

## 1 Introduction

Les travaux pratiques (TP) n° 1 à 5 sont conçus pour une utilisation sur un poste de travail avec un environnement Scratch 3.

- Le premier TP revisite le programme de cinquième en proposant des tracés de figures par déplacement du lutin et en utilisant le « stylo » de Scratch.

Il s'inspire du sujet de mathématiques du brevet donné en métropole le 2 septembre 2018.

- Le second TP continue le travail sur le déplacement et les tracés de figures par modification ou création de scripts.

Il s'inspire du sujet de mathématiques du brevet donné en Nouvelle-Calédonie le 2 décembre 2018.

- Le troisième TP propose des rappels sur la programmation événementielle et le contrôle du déplacement d'un lutin à l'aide des flèches du clavier et conduit à l'analyse d'un script.

Il s'inspire du sujet de mathématiques du brevet donné en Amérique du Nord le 7 juin 2017.

- Le quatrième TP aborde les variables informatiques numériques créées par l'utilisateur sous l'angle d'un programme de calcul.

Il s'inspire du sujet de mathématiques du brevet donné en Polynésie le 23 juin 2017.

- Le cinquième TP poursuit le travail sur les variables en détaillant leur usage et l'évolution de leur valeur dans différents scripts.

## TP n° 1 : déplacement et gestion du « style »

Sam a écrit le programme ci-dessous qui permet de tracer un rectangle.

Ce programme comporte deux variables, **Longueur** et **Largeur**, qui représentent les dimensions du rectangle.

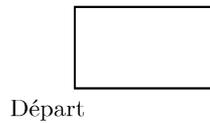


Figure 1

### Script A

```

Quand la touche a est pressée
  aller à x : 0 y : 0
  s'orienter en direction de 90
  effacer tout
  mettre Longueur à 50
  mettre Largeur à 30
  rectangle
  
```

### Bloc rectangle

```

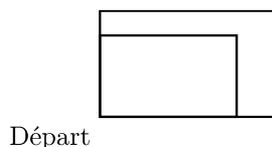
définir rectangle
  stylo en position d'écriture
  répéter ... fois
    avancer de Longueur pas
    tourner de ... degrés
    avancer de ... pas
    tourner de ... degrés
  relever le stylo
  
```

### Script B

```

1 Quand la touche b est pressée
2 effacer tout
3 mettre Longueur à 50 pas
4 mettre Largeur à 30
5 aller à x : 0 y : 0 pasa
6 s'orienter en direction de 90
7 rectangle
8 attendre 1 secondes
9 mettre Longueur à Longueur * ...
10 mettre Largeur à ... * ...
1 .....
  
```

1. Complétez le bloc « rectangle » ci-dessus avec des nombres et des variables pour que le script fonctionne.
2. Créez l'ensemble du programme dans Scratch 3 et sauvegardez-le sous le nom « *rectangle\_Nom\_Classe.sb3* » en remplaçant « *Nom* » par votre nom et « *Classe* » par votre classe.
3. Sam n'a pas terminé le script B qui doit produire la figure ci-dessous. Il s'agit d'ajouter un nouveau rectangle dont la longueur et la largeur sont chacune multipliée par 1,2 par rapport au précédent.  
Complétez les lignes 9 et 10 ainsi que l'instruction manquante en ligne 11.
4. Sauvegardez votre programme puis envoyez-le à votre professeur via le cahier de texte numérique.



## TP n° 2 : analyser et modifier un programme

Le chat  indique la position de départ. Le côté d'un carreau mesure 20 unités.

Ce bloc utilisateur « Initialiser » place le chat dans la position de départ, orienté vers la droite.

1. Ouvrez le programme « *TP\_4ACG\_tourner.sb3* » dans Scratch 3 et sauvegardez-le sous le nom « *tourner\_Nom\_Classe.sb3* », en remplaçant « *Nom* » par votre nom et « *Classe* » par votre classe.
2. Recopiez le script A et exécutez-le. Indiquez ci-contre le numéro de la figure tracée : . . . .
3. Créez une variable *D*, recopiez le script B et exécutez-le. Indiquez ci-contre le numéro de la figure tracée : . . . .
4. Créez et exécutez deux nouveaux scripts de façon à obtenir les deux figures restantes.
5. Sauvegardez votre programme et envoyez-le à votre professeur via le cahier de texte numérique.

Script A



Script B

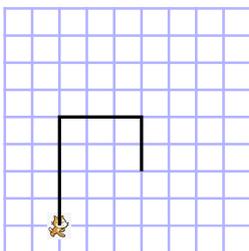
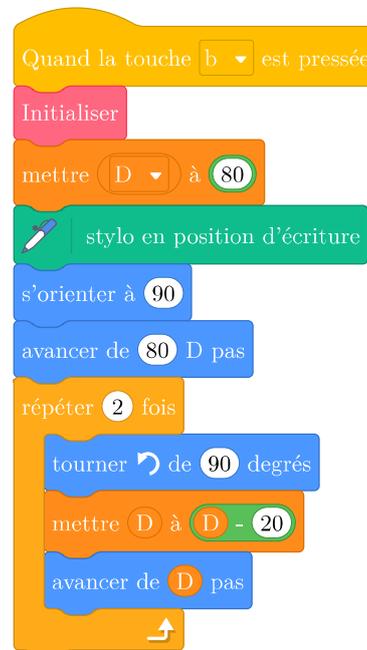


Figure 2

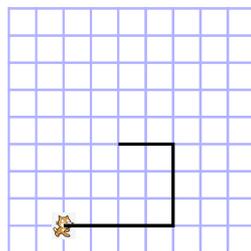


Figure 3

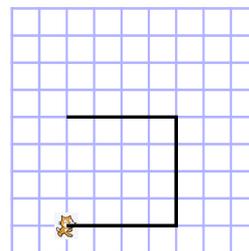


Figure 4

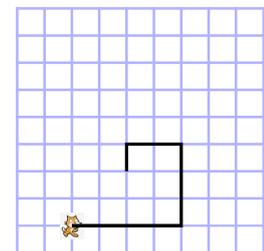
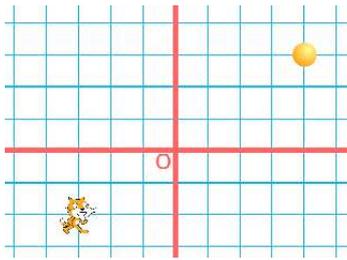


Figure 5

### TP n° 3 : programmation événementielle

L'image ci-dessous représente la position obtenue en cliquant sur le drapeau vert .



**Le but du jeu est de positionner le chat sur la balle.**

L'arrière-plan est un quadrillage d'éléments espacés de 30 unités.

Attention, dans le bloc bleu , la variable  $x$  n'est pas une variable créée par l'utilisateur : elle désigne l'abscisse du lutin.

1. Donner les coordonnées du centre de la balle représentée dans cette position : (...;...)

2. Dans cette question, le chat<sup>1</sup> est dans la position obtenue en cliquant sur le drapeau vert.

Voici les scripts du lutin « chat ».

a. Créez et testez le programme.

b. Expliquez pourquoi le chat ne revient pas à sa position de départ si le joueur appuie sur la touche  $\rightarrow$  puis sur la touche  $\leftarrow$ .

.....  
 .....  
 .....  
 .....

c. Le joueur appuie sur la succession de touches suivantes :  $\rightarrow \rightarrow \uparrow \leftarrow \downarrow$ .

Donner les coordonnées  $x$  et  $y$  du chat après ce déplacement :

(...;...)

d. Parmi les propositions de succession de touches ci-dessous, laquelle permet au chat d'atteindre la balle ?

e. Sauvegardez votre programme puis envoyez-le à votre professeur via le cahier de texte numérique.

```

    Quand drapeau vert est cliqué
    Mettre la taille à 35 % de la taille initiale
    aller à x : -90 y : -60

    Quand la touche flèche gauche est pressée
    ajouter -30 à x

    Quand flèche droite est pressée
    ajouter 60 à x

    Quand flèche haut est pressée
    ajouter 60 à y

    Quand flèche bas est pressée
    ajouter -30 à y

    Quand n'importe laquelle est pressée
    si touche le Ballon ? alors
    dire Je t'ai attrapé pendant 2 secondes
    Initialiser
    
```

Déplacement 1	Déplacement 2	Déplacement 3
$\rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \uparrow \uparrow \uparrow \uparrow$	$\rightarrow \rightarrow \rightarrow \uparrow \uparrow \uparrow \rightarrow \downarrow \leftarrow$	$\uparrow \rightarrow \uparrow \rightarrow \uparrow \rightarrow \rightarrow \downarrow \downarrow$

3. Que se passe-t-il quand le chat atteint la balle ?

.....

## TP n° 4 : programme de calcul

On considère le programme de calcul suivant :

- Choisir un nombre.
- Le multiplier par - 4.
- Ajouter 5 au résultat.

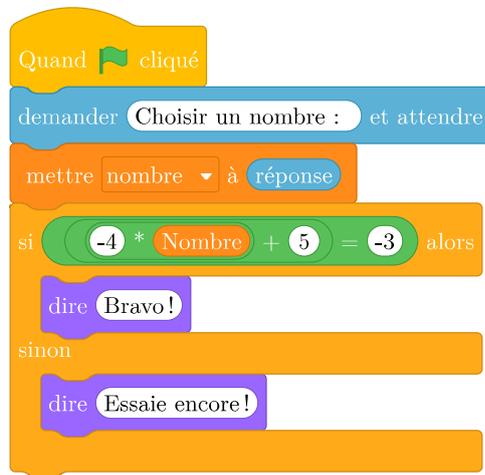
1. Vérifier que lorsque l'on choisit +10 avec ce programme, on obtient -35.

...

2. Salomé exécute un script où :

- l'utilisateur doit créer la variable (en orange foncé) **Nombre** ;
- l'utilisateur *de doit pas* créer la variable (en bleu) **réponse** : c'est une variable spéciale, gérée par Scratch, qui contient ce que l'utilisateur a tapé et validé au clavier après exécution d'un bloc **demande ..... et attendre**.

Saisissez et exécutez ce script :



a) Quelle sera la réponse du lutin si elle choisit le nombre 13 ? ...

b) Quelle sera la réponse du lutin si elle choisit le nombre -3 ? ...

3. Le programme de calcul peut se traduire par l'expression littérale  $(-4x + 5)$  avec  $x$  représentant le nombre choisi.

Résoudre l'équation suivante :  $-4x + 5 = -3$ .

...

4. À quelle condition, portant sur le nombre choisi, est-on certain que la réponse du lutin sera « Bravo » ?

...

5. Sauvegardez votre programme et envoyez-le à votre professeur via le cahier de texte numérique.

**TP n° 5 : variables**

**Exercice 1**

```

1 Quand la touche a est pressée
2 mettre x à 10
3 mettre A à 3 * x
4 mettre A à A + 2
5 mettre A à 4 * A
    
```

a. Parmi ces expressions littérales, cochez celle qui permet d'exprimer  $A$  en fonction de  $x$  dans le script ci-dessus :

- $4 \times x + 2 \times 3 \times x$
- $(3 \times x + 2) \times 4$
- $3 \times x + 2 \times 4$

b. Complétez le tableau donnant la valeur de chaque variable à la fin du bloc :

Numéro du bloc	x	A
1		
2	10	
3	...	...
4	...	...
5	...	...

c. Programmez ce script dans Scratch. Modifiez le bloc numéro 2 afin que  $x$  prenne la valeur 345 et indiquez la valeur de  $A$  obtenue à la fin de l'exécution du script.

.....

**Exercice 2**

a. Citez les variables utilisées dans le script ci-dessous. ....

b. Exprimez  $A$  en fonction de  $R$ . ....

c. Programmez le script et notez la valeur de  $A$  obtenue. ....

d. Interprétez le résultat.

.....

```

1 Quand la touche b est pressée
2 mettre pi à 3.14
3 mettre R à 5
4 mettre A à 2 * pi * R
    
```

**Exercice 3**

Le volume  $V$  d'un cône de hauteur  $h$  et dont la base est un disque de rayon  $r$  est donnée par :

$$V = \frac{\pi \times r^2 \times h}{3}$$

a. Dans Scratch, créez les variables  $pi$ ,  $r$  et  $h$  puis complétez le script ci-dessous afin de calculer  $V$  pour un rayon  $r = 4$  m et une hauteur  $h = 3$  m.

```

1 Quand la touche c est pressée
2 mettre pi à 3.14
3 mettre r à 4
    
```

b. Notez la valeur de  $V$  obtenue : .....

**Exercice 4**

```

1 Quand la touche d est pressée
2 mettre B à (x + 7) * (4 - x)
    
```

a. Exprimez  $B$  en fonction de  $x$  : .....

b. Programmez ce script, et complétez-le de façon à afficher la valeur de  $B$  pour  $x = 35$ .

$B =$  .....

**Sauvegardez votre programme et transmettez-le par le cahier de texte numérique.**

## 2 Les variables de l'utilisateur

### 2.1 Une variable utilisateur doit être déclarée

Une variable se comporte comme une boîte qui contient et conserve une information. Avec Scratch, une variable doit impérativement être déclarée avant toute utilisation. Lors de cette déclaration, l'utilisateur doit préciser le nom et la portée de la variable, à l'aide de cette boîte de dialogue :



#### Nom d'une variable

Le nom d'une variable peut contenir des lettres majuscules ou minuscules, des chiffres, mais aussi... des caractères blancs, des caractères de ponctuation et des caractères spéciaux. Profitons de cette liberté pour donner à chaque variable un nom compréhensible et si possible évocateur. Par exemple, concernant le rayon d'un cercle, on pourra déclarer une variable :

Rayon

Précisons enfin que Scratch distingue les majuscules et les minuscules lorsqu'il s'agit de nommer une variable ; ainsi, les trois variables suivantes sont différentes.

SOMME Somme somme

### 2.2 Portée d'une variable

La portée d'une variable définit la partie du programme pour laquelle cette variable est accessible et, lors de la déclaration d'une variable, il faut préciser sa portée :

- soit elle est accessible par tous les lutins ;
- soit elle est accessible uniquement par le lutin au sein de laquelle elle est créée.

Le premier choix doit être privilégié par défaut (c'est pourquoi il est présélectionné dans la boîte de dialogue ci-dessus) et le second doit être motivé par une particularité de l'algorithme que l'on veut programmer.<sup>2</sup>

### 2.3 Montrer une variable et sa valeur

#### Montrer et cacher une variable

Scratch permet d'afficher ou de cacher la valeur d'une variable à l'aide des blocs suivants.

<sup>2</sup> Par exemple, dans la programmation d'un jeu à deux joueurs représentés chacun par un lutin, ces deux lutins pourront être dotés d'une variable *Score* qui sera propre à chaque lutin.



Figure 6 – Montrer ou cacher une variable

Déclarons 4 variables  $a, b, c$  et  $d$  et exécutons le script suivant.



Figure 7 – Montrer ou cacher une variable

Les afficheurs des variables  $a, b$  et  $d$  seront alors visibles dans l'interface de Scratch.



Un double-clic sur l'afficheur d'une variable permet de choisir l'un des formats suivants :

- nom de la variable et valeur ;
- valeur seule ;
- nom de la variable, valeur et jauge permettant de modifier cette valeur.



Figure 8 – Trois formats pour un afficheur

### 3 Programmation séquentielle

#### 3.1 Affecter une valeur une variable

L'affectation est une opération élémentaire qui attribue une valeur à une variable.

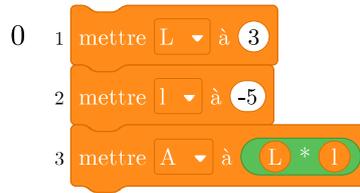
Le bloc suivant<sup>3</sup> affecte à la variable  $x$  la valeur  $-5$  :



3. Scratch propose un seconde bloc permettant d'affecter une variable (« ajouter à ... ») que nous n'utiliserons pas pour l'instant.

### Un premier exemple

Considérons le script suivant, créé pour calculer l'aire  $\mathcal{A}$  d'un rectangle de longueur  $L = 5$  m et de largeur  $l = 3$  m.



Ce script manipule trois variables dont les valeurs évoluent après l'exécution de chaque bloc comme l'indique le tableau suivant.

N° du bloc	L	l	A
1	...		
2	...	...	
3	...	...	...

### Un programme de calcul

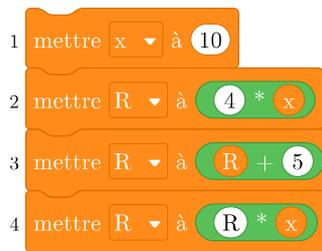
On considère le programme de calcul suivant.

- Choisir un nombre.
- Multiplier ce nombre par 4.
- Ajouter 5.
- Multiplier par le nombre de départ.

Il est associé à l'expression, en fonction de  $x$  :

$$(x \times 4 + 5) \times x$$

On peut lui associer un script Scratch qui calcule  $R$ , le résultat du programme de calcul, en plusieurs étapes lorsque  $x$  prend la valeur 10, par exemple.



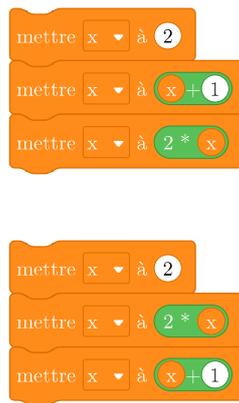
Le tableau suivant indique la valeur de la variable  $x$  après l'exécution de chaque bloc.

N° du bloc	x	R
1	...	
2	...	...
3	...	...

### 3.2 Particularité de l'enchaînement séquentiel

L'enchaînement séquentiel consiste à exécuter des instructions (ou les blocs) les unes à la suite des autres, dans l'ordre indiqué.

Ci-dessous<sup>4</sup> :



À l'exécution :

- Dans le premier script,  $x$  prend pour valeur successivement ...
- Dans le second script,  $x$  prend pour valeur successivement ...

La valeur de  $x$  à la fin de l'exécution n'est donc pas la même pour ces deux scripts.

### 3.3 De l'importance d'initialiser une variable

Prenons par exemple un programme simple, comportant une seule instruction ( $A$  et  $B$  étant deux variables déclarées).



Cette instruction a pour effet d'ajouter 1 à la valeur de  $x$ ... mais la valeur de  $x$  n'est pas précisée au préalable et en toute rigueur, on ne saurait prédire le résultat de l'opération  $x + 1$ .

Il convient donc d'initialiser explicitement  $x$  à la valeur souhaitée, par une instruction d'affectation ou de lecture.

Si la valeur de  $x$  doit être 17 au moment de l'exécution de bloc où l'on affecte  $B$ , un script correct pourrait être celui-ci.

4. Voir DSAP-exemple-execution-sequentielle.sb2



### 3.4 Lire et écrire

La lecture (ou saisie) permet de donner une valeur à une variable au moment où l'instruction de lecture est exécutée. Lors de la programmation, cette valeur pourra provenir d'une saisie au clavier, d'un clic de la souris ou de différents types de capteurs.

Nous nous intéresserons dans ce chapitre à la forme la plus courante, à savoir la lecture d'une valeur saisie par l'utilisateur au clavier.

La mise en œuvre est un peu particulière.



Figure 9 – Lire une valeur « au clavier »

L'instruction « demander ... et attendre » provoque l'affichage de la valeur d'un message sous forme d'une bulle de dialogue associée au lutin.



L'exécution du programme est alors suspendue jusqu'à ce que l'utilisateur entre une valeur puis la valide avec la touche « entrée » du clavier. Cette valeur est alors stockée dans une variable interne et prédéfinie de Scratch, *réponse*, dont le rôle est précisément de conserver la dernière valeur saisie à l'aide de l'instruction « demander ... et attendre ».

L'instruction « mettre  $x$  à *réponse* » permet de donner à  $x$  la valeur saisie.

La valeur de *réponse* est conservée jusqu'à la prochaine exécution d'une instruction « demander ... et attendre » ce qui pourrait permettre, par exemple, des initialisations multiples :



Figure 10 – À éviter !

Cependant, une bonne pratique consiste à stocker dès que possible le contenu de *réponse* dans une variable.