

Pour vous entraîner dans toutes les matières

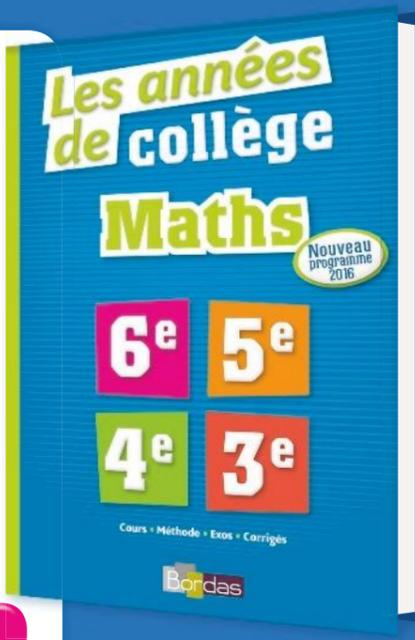
Un cahier complet de 360 pages

L'année de
La collection n°1 depuis 10 ans !

Toutes les matières en un seul cahier, pour s'entraîner et réviser à la maison :

- ✓ L'essentiel des leçons
- ✓ Des méthodes et des exemples
- ✓ Des exercices progressifs
- ✓ Tous les corrigés

Disponible en magasin et sur www.editions-bordas.fr



* Panels Livre GfK - CA 2005-2016 - Segment parascolaire : cumul cahiers de soutien et préparation aux examens, catégorie « tout en un », tous réseaux de distribution.

Un site d'entraînement en ligne

Bordas Soutien scolaire

L'entraînement en ligne du CP à la Terminale

Un site d'exercices et d'entraînement en ligne, pour apprendre à son rythme !

- ✓ Des fiches de cours, des exercices interactifs, des quiz et des vidéos pour s'entraîner sur ordinateur ou sur tablette
- ✓ Les exercices s'adaptent automatiquement à votre niveau
- ✓ Vous pouvez suivre tous vos résultats

À découvrir sur soutien.editions-bordas.fr



© Samuel Berges Photography/Shutterstock.com

ISBN 978-2-04-733445-4



Bordas



COLLECTION
LE
D
A
R
I
A
N
T
E
M
Y
R
I
A
N
T
E

programme
2016

Cahier **CYCLE 4** d'algorithmique



Bordas

SOMMAIRE

Séquence 1 ■ Les instructions

■ Algorithmique débranchée.....	4
■ Je m'entraîne à programmer – Niveau 1.....	6
■ Je m'entraîne à programmer – Niveau 2.....	8
■ Je m'entraîne à programmer – Niveau 3.....	10
■ Je programme pour résoudre des problèmes.....	12
■ Je prépare le brevet des collèves.....	13

Séquence 2 ■ Les boucles

■ Algorithmique débranchée.....	14
■ Je m'entraîne à programmer – Niveau 1.....	16
■ Je m'entraîne à programmer – Niveau 2.....	18
■ Je m'entraîne à programmer – Niveau 3.....	20
■ Je programme pour résoudre des problèmes.....	22
■ Je prépare le brevet des collèves.....	23

Séquence 3 ■ Les variables

■ Algorithmique débranchée.....	24
■ Je m'entraîne à programmer – Niveau 1.....	26
■ Je m'entraîne à programmer – Niveau 2.....	28
■ Je m'entraîne à programmer – Niveau 3.....	30
■ Je programme pour résoudre des problèmes.....	32
■ Je prépare le brevet des collèves.....	33

Séquence 4 ■ Les instructions conditionnelles

■ Algorithmique débranchée.....	34
■ Je m'entraîne à programmer – Niveau 1.....	36
■ Je m'entraîne à programmer – Niveau 2.....	38
■ Je m'entraîne à programmer – Niveau 3.....	40
■ Je programme pour résoudre des problèmes.....	42
■ Je prépare le brevet des collèves.....	43

Séquence 5 ■ Blocs d'instructions

■ Algorithmique débranchée.....	44
■ Je m'entraîne à programmer – Niveau 1.....	46
■ Je m'entraîne à programmer – Niveau 2.....	48
■ Je m'entraîne à programmer – Niveau 3.....	50
■ Je programme pour résoudre des problèmes.....	52
■ Je prépare le brevet des collèves.....	53

Séquence bilan ■ Les projets

Projet 1 ■ La chasse aux clés.....	54
Projet 2 ■ Le singe gourmand.....	56
Projet 3 ■ Air hockey.....	58
Projet 4 ■ Les aventures de Petit Scarabée.....	60
Projet 5 ■ Au clair de la Lune.....	62



Le logiciel **SCRATCH** est un langage intuitif de programmation par blocs, dont l'utilisation à travers ce cahier permettra d'appréhender et comprendre comment mettre en place des algorithmes ! Il permet de réaliser tout type de programmes et, en particulier, des jeux assez facilement. Ce cahier te permettra de progresser tout au long du cycle 4 et de comprendre ainsi comment sont programmés les objets qui t'entourent.

Voici ci-dessous une présentation de l'interface générale de Scratch :

The screenshot shows the Scratch interface with several callout boxes:

- La scène permet d'exécuter le programme.** (The stage allows for program execution.) - Points to the stage area.
- Dans ces menus se trouvent les instructions à faire glisser dans la zone de script. Il y a une couleur de blocs par menu.** (In these menus are the instructions to be dragged into the script area. There is a color for blocks per menu.) - Points to the menu categories.
- Zone de gestion et de création des lutins et/ou arrière-plans.** (Area for managing and creating sprites and/or backgrounds.) - Points to the sprite area.
- C'est dans la zone de script que l'on assemble les instructions du programme.** (It is in the script area that the program instructions are assembled.) - Points to the script area.

Ce logiciel, avec lequel nous allons apprendre à travailler, est téléchargeable et utilisable gratuitement sur :

<http://scratch.mit.edu/>



ALGORITHMIQUE débranchée

Je découvre

Une **séquence d'instructions** est une suite d'actions à exécuter dans un ordre donné.

Exemple 1 : Pour dessiner la lettre **L**, on peut exécuter les instructions suivantes.

- Prendre un crayon rouge
- Tracer un trait vertical vers le bas de longueur 1 cm
- Tracer un trait horizontal vers la droite de longueur 0,6 cm

Dans le logiciel Scratch, la séquence d'instructions ci-contre va dessiner la lettre **L** à l'écran.

```

SCRATCH
quand est cliqué
stylo en position d'écriture
mettre la couleur du stylo à [rouge]
mettre la taille du stylo à [5]
s'orienter à [180]
avancer de [100]
tourner [90] de 90 degrés
avancer de [60]

```

Exemple 2 : Pour tracer un carré rouge, on peut exécuter les instructions suivantes.

- Prendre un crayon rouge
- Tracer un trait vertical vers le bas de longueur 5 cm
- Tracer un trait horizontal vers la droite de longueur 5 cm
- Tracer un trait vertical vers le haut de longueur 5 cm
- Tracer un trait horizontal vers la gauche de longueur 5 cm

Dans Scratch, l'algorithme programmé ci-contre permettra de tracer un carré rouge à l'écran.

```

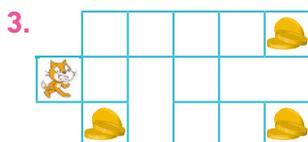
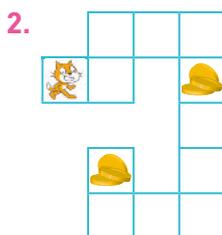
quand est cliqué
effacer tout
stylo en position d'écriture
mettre la couleur du stylo à [rouge]
mettre la taille du stylo à [5]
avancer de [100]
tourner [90] de 90 degrés

```

J'applique

Exercice 1 Le labyrinthe Difficulté ★

Dans chaque cas, aider Scratchy à récupérer le maximum de pièces d'or en utilisant les instructions ◀, ▶, ▲ et ▼ pour se déplacer.



Question 1.

.....

Question 2.

.....

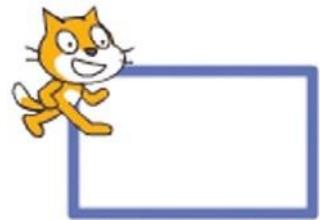
Question 3.

.....

JE M'ENTRAINE à programmer

Exercice 4 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui permet d'obtenir un rectangle comme représenté ci-contre dans le logiciel Scratch. (On considère que le stylo est déjà en position d'écriture, en bleu, de taille 5.)



Programme ①

```

quand [drapeau] est cliqué
  avancer de 100
  tourner [90] de 90 degrés
  avancer de 60
  tourner [90] de 90 degrés
  avancer de 100
  tourner [90] de 90 degrés
  avancer de 60
  tourner [90] de 90 degrés
  
```

Programme ②

```

quand [drapeau] est cliqué
  avancer de 100
  tourner [60] de 60 degrés
  avancer de 90
  tourner [60] de 60 degrés
  avancer de 100
  tourner [60] de 60 degrés
  avancer de 90
  tourner [60] de 60 degrés
  
```

Programme ③

```

quand [drapeau] est cliqué
  avancer de 100
  tourner [100] de 100 degrés
  
```

2. Expliquer brièvement pourquoi les deux autres programmes ne permettent pas de dessiner un rectangle.

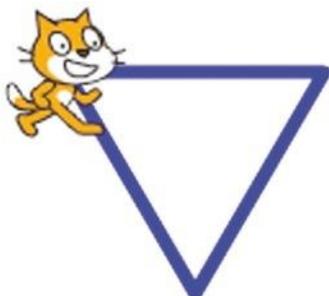
.....

.....

.....

Exercice 5 Je complète un programme

Compléter les zones blanches dans le programme ci-contre afin que celui-ci permette d'obtenir un triangle équilatéral de côté 100 pixels.



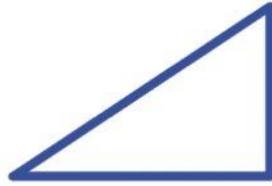
```

quand [drapeau] est cliqué
  effacer tout
  stylo en position d'écriture
  avancer de .....
  tourner [ ] de ..... degrés
  avancer de .....
  tourner [ ] de ..... degrés
  avancer de .....
  tourner [ ] de ..... degrés
  
```




Exercice 10 Un triangle rectangle

En utilisant principalement les instructions proposées, écrire un programme qui permette de tracer un triangle rectangle dont les côtés de l'angle droit mesurent 100 et 150 pixels.



Area for writing code instructions, consisting of a series of horizontal dotted lines.

Instructions principales

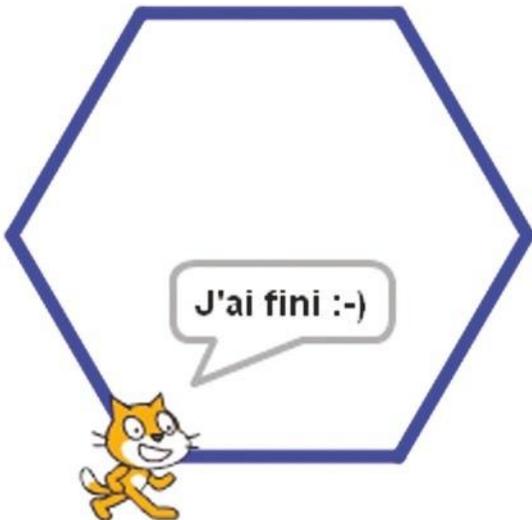
tourner ↻ de 90 degrés

avancer de 100

aller à x: 0 y: 0

Exercice 11 Hexagone

En utilisant principalement les instructions proposées, écrire un programme qui permette de tracer un hexagone en faisant une pause de une seconde entre chaque côté et en disant « J'ai fini :-) » à la fin du tracé.



Area for writing code instructions, consisting of a series of horizontal dotted lines.

Instructions principales

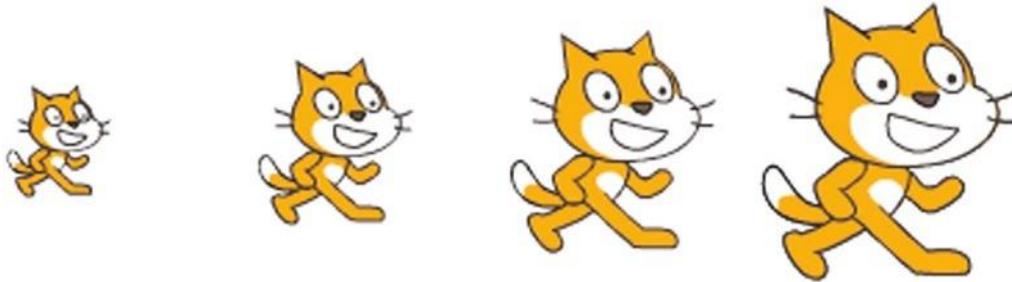
tourner ↻ de 60 degrés

attendre 1 secondes

dire J'ai fini :-)

JE M'ENTRAINE à programmer

Exercice 12 Je choisis le bon programme



1. Parmi les trois programmes suivants, cocher celui qui permet d'obtenir le dessin ci-dessus dans Scratch.

Programme ①

Programme ②

Programme ③

```

quand est cliqué
mettre à 100 % de la taille initiale
aller à x: -200 y: 0
effacer tout
estampiller
avancer de 100
ajouter -20 à la taille
estampiller
avancer de 100
ajouter -20 à la taille
estampiller
avancer de 100
ajouter -20 à la taille
estampiller

```

```

quand est cliqué
mettre à 40 % de la taille initiale
aller à x: -200 y: 0
effacer tout
estampiller
avancer de 100
ajouter 20 à la taille
avancer de 100
ajouter 20 à la taille
avancer de 100
ajouter 20 à la taille
estampiller

```

```

quand est cliqué
mettre à 40 % de la taille initiale
aller à x: -200 y: 0
effacer tout
estampiller
avancer de 100
ajouter 20 à la taille
estampiller
avancer de 100
ajouter 20 à la taille
estampiller
avancer de 100
ajouter 20 à la taille
estampiller

```

2. Expliquer brièvement ce que permettent d'obtenir les deux autres programmes.

.....

.....

Exercice 13 À la belle étoile

Écrire ci-contre un programme qui permette de réaliser le motif ci-dessous.



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Coup de pouce

Utiliser les points de coordonnées suivantes : (0 ; 90) ; (-120 ; 0) ; (120 ; 0) ; (75 ; -150) et (-75 ; -150).

Je programme pour résoudre des problèmes



Exercice 16 À quelle distance ?

Dans Scratch, dupliquer le lutin qui apparaît par défaut, puis saisir pour chacun des lutins les deux programmes ci-dessous.



```

quand [drapeau] est cliqué
  cacher
  aller à x: 0 y: 0
  
```



```

quand [drapeau] est cliqué
  mettre à 30 % de la taille initiale
  aller à x: 0 y: 0
  stylo en position d'écriture
  effacer tout
  s'orienter à 90
  avancer de 40
  s'orienter à 0
  avancer de 30
  dire distance de Lutin1
  aller à x: 0 y: 0
  s'orienter à 90
  relever le stylo
  aller à x: -100 y: 0
  
```

1. Préciser ce que fait le programme du « Lutin1 ».

.....

2. Préciser ce que fait le programme du « Lutin2 ».

.....

.....

.....

.....

3. En modifiant quelques paramètres dans ces programmes, donner dans chaque cas la longueur de l'hypoténuse d'un triangle rectangle dont les côtés de l'angle droit mesurent :

a. Côtés de l'angle droit : 50 pixels et 120 pixels → Hypoténuse :

b. Côtés de l'angle droit : 90 pixels et 120 pixels → Hypoténuse :

c. Côtés de l'angle droit : 48 pixels et 55 pixels → Hypoténuse :

d. Côtés de l'angle droit : 65 pixels et 72 pixels → Hypoténuse :

e. Côtés de l'angle droit : 99 pixels et 168 pixels → Hypoténuse :

4. Trouver les valeurs des côtés de l'angle droit du triangle rectangle ci-contre dont l'hypoténuse mesure 203 pixels, sachant que les deux côtés de l'angle droit ont moins de 10 pixels d'écart.

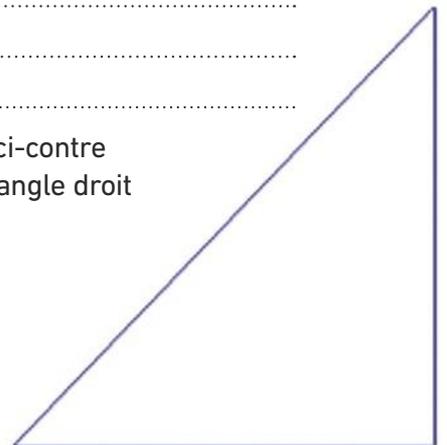
.....

.....

.....

.....

.....



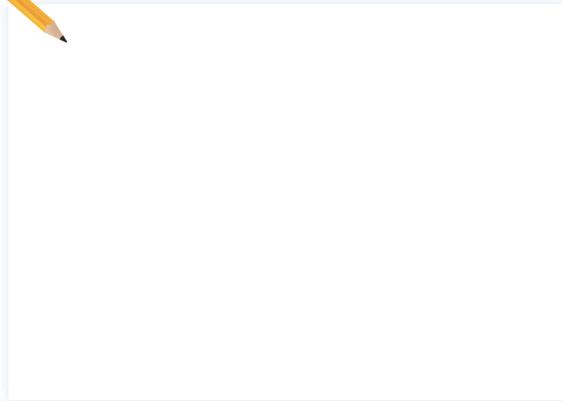
Je prépare le brevet des collèves

20 min

Notions : Instructions

Suivre pas à pas le programme proposé au stylo sur une feuille de brouillon.

- ▶ 1. Voici un programme qui permet de dessiner un chiffre à la façon des horloges digitales.
a. Faire ce dessin à main levée.



```
quand est cliqué  
cacher  
aller à x: 0 y: 0  
s'orienter à 90  
effacer tout  
stylo en position d'écriture  
avancer de -50  
tourner de 90 degrés  
avancer de 50  
tourner de 90 degrés  
avancer de 50  
tourner de 90 degrés  
avancer de 50  
tourner de 90 degrés  
avancer de 50
```

- b. De quel chiffre s'agit-il ?

.....

De même, suivre pas à pas le programme en remplaçant cette instruction.

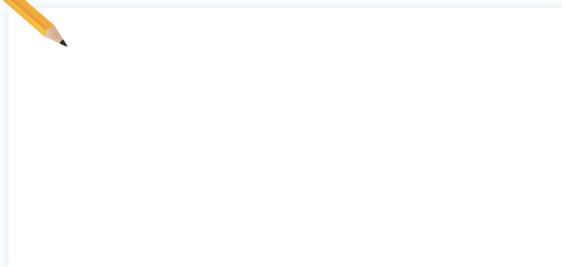
- ▶ 2. a. Quel dessin obtiendrait-on si l'on remplace dans le programme l'instruction

```
tourner de 90 degrés
```

par l'instruction :

```
tourner de 90 degrés ?
```

- b. Le dessiner.



Il faut se demander comment on peut obtenir un 6 à partir d'un 5 en s'interrogeant sur la position du stylo à la fin du programme donné.

- ▶ 3. Quelles instructions doit-on rajouter à la fin du programme pour que celui-ci dessine le chiffre 6 à la façon des horloges digitales ?

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

Récrire le programme et le suivre pas à pas.

- ▶ 4. Quel chiffre obtient-on si l'on change le sens de rotation des quatre

instructions « Tourner... », c'est-à-dire en remplaçant `tourner de 90 degrés`

par `tourner de 90 degrés` et en remplaçant `tourner de 90 degrés`

par `tourner de 90 degrés` ?

.....



ALGORITHMIQUE débranchée

Je découvre

Une **boucle** permet de faire répéter la même action un certain nombre de fois, ou bien jusqu'à ce qu'un évènement se produise, ou encore indéfiniment.

Exemple 1 : Pour monter un escalier de 10 marches, on pourrait utiliser cette suite d'instructions :

Monter une marche
Monter une marche

que l'on peut remplacer par :

Répéter 10 fois

Monter une marche

ou encore :

Répéter jusqu'à ce que l'on soit en haut de l'escalier

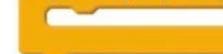
Monter une marche

SCRATCH

répéter fois



répéter jusqu'à



Exemple 2 : Pour programmer le clignotement d'une guirlande de Noël :

Répéter indéfiniment

Allumer les ampoules
Attendre 1 seconde
Éteindre les ampoules
Attendre 1 seconde



Coup de pouce

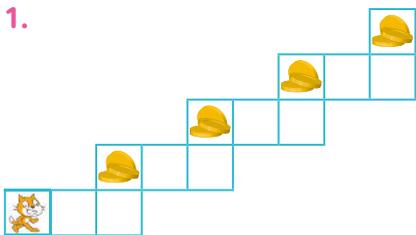
- Les boucles permettent donc de réduire la taille des programmes et des algorithmes que l'on crée.
- Les boucles **Répéter indéfiniment** permettent de programmer des actions permanentes.

J'applique

Exercice 1 Le labyrinthe Difficulté ★

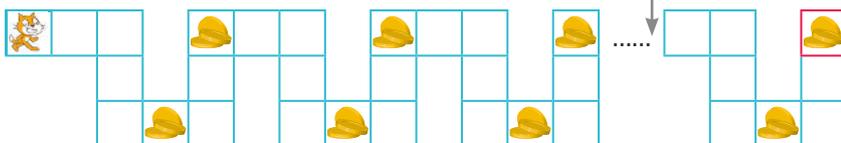
Dans chaque cas, aider Scratchy à récupérer le maximum de pièces d'or en utilisant les instructions , , et pour se déplacer à l'intérieur d'une boucle.

1.



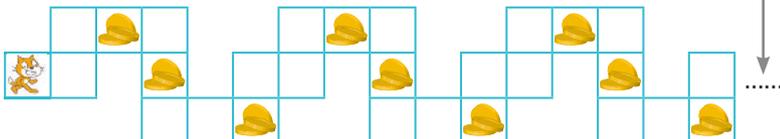
Ces pointillés « » signifient que l'on continue de la même manière un certain nombre de fois.

2.



Ces pointillés « » signifient que l'on continue de la même manière indéfiniment.

3.



Question 1.

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

Question 2.

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

Question 3.

.....
.....

Exercice 2 Les caisses automatiques Difficulté ★★

Dans les supermarchés actuels, on trouve de plus en plus de caisses automatiques qui remplacent progressivement les caissières. Pour rendre la monnaie aux clients, ces automates en libre-service suivent et produisent des algorithmes.

1. À l'aide de boucles et des deux instructions : **Donner une pièce de 1€** et **Donner une pièce de 20 cents**, à n'utiliser qu'une seule fois chacune, écrire un algorithme qui permette de rendre 3,80 € au client.

2. Le distributeur peut donner des pièces de 2 €, 1 €, 50 cents, 20 cents, 10 cents, 5 cents, 2 cents ou encore 1 cent. Écrire un algorithme contenant des boucles qui permette de rendre 9,48 € en donnant le moins de pièces possible au client.



Question 1.

.....

.....

.....

.....

Question 2.

.....

.....

.....

.....

.....

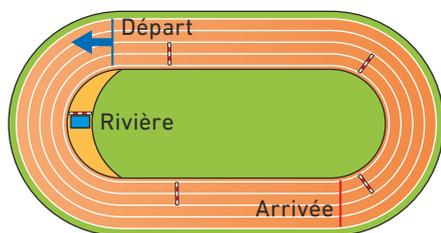
.....

.....

.....

Exercice 3 Le 3 000 mètres steeple Difficulté ★★★

Le 3 000 mètres steeple est une épreuve d'athlétisme assez spectaculaire : les coureurs doivent courir 3 000 m, soit 7 tours et demi de stade, sur lesquels sont disposées 5 barrières espacées de 80 m et dont l'une d'elle est suivie d'une « rivière ». Les premiers 225 m s'effectuent sans croiser de barrières : les juges les rajoutent après le passage des sportifs.



À l'aide de boucles et en utilisant les instructions ci-dessous, compléter l'algorithme ci-contre qui doit décrire de la façon la plus courte possible le parcours effectué par les coureurs.

- Courir ... mètres
- Sauter une barrière
- Sauter la rivière

Prendre le départ de la course

.....

.....

.....

.....

.....

.....

.....

.....

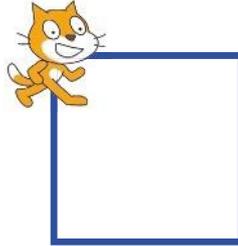
.....

Franchir la ligne d'arrivée

JE M'ENTRAINE à programmer

Exercice 4 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui permet d'obtenir le dessin ci-contre dans le logiciel Scratch.



Programme ①

```
quand cliqué
effacer tout
stylo en position d'écriture
répéter 4 fois
  avancer de 100
  tourner de 90 degrés
relever le stylo
```

Programme ②

```
quand cliqué
effacer tout
stylo en position d'écriture
répéter 4 fois
  avancer de 100
  tourner de 90 degrés
relever le stylo
```

Programme ③

```
quand cliqué
effacer tout
stylo en position d'écriture
répéter 4 fois
  tourner de 90 degrés
  avancer de 100
relever le stylo
```

2. Expliquer brièvement pourquoi les deux autres programmes ne permettent pas de dessiner un carré.

.....

.....

.....

.....

.....

.....

Exercice 5 Je complète un programme

Compléter les zones blanches dans le programme ci-contre afin que celui-ci permette d'obtenir un rectangle de longueur 150 pixels et de largeur 60 pixels.



```
quand est cliqué
effacer tout
stylo en position d'écriture
répéter ..... fois
  avancer de .....
  tourner de ..... degrés
  avancer de .....
  tourner de ..... degrés
```



Exercice 6 J'ordonne un programme

Remettre dans l'ordre les instructions ci-dessous afin de réaliser un programme dans Scratch qui permette d'obtenir un losange de longueur 100 pixels.

stylo en position d'écriture

répéter 2 fois

avancer de 100

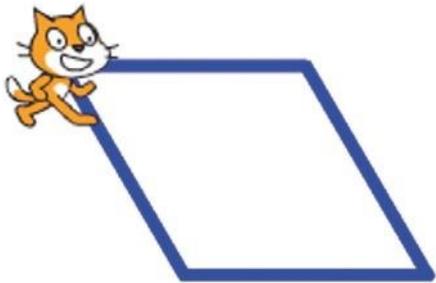
tourner de 60 degrés

avancer de 100

tourner de 120 degrés

quand est cliqué

effacer tout



.....

.....

.....

.....

.....

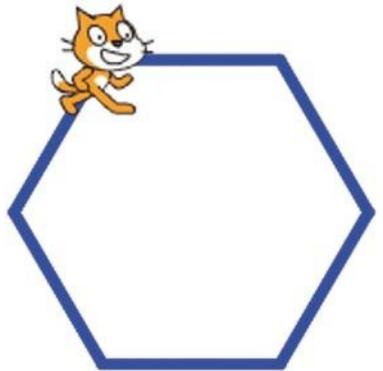
.....

.....

.....

Exercice 7 Je complète un programme à structure imposée

Compléter le programme ci-dessous afin d'obtenir un hexagone de côté 80 pixels comme le montre le dessin ci-contre. Pour cela, tester à l'aide du logiciel Scratch jusqu'à obtenir le programme désiré en respectant les instructions utilisées.



.....

.....

.....

.....

.....

.....

.....

.....

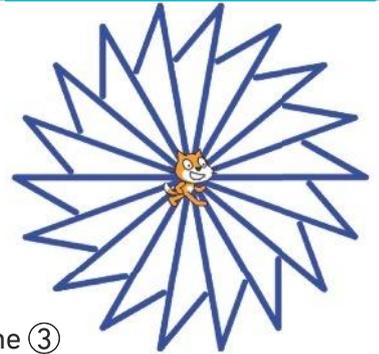
.....

.....

JE M'ENTRAINE à programmer

Exercice 8 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui permet d'obtenir le dessin ci-contre dans Scratch.



Programme ①

```

quand est cliqué
  aller à x: 0 y: 0
  s'orienter à 90
  mettre à 50 % de la taille initiale
  mettre la taille du stylo à 5
  effacer tout
  répéter 18 fois
    stylo en position d'écriture
    avancer de 150
    tourner de 30 degrés
    avancer de 60
    relever le stylo
    aller à x: 0 y: 0
    tourner de 50 degrés
  
```

Programme ②

```

quand est cliqué
  aller à x: 0 y: 0
  s'orienter à 90
  mettre à 50 % de la taille initiale
  mettre la taille du stylo à 5
  effacer tout
  stylo en position d'écriture
  répéter 18 fois
    avancer de 150
    tourner de 150 degrés
    avancer de 60
    relever le stylo
    aller à x: 0 y: 0
    tourner de 170 degrés
  
```

Programme ③

```

quand est cliqué
  aller à x: 0 y: 0
  s'orienter à 90
  mettre à 50 % de la taille initiale
  mettre la taille du stylo à 5
  effacer tout
  répéter 18 fois
    stylo en position d'écriture
    avancer de 150
    tourner de 150 degrés
    avancer de 60
    relever le stylo
    aller à x: 0 y: 0
    tourner de 170 degrés
  
```

2. Expliquer brièvement ce que permettent d'obtenir les deux autres programmes.

.....

.....

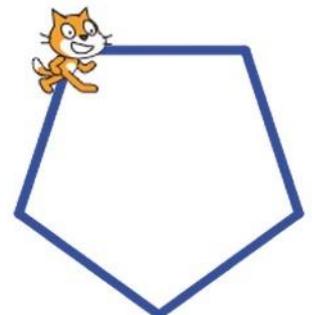
.....

.....

Exercice 9 Un joli pentagone !

En utilisant principalement les instructions ci-dessous, écrire un programme dans Scratch qui permette de dessiner le pentagone de côté 100 pixels ci-contre.

Instructions principales



.....

.....

.....

.....

.....

.....



Exercice 10 Le plus joli chemin n'est pas une ligne droite !

En utilisant principalement les instructions ci-dessous, écrire un programme où le lutin (caché) démarre tout à gauche de l'écran (son abscisse x sera de -240) et avance dans une direction aléatoire (comprise entre 0 et 180) de 10 pixels. La couleur changera aléatoirement à chaque déplacement (la couleur sera comprise entre 1 et 200).

Instructions principales

répéter jusqu'à

s'orienter à

nombre aléatoire entre et

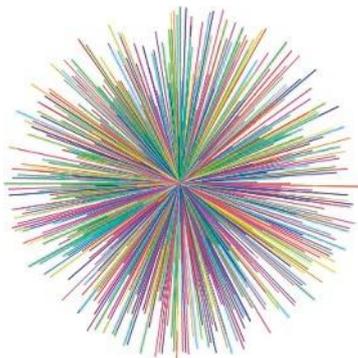


Coup de pouce

Le tracé doit s'arrêter quand le lutin atteint la droite de l'écran, c'est-à-dire qu'il dépasse l'abscisse $+240$.

Exercice 11 Une étoile scintillante

En utilisant principalement les instructions ci-dessous, écrire un programme où le lutin (caché) trace indéfiniment à partir du centre de l'écran un segment d'une longueur aléatoire comprise entre 0 et 180. La direction sera choisie aléatoirement entre 0 et 360 et la couleur sera choisie aléatoirement entre 1 et 200.



Instructions principales

répéter indéfiniment

stylo en position d'écriture

relever le stylo

JE M'ENTRAÎNE à programmer

Exercice 12 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui permet d'obtenir le dessin ci-contre dans Scratch.



Programme ①

```
quand [drapeau] est cliqué
s'orienter à 90
mettre à 40 % de la taille initiale
aller à x: -200 y: 0
mettre la taille du stylo à 5
effacer tout
répéter 10 fois
  stylo en position d'écriture
  répéter 4 fois
    avancer de 20
    tourner de 90 degrés
  relever le stylo
  avancer de 20
```

Programme ②

```
quand [drapeau] est cliqué
s'orienter à 90
mettre à 40 % de la taille initiale
aller à x: -200 y: 0
mettre la taille du stylo à 5
effacer tout
répéter 10 fois
  stylo en position d'écriture
  répéter 4 fois
    avancer de 20
    tourner de 90 degrés
  relever le stylo
  avancer de 40
```

Programme ③

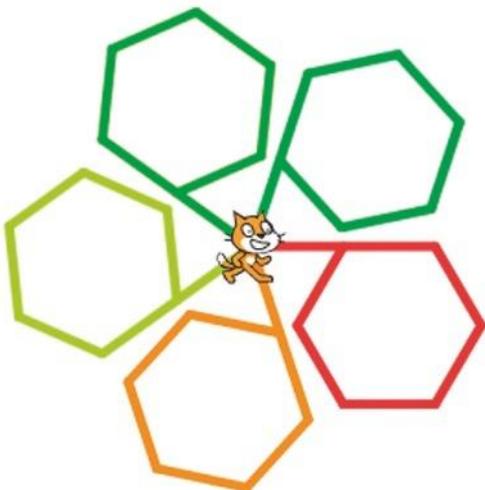
```
quand [drapeau] est cliqué
s'orienter à 90
mettre à 40 % de la taille initiale
aller à x: -200 y: 0
mettre la taille du stylo à 5
effacer tout
répéter 10 fois
  stylo en position d'écriture
  répéter 4 fois
    avancer de 20
    tourner de 90 degrés
  avancer de 40
  relever le stylo
```

2. Dessiner ci-dessous le motif qui serait obtenu avec les deux autres programmes.



Exercice 13 Un motif coloré

Écrire dans l'emplacement ci-contre un programme qui permette de réaliser le dessin ci-dessous où les 5 motifs ont des couleurs différentes.



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



Exercice 14 Des points de toutes les couleurs

Écrire dans l'emplacement ci-contre un programme qui permette de réaliser le dessin ci-dessous où les points ont des couleurs choisies aléatoirement.

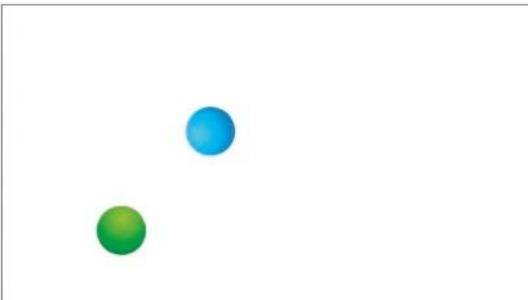


Area with horizontal dotted lines for writing code.

Coup de pouce
 Pour faire un point, il suffit de mettre la taille du stylo à 10, puis de mettre le stylo en position d'écriture et le relever.

Exercice 15 Les balles prisonnières

1. Programmer dans Scratch deux lutins de type « Balle » qui :
 - avancent indéfiniment ;
 - rebondissent quand le bord est atteint ;
 - changent de direction lorsqu'ils percutent l'autre balle.
2. Écrire ci-contre le programme correspondant à l'une des deux balles.



Area with horizontal dotted lines for writing code.

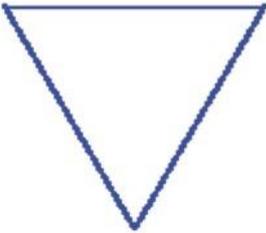
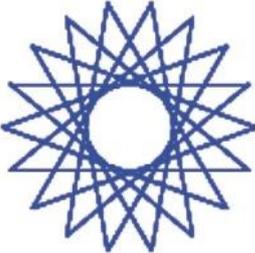
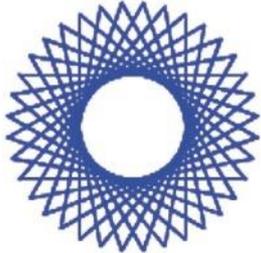
Coup de pouce
 On pourra programmer un lutin et le dupliquer pour n'avoir à modifier que très peu de choses.

Je programme pour résoudre des problèmes



Exercice 16 La tête dans les étoiles

À l'aide du programme ci-contre, en mettant différentes valeurs dans la zone de saisie en jaune, on a obtenu les dessins suivants.

Dessin ①	Dessin ②
	
Dessin ③	Dessin ④
	

```

quand est cliqué
  aller à x: 0 y: 0
  s'orienter à 90
  mettre à 50 % de la taille initiale
  montrer
  effacer tout
  stylo en position d'écriture
  répéter 100 fois
    avancer de 100
    tourner de [ ] degrés
  cacher
  
```

1. On a saisi les valeurs 120 ; 130 ; 140 et 150 pour obtenir les 4 dessins ci-dessus. Associer chaque valeur au dessin qui lui correspond.

.....

.....

2. Expliquer le lien entre le nombre de sommets de chaque dessin et le nombre saisi.

.....

.....

.....

.....

.....

.....

3. Peut-on changer le nombre de fois où la boucle se répète sans changer les dessins obtenus avec ces quatre nombres ? Si oui, quelle est la plus petite valeur que l'on peut saisir ?

.....

.....

4. Quel nombre de sommets pourra-t-on obtenir au maximum en changeant le nombre dans la case jaune ? Quel nombre doit-on saisir dans cette case jaune pour cela ?

.....

.....

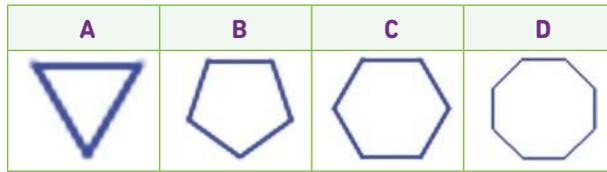
Je prépare le brevet des collèves

15 min

Notions : Instructions et boucles

Réfléchir à combien de fois s'exécute la boucle.

1. Préciser sans justification le motif obtenu en exécutant le programme ci-contre.



```

quand est cliqué
répéter 6 fois
  avancer de 20
  tourner de 60 degrés
    
```

Regarder combien de côtés comporte ce motif et de quel angle on doit tourner entre chacun de ces côtés.

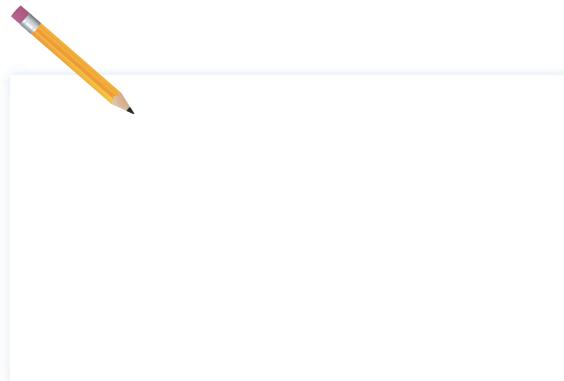
2. Écrire un programme qui permet d'obtenir le motif D.

```

.....
.....
.....
.....
    
```

Dessiner d'abord un motif et voir ensuite le décalage inséré avant de reproduire ce même motif. L'instruction S'orienter à 90° permet d'orienter le lutin vers la droite.

3. Dessiner le motif que permettra d'obtenir le programme ci-contre.



```

quand est cliqué
aller à x: -200 y: 0
s'orienter à 90
montrer
mettre à 50 % de la taille initiale
effacer tout
répéter 10 fois
  stylo en position d'écriture
  répéter 6 fois
    avancer de 20
    tourner de 60 degrés
  relever le stylo
  avancer de 40
cacher
    
```

En se basant sur le dessin précédent, regarder où se trouve le lutin lorsqu'il a terminé sa construction et en déduire l'utilité de l'instruction Avancer de 40.

4. Que faut-il changer dans l'instruction `avancer de 40` pour obtenir le dessin suivant ?



Chaque motif est plus épais que le précédent...

5. Où faut-il alors insérer l'instruction `ajouter 1 à la taille du stylo` pour obtenir le dessin suivant ?





ALGORITHMIQUE débranchée

Je découvre

Une **variable** est un espace de stockage dans laquelle on range une valeur (un nombre, un texte) pour l'utiliser dans un programme. Chaque variable est désignée avec un nom qui doit être défini avant l'écriture du programme.

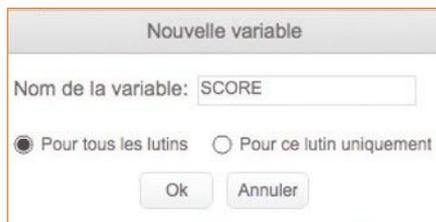
Exemple : Dans un jeu vidéo, les variables *score*, *nombre de vies* peuvent évoluer le long d'une partie.

SCORE 150

Nombre de vies 3



Dans Scratch, on peut **Créer une variable** à partir du menu **Données** : on commence par donner un nom à la **variable** pour ensuite l'utiliser dans des actions.



mettre SCORE à 0

ajouter à SCORE 1

montrer la variable SCORE

cacher la variable SCORE

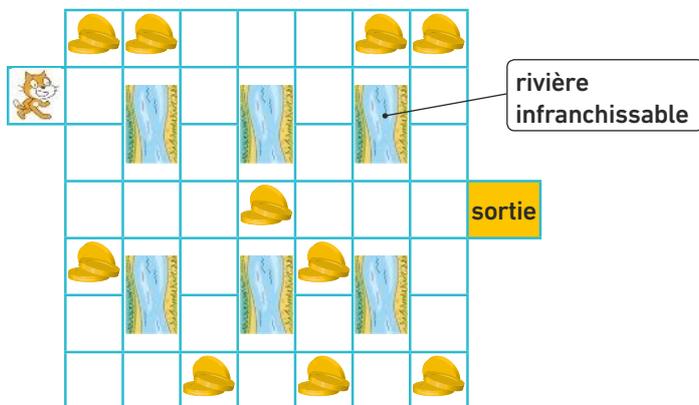
Coup de pouce

Une **variable** par défaut existe sur Scratch : la variable **réponse** qui enregistre la réponse de l'utilisateur à une question posée comme **demande** Donne un nombre **et attendre**.

Exercice 1 Le labyrinthe Difficulté ★

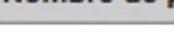
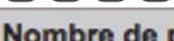
Jade, Fatou, Yassine et Killian jouent au labyrinthe de Scratchy. Indiquer le nombre de pièces d'or que chaque joueur aura récupérées en guidant Scratchy avec les instructions suivantes : ◀ ▶ ▲ ▼.

Remarque : on compte 3 pièces pour une case représentant les pièces d'or.



Yassine ▶ ▲

Répéter 6 fois



Nombre de pièces

Jade



Nombre de pièces

Fatou



Nombre de pièces

Killian ▶ ▼ ▼

Répéter 2 fois

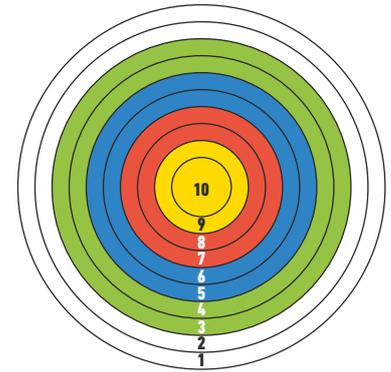


Nombre de pièces

Exercice 2 Le tir à l'arc Difficulté ★★

Un participant à une épreuve de tir à l'arc tire 5 flèches sur une cible avec des zones valant de 1 à 10 points. Le nombre total de points du joueur (*score total*) augmente donc à chaque flèche en ajoutant la valeur du nouveau tir (*valeur fleche*).

À l'aide d'une boucle et de deux variables *score total* et *valeur fleche*, compléter l'algorithme décrivant le déroulement d'une épreuve de tir à l'arc et permettant de calculer au fur et à mesure le score du tireur, puis d'annoncer son score final.



Mettre *score total* à

Répéter fois

.....

Annoncer

Exercice 3 L'horloge digitale Difficulté ★★★

Pour faire fonctionner une horloge digitale, il y a trois variables : *heures*, *minutes* et *secondes*.

Voici ci-contre un morceau de l'algorithme qui permet de gérer l'affichage de l'heure qu'il est (de 0 à 24 heures).

Compléter cet algorithme pour qu'il permette d'afficher les heures, minutes et secondes sur une journée complète (soit de 24 heures).

Répéter 60 fois

Afficher *secondes*

Ajouter 1 à *secondes*

Mettre *secondes* à 0

Ajouter 1 à *minutes*

.....



JE M'ENTRAINE à programmer

Exercice 4 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui permet de calculer le triple d'un nombre donné.

Programme ①

```

quand est cliqué
demander Donner un nombre. et attendre
mettre nombre à réponse
dire regroupe Son triple est nombre * 2
    
```

Programme ②

```

quand est cliqué
demander Donner un nombre. et attendre
mettre nombre à réponse
dire regroupe Son triple est nombre * 3
    
```

Programme ③

```

quand est cliqué
demander Donner un nombre. et attendre
mettre nombre à réponse
dire regroupe Son triple est nombre / 3
    
```

2. Expliquer pourquoi les autres programmes ne permettent pas de calculer le triple d'un nombre donné.

.....

.....

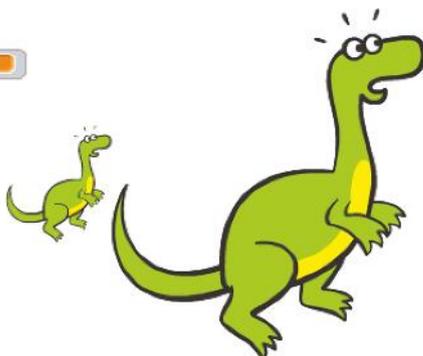
.....

.....

Exercice 5 Je complète un programme

Compléter les zones blanches dans le programme ci-contre afin que celui-ci permette d'obtenir un programme qui affiche un lutin à une taille de 100 %, et qui demande ensuite un coefficient d'agrandissement pour afficher le même lutin, sans effacer le premier, à une taille correspondant à l'agrandissement demandé.

coef 3



```

quand est cliqué
aller à x: -50 y: 0
mettre à 100 % de la taille initiale
demander Donner un coefficient d'agrandissement et attendre
mettre coef à .....
estampiller
aller à x: 100 y: 0
mettre à ..... * ..... % de la taille initiale
estampiller
    
```



Exercice 6 J'ordonne un programme

Remettre dans l'ordre les instructions ci-dessous afin de réaliser un programme dans Scratch qui permette au lutin *fusée* de décompter de 10 à 0 avant le décollage.

Décollage



mettre compteur à 10

répéter 11 fois

dire compteur pendant 1 secondes

ajouter à compteur -1

dire Décollage pendant 5 secondes

quand est cliqué

Large empty box with horizontal dotted lines for writing the Scratch code.

Exercice 7 Je complète un programme à structure imposée

Compléter le programme ci-dessous de façon à obtenir un programme qui demande un sujet, demande ensuite un verbe conjugué, puis demande un complément et, enfin, reconstitue la phrase en regroupant sujet, verbe conjugué et complément.

Diagram of a program structure with colored blocks (yellow, blue, yellow, blue, yellow, blue, yellow) and dotted lines for text input.

JE M'ENTRAINE à programmer

Exercice 8 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui permet d'obtenir la surface d'un rectangle dans Scratch.

Programme ①

```
quand [drapeau] est cliqué
demander [Quelle est la longueur ?] et attendre
mettre [long] à réponse
demander [Quelle est la largeur ?] et attendre
mettre [larg] à réponse
dire [regroupe] La surface est de [long + larg]
```

Programme ②

```
quand [drapeau] est cliqué
demander [Quelle est la longueur ?] et attendre
mettre [long] à réponse
demander [Quelle est la largeur ?] et attendre
mettre [larg] à réponse
dire [regroupe] La surface est de [long + larg * 2]
```

Programme ③

```
quand [drapeau] est cliqué
demander [Quelle est la longueur ?] et attendre
mettre [long] à réponse
demander [Quelle est la largeur ?] et attendre
mettre [larg] à réponse
dire [regroupe] La surface est de [long * larg]
```



2. Expliquer brièvement pourquoi les deux autres programmes ne permettent pas de calculer la surface d'un rectangle.

.....

.....

Exercice 9 L'âge de raison ?

En utilisant principalement les instructions à compléter ci-contre, écrire un programme dans Scratch qui demande l'année de naissance de l'utilisateur et qui lui calcule son âge.



Instructions principales

dire [] pendant [] secondes

demander [] et attendre

regroupe [] []

mettre [] à réponse

.....

.....

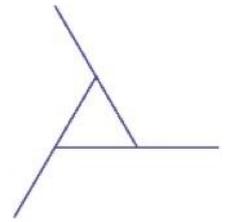
.....

.....

JE M'ENTRAINE à programmer

Exercice 12 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui permet d'obtenir le dessin ci-contre dans Scratch lorsque l'utilisateur répond 100 à la question posée.



Programme ①

```
quand a est cliqué
  aller à x: 0 y: 0
  s'orienter à 90
  effacer tout
  stylo en position d'écriture
  demander choisir un nombre entre 50 et 200 et attendre
  mettre depla à réponse
  répéter 3 fois
    avancer de depla
    avancer de -50
    tourner de 60 degrés
```

Programme ②

```
quand b est cliqué
  aller à x: 0 y: 0
  s'orienter à 90
  effacer tout
  stylo en position d'écriture
  demander choisir un nombre entre 50 et 200 et attendre
  mettre depla à réponse
  répéter 3 fois
    avancer de depla
    tourner de 120 degrés
```

Programme ③

```
quand c est cliqué
  aller à x: 0 y: 0
  s'orienter à 90
  effacer tout
  stylo en position d'écriture
  demander choisir un nombre entre 50 et 200 et attendre
  mettre depla à réponse
  répéter 3 fois
    avancer de depla
    avancer de -50
    tourner de 120 degrés
```

2. Expliquer brièvement pourquoi les deux autres programmes ne permettent pas de dessiner le motif ci-dessus.

.....

.....

.....

.....

Exercice 13 Le solveur d'équation

En mathématiques, on rencontre souvent des équations de type $ax + b = cx + d$ où x est une inconnue et a , b , c et d sont des nombres.

Écrire un programme qui demande les coefficients a , b , c et d et qui donne la solution de l'équation (quand elle existe et qu'elle est unique, c'est-à-dire pour $a \neq c$).

- a 7
- b -3
- c 4
- d 12

La solution de votre équation est 5



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



Exercice 14 C'est trop bon, les glaces !

Chez un glacier, on commande sur une borne tactile. Les prix sont les suivants :

- la glace une boule à 2,50 € ;
- la glace deux boules à 3,50 € ;
- la glace trois boules à 4,50 €.

Réaliser un programme sur Scratch permettant de cliquer sur les glaces souhaitées pour augmenter la facture et qui affiche l'addition finale lorsqu'on clique sur le gros bouton Demander l'addition.



Remarque : on peut indiquer ci-contre la programmation de chaque « lutin ».

Pour le lutin `glace1`, la programmation est :

.....

Pour le lutin `glace2`, la programmation est :

.....

Pour le lutin `glace3` la programmation est :

.....

Pour le lutin `addition` la programmation est :

.....

1. Tester le programme pour un client commandant 4 glaces une boule et 2 glaces trois boules. Pourra-t-il payer avec un billet de 20 € ?

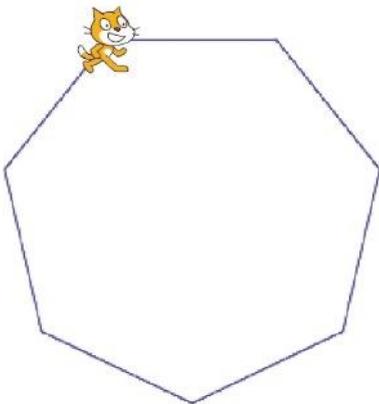
.....

2. Quel est le montant de la facture pour un client commandant 4 glaces de chaque taille ?

.....

Exercice 15 Polygone au choix

Écrire un programme qui permette de construire un polygone régulier avec un nombre de côtés et une longueur de côté choisis par l'utilisateur.



.....

Coup de pouce
 L'angle de rotation du lutin entre deux côtés d'un polygone à n côtés est $360/n$.

Avec le logiciel Scratch, on a réalisé le programme suivant ci-contre :

```

quand [drapeau] est cliqué
  aller à x: -150 y: 0
  s'orienter à 90
  effacer tout
  stylo en position d'écriture
  demander [Choisir un nombre entre 20 et 40] et attendre
  mettre [dépla] à [réponse]
  avancer de 30
  répéter 8 fois
    avancer de [dépla]
    avancer de -15
  relever le stylo
  
```

Remarque : S'orienter à 90 est une orientation horizontale, vers la droite.

1. Tom utilise ce programme et lorsque celui-ci lui demande de choisir un nombre entre 20 et 40, Tom saisit le nombre 25.

► a. Quelle est la forme géométrique du dessin obtenu ?

► b. Expliquer pourquoi la longueur du trait dessiné à l'écran à l'issue du programme réalisé par Tom est de 125 pixels.

c. Quelles sont les coordonnées (x ; y) du lutin à la fin du programme ?

► 2. Tom relance le programme et décide de changer le nombre choisi à la question posée pour obtenir un trait de longueur la plus grande possible.

a. Pour cela, quel doit être le nombre choisi à la question posée ?

b. Quelle longueur maximale peut-il obtenir pour le trait dessiné à l'écran ?

► 3. Parmi les coordonnées de positions suivantes, lesquelles sont possibles pour le lutin à l'issue du programme ? Justifier.

a. Lutin en position (-80 ; 0).

b. Lutin en position (0 ; -80).

c. Lutin en position (48 ; 0).

Comprendre que, dans la boucle, on avance et on recule horizontalement.

Comprendre qu'à la dernière répétition, on a avancé de 25 avant de reculer de 15, mais le trait est donc tracé plus loin que la position finale.

Remarquer que le nombre doit être compris entre 20 et 40.

Pour arriver à une abscisse de a, en tenant compte de la position de départ (abscisse à -150), on peut essayer plusieurs valeurs de « dépla » ou résoudre une équation du type $30 + 8(\text{dépla} - 15) = 150 + a$.



ALGORITHMIQUE débranchée

Je découvre

Une **instruction conditionnelle** est une instruction qui permet de tester une condition, puis d'effectuer une (ou plusieurs) action(s) en fonction du résultat « VRAI » ou « FAUX » de ce test. Elle s'utilise sous la forme d'un bloc d'instructions.

Exemple ① : Un jeune conducteur se déplace en ville au volant de son véhicule.

Lorsqu'il arrive à une intersection avec un feu tricolore, **Si** le feu est rouge **Alors** le conducteur s'arrête il s'arrête lorsque le feu est rouge.



Exemple ② : On organise une loterie avec des boules de différentes couleurs dans une boîte opaque.

On tire une boule au hasard sans connaître sa couleur. **Si** la boule tirée est blanche **Alors** le joueur a gagné **Sinon** il a perdu. Si la boule est blanche, le joueur a gagné un lot.



Coup de pouce

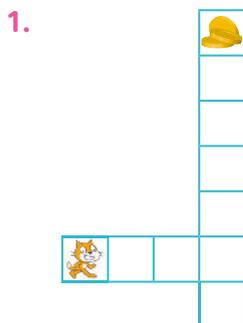
Pour tester la condition étudiée, on utilise des opérateurs de comparaison tels que « INFÉRIEUR À » ; « ÉGAL À » ; « SUPÉRIEUR À »



J'applique

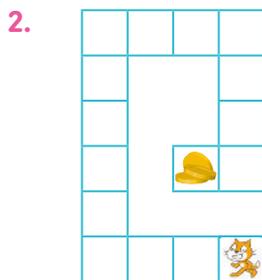
Exercice 1 Le labyrinthe Difficulté ★

Dans chaque labyrinthe, aider Scratchy à récupérer les pièces d'or en complétant les programmes correspondants, en utilisant des instructions conditionnelles et les instructions Avancer, Tourner à gauche et Tourner à droite pour se déplacer à l'aide d'une boucle.



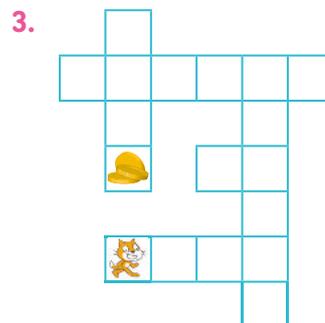
Répéter jusqu'à « pièces d'or »

Si chemin à gauche Alors



Répéter jusqu'à « pièces d'or »

Si chemin à droite Alors



Répéter jusqu'à « pièces d'or »

Si chemin à droite Alors

Exercice 2 What time is it ? Difficulté ★★

Les Anglais utilisent les abréviations AM (*ante meridiem* : avant le midi) pour désigner l'heure du matin et PM (*post meridiem* : après le midi) pour celle de l'après-midi, en écrivant toujours un nombre d'heures entre 1 et 12 (inclus). Ainsi, 16 h 30 à Paris s'écrit 4:30 PM à Londres.

On souhaite construire un algorithme qui demande l'heure en écriture française (en utilisant deux variables : *heures* et *minutes*) et affiche l'heure au format anglais.

Voici un morceau de l'algorithme qui permet de faire cette conversion :

Si *heures* = 0 Alors Dire « 12 : » et *minutes* et « AM »
Si *heures* > 12 Alors Ajouter -12 à *heures*
Dire *heures* et « : » et *minutes* et « PM »
Sinon

Compléter cet algorithme en utilisant éventuellement le morceau précédent.



Demander le nombre d'heures et attendre

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

Exercice 3 La séance de tirs au but Difficulté ★★★

Lors de certains matchs de football, en cas d'égalité à la fin du temps réglementaire, on procède à une séance de tirs au but (5 par équipe) pour désigner le vainqueur. Si les deux équipes sont encore à égalité à la fin de ces tirs au but, on continue alors en « mort-subite », c'est-à-dire jusqu'à ce qu'une équipe marque et l'autre non. À l'aide d'instructions conditionnelles et de deux variables *buts-A* et *buts-B*, compléter l'algorithme décrivant le déroulement de cette « mort-subite ».



Répéter jusqu'à *buts-A* différent *buts-B*

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

JE M'ENTRAINE à programmer

Exercice 4 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui demande un nombre et qui dit s'il est plus petit que 100 dans le logiciel Scratch.

Programme ①

```

quand [drapeau] est cliqué
demander [Donner un nombre.] et attendre
mettre [nombre] à [réponse]
si [nombre > 100] alors
  dire [Ce nombre est plus petit que 100.] pendant 2 secondes
    
```

Programme ②

```

quand [drapeau] est cliqué
demander [Donner un nombre.] et attendre
mettre [nombre] à [réponse]
si [nombre < 100] alors
  dire [Ce nombre est plus petit que 100.] pendant 2 secondes
    
```

Programme ③

```

quand [drapeau] est cliqué
demander [Donner un nombre.] et attendre
mettre [nombre] à [réponse]
si [nombre = 100] alors
  dire [Ce nombre est plus petit que 100.] pendant 2 secondes
    
```

2. Expliquer brièvement pourquoi les deux autres programmes ne permettent pas de dire si le nombre testé est plus petit que 100.

.....

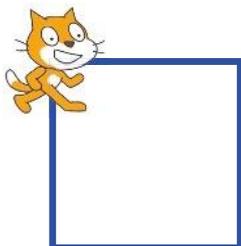
.....

.....

.....

Exercice 5 Je complète un programme

Compléter les zones blanches dans le programme ci-contre afin que celui-ci demande une valeur, puis trace un carré ayant pour longueur de côté la valeur saisie, mais uniquement si celle-ci est inférieure à 200.



```

quand [drapeau] est cliqué
demander [Donner une longueur.] et attendre
si [réponse < .....] alors
  répéter [.....] fois
    avancer de [.....]
    tourner [.....] degrés
  
```



Exercice 6 J'ordonne un programme

Remettre dans l'ordre les instructions ci-dessous afin de réaliser un programme dans Scratch qui permette de déplacer le lutin à droite et à gauche.

Instructions disponibles :

- quand est cliqué
- répéter indéfiniment
- touche flèche droite pressée?
- touche flèche gauche pressée?
- si alors
- si alors
- ajouter 5 à x
- ajouter -5 à x

Zone de programmation vide avec des lignes pointillées pour l'écriture du code.

Exercice 7 Je complète un programme à structure imposée

Compléter le programme ci-dessous qui permet de simuler le lancer de deux dés à 6 faces, qui annonce ensuite la somme des deux dés obtenus, puis qui dit finalement « CRAPS ! » si la somme est de 7.

dé1 5
 dé2 2
 CRAPS !



Zone de programmation avec des blocs pré-remplis et des lignes pointillées pour compléter le code.

JE M'ENTRAINE à programmer

Exercice 8 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui permet de tester dans le logiciel Scratch si le lutin touche la couleur rouge et le fait s'orienter à gauche dans ce cas.

Programme ①

```
quand [drapeau] est cliqué
si couleur [jaune] touchée? alors
  aller à x: 0 y: 0
```

Programme ②

```
quand [drapeau] est cliqué
si couleur [rouge] touchée? alors
  tourner de 90 degrés
```

Programme ③

```
quand [drapeau] est cliqué
si couleur [rouge] touchée? alors
  s'orienter à -90
```

2. Expliquer brièvement pourquoi les deux autres programmes ne permettent pas de tester si le lutin touche la couleur rouge et de le faire s'orienter à gauche dans ce cas.

.....

.....

.....

.....

.....

.....

Exercice 9 Table de multiplication

En utilisant principalement les instructions ci-contre, écrire un programme sur Scratch qui permet de poser à l'utilisateur une multiplication et de vérifier sa réponse.

Instructions principales

```
demande [ ] et attendre
```

```
regroupe [ ] [ ] = [ ] [ ] * [ ]
```

```
nombre aléatoire entre [ ] et [ ]
```

```
si [ ] alors
```

```
sinon
```

Quel est le produit de 8 par 7 ?



.....

.....

.....

.....

.....

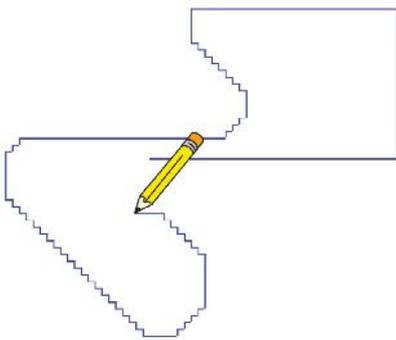
.....



Exercice 10 Déplacement en deux dimensions

En utilisant principalement les instructions ci-contre, écrire un programme qui permet de déplacer dans toutes les directions, à l'aide des 4 flèches du clavier, un lutin en forme de crayon et de tracer sur son passage.

Instructions principales



Area for writing Scratch code with horizontal dotted lines.

Exercice 11 Au cinéma ce week-end !

Au cinéma, il y a deux tarifs : 4,50 € pour les personnes de moins de 14 ans ou de plus de 70 ans, et 8 € pour les autres. En utilisant principalement les instructions ci-dessous, écrire un programme sur Scratch qui permet de donner le prix d'une place de cinéma en fonction de l'âge donné par l'utilisateur.

Instructions principales



Votre place de cinéma coûte 8 €



Area for writing Scratch code with horizontal dotted lines.

JE M'ENTRAINE à programmer

Exercice 12 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui permet de déterminer si un nombre donné par l'utilisateur est pair.

Programme ①

```
quand est cliqué
demander Donner un nombre entier. et attendre
mettre nombre à réponse
si 2 modulo nombre = 0 alors
  dire Ce nombre est pair. pendant 2 secondes
sinon
  dire Ce nombre est impair. pendant 2 secondes
```

Programme ②

```
quand est cliqué
demander Donner un nombre entier. et attendre
mettre nombre à réponse
si nombre / 2 = 0 alors
  dire Ce nombre est pair. pendant 2 secondes
sinon
  dire Ce nombre est impair. pendant 2 secondes
```

Programme ③

```
quand est cliqué
demander Donner un nombre entier. et attendre
mettre nombre à réponse
si nombre modulo 2 = 0 alors
  dire Ce nombre est pair. pendant 2 secondes
sinon
  dire Ce nombre est impair. pendant 2 secondes
```

2. Expliquer brièvement pourquoi les deux autres programmes ne permettent pas de déterminer si un nombre donné par l'utilisateur est pair.

.....

.....

.....

.....

.....

.....

.....

.....

Exercice 13 Devinette de nombre

Écrire un programme ci-dessous qui choisisse secrètement un nombre aléatoire entre 1 et 10, puis qui demande à l'utilisateur de le retrouver. Si l'utilisateur ne le trouve pas, le programme devra indiquer le nombre.

Coup de pouce

On pourra utiliser une variable pour y stocker un nombre aléatoire au début du programme

Domage, j'avais choisi le nombre 2.



```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```


Je prépare le brevet des collèves

20 min

Notions: Variables, instructions conditionnelles, opérateurs

Voici un algorithme programmé sur Scratch qui effectue une action à partir d'un nombre relatif donné par l'utilisateur.

```
quand est cliqué
demander Donner un nombre relatif. et attendre
si réponse < 0 alors
mettre nombre à dire à -1 * réponse
dire nombre à dire pendant 5 secondes
sinon
mettre nombre à dire à réponse
dire nombre à dire pendant 5 secondes
```

Comprendre le fonctionnement du programme sur deux exemples traités différemment par l'algorithme

1. a. Que fait le programme si l'utilisateur saisit 5 en réponse au nombre demandé ?

b. Que fait le programme si l'utilisateur choisit -10 au nombre demandé ?

Comprendre que le programme renvoie toujours un nombre positif (égal à la valeur absolue du nombre choisi) et savoir argumenter.

2. Pour chaque affirmation, expliquer, en justifiant, si elle est vraie ou fausse.

a. Pour n'importe quel nombre choisi par l'utilisateur, le lutin dit un nombre qui a la même valeur que le nombre choisi. Vrai Faux

b. Le lutin dit quelque chose uniquement si le nombre choisi par l'utilisateur est négatif. Vrai Faux

c. Pour n'importe quel nombre choisi par l'utilisateur, le lutin dit toujours un nombre positif. Vrai Faux

Connaître les critères de divisibilité et savoir utiliser un opérateur.

3. Sur Scratch, l'opérateur **a modulo b** calcule le reste de la division euclidienne de **a** par **b**. En utilisant cet opérateur (et éventuellement d'autres), on souhaite poursuivre le programme précédent pour dire si le **nombre à dire** est à la fois divisible par 2 et par 3. Proposer une programmation qui permette de faire cette suite du programme.



ALGORITHMIQUE débranchée

Je découvre

Un bloc est une **suite d'instructions** qui permet de remplacer plusieurs commandes par une seule. Dans le logiciel Scratch, l'onglet Bloc permet de définir les suites d'instructions souhaitées.

Exemple ①: Leïla souhaite tracer un carré de 5 cm de côté.

Elle répète 4 fois le processus suivant :
– elle utilise sa règle pour tracer un segment de 5 cm ;
– elle utilise son équerre pour faire un angle droit à l'extrémité de ce segment.

Pour un carré de côté 50 pixels, on peut créer le bloc ci-contre.

Un bloc d'instructions peut être paramétré, c'est-à-dire qu'il contient au moins une instruction qui contient une information à préciser pour utiliser ce bloc.



Exemple ②: Leïla souhaite tracer un carré de longueur donnée.

Elle répète 4 fois le processus suivant :
– avec sa règle elle trace un segment de la longueur choisie ;
– elle utilise son équerre pour faire un angle droit à l'extrémité de ce segment.

Pour tracer un carré dont le côté a une longueur précisée dans le programme, on peut créer le bloc ci-contre.



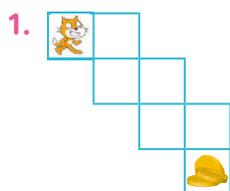
Coup de pouce

Dans ce cas, si on définit la longueur par 50, l'instruction **carré 50** donnera un carré de 50 pixels de côté.

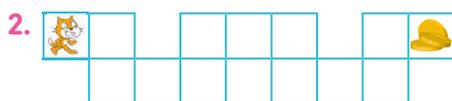
J'applique

Exercice 1 Le labyrinthe Difficulté ★

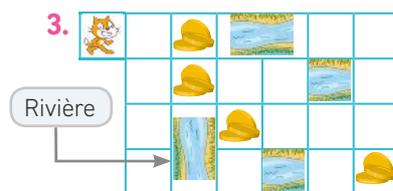
Dans chaque labyrinthe, aider Scratchy à récupérer toutes les pièces d'or en définissant le bloc Prog1 que l'on répètera deux ou trois fois. On utilisera les instructions de déplacement :



```
Répéter 3 fois
|
Prog1
Définir Prog1 :
.....
```



```
Répéter 2 fois
|
Prog1
Définir Prog1 :
.....
```



```
Répéter 3 fois
|
Prog1
Définir Prog1 :
.....
```

Exercice 2 De bon matin !

Difficulté ★★

Le matin, avant d'aller travailler, on effectue plusieurs ensembles d'actions dont l'ordre peut varier selon les personnes. Voici le programme de Léo, 13 ans, un matin en semaine :

- ① Prendre un petit-déjeuner
- ② Faire sa toilette
- ③ S'habiller
- ④ Aller au collège en vélo.

Chacun de ces ensembles d'actions peut être composé de plusieurs actions.

Compléter un algorithme qui donne, dans l'ordre, les ensembles d'actions que l'on peut effectuer chaque matin et, pour chacun de ces ensembles, détailler les actions qu'il contient.



- ①
- ②
- ③ Faire ma toilette
- ④

①

.....

.....

.....

.....

Mettre des chaussures

②

.....

Préparer une tartine de pain avec de la confiture

.....

.....

.....

③

.....

Me laver les dents

.....

④ Aller au collège en bus

.....

.....

.....

Descendre à l'arrêt « collège »

Exercice 3 Le 4 × 100 mètres nage libre

Difficulté ★★★

Une épreuve de natation olympique pour laquelle l'équipe de France masculine est parmi les meilleures du monde est le 4 × 100 mètres nage libre. Cette épreuve de relais est disputée dans une piscine de 50 m de longueur par des équipes de 4 nageurs, qui nagent chacun leur tour un aller-retour.

Décrire cette course par un programme court qui utilise le bloc défini ci-contre.



Définir **Nager son 100 mètres**

- Plonger dans la piscine
- Nager 50 m en ligne droite
- Toucher le bord de la piscine
- Faire demi-tour
- Nager 50 m en ligne droite
- Toucher le bord de la piscine

Premier nageur sur le plot de départ

.....

Nager son 100 mètres

Nageur suivant sur le plot de départ

.....

Arrêter le chronomètre

JE M'ENTRAINE à programmer

Exercice 4 Je choisis le bon programme

1. Parmi les trois programmes suivants, cocher celui qui permet de réaliser un triangle équilatéral dont les côtés mesurent 100 pixels.

Programme ①

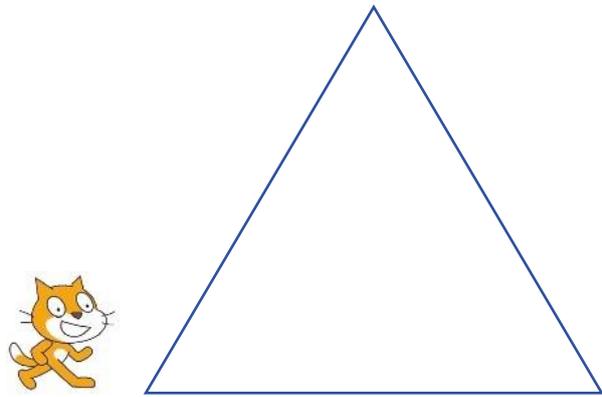
```
définir triangle
stylo en position d'écriture
répéter 3 fois
  avancer de 100
  tourner de 120 degrés
```

Programme ②

```
définir triangle
stylo en position d'écriture
répéter 3 fois
  avancer de 120
  tourner de 100 degrés
```

Programme ③

```
définir triangle
stylo en position d'écriture
répéter 3 fois
  avancer de 100
  tourner de 60 degrés
```



2. Expliquer brièvement pourquoi les deux autres programmes ne permettent pas de construire le triangle équilatéral attendu.

.....

.....

.....

.....

Exercice 5 Je complète un programme

Compléter les zones blanches des instructions du programme ci-contre afin que celui-ci récite la table de multiplication de 7 (de 1×7 à 10×7).

```
définir Table de 7
mettre Facteur à 0
répéter [ ] fois
  ajouter à Facteur [ ]
  dire [ ] * Facteur pendant 1 secondes
```



Exercice 6 J'ordonne un programme

Remettre dans l'ordre les instructions ci-dessous afin de créer un bloc d'initialisation dans Scratch qui permette au lutin de revenir au centre, d'effacer tout et d'être en position d'écriture avec une taille 1 et une couleur rouge pour le stylo.

Area with horizontal dotted lines for writing the program instructions.










Exercice 7 Je complète un programme à structure imposée

Compléter la programmation du bloc ci-dessous de façon à obtenir un programme qui trace un décagone régulier (c'est-à-dire polygone à 10 côtés égaux) de 50 pixels de côté.

Diagram of a Scratch script block with several empty slots for code blocks:

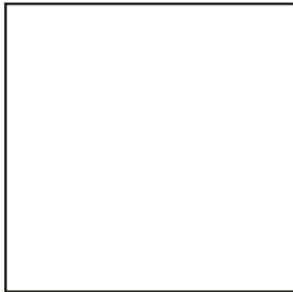
- Slot 1: Empty (purple)
- Slot 2: Empty (green)
- Slot 3: Empty (green)
- Slot 4: Empty (yellow)
- Slot 5: Empty (purple)
- Slot 6: Empty (purple)
- Slot 7: Empty (yellow)

JE M'ENTRAINE à programmer

Exercice 8 Je choisis le bon programme

Le programme ci-contre permet de construire un carré de côté, de taille et de couleur qui sont à définir.

1. Parmi les trois programmes ci-dessous, cocher celui qui permet de réaliser un carré de côté 60 pixels avec un stylo de taille 5 et avec la couleur numéro 80 en utilisant ce bloc.



```

définir carré côté taille couleur
stylo en position d'écriture
mettre la couleur du stylo à couleur
mettre la taille du stylo à taille
répéter 4 fois
  avancer de côté
  tourner de 90 degrés
    
```

Programme ①

```

quand est cliqué
  effacer tout
  carré 60 80 5
    
```

Programme ②

```

quand est cliqué
  effacer tout
  carré 60 5 80
    
```

Programme ③

```

quand est cliqué
  effacer tout
  carré 80 5 60
    
```

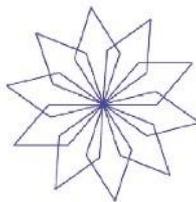
2. Expliquer brièvement pourquoi les deux autres programmes ne permettent pas de construire le carré attendu.

.....

.....

Exercice 9 Une fleur de losanges

En utilisant principalement les instructions à compléter ci-dessous, écrire dans Scratch un bloc qui permet de réaliser un motif initial en forme de losange de 50 pixels de côté, puis écrire un programme qui trace plusieurs fois ce losange afin d'obtenir le dessin ci-contre en utilisant le bloc défini.



Instructions principales

définir Losange

répéter fois

s'orienter à

Losange

avancer de

tourner de degrés

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



Exercice 10 Un programme de calcul

En utilisant principalement les instructions à compléter ci-contre, programmer un bloc dans Scratch qui permette d'effectuer, à partir d'un nombre donné, le programme de calcul suivant :

- ajouter 6 au nombre donné ;
- multiplier le résultat par 3 ;
- diviser le résultat par 2.

Instructions principales

Le nombre 4 donne un résultat de 15



.....

.....

.....

.....

.....

.....

Exercice 11 L'année de naissance

En utilisant principalement les instructions à compléter ci-dessous, écrire un bloc paramétré dans Scratch qui utilise le prénom, le nom et l'âge d'une personne pour faire une phrase donnant son année de naissance.

Pac Man est né en 1980



Instructions principales

.....

.....

.....

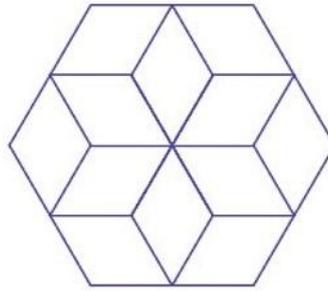
.....

.....

JE M'ENTRAINE à programmer

Exercice 12 Je choisis le bon programme

1. Parmi les trois programmes ci-dessous, choisir celui qui permet de réaliser la figure ci-contre en utilisant ce programme et en partant du centre de la figure.



```

quand est cliqué
  stylo en position d'écriture
  répéter 6 fois
    motif
    tourner de 60 degrés
  
```

Programme ①

```

définir motif
  répéter 2 fois
    avancer de 50
    tourner de 120 degrés
    avancer de 50
    tourner de 60 degrés
  
```

Programme ②

```

définir motif
  répéter 4 fois
    avancer de 50
    tourner de 90 degrés
  
```

Programme ③

```

définir motif
  répéter 6 fois
    avancer de 50
    tourner de 60 degrés
  
```

2. Expliquer brièvement pourquoi les deux autres programmes ne permettent pas de construire la figure attendue.

.....

.....

.....

.....

Exercice 13 Rendez la monnaie !

1. Écrire un bloc paramétré qui permet de calculer, lors d'un achat dans un commerce, la somme d'argent à rendre au client.

Coup de pouce

S'il manque de l'argent, il faut que le programme le signale.

```

définir Monnaie Prix Argent Reçu
  
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Tester ce bloc pour l'achat d'un article de 43,37 € avec un billet de 50 €. Que répond le programme ?

.....

3. Tester ce bloc pour l'achat d'un article de 31,26 € avec un billet de 20 €. Que répond le programme ?

.....



Exercice 14 À la recherche du maximum !

La capture d'écran ci-dessous est le début d'un bloc paramétré qui permet d'obtenir le plus grand parmi trois nombres donnés.

Compléter ce bloc d'instructions qui stockera le nombre maximum recherché dans une variable nommée **max**.



Définir Maximum **nb1 nb2 nb3**

.....

.....

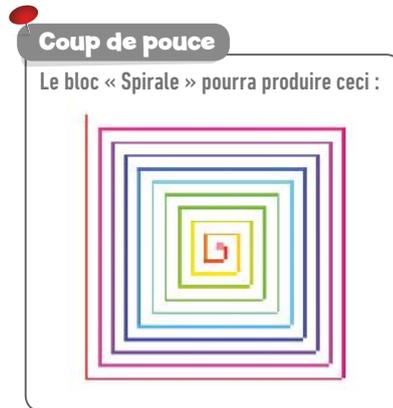
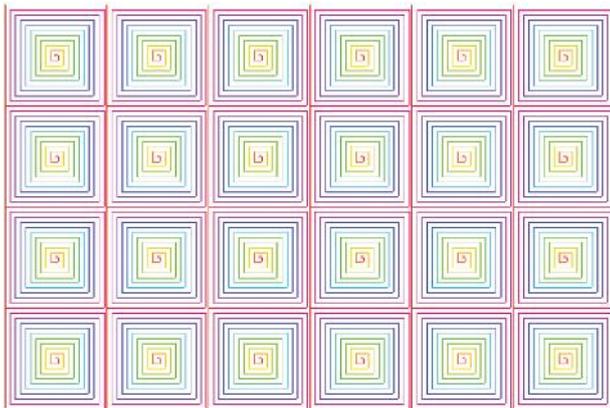
.....

.....

.....

Exercice 15 Un pavage de l'écran

Écrire les programmations des quatre blocs qui permettent au programme ci-contre de produire le pavage de l'écran comme représenté ci-dessous (les dimensions de ce grand rectangle sont de 480 pixels de large sur 320 pixels de haut).



Définir initialisation

.....

.....

.....

Définir **dépla** horizontal

.....

.....

.....

Définir **dépla** vertical

.....

.....

.....

Définir spirale

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



Je programme pour résoudre des problèmes

Exercice 16 Une étude de triangles

1. Écrire un programme dans Scratch qui demande à l'utilisateur de donner trois longueurs, puis qui détermine si ces trois longueurs permettent de construire un triangle.



Large area with horizontal dotted lines for writing code.

Coup de pouce

On pourra utiliser un bloc maximum qui permet d'obtenir le plus grand des 3 nombres donnés.

2. Les longueurs 15 ; 41 et 22 permettent-elles de tracer un triangle ?

Two lines of horizontal dotted lines for writing the answer.

3. Les longueurs 23 ; 37 et 41 permettent-elles de tracer un triangle ?

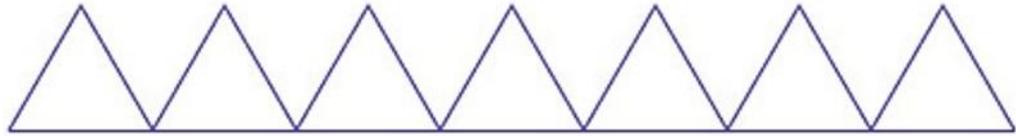
Two lines of horizontal dotted lines for writing the answer.

4. Écrire un bloc `trianglerectangle` qui pourrait compléter le programme précédent afin de déterminer si les trois longueurs données par l'utilisateur forment un triangle rectangle.



Large area with horizontal dotted lines for writing code.

Pour réaliser la frise ci-dessous, on a défini un motif en forme de triangle équilatéral, puis utilisé un programme pour le répéter.



Identifier un motif initial dans une frise
Comprendre quel est le bloc qui permet de tracer ce motif.

► 1. Cocher le bloc qui a permis de construire le triangle initial.

Remarque : s'orienter à 90 correspond à une orientation vers la droite.

Programme (A) Programme (B) Programme (C)

```

définir triangle
stylo en position d'écriture
s'orienter à 90
répéter 3 fois
  avancer de 60
  tourner de 120 degrés
    
```

```

définir triangle
stylo en position d'écriture
s'orienter à 90
répéter 3 fois
  avancer de 60
  tourner de 60 degrés
    
```

```

définir triangle
stylo en position d'écriture
s'orienter à 90
répéter 3 fois
  avancer de 60
  tourner de 120 degrés
    
```

Comprendre que le motif est répété un certain nombre de fois et identifier le décalage entre chacun d'eux.

► 2. Compléter le programme ci-dessous de façon à obtenir la frise attendue.

```

quand est cliqué
effacer tout
mettre la taille du stylo à 1
aller à x: -200 y: 0
répéter ..... fois
  triangle
  relever le stylo
  avancer de .....
    
```

Compter le nombre de segments et retrouver leur longueur dans le programme.

► 3. Quelle est la longueur totale de tous les traits tracés dans cette frise ?

.....

4. On souhaite réaliser la frise ci-dessous.



Identifier le bon endroit pour ajouter une instruction qui modifie la taille du stylo.

► Pour cela, on envisage d'insérer l'instruction `ajouter 1 à la taille du stylo` dans le programme précédent. Où faut-il insérer cette instruction ?

.....
.....



PROJET 1 La chasse aux clés

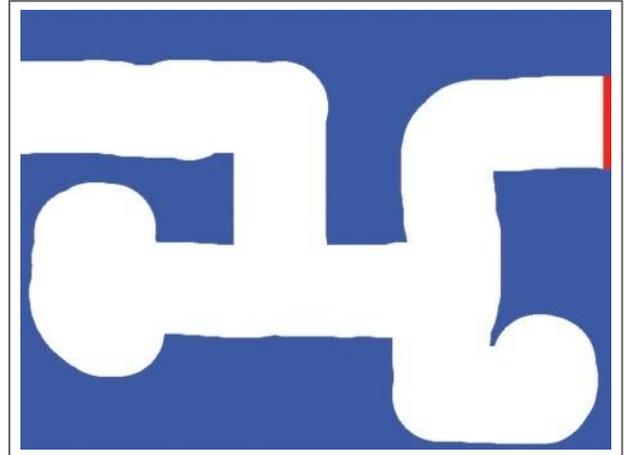
Dans ce jeu, on pilote Tera, un petit personnage, pour qu'il récupère les clés qui lui permettront de sortir d'un labyrinthe, sans se faire dévorer par le fantôme...

Étape 1 Créer un fond pour le décor du jeu

Dans le logiciel Scratch, réaliser un arrière-plan du même style que celui ci-contre. Il faut qu'il contienne des zones un peu plus larges pour y positionner les clés et un rectangle rouge qui délimitera la sortie du labyrinthe.

Coup de pouce

Utiliser l'outil  pour colorier le fond de l'arrière-plan et l'outil  pour gommer de la couleur et ainsi créer le labyrinthe. L'outil  permet de créer le petit rectangle rouge.

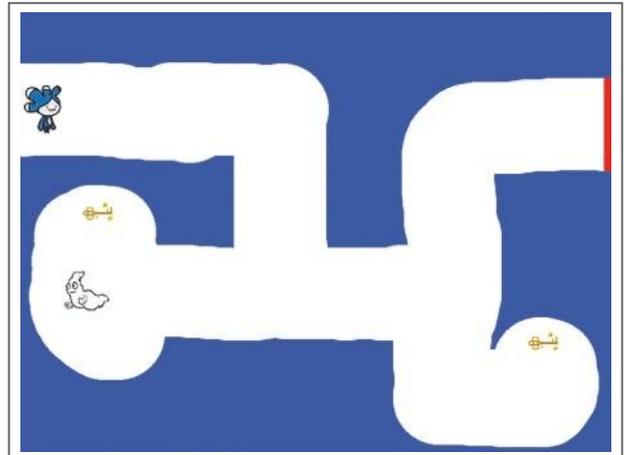


Étape 2 Mise en place des lutins

- Positionner le lutin Tera  au début du labyrinthe, puis positionner deux lutins symbolisant les clés à ramasser et un lutin fantôme Ghost1 à côté de l'une de ces clés.
- Pour tous les lutins, faire en sorte qu'ils débutent à leur place à chaque clic sur le drapeau vert.
- Pour les lutins Tera et Ghost1, fixer le sens de rotation uniquement à **position à gauche ou à droite**.

Coup de pouce

Utiliser l'instruction 



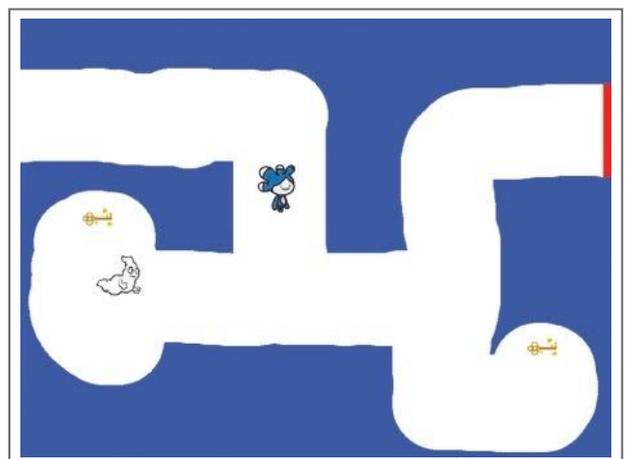
Étape 3 Programmer le déplacement de Tera

- Programmer le déplacement de Tera pour que le personnage puisse se déplacer à l'aide des flèches du clavier dans les quatre directions par tranche de 10 pas.
- Lorsque la couleur bleu est rencontrée, faire reculer Tera de 10 pas pour qu'il ne puisse donc pas sortir du labyrinthe proposé.

Coup de pouce

Utiliser l'orientation pour gérer le déplacement de Tera et l'instruction  pour le faire reculer.

 (90) à droite
(-90) à gauche
(0) vers le haut
(180) vers le bas

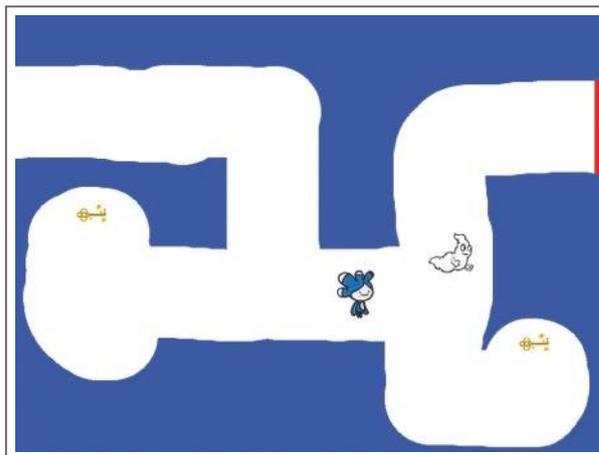


Étape 4 Programmer le mouvement aléatoire du fantôme

Programmer le fantôme pour que, de façon infinie, il s'oriente au hasard et qu'il avance jusqu'à ce qu'il touche un bord du labyrinthe (couleur bleue) ou bien qu'il touche le bord de l'écran.

Coup de pouce

L'instruction **rebondir si le bord est atteint** sera bien utile ici. Il faut également penser à faire reculer le fantôme lorsqu'il touche la couleur bleue.

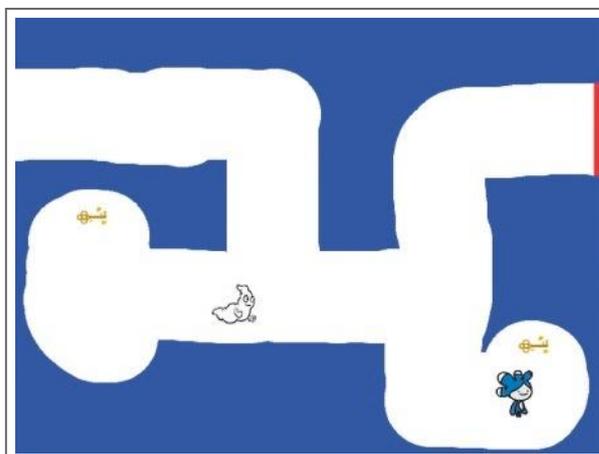


Étape 5 Programmer les clés

- Programmer chaque clé de la même manière. Elles doivent tester indéfiniment si elles sont touchées par Tera.
- Lorsque Tera touche une clé :
 - la clé disparaît ;
 - un petit son est joué ;
 - une variable augmente de 1 (cela permettra à la fin de vérifier que Tera les a bien toutes ramassées).

Coup de pouce

Penser à ce que chaque clé se montre au début du jeu et à remettre la variable CLES à zéro au début de chaque partie

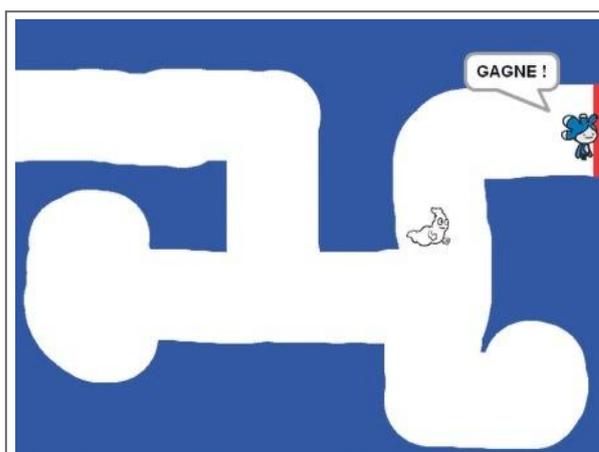


Étape 6 Programmer la fin de partie

- Une partie peut se terminer de deux façons :
- si Tera touche le fantôme, il doit crier « Ahhhhh ! », disparaître et la partie se termine ;
 - si Tera touche la ligne rouge en ayant bien récolté les deux clés, il doit dire « GAGNÉ ! » et la partie s'arrête.

Coup de pouce

Pour stopper la partie, l'instruction **stop tout** est bien utile...



Quelques évolutions possibles à ce projet

- Intégration d'un temps limite pour sortir du labyrinthe.
- Intégration d'autres niveaux de jeu à faire lorsque Tera sort du labyrinthe.
- Dans un jeu à plusieurs niveaux, donner un certain nombre de vies à Tera et lui permettre d'en gagner.
- Rajouter d'autres ennemis dans les tableaux (chauve-souris, araignées, etc.).
- À vos idées !...

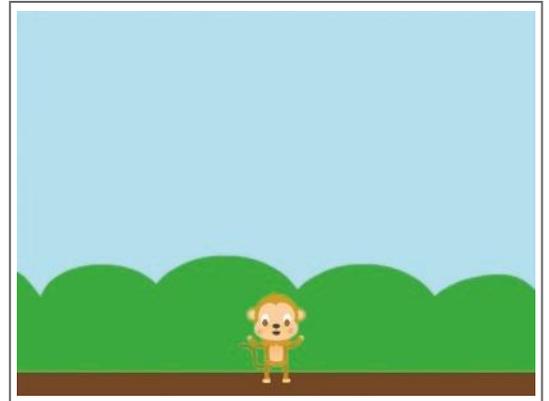


PROJET 2 Le singe gourmand

Un petit singe très gourmand adore les bananes ! Pour en ramasser le plus possible, il devra éviter les pastèques qui risquent de lui tomber sur la tête et l'assommer...

Étape 1 Créer un fond pour le jeu et gérer le déplacement du singe

- Choisir l'arrière-plan dans la bibliothèque du logiciel Scratch et sélectionner le lutin Monkey2 également dans la bibliothèque.
- Programmer Monkey2 pour qu'il apparaisse en bas au centre de l'écran à 50 % de sa taille et qu'il se déplace vers la droite ou vers la gauche grâce aux flèches de déplacement du clavier, mais sans jamais sortir de l'écran.



Coup de pouce

On peut vérifier lors de l'appui sur l'une des deux flèches ◀ et ▶ que l'abscisse du lutin n'a pas dépassé une certaine valeur pour éviter qu'il ne sorte de l'écran.

Étape 2 Faire tomber des bananes et des pastèques

- Rajouter un lutin *banane* (à 50 % de sa taille initiale) et un lutin *pastèque* (à 80 % de sa taille initiale).
- Programmer le lutin banane pour, qu'indéfiniment, cette banane apparaisse aléatoirement en haut de l'écran et glisse verticalement en un temps aléatoire jusqu'en bas de l'écran, où elle disparaît.
- Programmer de même le lutin pastèque, mais mettre une pause aléatoire avant qu'elle apparaisse.
- Dupliquer ensuite ce lutin pour augmenter le nombre de pastèques qui tombent.



Coup de pouce

Pour programmer une pause, on peut utiliser l'instruction

attendre nombre aléatoire entre 0 et 2 secondes

Étape 3 Gérer les contacts avec les bananes

Programmer le lutin banane pour qu'il se cache lorsqu'il touche le singe et qu'il ajoute 1 au nombre de bananes attrapées.

Coup de pouce

Créer une variable <BANANES> qu'il faudra remettre à zéro au début de chaque partie. Il est préférable de tester le contact dans une autre boucle infinie afin d'éviter quelques erreurs de comptage.



Étape 4 Gérer les contacts avec les pastèques

- Chaque contact avec une pastèque doit enlever une vie. Il faut donc créer une variable <VIES> qui démarre toujours à 5, par exemple, et à laquelle on enlève 1 chaque fois qu'une pastèque touche le singe.
- Le singe changera 3 fois de costume pour montrer qu'il a pris un coup.

Coup de pouce

Pour faire changer de costume, l'instruction **costume suivant** utilisée dans une boucle avec un tout petit temps d'attente est assez efficace.



Étape 5 Ajouter la possibilité de gagner des vies

- Ajouter un lutin en forme de cœur qui devra tomber comme les bananes et les pastèques et qui fera ajouter 1 aux vies lorsque il est attrapé par le singe.
- En revanche, plus le singe a de vies et plus les apparitions des cœurs doivent être rares.

Coup de pouce

On peut simuler le tirage au sort toutes les secondes d'un nombre entre 1 et 6 plus le nombre de vies pour raréfier les apparitions du cœur. Celui-ci n'apparaissant que quand le 1 sort par exemple.



Étape 6 Programmer la fin de partie

- La partie se termine quand le singe n'a plus de vies.
- Le singe annonce alors le nombre de bananes que le joueur a réussi à attraper.

Coup de pouce

On peut envoyer un message à tous les lutins lorsque le nombre de vies est égal à 0 pour leur dire de se cacher et de stopper leurs scripts.



Quelques évolutions possibles à ce projet

- Intégration d'un temps limite de jeu.
- Mémoriser le meilleur score quand le joueur a perdu.
- Faire un jeu à 2 joueurs où 2 singes pilotés au clavier doivent attraper le plus de bananes possibles.
- Ajouter d'autres fruits qui tombent, certains valant plus de points que d'autres...
- À vos idées !...

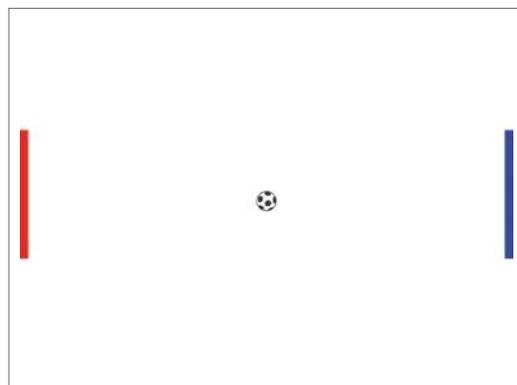


PROJET 3 Air hockey

Ce jeu se joue à deux joueurs : chacun protège ses cages et doit tenter d'envoyer le palet dans les cages de l'adversaire. Bonne chance !

Étape 1 Créer l'aire de jeu

- Créer l'aire de jeu du Air hockey en prenant le lutin Ball-Soccer pour représenter le palet et le lutin Paddle pour représenter les cages.
- Au démarrage, positionner le palet au centre de l'écran à 30 % de sa taille initiale.
- Colorier le Paddle en rouge, le faire pivoter pour qu'il soit à la verticale, le dupliquer et colorier sa copie en bleu. Les positionner au démarrage de chaque côté centrés verticalement à 150 % de leur taille initiale.



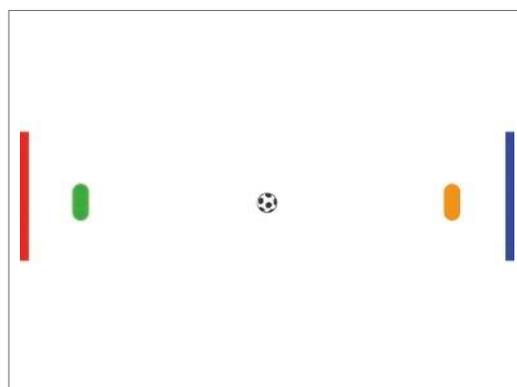
Coup de pouce

Pour faire pivoter le Paddle, utiliser l'instruction

fixer le sens de rotation à 360° et orienter le lutin vers le haut.

Étape 2 Positionner et gérer les poignées

- Pour les poignées (les crosses de air hockey), utiliser le lutin Button2. Dupliquer ce lutin et colorier chaque poignée d'une couleur différente.
- Au démarrage, placer ces poignées à 30 % de leur taille initiale, centrées devant chacune des cages et programmer leur déplacement à l'aide des touches [A] et [Q] pour la poignée de gauche et des flèches ▲ et ▼ pour la poignée de droite.

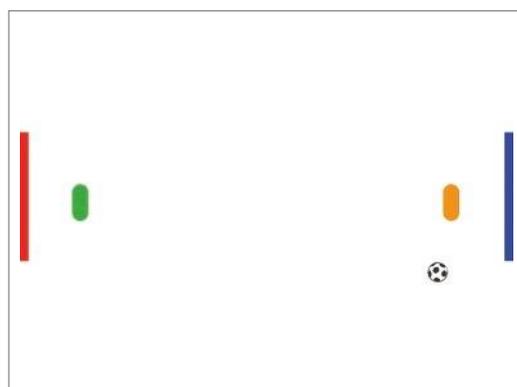


Coup de pouce

Pour obtenir un déplacement fluide, utiliser une boucle infinie contenant des tests du style « Si <Touche...Pressée> ».

Étape 3 Programmer le déplacement du palet

- Le palet doit démarrer au bout d'une seconde, s'orienter au hasard vers la droite ou la gauche, avec un angle au hasard au maximum égal à 30° par rapport à l'horizontale, vers le haut ou le bas.
- Le palet avance alors indéfiniment, en rebondissant sur les bords d'une vitesse qui augmente toutes les 3 secondes.



Coup de pouce

Créer une variable <VITESSE> qui débute à 10 et qui augmente de 1 dès que le chronomètre dépasse les 3 secondes.

chronomètre

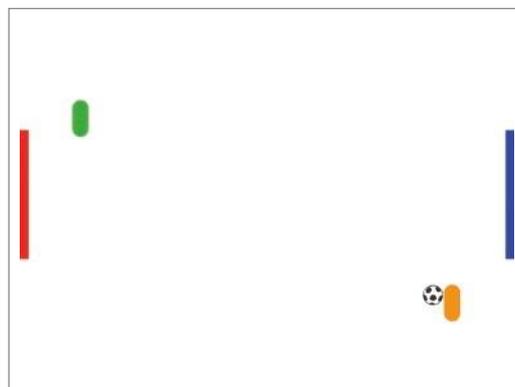
réinitialiser le chronomètre

Étape 4 Gérer le contact du palet avec les poignées

Quand le palet touche l'une des deux poignées, il doit rebondir dessus (sa direction est multipliée par -1) et on rajoutera aléatoirement un angle compris entre -20° et 20° pour rendre le jeu plus intéressant.

Coup de pouce

Pour éviter que la balle rebondisse plusieurs fois sur le palet, rajouter l'instruction **attendre 0.5 secondes** juste après le changement de direction pour qu'il ne teste pas à nouveau immédiatement le contact avec la poignée.

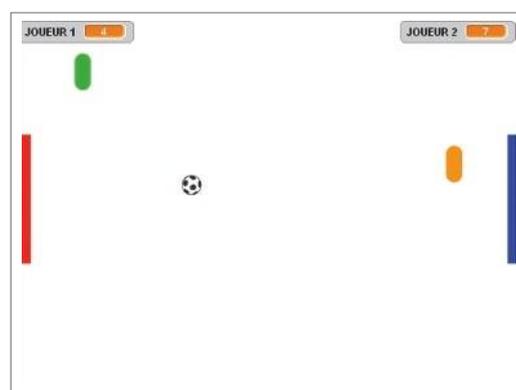


Étape 5 Programmer la gestion du score

Ajouter deux variables <JOUEUR1> et <JOUEUR2> qui augmentent de 1 chaque fois que le palet touche les buts adverses.

Coup de pouce

Utiliser la même astuce qu'à l'étape 4 pour qu'un seul point soit comptabilisé lorsque le palet atteint l'une des deux cages.

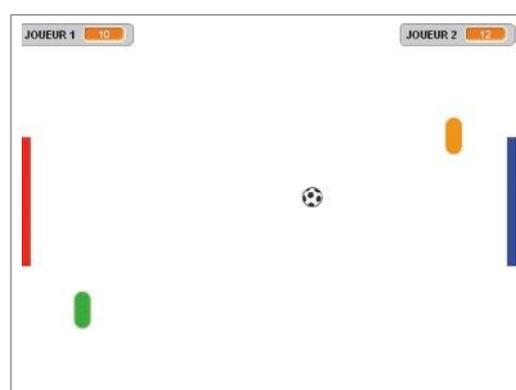


Étape 6 Gérer globalement la partie

Lorsqu'un but est marqué, faire en sorte que le palet retourne au centre de l'écran, qu'il attende une seconde et qu'il se relance dans une direction au hasard, mais sans que les scores des joueurs se remettent à zéro.

Coup de pouce

Utiliser l'envoi de messages pour faire démarrer une nouvelle phase de jeu lorsqu'un but est marqué ou lorsque la partie commence.



Quelques évolutions possibles à ce projet

- Déclarer un joueur gagnant lorsqu'il a atteint un certain score.
- Faire rétrécir le palet d'un joueur quand son score dépasse un certain seuil ou quand son avance par rapport à l'autre joueur est trop grande.
- Faire apparaître des bonus qui font grossir ou rétrécir le palais, les cages, les poignées, etc.
- À vos idées !...



PROJET 4 Les aventures de Petit Scarabée

On peut dire que la démarche de Tom le petit scarabée est hésitante ! Il va plutôt vers la gauche, ou plutôt vers la droite...

Dans le jeu suivant, on va programmer un déplacement permettant de faire « monter » Tom vers le haut de l'écran, mais avec des pas aléatoires orientés un peu à gauche ou un peu à droite, tout en permettant au joueur de « parier » sur sa position finale.

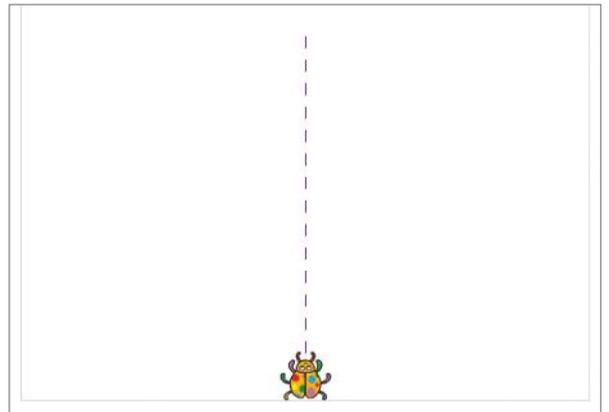
Étape 1 Tracer la ligne verticale de séparation

- Prendre un scarabée pour lutin et le positionner en bas de l'écran.
- À l'aide d'un bloc d'instructions, lui faire tracer une ligne pointillée verticale qui sépare l'écran en deux.

Coup de pouce

Penser à réduire la taille du lutin.

mettre à % de la taille initiale



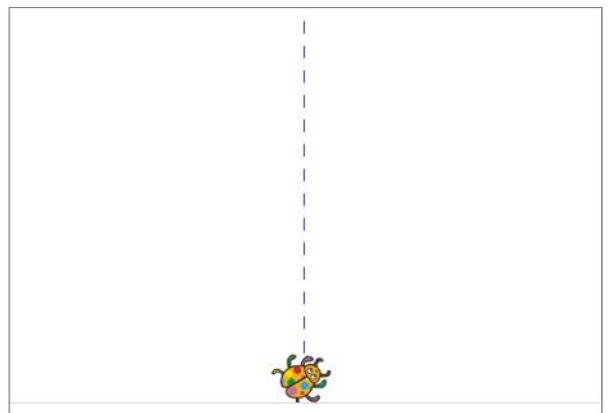
Étape 2 Faire un pas « aléatoire vers le haut »

À l'aide d'une variable dans laquelle on stocke un nombre aléatoire, programmer un déplacement de Tom qui l'oriente soit à 60° (vers le « haut et à droite »), soit à -60° (vers le « haut et à gauche ») et qui le fait avancer d'un pas.

Coup de pouce

On pourra utiliser l'opérateur « Nombre aléatoire entre 1 et 2 ».

nombre aléatoire entre 1 et 2



Étape 3 Obtenir le déplacement complet

En répétant ce déplacement un certain nombre de fois, permettre au scarabée d'aller jusqu'en haut de l'écran en laissant une trace du chemin emprunté.

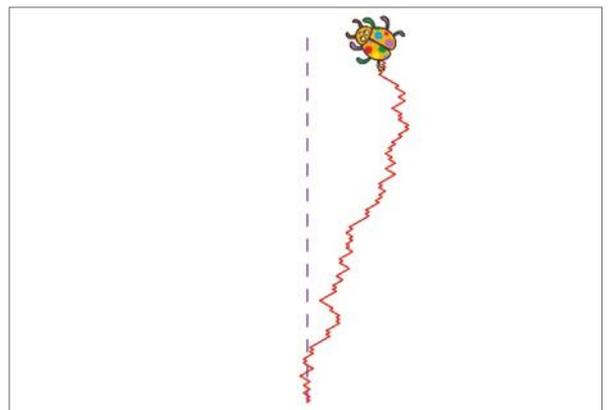
Coup de pouce

On veillera à ne pas dépasser la hauteur de l'écran.

répéter jusqu'à



ordonnée y

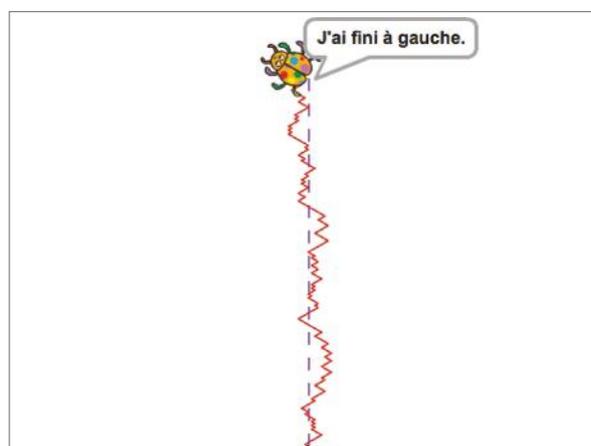


Étape 4 Tester la position finale

En récupérant l'abscisse de la position finale de Tom, lui faire dire s'il a terminé sa course « à droite » ou « à gauche » de la ligne verticale.

Coup de pouce

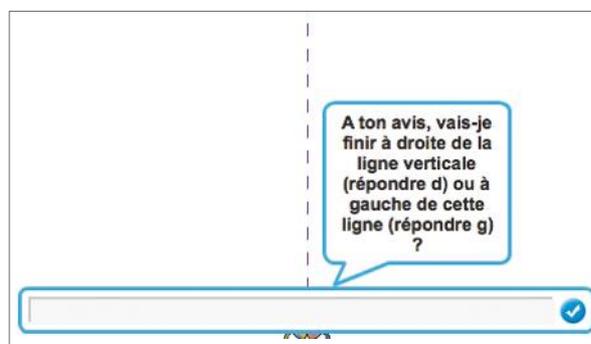
Penser à tester s'il termine exactement au milieu et, dans ce cas, à le faire se déplacer une fois de plus.

**Étape 5** Programmer un pari

- Avant le départ de la course de Tom, demander à l'utilisateur s'il parie plutôt sur le fait que le scarabée finira « à gauche » ou « à droite ».
- Faire afficher le pari en haut de l'écran en utilisant une variable qui affiche GAUCHE ou DROITE en fonction du choix.

Coup de pouce

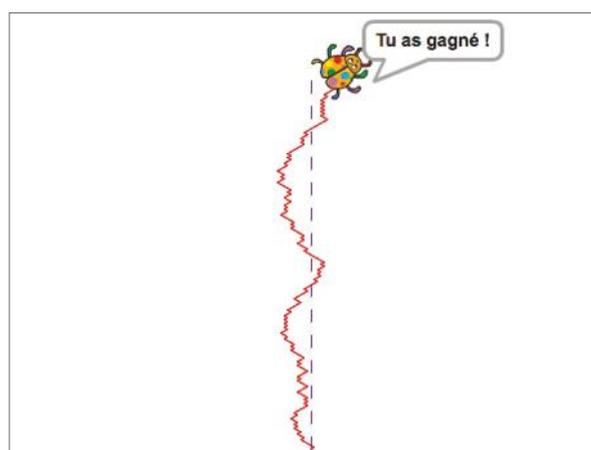
On pourra répéter la question jusqu'à ce que la réponse soit « d » ou « g » pour être certain d'avoir un pari correct.

**Étape 6** Pari gagné ou pari perdu ?

Comparer le « pari » de l'utilisateur à la position obtenue en fin de course et dire s'il a gagné ou perdu.

Coup de pouce

On pourra utiliser plusieurs instructions conditionnelles.

**Quelques évolutions possibles à ce projet**

- Demander à l'utilisateur s'il veut rejouer.
- Prévoir une cagnotte de départ qui permet à l'utilisateur de miser une certaine somme (on gagne deux fois la mise si le pari est correct).
- Demander à mi-parcours une autre possibilité de miser.
- À vos idées !...



PROJET 5 Au clair de la Lune

Scratch permet de jouer de la musique. Il possède même toute une gamme « d'instruments ». Ce projet va permettre de programmer un jeu pour reconnaître quel instrument joue l'air d'« Au clair de la Lune ».

Étape 1 Programmer « Au clair de la Lune »

Programmer, à l'aide d'un bloc d'instructions, le fameux air « Au clair de la Lune » dont voici ci-contre une partition qui est composée sur le logiciel Scratch.

Coup de pouce

Ce bloc d'instruction ne doit pas s'exécuter pour l'instant.

jouer la note 60 pendant 0.5 temps

jouer la note 60 pendant 0.5 temps

jouer la note 60 pendant 0.5 temps

jouer la note 62 pendant 0.5 temps

jouer la note 64 pendant 1 temps

jouer la note 62 pendant 1 temps

jouer la note 60 pendant 0.5 temps

jouer la note 64 pendant 0.5 temps

jouer la note 62 pendant 0.5 temps

jouer la note 62 pendant 0.5 temps

jouer la note 62 pendant 0.5 temps

jouer la note 60 pendant 0.5 temps

Étape 2 Choisir un instrument au hasard

- Faire dire un message d'accueil au lutin pour présenter le jeu.
- Lui faire choisir au hasard un instrument parmi les 10 premiers prédéfinis dans les instruments de Scratch.
- Sélectionner cet instrument et jouer le morceau du bloc « Au clair de la Lune ».

Coup de pouce

On pourra créer une variable pour y stocker un nombre aléatoire entre 1 et 10. Pour information, voici la liste des dix premiers instruments de Scratch dans la capture ci-contre.

- (1) Piano
- (2) Piano électrique
- (3) Orgue
- (4) Guitare
- (5) Guitare électrique
- (6) Basse
- (7) Pizzicato
- (8) Violoncelle
- (9) Trombone
- (10) Clarinette

Je vais choisir un instrument de musique au hasard et jouer un air bien connu...



Étape 3 Faire apparaître des images des instruments

Faire apparaître une image de tous les instruments possibles et demander à l'utilisateur de cliquer sur le bon instrument.

Coup de pouce

On pourra utiliser une autre variable qui prend la valeur de l'instrument sur lequel on clique.

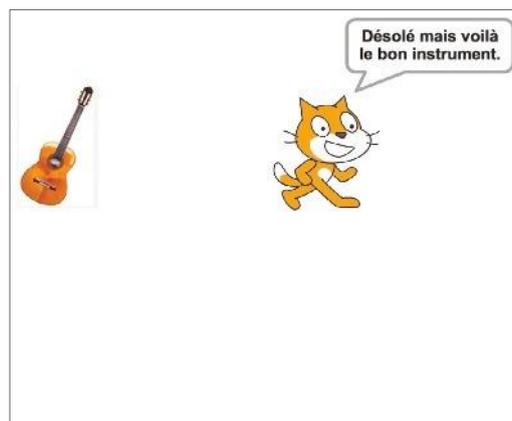


Étape 4 Comparer les instruments

- Comparer l'instrument cliqué avec l'instrument choisi au hasard par Scratch.
- Si c'est le bon instrument, faire dire un message de félicitations par le lutin.
- Si ce n'est pas le bon instrument, alors mettre en évidence le bon instrument par un message ou par un effet visuel.

Coup de pouce

On pourra envoyer un message pour cacher ou montrer certains instruments selon la réponse.



Étape 5 Compter les points

Faire recommencer le jeu pour que l'utilisateur joue cinq fois et marque un point à chaque bonne réponse.

Coup de pouce

Il sera peut-être alors utile de déclencher les questions par un envoi de message afin de mieux gérer les transitions entre ces questions.



Étape 6 Utiliser le chronomètre

Utiliser un chronomètre de façon à ce qu'une réponse rapide rapporte plus de points.

Coup de pouce

On pourra, par exemple, pour chaque bonne réponse rapide, gagner un bonus égal à 5 points moins le nombre de secondes utilisées pour répondre.



Quelques évolutions possibles à ce projet

- Programmer d'autres morceaux de musique et les jouer au hasard.
- Utiliser plus d'instruments.
- Faire un jeu à deux joueurs.
- À vos idées !...

Nom :

Prénom :

Établissement :

.....

Crédits iconographiques

Couv	Nik Merkulov, harper kt, Naty Kash Nataliia/SHUTTERSTOCK
5 ht g	Grafvision/SHUTTERSTOCK
5 bas g	G.Tatiana/SHUTTERSTOCK
15 ht g	Syda Productions/SHUTTERSTOCK
15 bas d	A. Hassenstein/Getty Images for IAAF/AFP
25 bas d	J. Dharmasena/SHUTTERSTOCK
35 ht g	IR Stone/SHUTTERSTOCK
35 bas g	Tungphoto/SHUTTERSTOCK
45 ht g	C. Rose / GETTY IMAGES EUROPE//AFP
45 ht d	A. Zveige/FOTOLIA by ADOBE

Direction éditoriale : Julien Barret

Édition : Malik Agina

Couverture : Christine Soyez

Conception graphique : Pierre Taillemite

Recherche iconographique : Lorena Martini

Fabrication : Jean-Philippe Dore

Réalisation et schémas : Soft Office

À la mémoire de Marcel-André BOULLIS, professeur de mathématiques.