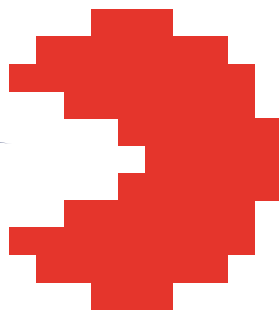


Au Secours!

C'est l'ordinateur qui me guide et les monstres me mangent toujours



Le *code de PacMan* est composé d'une série d'ordres que l'ordinateur va exécuter à chaque étape du jeu pour décider comment déplacer PacMan.

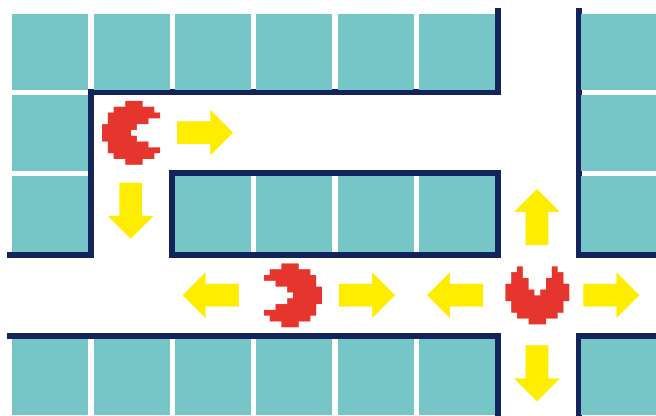
reste

Lorsqu'aucun ordre n'est donné à l'ordinateur, PacMan ne va pas bouger. C'est ce qui se passe au début du jeu.

On peut aussi demander expressément à l'ordinateur de ne pas déplacer PacMan, en donnant l'ordre «reste».

bouge

Pour que PacMan survive, il faut qu'il bouge. Pour cela, on donne à l'ordinateur l'ordre «bouge». Le langage de programmation assure que PacMan ne va pas entrer dans un mur. De cette façon, il y a à chaque pas au moins deux directions possibles. Au plus, dans un croisement complet, il y a quatre directions possibles.



Choisir une direction

Si l'on ordonne à l'ordinateur de bouger, il ne saura pas encore dans quelle direction aller. Faute de mieux, il choisira une direction au hasard.

... vers ...

... telQue ...

Pour que l'ordinateur choisisse une direction, on utilise l'ordre «vers» ou «telQue» — ce sont des synonymes — suivi d'une condition sur la direction à choisir. De toutes les directions vers lesquelles l'ordinateur pouvait bouger, il en choisira une qui satisfait la condition que vous avez ordonnée. S'il n'en existe pas, PacMan ne bougera pas.

Dans ce cas, l'ordre n'est donc plus composé d'un seul mot, mais de trois. C'est le début d'une phrase dans le langage de programmation.

bouge vers àDroite

... àGauche

... enHaut

... enBas

Ce sont là les conditions les plus simples : PacMan ne se déplacera dans une direction **que** s'il s'agit de la direction donnée. Par exemple, si vous ordonnez de «bouge vers àDroite» et qu'il y a un mur à droite, PacMan ne bougera pas.

Les conditions plus avancées sont expliquées sur la prochaine page.

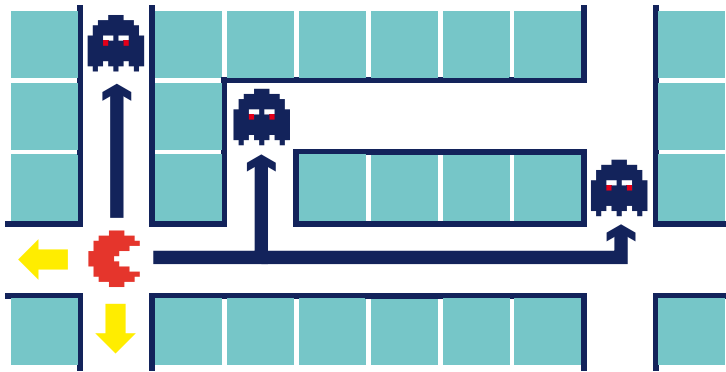
Conditions intelligentes

bouge telQue versUnMonstre

Cette condition choisira, parmi les directions disponibles, celle pour laquelle le chemin vers n'importe quel monstre sera le plus court.

... loinDesMonstres

Celle-ci choisira la direction pour laquelle le chemin vers tous les monstres sera le plus long.



Attention toutefois : cette condition ne prend en compte que la position actuelle des monstres. Quand les monstres bougeront au prochain tour, il se peut qu'ils bouchent de bons chemins et que PacMan se trouve pris au piège.

En mode avancé, d'autres conditions sont disponibles pour trouver les points et les cerises :

... loinDesPoints

... versUnPoint

... loinDesCerises

... versUneCerise

Prévoir

bouge telQue enSécuritéPendant(10)

Cette condition choisira, parmi les directions disponibles, celle pour laquelle PacMan est sûr de survivre sur une certaine distance (10 cases dans l'exemple ci-dessus). Mais que se passe-t-il s'il n'y a pas de direction où l'ordinateur peut être **sûr** que PacMan survivra ? PacMan ne bougera pas.

... sinon bouge ...

En ajoutant un ordre à exécuter si le premier ordre est impossible, on peut faire en sorte que PacMan bouge suivant une autre stratégie quand la première ne fonctionne pas.

La situation change

Des situations différentes demandent des stratégies différentes (par exemple quand PacMan est chasseur en mode avancé). Pour changer de stratégie par rapport à la situation, on utilise des ordres conditionnels (ordre «si ... sinon»).

siChasseur (...) sinon (...)

L'ordinateur exécutera l'ordre de gauche si PacMan est chasseur, et l'ordre de droite autrement.

siChassé (...) sinon (...)

siMonstresLoin (...) sinon (...)

siMonstresPrès (...) sinon (...)

Distances

Dans la condition « en sécurité pendant », on peut donner un nombre de cases. Mais au lieu d'utiliser un nombre fixé, on peut aussi utiliser un nombre calculé par l'ordinateur par rapport à la position de PacMan.

```
distanceVersPoint
```

```
distanceVersCerise
```

```
distanceVersMonstre
```

Ces ordres demandent à l'ordinateur de calculer la distance depuis PacMan vers un point, une cerise ou les monstres, et de se servir de ce nombre pour exécuter l'ordre.

```
si (...) (...) sinon (...)
```

Les distances peuvent aussi servir dans la « condition universelle ». L'ordre juste après le « si » est ici une condition, par exemple de la forme suivante :

```
distanceVersPoint < distanceVersMonstre
```

Ce qui se lit « la distance vers le point le plus proche est moins que la distance vers le monstre le plus proche ».

Chainer les conditions

Si une condition retourne plusieurs directions (par exemple avec l'ordre « en sécurité »), on peut ajouter une condition supplémentaire pour préciser la direction à choisir. Pour cela, il suffit d'ajouter un autre « telQue » ou « vers » après l'ordre. La condition de ce dernier ne s'appliquera pas sur toutes les directions possibles, mais seulement sur celle sélectionnées pas la condition précédente.

```
bouge vers enSécuritéPendant(10)  
telQue versUnPoint
```

.....

En choisissant avec attention les ordres pour l'ordinateur, vous pourrez sauver PacMan. Pour cela, il faut que PacMan survive pendant assez de temps.

Si votre stratégie semble marcher, mais que PacMan se fait manger avant la fin du temps, essayez encore une fois : les monstres ne sont pas très attentifs, et ils ont peut-être eu de la chance cette fois. La prochaine fois, ce sera PacMan et votre stratégie qui auront de la chance.

.....

Si vous avez besoin d'aide, ou si votre code est faux et que vous ne comprenez pas pourquoi, n'hésitez pas à demander à l'un d'entre nous. Nous n'aimons pas voir PacMan se faire manger, et nous sommes heureux de vous aider pour le sauver.