.
- **S'assurer (Insurance)** Si la carte visible du croupier est un as, le joueur peut s'assurer contre un blackjack du croupier. Le joueur paye alors la moitié de sa mise initiale comme assurance. Si le croupier fait effectivement un blackjack, le joueur récupère son assurance et la banque lui paye la même somme sinon le joueur perd son assurance. Ensuite les gains ou pertes du joueur sont

déterminés normalement.

Une fois que tous les joueurs ont passé, c'est au tour du croupier. Le croupier suit le principe du "pioche à 16, reste à 17" : tant que la valeur de sa main est inférieure ou égale à 16, il pioche, dès qu'elle atteint 17, il s'arrête. Si le croupier dépasse 21, tous les joueurs encore en jeu gagnent. Sinon les joueurs ayant un jeu de valeur supérieur à celui du croupier gagnent et ceux ayant un jeu de valeur inférieur perdent. Le croupier récupère la mise des joueurs ayant perdu, les joueurs ayant gagné récupèrent leur mise et la banque leur paye la même somme, les joueurs à égalité avec le croupier récupère leur mise. Un blackjack est plus fort qu'un 21 obtenu d'une autre manière (en trois cartes ou plus ou après un split). Si un joueur gagne avec un blackjack, la banque lui paye 1,5 fois sa mise au lieu d'une fois.

2 Implémentation

A fin de vous guider dans le développement du jeu, nous donnons quelques consignes que vous devrez prendre en compte pour sa conception. Vous pourrez dans un premier temps vous restreindre à un seul joueur.

2.1 Structure de données

Vous devez définir le type de variable qui sera utilisé pour le développement du jeu. Les principales variables à définir sont les suivantes :

- *Jeu de Cartes :* Vous devez trouver une manière de coder la valeur de chaque carte et de créer un jeu pour les utiliser dans le Blackjack.
- *Main :* Les cartes qui sont jouées par le joueur et par le croupier. Il peut être utile de différencier les deux cas.
- *Joueur :* Elle représente un joueur caractérisé par son budget et la ou les main(s) qu'il a sur la table.
- *Table :* C'est le lieu de jeu. Elle est composée par les joueurs et le croupier.

Ces variables ne sont que des exemples et peuvent varier en fonction de votre façon de concevoir le jeu.

2.2 Fonctions du jeu

Vous devez définir des fonctions qui seront utilisées pendant le développement du jeu. Voici, une petite liste :

- *melanger :* Fonction qui mélange le jeu de cartes.
- *distribuer :* Fonction qui simule l'action de distribuer les cartes au croupier et aux joueurs.
- *mise :* Fonction qui simule la mise du joueur selon son argent.
- *demander :* Fonction qui simule la demande de cartes du joueur.
- *somme :* Fonction qui somme la main du joueur et du croupier. Attention à toujours donner la valeur optimale d'une main si des as sont présents.
- *resultat :* Fonction qui calcule les gagnants et perdants du tour et met à jour leur budget.

Ce sont les principales fonctions que vous devez développer. D'autres fonctions sont à prévoir comme *split*, *assurer*, *doubler* qui sont liées aux fonctions *distribuer* et *demander*.

2.3 Détails pratiques

Nous demandons une interface **textuelle** qui montrera la table, le jeu, la mise des joueurs et le jeu de croupier. Il doit y avoir un menu qui permettra de décider aux joueurs les options de jeu selon sa main (demander une carte, partager le jeu, assurer la mise, etc).

Il existe plusieurs manières d'effectuer le mélange d'un jeu de carte. La plus simple est de sélectionner deux positions au hasard, d'échanger les cartes se trouvant à ces positions et de répéter ce procédé un grand nombre de fois (au moins 1000). Pour obtenir un nombre aléatoire, on peut utiliser la fonction suivante :

```
/**
 * Renvoie un entier aléatoirement choisi entre 0 (compris) et max (non
 * compris).
```

```
 * @param max La borne sur l'entier pouvant être généré.
 * @return Un entier aléatoire.
 */
public static int randomInt(int max) {
    java.util.Random gen = new java.util.Random();
    return gen.nextInt(max);
}
```

Nous donnons les deux fonctions suivantes pour lire l'entrée standard :

```
/**
 * Lit une ligne de l'entrée standard.
 * @return Renvoie la ligne lue ou une chaine vide si un problème est survenu.
 */
public static String readLine() {
    java.util.Scanner scanner = new java.util.Scanner(System.in);
    scanner.useDelimiter("\n");
    String input;
    try {
        input = scanner.next();
    } catch (Exception e) {
        input = "";
    }
    return input;
}

/**
 * Lit une ligne de l'entrée standard.
 * @return Renvoie le premier entier lu ou -1 si un problème est survenu.
 */
public static int readInt() {
    java.util.Scanner scanner = new java.util.Scanner(System.in);
    int input;
    try {
        input = scanner.nextInt();
    } catch (Exception e) {
        input = -1;
    }
    return input;
}
```

3 Evaluation

Le projet sera developpé **individuellement**. L'évaluation sera basée sur les critères suivants :

* Qualité de conception
* Qualité du code (organisation, commentaires inclus, bien indenté, etc)
* Fonctionnalité
* Rapport