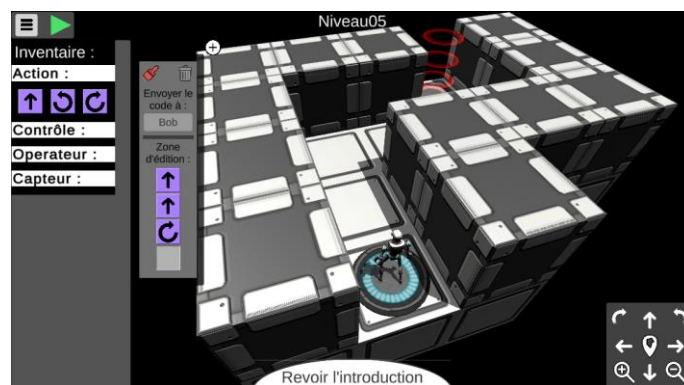


SPY

Fiche d'identité du projet

SPY c'est quoi ?

SPY est un jeu sérieux sur la pensée informatique conçu pour des élèves de cycle 3. Il est gratuit et libre d'utilisation. Le joueur doit apprendre à programmer un robot pour l'aider à atteindre la sortie d'un labyrinthe.



De quoi ai-je besoin pour jouer à SPY ?

Un simple ordinateur avec un navigateur internet : <https://webia.lip6.fr/~muratetm/SPY/>

Si vous avez une tablette récente vous pouvez aussi y jouer en tactile.

Il est fait par qui ?

SPY est développé par une équipe de recherche publique. L'équipe [MOCAH](#) du LIP6 (Sorbonne Université / CNRS). Il est développé à des fins de recherches en vue de fournir une ressource utile aux élèves de cycle 3 et à leurs enseignants.

Quel est la licence du jeu ?

SPY est sous licence GPL3... Ça veut dire quoi ? Que son code source est libre et ouvert. Tout ce passe à l'adresse suivante : <https://github.com/Mocahteam/SPY>

Présentation détaillée du jeu

Table des matières

1	Les principes de jeu	2
1.1	Blocs de programmation	2
1.2	Zone d'édition	3
1.3	Obstacles	4
1.4	Contrôle de l'exécution des programmes	4
1.5	Nombre d'exécutions	4
1.6	Brouillard et texte d'introduction	4
2	Les scénarios.....	5

1 Les principes de jeu

1.1 Blocs de programmation

Les blocs de programmation sont les éléments fondamentaux de SPY dans la mesure où ils constituent les briques de base composant les programmes exécutés par les robots. Les blocs de programmation sont répartis en quatre catégories :

Action :



Les blocs d'action : Ces blocs permettent de définir les actions pouvant être réalisées par les robots à savoir : « Avancer », « Pivoter à gauche », « Pivoter à droite », « Faire demi-tour », « Activer un terminal », et « Attendre ». A noter que toutes ces actions sont atomiques à l'exception de l'action « Faire demi-tour » qui peut être décomposée par deux actions « Pivoter à droite » ou deux actions « Pivoter à gauche ».

Contrôle :



Les blocs de contrôle : Ces blocs permettent de contrôler les blocs d'action à exécuter à savoir : « Si Alors », « Si Alors Sinon », « Répéter n fois », « Tant que » et « Répéter indéfiniment ».

Capteur :



Les capteurs : Ces blocs donnent des informations sur l'environnement avoisinant les robots. Les capteurs renvoient des valeurs booléennes qui peuvent être exploitées dans les blocs de contrôle « Si Alors », « Si Alors Sinon » et « Tant que ». Les capteurs permettent au robot de savoir si un mur se trouve en face de lui, à sa gauche ou à sa droite ; si un passage se trouve en face de lui, à sa gauche ou à sa droite ; si une sentinelle, une zone surveillée ou une porte se trouve en face de lui ; et si un terminal ou une sortie se trouve sur sa position.

Opérateur :



Les opérateurs : Ces blocs permettent de combiner les capteurs. Nous retrouvons les classiques opérateurs booléens : « Non », « Ou » et « Et ».

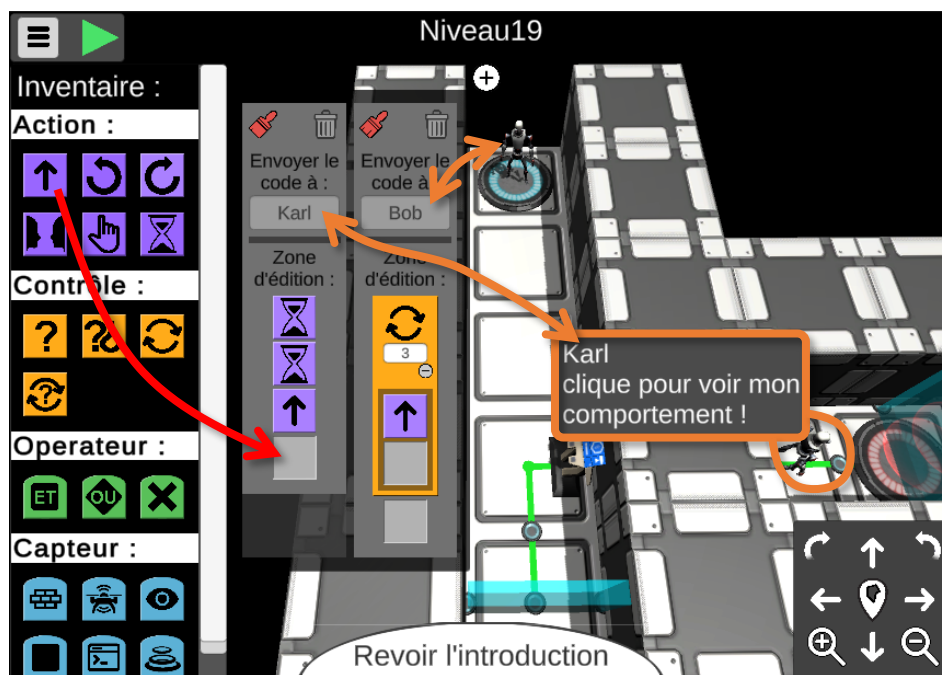
Limitation des blocs de programmation : SPY donne la possibilité de paramétrer pour chaque niveau quels blocs de programmation seront disponibles et en quelle quantité. Cette fonctionnalité peut être utile à la fois pour réduire la complexité d'un niveau ou au contraire l'augmenter. Par exemple, un niveau d'introduction pourra ne contenir que les blocs utiles à sa résolution, évitant ainsi au joueur de devoir choisir parmi des blocs potentiellement inutiles. Inversement limiter l'accès à certains blocs (ou leur quantité) peut être utile pour forcer le joueur à utiliser certains blocs (par exemple, demander de faire avancer un robot de plusieurs cases en limitant le nombre de bloc « Avancer » à 1 et un donnant accès au bloc de contrôle « Répéter n fois »).



1.2 Zone d'édition

La zone d'édition est la seconde fonctionnalité fondamentale de SPY. Elle accueille les blocs de programmation permettant de construire les solutions aux problèmes posés.

Par défaut, chaque robot est associé à une zone d'édition. Il est cependant possible de rompre cette association afin de demander au joueur de le faire. Dans ce cas, le joueur doit indiquer dans la zone d'édition le nom du robot auquel elle est associée afin que le programme lui soit envoyé lors du lancement de l'exécution.



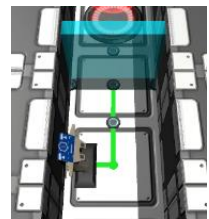
Une zone de programmation peut être préconstruite. Le programme ainsi proposé au joueur peut être complet, partiel ou bogué. Le joueur devra en conséquence la compléter ou la corriger le cas échéant.

Enfin, SPY offre la possibilité de ne proposer au joueur qu'une seule zone de programmation pour plusieurs robots. Ceci permet de construire des niveaux où le joueur devra trouver une solution fonctionnelle non pas pour un robot mais pour deux robots (ou plus) pouvant être dans des labyrinthes différents. Le joueur pourra ainsi se confronter à la réalisation de solutions plus génériques.

1.3 Obstacles

Pour résoudre les différents niveaux du jeu, le joueur doit programmer un ou plusieurs robots pour les aider à atteindre une position particulière tout en évitant des obstacles. Ces obstacles ont été conçus avec une intention didactique :

- Des sentinelles surveillent des zones du labyrinthe. Si un robot se trouve sur l'une de ces zones surveillées, le robot est détecté ce qui met fin à la partie. Le joueur doit donc programmer le robot pour qu'il évite ces zones surveillées. Par ailleurs, comme pour les robots, les sentinelles peuvent contenir un programme préconstruit ce qui rend les zones surveillées dynamiques. Le joueur peut donc sélectionner chaque sentinelle du jeu, observer le programme qui la compose, comprendre cette séquence d'action, anticiper les mouvements de la sentinelle et programmer son robot en conséquence.
- Des portes peuvent être ouvertes et fermées à l'aide de terminaux. Cette mécanique a été introduite pour engager un processus de résolution en étapes. Lorsqu'une porte bloque le passage, le joueur doit décomposer sa solution en sous-étapes (objectif 1 : activer le terminal pour ouvrir la porte ; et objectif 2 : atteindre la sortie). Les portes permettent également de manipuler des objets dont l'état peut changer (ouvert ou fermé).



1.4 Contrôle de l'exécution des programmes

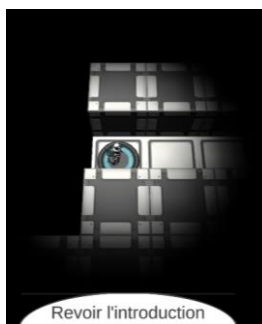
Lorsque le joueur souhaite tester sa solution il clique sur un Play ► pour lancer la simulation. Chaque zone de programmation envoie son programme au robot ciblé s'il existe et chaque robot/sentinelle exécute ses actions en parallèle. Le joueur peut suivre l'exécution des programmes en observant les robots/sentinelles bouger dans la scène en fonction des actions en cours d'exécution. Le joueur peut mettre en pause la simulation ⏸ à tout moment et exécuter les programmes pas à pas. ▶

1.5 Nombre d'exécutions

Trouver la solution à un problème en un seul coup (trouver un programme qui permet de déplacer le robot directement du point de départ au point d'arrivée) peut être une tâche complexe notamment sur des niveaux contenant des sentinelles en mouvement. SPY permet donc de résoudre un niveau en plusieurs coups. Le joueur peut définir une première séquence de blocs, l'exécuter, observer la nouvelle situation, ajouter de nouveaux blocs, exécuter, observer... Il est donc possible de construire le programme solution étape par étape.

Cependant, il peut être pertinent de limiter ce nombre d'exécutions pour amener progressivement le joueur à anticiper plusieurs actions en avance. Il est donc possible pour chaque niveau de SPY de définir le nombre autorisé d'exécutions pour le résoudre. ⏸

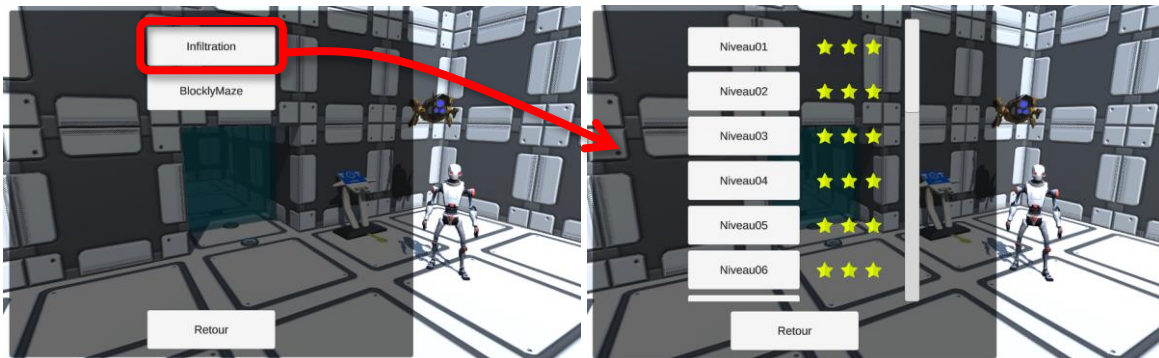
1.6 Brouillard et texte d'introduction



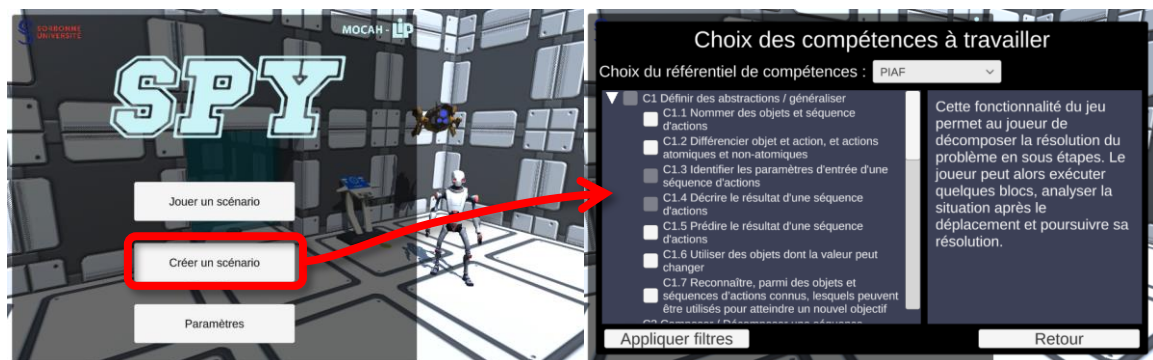
Dans un niveau classique de SPY le joueur a une vue omnisciente de la situation de jeu ce qui lui permet de planifier ses actions en vue d'atteindre l'objectif du niveau décrit dans les textes d'introduction. Le brouillard permet de modifier cette règle en cachant la vue d'un robot à son entourage proche. Dans ce cas, le texte d'introduction présentant l'objectif de la mission joue un rôle fondamental, car il devra contenir des indices permettant au joueur de trouver la solution. Par exemple, l'algorithme peut être donné dans le texte d'introduction en langage naturel et le joueur doit le traduire à l'aide du langage formel du jeu.

2 Les scénarios

SPY propose deux scénarios. Le premier intitulé « Infiltration » est un scénario original composé de 20 niveaux. Le second « BlocklyMaze » est une reprise du jeu original du même nom¹.



Un éditeur de scénario est intégré au jeu qui permet de composer son propre scénario à partir des niveaux intégrés à la base de donnée. Les niveaux peuvent être filtrés par compétences. Actuellement le référentiel PIAF² est intégré au jeu.



¹ <https://blockly.games/maze>

² <https://piaf.loria.fr/>