

Travail pratique # 4

L'interface du jeu de dames

Jean-Francis Roy

Date de remise : Mardi 22 avril 2014 à 23h00
Pondération de la note finale : 12%

1 Objectifs

Ce travail a comme principaux objectifs de vous familiariser avec les interfaces graphiques (bibliothèque tkinter fournie avec Python), avec la gestion des exceptions et avec la réutilisation/amélioration/maintenance de code existant dont vous n'êtes pas l'auteur.

2 Organisation du travail en équipe

Nous privilégions le travail en équipe car il fait partie de votre formation académique et vous permettra de partager la charge du travail. Ceci dit, si certains veulent faire le TP individuellement, libre à eux de le faire. Aussi, pour ceux qui n'arrivent pas à trouver un coéquipier, nous les invitons à utiliser le forum du cours.

Chaque coéquipier doit contribuer à part égale au développement de ce TP. Laisser son coéquipier faire tout le travail (peu importe les raisons) est inacceptable : vous passerez à côté des objectifs de ce cours. De la même manière, il ne faut pas non plus trop en faire : il faut apprendre à travailler en équipe !

3 Problème à résoudre

Vous devez réaliser une interface graphique sur un jeu de dames similaire à celui que vous avez développé au TP#3. Cette fois-ci, nous ne vous fournissons pas la modélisation du problème : à vous de créer votre propre solution.

3.1 Le code fourni

En pièce jointe à ce TP est fourni un jeu de dames générique sans interface (ni graphique, ni en console). Ce code est similaire à ce que vous deviez rendre au TP#3, mais n'en est pas un corrigé. Notamment, ce code diffère de votre TP#3 aux éléments suivants :

- La classe Piece
 - Une pièce possède maintenant un attribut `nom` qui est unique pour chaque pièce. Cet attribut est pratique dans l'interface graphique : on peut l'utiliser comme *tag* lorsqu'on ajoute une pièce dans un canvas, pour pouvoir récupérer une pièce par la suite, notamment pour la supprimer du jeu.
 - Le nom d'une pièce contient deux caractères et un numéro. Le premier caractère est P (pion) ou D (dame), et le second est B (blanc) ou N (noir). Le numéro est attribué automatiquement lorsqu'on crée une nouvelle pièce.
- La classe Damier
 - La méthode `deplacer` lance maintenant une exception si le déplacement est invalide. Cette méthode retourne maintenant `True` si une pièce a été prise lors du déplacement, et `False` autrement.
 - La méthode de chargement à partir d'une chaîne de caractères peut maintenant lancer une exception en cas d'erreur.
 - Deux nouvelles méthodes : `position_peut_prendre_une_piece_adverse` et `joueur_peut_prendre_une_piece_adverse`, qui seront bien pratiques !
- La classe Partie
 - Nous avons retiré toutes les méthodes reliées à l'interface en console.
 - Nous avons retiré les méthodes `tour` et `jouer`.
 - Les méthodes `valider_position_source` et `valider_position_cible` ne retournent rien si la position est valide, et lancent une exception autrement. On les utilisera donc dans un bloc `try` dans l'interface, et on gèrera les erreurs au besoin dans un bloc `except` !
 - Les méthodes de chargement et de sauvegarde peuvent lancer des exceptions, et vous avez maintenant accès à des méthodes pour lire une partie à partir d'une chaîne de caractères, ou sauvegarder dans une chaîne de caractères.
 - Deux nouvelles méthodes : `passer_au_joueur_suivant` et `joueur_courant_peut_prendre_piece_adverse`.
- De nouveaux types d'exception sont définis et utilisés dans le code. À vous bien les gérer dans votre interface !

Nous vous fournissons également le corrigé du TD#10 dans le dossier `interface_dames`. Réutilisez-le, mais vous vous rendrez compte qu'il faut le modifier ! Par exemple, avez-vous besoin du dictionnaire de pièces de ce laboratoire ? Probablement pas, puisque votre interface graphique inclura aussi une instance de la classe `Partie`, qui elle inclut un `Damier`, qui lui inclut un dictionnaire de pièces... **Prenez le temps de bien y penser avant de commencer à programmer.**

Nous vous fournissons finalement un fichier `tp4.py` qui sera le point d'entrée du programme.

4 L'interface graphique à programmer

En réutilisant et en améliorant le code fourni, construisez une interface complète ayant les éléments suivants :

- Un damier permettant de jouer une partie de dames. On clique sur une pièce et ensuite sur une position cible pour faire un déplacement ;
- Un élément visuel permettant de savoir à qui est le tour et les autres informations pertinentes ;
- Une liste des coups joués ;
- Des boutons permettant de :
 - Charger une partie (en ignorant la liste des coups joués) ;
 - Sauvegarder une partie (en ignorant la liste des coups joués) ;
 - Charger un fichier de partie avec liste de coups joués ;
 - Sauvegarder un fichier de partie avec liste de coups joués ;
 - Démarrer une nouvelle partie ;
 - Quitter le jeu.
- Une zone de texte indiquant les éventuelles erreurs.

Notez que les méthodes déjà fournies permettent entre autre de valider les positions sources et cibles, de faire les déplacements, de charger et sauvegarder une partie, etc. À vous de voir ce que vous pouvez réutiliser !

Chargement et sauvegarde avec liste de coups joués. Un fichier de sauvegarde d'une partie avec les coups joués doit contenir comme première partie la sauvegarde de la partie en cours. Vous avez déjà accès à une méthode pour le faire. Le fichier contient ensuite une ligne permettant de délimiter le reste du fichier. **Utilisez par exemple le caractère #.** Par la suite, le fichier contient la liste des coups joués jusqu'à présent. Le format de cette partie est libre, mais assurez-vous d'utiliser le même format pour la sauvegarde et le chargement !

Fabriquez votre interface comme vous le désirez : vous êtes complètement libres ! Quelques points seront attribués à la beauté et la fonctionnalité de l'interface. La figure 1 montre un exemple minimal d'interface.

4.1 La modélisation de la solution

Cette fois-ci, vous êtes complètement libres au niveau de la modélisation de votre solution. Nous vous recommandons fortement de bien séparer les fonctionnalités en diverses méthodes de la classe contenant la fenêtre principale.

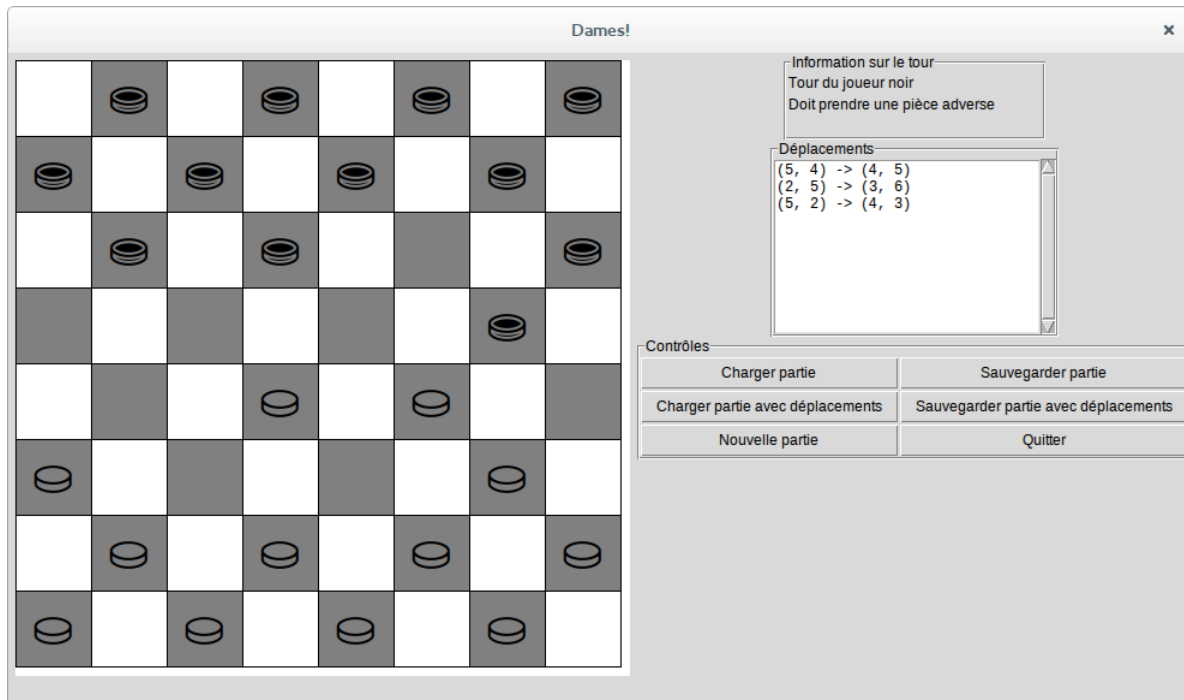


FIG. 1 : Exemple d'interface pour le TP4. Cet exemple n'est pas très joli, mais il est tout de même fonctionnel !

5 Ce que vous devez rendre

Vous devez rendre un fichier .zip comportant tous les fichiers nécessaires à l'exécution de votre programme. Le fichier principal, `tp4.py`, doit se trouver à la racine du fichier .zip.

Le nom du .zip doit respecter le format suivant : `TP4_Matricule.zip` où `Matricule` est le numéro matricule de l'un des membres de l'équipe.

La remise de ce TP doit se faire via le lien "remise de travaux" du site Web qui vous redirige vers votre guichet étudiant (aucune remise par courriel n'est acceptée). Il est important de respecter la date et l'heure de la remise, car tout travail remis en retard se verra attribué la note de 0. Veuillez aussi noter que Pixel n'autorise aucun retard. Assurez-vous également, après la remise, d'avoir déposé le bon fichier et que Pixel vous a bien confirmé la remise par courriel. Tout manquement à ces vérifications pourrait entraîner une note de 0 à ce travail.

6 Remarques

Qualité du code : Remarquez qu'aucun point ne pourra être comptabilisé si le programme principal ne s'exécute pas. En effet, il n'est pas acceptable de remettre un programme qui plante à un client. **En conséquence, si le programme plante lors de l'exécution d'un élément de votre programme, une note de 0 sera attribuée à cet élément.**

Par exemple : L'utilisateur tente de jouer une partie chargée. Le chargement plante, puis après s'être rabattu sur l'option de création de partie, le programme plante à nouveau. Une note de 0 sera attribuée au chargement du jeu et d'une nouvelle partie.

Autre exemple : S'il n'est ni possible de créer une nouvelle partie, ni possible de charger une partie parce que ces deux éléments plantent, les 100 points du programme principal seront perdus puisque le correcteur sera dans l'incapacité de les tester adéquatement. Vérifiez donc avec précaution votre programme.

Plagiat : Tel que décrit dans le plan de cours, le plagiat est interdit. Une politique stricte de tolérance zéro est appliquée en tout temps et sous toute circonstance. Tous les cas seront référés à la direction du département.

7 Les points qui seront vérifiés lors de la correction

- Le fonctionnement de votre implantation ;
- La présence des fonctionnalités demandées ;
- La qualité du code remis ;
- La qualité de l'interface.

8 Évaluation

Le travail est noté sur 100 points. Chaque fonctionnalité vaudra un certain nombre de points qui sera déterminé en fonction de la difficulté qu'elle représente, puis une partie des points sera attribuée à la qualité du code et de l'interface. Un barème plus détaillé est disponible en pièce jointe à ce document.

Bon travail !