

Jeu des pirates

Pour la classe GetPower :

Méthode privée creerPouvoir : Cette méthode crée une instance de Power en fonction du pouvoir passé en paramètre. Elle est utilisée pour créer les différents pouvoirs disponibles dans le jeu.

Méthode action :

- Cette méthode définit l'action à effectuer lorsqu'un joueur atterrit sur la case de pouvoir.
- Si le joueur possède le pouvoir « Chuunins », il essaie d'obtenir un nouveau pouvoir de manière aléatoire (en évitant les doublons), et l'applique au joueur. Si le joueur échoue à obtenir un nouveau pouvoir, un message est affiché.
- Sinon, le joueur reçoit 5 points de vie supplémentaires s'il possède déjà un pouvoir.
- Ensuite, un message est affiché pour informer le joueur de ce qui s'est passé, et les statistiques sont éventuellement mises à jour en fonction du pouvoir obtenu.
- Enfin, un récapitulatif est affiché, montrant les détails du joueur et de l'autre joueur.

Méthode privée getPowerName : Cette méthode retourne le nom du pouvoir en fonction de l'index passé en paramètre.

Pour la classe CardBox :

Méthode action :

- Cette méthode définit l'action à effectuer lorsqu'un joueur atterrit sur la case de carte.
- Un message est affiché pour informer le joueur qu'il est sur une case carte.
- Des cartes sont tirées du paquet de cartes ('deck') et affichées à l'écran.
- Le joueur est invité à choisir une carte parmi celles tirées, et l'effet de la carte choisie est appliqué sur le joueur.
- Un récapitulatif est affiché, montrant les détails du joueur et de l'autre joueur.

Pour la classe Deck :

Méthode privée initializeDeck : Cette méthode ajoute différentes cartes avec des effets différents à la liste 'cards', initiant ainsi le paquet de cartes.

Méthode privée putBackCards : Cette méthode remet toutes les cartes de 'cards' dans la pile de cartes 'cardsPile'.

Méthode tirageCards :

- Cette méthode tire trois cartes aléatoirement de la pile de cartes 'cardsPile' et les ajoute à une nouvelle liste 'drawnCards'.
- Elle retire également les cartes tirées de la pile et réinitialise la pile en appelant 'putBackCards()'.
- Elle renvoie ensuite la liste 'drawnCards'.

Méthode applyCardEffect :

- Cette méthode applique l'effet de la carte choisie sur le joueur ('Pion') passé en paramètre.
 - Elle utilise une instruction `switch` pour déterminer l'effet de chaque carte et appliquer les modifications appropriées au joueur.

Et pour finir la case Versus :

Méthode action :

- Cette méthode définit l'action à effectuer lorsqu'un joueur atterrit sur la case de combat.
- Elle récupère l'autre joueur à combattre, affiche un message indiquant le déclenchement du combat, affiche un récapitulatif des joueurs, puis appelle la méthode `combat` pour simuler le combat.
- Si le joueur adverse a ses points de vie réduits à zéro ou moins, le joueur actuel est déclaré vainqueur.
- Sinon, le combat est répété avec le joueur adverse attaquant à son tour.
- Un autre récapitulatif est affiché après chaque combat.

Méthode combat :

- Cette méthode simule le combat entre deux joueurs.
- Elle affiche les actions disponibles pour le joueur en cours (attaque et les capacités spéciales) et lui permet de choisir une action à effectuer.
- En fonction du choix du joueur, les dégâts sont infligés à l'adversaire ou les capacités spéciales sont utilisées.
- Si le joueur n'a pas suffisamment de chakra pour utiliser une capacité spéciale, un message d'avertissement est affiché et le joueur doit choisir une autre action.
- Le combat se termine lorsque l'un des joueurs n'a plus de points de vie.

Diagramme de Séquence pour CardBox

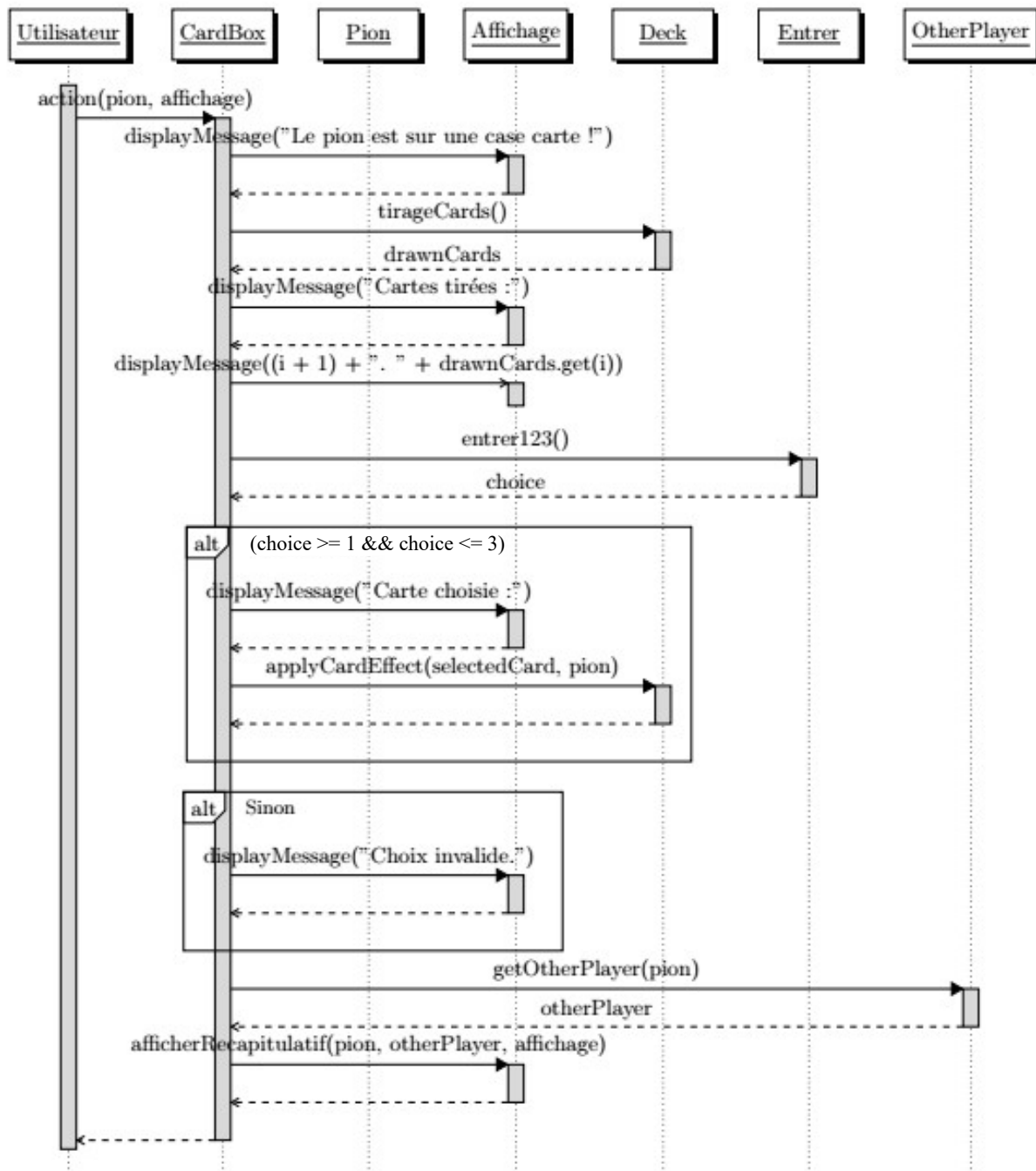
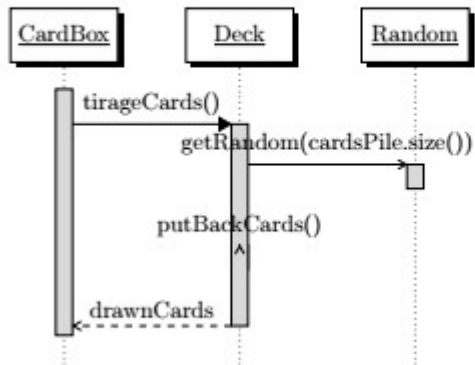


Diagramme de Séquence pour Deck

Tirage des Cartes



Application de l'Effet de la Carte

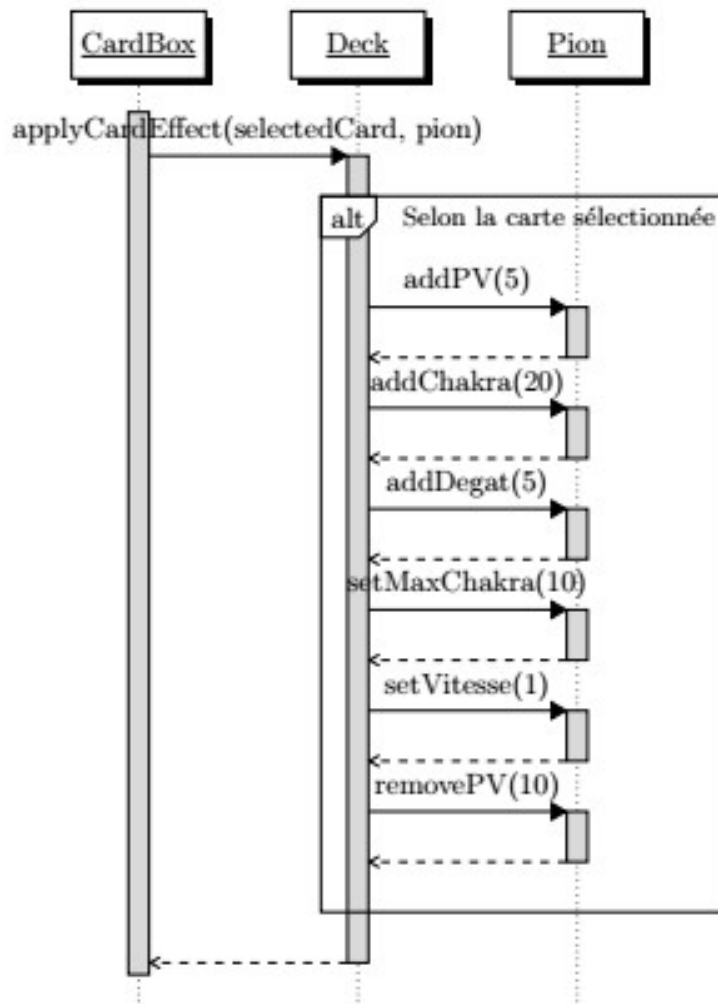


Diagramme de Séquence pour GetPower

