

UNIVERSITE DE NICE SOPHIA ANTIPOLIS
DEPARTEMENT INFORMATIQUE
LICENCE 2

Rapport sur l'implémentation du projet Blackjack

Auteur : CONDE Oumar

18/12/2011

Présentation de l'implémentation du jeu blackjack

Présentation des données

Le jeu est constitué des données objets suivants :

1- Une Carte

Elle est caractérisée par un type **String**, qui détermine le nom de la carte. Exemple : As de pique. Une valeur de type **int**, qui représente la valeur de la carte. 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.

En plus des getters et setters, elle a une méthode **toString** pour afficher une carte

2- Une Main

Elle est caractérisée par une liste de cartes qu'on appelle main.

Les méthodes

En plus des getters et setters, il y a les méthodes suivantes :

ajouterCarte : Qui permet d'ajouter une carte à la liste de cartes. Elle prend en paramètre une carte et ne renvoie rien.

somme : Elle totalise le nombre de points des cartes tout en gardant une valeur optimale pour les As. Elle renvoie un entier.

estPartagePossible : Vérifie la possibilité de partager un jeu pour un joueur.

toString : pour afficher une main

3- Un Joueur

Il est caractérisée par :

- Une main de type **Main**, qui détermine la main du joueur
- Un budget de type **double**, qui détermine le budget du joueur pour le jeu
- Un nom de type **String**, qui détermine le nom du joueur
- Une mise de type **double**, qui représente la mise du joueur
- Une variable « estAssurance » de type **boolean**, initialisée à **false**, qui est mise à **true** quand le joueur demande une assurance
- Une variable « estPartager » de type **boolean**, initialisée à **false**, qui est mise à **true** quand le joueur décide de partager son jeu

Les méthodes

En plus des getters et setters, il y a les méthodes suivantes :

ajouterCarte : Permet d'ajouter une carte à la main courante du joueur. Elle prend en paramètre une carte et ne renvoie rien.

estBudgetSuffisant : Vérifie que le budget du joueur est suffisant par rapport à une mise données. Elle prend en paramètre un montant et retourne un **boolean**.

miser : Effectue l'opération de mise pour un joueur tout est prenant en compte la méthode **estBudgetSuffisant**.

toString : Affiche un joueur

4- Une table

Elle représente le terrain de jeu, elle est caractérisée par :

Des joueurs de type **ArrayList<Joueur>**, qui représente les joueurs du jeu

Des cartes de type **ArrayList<Carte>**, qui représente les cartes du jeu, elle représente également le sabot après avoir défaussé des cartes.

Un croupier de type **Joueur**

Une défausse de cartes de type **ArrayList<Carte>**, qui représente la défausse

NB : J'ai choisi des listes à la place des tableaux, car elles sont plus flexibles dans la gestion des données.

Les méthodes

En plus des getters et setters, il ya les méthodes suivantes :

supprimerJoueur : Permet de supprimer un joueur existant. Elle prend en paramètre un joueur et retourne rien.

debarrasser : Permet de débarrasser la table de jeu.

melanger : mélange toutes les cartes du jeu (52 cartes x 6 au début d'une partie)

mise : permet à chaque joueur de miser

distribuer : distribue deux cartes à chaque joueur et une carte au croupier. Elle offre également à chaque joueur de doubler sa mise.

menuChoixHit : méthode statique, elle affiche le menu de choix à un joueur pour faire un hit ou un stand.

faireUnTour : déroule un tour pour une partie. Elle commence par distribuer les cartes ensuite chaque joueur joue à tour de rôle, puis le croupier fait le « pioche à 16, rester à 17 ». Toutes les opérations sont possibles sauf l'action « **partager** » que je n'ai pas eu le temps de finaliser.

pioche16Reste17 : principe du « pioche à 16, reste à 17" : tant que la valeur de sa main est inférieure ou égale à 16, il pioche, des qu'elle atteint 17, il s'arrête.

resultat : calcule les perdants et les gagnant d'un tour.

defausserUneCarte : supprimer la première carte du jeu et l'ajoute dans la defausse.

donnerCarte : donne une carte à un joueur.

creerSabot : défausse les cinq premières cartes du jeu.

validerTour : valide un tour, c'est-à-dire débarrasser la table et vérifie que le sabot contient plus de 52 cartes. Si le sabot contient moins de 52 cartes elle récupère les cartes de la défausse.

randomInt : génère un nombre aléatoire.

creer52Cartes : crée un jeu de 52 cartes (l'usine de fabrication des cartes).

toString : Affiche la table de jeu

5- Blackjack

Contient la méthode main du jeu.

Comment jouer

Pour l'instant le jeu n'est limité qu'à un tour, c'est-à-dire :

On commence par démarrer une nouvelle partie (choix 1 du menu)

Puis par saisir le nombre de joueur

Ensuite chaque joueur saisie sa mise et la partie démarre. Au début de la partie chaque joueur peut décider de doubler sa mise.

Les joueurs jouent à tour de rôle. Au tour d'un joueur, il peut décider de faire des hits, un stand, une Insurance. Quand tous les joueurs ont passé, le croupier applique le principe du « pioche à 16, rester à 17 ».

A la fin du tour on affiche les gagnants et les perdants.