

# Travail pratique # 4

## PyChecs 2

(optionnellement en équipes de 2)

Honoré Hounwanou

Date de remise : le vendredi 24 Avril 2020  
Pondération de la note finale : 16%

## 1 Objectifs

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

Pour ce travail, nous vous fournissons une solution du TP3, que vous avez le loisir d'utiliser et de modifier pour concevoir votre TP4, si vous ne désirez pas participer au concours Pierre-Ardouin (voir section 2). Nous vous fournissons également le solutionnaire du laboratoire 11 : un échiquier implémenté dans un `Canvas` de la librairie `tkinter`. Ces deux sous-projets ne sont pas liés entre deux : une partie du travail est d'utiliser et d'adapter ces éléments pour former un tout.

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

## 2 Le prix Pierre Ardouin

Depuis l'automne 2013, le département d'informatique et de génie logiciel a mis en place un concours récompensant l'équipe qui aura produit le meilleur TP/projet dans le cadre d'un cours. Ces travaux de session ont l'envergure d'un mini-projet qui est admissible par rapport aux normes fixées par le département. À la suite des évaluations des travaux, l'enseignant du cours détermine l'équipe gagnante; chaque membre de l'équipe gagnante reçoit alors une bourse de 50\$ ainsi qu'une attestation remise par le département.

De plus, le département d'informatique et de génie logiciel a mis en place une bourse Élite, appelée bourse «Pierre Ardouin», qui vise à récompenser le meilleur projet de session, tous cours confondus. Deux principaux critères guident le choix des évaluateurs dans l'identification du lauréat : l'excellence du travail (par rapport à ce qui est demandé dans l'énoncé) et l'aspect créativité/innovation. Il est actuellement prévu une bourse de 200\$

pour récompenser chaque membre de l'équipe «élite» gagnante (pour un maximum de 1000\$ pour toute l'équipe). Aussi, le département veille à publier l'information sur un site Web dédié : <http://www.ift.ulaval.ca/vie-etudiante/prix-pierre-ardouin>.

À la deuxième moitié du mois de mai de chaque année universitaire, le département organise une cérémonie pour honorer les finalistes et le lauréat du prix «Pierre Ardouin» des sessions d'automne et d'hiver, et leur remettre une attestation.

Les travaux pratiques 3 et 4 seront évalués pour le prix Pierre Ardouin. **Si vous désirez y participer, vous devez utiliser le code de *votre* TP3 (et non la solution fournie) pour construire votre TP4.** Vous avez par contre le droit d'utiliser le code du laboratoire 11.

### 3 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 n'ont pas encore trouvé de coéquipier ou qui veulent former une nouvelle équipe, nous vous invitons à utiliser le forum du cours dans la section prévue à cet effet. Chaque coéquipier doit contribuer à part égale 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! **Chaque équipe doit être formée sur le portail des cours, sur la page du TP4.**

Nous vous rappelons que nous fournissons sur le site Web du cours un tutoriel pour utiliser des technologies permettant de partager votre code avec votre coéquipier. Chaque membre d'une équipe possède sur son ordinateur une copie locale du code, mais le code est synchronisé par un service dans le «*cloud*» et nous pouvons également voir l'historique (qui a fait quoi et quand). De plus, PyCharm intègre ces outils de manière très conviviale.

### 4 Le problème à résoudre

Vous devez créer une interface graphique permettant de jouer aux échecs, en utilisant la librairie `tkinter`. Vous devez considérer au minimum 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. Un sous-ensemble des points est accordé à des fonctionnalités obligatoires, et le reste des points sera accordé en fonction des fonctionnalités supplémentaires que vous aurez choisies. Nous vous fournissons les fichiers suivants :

```
pychecs2/  
  echecs/  
    __init__.py  
    piece.py  
    echiquier.py  
    partie.py  
  
  interface/  
    __init__.py  
    exemple.py  
  
  __init__.py  
  __main__.py
```

Les fichiers `__init__.py` sont des fichiers spéciaux indiquant à Python que le dossier est un *package*, c'est à dire un dossier contenant des modules qui peuvent être importés à l'aide de `from nom_package.nom_module import nom_classe`. Le fichier `__main__.py` est le programme principal à partir duquel exécuter le TP. Vous devez l'adapter à vos besoins : c'est ce fichier qu'exécuteront les correcteurs de votre travail.

Les prochaines sous-sections expliquent les étapes à réaliser (dans l'ordre) pour bien réussir ce TP.

## 4.1 Analyser le solutionnaire du TP3

Pour ce TP4, nous vous fournissons une solution du TP3, que vous devez d'abord comprendre, car vous devrez la modifier pour combler vos besoins. Notez que pour participer au prix Pierre-Ardouin, vous devez plutôt utiliser **votre** solution du TP3, mais vous avez le droit de l'améliorer après avoir pris connaissance de la solution jointe avec le TP4.

Notez également que vous pourriez avoir une solution très différente de la nôtre, ce qui ne veut pas dire que la nôtre est meilleure. Profitez-en pour les comparer et juger des avantages/désavantages de votre solution par rapport à la nôtre.

## 4.2 Ajouter des classes d'exception

Nous vous demandons d'utiliser la gestion des exceptions pour réaliser votre TP4. Vous devez donc créer de nouvelles classes d'exceptions, puis de modifier la méthode `deplacer` de la classe `Echiquier` afin que celle-ci lance une exception appropriée si jamais le déplacement demandé est invalide.

De cette manière, le code de votre interface pourra appeler la méthode `deplacer` **sans avoir à valider les positions préalablement** : si le déplacement demandé était invalide, le déplacement ne sera pas effectué et une exception sera reçue.

Vous pouvez également modifier les autres méthodes des diverses classes, ou en ajouter, puis lancer des exceptions appropriées selon le cas. Ces exceptions devront être capturées dans l'interface graphique, dans le but de valider les sélections et déplacements. Des exemples

de classes d'exception que vous pourriez créer sont `AucunePieceAPosition`, `DeplacementInvalide`, `PieceDeMauvaiseCouleur`, etc.

### 4.3 Analyser le solutionnaire du TD11

Nous vous fournissons également un solutionnaire du travail dirigé #11. Celui-ci n'est pour l'instant aucunement lié au code du TP3. Comprenez le code de ce solutionnaire, c'est à partir de celui-ci que vous travaillerez à implémenter votre jeu d'échecs.

### 4.4 Adapter la classe `CanvasEchiquier`

Trouvez les éléments de la classe `CanvasEchiquier` qui sont redondants avec les éléments du TP3. Vous devez ensuite modifier (ou supprimer) les éléments qui ne vous sont pas utiles, et bien sûr lier ce module avec les classes disponibles dans le TP3. Par exemple, a-t-on besoin d'un dictionnaire contenant des pièces? Nous pourrions considérer qu'il est plus intéressant de fournir en argument au constructeur une instance de la classe `pychecs2.echecs.echiquier.Echiquier...`

### 4.5 Ajouter des fonctionnalités

Une fois les étapes précédentes réalisées, il est temps d'ajouter des fonctionnalités à cette interface qui, pour l'instant, ne permet pas de jouer aux échecs.

#### 4.5.1 Fonctionnalités obligatoires

Ces fonctionnalités doivent être présentes dans le programme que vous remettrez. Les points perdus ici ne peuvent pas être récupérés en accumulant des fonctionnalités facultatives.

Ces fonctionnalités seront évaluées selon le fait qu'elles **fonctionnent** ou non (et non la manière dont vous les avez implémentées, vous êtes totalement libres à ce niveau). Nous les testerons directement via votre interface graphique. Si une fonctionnalité fait planter votre programme, lance des erreurs dans la console ou ne se comporte pas comme prévu, vous perdrez une partie ou la totalité des points.

Fonctionnalité	Pondération
Les pièces peuvent être déplacées sur l'échiquier	10 points
La pièce sélectionnée est facilement identifiable	5 points
Le tour des joueurs est alterné correctement	5 points
Les déplacements invalides ne sont pas acceptés	5 points
On peut changer d'idée de pièce source sans compléter le déplacement	5 points
La fin de la partie est détectée et affichée	5 points
On peut sauvegarder une partie via un bouton	5 points
On peut charger une partie via un bouton	5 points
On peut démarrer une nouvelle partie	5 points
De l'information sur la partie est affichée (au tour de quelle couleur)	5 points
Qualité des messages d'erreur donnés à l'utilisateur	5 points
Utilisation de la gestion des exceptions	8 points
Qualité du code (noms de variables, style, commentaires, documentation)	2 points
Total	70 points

#### 4.5.2 Fonctionnalités facultatives

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 une note de 100%. Vous pouvez par contre récupérer des points qui auraient pu être perdus dans une autre fonctionnalité facultative!

Fonctionnalité	Pondération
1. Option de lire les règlements du jeu	2 points
2. Affichage d'une liste des déplacements effectués	3 points
3. Permettre d'annuler le dernier mouvement	5 points
4. Sauvegarde et possibilité de «revoir» une partie enregistrée	5 points
5. Après la sélection, afficher les déplacements et prises valides	10 points
6. Jouer contre l'ordinateur	10 points
7. Permettre de changer le thème (couleurs, etc.)	5 points
8. «Drag-and-drop» de pièces	5 points
9. Afficher un compteur de temps de jeu pour chaque joueur	5 points
10. Afficher les pièces que chaque joueur a perdu	5 points
11. Remplacer les caractères unicode par de vraies images	5 points
12. Demander si on veut sauvegarder avant de quitter/charger	5 points
13. Règles avancées : le «roque»	5 points
14. Règles avancées : prise «en passant»	5 points
15. Règles avancées : détection des mises en échec	5 points
16. Règles avancées : empêcher un mouvement qui provoque un échec	5 points
17. Règles avancées : détecter une partie nulle de type «pat»	10 points
18. Règles avancées : détecter «échec et mat»	10 points
19. Règles avancées : «promotion» du pion	5 points
20. Fenêtre de bienvenue, avec explications du jeu et sélection d'options	5 points
Total	110 points

Si vous avez d'autres idées de fonctionnalités intéressantes, demandez-nous de les ajouter dans cet énoncé!

**Note importante:** Pour que vos fonctionnalités facultatives soient évaluées, vous devez les énumérer la liste de celles que vous avez implémentées dans la zone appropriée du fichier `correction.txt`. Pour les fonctionnalités de règles avancées, vous devez également fournir un fichier de partie sauvegardée nommé `regle_XX.txt`, où `XX` est remplacé par le numéro de la règle dans la liste ci-haut. Vous devez indiquer dans le fichier `correction.txt` quel déplacement effectuer, une fois le fichier de sauvegarde ouvert, pour valider que la règle a bien été programmée. Attention par contre : les règles avancées ont beaucoup de subtilités, et vous pourriez perdre plusieurs points si nous nous rendons compte que vous ne les avez pas toutes considérées.

## 5 Modalités d'évaluation

Ce travail sera évalué sur 100 points, et les points seront attribués pour chaque fonctionnalité. Une fonctionnalité qui plante à l'exécution ou qui ne fait rien ne sera pas considérée. 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. Des pénalités pourront être appliquées si les demandes ne sont pas respectées.

## 6 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. Des logiciels comparant chaque paire de TPs pourraient être utilisés pour détecter les cas de plagiat.

**Retards :** Tout travail remis en retard peut être envoyé par courriel à l’enseignant si le portail des cours n’accepte pas la remise. 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. **N’oubliez pas de créer votre équipe sur le portail des cours** dans les temps. Une omission de votre part pourrait vous mettre en situation de retard.

**Remises multiples :** Il vous est possible de remettre votre TP plusieurs fois sur le portail des cours, **en conservant le même nom de fichier**. La dernière version sera considérée pour la correction. Ne laissez pas plusieurs fichiers avec des noms différents sur le portail.

**Respect des normes de programmation :** Nous vous demandons de prêter attention au respect des normes de programmation établies pour le langage Python, notamment de nommer vos variables et fonctions en utilisant la convention suivante : `ma_variable`, `fichier_entree`, etc. Utiliser **PyCharm** s’avère être une très bonne idée ici, car celui-ci nous donne des indications sur la qualité de notre code (en marge à droite, et souligné).

## 7 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 `correction.txt`, complété avec vos noms et matricules, **et incluant la liste des numéros fonctionnalités facultatives que vous avez implémentées**. Ne remettez aucun fichier ou dossier non nécessaire. Cette archive doit être déposée sur le site Web du cours.

**Bon travail!**