

UNIVERSITÉ LAVAL

Introduction à la programmation

IFT-1004



Travail pratique #4
Un jeu de dames avec interface graphique
(Rapport écrit)

Travail remis à :

M. Pascal GERMAIN

Travail effectué par :

Ariane Fiset (111 236 136)

Pascal de Le Rue (111 187 065)

Session été 2020

Remis le 26 juillet 2020

4.1 Fonctionnalités obligatoires

Lors de la première partie du travail pratique, nous avons fait plusieurs choix et modification afin d'obtenir un jeu fonctionnel en tout temps. De plus, ce jeu comporte des fonctionnalités qui facilitent son utilisation, tel que des boutons ou des zones de textes. Les fonctionnalités du code ont été faites de la façon suivante :

(1) Afin que les pièces puissent se déplacer sur le damier, nous avons tout d'abord initialisé trois variables, soit un booléen afin de savoir si une pièce est sélectionnée ou non, une variable de pièce sélectionnée et une de position sélectionnée. Nous avons ensuite trouvé le numéro de ligne et de colonne associé à cette position pour ensuite déplacer la pièce à la position sélectionnée et finalement actualisé l'affichage du damier. (2) Afin de bloquer les déplacements invalides, nous avons tout simplement fait appel à la classe *Partie* afin de pouvoir réutiliser les 'règlement' déjà codés. (3) Encore une fois, nous avons utilisé la classe *Partie* afin que le joueur puisse choisir une nouvelle pièce source après un mouvement invalide. (4, 5) Afin qu'une prise soit obligatoirement faite si elle est possible et que les prises multiples soient gérées correctement, nous avons utilisé les fonctions *tests* disponibles dans la classe *Damier* du fichier *damier* afin d'obliger le joueur à faire une prise lorsque nécessaire. (6) L'alternance du tour des joueurs a été réalisée en réutilisant la méthode *tour* de la classe *Partie*, celle-ci gérant l'alternance des joueurs. (7) Un bouton *Nouvelle partie* a été ajouté à l'interface à l'aide de la classe *Button* de *tkinter*. Lorsqu'un utilisateur appuie sur ce bouton, la fonction *nouvelle_partie* est lancée et la classe *FenetrePartie* est supprimé et ensuite initialiser afin de redémarrer une nouvelle partie. (8) Un bouton *Quitter* est aussi ajouté à l'interface afin de permettre à l'utilisateur de Quitter la partie. Ce bouton ne fait qu'effectuer la commande *quit* ce qui revient à la même chose qu'un clic sur le X de la page. (9) Une rétroaction est donnée à l'utilisateur en tout temps. Cette fonctionnalité est effectuée, car tous les déplacements découlent de la classe *Partie* qui effectuait déjà ce type de messages. Ces messages sont donc maintenant transmis à l'interface graphique du jeu à l'aide du module *Label*. De plus, nous avons ajouté un message lorsqu'un des joueurs gagne la partie. (10) Finalement, l'affichage de la couleur du joueur courant est réalisé à l'aide de l'ajout d'une étiquette *Label* en se basant sur les données de la fonction *joueur_courant*.

L'attribut *self.couple_de_position* a été créé dans la classe *Partie* du fichier *partie* afin de modifier cette valeur à partir de la classe *FenetrePartie* du fichier *interface_dames*. Nous avons ensuite modifié la méthode *tour* de la classe *Partie* en remplaçant l'appel de la méthode *demande_positions_deplacement* par une simple attribution de *self.couple_de_position* au tuple *position_source, position_cible* (variables locales de la méthode *tour*). Le bloc de code servant à l'affichage du damier dans l'onglet run de l'interface *PyCharm* a été déplacé à la fin de la méthode *tour* afin de garder la cohérence entre la console et l'affichage graphique. À la fin de la méthode *tour* nous avons également ajouté une ligne afin de réinitialiser l'attribut *self.position_source_selectionnee* de la classe *Partie*. Dans la méthode *deplacer* de la classe *damier*, la ligne à laquelle un pion devient une reine a été modifiée. Initialement, cette ligne a été fixée à 7 tandis que maintenant elle n'est plus fixe et se modifie en conséquence du dimensionnement du damier.

De plus, afin de nous permettre un appui visuel lors de notre programmation, nous avons pris le choix de garder l’affichage que retournait initialement le code dans l’onglet *run* de l’interface *PyCharm*. Cet appui nous permettait de comparer ce code qui était initialement fonctionnel en rapport avec l’affichage graphique.

4.2 Fonctionnalités optionnelles

Dans ce travail pratique, nous avons choisi les fonctionnalités ainsi que la façon de les résoudre suivante :

(1) Tout d’abord, nous avons ajouté l’option de lire les règlements. Cette option impliquait l’ajout d’un nouveau bouton *Règlements* qui effectue l’ouverture d’une nouvelle fenêtre listant tous les règlements du jeu. La création de cette nouvelle fenêtre a été exactement le même principe que l’ouverture de la partie. (5) Par la suite, nous avons ajouté l’option de revoir tous les déplacements qui ont été effectués précédemment dans la partie. Cette fonctionnalité ouvre une nouvelle fenêtre (de la même façon que *Règlements*) pour ensuite lister tous les déplacements faits précédemment. Cette liste est une chaîne de caractères qui est mise à jour à chaque déplacement. (9) La génération d’un damier aléatoire se fait avant le début de la partie. Une fenêtre apparaît et dans celle-ci il y a un bouton *Générer un damier aléatoire*. Ce bouton fait en sorte qu’un nombre de colonnes, un nombre de lignes ainsi que les emplacements de pions et reines sont positionnés aléatoirement dans le damier. (14) Un mode tricherie est aussi ajouté au code. Ce mode permet aux joueurs de rendre ses pions en dame, que ce soit tous pour les deux couleurs ou pour seulement que l’une d’entre elles. Un bouton *Tricherie* a été ajouté à la fenêtre principale et lorsque le joueur appuie sur celui-ci, une autre fenêtre s’ouvre et offre à l’utilisateur les trois options de tricherie. (16) Afin de permettre à l’utilisateur de choisir le dimensionnement de damier voulu, dans la même fenêtre que *Générer un damier aléatoire* deux bandes déroulantes s’affichant demandant à l’utilisateur le nombre de colonnes et de lignes que celui-ci souhaite avoir comme damier. Cette fonctionnalité est possible en changeant, certainement, la dimension de damier (qui est toujours par défaut 8 x 8) ainsi qu’en repositionnant correctement les pions et les dames dans le damier. Lorsqu’on utilise la fonctionnalité *Générer un damier aléatoire* cette fonctionnalité de redimensionnement est aussi utilisée. (18, 19) Finalement, l’interface de jeu de dames permet de sauvegarder et de reprendre une partie sauvegarder à l’aide des boutons *Sauvegarder* et *Charger* qui se situent dans la fenêtre principale du jeu. Lorsqu’un utilisateur appuie sur le bouton *Sauvegarder*, un fichier texte est enregistré dans le répertoire courant. Ce fichier texte inclut le positionnement des pièces ainsi que le type de la pièce. Ensuite, si un utilisateur appuie sur le bouton *Charger*, l’ordinateur réutilise le fichier texte créé précédemment afin de créer un dictionnaire qui peut ensuite être utilisé par le reste du code.