

Travail pratique # 4

Pymafia (la suite)

(préférentiellement en équipe de 2)

Simon Hardy

Date de remise : le mardi 27 avril 2021
Pondération de la note finale : 12%

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 davantage avec la création d'interfaces graphiques.

Ce travail vous permettra de valider votre compréhension de la matière des modules 1 à 7, inclusivement.

2 Le problème à résoudre

Vous devez créer une interface graphique permettant de jouer au jeu de dés Mafia en utilisant la librairie `tkinter` (une autre librairie peut être utilisée, mais nous n'offrirons pas de support dans ce cas). Vous devez considérer les mêmes règles et la même logique de jeu que pour le TP3, au besoin revoir l'énoncé du TP3. Nous vous suggérons la structure de fichiers suivante :

TP4/

```
pymafia/  
  __init__.py  
  __main__.py  
  de.py  
  joueur.py  
  joueur_humain.py  
  joueur_ordinateur.py  
  partie.py  
  __main__.py  
  pymafia_gui.py
```

Le package `pymafia` est celui que vous avez développé pour le TP3. Si vous n'avez pas été en mesure de terminer un TP3 fonctionnel, nous fournissons une solution. Comme nous vous l'avons expliqué au dernier TP, le fichier `__init__.py` est un fichier spécial 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 nouveau fichier `__main__.py` à l'extérieur du package `pymafia` est le programme principal à partir duquel exécuter le TP4. Vous devez l'adapter à vos besoins : c'est ce fichier qu'exécutera le correcteur ou la correctrice de votre travail.

Pour ce TP, nous vous avons fourni une interface de base lors du TD11. Vous êtes libres de l'utiliser ou non. Vous pouvez renommer le fichier `TD_pymafia.py` pour démarrer. Les prochaines sous-sections expliquent les étapes à réaliser pour bien réussir ce TP.

2.1 Programmer une interface graphique pour le jeu `pymafia`

À l'aide des widgets de Tkinter vus en classe, programmez une interface graphique qui permettra de jouer au jeu de dés. Entre autres, vous pourriez avoir besoin de cadres et d'étiquettes pour afficher des informations et possiblement des boutons pour certaines actions. Consultez l'énoncé du TD11 pour plus d'informations sur une proposition d'interface.

Pour obtenir tous les points du TP, vous aurez besoin de widgets qui permettent de configurer une partie (choix du nombre de joueurs). En fonction de l'information obtenue, vousinstancierez un objet partie de la classe `Partie`. Si une partie est déjà en cours, votre programme doit vérifier avec l'utilisateur s'il veut vraiment démarrer une nouvelle partie (possiblement avec un `messagebox`). Un deuxième widget est utilisé pour afficher les instructions du jeu dans un `messagebox` et finalement un dernier widget permet de quitter le jeu (on vérifie avec l'utilisateur si c'est bien son intention). Ceci peut être fait avec un menu, ou avec des boutons ajoutés à l'interface.

2.2 Déroulement du jeu

Le jeu se déroule comme pour le TP3 : avant le début de la partie, il faut déterminer un premier joueur. On demande à ce dernier de déterminer le sens. Ensuite, chaque joueur joue à tour de rôle. Lorsque joués, les dés de valeur 1 et 6 sont déplacés. Les dés des joueurs doivent être mis à jour dans l'interface lorsque ceux-ci les lancent et quand les dés doivent être déplacés. Un widget doit être présent pour indiquer le numéro de la ronde en cours, tout comme le score des joueurs doit aussi être présent dans l'interface.

Lorsqu'une ronde se termine (un joueur n'a plus de dés), un message s'affiche pour informer les joueurs de la fin de la ronde. Les joueurs perdants lancent leurs dés et leur score est diminué en conséquence (nous suggérons un score initial de 50 points). Ces points sont ajoutés au score du gagnant de la ronde. Ceux qui veulent programmer le jeu complet, les joueurs dont le score tombe à 0 se retirent du jeu et l'interface doit indiquer ce retrait. Cette dernière fonctionnalité est facultative. Lorsque le nombre maximal de rondes est atteint, la partie se termine et un message indique le ou les gagnants. Contrairement au TP3, ter-

miner une partie ne doit pas fermer le jeu. L'utilisateur peut démarrer une nouvelle partie ou quitter en choisissant le bouton/menu associé.

2.3 Connecter l'interface au package `pymafia`

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, démarrer une partie devrait provoquer l'instanciation d'un objet de la classe `Partie`, qui sera un attribut de votre classe principale. Les joueurs, les dés, le joueur courant, le joueur suivant, le numéro de la ronde, etc, tous ces objets et attributs seront toujours gérés par l'objet `partie`. Il faudra faire appel aux méthodes appropriées de la classe `Partie`. Par exemple, les méthodes `determiner_joueur_suivant()`, `reinitialiser_dés_joueurs()`, `deplacer_les_dés_1_et_6()` et `verifier_si_fin_de_ronde()` sont des méthodes que votre interface devrait utiliser. Vous y accéderez par l'intermédiaire de l'attribut `partie` de l'interface. Lors de la présentation du TP4 en classe, nous vous proposerons une organisation du code qui tirera profit de la programmation événementielle.

2.4 Jeu de base

Pour faciliter la réalisation de ce TP, nous nous attendons à un jeu qui permet à 2 à 4 joueurs humains de jouer. Il est facultatif de permettre à plus de joueurs de jouer ou d'inclure des joueurs ordinateur. Dans le jeu, lorsque plusieurs joueurs doivent lancer les dés, vous pouvez aussi leur permettre de lancer les dés un après l'autre, plutôt que tous puissent lancer les dés quand bon leur semble. Le retrait de joueurs est aussi facultatif.

3 Modalités d'évaluation

Ce travail sera évalué sur 12 points. Les éléments de l'interface doivent fournir des fonctionnalités qui sont opérantes. 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 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 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.

Fonctionnalité	Points
L'interface permet de redémarrer une partie (avec validation)	1 point
L'interface demande le nombre de joueurs de la partie	1 point
L'interface a un widget pour afficher les instructions du jeu	1 point
L'interface a un widget pour quitter le jeu (avec validation)	1 point
L'interface permet à chaque joueur de lancer leurs dés à tour de rôle avec activation des boutons et affichage du lancer de dés	1 point
L'interface affiche le numéro de la ronde et le score des joueurs	0,5 point
Un joueur est désigné premier joueur (au hasard ou aux dés), avec affichage	1 point
Le jeu demande au premier joueur de déterminer le sens de la partie	0,5 point
Le jeu fait le déplacement des dés de valeur 1 et 6, avec mise à jour des dés	1 point
Le jeu détecte la fin d'une ronde et calcule l'échange de points selon les dés	1 point
Le jeu détecte la fin d'une partie et désigne un ou des gagnants.	1 point
Qualité du code (noms de variables, style, commentaires, documentation)	2 points
Total	12 points

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).

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 `correction.txt`, complété avec vos noms et numéros d'étudiant. Ne remettez aucun fichier ou dossier non nécessaire. Cette archive doit être déposée sur le site Web du cours.

Bon travail!