

## Algorithmique et programmation à l'école primaire

L'initiation à la programmation constitue une nouveauté importante. Il s'agit, pour l'école primaire, d'amorcer un travail qui sera poursuivi en sixième de collège et au cycle 4.

## I. Contextualisation : renvoi aux textes, programmes

L'initiation à la programmation s'inscrit dans les objectifs du socle commun de connaissances, de compétences et de culture où il est précisé dans **le domaine 1, les langages pour penser et communiquer et la composante 3, comprendre, s'exprimer en utilisant les langages mathématiques, scientifiques et informatiques** que l'élève :

- sait que des langages informatiques sont utilisés pour programmer des outils numériques et réaliser des traitements automatiques de données ;
- connaît les principes de base de l'algorithmique, de la conception des programmes informatiques et les met en œuvre pour créer des applications simples.

L'initiation à la programmation apparaît également dans les programmes au sein du thème « **Espace et géométrie** » :

	Cycle 2	Cycle 3	
<i>Attendus de fin de cycle</i>	(Se) repérer et (se) déplacer en utilisant des repères	(Se) repérer et (se) déplacer dans l'espace en utilisant ou en élaborant des représentations	Reconnaître et utiliser quelques relations géométriques
<i>Situations et activités</i>	Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran.	Travailler avec de nouvelles ressources comme... des logiciels d'initiation à la programmation...	Exemples de matériels : ... logiciels de programmation.

Le cadre de référence des compétences numériques fait aussi mention de la programmation :

	Niveau 1 (CE2)	Niveau 2 (6è)
<i>Domaine</i>	3 : Création de contenu	
<i>Compétence</i>	3.4 : Programmer	
<i>Descripteurs</i>	Lire et construire un algorithme qui comprend des instructions simples.	Réaliser un programme simple.

## II. Éclairages notionnels et didactiques


## a. L'algorithme

Un algorithme est une suite d'opérations, d'instructions permettant de résoudre un problème. Il permet d'appréhender globalement le problème à résoudre. Il peut être élaboré en « débranché ». Il est d'ailleurs conseillé de faire d'abord réfléchir les élèves sur l'algorithme avant de recourir au codage.

## b. La pensée récursive

La récursion est un concept fondamental de l'informatique. Penser récursif, c'est s'attacher à résoudre des problèmes en les décomposant en sous-problèmes de même nature jusqu'à ne plus avoir que des problèmes triviaux puis composer ces solutions partielles en une solution globale au problème original (Pixees).



## c. Les types de déplacement

Déplacement absolu	Déplacement relatif
	
L'effet des instructions <b>ne dépend pas</b> de l'orientation initiale du « mobile » qui les reçoit. Par exemple : « avance de deux pas vers la fenêtre ». (Éduscol)	L'effet des instructions <b>dépend</b> de l'orientation initiale du « mobile » qui les reçoit. Par exemple : « effectue un quart de tour vers la droite ». (Éduscol)

## d. La problématique des pictogrammes polysémiques

Il convient d'utiliser avec une extrême prudence les pictogrammes traduisant les déplacements.

Exemple :


- Dans le cas d'un déplacement relatif, « → » pourrait signifier « avancer » si le mobile est dans la position suivante  ou « pivoter vers la droite » si le mobile est positionné ainsi .
- Dans le cas d'un déplacement absolu, le même pictogramme  signifierait alors « aller à droite », sans prise en compte de l'orientation initiale du mobile.

L'utilisation des termes « avancer, reculer, pivoter à droite, à gauche... » en lieu et place des pictogrammes peut s'avérer être plus pertinent.

## e. Le processus de décentration

La décentration est la capacité de différencier et de se représenter mentalement le point de vue d'autrui. Il s'agit d'une notion-clé à la base du développement de l'orientation et du déplacement dans l'espace.

#### f. Les différentes « tailles » d'espace (du proche au lointain)

Micro-espace	Meso-espace	Macro-espace
		
Le micro-espace est l'espace proche et extérieur au sujet.	L'individu élabore une vision globale d'un lieu en coordonnant les points de vue obtenus, au fur et à mesure de ses déplacements.	Le macro-espace, ne peut être appréhendé de manière directe et instantanée étant donné l'importance de sa surface.

Les types de déplacement, le processus de décentration et les différentes tailles d'espace sont des éléments à prendre en compte pour la conception de repères de progressivités pour construire le concept d'espace.

#### III. Repères de progressivité (Éduscol)

##### a. Au cycle 1

- Dans des récits, descriptions ou explications, apprendre à utiliser des marqueurs spatiaux adaptés (devant/ derrière, droite/gauche, dessus/dessous...).
- Se situer par rapport à d'autres, à des objets repères.
- Apprendre à situer des objets par rapport, à soi, entre eux, à des objets repères.
- Dans un environnement bien connu, réaliser un trajet, un parcours à partir de sa représentation (dessin ou codage)

Ce travail leur permet de développer l'aptitude à émettre des instructions élémentaires de déplacement, instructions qu'ils apprendront à associer dans les cycles suivants pour construire des programmes de déplacement.

##### b. Au cycle 2

- Coder et décoder pour prévoir.
- Représenter et réaliser des déplacements dans des espaces familiers, sur un quadrillage, sur un écran.
- À partir du CE1, coder des déplacements à l'aide d'un logiciel de programmation adapté.

##### c. Au cycle 3

- Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran.

#### IV. Conseils méthodologiques et pédagogiques

##### a. La démarche de résolution de problème

Elle doit être favorisée et explicitée par l'enseignant. L'apprenant en s'appropriant la démarche et en l'appliquant à la situation-problème, structure sa pensée, renforce son autonomie et développe ses capacités à chercher.

##### b. La méthode essais-erreurs

Les situations faisant appel à la programmation informatique ou de robots sont auto-validantes. Dans ce cadre, la méthode essais-erreurs est la modalité de résolution à favoriser.

##### c. Le statut et la gestion de l'erreur

L'erreur est formative. Partant de ce constat, il s'avère nécessaire d'apprendre aux élèves à les gérer. C'est dans ce cadre que la notion d'algorithme trouve sa place. Pour que l'élève dépasse l'erreur, il devient nécessaire de développer chez lui ses capacités d'anticipation (appréhender globalement le déplacement) et de séquençage (identifier chaque élément du parcours) avant de passer au codage. Cet apprentissage, qui permet de développer la pensée logique, peut se faire en « débranché ».

##### d. L'étayage de l'enseignant

Face à un obstacle que rencontre l'élève, l'étayage consiste à lui apporter une béquille pour l'aiguiller vers la solution **sans toutefois la lui apporter directement**.

Les propositions de séquences d'introduction à la programmation (Scratch JR au cycle 2 et Scratch au cycle 3) apportent plusieurs pistes de réflexion sur les modalités d'étayage, la mise en place de la méthode essais-erreurs et de la démarche de résolution de problème.

##### e. L'évaluation positive

Le développement des usages du numérique fait évoluer les modalités d'enseignement, mais également d'évaluation. Prendre des photos du travail des élèves, le sauvegarder dans la mémoire des machines, réaliser des captures d'écran sont des opérations techniques extrêmement simple à mettre en œuvre. Pourtant, effectuées à différents moments du projet, elles favorisent la régulation de l'acquisition des connaissances et des compétences, attestent de la progression de l'élève et évitent « l'évaluation sanction ».

##### f. La programmation et le plurilinguisme

L'initiation à la programmation pourra être une opportunité pour mener des travaux interdisciplinaires, par exemple, en créant des animations où les personnages dialoguent en langues régionales ou étrangère.

Des activités faisant appel à l'usage des instructions de déplacements dans les trois langues à l'oral puis à l'écrit, sont à favoriser dès le cycle 1 (exemple : avancer, *haere i mua*, *move forward*).

##### g. Les activités débranchées

Il est possible d'initier les élèves à la programmation sans forcément faire usage d'ordinateurs, de tablettes ou de robots. Des exemples d'activités de ce type, que l'on qualifie de « débranchées » sont proposés dans le document d'accompagnement « Initiation à la programmation aux cycles 2 et 3 » d'Éduscol.

## V. Exemples d'outils d'initiation à la programmation

	Cycle 1	Cycle 2	Cycle 3
Applications		Scratch JR	Scratch
Robots	Blue-Bot	Blue-Bot	Thymio II

## VI. Sitographie

- Stratégies d'enseignement pour l'initiation à la programmation.

Ressource d'accompagnement « Initiation à la programmation aux cycles 2 et 3 »

([http://cache.media.eduscol.education.fr/file/Initiation\\_a\\_la\\_programmation/92/6/RA16\\_C2\\_C3\\_MATH\\_initiation\\_programmation\\_doc\\_maitre\\_624926.pdf](http://cache.media.eduscol.education.fr/file/Initiation_a_la_programmation/92/6/RA16_C2_C3_MATH_initiation_programmation_doc_maitre_624926.pdf))

- Comprendre le numérique au travers d'activités débranchées.

Pixees, science participative, chaîne *YouTube* (<https://www.youtube.com/user/Scienceparticipative>)

- Ressources numériques clés en main qui ont fait leur preuve sur le terrain

<https://pixees.fr/utiliser-une-ressource/>

- Enseigner l'informatique à l'école et au collège

Manuel « 1,2,3 codez » de la fondation la Main à la pâte (<http://www.fondation-lamap.org/fr/123codez>)

- Prim à bord, le portail du numérique pour le premier degré

<http://eduscol.education.fr/primabord/>

- Apprendre à coder

<https://studio.code.org>

## VII. Annexes

Initiation à la programmation aux cycles 2 et 3 avec Scratch JR et Scratch (propositions de séquençage élaborées par l'équipe de la circonscription pédagogique n° 1).

# Scratch Jr au cycle 2 : anticiper et séquencer

Tihiura FALCHETTO

## I. Présentation

### A. Cycle

- Cycles des apprentissages fondamentaux

### B. Niveaux concernés

- CP au CE2

### C. Matériel

- Tablettes (iPad ou Android) dans lesquelles

## II. Programmes du cycle 2

### A. Domaine d'enseignement

- Mathématiques
- Espace et géométrie

### B. Compétences travaillées

#### 1. Chercher (domaines 2 et 4 du S4C)

- S'engager dans une démarche de résolution de problèmes en observant, en posant des questions, en manipulant, en expérimentant, en émettant des hypothèses, si besoin avec l'accompagnement du professeur après un temps de recherche autonome.
- Tester, essayer plusieurs pistes proposées par soi-même, les autres élèves ou le professeur.

#### 2. Modéliser (domaines 1, 2 et 4 du S4C)

- Utiliser des outils mathématiques pour résoudre des problèmes concrets, notamment des problèmes portant sur des grandeurs et leurs mesures.

#### 3. Raisonner (domaines 2, 3 et 4 du S4C)

- Anticiper le résultat d'une manipulation, d'un calcul, ou d'une mesure.
- Tenir compte d'éléments divers (arguments d'autrui, résultats d'une expérience, sources internes ou externes à la classe, etc.) pour modifier son jugement.
- Prendre progressivement conscience de la nécessité et de l'intérêt de justifier ce que l'on affirme.

#### 4. Communiquer (domaines 1 et 3 du S4C)

- Utiliser l'oral et l'écrit, le langage naturel puis quelques représentations et quelques symboles pour expliciter des démarches, argumenter des raisonnements.

### C. Attendus de fin de cycle

- (Se) repérer et (se) déplacer en utilisant des repères et des représentations.

### D. Exemples de situations, d'activités et de ressources pour l'élève

- Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran.

## III. Socle commun de connaissances de compétences et de culture

### A. Domaine 1 : les langages pour penser et communiquer

#### 1. Comprendre, s'exprimer en utilisant les langages mathématiques, scientifiques et informatiques.

- Il sait que des langages informatiques sont utilisés pour programmer des outils numériques et réaliser des traitements automatiques de données.

- Il connaît les principes de base de l'algorithmique et de la conception des programmes informatiques.
- Il les met en œuvre pour créer des applications simples.

## IV. Éléments à prendre en compte

### A. Apprendre aux élèves à anticiper et à développer ses capacités de séquençement.

#### 1. Proposer aux élèves une démarche de codage :

##### a) Démarche 1

- Je visualise le point de départ et d'arrivée ;
- J'image le trajet de Scratch ;
- Je place la première instruction et indique le nombre de case du déplacement ;
- Je teste la première partie du programme ;
- Je poursuis avec la deuxième instruction ou je rectifie ma première instruction.

##### b) Démarche 2

- Je visualise le point de départ et d'arrivée ;
- J' imagine le trajet de Scratch ;
- Je place les instructions de déplacement ;
- J'indique le nombre de case du déplacement ;
- Je lance mon programme.

#### 2. Proposer aux élèves une démarche de gestion des erreurs :

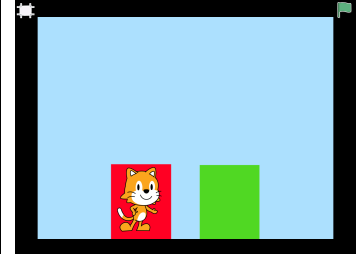
- Je lance mon programme ;
- Je repère mon erreur et identifie l'instruction en surbrillance ;
- Je cherche l'origine de mon erreur :
  - sens du déplacement ;
  - nombre de case du déplacement.
- Je corrige mon erreur ;

#### 3. Concevoir un affichage référentiel à étoffer au fur et à mesure des séances (consignes de travail, contraintes, démarches de codage et de gestion des erreurs, instructions utilisées...)

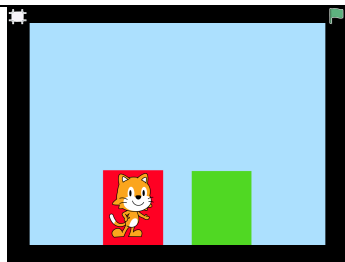
## V. Proposition de situations progressives


Il serait intéressant, pour débiter le projet, de présenter aux élèves un programme abouti réalisé avec Scratch Jr. De cette manière, les élèves sauront ce qu'il leur sera possible de réaliser en fin de séquence.

### A. Situation 1

	<ol style="list-style-type: none"> <li>1. Tâche de l'élève           <ul style="list-style-type: none"> <li>• Déplacer Scratch de la zone rouge à la zone verte.</li> </ul> </li> <li>2. Objectif de l'enseignant           <ul style="list-style-type: none"> <li>• Découvrir Scratch Jr librement.</li> </ul> </li> <li>3. Remarques           <ul style="list-style-type: none"> <li>• Présenter l'application aux élèves.</li> <li>• Les laisser la découvrir librement.</li> <li>• La majorité des élèves va pointer le chat et le déplacer au doigt.</li> <li>• Annoncer aux élèves qu'ils pourront tout essayer, appuyer sur tous les menus, se tromper et recommencer (approche de la méthode essai-erreur).</li> </ul> </li> </ol>
--	---


## B. Situation 2



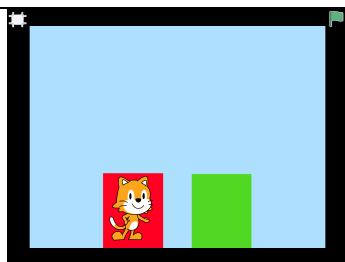
1. Tâche de l'élève
  - Déplacer Scratch de la zone rouge à la zone verte.
2. Objectif de l'élève
  - Découvrir et utiliser l'instruction *aller à droite* .
3. Contrainte
  - Ne pas toucher à Scratch.
4. Remarques
  - Laisser les élèves découvrir librement les différents menus puis les orienter vers le menu bleu contenant les instructions de déplacement.
  - Après plusieurs appuis sur les instructions de déplacement, les élèves constateront que rien ne se passe. Expliquer alors aux élèves l'utilité de la *zone de programmation* et la nécessité d'y déplacer les instructions.





• Afin de faciliter les recherches et aider les élèves à structurer leur pensée, il serait intéressant de noter leurs hypothèses et de barrer celles qui sont erronées, par exemple :

- ☒ Je déplace l'instruction sur Scratch.
- ☒ Je déplace l'instruction dans la case verte.
- ☐ Je déplace l'instruction dans la zone blanche (zone de programmation).
- A ce stade, la majorité des élèves placeront l'instruction *aller à droite*  dans la zone de programmation et appuieront plusieurs fois sur cette dernière jusqu'à ce que Scratch arrive dans la zone verte.

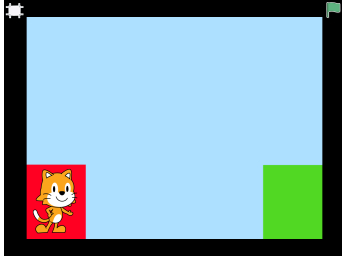
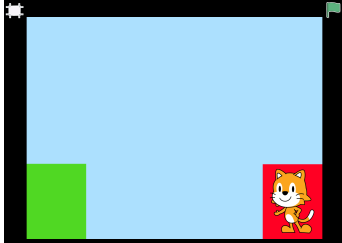
## C. Situation 3



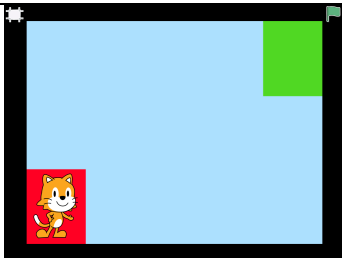




1. Tâche de l'élève
  - Déplacer Scratch de la zone rouge à la zone verte.
2. Objectif de l'enseignant
  - Associer plusieurs instructions de déplacement.
  - Introduire l'instruction retourner à la position initiale .
3. Contrainte
  - Appuyer une seule fois sur les instructions.
4. Remarques
  - L'idée, pour cette situation, est de leur faire utiliser plusieurs fois l'instruction *aller à droite*  en les assemblant à la manière d'un « puzzle ».
  - Faire comprendre aux élèves que les acquis des situations précédentes sont à mobiliser pour les nouvelles situations.

## D. Situation 4

Avant de débiter la situation 4, il serait intéressant de montrer aux élèves comment créer un nouveau projet et insérer des arrières plans.

	<ol style="list-style-type: none"> <li><b>Tâche de l'élève</b> <ul style="list-style-type: none"> <li>Déplacer Scratch de la zone rouge à la zone verte.</li> </ul> </li> <li><b>Objectif de l'enseignant</b> <ul style="list-style-type: none"> <li>Réduire le nombre d'instructions lorsqu'il s'agit de déplacer plusieurs fois de suite un lutin dans une même direction.</li> </ul> </li> <li><b>Contrainte</b> <ul style="list-style-type: none"> <li>Utiliser une seule instruction de déplacement.</li> </ul> </li> <li><b>Remarques</b> <ul style="list-style-type: none"> <li>Déplacement à 1 direction.</li> <li>Au départ, laisser les élèves utiliser plusieurs instructions. Puis, leur faire remarquer qu'après en avoir déplacé une douzaine, la zone de programmation devient surchargée. À ce moment-là, orienter la réflexion vers la signification du chiffre 1 situé sous l'instruction.</li> <li>Définir avec les élèves les dimensions du quadrillage (20 cases par 15) et attirer leur attention sur les coordonnées de départ (mises en surbrillance) du chat.</li> </ul> </li> <li><b>Prolongement</b>  </li> </ol>
--	--

#### E. Situation 5

	<ol style="list-style-type: none"> <li><b>Tâche de l'élève</b> <ul style="list-style-type: none"> <li>Déplacer Scratch de la zone rouge à la zone verte.</li> </ul> </li> <li><b>Objectifs de l'enseignant</b> <ul style="list-style-type: none"> <li>Découvrir et utiliser l'instruction <i>monter</i> .</li> <li>Coder un déplacement à deux directions.</li> <li>Savoir déterminer le nombre de cases d'un déplacement.</li> </ul> </li> <li><b>Contrainte</b> <ul style="list-style-type: none"> <li>Utiliser le minimum d'instructions.</li> </ul> </li> <li><b>Remarques</b> <ul style="list-style-type: none"> <li>Déplacement à 2 directions.</li> <li>Découverte de l'usage de l'instruction <i>monter</i> .</li> <li>Indiquer aux élèves que les instructions doivent être « attachées » les unes aux autres.</li> <li>Il serait intéressant de proposer les démarches de codage et de gestion des erreurs à partir de cette situation.</li> <li>Expliquer aux élèves comment déterminer le nombre de cases d'un déplacement (différence entre les coordonnées d'arrivée et de départ).</li> <li>Les deux solutions possibles avec le minimum d'instructions sont :   et . </li> <li>Profiter de cette situation pour exploiter et réguler les propositions comportant trop d'instructions :</li> </ul> </li> </ol>
---	---

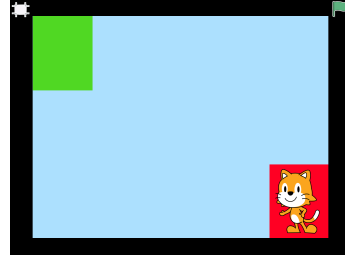


Exemple :

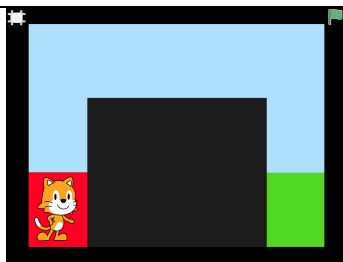


- Proposer aux élèves la démarche de codage et de gestion des erreurs.

#### 5. Prolongement




### F. Situation 6



#### 1. Tâche de l'élève

- Déplacer Scratch de la zone rouge à la zone verte.


#### 2. Objectifs de l'enseignant

- Découvrir et utiliser l'instruction *descendre*  ;
- Coder un déplacement à trois directions.

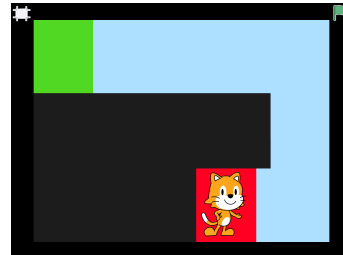
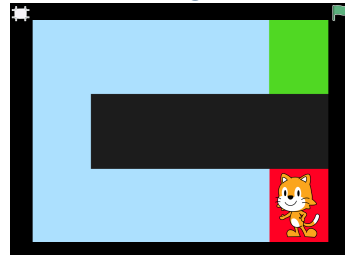
#### 3. Contrainte

- Scratch ne doit pas toucher le « mur noir ».

#### 4. Remarques

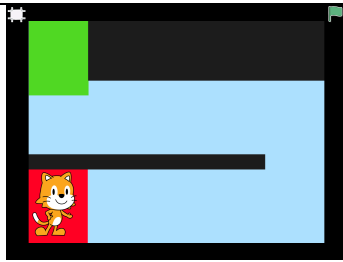
- Déplacement à 3 directions ;
- Usage de l'instruction *descendre*  ;
- Expliquer que le nombre de cases de déplacement vers le haut est égal au nombre de cases de déplacement vers le bas.

#### 5. Prolongements





## G. Situation 7




### 1. Tâche de l'élève

- Déplacer Scratch de la zone rouge à la zone verte.

### 2. Contrainte

- Utiliser le minimum d'instructions ;
- Ne pas traverser les murs noirs.

### 3. Objectifs de l'enseignant

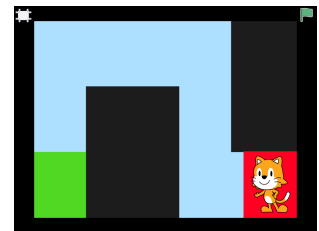
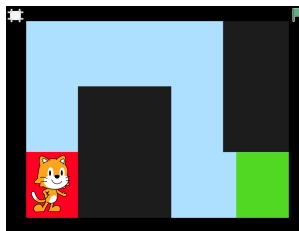
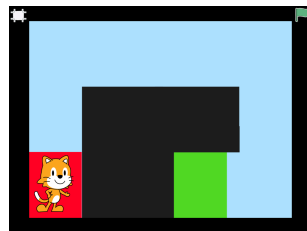
- Découvrir et utiliser l'instruction *aller à gauche*  ;
- Coder un déplacement à quatre directions ;

### 4. Remarques

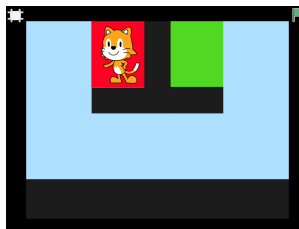
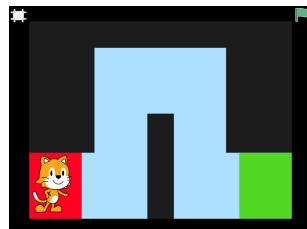
- Déplacement à 4 directions
- Usage de l'instruction « Déplacement à gauche ».

### 5. Prolongements

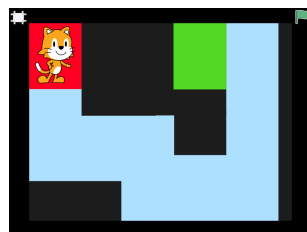
#### 4 déplacements



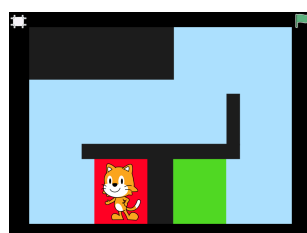
#### 5 déplacements





#### 6 déplacements









#### 7 déplacements



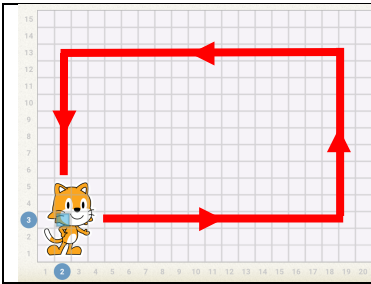
## H. Situation 8


	<ol style="list-style-type: none"><li>1. Tâche de l'élève<ul style="list-style-type: none"><li>• Faire suivre le chemin au ballon/</li></ul></li><li>2. Contrainte<ul style="list-style-type: none"><li>• Utiliser le minimum d'instructions.</li></ul></li><li>3. Objectifs de l'enseignant<ul style="list-style-type: none"><li>• Découvrir que certaines combinaisons de déplacements peuvent se répéter ;</li><li>• Découvrir et utiliser l'instruction <i>répéter</i> .</li></ul></li><li>4. Remarque<p>Pour introduire l'instruction , il serait intéressant de transcrire le programme en langue française :</p><p><i>Répéter 5 fois :</i></p><ul style="list-style-type: none"><li>• Monter de 2 cases</li><li>• Aller à droite de 4 cases.</li></ul><p>« Monter de deux cases » et « aller à droite de 4 cases » sont des instructions connues, demander alors aux élèves de trouver l'instruction qui signifie « Répéter 5 fois ».</p></li></ol>
--	--

## I. Situation 9 : découverte des instructions les plus usuelles

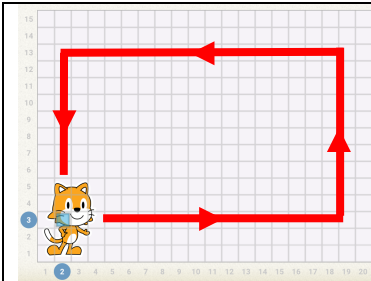
1. Tâche de l'élève
  - Tester une instruction et déterminer sa fonction
2. Objectif de l'enseignant
  - Découvrir et utiliser les instructions les plus usuelles.
3. Remarques
  - Afficher les instructions et laisser les élèves les tester afin de déterminer leur fonction ;
  - Il est possible de fournir aux élèves une feuille contenant la liste des instructions. Ils pourront ainsi y noter leurs idées ;
  - Faire confronter les propositions et convenir d'une définition commune pour chaque instruction :
    - Agrandir 
    - Rétrécir 
    - Taille normale 
    - Cacher 
    - Afficher 
    - Définir la vitesse 


## J. Situation 10



1. Tâche de l'élève
  - Faire 6 tours de la zone d'évolution.
2. Objectif de l'enseignant
  - Maîtriser l'instruction *répéter* .

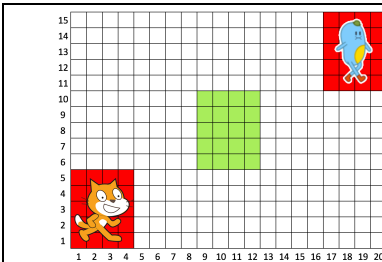
## K. Situation 11




1. Tâche de l'élève
  - Faire le tour perpétuel de la zone d'évolution.
2. Objectif de l'enseignant
  - Découvrir et comprendre l'instruction *répéter toujours* .

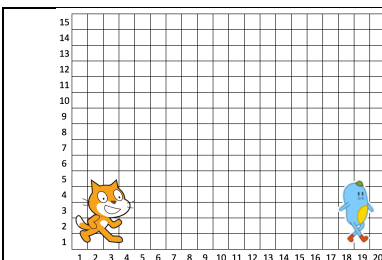
## L. Situation 12



Cette situation doit être précédée par une séance d'apprentissage sur l'élaboration puis l'insertion d'un arrière-plan.



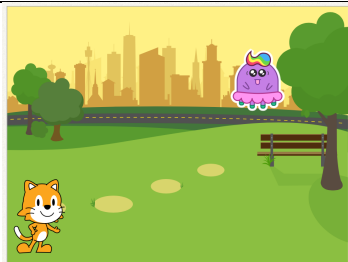
1. Tâche de l'élève
  - Déplacer Scratch et Tic des zones rouges à la zone verte
2. Contraintes
  - Les deux personnages doivent partir et arriver en même temps.
3. Objectifs de l'enseignant
  - Insérer un nouveau lutin ;
  - Découvrir l'instruction de départ *lancer le programme*  ;
  - Assigner des instructions différentes aux lutins.



## M. Situation 13



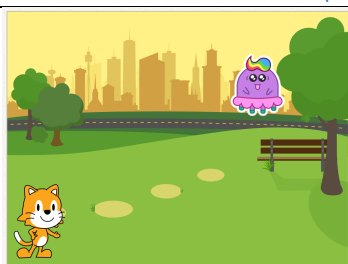
1. Tâche de l'élève
  - Déplacer Scratch vers Tic et Tic vers Scratch
2. Contraintes
  - Faire sauter Scratch lorsqu'il touche Tic ;
3. Objectifs de l'enseignant
  - Déclencher un programme sous condition ;
  - Découvrir et utiliser les instructions *lancer le programme si collision*  et *sauter* .
4. Remarque
  - Le saut a lieu lorsque les deux personnages entrent en collision. Pour valider la situation, déplacer Scratch vers le haut de manière à ce qu'il ne touche plus Tic puis lancer le programme. Si Scratch saute dans le vide, le programme n'est pas valable.


## N. Situation 14



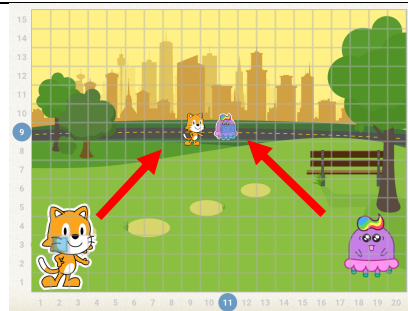
- Tâche de l'élève
  - Lorsque Scratch est touché, le faire se déplacer vers Tac puis lui faire dire « Bonjour Tac ! » (info-bulle).
- Objectifs de l'enseignant
  - Afficher une *bulle de texte écrit*  ;
  - Personnaliser le texte contenu dans la bulle.
  - Réponse vocale de Tac (son enregistré).
  - Découvrir et utiliser l'instruction .
  - Insérer un nouveau lutin : Tac ;
  - Insérer un « arrière-plan » prédéfini.





## O. Situation 15 (suite de la situation 14)



1. Tâche de l'élève
  - Faire répondre Tac à Scratch : « Bonjour Scratch, comment vas-tu ? »
2. Objectifs de l'enseignant
  - Faire *attendre*  le programme avant le déclenchement d'une nouvelle instruction.
3. Remarques
  - Bâtir ce programme à partir de celui de la situation précédente.

## P. Situation 16 : insérer un son



1. Tâche de l'élève
  - Déplacer Scratch et Tac vers la route ;
  - Lors du déplacement, rétrécir progressivement les personnages pour marquer l'effet de perspective ;
  - Élaborer (enregistrements sonores et/ou bulles de texte écrit) un dialogue d'au moins 6 échanges.
2. Objectifs de l'enseignant
  - Utiliser les fonctions *rétrécir* , *message envoyé*  et *message reçu* .
  - Combiner plusieurs déplacements et mouvements.
  - Insérer un son et le jouer au moment opportun .

A l'issue de cette séquence, les élèves maîtriseront les éléments de base de l'application Scratch Jr. Il leur sera possible alors de réaliser des défis.

# Découvrir Scratch au C3 : la pensée computationnelle

Tihiura FALCHETTO – novembre 2017

## I. Présentation

### A. Cycle

Cycles des consolidations

### B. Niveaux

- CM1
- CM2
- 6è

### C. Domaines d'enseignement

- Mathématiques
- Espace et géométrie
  1. Attendus de fin de cycle : Reconnaître et utiliser quelques relations géométriques.
    - a) *Connaissances et connaissances associées*
  - Effectuer des tracés correspondant à des relations de perpendicularité ou de parallélisme de droites et de segments.
  - Déterminer le plus court chemin entre deux points (en lien avec la notion d'alignement).
  - Déterminer le plus court chemin entre un point et une droite ou entre deux droites parallèles (en lien avec la perpendicularité).
    - Alignement, appartenance.
    - Perpendicularité, parallélisme (construction de droites parallèles, lien avec la propriété reliant droites parallèles et perpendiculaires).
    - Égalité de longueurs.
    - Égalité d'angles.
    - Distance entre deux points, entre un point et une droite.
  - b) *Exemples de situations, d'activités et de ressources pour l'élève*
- Situations conduisant les élèves à utiliser des techniques qui évoluent en fonction des supports et des instruments choisis ; par exemple pour la symétrie axiale, passer du pliage ou de l'utilisation de papier calque à la construction du symétrique d'un point par rapport à une droite à l'équerre ou au compas.
  - (1) *Exemples de matériels*
- papier/crayon
- logiciels de géométrie dynamique
- *logiciels d'initiation à la programmation*
- logiciels de visualisation de cartes, de plans.
  2. Attendus de fin de cycle : (Se) repérer et (se) déplacer dans l'espace en utilisant ou en élaborant des représentations
    - a) *Connaissances et connaissances associées*
- Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran.

b) Exemples de situations, d'activités et de ressources pour l'élève

- Situations donnant lieu à des repérages dans l'espace ou à la description, au codage ou au décodage de déplacements.
- Travailler avec de nouvelles ressources comme les systèmes d'information géographique, *des logiciels d'initiation à la programmation...*

D. Objectifs généraux

- Résoudre des problèmes dans un cadre motivant ;
- Adopter une démarche scientifique ;
- Structurer la pensée ;
- Se familiariser avec la méthode essai-erreur ;
- Développer son sens logique, l'anticipation et ses capacités de séquençement.

## II. Proposition de séances et situations progressives

### A. Séance 1 : Utiliser Scratch librement

#### 1. Objectifs spécifiques et apprentissages techniques

- Découvrir et savoir lancer et utiliser l'application Scratch ;
- Découvrir et connaître l'environnement de programmation de Scratch ;
- Construire une première définition du mot « programmer »
  - *Définition attendue en fin de séquence* : donner des ordres à un ordinateur afin qu'il réalise des actions.


#### 2. Déroulement succinct : 30 minutes

Enseignants	Élèves
<i>a) Mise en situation : 5 minutes</i>	
<ul style="list-style-type: none"> <li>• Écrire le mot <i>programmer</i> au tableau.</li> <li>• Noter les hypothèses des élèves sur une grande feuille.</li> <li>• Expliquer aux élèves qu'ils devront construire et affiner la définition du mot <i>programmer</i> au fil des séances.</li> </ul>	<ul style="list-style-type: none"> <li>• Débattre sur la signification du mot au tableau.</li> <li>• Donner des mots-clés</li> </ul>
<i>b) Découverte : 10 minutes</i>	
<ul style="list-style-type: none"> <li>• Expliquer aux élèves comment lancer Scratch.</li> <li>• Variable didactique n°1 : Après un premier bilan d'étape, si aucun élève n'a focalisé son attention sur les possibilités de déplacement du lutin, orienter leur attention vers les instructions de couleur bleue. Par contre, si les élèves découvrent que Scratch peut se déplacer en cliquant sur lui, ajouter la contrainte de faire déplacer Scratch sans le toucher à la souris.</li> </ul>	<ul style="list-style-type: none"> <li>• Mettre en route les ordinateurs et lancer Scratch.</li> <li>• Découvrir l'application librement.</li> </ul>

<ul style="list-style-type: none"> <li>Variable didactique n°2 : Après un second bilan d'étape, procéder de même que précédemment pour le rôle de la zone de script.</li> </ul>	
<i>c) Restitution : 10 minutes</i>	
<ul style="list-style-type: none"> <li>Projeter l'interface de l'application.</li> <li>Noter les propositions intéressantes des élèves au tableau.</li> <li>Institutionnaliser le vocabulaire relatif aux différentes parties de l'environnement de Scratch.</li> <li>Afficher le référent <i>01 L'environnement de programmation de Scratch</i></li> <li>Noter la procédure sur une feuille.</li> </ul>	<ul style="list-style-type: none"> <li>Définir les différentes zones de l'environnement de Scratch (scènes, catégories de blocs, zone de script, liste des lutins...) et son intérêt (contrôler et agir sur le chat).</li> <li>Déterminer comment agir sur le lutin (faire bouger le lutin) sans le toucher : <ul style="list-style-type: none"> <li>Je déplace les blocs de la palette vers la zone de script.</li> <li>Je clique sur les instructions.</li> </ul> </li> </ul>
<i>d) Retour à la mise en situation et synthèse : 5 minutes</i>	
	<ul style="list-style-type: none"> <li>Construire collectivement une première définition du mot programmer ou, si les élèves rencontrent encore des difficultés, se contenter de glaner des mots-clefs.</li> </ul>


## B. Séance 2 : Faire avancer et reculer Scratch

### 1. Objectifs spécifiques et apprentissages techniques

- Réaliser son premier programme Scratch ;
- Comprendre et utiliser le bloc  ;
- Appréhender la notion de pixel ;
- Dans les instructions de déplacement et d'orientation : comprendre la signification du signe moins (–) lorsqu'ajouté avant le nombre.



### 2. Déroulement succinct (25 minutes)

<i>Enseignants</i>	<i>Élèves</i>
<i>a) Rappel : 2 minutes</i>	
<ul style="list-style-type: none"> <li>Comment contrôler le lutin ?</li> </ul>	<ul style="list-style-type: none"> <li>Déplacer les blocs de la palette vers la zone de script, puis cliquer sur l'instruction.</li> </ul>
<i>b) Situation problème : 13 minutes</i>	
<ul style="list-style-type: none"> <li>Comment faire avancer et reculer Scratch ? <ul style="list-style-type: none"> <li>Variable didactique n° 1 : ne pas toucher à Scratch.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Rechercher la solution au problème.</li> </ul>

<ul style="list-style-type: none"> <li>○ <i>Variable didactique n°2 : utiliser uniquement l'instruction</i> </li> <li>• Expliquer que la méthode essais-erreurs est privilégiée. <ul style="list-style-type: none"> <li>○ Leur montrer comment créer un nouveau projet afin de réinitialiser leur programme.</li> <li>○ Leur dire que si une manipulation ne fonctionne pas une fois, inutile d'insister, il faut essayer autre chose.</li> </ul> </li> </ul>	
<i>Étayage possible de l'enseignant</i>	<i>Obstacles que peuvent rencontrer les élèves</i>
<ul style="list-style-type: none"> <li>• Faire référence à la procédure de programmation évoquée lors de la séance précédente et du rappel et pointer l'affichage correspondant.</li> <li>• Mettre en évidence l'absence de l'instruction <i>reculer de ...</i> et orienter la réflexion vers le sens des opérateurs + et – en s'aidant de la frise numérique. <ul style="list-style-type: none"> <li>○ Exemple : si je suis sur la case du nombre 10. Quelle opération j'utilise pour avancer vers les cases 11, 12... ? Quel est son signe ? Maintenant, si je veux reculer. Quelle signe j'utilise ?</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Les élèves ne savent plus comment contrôler le lutin.</li> <li>• Les élèves ne trouvent pas comment faire reculer le lutin.</li> </ul>
<i>c) L'unité de déplacement : 5 minutes</i>	
<ul style="list-style-type: none"> <li>• Dans l'instruction <i>avance de 100</i>. Que représente le nombre 100 ?</li> <li>• Projeter une capture d'écran de l'interface à l'aide d'un logiciel de retouche d'image et zoomer au maximum pour faire apparaître les pixels et ainsi aider les élèves à appréhender cette notion.</li> </ul>	
<i>d) Synthèse : 5 minutes</i>	
<ul style="list-style-type: none"> <li>• Afficher le référent <i>02 Avancer et reculer</i>.</li> </ul>	<ul style="list-style-type: none"> <li>• Expliciter le contenu du référent <i>02 Avancer et reculer</i>.</li> </ul>

### C. Séance 3 : Faire pivoter Scratch

#### 1. Objectif spécifique et apprentissage technique

- Comprendre et utiliser les blocs  et  ;
- Connaître quelques mesures d'angles remarquables (0°, 90°, 180°, 270°)



## 2. Déroulement succinct (40 minutes)

<i>Enseignants</i>	<i>Élèves</i>
<i>a) Rappel : 2 minutes</i>	
<ul style="list-style-type: none"> <li>• Comment contrôler le lutin ?</li> </ul>	<ul style="list-style-type: none"> <li>• Déplacer les blocs de la palette vers la zone de script, puis cliquer sur l'instruction.</li> </ul>
<i>b) Situation problème : 13 minutes</i>	
<ul style="list-style-type: none"> <li>• Afficher Scratch dans différentes orientations (regard vers la droite, puis le haut, puis vers la gauche et enfin vers le bas) et demander aux élèves : Comment faire pour <i>pivoter</i> Scratch ?</li> <li>• Préciser que Scratch ne s'est pas déplacé. Il a tourné sur lui-même : il a <i>pivoté</i>.</li> <li>• Rappeler que la méthode essais-erreurs est privilégiée.</li> </ul>	<ul style="list-style-type: none"> <li>• Rechercher la solution au problème.               <ul style="list-style-type: none"> <li>○ Décomposer le problème en 3 sous-problèmes                   <ul style="list-style-type: none"> <li>○ 1<sup>er</sup> problème : SAM de 0 à 90°</li> <li>○ 2<sup>ème</sup> problème : SAM de 0 à 180°</li> <li>○ 3<sup>ème</sup> problème : SIAM de 0 à 90°</li> </ul> </li> <li>○ Variable didactique n°1 : Certains élèves réussiront à résoudre le problème en cliquant six fois sur l'instruction de rotation par défaut qui est de 15 degrés. Préciser alors qu'il faut résoudre les problèmes en cliquant une seule fois sur l'instruction.</li> </ul> </li> </ul>
<i>Étayage possible de l'enseignant</i>	<i>Obstacles que peuvent rencontrer les élèves</i>
<ul style="list-style-type: none"> <li>• Expliquer que du texte peut-être saisi dans les zones blanches de chaque instruction.</li> <li>• Représenter les regards de Scratch à 0° et 90° par deux traits formant un angle droit et faire remarquer aux élèves qu'ils connaissent cet angle et sa mesure. Pour rafraîchir leur mémoire, leur montrer éventuellement une équerre.</li> <li>• Leur faire remarquer qu'ils peuvent additionner la mesure de deux droits pour trouver une nouvelle mesure d'angles remarquables.</li> <li>• Cliquer plusieurs fois sur l'instruction <i>tourner de 90 degrés (ou 180 degrés)</i> pour bien s'approprier l'effet des instructions sur le lutin.</li> </ul>	<ul style="list-style-type: none"> <li>• Les élèves ne savent pas comment augmenter la mesure de l'angle.</li> <li>• Les élèves ne trouvent pas la mesure de l'angle droit.</li> <li>• Les élèves ne trouvent pas la mesure de l'angles plat.</li> <li>• Les élèves appréhendent difficilement les pivotements par quart de tour (ou demi-tour)</li> </ul>
<i>c) Pivoter à partir d'une orientation initiale différente : 5 minutes</i>	
<ul style="list-style-type: none"> <li>• Modifier l'orientation initiale de Scratch et demander aux élèves quelle sera</li> </ul>	<ul style="list-style-type: none"> <li>• Répondre en utilisant l'ordinateur comme outil d'aide.</li> </ul>

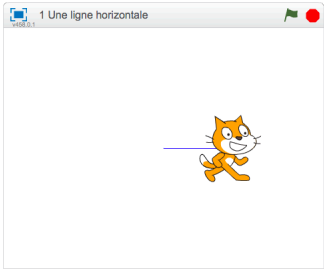
l'orientation finale de Scratch si on le pivote de 90°, 180° dans un sens ou dans un autre.	
<i>d) Exercice d'application : 10 minutes</i>	
<ul style="list-style-type: none"> <li>L'ordinateur peut être utilisé pour aider à réaliser l'exercice.</li> </ul>	<ul style="list-style-type: none"> <li>Associer chaque image aux instructions qui conviennent.</li> </ul>
<i>e) Correction et synthèse : 10 minutes</i>	
<ul style="list-style-type: none"> <li>Corriger l'exercice</li> <li>Afficher les référents <i>S03a Pivoter, S03b Pivoter, S03c Pivoter et S03d Pivoter.</i></li> </ul>	

#### D. Séance 4 : réalisation d'une figure simple (1)

##### 1. Objectifs spécifiques et apprentissages techniques

- Comprendre et utiliser l'instruction *stylo en position d'écriture* ;
- Réaliser une figure simple ;
- Savoir emboîter les blocs ;

##### 2. Déroulement succinct : 30 minutes

<i>Enseignants</i>	<i>Élèves</i>
<i>a) Rappel : 2 minutes</i>	
<ul style="list-style-type: none"> <li>Comment contrôler le lutin ?</li> <li>Signification des instructions <i>avancer de 100</i>, <i>tourner de 90 degrés</i> et <i>tourner de 90 degrés</i> et du signe moins.</li> </ul>	<ul style="list-style-type: none"> <li>Déplacer les blocs de la palette vers la zone de script, puis cliquer sur l'instruction.</li> <li>Avancer ou reculer (avec le moins) de 100 pixels ; pivoter d'un certain nombre de degrés.</li> </ul>
<i>b) Situation problème : 13 minutes</i>	
<ul style="list-style-type: none"> <li>Projeter une capture d'écran de la situation problème</li> </ul>  <ul style="list-style-type: none"> <li>Attirer l'attention des élèves sur le trait bleu et leur expliquer aux élèves qu'ils devront découvrir une nouvelle instruction.</li> <li>Rappeler que la méthode essais-erreurs est privilégiée.</li> </ul>	<ul style="list-style-type: none"> <li>Se projeter vers la résolution du problème en identifiant les différentes étapes du déplacement.</li> <li>Constater que le déplacement de Scratch est marqué.</li> </ul>
<i>Étayage possible de l'enseignant</i>	<i>Obstacles que peuvent rencontrer les élèves</i>

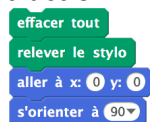
- Dans la majorité des cas, les élèves auront repéré l'angle droit. Leur demander alors comment il leur a été possible de visualiser l'angle droit : grâce au trait bleu.

- Déterminer avec eux les points de départ et d'arrivée de Scratch et décomposer le déplacement :
  - Stylo en position d'écriture ;
  - Avancer de 100 pixels ;

- Après avoir découvert l'instruction `stylo en position d'écriture`, leur expliquer qu'ils ne doivent résoudre le problème qu'avec les instructions qu'ils connaissent.

- Expliquer aux élèves qu'ils doivent résoudre le problème en « un clic » de souris. Attirer l'attention des élèves sur la forme en pièces puzzle des instructions.

- Afficher le référent « Réinitialiser la scène ». Leur montrer l'effet obtenu et leur expliquer brièvement la fonction de chaque instruction.



- Leur demander de lire les intitulés des catégories. Ils trouveront facilement l'instruction dans la catégorie « Stylo ».

- Mimer la situation avec une feuille et un feutre tenu en position relevée puis en position d'écriture. Leur montrer que lorsque le feutre est posé, rien ne se voit sur la feuille. Il faut associer l'instruction `stylo en position d'écriture` à l'instruction `avancer de 100` pour constater son effet.

- (Par les élèves ou par l'enseignant) Sur une feuille, détailler et numéroté les étapes permettant de réaliser la

- Après le temps d'analyse de la figure, les élèves ne font pas attention au trait bleu.

- Les élèves ont des difficultés à analyser la figure.

- Les élèves cherchent à utiliser de nouvelles instructions

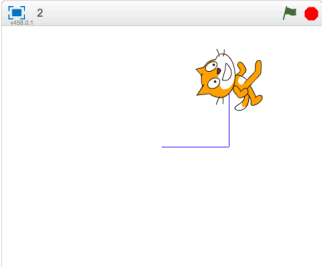
- Dans la zone de script, les instructions ne sont pas assemblées.

- Redémarrer Scratch pour réinitialiser la scène commence à faire perdre du temps et de l'efficacité.

- Les élèves ne trouvent pas l'instruction `stylo en position d'écriture`.

- Les élèves ne savent pas à quel niveau du programme insérer l'instruction `stylo en position d'écriture`.

- Les élèves n'arrivent pas avec à ordonner les instructions.

<p>figure et insister auprès des élèves qu'il faille respecter rigoureusement l'ordre des instructions :</p> <ol style="list-style-type: none"> <li>1. Stylo en position d'écriture</li> <li>2. <i>Avancer de 100</i></li> </ol> <ul style="list-style-type: none"> <li>• Leur demander de détacher temporairement les instructions et de les tester les unes après les autres dans l'ordre afin d'identifier celles qui sont à corriger.</li> <li>• Leur proposer le problème suivant.</li> </ul> 	<ul style="list-style-type: none"> <li>• Les élèves utilisent des instructions erronées.</li> <li>• Certains élèves terminent rapidement leur défi.</li> </ul>
<p><i>c) Synthèse : 10 minutes</i></p>	
<ul style="list-style-type: none"> <li>• Noter les étapes de la procédure sur une grande feuille.</li> <li>• Faire un retour sur une définition possible du mot « Programmer ».</li> </ul>	<ul style="list-style-type: none"> <li>• Expliciter la procédure utilisée pour réaliser la figure.</li> </ul>

*Les situations ci-dessous peuvent être mises en œuvre selon les modalités décrites précédemment. Seules les nouvelles idées sont mentionnées.*

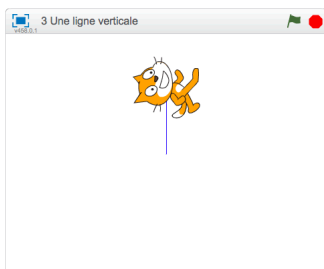
#### E. Situation 5 : réalisation d'une figure simple (2)

##### 1. Objectif(s) et apprentissage(s) technique(s) :

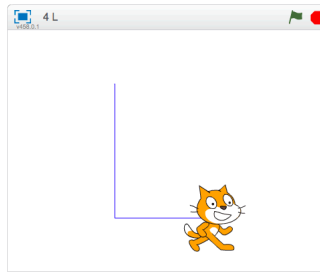
- Comprendre l'importance de définir la position et l'orientation de départ du mobile.

##### 2. Déroulement succinct : 30 minutes

- Projeter d'abord la capture d'écran et laisser les élèves chercher la solution au problème.
- La vidéo ralentie peut être utilisée pour montrer qu'avant le premier déplacement, il y a orientation initiale du mobile.




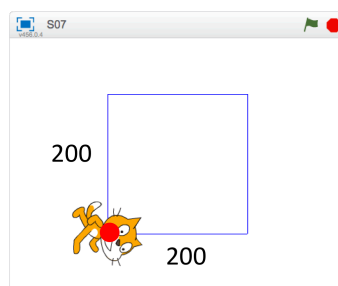
- Prolongement




## F. Situation 6 : Le carré et l'instruction répétition

### 1. Objectif(s) et apprentissage(s) technique(s) :

- Identifier les instructions qui se répètent ;
- Comprendre et utiliser l'instruction .



### 2. Déroulement succinct : 30 minutes

- Poser le problème aux élèves ;
- Faire identifier les instructions qui se répètent ;
- Utiliser l'instruction .

## G. Situations 7 et suivantes.

Proposer aux élèves des défis faisant appel aux angles droits et angles plats.