

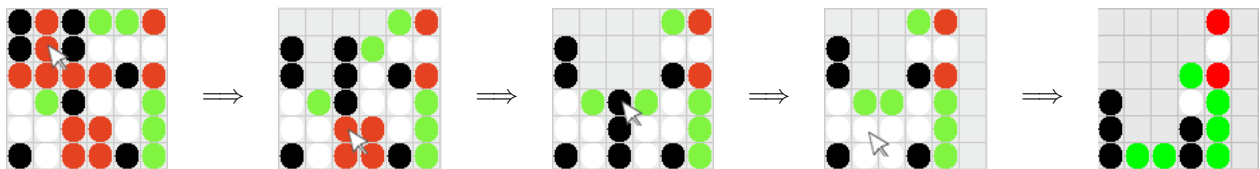
Mini projet JAVA

Master SDI – MSY05

Bubbles

Il s'agit d'un petit jeu simple, dans lequel le joueur doit prendre un maximum de billes de couleur d'un plateau vertical. Le plateau est une grille rectangulaire (de taille configurable), initialement remplie complètement de billes de 4 couleurs différentes aléatoirement choisies (la probabilité de chaque couleur est identique).

Le joueur enlève en fait des groupes de billes contiguës, de même couleur. Il doit retirer des groupes de 3 billes au moins. Plus le groupe comporte de billes, plus le nombre de points gagné est important. Lorsqu'une bille est retirée, les billes situées au dessus descendent. De même, lorsqu'une colonne de billes devient vide, les colonnes situées à sa droite sont décalées vers la gauche (pour combler le vide). L'illustration ci-dessous indique ce qui se passe, sur un plateau de taille 6×6 lorsqu'on retire successivement les billes désignées par le curseur.



On compte $1 + 2 = 3$ points si le groupe pris compte 3 billes, $1 + 2 + 3 = 6$ s'il en compte 4, et ainsi de suite $1 + 2 + \dots + (n - 1)$ s'il en compte n . Ainsi dans l'exemple ci-dessus le score passera successivement de 0 (au départ) à 15 après le premier mouvement, puis 21, 24 et 52.

La première partie concerne la modélisation, dans un second temps vous aurez à implémenter une interface graphique.

Modelisation

Construire un modèle objet pour représenter se problème.

1. On définira tout d'abord une classe `Case` pour représenter chaque case du tableau (ayant au moins un attribut pour représenter la couleur de la bille de cette case). Un constructeur initialisera une case avec une valeur aléatoire, et la méthode `toString` affichera une lettre représentant la couleur de la case. Une autre méthode permettra de changer la couleur de la case. Un second attribut pourra être utilisé pour savoir si la case est occupée ou vide.
2. Ensuite on définira une classe `Plateau`, qui représentera le plateau de jeu. Cette classe pourra, par exemple, avoir un attribut de type tableau à 2 dimensions de `Case`.
3. Implémenter la méthode `toString` de `Plateau`, permettant d'afficher le plateau en utilisant des lettres pour les billes. L'état du jeu proposé ci-dessus après le premier mouvement serait donc comme indiqué ci-contre.
4. Implémenter la méthode `Equals` de `Case` pour tester si 2 cases contiennent la même couleur.
5. Implémenter une méthode `chercheVoisinsCouleur` (`Case c`) renvoyant la liste des cases voisines d'une case donnée, ayant la même couleur que celle-ci.
6. Implémenter une méthode `retireBillesCouleur` (`Case c`) qui, en utilisant la méthode précédente, met à jour le modèle en retirant les billes à retirer, si c'est possible (nombre ≥ 3).
7. Définir une classe `Jeu`, chargée d'initialiser le jeu et de compter les points.
8. Construire une classe de test, utilisant la classe `Console`, pour afficher le plateau, puis demander à l'utilisateur, dans une boucle, de saisir les coordonnées des billes à retirer, mettre à jour le plateau, et le réafficher.
9. (*facultatif*) Implémenter une méthode chargée de déterminer si un mouvement est encore possible (on ne peut retirer de billes que si un groupe de 3 billes contiguës de même couleur est présent).

```
+-----+
|   vr |
| n nvbb |
| n nbnr |
| bvnbbv |
| bbrrbv |
| nbrrnv |
+-----+
```