

Projet : le jeu Cascade

À rendre pour le mardi 15 janvier 2001

Cascade est un jeu de reflexion pour un joueur, avec une grille contenant des billes bleues, noires et rouges (ou +, o et *). Au départ, la grille est pleine, contenant chacune des couleurs en proportion identique, et le but est de la vider, en éliminant successivement toutes les billes.

Comment supprimer les billes. Chaque bille a plusieurs voisines immédiates: en haut, en bas, à gauche et à droite. À chaque étape du jeu, une bille est sélectionnée, et celle-ci entraîne dans sa disparition toutes les billes voisines de même couleur, qui elles-mêmes vont entraîner la disparition de leurs voisines de même couleur et ainsi de suite. Par exemple, considérons la configuration ci-dessous.

	1	2	3	4	5	6	7	8	9
4	*	*	o	o	+	+	+	*	*
3	o	*	+	*	*	o	+	o	+
2	o	*	o	*	*	+	+	+	+
1	+	o	o	*	o	*	+	o	o

Si on joue la bille + de position (2,6), cela provoque l'élimination de 10 billes.

	1	2	3	4	5	6	7	8	9
4	*	*	o	o				*	*
3	o	*	+	*	*	o		o	
2	o	*	o	*	*				
1	+	o	o	*	o	*		o	o

Ensuite les billes restantes roulent du haut vers le bas de manière à combler les positions libérées, puis les colonnes vides sont supprimées. On obtient ainsi la prochaine configuration de la grille, et il ne reste plus qu'à jouer de nouveau.

<table style="border-collapse: collapse; width: 100%;"> <tr><td style="padding: 2px;">*</td><td style="padding: 2px;">*</td><td style="padding: 2px;">o</td><td style="padding: 2px;">o</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;">+</td><td style="padding: 2px;">*</td><td style="padding: 2px;">*</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;">*</td></tr> <tr><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;">*</td><td style="padding: 2px;">o</td><td style="padding: 2px;"></td><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td></tr> <tr><td style="padding: 2px;">+</td><td style="padding: 2px;">o</td><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;"></td><td style="padding: 2px;">o</td><td style="padding: 2px;">o</td></tr> </table>	*	*	o	o						o	*	+	*	*				*	o	*	o	*	*	o		o	*	+	o	o	*	o	*		o	o	descente vers le bas	<table style="border-collapse: collapse; width: 100%;"> <tr><td style="padding: 2px;">*</td><td style="padding: 2px;">*</td><td style="padding: 2px;">o</td><td style="padding: 2px;">o</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;">+</td><td style="padding: 2px;">*</td><td style="padding: 2px;">*</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;">*</td></tr> <tr><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;">*</td><td style="padding: 2px;">o</td><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;"></td></tr> <tr><td style="padding: 2px;">+</td><td style="padding: 2px;">o</td><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;">o</td><td style="padding: 2px;">*</td><td style="padding: 2px;">o</td><td style="padding: 2px;">o</td><td style="padding: 2px;"></td></tr> </table>	*	*	o	o						o	*	+	*	*				*	o	*	o	*	*	o	o	*		+	o	o	*	o	*	o	o		suppression des colonnes vides
*	*	o	o																																																																								
o	*	+	*	*				*																																																																			
o	*	o	*	*	o		o	*																																																																			
+	o	o	*	o	*		o	o																																																																			
*	*	o	o																																																																								
o	*	+	*	*				*																																																																			
o	*	o	*	*	o	o	*																																																																				
+	o	o	*	o	*	o	o																																																																				

Il n'est pas possible de sélectionner une bille isolée, c'est-à-dire une bille dont toutes les voisines sont de couleur différente.

Calcul du score. Le score d'un coup est fonction du nombre de billes éliminées :

- 1 bille → *interdit*
- 2 billes → 3 points
- 3 billes → 6 points
- 4 billes → 10 points
- ...

Autrement dit, éliminer n billes en un coup rapporte $n(n+1)/2$ points. Pour réussir un score élevé, on a donc intérêt à chercher à éliminer le maximum de billes en peu de coups.

Fin de la partie. La partie s'arrête quand plus aucun coup n'est possible : soit la grille est vide, et dans ce cas on bénéficie d'un bonus de **500 points**, soit il reste des billes, mais celles-ci sont isolées.



Exemple de fin de partie : toutes les billes sont isolées

Votre travail

Le projet consiste en la réalisation d'un paquetage `cascade` dont la spécification est la suivante.

```
package Cascade is
  procedure Jouer(Hauteur, Longueur : in Positive);
end Cascade;
```

La procédure `Jouer` doit permettre de mener une partie complète :

- créer une configuration initiale aléatoire pour une grille de `Hauteur × Longueur` positions,
- jouer, coup par coup, en saisissant les coordonnées de la bille sélectionnée, et en affichant la grille mise à jour avec le nouveau score,
- annoncer la fin de la partie.

Pour générer aléatoirement la grille initiale, il faut simuler un tirage au sort sans remise dans un sac. Initialement, le sac contient `Hauteur × Longueur` billes dont un tiers sont bleues, un tiers noires et un tiers rouges (à +1 près si le nombre de positions dans la grille n'est pas divisible par 3), et on y tire les billes une à une. Pour le générateur aléatoire de nombres entiers, vous pouvez utiliser le paquetage disponible à

```
~touzet/ADA/Random
```

Il vous faut également réfléchir aux algorithmes à mettre en œuvre pour détecter toutes les billes appelées à disparaître lors d'un coup, et pour gérer le roulement des billes vers le bas et le décalage de colonnes.

Comment avoir 18/20 au projet ?

La qualité d'un projet dépend des choix de programmation que vous ferez après avoir analysé le sujet.

La première question à se poser est celle de la structure générale du programme, avec une bonne organisation en modules. Ensuite, réfléchissez au choix des structures de données, à l'utilisation d'exceptions, de sous-types ... C'est également une qualité si le programme est résistant aux erreurs de manipulations.

La démarche à suivre pour rendre le projet

- Le projet doit être rendu par courrier électronique à l'adresse `touzet@lif1.fr` Vous devez envoyer chacun des fichiers source en fichiers textes attachés, mais ne joignez-pas l'exécutable !
- Si le projet comporte plusieurs modules, créez un fichier texte `readme.txt` dans lequel vous précisez quelle est la fonction de chacun des modules, ainsi que les dépendances entre modules.
- Si vous travaillez le projet chez vous, compilez-le et testez-le sur les machines de la salle de TP avant de le rendre.
- Toujours par souci de compatibilité avec d'autres systèmes d'exploitation, Linux par exemple, veillez à ce que le nom de vos fichiers soit en *minuscules*. Cette remarque est particulièrement importante si vous avez travaillé le projet sous Windows.
- Le code du projet doit être correctement indenté et commenté (commande `--`) : il faut commenter chaque paquetage, ainsi que les fonctions et procédures à l'intérieur des paquets..
- L'interface du programme doit être explicite et guider l'utilisateur.

Exemple de partie de Cascade

```
* * o o + + + * *  
o * + * * o + o +  
o * o * * + + + +  
+ o o * o * + o o
```

36 billes
Position jouee ? 2 6

```
* * o o  
o * + * * *  
o * o * * o o *  
+ o o * o * o o
```

Score : 55 (reste 26 billes)
Position jouee ? 3 4

```
* * o  
o * + *  
o * o o o *  
+ o o o o * o o
```

Score : 70 (reste 21 billes)
Position jouee ? 1 2

```
*  
o * *  
o * o o o *  
+ * + * o o
```

Score : 85 (reste 16 billes)
Position jouee ? 2 3

```
*  
o *  
o *  
+ * + * * *
```

Score : 100 (reste 11 billes)
Position jouee ? 2 1

```
*  
* *  
+ * + * * *
```

Score : 103 (reste 9 billes)
Position jouee ? 1 2

```
+ + * * *
```

Score : 113 (reste 5 billes)
Position jouee ? 1 2

```
* * *
```

Score : 116 (reste 3 billes)
Position jouee ? 1 1

C'est fini : la grille est vide !
Score final : 622 (36 billes)