

# Travail pratique # 4

## Un jeu de dames avec interface graphique

Pascal Germain

Date de remise : le dimanche 26 juillet 2020  
Pondération de la note finale : 20%

### 1 Objectifs

Ce travail a comme objectif de vous pratiquer avec une notion importante du développement d'applications d'envergure : la réutilisation de modules créés par un autre programmeur. Ce faisant, vous vous familiariserez davantage avec la création d'interfaces graphiques et la gestion des exceptions.

Une partie de la modélisation est déjà faite pour vous, via les modules déjà fournis avec cet énoncé. Le reste de la modélisation de votre programme est libre : vous pouvez créer de nouvelles classes, méthodes, attributs ou modules, ainsi que modifier les éléments déjà fournis.

Ce travail vous permettra de valider votre compréhension de la matière des modules 1 à 9, inclusivement.

### 2 Organisation du travail en équipe

Nous vous demandons de travailler en équipe, car c'est un objectif de votre formation académique, et vous pourrez ainsi vous partager la charge de travail. Pour ceux qui ont perdu leur coéquipier suite à un abandon ou pour ceux qui désirent changer de coéquipier, nous vous invitons à utiliser le forum du cours dans la section prévue à cet effet. Chaque coéquipier doit contribuer à parts égales au développement de ce travail. 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 ! Nous vous suggérons encore une fois d'utiliser Git et BitBucket pour réaliser ce travail de manière efficace.

### 3 Le problème à résoudre

Vous devez créer une interface graphique permettant de jouer aux dames, en utilisant la librairie `tkinter`. Vous devez considérer les mêmes règles et la même logique de jeu que pour le TP3, assurez-vous donc de revoir l'énoncé du TP3 en cas de doute. Une partie des points est accordée pour des fonctionnalités obligatoires et une autre partie pour des fonctionnalités optionnelles choisies parmi une liste de propositions. Pour effectuer ce travail, vous serez aussi à concevoir vous-même votre propre modélisation. Nous vous demandons de rédiger un court rapport expliquant les choix effectués.

Nous vous fournissons les fichiers suivants :

- `interface_dame.py` est la base du programme que vous devez compléter et modifier pour intégrer les fonctionnalités demandées. Il est possible d'ores et déjà d'exécuter ce programme, qui affiche une fenêtre contenant un damier. Vous constaterez que nous avons programmé un évènement qui détecte lorsque l'utilisateur clique sur une pièce, mais que le jeu de dames n'est pas fonctionnel. Commencez par bien lire et comprendre le code fourni. Ensuite, modifiez-le autant que vous le désirez.
- `canvas_damier.py` contient la définition de la classe « CanvasDamier », qui est un « widget » créé pour afficher à l'écran un objet « Damier » tel que programmé au TP3. Il est aussi possible de modifier ce fichier si vous le désirez (mais ce n'est pas nécessairement requis).
- `piece.py`, `position.py`, `damier.py` et `partie.py` sont les fichiers du TP3. Lors de la publication de l'énoncé, nous avons recopié les fichiers incomplets que l'on vous fournissait comme base du TP3. Ceux-ci sont suffisants pour exécuter le squelette d'interface. Vous êtes invités à les remplacer par les fichiers que vous avez complétés lors du TP3. Nous rendrons disponible le 8 juillet 2020 une solution au TP3 (c'est-à-dire les quatre fichiers complétés) que vous pourrez (ou non) utiliser dans le TP4 au lieu de votre propre code. Vous êtes libre de modifier le code de ces fichiers à votre guise si cela facilite l'implémentation des fonctionnalités du TP4.

En résumé, contrairement aux instructions des travaux pratiques précédents, nous vous laissons la liberté de concevoir votre propre modélisation pour ce travail. Vous pouvez modifier les fichiers fournis, incluant les arguments et les valeurs de retour des méthodes programmées précédemment. Avant de tout modifier, nous vous conseillons fortement de bien réfléchir aux choix qui devront être faits. Ces choix doivent être convenablement décrits dans le rapport accompagnant le code. Comme d'habitude, le code doit aussi être judicieusement commenté.

## 4 Modalités d'évaluation

Ce travail sera évalué sur 100 points distribués selon le barème suivant.

Critère	Pondération
Fonctionnalités obligatoires	40 points
Fonctionnalités optionnelles	30 points
Qualité du code	10 points
Rapport écrit	20 points
Total	100 points

Notez qu'un programme qui plante au début de l'exécution ou qui ne fait rien ne sera pas considéré. Un TP qui ne s'exécute pas correctement (et ainsi nous empêche de valider si l'une ou l'autre des fonctionnalités est présente ou non) pourrait donc recevoir une note de 0.

### 4.1 Fonctionnalités obligatoires

Dans un premier temps, on vous demande d'ajouter certaines fonctionnalités à l'interface fournie qui, pour l'instant, ne permet pas de jouer aux dames. Ces fonctionnalités doivent obligatoirement être présentes dans le programme que vous remettrez.

Fonctionnalités obligatoires	Pondération
Les pièces peuvent être déplacées sur le damier	4 points
Les déplacements invalides ne sont pas acceptés	4 points
On peut choisir une nouvelle pièce source après un mouvement invalide	4 points
Si une prise est possible, elle est obligatoire	4 points
Les prises multiples sont gérées correctement	4 points
Le tour des joueurs est alterné correctement	4 points
On peut créer une nouvelle partie	4 points
On peut quitter le jeu	4 points
Une rétroaction est donnée à l'utilisateur (messages)	4 points
Affiche la couleur du joueur qui doit faire un déplacement (tour courant)	4 points
Total	40 points

### 4.2 Fonctionnalités optionnelles

Choisissez parmi les fonctionnalités suivantes celles que vous désirez intégrer à votre jeu. Certaines fonctionnalités sont plus difficiles à réaliser que d'autres, et vous ne pouvez pas dépasser 30% de la note totale pour les fonctionnalités optionnelles.

Fonctionnalités optionnelles	Pondération
1. Option de lire les règlements du jeu	2 points
2. Permettre à l'utilisateur de changer le nom des joueurs	2 points
3. Calcul et affichage de statistiques de jeu	3 points
4. Afficher un compteur de temps de jeu par joueur	3 points
5. Affichage d'une liste des déplacements effectués	3 points
6. Permettre des raccourcis clavier	4 points
7. Sauvegarde des déplacements et possibilité de revoir cette partie	5 points
8. Damier plus convivial, par exemple surligner les positions disponibles	5 points
9. Génération d'un damier aléatoire	5 points
10. Faire un mouvement automatique lorsqu'il n'y a qu'une possibilité	5 points
11. Permettre de changer le thème (couleurs, etc.)	5 points
12. « Drag-and-drop » de pièces	5 points
13. Annuler le dernier mouvement	5 points
14. Mode tricherie <sup>1</sup>	5 points
15. Permettre plusieurs styles de jeu (international, Canadien, etc.)	5 points
16. Choisir les dimensions du damier (nombre de lignes et colonnes)	5 points
17. Afficher les positions sources valides (en considérant les règles)	5 points
18. Sauvegarder la partie dans un fichier texte	5 points
19. Charger la partie à partir d'un fichier texte	5 points
20. Jouer contre l'ordinateur	20 points
Total	Max 30 points

Si vous avez d'autres idées de fonctionnalités intéressantes, écrivez au professeur pour lui suggérer !

### 4.3 Qualité du code

Une portion de 10% de la note est attribuée à la qualité du code fourni. On considérera pour l'évaluation de ce critère le code d'en son ensemble (à la fois l'implémentation des fonctionnalités obligatoires et optionnelles). Les deux aspects suivants seront évalués.

**Respect des normes de programmation :** Nous vous demandons de prêter attention au respect des normes de programmation établies pour le langage Python. Entre autre, les noms des variables, classes et méthodes doivent être significatives.

**Documentation du programme :** Vous devez documenter toutes les méthodes et les interfaces des classes à l'aide de « docstrings ». Lorsque vous modifiez une méthode qui est fournie dans l'énoncé, ou encore une méthode provenant du TP3, vous devez aussi mettre à jour la documentation correspondante.

<sup>1</sup>Exemples : transformer pions en dames, détruire une pièce ennemie au hasard, ramener une pièce à la vie, permettre des déplacements interdits, etc.

## 4.4 Rapport écrit

Nous vous demandons de remettre un (court) rapport écrit **d'une à deux pages** qui explique les **choix de modélisation** que vous avez faits. Décrivez et **justifiez** les modifications que vous avez apportées aux fichiers de code fournis avec l'énoncé.

Pour chacune des **fonctionnalités optionnelles** implémentées dans votre programme, le rapport doit **décrire brièvement les choix qui ont été faits** pour intégrer cette fonctionnalité dans l'interface graphique (gestion des événements, utilisation de « widgets » spécifique, etc.).

Le rapport vaut pour 20% de l'évaluation. Il doit être remis sous la forme d'un fichier pdf portant le nom **rapport.pdf**. Les critères évalués seront les suivants.

Critère	Pondération
Description et justification des choix de modélisations	10 points
Description et justification des modifications au code fourni	5 points
Qualité de la langue	5 points
Total	20 points

## 5 Ce que vous devez rendre

Vous devez rendre une archive **.zip** contenant tous les fichiers nécessaires à l'exécution de votre TP. Vous devez également fournir le fichier **rapport.pdf** et le fichier **correction.txt**. Ce dernier doit être complété avec vos noms et numéros de dossiers, ainsi **la liste des numéros fonctionnalités facultatives que vous avez implémentées**. Vous pourrez être pénalisés pour avoir remis des fichiers ou dossiers non nécessaires.

Cette archive doit être remise via l'utilitaire de remise de travaux, disponible sur le site Web du cours. Ne remettez qu'un travail par équipe.

### Remarques

**Plagiat :** Tel que décrit dans le plan de cours, le plagiat est strictement interdit. Ne partagez pas votre code source à quiconque. Une politique stricte de tolérance zéro est appliquée en tout temps et sous toute circonstance. Tous les cas détectés seront référés à la direction de la faculté.

**Retards :** Pixel n'acceptera pas les travaux remis en retard. Tout travail remis en retard peut être envoyé par courriel à l'enseignant. Une pénalité de 10% sera appliquée pour un travail remis le lendemain de la remise, puis une pénalité de 25% sera attribuée à un TP remis le surlendemain. Une note de 0 sera attribuée pour un plus long retard.

**Remises multiples :** Il vous est possible de remettre votre TP plusieurs fois sur Pixel. La dernière version sera considérée pour la correction.

Bon travail!