

Conception et programmation objets avancées

Sujet 3 : design pattern stratégie

Vous devez programmer un jeu où des animaux se déplacent et combattent. Les animaux peuvent aussi attaquer : se déplacer puis combattre ou disparaître en se cachant dans la haie. Dans la version courante de votre jeu, vous avez 3 types d'animaux – écureuil, tortue et mouffette – qui ont chacun une technique de déplacement et chacun une façon de combattre :

- un écureuil se déplace dans les arbres et bombarde de noix ;
- une tortue marche lentement et rentre dans sa carapace pour rouler et faire tomber ses adversaires ;
- une mouffette trotte et projette du liquide malodorant sur ses ennemis.

1. Concevez un diagramme de classes détaillé pour représenter ces données.

On vous demande maintenant d'ajouter deux nouveaux types d'animaux à votre jeu :

- un porc-épic qui trotte comme la mouffette mais combat en lançant ses aiguilles ;
- un hérisson qui trotte et combat comme la tortue.

2. Modifiez votre diagramme de classes de façon détaillée en conséquence.

3. Répérez les méthodes qui seront rigoureusement identiques.

On vous demande maintenant de modifier le code de la méthode qui permet de trotter pour prendre en compte le relief.

4. Quelles méthodes devrez-vous modifier ? Devrez-vous faire plusieurs fois la même intervention dans des classes différentes ?

5. Est-ce que cette façon de faire vous permet de fournir un code facilement extensible et maintenable ?

Vous allez donc utiliser le *design pattern stratégie* pour séparer ce qui varie (les stratégies de déplacement et de combat) de ce qui ne varie pas (tous les animaux en voyant un ennemi vont se déplacer et combattre et ils peuvent tous se cacher sous la haie). Les parties variables sont déplacées dans deux **interfaces** : une interface *StrategieDeplacement* contenant une méthode *seDeplacer()* et une interface *StrategieCombat* contenant une méthode *combattre()*. La classe *Personnage* possède donc une stratégie de déplacement et une stratégie de combat.

6. En vous aidant du cours, donnez le nouveau diagramme de classes détaillé.

7. Codez les différentes classes (en vous aidant au maximum de l'IDE) puis créez une classe Jeu qui permet de tester votre code.

8. Ajoutez un nouveau type d'animal : un opossum qui se déplace comme l'écureuil dans les arbres et évite le combat en faisant le mort. Qu'avez-vous dû ré-écrire ?

9. Si vous deviez modifier la méthode de déplacement dans les arbres, quelle(s) classe(s) devriez-vous modifier ? Devriez-vous modifier toutes les classes des animaux qui utilisent ce type de déplacement ? Quel avantage y voyez-vous ?

10. L'écureuil peut parfois changer de stratégie de déplacement : il peut se mettre à trotter. Ajoutez le nécessaire et testez.