

RAPPORT DE PROGRAMMATION, BONUS

1. Interface de lancement de la partie
2. Création d'un joueur jouant très mal
3. Implémentation des annonces dans une partie de jass.
4. Amélioration du *MctsPlayer*, prenant en compte les annonces et les règles du jass.
5. Implémentation de quelques raccourcis de triches.

1. Pour utiliser l'interface, lancer la méthode *main* de la classe *Launcher* du package *gui*.
 - a. S'affichera d'abord une première interface, vous permettant de choisir si vous voulez lancer une partie *locale*, ou *distante*. Si vous sélectionnez *distante*, tout se passe comme si vous aviez lancé *RemoteMain*. Si vous sélectionnez *locale*, s'ouvre l'interface de lancement d'une partie locale.

Dans cette interface, vous choisissez le type des joueurs à l'aide de boutons, notamment le niveau des IA, puis vous remplissez les paramètres optionnels à la main -des *TextField*, qui empêchent de rentrer des valeurs interdites à l'aide de *Listener*. Cela fait, vous pouvez maintenant cliquer sur le bouton *démarrer* (ce que vous pouviez faire auparavant, mais cela ne lançait pas de partie).

2. Le joueur mauvais -*WorstMctsPlayer*- s'inspire grandement (totalement) du *MctsPlayer*, mais joue simplement la pire carte à laquelle il pense, plutôt que la meilleure.
3. Ce système d'annonce respecte les règles du jass. Son implémentation utilise l'exercice « Annonce au Jass » mais complète la classe *MeldSet* : attribut public statique *EMPTY_SET*, méthode *cards()*, méthode *allIn* prenant un *CardSet*, et méthode *packed()* et *from(packed)* qui permettent d'échanger à distance. Notons l'imprévu suivant : *MeldSet* est le set de 78 annonces donc il m'a [Antoine] fallu 2 long pour emballer un *MeldSet*. Etant donné que le panneau pour le choix des annonces n'apparaît qu'une fois par tour, lors de l'appel de la méthode *announcement*, le panneau est créé à cet appel (conseil d'un assistant), puis supprimé (via un *BooleanBinding* du constructeur de *GraphicalPlayer*). La mise à jour des annonces dans *GraphicalPlayer* est faite de manière identique à *updateTrick*.

Les annonces sont appelées, récoltées dans *JassGame*.

NB : lorsque l'on choisit une annonce, il faut cliquer sur le texte de cette annonce, pas juste dans la case correspondante.

4. L'amélioration du *MctsPlayer* : lors de la simulation de la partie aléatoire (*simulateToEndOfTurn*) le choix des cartes pour chaque joueur (*playableCards* auparavant) est restreint si possible par a) et b), ou seulement par a), ou seulement par b) dans cet ordre.
 - a. L'ensemble des cartes sauf les cartes qui apparaissent dans les annonces des autres joueurs.

- b. L'ensemble des cartes sauf les cartes que le joueur ne peut pas avoir d'après les règles du jass (voir wikipedia ou la classe `JassReducerOfSet`).

En effet nous obtenons les résultats suivants en opposant 2 *MctsPlayerSmart* à 2 *MctsPlayer*.

<i>MctsPlayerSmart</i> VS <i>MctsPlayer</i>	100 itérations, 1000 parties Prise en compte des annonces seulement (a)	100 itérations 1000 parties Prise en compte des règles seulement (b)	100 itérations 1000 parties Prise en compte de (a) et (b)	100_000 itérations 100 parties Prise en compte de (a) et (b)
Points d'écart/partie	+30	+25	+44	+60
Taux de victoire <i>MctsPlayerSmart</i>	55%	54%	57%	60%

5. L'Implémentation des raccourcis de triches est similaire à celle des annonces et a les fonctionnalités suivantes : à n'importe quel moment du jeu, si un joueur appui sur une touche spécifique du clavier :

- « 1 » vous donnera la meilleure main possible au prochain tour, à savoir celle avec les 9 atouts.
- « 2 » vous rajoute 100 points à la fin de ce pli.
- « 8 » vous fera commencer au prochain tour.
- « 9 » vous fait gagner à la fin de ce pli.

Nb : il ne peut y avoir qu'un raccourci de triche en cours par joueur en même temps. Et il est possible d'écraser le précédent raccourci de triche mémorisé à chaque nouveau pli.