Rapport du projet de Programmation 2 : SnapEIID

Arnold NGNOUBA NDIE TCHEGOU
et
Maryam AIT DAOUD

Nous avons programmé en JAVA une structure de projet basée sur l'architecture modèle-vue-contrôleur pour un jeu de bataille de cartes. Nous décrivons les fonctionnalités ci-dessous :

- **PACKAGE MODELE**
  - *La Class BasicCard.java*
    - Représente les cartes sans effets.
    - La méthode **getDescription( )** renvoie la description complète de la carte sous forme d'une chaine de caractères.
  - *La Class CardEffet.java*
    - Représente les cartes avec effets.
    - Redéfinie la méthode **effetRevele** de la class AbstractCard.java afin d'implémenter les effets des cartes comme suit :
      - **Ant-Man** : Lors de sa révélation, si 3 cartes sont posées au même emplacement que cette carte, elle ajoute +3 de puissance à chacune des cartes.
      - **Medusa** : Lors de sa révélation, si elle est posée à l'emplacement du milieu elle ajoute +2 de puissance à chacune de cartes qui se trouvent au même emplacement.
      - **Star-Lord** : Lors de sa révélation, si l'adversaire a posé une carte au même emplacement que cette carte à ce tour, elle augmente sa valeur de +3.
      - **Punisher** : Lors de sa révélation, elle diminue de -1 la valeur de chacune des cartes sur le même emplacement du côté de l'adversaire.
      - **Blade** : Lors de sa révélation, elle détruit toute les cartes de l'adversaire sur le même emplacement du côté de l'adversaire. !!!Les cartes détruites sont enlevées du plateau et placées dans le Deckdetruit de l'adversaire.
      - **Cable** : Lors de sa révélation, elle pioche une carte de façon aléatoire dans le Deckdetruit en se servant de seed (fournie par ExchangeConnector) et la pose sur le même emplacement.
      - **Deathlock (Carte avec effet continu)** : après sa révélation, il n'est plus possible pour l'adversaire de poser une carte au même emplacement sur son plateau.
    - Redéfinie la méthode **reveal** de class AbstractCard.java afin d'appliquer les effets des cartes avec effets. Pour les cartes avec effets révélé, on utilise un booléen (**reveal_ettect**) pour s'assurer que l'effet de la carte ne s'applique qu'une seule fois et non à chaque tour de la partie.
  - *La Class Game.java*
    - Afin d'implémenter les effets mentionnés ci-dessus on utilise les trois de booléens tableaux suivants :
      - **marker** : tableau de booléen permettant de savoir si une carte a été ajoutée à un emplacement du plateau. La case du tableau correspondant à l'indice d'un emplacement (le Board) est marquée true si une carte a été ajoutée à celle-ci. Ce tableau nous sert à implémenter l'effet de la carte **Star-lord**.
      - **marker_destroy** : tableau de booléen permettant de savoir sur quel emplacement l'adversaire a détruit des cartes. La case du tableau correspondant à l'indice d'un

emplacement (le Board) est marquée true si ses cartes ont été détruites. Les deux joueurs doivent ensuite s'échanger le tableau marker_destroy pour mettre à jour leur affichage de UIgrapique.

- **verou** : tableau de booléen permettant de savoir si un emplacement est verrouillé par la carte à effet continu Deadlock. La case du tableau correspondant à l'indice d'un emplacement (le Board) est marquée true si la carte Deadlock a été ajoutée à celle-ci. Les deux joueurs doivent ensuite s'échanger le tableau verou pour afin d'implémenter l'effet de Deadlock.

- Dans la méthode **tryPlay( )**, nous avons pu implémenter les contraintes du nombre maximum de cartes égale à 4 et du cout de la carte valide. Nous avons rajouté à ces deux contraintes, celle sur le verou (le verou doit être false pour qu'une carte soit placée sur un emplacement).
- Dans la méthode tryPlay ( ) nous avons également ajouté l'effet de la carte **Unlock** qui consiste à désactiver le verou à un emplacement si ce dernier avait été activé par la carte Deadlock, et permet ainsi au joueur de poser des cartes sur cet emplacement.
- La méthode **getWinner ( ),** nous renvoie exactement le gagnant et le perdant au bout d'une partie. En cas d'agalité (les deux joueurs ont le même score sur les emplacements et le même score total) elle affiche égalité pour les deux joueurs.

➢ PACKAGE CONTROLEUR

- *La Class Controller.java*
  - La méthode **setupGame( )** initie connexion entre les deux joueurs.
  - La méthode **startGame( )** affecte le Deck de chaque joueur à la LinkedList correspondant de chaque côté avant le début de la partie.
  - Dans la méthode **endTurn ( )** nous avons implémenté :
    - L'échange des LinkedList correspondant aux coups_valides pour un tour de jeu, entre les deux joueurs.
    - Nous avons ajouté les propositions des coups des joueurs sur le plateau de chaque côté et nous avons appliqué les effets des cartes avec Effets, avant de mettre à jour l'affichage et passer au prochain tour.
  - La méthode **endGame ( )** met fin à la partie au bout de 6 tours et affiche le vainqueur (ou égalité en cas d'égalité).

➢ PACKAGE VUE

- Dans la classe *CardPanel2.java*, nous avons ajouté une image à chacune de nos cartes en se servant de la méthode **paintComponent**.

➢ DIFFICULTES

- Notre principale difficulté a été l'amélioration de l'interface graphique, notamment :
  - Nous avons essayé sans succès d'ajouter une JList à l'interface graphique afin de permettre aux joueurs de constituer leurs Deck.
  - Nous ne savions pas exactement dans quelle classe ceci devrait être ajoutée et comment le faire.