

# TP n° 10 : programmation d'un démineur

PTSI B Lycée Eiffel

juin 2018

Dans ce TP, nous allons tenter de programmer un démineur rudimentaire (en mode texte, mais les plus motivés pourront ensuite se pencher sur les modules Python permettant de faire un peu de graphisme, représenter une grille rectangulaire n'étant pas encore d'une difficulté insurmontable). La grille de démineur sera représentée par un tableau (donc une liste de listes) de  $n + 2$  lignes et  $p + 2$  colonnes (on ajoute un bors à la « vraie » grille dans lequel on ne mettra jamais de mines, ce qui rend plus facile la programmation ensuite ; bien sûr, on n'affichera pas ce bord vide au joueur quand on lui montrera la grille de jeu).

1. Écrire une fonction **creegrille**( $n, p, k$ ) qui crée une grille de  $n + 2$  lignes et  $p + 2$  colonnes ne contenant initialement que des 0, puis modifier la grille pour faire apparaître  $k$  chiffres 9 à des endroits distincts de la grille, sans modifier les bords (ces chiffres 9 représenteront les mines). On peut bien sûr recourir à certaines fonctions du module random.
2. Écrire une fonction **minesvoisines**(**grille**,  $i, j$ ) qui renvoie le nombre de mines situées sur des cases adjacentes à la case de la ligne  $i$  et de la colonne  $j$  dans le tableau grille. On supposera que la case correspondante n'est pas située sur le bord de la grille (donc elle a nécessairement exactement huit cases adjacentes).
3. En déduire un programme **grillecomplete**(**grille**) qui, à partir d'une grille ne contenant que des 9 aux positions des mines, crée une grille à  $n$  lignes et  $p$  colonnes contenant le nombres de mines adjacentes à chaque case (en gardant bien sûr des chiffres 9 pour les cases contenant des mines).
4. Écrire le programme complet permettant de jouer au démineur, en procédant par exemple de la façon suivante :
  - on demande au jouer de choisir le nombre  $n$  de lignes de la grille, son nombre  $p$  de colonnes, et le nombre  $k$  de mines dans la grille.
  - on crée la grille à l'aide du programme creegrille, puis la grille des mines adjacentes à l'aide de la fonction grillecomplete. Dans le même temps, on crée une autre grille (la grille de jeu du joueur) à  $n$  lignes et  $p$  colonnes ne contenant que des 9 (ce qui indique ici une case n'ayant pas encore été déminée).
  - on crée une variable  $c$  contenant le nombre de cases restant à déminer (qui vaut donc initialement  $np - k$ ).
  - à chaque tour de jeu, on affiche la grille du joueur, puis on lui demande de choisir une case à déminer. Si cette case contient une mine (chiffre 9 dans la grille complète), on arrête le jeu en insultant le joueur. Si cette case a déjà été explorée, on a aussi le droit d'insulter le joueur, mais on ne fait rien. Sinon, on remplace dans la grille du joueur le 9 de la case par le nombre présent dans la grille complète, et on diminue d'une unité la variable  $c$ .
  - si on réussit à atteindre  $c = 0$  sans exploser, on a droit à un message de félicitations.
5. Tenter d'améliorer le programme pour laisser à chaque tour de jeu le choix au joueur entre déminer une nouvelle case ou poser un drapeau sur une case où il pense avoir repéré une mine (on représentera le drapeau dans la case du joueur par un caractère autre que les chiffres déjà utilisés pour le reste de la grille).

6. (plus dur) Améliorer le programme pour que, si le joueur démine une case contenant un 0 (aucune mine sur les cases adjacentes), les cases adjacentes soient déminées automatiquement lors du même tour de jeu (ce que fait le démineur Windows par exemple).
7. Si tout cela était beaucoup trop facile pour vous, il est temps de programmer une version graphique, par exemple à l'aide du module tkinter, voire même Pygame.