

Examen de MCPOOA

Partie C++

Documents non autorisés

La compréhension du sujet fait partie de l'examen

Ce sujet a pour but de modéliser et programmer en C++ une version simplifiée du jeu de l'oie.

Le **jeu de l'oie** se compose d'un plateau comportant des cases numérotées de 1 à `tailleMax` (implanté par un tableau dont la taille sera fixée à la construction). Le jeu comprend également un dé (cf. annexe) et une liste de joueurs. Lors de la construction du jeu, des cases de types différents doivent être stockées dans le plateau, au moins une case de chaque type (cf. § suivant). On désire pouvoir :

- Lancer le dé qui retourne le nombre de cases dont le joueur doit avancer.
- Enregistrer un joueur dans le jeu.
- Eliminer un joueur du jeu.
- Coder un opérateur d'indilage qui retourne un pointeur sur la case d'indice `i` du plateau.
- Lancer le jeu en faisant jouer les joueurs à tour de rôle et en retournant le gagnant éventuel : le joueur joue ; s'il est éliminé, alors il est retiré de la liste ; s'il est déclaré gagnant, la partie se termine et le vainqueur est retourné.

Une **case** comporte trois attributs : un nom, le numéro de la case du plateau et une référence sur le jeu. Prévoir un constructeur et une méthode d'affichage du nom et du numéro de la case. L'arrivée d'un joueur sur une case est décrite par la méthode **action** : le joueur effectue l'action associée à cette case, puis un pointeur sur cette case est retourné. L'action associée à une case dépend en fait du type de la case :

- case *simple* : aucune action.
- case Saut de `n` : le joueur avance de `n` cases et effectue l'action de la nouvelle case.
- case Joker : le joueur rejoue.
- case Défaite : le joueur est éliminé du jeu.
- case Victoire : la levée d'une exception retourne le joueur gagnant.

Chaque **joueur** est repéré par une couleur et le numéro de la case du plateau sur laquelle il se trouve. Il dispose également d'une référence sur le jeu. Initialement, le joueur se place sur la première case du plateau. Il dispose des méthodes :

- avancer de `n` cases qui retourne un pointeur sur la nouvelle case obtenue après déplacement.
- jouer : le joueur lance le dé du jeu et avance du nombre de cases indiqué sur le dé. Ensuite, il effectue le traitement associé à l'arrivée sur la nouvelle case. Ce traitement est réalisé par la méthode **action** de la classe `Case`.
- `getJeu` qui renvoie une référence sur le jeu.

1. Donner le code des classes **Case**, **Saut** et **Défaite** .
2. Donner le code de la classe **Joueur**. Coder également un opérateur d'affichage d'un joueur.
3. Donner le code de la classe **JeuOie**. Ecrire un constructeur, un constructeur par copie et un destructeur.
4. On souhaite modifier la fonction jouer de la classe `Joueur` de manière à faire des statistiques sur le nombre de cases Saut rencontrées lors d'une partie. Codez cette nouvelle fonction jouer.
5. Donnez le code d'un programme principal pour lancer le jeu de l'oie limité à quelques cases et à deux joueurs.
6. Le concept de virtualité est-il utilisé dans ce sujet? Expliquez le fonctionnement de ce concept et justifiez votre réponse.
7. Donner le code des classes **Joker** et **Victoire**.

Annexes

```
class De {  
    int valeur;  
public:  
    int lancer() ;  
};
```

Exemple d'utilisation d'une liste d'entiers

```
#include <list>  
using namespace std;  
  
void main ()  
{  
    list<int> l;  
    l.push_back(2); // insertion en fin de liste  
    l.remove(2) ; // suppression de chaque occurrence de la valeur 2 dans la liste  
    // affichage des éléments de la liste  
    for (list<int>::iterator it = l.begin(); it !=l.end(); it++)  
        cout << *it;  
}
```