Rapport

(Détaillant les structures de données et algorithmes utilisés, la structure du programme, les choix techniques effectués)

Structure du programme :

Nous avons un programme principal qui permet de lancer les parties avec différent paramètre (taille du plateau et IA). Les fichiers correspondants à l'utilisation du plateau se trouve dans le répertoire src. Pour le menu les fichiers se trouvent dans le répertoire IHM.

Menu principal :

Pour la création des boutons on récupère les coordonnées du clique et on vérifie si ce clique se trouve dans les coordonnées des boutons, si oui on lance le programme adapté.

Plateau :

Nous avons décidé de partir sur un système de tableau pour stocker nos données. Les coordonnées des points placés sont récupérées pour être ajouté dans le tableau. Par exemple pour un point de coordonnée (345 ;210), on divise par 40 ce qui nous donne (8,6 ; 5,3) ont arrondie ensuite pour récupérer des coordonnées qu'on va mettre dans notre tableau et ainsi pouvoir les placer correctement sur notre plateau.

Pour savoir à quel camp correspond le point placé on utilise comme valeur dans notre tableau : 0 pour aucun pion, 1 pour un point noir et 2 pour un pion blanc.

On check ensuite que les coordonnées sont bien dans la zone du plateau. Si le clique est hors zone, l'utilisateur doit recliquer. Nous utilisons une fonction qui check si l'emplacement est libre pour pouvoir poser notre pion. Elle nous retourne 1 si l'emplacement est disponible, elle nous retourne 0 si l'emplacement n'est pas disponible. Si le point est dans la zone et est disponible on ajoute au tableau ses coordonnées. On vérifie quel joueur doit jouer avec notre variable « tourJoueur » si elle est à 1 le noir joue si elle est à 2 le blanc joue. On choisit donc la couleur du pion à placer et on affiche le point sur le plateau.

Notre idée pour réaliser les règles du jeu de Go était de vérifier les libertés disponibles autour des pions ennemis. Nous avons une fonction qui vérifie les libertés disponibles autour du pion. Elle check le nord, sud, est, ouest. S'il n'y a aucun pion autour elle nous retourne 1 sinon 0.

Pour transformer un pion en une autre couleur, on récupère les coordonnées du point à modifier et on change la couleur en fonction de la valeur du tableau.

Arnaud LAPENNA & Julien GROLL

Pour les coups suicides nous vérifions avant de poser un pion qu'il n'a aucune liberté et qu'il est encerclé par les ennemis. Si c'est le cas nous l'empêchons de poser un pion.

Pour l'I.A nous simulons des coordonnées avec un random de x et de y. Et nous appliquons le même traitement que pour le traitement des pions. On vérifie que les coordonnées sont dans la zone (normalement le random ne dépasse pas 13) et que les coordonnées pointent sur un emplacement libre.

Pour le compteur de point, nous parcourons le tableau à chaque clique et nous incrémentons deux variables (noir et blanc) à chaque fois que nous rencontrons un nombre de la couleur (1=noir, 2=blanc). Pour l'afficher dynamiquement (à chaque nouveau pion) nous devons redraw la fenêtre pour la mettre à jour. On parcourt à nouveau le tableau pour réafficher tous les points placés précédemment.