Sarah Lemieux-Montminy 111154516

Gabriel Morin 536797778

**RAPPORT TP4**

Le présent rapport justifiera les choix de modélisation du travail pratique 4. Il abordera d’abord l’organisation générale des différentes fenêtres, ensuite le fonctionnement de l’interface du jeu et finalement, l’implémentation des fonctionnalités optionnelles.

Tout d’abord, les différentes fenêtres sont gérées à partir de la fenêtre de base Tkinter créée avec la classe JeuDames. La fenêtre programmée par la classe FenetreDemarrer est la première à être affichée à l’utilisateur et elle est suivie par celle de la classe FenetreConfiguration une fois qu’un clic droit a été effectué sur le bouton démarrer. Ce dernier interface permet à l’utilisateur de choisir les réglages du jeu qui constituent des fonctionnalités optionnelles dont il sera question plus loin. Cette fenêtre a été créée principalement par soucis de clarté et de cohérence. En effet, elle permet de rendre plus agréable visuellement la fenêtre du jeu de dame puisqu’elle est ainsi libérée de plusieurs boutons et l’utilisateur peut dès le départ choisir les options qu’il préfère. Une fois qu’un clic droit sur le bouton « jouer » a été réalisé, la fenêtre programmée par la classe FenetrePartie apparaît avec l’interface de jeu en soi. De cette fenêtre, il est toujours possible de retourner à la fenêtre programmée par la classe FenetreConfiguration pour réajuster les réglages grâce à un bouton dédié à cet usage. Les différentes fenêtres sont rappelées à l’aide du « controller », qui permet notamment de réutiliser les variables d’une fenêtre à l’autre. Bien que le choix de cette technique n’est certainement pas optimal, c’est celui qui a été retenu pour palier à l’organisation de départ choisie des interfaces.

Ensuite, tel que mentionné précédemment, l’interface du jeu de dame est celui affiché par la classe FenetrePartie. Les pièces sont déplacées grâce à un clic sur la position source, puis avec un clic sur la position cible. Des variables membres sont créées pour garder les positions sélectionnées et les valider avec les méthodes valider\_position\_source() et valider\_position\_cible() qui ont été légèrement modifiées depuis le TP3 pour répondre aux besoins du jeu. Une fois les positions vérifiées, le déplacement est effectué grâce à la méthode tour(). Tous les messages indiquant à l’utilisateur les actions à suivre et les erreurs commises, s’il y a lieu, sont affichées dans un frame situé à droite du damier. Ces messages sont gérés pour la plupart dans la méthode selectionner(). Il existe aussi une colonne de boutons comportant un « Nouvelle partie » sous ce frame. Il réinitialise simplement toutes les variables membres et le damier de façon à retrouver le jeu de départ. C’est dans cette même colonne de boutons que plusieurs permettent la réalisation de différentes fonctionnalités optionnelles.

En effet, plusieurs options sont proposées à cet endroit. La première est celle d’afficher les positions sources possibles, qui permet de mettre en orange les cases des pièces pouvant être déplacées. Plusieurs méthodes ont été utilisées dans la classe FenetrePartie pour y arriver, mais l’essentiel du code de la création des cases de couleurs se situe dans le fichier canvas\_damier, plus précisément dans les méthodes afficher\_positions\_sources\_blanc() et afficher\_positions\_sources\_noir(). Par ailleurs, la méthode fournie dessiner\_cases() a aussi été modifiée pour faire fonctionner le tout. Ainsi, un clic droit sur le bouton correspondant permet de colorer les cases des positions sources possibles et un double-clic annule la commande. Cette option semblait importante pour aider un joueur en difficulté, mais tout de même lui permettre de revenir à l’état initial si l’option lui déplaisait après un moment. En outre, l’option de rendre les cases des positions cibles en vert a été ajoutée dans l’optique de rendre le damier plus convivial. Elle fonctionne d’ailleurs d’une façon presque identique à la précédente.

Sous le bouton « Afficher les positions source possibles » se trouve celui d’annuler le dernier mouvement, qui permet de faire exactement ce qu’il indique. Il a été créé grâce à une méthode du même nom dans la classe FenetrePartie() qui garde les dernières positions utilisées dans des variables membres. De par ce procédé, cette option ne peut être utilisée qu’une seule fois par tour, ce qui répondait aux besoins du jeu.

Ensuite, le bouton « Afficher l’historique » crée un fichier text des mouvements exécutés tout au long de la partie. Il peut par ailleurs être sauvegardé. Finalement, les statistiques du jeu sont affichées à chaque tour grâce à différentes variables membres.

En ce qui a trait aux options de la fenêtre de réglages, la fonctionnalité du choix de la couleur du thème a été implémentée grâce à des variables membres dans la classe FBase qui sont utilisées ensuite dans la création des fenêtres. Il en va de même pour les noms des joueurs; ils peuvent être changés par l’utilisateur grâce aux Entry et sont ensuite introduits dans des variables membres qui sont réutilisées dans l’interface de jeu. Finalement, un bouton « Règlements » permet d’afficher les règles du jeu dans une nouvelle fenêtre.

Ainsi, l’utilisation de plusieurs fenêtres et l’implémentation de neuf fonctionnalités optionnelles permettent de rendre l’expérience du jeu de dames plus agréable à l’utilisateur.