



CENTRE SCOLAIRE SAINTE-JULIENNE

TA 11 – Jouer au ballon

Exercices Robotprog – Série 8 – Énoncés

Mise en situation

Une institutrice primaire souhaite faire découvrir la programmation robotique à ses élèves, elle t'a demandé de lui fournir les solutions qui répondent à ses besoins.

Objets d'apprentissage

Appliquer	Transférer
<ul style="list-style-type: none">• Utiliser des fonctions prédéfinies (bibliothèque) en vue d'animer un objet réel ou virtuel• Tester la séquence d'instructions conçue• Commenter des lignes de codes	<ul style="list-style-type: none">• Écrire un logigramme d'actions d'un objet réel ou virtuel intégrant structure répétitive et opérateurs logiques• Améliorer une séquence pour répondre à un besoin défini• Corriger une séquence défaillante proposée pour atteindre un but défini
Connaître	
<ul style="list-style-type: none">• Expliquer la notion d'expression• Expliquer la notion d'instruction• Expliquer la notion de séquence• Expliquer la notion de variable• Expliquer la notion de répétition	

Tâches à accomplir

1. Dessiner sur papier l'enchaînement des déplacements;
2. Tester et corriger avec RobotProg votre solution;
3. Enregistrer la solution finale.

Situation de départ

Le robot se dresse dans un environnement avec ou sans murs avec une ou plusieurs prises et pour seules actions possibles: Avancer(), TournerADroite() et Recharger(). De plus, il est possible de tester la position en X et en Y du robot avec les fonctions xRobot() et yRobot().

Pour informer le robot quant à l'existence d'une prise sur la case où il se trouve, la fonction RobotSurUnePrise() renvoie Vrai ou Faux. La fonction Energie() renvoie la réserve d'énergie du robot.

Le ballon apparaît sur le terrain uniquement quand le programme utilise un mot-clé relatif au ballon. Au début de l'exécution, le ballon est placé sur une case déterminée **au hasard**. La position du ballon sur le terrain est donnée par les deux mots-clés **xBallon** et **yBallon**. Ainsi, pour faire apparaître le ballon, il suffit d'écrire $x = xBallon$ et $y = yBallon$.

On peut tester si le robot est dans la même case que le ballon en utilisant la condition: $xRobot = xBallon$ et $yRobot = yBallon$. La condition logique **BallonSurLeSol** est vraie si le ballon est sur le sol.

Les commandes du robot relatives au ballon sont: PrendreBallon, PoserBallon et LancerBallon. Ces commandes doivent être écrites dans des blocs de commande éditables.

PrendreBallon: le robot prend le ballon s'il se trouve sur la même case que le ballon.

PoserBallon: le robot pose le ballon sur la case qu'il occupe.

LancerBallon: le robot lance le ballon trois cases devant lui. Si le ballon sort du terrain, il est relancé automatiquement sur une case déterminée au hasard.

Le plateau de jeu comporte au minimum 10 lignes et 10 colonnes.

Ex 1

Le robot se trouve n'importe où sur un plateau sans obstacles.

Le robot doit se déplacer vers le ballon.

Le programme principal va faire apparaître le ballon et appeler le sous-programme

AllerALaCaseXY tant qu'il n'a pas atteint sa destination.

Ce sous-programme appelle lui-même un sous-programme **AllerVersXY** qui va orienter le robot sur l'axe des X et l'axe des Y.

Pour orienter le robot, il est possible de tester:

dxRobot: indique l'orientation sur l'axe des X:

Les valeurs possibles sont:

- -1: le robot se déplace de la droite vers la gauche;
- 0: Le robot est bien positionné sur l'axe des X;
- 1 : Le robot se déplace de la gauche vers la droite.

dyRobot: indique l'orientation sur l'axe des Y:

Les valeurs possibles sont:

- -1: le robot se déplace du bas vers le haut;
- 0: Le robot est bien positionné sur l'axe des X;
- 1 : Le robot se déplace du haut vers le bas.

Ex 2

Le robot doit aller chercher le ballon comme dans l'ex 1, prendre le ballon, l'apporter aux coordonnées (1,1) et le déposer.

Il faut prévoir un sous-programme **AllerChercherBallon** qui reprend la solution de l'ex1 et un sous-programme **AllerDeposerBallon** qui va déposer le ballon aux coordonnées renseignées.

Le plateau contient quelques prises et le robot dispose au début d'une énergie de 350.

Ex 3

Il faut créer un nouveau terrain (modifier terrain puis nouveau terrain) a enregistré sous le nom basket.

Le robot doit aller chercher le ballon, aller se positionner devant le panier le plus proche et lancer le ballon.

Il faut reprendre la logique de l'ex 2 en remplaçant AllerDeposerBallon par **AllerLancerBallon**.

Le sous-programme AllerLancerBallon va appeler le sous-programme **ChercherPanierPlusProche**, puis le sous-programme **AllerALaCaseXY** et pour finir le sous-programme **RegarderPanier**. ChercherPanierPlusProche va renseigner le panier le plus proche du Robot en initialisant les variables X et Y.

La position des paniers est la suivante :

- Panier du haut=(6,1);
- Panier du bas=(6,11);
- Panier de Gauche=(1,6);
- Panier de Droite=(11,6).

Ne pas oublier de tenir compte des 3 cases nécessaires pour que le robot effectue son tir!

RegarderPanier va réorienter le robot vers le panier qui se tient devant lui. Il faut le faire tourner à droite tant qu'il n'a pas atteint la bonne orientation.

Ex4

Enregistrer une copie de l'ex3 dans un projet, préparer le terrain basket pour la réception d'un deuxième robot placé également aléatoirement et lancer le programme.

