

M1 Informatique - Ingénierie Logicielle

TD-TP

Séparer l'émission de commandes de leur exécution.

Prétexte

Un exercice classique donné en mathématiques au collège consiste à chercher à obtenir un volume d'eau donné, à partir d'un certain nombre de bidons dont on connaît le volume, et d'une source d'eau (supposée inépuisable). Lors du jeu, on peut remplir un bidon (à ras bord), le vider (complètement), ou le transvaser dans un autre (on met dans le deuxième bidon soit la totalité du volume du premier bidon si c'est possible, soit sinon le volume exact nécessaire pour remplir complètement le deuxième bidon). On suppose toujours être capable d'effectuer ces opérations sans pertes liées à des éclaboussures, ou au fait qu'il reste toujours des gouttes d'eau au fond d'un récipient que l'on vient de vider.

Informatisation

On souhaite réaliser un simulateur de ce jeu, permettant de créer un jeu à n bidons de contenances diverses, dans lequel un joueur peut exécuter les actions précédemment listées afin d'obtenir un volume donné d dans le premier bidon de la partie de jeu en cours (Il sera convenu que le premier bidon d'une partie à une contenance supérieure ou égale à d).

```
1 int[] capacitesBidons = {200, 100, 50};
2 Partiel p =
3     new Partiel(
4         3, //nombre de bidons
5         capacitesBidons, //capacité des bidons
6         150); //volume à atteindre
7 p.jouer();
```

La méthode `jouer()` simule le comportement interactif d'un joueur et les commandes qu'il exécute. On y appelle explicitement les méthodes correspondant aux actions. On pourra obtenir un résultat ressemblant à ce qui est ci-dessous. La suite d'action n'a pas à être optimale pour gagner.

```
1 Gagné avec la solution : [RemplirBidon-2, TransvaserBidon-2-1,
2 RemplirBidon-2, TransvaserBidon-2-1, undo-last, RemplirBidon-3, TransvaserBidon-3-1]
```

Contraintes

On souhaite pouvoir revenir en arrière (ce qui est en pratique impossible hors du contexte de simulation informatique) en défaisant la dernière commande effectuée, et ce plusieurs fois de suite.

On souhaite pouvoir connaître à l'issue d'une partie la séquence d'action qui a été réalisée pour arriver au but (sans compter les retours en arrière bien sûr).

On souhaite pouvoir rejouer dans une nouvelle partie, une séquence d'actions enregistrée lors d'une partie précédente.

Il est à noter que ces contraintes sont extrêmement classiques. Presque tous les logiciels interactifs les satisfont (penser au générique `ctrl-z` "undo").

Vous pourrez vous demander si une architecture logicielle type solutionnant ce problème récurrent a été proposée, et si oui l'étudier.