

Note sur SPY

Fonctionnalités :

Liste des fonctionnalités développées :

1. Restreindre les blocs que l'on peut utiliser dans un niveau / limiter le nombre de bloc disponible
2. Désactiver le Drag&Drop
3. Initialiser une séquence complète, incomplète, erronée ou non optimale pour un robot, le joueur doit la compléter, corriger ou optimiser.
4. Résoudre un niveau en plusieurs exécutions / limiter le nombre d'exécution possible pour terminer un niveau.
5. Utiliser des Blocs actions (présent dans l'inventaire ou dans un script pré-construit)
 - 5.1. Bloc Avancer
 - 5.2. Bloc Pivoter à gauche
 - 5.3. Bloc Pivoter à droite
 - 5.4. Bloc Faire un demi-tour
 - 5.5. Bloc Attendre
 - 5.6. Bloc Activer
6. Utiliser des structures de contrôle (présent dans l'inventaire ou dans un script pré-construit)
 - 6.1. Utiliser la structure de contrôle boucle « For »
 - 6.2. Utiliser la structure de contrôle « While »
 - 6.3. Utiliser la structure de contrôle « If »
 - 6.4. Utiliser la structure de contrôle « If... Else... »
7. Utiliser des capteurs renvoyant une valeur Vrai/Faux (présent dans l'inventaire ou dans un script pré-construit)
8. Combiner les capteurs à l'aide d'opérateurs « Et », « Ou », « Non » (présent dans l'inventaire ou dans un script pré-construit)
9. Pouvoir renommer le nom d'un script afin de le lier à un robot.
10. Définir plusieurs colonnes de programmation
11. Avoir plusieurs robots contrôlés par le même programme
12. Pouvoir ouvrir/fermer des portes (ie avoir une porte connectée à un terminal)
13. Définir un programme pour un ennemi et pouvoir le consulter
14. Zone de détection à éviter
15. Brouillard

Compétences PIAF en jeu :

C1.1 – Nommer des objets et séquences d'actions : Être capable de donner des noms à des objets, des actions et des séquences d'actions

F9 : permet de nommer de manière approprié un programme afin qu'il contrôle le bon robot

C1.2 – Différencier (i) objet et action, et (ii) actions atomiques et non-atomiques

F5.4 et (F5.2 ou F5.3) : Le bloc « Faire demi-tour » est une action non-atomique qui peut être décomposée par les actions atomiques « Pivoter à gauche » ou « Pivoter à droite ».

F5 (actions) et F7 (capteurs) : les actions et les capteurs peuvent être considérés comme des métaphores des concepts informatiques d'expression et d'instruction. L'expression (action) n'a pas de valeur et est représentée par des verbes, l'instruction (capteur) donne une information sur le monde.

C1.3 – Identifier les paramètres d'entrée d'une séquence d'actions

Non traité dans SPY

C1.4 – Décrire le résultat d'une séquence d'actions

Non traité dans SPY

C1.5 – Prédire le résultat d'une séquence d'actions

F13 et F14 : Le programme d'un ennemi ne peut être exécuté seul, le joueur doit donc se représenter les conséquences de la séquence d'action pour pouvoir agir.

C1.6 – Utiliser des objets dont la valeur peut changer

F7 et (F6.3 ou F6.4 ou F6.2) et F5 : La valeur des capteurs change en fonction du contexte

F5.6 et F12 : la manipulation des portes permet d'utiliser des actions qui changent l'état (ouvert/fermé) d'un objet du jeu

Note : Cette compétence correspond normalement à la notion de variable et aux notions de déclaration et d'affectation. Avec ces propositions on ne traite que très partiellement cette compétence

C1.7 – Reconnaître, parmi des objets et séquences d'actions connus, lesquels peuvent être utilisés pour atteindre un nouvel objectif

F2 et F3 et F9 : Proposer plusieurs séquences d'action qu'on ne peut pas éditer (Drag&Drop désactivé). Une seule bonne la solution. Le joueur doit les comprendre et nommer la bonne pour l'envoyer au robot. Extension possible avec plusieurs robots, sur le même canal de communication ou pas. Contrairement à C4.2, ici un seul programme par robot est bon (utiliser les métas données pour les différencier).

C2.1 – Ordonner une séquence d'actions pour atteindre un objectif

F1 et F5 : Ne donner au joueur que les blocs utiles (et en bonne quantité) à la résolution d'un problème lui permet de se concentrer uniquement sur une tâche d'ordonnancement. L'apprenant n'a pas besoin d'identifier toutes les parties nécessaires pour atteindre l'objectif, mais seulement de les mettre dans le bon ordre.

C2.2 – Compléter une séquence d'actions pour atteindre un objectif simple

F3 et F5 et non F6 : Pour identifier C2.2 dans le niveau il faudra noter F3 comme une séquence buggée pour savoir que la séquence donnée est une séquence incomplète. L'absence de F6 permet de rester sur des objectifs simples.

C2.3 – Créer une séquence d'actions pour atteindre un objectif simple

F5 et non F6 : Le joueur ne manipule que des actions simples. L'absence de F6 permet de rester sur des objectifs simples.

C2.4 – Créer une séquence d'actions pour atteindre un objectif complexe

F5 et F6 : Le joueur combine des actions simples avec des structures de contrôle

C2.5 – Combiner des séquences d'actions pour atteindre un objectif

Non traité dans SPY

C2.6 – Décomposer des objectifs en sous-objectifs plus simples

C'est un principe de base de SPY de pouvoir résoudre un niveau en sous étapes sauf si F4 est utilisé avec une limite de 1 auquel cas on souhaite que le joueur résolve le niveau en une seule étape (pas de décomposition possible).

C3.1 – Répéter une séquence d'actions un nombre donné de fois

F6.1 : Etre capable d'utiliser de manière appropriée les répétitions d'actions un certain nombre de fois afin d'atteindre un objectif

C3.2 – Répéter une séquence d'actions jusqu'à ce qu'un objectif soit atteint

F6.2 et F7 : Etre capable de contrôler les répétitions d'actions jusqu'à ce qu'une condition ne soit plus remplie afin d'atteindre un objectif

C3.3 – Intégrer une condition simple dans une séquence d'actions

(F6.3 ou F6.4) et F7 : Etre capable d'utiliser des conditions simples (reposant sur un capteur) pour permettre ou interdire certaines séquences d'actions

C3.4 - Intégrer une condition complexe dans une séquence d'actions

F8 et F7 et (F6.3 ou F6.4 ou F6.2) : Être capable d'utiliser des conditions complexes (c'est-à-dire reposant sur plusieurs capteurs, qui sont combinés au moyen des opérateurs logiques « Et », « Ou », « Non ») pour permettre (ou interdire) certaines séquences d'actions.

C4.1 – Comparer deux objets selon un critère donné

Non traité dans SPY

C4.2 – Comparer deux séquences d'actions selon un critère donné

F2 et F3 et F9 et F10 : Proposer plusieurs séquences d'actions qui ne peuvent être éditées (Drag&Drop désactivé) et qui fonctionnent pour résoudre un niveau et donner un critère dans la consigne. Le joueur doit sélectionner la bonne séquence d'action en la nommant pour l'envoyer sur le robot en vue de valider (ou pas) son choix. Contrairement à C1.7 tout programme proposé résout la situation (utiliser les métas donnés pour les différencier).

Note : comment évaluer le critère ?

C4.3 – Améliorer une séquence d'actions par rapport à un critère donné

F3 : Pour identifier C4.3 dans le niveau il faudra noter F3 comme nonOptimal pour savoir que la séquence donnée est non optimale. Dans la consigne demander de trouver un chemin avec le moins de virages possibles.

Note : comment évaluer le critère ?

C5.1 – Représenter des objets ou séquences d'actions au moyen d'une représentation formelle

Non traité dans SPY

C5.2 – Traduire des objets ou séquences d'actions entre représentations formelles

Non traité dans SPY

C6.1 – Vérifier si une séquence d'actions atteint un objectif donné

C'est un principe de base de SPY, vérifier par l'exécution que la séquence d'action a bien permis d'atteindre un objectif donné

C6.2 – Repérer des erreurs dans une séquence d'actions

Non traité dans SPY (*C'est une compétence sous-jacente mobilisée par le joueur pour déboguer ses programmes, mais on n'a pas de fonctionnalité permettant explicitement de pointer l'erreur*).

C6.3 – Corriger une séquence d'actions pour atteindre un objectif donné

F3 : Pour identifier C6.3 dans le niveau il faudra noter F3 comme bugée pour savoir que la séquence donnée est erronée.

C6.4 – Étendre ou modifier une séquence d'actions pour atteindre un nouvel objectif

Possible dans une scénarisation de niveau en exploitant F3, on peut donner le même programme qu'au niveau précédent et demander de l'adapter à la nouvelle situation.

F11 et F3 : donner une ligne de programme pour contrôler deux robots, le script pré rempli permet à un robot d'atteindre la sortie (c'est une solution pour un objectif donné). Le joueur doit modifier le script pour que les deux robots atteignent la sortie.