

« Au secours ! Je ne trouve pas mon chemin . . . »

Sébastien Combéfis et Chantal Poncin

Afin de vous sensibiliser à l'informatique, et en particulier à l'algorithmique et à la programmation, nous vous proposons ici un problème qu'il vous faudra tenter de résoudre.

Bob est à la recherche de *Barbie Ingénieure en Informatique*. Malheureusement, elle s'est perdue dans un mystérieux pays, peuplé de lapins diaboliques ! Vous allez devoir aider Bob à retrouver Barbie Geek (qu'on va appeler Alice dans la suite du document), en lui donnant des indications quant au parcours qu'il va devoir suivre pour la retrouver, tout en évitant de se faire manger par un méchant lapin. La figure 1 montre un exemple d'un terrain.

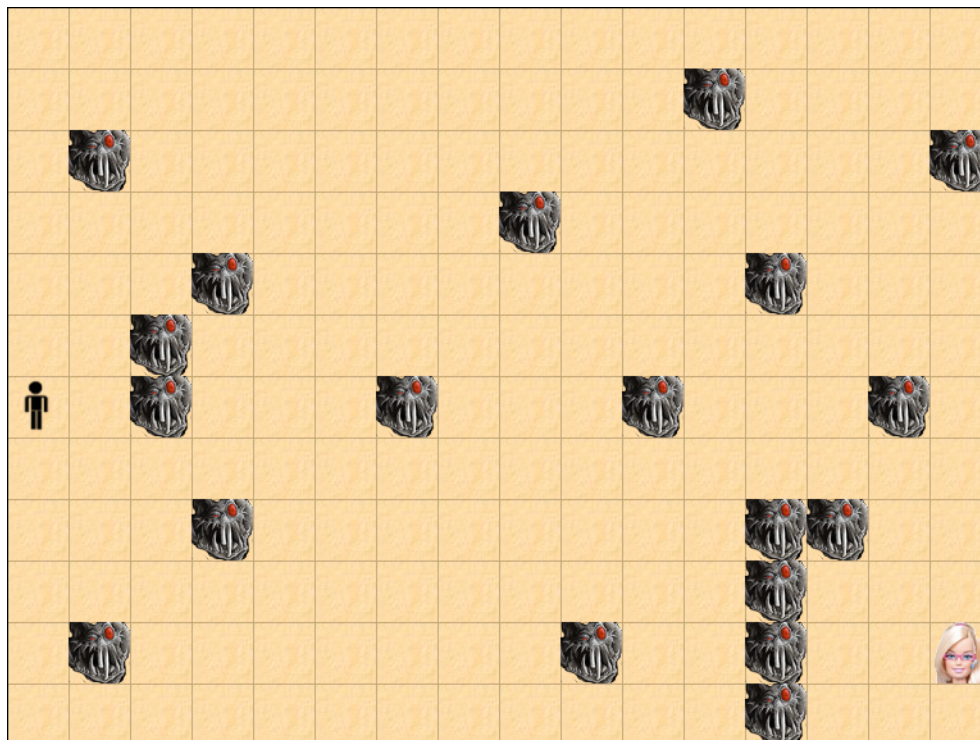
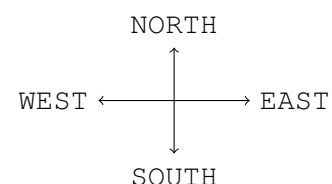


FIGURE 1. Un exemple de terrain.

Bob ne va pouvoir comprendre qu'un certain nombre d'opérations élémentaires, et c'est sur base de celles-ci que vous allez devoir l'aider à retrouver Alice. Tout d'abord, vous pouvez connaître l'orientation de Bob, par rapport aux points cardinaux. Initialement, Bob est dirigé vers l'est.



Vous pouvez également savoir s'il y a un lapin devant Bob, selon sa direction courante. Enfin, vous pouvez savoir si Bob se trouve contre un bord du terrain ou s'il se trouve sur la case contenant Alice. Toutes ces opérations sont des *questions* et permettent d'obtenir une information sur Bob, par rapport à son *environnement*.

On peut également donner des ordres à Bob. On peut lui demander d'avancer d'une case selon sa direction courante. S'il est contre un mur, il ne bouge pas de position. Par contre, s'il se dirige vers une case avec un méchant lapin, il se fait manger et meurt ! On peut également demander à Bob de tourner sur lui-même de 90 degrés, vers sa gauche ou vers sa droite. Ainsi, s'il regarde vers le Nord et que vous lui demander de tourner de 90 degrés vers la gauche, il se retrouve vers l'Ouest. Enfin, vous pouvez lui demander de sauver Alice. Ceci n'aura bien entendu un effet que lorsque Bob se trouve sur la case contenant Alice.

On va noter les différentes opérations qu'il est possible de faire :

- move fait avancer Bob d'une case selon sa direction courante ;
- turnLeft fait tourner Bob sur place de 90 degrés vers sa gauche ;
- turnRight fait tourner Bob sur place de 90 degrés vers sa droite ;
- canMove teste si Bob peut avancer (n'est pas contre un bord) ;
- isFacingRabbit teste si Bob se trouve face à un méchant lapin ;
- isOnAlice teste si Bob se trouve sur Alice ;
- saveAlice sauve Alice ;
- getDirection renvoie la direction courante de Bob (EAST, WEST, SOUTH ou NORTH).

Sur base de ces opérations, on va pouvoir écrire des *algorithmes* qui sont donc une définition rigoureuse d'une procédure à suivre, pour atteindre un but. Commençons avec un algorithme simple qui avance tout droit tant que Bob n'est pas sur la case de la princesse. Une fois sur cette case, il la sauve ! On suppose également qu'il n'y a pas de lapin.

```
TANT QUE PAS Bob isOnAlice FAIRE
|
|   Bob move
|
Bob saveAlice
```

On vous demande de partir d'une situation simple, pour laquelle les lapins sont isolés, et Alice se trouve à la même hauteur que Bob. Réfléchissez à un algorithme qui permet à Bob de retrouver Alice, en évitant tous les méchants lapins.

