

Marcel le robot

... une idée farfelue de Marie Duflot-Kremer

créée en avril 2014

dernière modification le 25 août 2017

Le document que vous êtes en train de consulter n'est pas une référence très finalisée, ni un guide strict à suivre. Il regroupe le plan du tapis, des idées de programmes ainsi que quelques idées d'étapes pour réaliser le jeu du robot, appelé ici Marcel, en référence à la première école où cette activité a été testée : l'école primaire spécialisée Marcel Leroy à Nancy. La diffusion de ce document est libre, vous pouvez suggérer des améliorations/enrichissements à marie.duflot-kremer@loria.fr. Si ce document vous a été utile, vous pouvez également me le signaler car j'envisage de mentionner sur ma page médiation les écoles/associations/etc. qui ont testé et approuvé l'activité. La page <https://members.loria.fr/MDuflot/> permet de trouver (section médiation/activités) une liste d'autres activités pour faire découvrir différents aspects de l'informatique, réalisables en grande majorité sans ordinateur.

Étape 1 _____ Découverte du langage

Cette étape peut être réduite ou supprimée en fonction de l'âge des participants. D'expérience elle est appréciée et utile pour les petits (testée principalement en Grande section et CP). Tous les enfants se mettent sur le tapis, chacun sur un point, et suivent les indications données par l'animateur, à choisir entre un pas à droite, à gauche, en avant ou en arrière.

Dès qu'un enfant se retrouve dans le décor (rivière, lac, forêt ou volcan) ou bien sort du tapis, il se met sur le côté. Quand tout le monde est sur le côté on peut recommencer.

Bien préciser :

- qu'un pas c'est se déplacer d'un point, pas deux
- rappeler gauche et droite pour ceux qui pourraient se tromper
- pas de déplacement en diagonale
- on reste toujours dans la même orientation. On fait des pas de côté mais on ne tourne pas sur soi-même
- on est un robot, donc on exécute les instructions sans les changer, même si elles nous envoient dans le décor

Pour la suite de l'activité, on peut aider les enfants ayant des problèmes de latéralisation en surlignant sur les programmes de deux couleurs différentes les flèches correspondant aux pas ou aux quarts de tours à droite et à gauche, puis en mettant des bracelets (un morceau de tissu noué fait l'affaire) de couleur correspondante sur les poignets de l'enfant. La lecture du programme ressemble alors à : "un pas en avant, un pas du côté vert, un pas en arrière, un pas du côté rose, etc."

Étape 2 _____ Exécution de programmes

Des exemples de programmes sont donnés en fin de document, à découper (et plastifier si on veut qu'ils durent plus longtemps).

On met le point de départ comme montré sur le plan (tout en bas, 7ème point à partir de la gauche). Tous les programmes vont démarrer de là.

Pour chaque programme on va répartir la tâche entre un enfant (ou plusieurs) qui va jouer le rôle de la mémoire de l'ordinateur : lire les instructions une par une, et un autre qui va jouer le processeur de

l'ordinateur (ou le robot) et donc exécuter le programme. Il faut bien préciser que les deux jouent le rôle d'un ordinateur, et que donc ils doivent effectuer très précisément les tâches qui leur ont été attribuées. Il n'est pas question de faire un pas de trop ou d'ajouter une flèche en cours de route. L'ordinateur ne fait pas cela.

Dans les exemples les programmes 2 et 4 ont chacun un "bug". Cela permet d'expliquer que si le programme est faux, l'ordinateur va l'exécuter bien sagement et faire consciencieusement un truc faux. Il ne s'est pas trompé, c'est le programmeur qui a fait une erreur. Dans chacun de ces deux programmes, rajouter une seule flèche (vers la droite dans le 2 ou vers le haut dans le 4) permet de corriger le programme. Cela montre bien qu'une toute petite erreur peut entraîner un gros problème (robot noyé), mais que parfois il n'y a pas grand chose à changer pour avoir un programme correct. Cela permet également de mentionner qu'il arrive à tous les informaticiens de faire des erreurs dans leurs programmes (ils sont très long et complexes en général) mais qu'un bon informaticien va tester/vérifier son programme jusqu'à trouver et corriger les bugs.

Étape 3 _____ Changement de jeu d'instructions

Les programmes 1 à 4 permettent de se promener sur le tapis, mais on peut avoir envie d'en faire plus/ de faire les choses différemment.

1. disposer sur le tapis les objets (pierres/feuilles/cailloux) comme montré sur le plan et ajouter une instruction ramasser (ou juste R) qui permet de prendre tous les objets situés sur le point où l'on se trouve,
2. regrouper les flèches identiques consécutives, par exemple écrire $\leftarrow \times 4$ au lieu de $\leftarrow \leftarrow \leftarrow \leftarrow$. Cela évite des erreurs en comptant les flèches et rend les programmes plus compacts (cf programme 5),
3. remplacer les instructions "pas de côté" par des instructions de quart de tour à droite ou à gauche (cf programme 6).

Ces variantes, ainsi que les activités ci-dessous qui en découlent sont à considérer en partie ou en totalité en fonction du temps disponible et de la facilité avec laquelle les enfants ont abordé les étapes précédentes. On peut ensuite :

- réécrire tout ou partie des programmes 1 à 4 (corrigés le cas échéant) avec tout ou partie des modifications ci-dessus.
- trouver comment traduire de manière systématique (tiens, on retrouve un algorithme) tout programme écrit uniquement à l'aide de flèches en un programme écrit avec des flèches et des tours
- voir que le quart de tour à droite et le pas en arrière ne sont plus nécessaires si on a le pas en avant et le quart de tour à gauche (ces deux instructions sont suffisantes pour tout faire). Par contre les programmes avec un jeu d'instruction plus petit sont parfois plus longs (on peut donner des exemples).

Étape 4 _____ Ecriture de programmes

Une fois que les enfants ont apprivoisé le langage de programmation on peut aborder la phase la plus intéressante (en fixant les instructions autorisées ou en les laissant choisir parmi tout ce qu'on a vu). C'est donc à eux d'écrire leurs propres programmes (utiliser une ardoise par exemple, plus facile pour corriger qu'une feuille de papier).

Voici une liste d'exemples de programmes à proposer :

- ramasser trois objets, peu importe lesquels,
- revenir au point de départ avec deux objets dans les mains,
- ramasser trois cailloux et aller au bord de la forêt,
- ramasser un objet de chaque sorte,
- terminer sur un pont avec dans ses mains une fleur, un caillou et une feuille,
- ramasser toutes les feuilles et faire le tour de la forêt,
- ramasser (dans n'importe quel ordre) deux fleurs et deux feuilles,
- ramasser, dans cet ordre, une feuille puis une fleur puis une feuille puis une fleur (et rien d'autre).

En faisant cela on peut remarquer qu'il existe plusieurs programmes possibles pour faire la même chose. On peut si on a le temps et pour des enfants de fin de primaire ou au-delà essayer de comparer deux programmes qui effectuent la même tâche, en termes d'efficacité (en informatique on parle de complexité).

Un programme peut faire le même travail qu'un autre et être plus court à écrire, ou nécessiter moins de pas, ou de tours. Parfois il est assez évident qu'un programme qui va droit au but est plus efficace qu'un autre qui ferait un tour du lac, un tour de la forêt et un tour du volcan avant d'effectuer ce qu'on lui demande. D'autre fois c'est plus difficile de comparer. Entre deux robots, peut-être qu'un avance super vite mais se baisse très lentement. On voudra alors plutôt l'envoyer dans la presqu'île du lac pour ramasser trois objets d'un coup, plutôt que de faire un chemin moins long mais de se baisser trois fois.

De manière générale entre deux programmes corrects on va naturellement préférer le plus efficace. Suivant les cas l'efficacité peut se mesurer en termes de temps/nombre d'étapes à exécuter, ou d'énergie consommée, ou encore d'espace mémoire utilisé.

Programme 1 :

↑ ← ↑ ← ↑ ↑ ↑ ←
← ← ↓ ← ↓ ↓ ↓ →
→ → ↑

Programme 2 :

↑ ↑ ↑ ↑ ← ← ↑ ←
← ↑ ↑ ↑ → → → ↓
→ → ↓ ↓ ↓ ← ← ↓
↓ ↓ ↓

Programme 3 :

↑ ↑ ← ← ↑ ↑ ↑ ←
← ↑ ↑ ↑ ← ← ↑ ↑

Programme 4 :

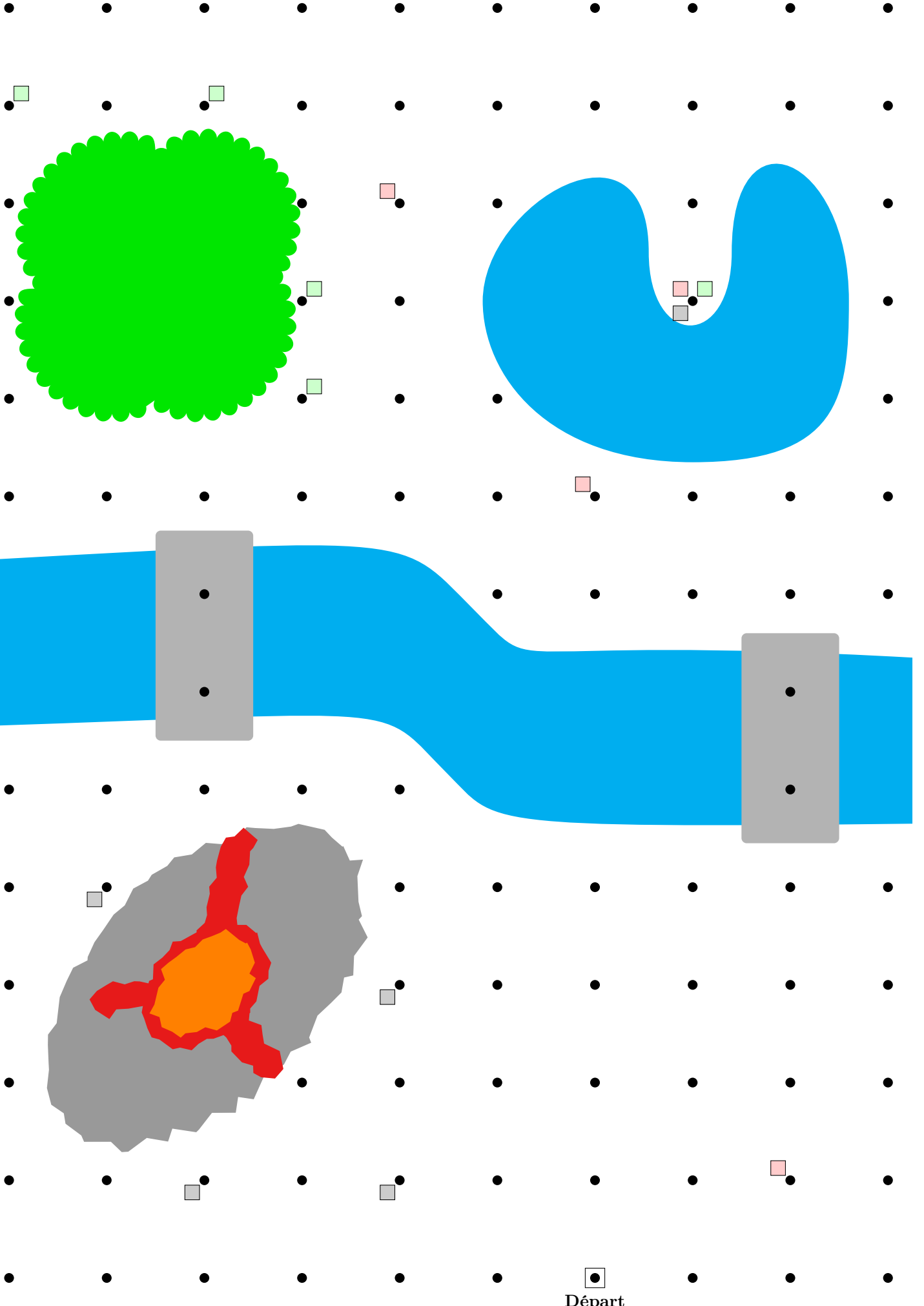
↑ ↑ → ↑ ↑ → ↑ ↑
↑ ↑ ← ← ← ↑ ← ↑
↑ ↑ → → → ↓ ↓

Programme 5 :

→ **x2** ↑ **x8** ← **x4** ↑ **x2** ← **x1** **R**


Programme 6 :

↑ ↷ ↑ **x 2** ↶ ↶ **R**
↑ **x 4** ↷ ↑ **x 2** **R**



 feuille/leaf

 caillou/stone

 fleur/flower


Départ