

Construisons un robot !

Nous allons construire un robot (virtuel) qui pourra se déplacer sur une carte et effectuer diverses actions. Pour faciliter le travail, nous allons utiliser les concepts de la programmation orientée objet.

Créer une classe Robot. Le robot aura les caractéristiques suivantes :

- Un nom
- Des points de vie (initialisés à 100 dans le constructeur)
- Une vitesse maximale de déplacement (initialisée à 3 par défaut dans le constructeur)
- Une position, représentée par des coordonnées X et Y (initialisée à (0,0) dans le constructeur)

Une fois la classe créée, nous pouvons générer un ou plusieurs robots basés sur ce modèle. Dans la fonction `main()`, générer un robot en y définissant son nom. Le nom sera appliqué par le constructeur. Les autres caractéristiques prendront les valeurs par défaut (100 points de vie, etc.).

Robot a un ami

Nous venons de créer une classe (Robot) et un objet basé sur cette classe. Nous pouvons en réalité créer une infinité d'objets à partir d'une même classe.

Notre robot se sent un peu seul. Ajoutez un second robot dans le `main()` et donnez-lui un nom différent pour ne pas les mélanger.

Des robots qui parlent

Créer une méthode (fonction de classe) que l'on appellera `sePresenter()`. Cette méthode fait parler le robot dans la console et lui fait dire quelque chose de la forme :

« Bonjour, je m'appelle Terminator. J'ai 100 points de vie. Je me déplace à 3 cases par seconde. Je suis à la case de coordonnées (0 ; 0) »

Une fois la méthode créée, retourner dans le `main()` et demander à chaque robot de se présenter.

Une armée de robots

Qu'est-ce qui est 2 fois mieux que 2 robots ? 4 robots !

Pour bien les ranger en ordre de bataille, nous allons assembler nos robots dans un Array dans le `main()`. Créez 4 robots dans un Array et faites-les se présenter à tour de rôle grâce à une boucle.

A la conquête du monde

Nos robots veulent maintenant se déplacer pour conquérir le monde. Créer une fonction `seDeplacer()` qui permet de se déplacer vers le haut, le bas, la gauche ou la droite sur la carte. En plus de la direction, on doit pouvoir choisir de se déplacer de 1 case au nombre maximum de cases correspondant à la vitesse maximale de déplacement du robot (3 par défaut).

Retourner dans le `main()` et faire déplacer les robots dans toutes les directions au choix. Après les avoir fait déplacer, les faire parler à nouveau pour vérifier qu'ils sont bien dans des cases aux coordonnées différentes.

En mouvement brownien

Nous voulons permettre à nos robots d'effectuer plusieurs tours de déplacement d'affilée.

Créer une fonction `seDeplacerAleatoirement()` qui prend en paramètre un nombre de coups de déplacement. Elle choisira aléatoirement de se déplacer de 1 ou plusieurs cases dans des directions différentes.

Par exemple, si on appelle :

`robot.seDeplacerAleatoirement (4) ;`

... le robot se déplacera aléatoirement 4 fois. Par exemple : une fois vers le haut de 1 case, une fois vers la gauche de 3 cases, une autre fois vers la gauche de 2 cases et une fois vers le bas de 2 cases.

Faire parler à nouveau nos robots pour vérifier qu'ils ont bien bougé.

Des robots uniques

Leur nom ne leur suffit pas. Nos robots aimeraient bien avoir un peu plus de personnalité.

Ajouter un second constructeur qui permet de définir, en plus du nom, les points de vie, la vitesse et la position initiale du robot.

Faire en sorte ensuite que, lors de la création des objets dans le `main()`, ceux-ci aient des caractéristiques différentes de votre choix.

Observer le comportement du `main()` maintenant. Les robots se disent-ils bien placés à des cases différentes de (0 ; 0) au départ ? Vont-ils bien se déplacer à la nouvelle vitesse maximale qu'on leur a définie ?