TP Programmation Orientée Objet
Langage Python

Sujet

On souhaite modéliser un jeu de cartes faisant intervenir différents types de personnages. Ces personnages sont caractérisés par leur nom et un nombre de points de vie initial attribué aléatoirement. Ces personnages peuvent combattre et/ou soigner. Dans le cadre de ce TP les personnages se restreindront à :

- des guerriers pouvant combattre
- des soigneurs pouvant soigner
- et des paladins pouvant combattre et soigner.

Les paladins ne combattent ni ne soignent de la même façon que les guerriers ou les soigneurs.
Les personnages pouvant combattre ont des points d'attaques qui déterminent le nombre de points de vie que perdra leur adversaire lorsqu'ils le blessent.
Les personnages pouvant soigner ont des points de soin qui déterminent le nombre de points de vie qu'ils redonnent à un personnage lorsqu'ils le soignent.
On associe également aux personnages un message permettant de décrire chaque action (combat et soin) qu'ils font.
Le module `random` propose pas mal de fonctionnalités autour des nombres aléatoires. Par exemple la fonction `randint(a, b)` retourne un entier aléatoire compris entre a et b.

Question 1
Modéliser les différents personnages, en veillant à avoir plusieurs personnages de chaque type (par exemple plusieurs guerriers avec des caractéristiques/attaques différentes). Cette modélisation devra utiliser des classes abstraites et interfaces et avoir au moins 4 niveaux de hiérarchie.
Coder en Python cette modélisation.

Question 2
Ajouter une méthode `allInfo()` qui retourne une chaîne de caractères qui servira à l'affichage des caractéristiques des personnages. Vous pourrez utiliser les méthodes d'introspection qu'offre Python (`type, isinstance, issubclasse`, etc.).

Question 3
Implémenter des guerriers, soigneurs et paladins qui combattront et soigneront différemment les uns des autres. Voici quelques exemples :

- l'attaquant ne blesse son adversaire que si celui-ci est proche : à chaque attaque on pourra déterminer aléatoirement la distance entre les personnages
- certains combattant font plus de dommages lorsqu'ils sont loin
- certains combattants peuvent subir eux-mêmes des dommages lorsqu'ils attaquent

— le soigneur ne peut soigner que si le blessé est proche
— le soigneur perd des points de vie lorsqu'il soigne
— la récupération du blessé prend plusieurs jours
— ...

Question 4
Tester votre programme avec plusieurs personnages.

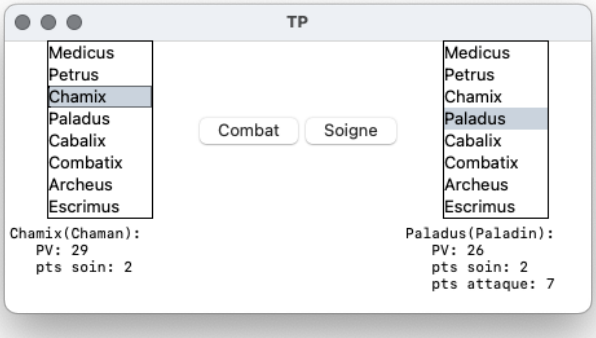
Vous devez arriver ici à la fin du premier TP

Question 5
Modifier votre programme pour que l'ensemble des personnages intervenant dans votre jeu soit dynamique en utilisant des collections. Vous pourrez éventuellement rajouter une méthode permettant de créer à la volée des personnages.

Question 6
Ajouter ce qu'il faut pour traiter les exceptions.

Vous devez arriver ici à la fin du deuxième TP

Question 7
Utiliser le module `tkinter` pour écrire la classe `Application` permettant d'obtenir l'interface graphique suivante en utilisant les *widgets* `Listbox, Text, Button` et en les plaçant en utilisant le *layout* `grid` :



Question 8
Ecrire la méthode permettant d'afficher les caractéristiques du personnage sélectionné dans la `Listbox` dans le *widget* `Text` correspondant.

Question 9
Ecrire la méthode permettant de déclencher l'action entre les 2 personnages correspondants et de mettre à jour l'affichage des *widgets* `Text` correspondants.

Question 10
Connecter ces méthodes aux *widgets* correspondants à l'aide de la fonction `bind`.

Question 11
Utiliser la méthode `messagebox.showinfo` pour afficher les exceptions dans une fenêtre *popup*.