

# Scénario pédagogique sur la PIA

## Pensée informatique et algorithmique

# <u>Titre</u>: Suis et construis des algorithmes<sup>1</sup>

# **Informations pratiques**

#### Fonctionnement:

En ateliers

### Nombre (idéal) d'apprenants :

- 4 par atelier
- 5 ateliers de 4 élèves dont un de robotique sont mis en place en classe

### Âge des apprenants :

• De 10 à 12 ans

#### Durée du scénario:

4 séquences de 50 minutes

### Discipline principale du scénario :

• Pensée informatique dans le fondamental

# **Description**

L'activité vise, au moyen de situations de manipulation du logiciel ainsi que d'observation du robot, à amener les élèves à programmer les déplacements du robot qu'ils ont observés et à découvrir les caractéristiques d'un algorithme.

Au terme des activités, les élèves seront capables de rédiger un algorithme qui correspond à une suite d'actions précises effectuées par le robot, ainsi que de programmer cette suite d'actions avec le logiciel Lego Spike.

<sup>&</sup>lt;sup>1</sup> Scénario inspiré du scénario créé collaborativement avec Elaimrany, K. et Angenot, L.. (2019).















# Compétences/Objectifs spécifiques à la PIA

### Compétences spécifiques à la PIA (cf. référentiel PIAF<sup>2</sup>)

### 5.1. **5. Manipuler des représentations formelles**

- → 5.1. Représenter des objets ou séquences d'actions au moyen d'une représentation formelle
- ⇒ Observation : représenter les actions observées (les mouvements/actions d'un robot qui sont observées) à une représentation textuelle.
  - ⇒ L'élève sera capable, <u>suite à ses observations</u>, <u>de rédiger l'algorithme effectué</u> <u>par le robot</u>.

### 5.2. **5. Manipuler des représentations formelles**

- → 5.2. Traduire des objets ou séquences d'actions entre représentations formelles
- ⇒ Programmation : traduire une séquence d'actions d'une représentation textuelle en langage de programmation (texte → blocs d'action).

L'élève sera capable de <u>programmer les déplacements du robot en partant d'algorithmes écrits.</u>

### 4.2. 4. Évaluer des objets ou des séquences d'actions

- → 4.2. Comparer deux séquences d'actions selon un critère donné
- ⇒ Comparaison du langage de deux séquences d'actions selon un critère.
  - ⇒ L'élève sera capable de comparer le langage de différentes séquences d'actions rédigées selon le critère donné (la précision du langage et la réduction du nombre de commandes).

Cette compétence n'est pas évaluée.

#### 3.3. **3. Contrôler une séquence d'actions**

- → 3.3. Intégrer une condition simple dans une séquence d'actions
- ⇒ L'élève sera capable d'utiliser une condition simple dans la rédaction d'un algorithme et d'un programme.

#### 3.1. **3. Contrôler une séquence d'actions**

→ 3.1. Répéter une séquence d'actions un nombre donné de fois

<sup>&</sup>lt;sup>2</sup> Les compétences présentées dans ce tableau proviennent du référentiel créé dans le cadre du projet Erasmus+ PIAF (disponible à l'adresse suivante : <a href="https://piaf.loria.fr/contributions/">https://piaf.loria.fr/contributions/</a>). Celles-ci font d'ailleurs référence aux numéros des compétences du référentiel (par. ex. : 5. Traduire d'une représentation à une autre).













- ⇒ L'élève sera capable <u>d'utiliser la boucle dans la rédaction d'un algorithme et d'un</u> programme.
- 4.3 4. Évaluer des objets ou des séquences d'actions
  - → 4.3 Améliorer une séquence d'actions selon un critère donné.

Cette compétence n'est pas évaluée.

- 6.1 **6.** Construire une séquence d'actions de manière itérative
  - → 6.1. Vérifier si une séquence d'actions atteint un objectif donné
  - ⇒ Tester un programme avec le logiciel Lego Spike.
    - ⇒ L'élève sera capable de tester son programme sur le logiciel Spike.
- 6.2. **6.** Construire une séquence d'actions de manière itérative
  - → 6.2. Repérer des erreurs dans une séquence d'actions
- 6.3 **6.** Construire une séquence d'actions de manière itérative
  - → 6.3. Corriger une séquence d'actions par rapport à un objectif donné

L'élève sera capable <u>de corriger le programme sur le logiciel Lego Spike ou l'algorithme</u> rédigé afin qu'ils correspondent aux mouvements du robot de l'enseignant.

# <u>Prérequis à l'activité</u>

- Conseil: une série d'activités de prise en main relative à la construction de robots avec le kit Lego Spike Prime (briques, moteurs et capteurs) et l'interface du logiciel Spike Education basé sur Scratch.
- Former les élèves à utiliser la section d'aide du logiciel Spike Education.











# Ressources numériques

# Matériel technique et didactique

Technique	Didactique
Deux ordinateurs sur lesquels est installé le logiciel pour programmer le robot Lego Spike ( <a href="https://education.lego.com/en-us/downloads/spike-prime/software">https://education.lego.com/en-us/downloads/spike-prime/software</a> )	<ul> <li>⇒ Feuilles sur lesquelles les élèves rédigeront leurs algorithmes (Annexe 4).</li> </ul>
Selon la maîtrise qu'ont les élèves du logiciel Scratch, l'enseignant décidera s'il laisse les apprenants chercher parmi tous les blocs (niveau très avancé), si les apprenants doivent chercher parmi des blocs fournis (niveau avancé – Annexe 6) ou si les apprenants doivent remettre dans l'ordre les blocs fournis (niveau débutant – Annexe 5).	
Trois robots Lego Spike → Modèle : Rhino Spike.  Pour les moments d'observation (étapes 1) : l'enseignant aura programmé deux robots (voir programmes en Annexes 1 et 3).	Matériel d'écriture : feuilles, stylos et tapis de sol pour la circulation plus fluide du robot.
Pour les moments de comparaison (étapes 3), l'enseignant aura un seul robot mais les deux projets ouverts dans le logiciel afin de comparer facilement sa programmation à celle du groupe 1 et puis à celle du groupe 2.	











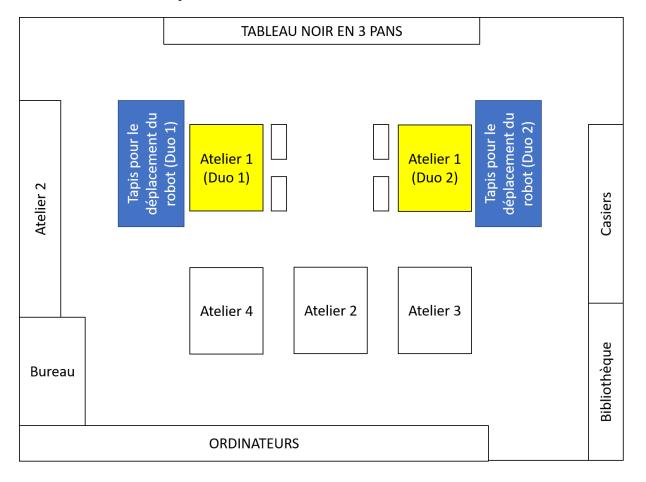
# Organisation de la classe

⇒ Cinq groupes de quatre ateliers tournants sur plusieurs semaines afin que tous passent par l'atelier « robot ».

1) ATELIER: activité PIAF Lego Spike

2) ATELIER: sujet au choix3) ATELIER: sujet au choix4) ATELIER: sujet au choix

Schéma illustrant la disposition de la classe













# Déroulement du scénario (activités)

Étapes des activités	Description (Consignes - Rôle du formateur – Activité de l'apprenant)	Compétences PIA (n°) et autres
	ation du robot et écriture d'algorithmes (50')	
1. Explication de la consigne et rédaction des algorithmes (20')	Première observation des déplacements du robot et rédaction des algorithmes  Consigne de l'enseignant  « l'ai programmé deux robots pour qu'ils effectuent une série d'actions bien précises. Votre MISSION sera d'observer très attentivement un des robots, afin de séquencer par étapes ses différentes actions et de les écrire en utilisant des phrases simples (donner un exemple : le robot recule de 30 cm). »  Matériel : robot + le tapis, une feuille (Annexe 4) et un crayon par duo.  Déroulement  1. Attribuer une table, un tapis et un robot (déjà programmé) à chaque duo.  2. Leur fournir la feuille sur laquelle ils écriront leurs algorithmes (Annexe 4).  3. Montrer une première fois le déplacement du robot auprès de chaque groupe. Les élèves devront se mettre d'accord et rédiger en groupe les différentes actions (l'algorithme) réalisées par le robot.  Chaque robot aura été programmé différemment en termes de paramètres d'actions, mais	5.1. (3.1.)









2. Comparaison des déplacements des deux robots (20')	ils auront les mêmes actions (ex. avancer de x ou y cm, tourner à droite pendant x ou y secondes).  Algorithme du robot # 1 : voir Annexe 1.  Algorithme du robot # 2 : voir Annexe 1.  Les groupes peuvent lancer le programme du robot autant de fois que nécessaire.  A ce stade, il est important que chaque groupe ne puisse pas voir le robot des voisins.  Les étudiants peuvent utiliser du matériel pour mesurer les distances parcourues par exemple.  Alternative proposée par une enseignante : donner les mêmes actions avec les mêmes paramètres et, pour la comparaison des algorithmes, demander à chaque groupe d'écrire son algorithme au tableau afin de les comparer.  Afin de mettre en évidence la différence au niveau des paramètres et la similitude au niveau des actions, l'enseignant invite les élèves à se rassembler pour observer la comparaison des comportements des deux robots.  Le programme de chaque robot est lancé à tour de rôle.  Les élèves des deux groupes sont invités à s'exprimer sur ce qu'ils viennent d'observer. Les élèves doivent en arriver à la conclusion que les actions sont effectivement identiques mais pas les paramètres (distance, du temps). « Effectuent-ils par les mêmes actions ? » « Quelles différences observez-vous ? » « Pouvons-nous dire que ce sont les actions qui sont différentes ? »	4.2. 5.1. (3.1.) (6.2.) 6.3.
	Les élèves sont ensuite invités à retourner dans leur atelier respectif afin d'affiner leur algorithme et ce, dans le but qu'il corresponde au mieux à la réalité de leur robot.	0.3.
3. Mise en évidence des critères pour	Ils sont dans un premier temps invités à mettre en évidence les critères pour rédiger un algorithme : « Vous avez eu l'occasion de rédiger les instructions qu'a reçu le robot pour	(4.2.)
rédiger les algorithmes (15')	réaliser ce que vous avez observé. Selon vous, quels sont les critères pour que ces instructions soient bien comprises et effectuées correctement ? »	4.3.









	L'enseignant peut décider de comparer les algorithmes réalisés par les groupes afin d'en	(3.1.)
	discuter et de mettre en évidence les critères d'un bon algorithme :	(0.1.)
	- L'algorithme est découpé en étapes clairement délimitées.	
	- L'ordre des étapes/actions de l'algorithme respecte l'ordre des déplacements du robot.	
	<ul> <li>Le vocabulaire utilisé est précis (précision des distances, du temps,) et sans ambiguïté.</li> </ul>	
	L'enseignant peut introduire, en fonction de ce qui est dit et observé, la notion d'optimisation (rédiger le programme le plus court) afin de faire émerger que certains comportements se répètent.	
	Les élèves sont invités à changer leur séquence d'actions en se basant sur ce qui a été dit lors du regroupement.	
4. Test des algorithmes avec le	L'enseignant explique aux élèves qu'ils vont devoir utiliser un logiciel pour vérifier que ce qui a été rédigé corresponde bien aux actions du robot.	5.2.
logiciel Lego Spike	Spike « Lors de l'activité 1, vous avez observé les déplacements du robot à partir de programmes déjà créés. Maintenant, c'est à vous de jouer! Vous allez utiliser le langage de programmation Scratch et demander au robot de réaliser les actions que vous avez rédigées dans votre groupe. »	6.1.
(25')		(6.2.)
		(6.3.)
	L'enseignant ouvre le logiciel Spike sur l'ordinateur de chaque duo et vérifie que chaque robot est bien connecté. Il ouvre sur chaque ordinateur le projet comprenant les différents blocs qui pourraient être utilisés par les élèves (Annexe 5 ou Annexe 6).	(3.1.)
	Il laisse ensuite les duos manipuler le logiciel afin qu'ils atteignent l'objectif donné : traduire l'algorithme écrit en langage Scratch et le tester avec son robot.	
	Les élèves ont la liberté de tester plusieurs fois ce qu'ils ont programmé. Dès qu'ils pensent avoir fini, ils en informent l'enseignant.	
	Anticipation des difficultés et pistes de solution	











T	
<ol> <li>Problème de déplacement des blocs via l'utilisation de la souris (montrer l'utilisation directement aux enfants);</li> </ol>	
2) Problème technique avec le robot (essai de résolution par l'enseignant) ;	
3) Incompréhension du vocabulaire utilisé sur les blocs (explications/traduction/synonyme);	
Remarque : il a été décidé de programmer les rotations en secondes car les degrés ne correspondent pas à l'amplitude réalisée par le robot.	
L'enseignant regroupe les deux duos et montre d'abord les actions du robot du duo 1 en	3.1.
comparaison à celles de son robot (et dont le programme correspond). Ensuite, les actions du robot du duo 2 sont comparées à celles de son robot (deuxième programme).	4.2.
À cette étape plusieurs observations sont possibles :	(4.3.)
- Les robots font exactement ce que les robots de l'enseignant font. Dans ce cas, il proposera	(5.2.)
de comparer les programmes des duos à ceux de ses robots afin de mettre en évidence de	6.1.
nouvelles caractéristiques d'un bon algorithme (voir plus haut) et afin de mettre en évidence la boucle (optimisation – réduction du nombre de lignes de code) si cela n'a pas	(6.2.)
été fait : « Comment pouvons-nous réduire votre programme ? », « Y a-t-il des actions que	(6.3.)
Si cela est nécessaire, l'enseignant pourrait également montrer un ou deux exemples de programme avec boucle (pas ceux des robots ici) afin de l'/les observer ensemble.	
- Le ou les robots ne font pas exactement ce que le robot de l'enseignant fait. Dans ce cas, l'enseignant fait réfléchir les élèves sur l'origine du problème : les actions écrites ne sont pas assez claires et/ou le programme ne correspond pas à celui du robot de l'enseignant. Il propose alors aux élèves de comparer leurs algorithmes écrits. Il fera en sorte de faire réfléchir les élèves sur ce qui pourrait permettre de rendre un algorithme plus clair et plus précis (caractéristiques d'un bon algorithme) : « Quelles sont les éléments importants pour	
	directement aux enfants);  2) Problème technique avec le robot (essai de résolution par l'enseignant);  3) Incompréhension du vocabulaire utilisé sur les blocs (explications/traduction/synonyme);  Remarque : il a été décidé de programmer les rotations en secondes car les degrés ne correspondent pas à l'amplitude réalisée par le robot.  L'enseignant regroupe les deux duos et montre d'abord les actions du robot du duo 1 en comparaison à celles de son robot (et dont le programme correspond). Ensuite, les actions du robot du duo 2 sont comparées à celles de son robot (deuxième programme).  À cette étape plusieurs observations sont possibles :  - Les robots font exactement ce que les robots de l'enseignant font. Dans ce cas, il proposera de comparer les programmes des duos à ceux de ses robots afin de mettre en évidence de nouvelles caractéristiques d'un bon algorithme (voir plus haut) et afin de mettre en évidence la boucle (optimisation – réduction du nombre de lignes de code) si cela n'a pas été fait : « Comment pouvons-nous réduire votre programme ? », « Y a-t-il des actions que l'on pourrait regrouper ensemble ? ».  Si cela est nécessaire, l'enseignant pourrait également montrer un ou deux exemples de programme avec boucle (pas ceux des robots ici) afin de l'/les observer ensemble.  - Le ou les robots ne font pas exactement ce que le robot de l'enseignant fait. Dans ce cas, l'enseignant fait réfléchir les élèves sur l'origine du problème : les actions écrites ne sont pas assez claires et/ou le programme ne correspond pas à celui du robot de l'enseignant. Il propose alors aux élèves de comparer leurs algorithmes écrits. Il fera en sorte de faire réfléchir les élèves sur ce qui pourrait permettre de rendre un algorithme plus clair et plus











	comparer ce qui est écrit et ce que le robot fait. Il pourra également faire réfléchir quant à la boucle : « Peut-on réduire votre algorithme ou avez-vous trouvé comment le faire ? ». Une fois corrigé, il proposera aux élèves de tester leur nouvel algorithme avec le logiciel.  Si finalement les algorithmes mettent clairement en évidence les actions du robot, il proposera de jeter un œil au(x) programme(s) du/des duos(s) qui ne corresponde(nt) pas aux actions du robot modèle. Il utilisera son robot pour comparer ses actions à celles du programme des élèves.	
	Durant cette phase, l'enseignant pourra mettre des mots sur ce qu'est un algorithme (= Description précise d'une séquence d'actions à réaliser pour atteindre un objectif).	
Prolongement	Cette activité peut également se reproduire avec les programmations en Annexe 2. Celle-ci mettent en évidence une difficulté supplémentaire : la boucle dans une boucle.	
Activité 2 - observe	ation du robot et écriture d'algorithmes avec la condition (50')	
1. Mise en situation (50')	Cette activité passe exactement par les mêmes étapes que celles de l'activité 1. Les seules différences sont les programmes utilisés car ceux-ci incluent la condition.	5.1.
	Tout comme pour l'activité 1, deux robots sont programmés avec les mêmes actions mais avec des valeurs différentes (Annexe 3). Chaque duo observe le comportement du robot qui leur a été attribué. Pour cela, l'enseignant dispose le robot face à un mur et l'actionne. Les élèves peuvent relancer plusieurs fois le programme du robot afin de rédiger l'algorithme sur la feuille distribuée préalablement par l'enseignant (Annexe 4).	3.3. (3.1.) (6.2.) (6.3.)
	Ensuite, un rassemblement des deux groupes est organisé afin que ceux-ci comparent les comportements des deux robots. La conclusion attendue est que les actions sont identiques, mais que les valeurs sont quant à elles différentes. Les élèves sont alors invités à préciser leurs algorithmes écrits.	
	Cette première étape se termine par la comparaison des algorithmes des duos. L'enseignant veille lors de cette discussion que la condition soit mise en avant : « Comment avez-vous exprimé le moment où le robot se dirige vers le mur ? ». Il est possible que les élèves aient mis	











	une distance à la place d'une condition. Dans ce cas, l'enseignant place le robot loin du mur et lance le programme. Les élèves se rendront compte qu'il n'y a pas de distance programmée à ce moment-là et devront trouver une autre manière d'exprimer cette étape. De plus, un autre élément qui devrait interroger les élèves est le fait que le programme se relance à chaque fois après que le bouton ait été pressé.  Alternative proposée par une enseignante : proposer les mêmes actions avec les mêmes paramètres et, pour la comparaison des algorithmes, demander à chaque groupe d'écrire son algorithme au tableau afin de les comparer.	
2. Test des	Cette deuxième étape correspond à la traduction de l'algorithme écrit en langage Scratch avec	5.2.
algorithmes avec le	le logiciel Lego Spike. L'enseignant prépare à nouveau les ordinateurs pour les deux groupes ainsi que les projets (Annexes 5 et 6). Une fois que les duos pensent avoir fini, ils en informent	3.3.
logiciel (25')	l'enseignant.	3.1.
		6.1.
		(6.2.)
		(6.3.)
3. Mise en commun	La troisième étape est la mise en commun :	(5.2.)
de l'activité 2 (25')	- Si les robots des duos font exactement la même chose que celui de l'enseignant, alors l'enseignant proposera aux élèves de comparer les programmes des duos à ceux du robot de l'enseignant. L'enseignant souligne particulièrement l'utilisation de la condition et demande aux élèves d'essayer de la définir : « Ce qui a été utilisé ici est la condition, à quoi sert-elle ? Comment pourrait-on la définir ? ». L'enseignant amène les élèves à mettre des	3.3.
		3.1.
		6.1.
	mots sur celle-ci : la condition permet de faire en sorte que le robot réagisse ou non d'une	(6.2.)
	certaine manière quand un ou plusieurs critères sont respectés. Il mettra également en évidence l'utilisation de la boucle, et notamment celle qui se répète indéfiniment.	(6.3.)
	- Si un ou les deux robots ne font pas la même chose que celui de l'enseignant, les élèves sont invités à réfléchir sur l'origine du problème en travaillant d'abord sur la rédaction écrite et ensuite sur le programme Scratch (dans le cas où la rédaction écrite est correcte).	4.2.









	Si l'algorithme écrit pose problème, l'enseignant invite les élèves à essayer d'identifier ce qui manque ou de comprendre ce qui n'a pas été assez précisé. Si cela est nécessaire, il pointe le moment où la condition est utilisée et demande aux élèves de mettre des mots sur ce passage. Ensuite, il propose aux élèves d'essayer à nouveau sur Scratch.	
	- Si c'est le programme sur le logiciel Lego qui pose problème, l'enseignant propose de comparer le programme aux actions de son robot. Si cela n'a pas été fait, il pointe le moment où la condition est utilisée et demande aux élèves de mettre des mots sur ce passage : « Qu'avez-vous utilisé lorsque le robot arrive face au mur ? » « Que dire oralement pour ce passage ? ».  Il est également possible que les élèves n'aient pas trouvé comment répéter indéfiniment le programme. Dans ce cas, l'enseignant fera réfléchir les élèves sur ce que le robot fait : « Que fait le robot après avoir touché le mur ? » « Et ensuite ? » « Avec vos mots, expliquez ce qui a été demandé au robot pour qu'il recommence. »	
Idée de prolongement	Possibilité de demander aux élèves de créer leur propre programme avec boucle(s) et condition(s). Pour cela, ils pourront créer leur robot et fixer un objectif qu'ils souhaitent atteindre.	2.4. 3.3.
	Ce type d'activité permettra (probablement) aux étudiants d'utiliser de nouvelles commandes, et d'utiliser la boucle et la condition de manière plus complexe (utiliser des opérateurs, utiliser « sialorssinon », utiliser plusieurs boucles dans une condition, utiliser une condition dans une condition, etc.).	3.1.









# **Évaluation**

Compétences/ Objectifs PIA	Activités d'évaluation	Critères d'évaluation
5.1.	Écriture de l'algorithme à partir de l'observation des comportements d'un robot (Annexe 7)	<ol> <li>L'algorithme est découpé en étapes clairement délimitées.</li> <li>L'ordre des étapes/actions de l'algorithme respecte l'ordre des déplacements du robot.</li> <li>Le vocabulaire utilisé est précis et correct.</li> </ol>
5.2.	Traduction de l'algorithme rédigé en langage Scratch L'élève a la possibilité de tester avec son robot autant de fois qu'il le souhaite.	Le programme correspond à l'algorithme rédigé par l'élève
3.1.	Cette étape se déroule en même temps que les activités proposées pour évaluer les compétences 5.1 et 5.2. :  L'enseignant demande aux élèves de créer un algorithme et un programme utilisant la boucle afin de réduire leur programme.	Utilisation de la boucle dans l'algorithme rédigé Utilisation de la boucle dans le programme Scratch
3.3.	Cette étape se déroule en même temps que les activités proposées pour évaluer les compétences 5.1 et 5.2. : Le programme à traduire contient une condition. Si des blocs spécifiques ne sont pas préparés, l'enseignant informera les élèves que leur programme doit contenir une condition.	Utilisation de la condition dans l'algorithme rédigé Utilisation de la condition dans le programme Scratch









# Critères d'évaluation

Critères	C'est réussi si	Résultat
1) L'algorithme est découpé en étapes clairement délimitées.	Je sépare bien chaque étape en passant à la ligne et en recommençant une nouvelle phrase.	
2) L'ordre des étapes de l'algorithme respecte l'ordre des déplacements du robot.	Je note les étapes dans l'ordre.	
3) Le vocabulaire utilisé est précis et correct.	J'utilise des termes précis et sans ambigüité.	





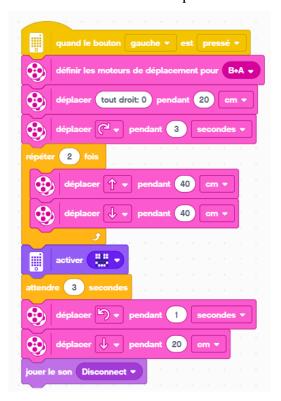




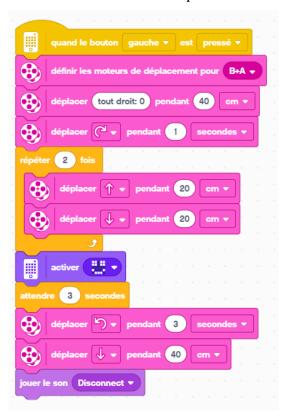


## ♦ Annexe 1 : Algorithmes utilisés

### Activité 1 - Groupe 1



Activité 1 – Groupe 2







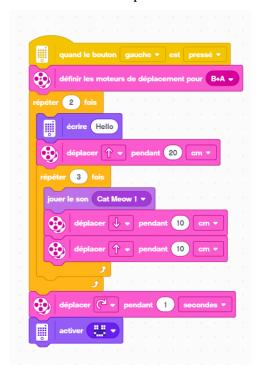






## ◆ Annexe 2 : Prolongement

## Groupe 1



## Groupe 2

```
2 fois
déplacer ↑ ▼ pendant 10 cm ▼
  déplacer ( → pendant 3 secondes
```





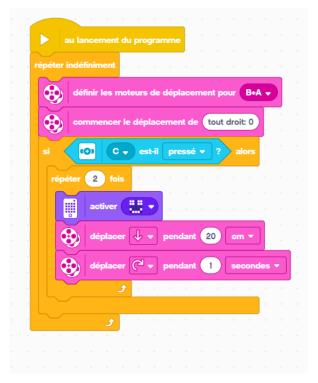






## ♦ Annexe 3 : Algorithmes utilisés lors de l'activité 2

### Activité 2 – Groupe 1



### Activité 2 – Groupe 2

```
encer le déplacement de (tout droit: 0
répéter 2 fois
```











### ♦ Annexe 4

# Que fait notre robot ?

Rédigeons notre algorithme (les étapes par lesquelles passe le robot)





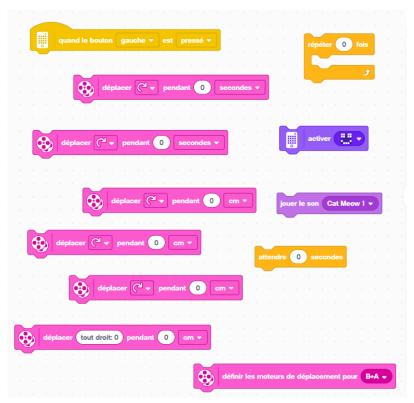







## ♦ Annexe 5 : Blocs à préparer pour les débutants

### Activité 1



### Activité 2

```
C → est-il pressé ▼
                       ent de (tout droit: 0
                                 er 🔁 ▼ pendant 0 secondes ▼
```





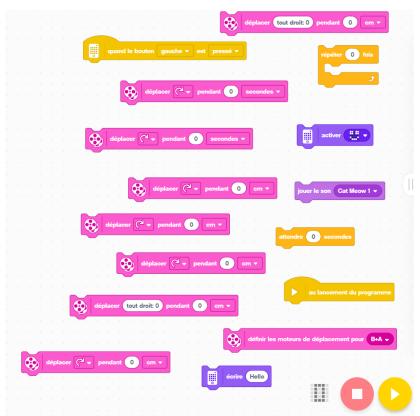






## ♦ Annexe 6 : Blocs à préparer pour les plus avancés

### Activité 1



### Activité 2

```
déplacer v pendant 0 cm v déplacer v pendant 0 cm v déplacer v pendant 0 secondes v
```











### ♦ Annexe 7

