

Travail pratique # 4

PyMineur : le retour

(en équipe de 2 ou 3)

Enseignant : Pascal Germain

Date de remise : le vendredi 18 décembre 2020
Pondération de la note finale : 15%

1 Objectifs

Ce travail a comme objectif de vous faire pratiquer une notion importante du développement d'applications d'envergure : la réutilisation de modules. Ce faisant, vous vous familiariserez aussi avec la création d'interfaces graphiques.

Ce travail vous permettra de valider votre compréhension de la matière des modules 1 à 6, 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. Par défaut, nous considérons que vous conservez les mêmes équipes que pour le TP3. Pour modifier votre équipe, merci de contacter le professeur par courriel.

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 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 !

3 Le problème à résoudre

Vous devez créer une interface graphique permettant de jouer au jeu démineur 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. Une partie des points est accordée pour des fonctionnalités obligatoires et une autre partie pour des fonctionnalités optionnelles choisies parmi une liste de propositions.

3.1 Fichiers fournis

Nous vous fournissons les fichiers suivants :

- `interface_partie.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 par le biais du fichier `principal.py`, qui ne fait que lancer l'interface. Vous constaterez que nous avons déjà programmé un événement qui détecte lorsque l'utilisateur clique sur une case, mais que le jeu de démineur n'est pas fonctionnel. Commencez par bien lire et comprendre le code fourni. Ensuite, modifiez-le autant que vous le désirez. Cette partie du code fournie n'est pas documentée pour vous, mais le code remis doit être pleinement documenté.
- `bouton_case.py` contient la définition de la classe « BoutonCase », qui est un « widget » créé pour afficher à l'écran un objet « Case » tel que programmé au TP3. Il est aussi possible de modifier le code de ce fichier si vous le désirez, mais ce n'est pas nécessairement requis. Dans tous les cas, ajoutez les commentaires requis pour documenter la classe BoutonCase.
- `tableau.py` et `case.py` sont les fichiers du TP3 contenant les classes « Tableau » et « Case ». Lors de la publication de l'énoncé, nous avons recopié le fichier `tableau.py` incomplet que l'on vous fournissait comme base du TP3. Celui-ci est suffisant pour exécuter le squelette d'interface. Vous êtes invité à le remplacer par le fichier que vous avez complété lors du TP3. Nous rendrons disponible le 27 novembre 2020 une solution au TP3 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. Le code remis doit aussi être judicieusement commenté.

3.2 Déroulement du jeu

Le jeu se déroule comme pour le TP3 : le joueur dévoile des cases en essayant d'éviter les mines. S'il dévoile une mine, la partie est perdue et un message s'affiche pour informer le joueur de sa défaite. S'il dévoile toutes les cases sans mine, la partie est gagnée et un message s'affiche pour informer le joueur de sa victoire. Peu importe l'issue de la partie, toutes les cases sont dévoilées. Contrairement au TP3, terminer une partie ne doit pas fermer le jeu. L'utilisateur peut démarrer une nouvelle partie ou quitter en cliquant sur le bouton associé.

3.3 Connecter l'interface à la classe Tableau

Il ne faut pas reprogrammer les fonctionnalités du jeu dans le fichier de l'interface ! Il faut plutôt réutiliser le code que vous avez déjà. Par exemple, le dévoilement d'une case devrait utiliser la méthode `devoiler_case()` de la classe `Tableau`. D'autres méthodes des classes `tableau` et `Case` vous seront utiles pour savoir si une case contient une mine, pour obtenir le nombre de mines voisines ou pour déterminer si la partie est terminée. Cela dit, vous pouvez modifier ou bonifier les classes `tableau` et `Case` pour qu'elles répondent mieux à vos besoins.

3.4 Fonctionnalités de l'interface

En plus de pouvoir jouer au démineur, votre interface doit posséder au minimum les fonctionnalités suivantes.

- Un bouton « Nouvelle partie » (ou un menu) permet de redémarrer la partie en tout temps.
- Un bouton « Quitter » (ou un menu) permet de quitter le programme en tout temps. Avant de quitter, une fenêtre de dialogue doit demander la confirmation de l'utilisateur avant de quitter.
- Un bouton « Instructions » (ou un menu) doit afficher à l'utilisateur les instructions du jeu.
- Votre interface doit offrir un moyen de changer la configuration du tableau (le nombre de lignes, de colonnes et de mines).
- Votre interface doit permettre de sauvegarder l'état de la partie **dans un fichier texte** et de continuer une partie en chargeant une partie à partir d'un fichier texte. La sauvegarde et le chargement doivent se faire à l'aide des boîtes de dialogues offertes par le module « `filedialog` » de `tkinter`. Attention : Ceci implique que vous devez définir un format de fichier texte pour sauvegarder l'état d'une partie, c'est-à-dire les dimensions du tableau, les positions des mines, les cases dévoilées, etc.

4 Modalités d'évaluation

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

Critère	Pondération
Fonctionnalités obligatoires	60 points
Fonctionnalités optionnelles	30 points
Qualité du code	10 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 au démineur. Ces fonctionnalités doivent obligatoirement être présentes dans le programme que vous remettrez.



Fonctionnalité	Pondération
Le déroulement du jeu et le dévoilement des cases sont adéquats	20 points
On peut choisir la taille de damier et le nombre de mines	10 points
On peut démarrer une nouvelle partie en tout temps	5 points
On peut afficher les instructions du jeu	5 points
Le bouton quitter est fonctionnel (incluant un message de validation)	3 points
Un message annonce la victoire/défaite du joueur	2 points
La solution est affichée lors d'une défaite	3 points
Un compteur de tour est mis à jour	2 points
La sauvegarde et chargement de la partie sont fonctionnels	10 points
Total	60 points

4.2 Fonctionnalités optionnelles

Choisissez parmi les fonctionnalités suivantes celles que vous désirez intégrer à votre jeu. Vous ne pouvez pas dépasser 30% de la note totale pour les fonctionnalités optionnelles. Vous pouvez par contre récupérer des points qui auraient pu être perdus dans une autre fonctionnalité optionnelle !

Important. Lors de la remise du travail, le fichier `correction.txt` doit inclure la liste des fonctionnalités optionnelles que vous avez programmées.

	Fonctionnalités optionnelles	Pondération
0	Dévoilement en cascade des cases vides (voir l'énoncé du TP3, section 4.5)	10 points
1	Placer un drapeau avec le clic droit de la souris	5 points
2	Permettre des raccourcis clavier	5 points
3	Amélioration des graphiques (images de mines, chiffres de couleur, etc)	5 points
3+	Options permettant de personnaliser les couleurs de l'interface	+10 points
4	Afficher un chronomètre de temps de jeu	5 points
5	Calcul et affichage d'un pointage basé sur des statistiques de jeu	5 points
5+	Maintenir un « tableau des meneurs » entre les parties	+10 points
6	Mode de jeu « contre-la-montre » : temps de jeu maximum	10 points
7	Mode de jeu « campagne » : tableaux de plus en plus difficiles	10 points
7+	Mode de jeu « campagne » avec une histoire entre chaque tableau	+10 points
8	Possibilité d'annuler le dernier coup joué	10 points
8+	Possibilité d'annuler un nombre arbitraire de coups joués	+10 points
9	Possibilité de revoir le déroulement d'une partie	10 points
10	Mode conception de niveau	15 points
10+	Sauvegarde et chargement des niveaux conçus	+10 points
11	Ajouts d'effets sonores	5 points
12	Assistance : Dévoilement d'une case minée	5 points
13	Assistance : Coloration des cases qui peuvent logiquement être dévoilées	15 points
Total		Max 30 points

Notez que certaines fonctionnalités facultatives sont divisées en deux. Par exemple, implémenter les fonctionnalités 3 et 3+ vous donnera un total de 15 points. D'autres fonctionnalités se combinent naturellement entre elles. Par exemple, la fonctionnalité 4 (afficher un chronomètre) se combine très bien avec 5, 5+ et 6. Vous pouvez tirer profit de ces combinaisons pour faciliter votre travail! Aussi, certaines fonctionnalités sont plus difficiles à réaliser que d'autres. Prenez le temps de bien réfléchir à vos choix avant de commencer à programmer.

Si nécessaire, utilisez le forum pour demander des précisions sur une fonctionnalité optionnelle. Nous vous permettons aussi de suggérer d'autres fonctionnalités intéressantes; utilisez le forum pour les suggérer au professeur!

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 autres, les noms des variables, classes et méthodes doivent être significatifs.

Documentation du programme : Vous devez documenter toutes les méthodes et les in-

terfaces 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.

5 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 TP’s 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 25% sera appliquée pour un travail remis le lendemain de la remise. Une note de 0 sera attribuée pour un plus long 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.

6 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.

Important. Le fichier `correction.txt` doit inclure la liste des fonctionnalités optionnelles que vous avez programmées.

Ne remettez aucun fichier ou dossier non nécessaire. Cette archive doit être déposée sur le site Web du cours.

Bon travail !