

Sur ce second exemple, on peut vérifier que l'algorithme décide bien entre la droite et la gauche pour l'effondrement.

Au départ, les deux colonnes de hauteur 2 sont égales, et la colonne instable est celle du centre. Le programme choisira donc au hasard l'un des deux côtés (voir animation). Cependant, au second tour de la boucle principale, le choix se portera sur le côté qui n'a pas été pris précédemment, car c'est avec celui que la dénivellation est alors la plus forte.

Puis l'effondrement se poursuit naturellement et deux blocs sont éjectés de chaque côté, deux dans chaque « conteneur à carrés éjectés ».

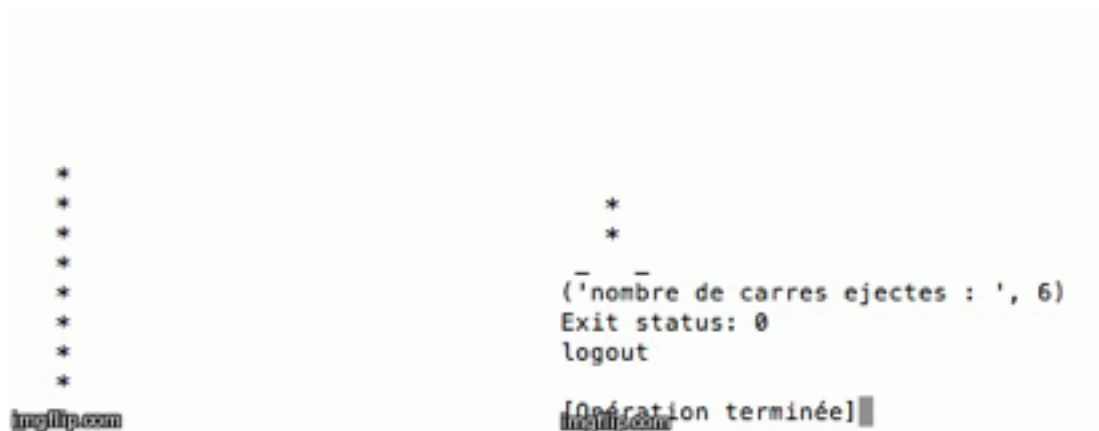
## Troisième configuration :

Cette troisième configuration illustre bien que le programme sait déterminer à quel moment



exactement la montagne est stable, et qu'il ne s'arrête pas avant. Ici, il commence par effondrer deux fois la dernière colonne, puis recherche à nouveau une colonne instable (la première), avant de l'effondrer sur la seconde (voir animation).

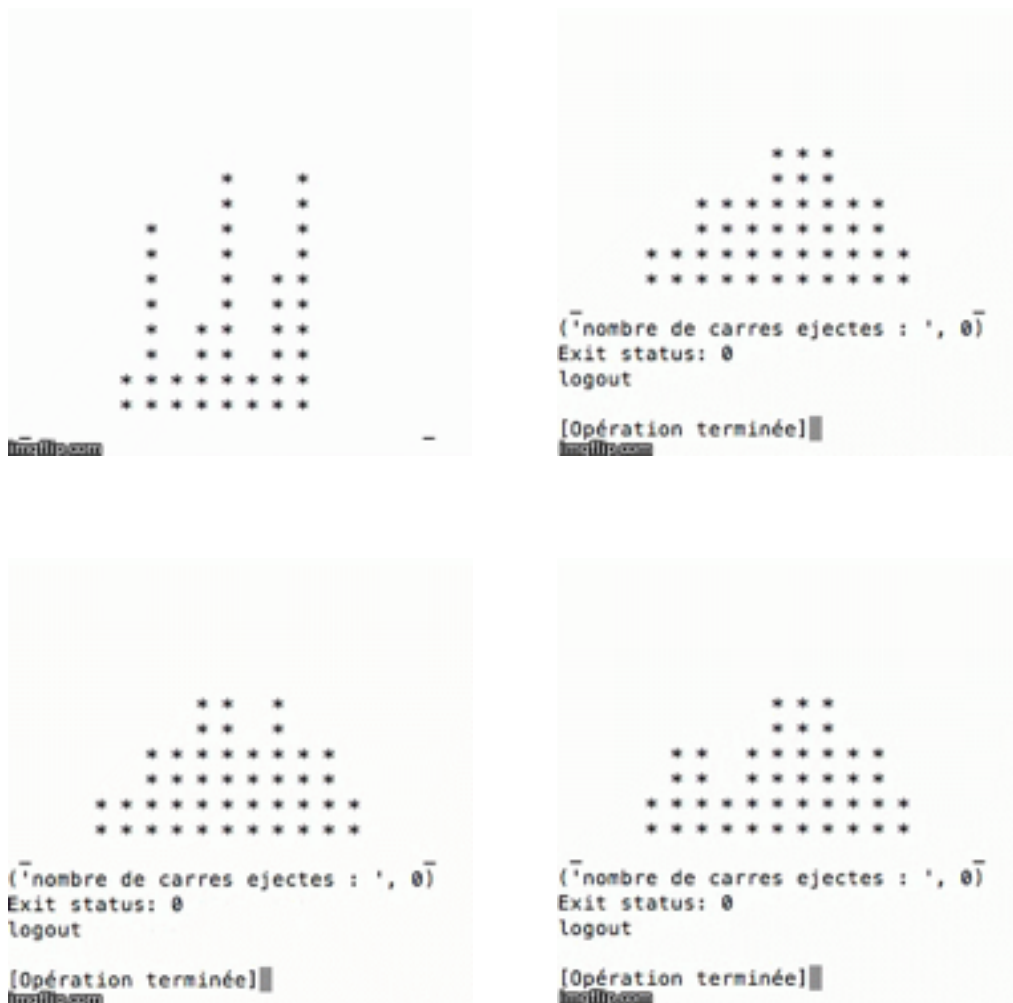
## Quatrième configuration :



Cet exemple est intéressant, dans le sens où il souligne particulièrement une des propriétés du programme. En effet, notre algorithme considère les « conteneurs à cubes éjectés » (symbolisés par des underscore « \_ » ) comme des colonnes de hauteur 0 pour procéder à l'avalanche. Ce qui explique qu'à la fin de l'effondrement de cette montagne-ci, il reste deux cubes : la dénivellation, à gauche comme à droite, n'exède pas deux unités.

Remarque : pour l'affichage, cependant, le programme les considère comme des colonnes de hauteur -1, pour pouvoir les différencier d'éventuelles colonnes vides, et ainsi les écrire avec le underscore « \_ ».

## Cinquième configuration :



Ce dernier exemple, quant à lui, montre en premier lieu que l'algorithme fonctionne sur des montagnes un peu plus grandes et plus complexes. Mais il est surtout intéressant dans le sens où il montre le côté aléatoire de l'algorithme, qui doit choisir au hasard quand les deux colonnes voisines d'une colonne instables sont égales.

Ainsi, en lançant trois fois l'algorithme, nous avons obtenu ces trois configurations finales et stables différentes. Les conteneurs sont éloignés pour n'avoir aucun blocs éjectés, pour pouvoir se concentrer uniquement sur la composante aléatoire du programme.