

# FMTTN P5-P6-S1 Algorithme, logigramme et programmation

## Des logigrammes pour réfléchir avant de programmer

### Objectifs

- Préparer un programme à l'aide de logigrammes
- Exercer plusieurs compétences en programmation
- Faire le lien avec des compétences mathématiques.

Les attendus du référentiel repris dans le programme sont (plus que ) rencontrés.

### Compétences exercées

Programmation et algorithmique:

- Le logigramme, l'algorithme
- Des boucles
- Des conditions
- Une variable et un champ de valeur booléenne (vrai-faux), etc.

Mathématiques :

- Des déplacements simples à l'aide de repères dans le plan orthonormé
- Un test binaire : Si ... alors
- Les variables

## 1 Activité déconnectée / préparation du logigramme

### 1.1 Organisation et mise en place de l'activité

- 1 période.
- Mettre les élèves par groupes de deux.

### 1.2 Proposition de déroulement

#### L'enseignant assigne une mission aux élèves

« Je vais jouer devant vous un petit jeu (très simple...). Votre mission est de programmer ce jeu à l'identique. Observez bien ce qu'il se passe quand je joue et notez un maximum d'informations. Vous en aurez besoin pour la suite. »

Le programme est disponible ici : <https://scratch.mit.edu/projects/889898724>

#### Les élèves préparent des logigrammes

Les élèves doivent noter sur une feuille une succession d'actions ou d'opérations que chaque objet réalise. Si les élèves n'ont jamais réalisé de logigramme, il est préférable de ne pas leur imposer

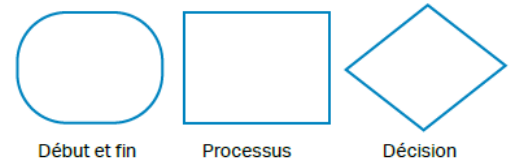
d'utiliser les symboles recommandés par le référentiel et de ne pas utiliser le mot logigramme à ce stade. L'enseignant présentera ces symboles à l'étape suivante.

L'enseignant suggère ceci aux élèves :

1. Identifier les objets qui se déplacent et décrire précisément ces déplacements.  
Il y a trois objets qui se déplacent : le poussin et les deux œufs.
2. Identifier les objets qui changent d'apparence et décrire ces changements, à quelles conditions et à quels moments ils ont lieu.

### L'enseignant et les élèves réalisent les logigrammes

L'enseignant présente les formes à utiliser pour les logigrammes. Voici les formes suggérées par le référentiel.



Je propose d'affiner un peu :

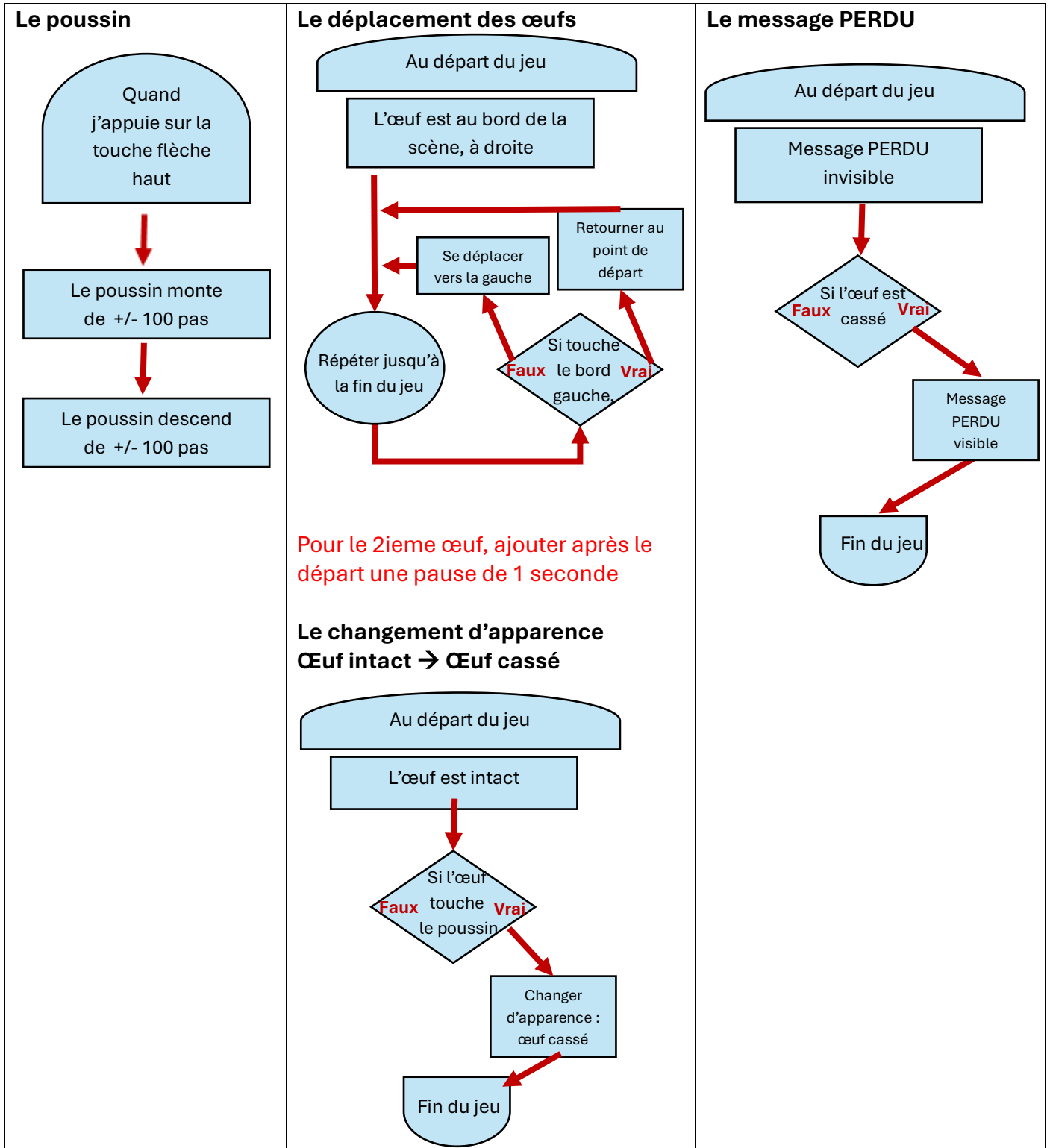
	<p><b>Début et fin</b> Le début d'un processus est généralement déclenché par un événement :</p> <ul style="list-style-type: none"> <li>- Quand j'appuie sur un bouton, ...</li> <li>- Quand j'ai mis mon tablier, je commence ma préparation culinaire</li> <li>- Quand je mets le contact de ma voiture...</li> </ul>
	<p><b>Une action</b>, un déplacement, un mouvement, un changement d'apparence, effectuer un calcul, accélérer (ma voiture...), mélanger deux ingrédients...</p>
	<p><b>Décision</b> Généralement, il s'agira d'une condition.</p> <ul style="list-style-type: none"> <li>- Si le feu est rouge, je m'arrête</li> <li>- Si je n'ai pas de sel, j'ajoute un peu de bouillon</li> <li>- Si j'ai collecté assez de points, je peux passer au niveau de jeu suivant....</li> </ul>
	<p><b>Boucle</b> Répéter un certain nombre de fois une ou plusieurs actions....</p>

### 1.3 Les logigrammes pour le jeu du poussin

Voici une proposition de logigrammes à élaborer avec les élèves avant qu'ils se mettent à programmer le jeu du poussin.

Il faudra insister sur le fait que pour programmer une application, on va généralement écrire plusieurs programmes qui « tournent » simultanément pour un ou plusieurs objets.

Dans le cas qui nous occupe, 4 objets auront leurs propres programmes : le poussin, les deux œufs et un message qui apparaît à la fin du jeu.



On pourrait encore ajouter dans ces logigrammes la gestion de la variable score.

**Attention !** Le but d'un logigramme n'est pas de préfigurer le futur programme mais de décrire un processus ou un ensemble d'actions ( l'algorithme).

## 2 La programmation du jeu : les cartes

On présume que les élèves sont déjà familiarisés avec Scratch et/ou sont passés par l'Activité 1.

La plupart des figures ci-dessous sont extraites des cartes.

Rappeler le principe :

- Une carte défi est au recto et au verso, les consignes.
- On peut créer, proposer des variantes à une condition : avoir a fini tout le jeu de cartes et montré au professeur un programme qui fonctionne.

Ce dernier vérifie que les élèves ont compris à quoi servent les différentes procédures.

### 2.1 Cartes 3-4

#### Charger les sprites et l'arrière-plan

Choix de l'arrière plan et des sprites : rien de nouveau.

- Veiller à ce que les élèves ne chargent qu'un œuf, le deuxième sera obtenu ultérieurement par duplication.
- Vérifier que les élèves comprennent bien que chaque sprite va avoir son propre code.

#### Placer le poussin au milieu de la scène, l'initialisation

On introduit ici les déplacements à l'aide d'un plan ou repère orthonormé. L'élève est invité à placer le poussin sur le sol de couleur brune et au milieu de la scène sur une ligne horizontale. Sur l'axe horizontal (X), on compte de 0 à 240 pas vers la droite et de 0 à -240 vers la gauche. La largeur de la scène est donc de 480 pas. À ce stade, il peut placer le poussin en suivant les consignes et dans même avoir compris. On fera le point sur l'utilisation du repère orthonormé plus tard.

**Place le poussin au milieu de la scène.**

Observe les **coordonnées X et Y** et **1** remplace la valeur de X par 0 et la valeur de Y par -122.



The image shows a Scratch stage with a chick sprite on a brown ground. A coordinate system is overlaid with X ranging from -240 to 240 and Y at -122. A red circle with the number 1 highlights the X coordinate field in the sprite's properties.

## 2.2 Cartes 5-6

### Le poussin saute et les œufs défilent

Deux mouvements sont introduits : un mouvement vertical et un mouvement horizontal.

Insister auprès des élèves pour qu'ils fassent les essais avec les 4 blocs de déplacement. Ils peuvent le faire aussi bien avec l'œuf qu'avec le poussin. Ils pourront supprimer ces 4 blocs quand ils auront terminé.

### Essaie les blocs avant de les utiliser !



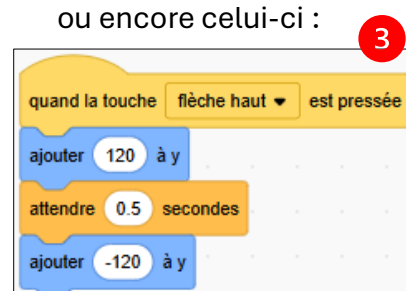
Tu vas devoir utiliser les blocs **ajoute ... à x** et **ajoute ... à y** pour programmer les sauts du poussin et la glissade de l'œuf de gauche à droite.

Place ces blocs comme indiqué dans la fenêtre des scripts du poussin et essaie-les en cliquant dessus. Modifie les valeurs et essaie à nouveau. Remarque les signes négatifs... Ajouter -20 à x pourrait aussi se lire Retirer 20 à x...



### Saut du poussin

On peut demander aux élèves pourquoi le code proposé est celui-là et pas celui-ci :



Le mieux est de tester les trois et comparer

### Explication :

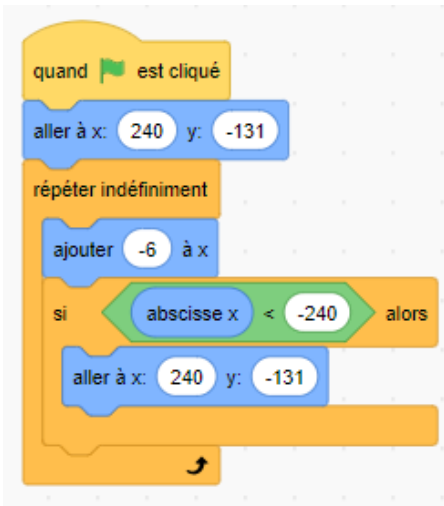
Script 2 : les deux instructions **ajouter 120 à y** et **ajouter -120 à y** vont être effectuées mais cela ira trop vite pour que l'utilisateur perçoive le mouvement. On peut le vérifier en ajoutant une pause 3 mais comme tel, le mouvement sera peu réaliste.

Pour avoir un mouvement fluide et un saut plus réaliste, il faut afficher plusieurs images sur le temps de la montée et de la chute du poussin 1, ce que l'on fait avec le code proposé :



Pour information : Scratch a prévu que les instructions qui se trouvent dans une boucle sont envoyées à l'écran **30 fois par seconde**<sup>1</sup>.

<sup>1</sup> Cela correspond au taux de rafraîchissement de nos écrans d'ordinateurs ou de TV. Les taux de rafraîchissement les plus courants sur nos écrans sont 25 FPS ou 30 FPS. FPS : Frame per second.



## L'œuf défile de gauche à droite

Dans ce code, il y a trois nouveautés : le test **si... alors**, le bloc opérateur  $< -240$  et le bloc de valeur **abscisse x** qui renvoie la position en X ( abscisse) de l'œuf.

En général, les jeunes comprennent assez facilement comment fonctionne ce test qui peut se lire aisément comment on lit une phrase : « Si l'abscisse X de l'œuf est inférieure à -240, alors... ». Pas besoin d'en dire plus à ce stade...

Mais s'il le juge utile, l'enseignant.e peut faire cette démonstration :

- Placer ces blocs **abscisse x**  $< -240$  dans l'éditeur de code de l'œuf et placer l'œuf plus ou moins au milieu de la scène
- Cliquer successivement sur le premier : puis sur le second . Lire les valeurs renvoyées.
- Déplacer l'œuf à  $X = -245$  en tapant cette valeur dans le champ x : des propriétés et refaire le même test.



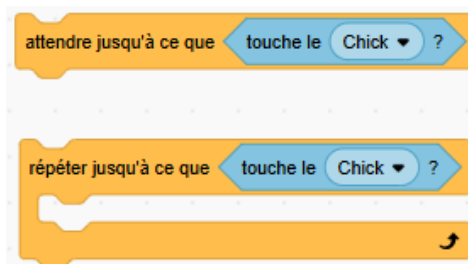
Le bloc vert est un bloc booléen, il ne peut renvoyer que deux valeurs : Vrai ou Faux.

## 2.3 Cartes 7-8 : le poussin casse l'œuf si...

Un deuxième script<sup>2</sup> est ajouté pour l'œuf.

Le bloc **attendre jusqu'à ce que ...** est une boucle qui n'effectue aucune action, elle attend que la condition soit rencontrée pour passer à l'instruction suivante.

Ces deux instructions ont le même effet :



La page 8 propose une nouvelle fois de tester un bloc booléen. Il est important d'inciter les jeunes à tester des blocs isolés pour comprendre leurs effets ou les valeurs qu'ils contiennent.

## 2.4 Cartes 9-10 : l'ajout du deuxième œuf est une formalité...

En dupliquant le premier œuf comme indiqué à la page 10, on duplique aussi son code, ce qui fait gagner du temps.

<sup>2</sup> Le terme script en programmation est utilisé pour désigner une série d'instructions ou une partie de programme.  
J.P. Bihin, décembre 2024

Il ne reste plus alors qu'à ajouter une petite pause avant de laisser le deuxième œuf démarrer et éviter qu'ils soient superposés.

## 2.5 Cartes 11-12 : ajouter un message « Perdu »

Pas de commentaire particulier. Les explications fournies par la carte sont assez détaillées.

Un nouveau bloc apparaît : **envoyer à tous....** (un message). Il fait la paire avec **quand je reçois ...** (un message). Son rôle est d'envoyer un message qui pourra déclencher des scripts et ce, pour tous les sprites. Dans ce cas, quand un œuf se casse, il envoie le message « œuf cassé » qui, une fois intercepté par le sprite **Annonce** affichera « Perdu » et terminera le jeu.

## 3 Mathématiques : le repère orthonormé

L'utilisation d'un repère orthonormé (axe X des abscisses et Y des ordonnées) est au programme de la première secondaire. Cela n'empêche pas de l'introduire pour des élèves de P5-P6. En programmant avec Scratch, ils apprendront en pratiquant.

Les nombres négatifs seront une source de difficulté. On peut expliquer aux élèves qu'ils n'auront pas besoin de calculer avec des nombres négatifs puisque c'est Scratch qui le fera. Ils apprendront à lire les coordonnées, comprendront qu'ajouter un nombre négatif à X revient à aller vers la gauche sur l'axe horizontal. Ils feront vite le lien avec la lecture sur un thermomètre.

Si vous souhaitez faire le point et exercer les élèves à se repérer dans le plan orthonormé proposé par Scratch, je vous propose deux petits ateliers.

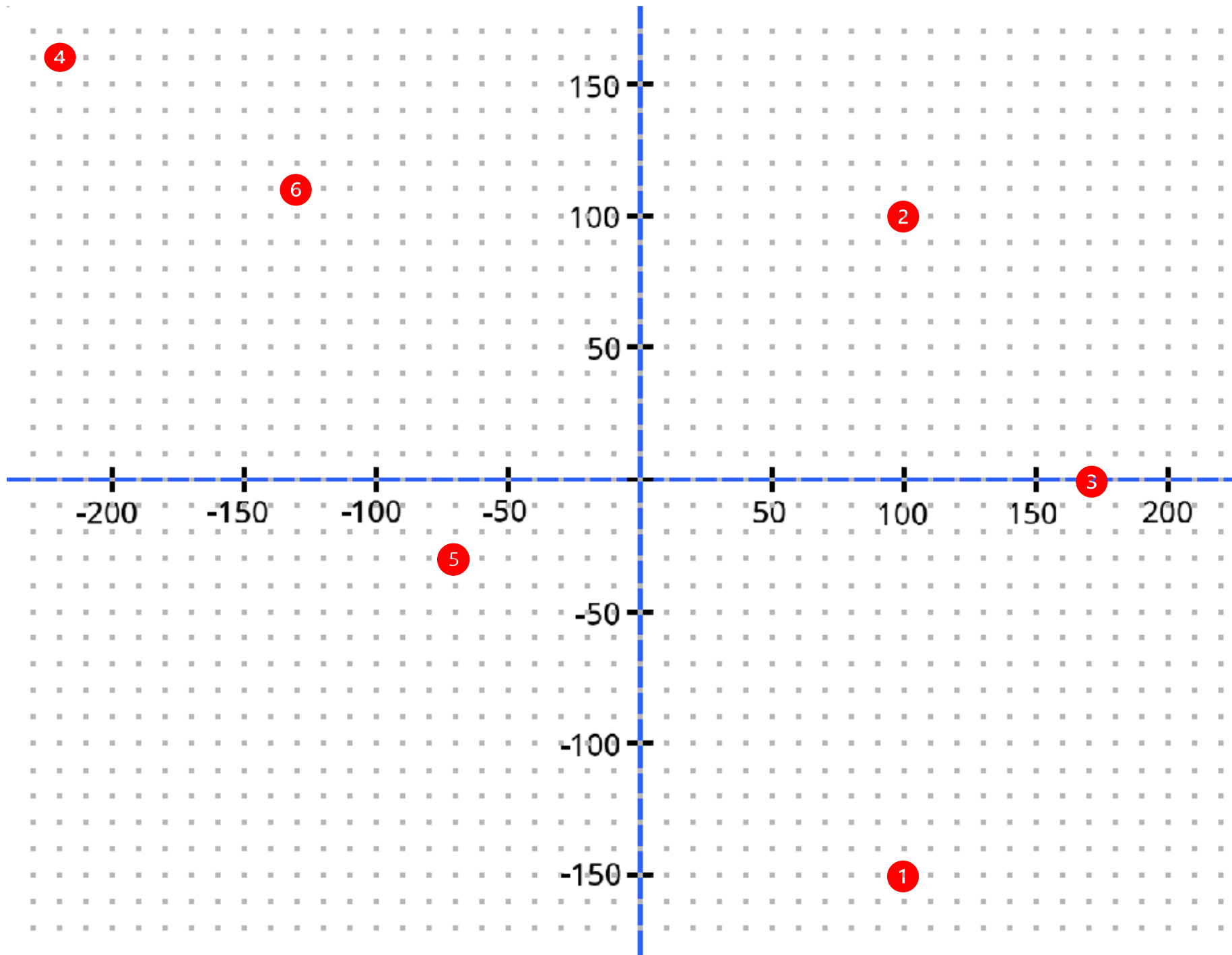
### 3.1 S'entraîner sur la scène de Scratch

Voici un petit jeu sur Scratch : <https://scratch.mit.edu/projects/649398441> . À jouer en plein écran !

### 3.2 S'entraîner sur papier

En annexe, une grille qui représente la scène de Scratch. L'enseignant note quelques paires de coordonnées au tableau en les numérotant. L'élève place chaque numéro sur le point correspondant à chaque paire de coordonnées.

Attention : avertir les élèves que les nombres sont tous arrondis à la dizaine.



Complète avec les bonnes coordonnées

	X	Y
1	100	-150
2	.....	.....
3	.....	.....
4	.....	.....
5	.....	.....
6	.....	.....

Place ces points dans le plan

	X	Y
a	0	30
b	-200	-50
c	70	70
d	70	-40
e	240	-180
f	-240	-240