2023

# CAHIER D'ACTIVITÉS SCRATCH

*8ème année*

DE BASE

ENSEIGNANT:
MAHJOUBI BILEL

EMAIL:MAHJOUBI.BILEL@GMAIL.COM

## SÉQUENCES

1. Instructions et algorithme
2. Utilisation des boucles
3. Utilisation des instructions conditionnelles
4. Utilisation des variables
5. Utilisation d'un bloc d'instructions

## PROJETS

1. Le crabe aux pinces magiques
2. Le mage et la grenouille
3. Le chiffre de César

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

**Exemple :** Marc joue à un célèbre jeu de football sur console. Il communique des **instructions** aux joueurs à l'aide d'une manette. Voici la **séquence** d'instructions qui permet de célébrer un but en faisant des saltos arrière :

- *Maintenir la touche* R2 .
- *Appuyer deux fois brièvement sur la touche* □ .

Un **algorithme** est une suite finie d'instructions permettant de résoudre un problème.

**Exemple :** Une recette de cuisine est un algorithme.
En effet, on dispose d'ingrédients au départ, on applique les instructions données par la recette et on obtient le plat désiré à la fin.

## J'applique

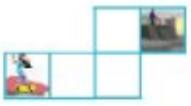
### 1 À la cantine Niveau 1

Prendre du pain — Choisir un fruit — Choisir une entrée — Choisir un plat principal — Prendre un plateau — Choisir un laitage — Passer la carte de cantine

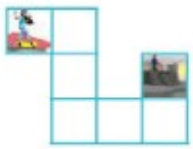
Écris un algorithme de ton passage à la cantine en remettant dans l'ordre ces instructions.

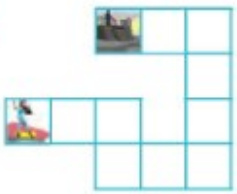
### 2 Julie la skateuse Niveau 2

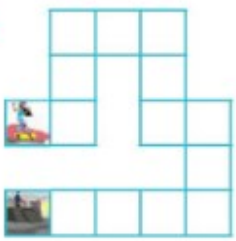
Ici, pour que Julie la skateuse retrouve le skate park, elle doit suivre la séquence d'instructions suivante : ▶ ▶ ▲ ▶



Dans chacun des cas ci-dessous, écrire l'algorithme permettant à Julie de retrouver le skate park à l'aide des quatre instructions : ◀ ▶ ▲ ▼

1. 

2. 

3. 

### 3 Le chemin du collège Niveau 3

On peut programmer les algorithmes avec un logiciel dédié comme le logiciel **Scratch**. Ce logiciel, avec lequel nous allons apprendre à travailler toute l'année, est téléchargeable et utilisable gratuitement : http://scratch.mit.edu/

**Exemple**

Ce programme permet de faire tracer au lutin un carré de côté 100 à chaque fois que l'utilisateur appuie sur 


Entre ce programme dans Scratch et exécute-le afin de vérifier qu'il produit bien le résultat attendu.

## Construire des rectangles Niveau 1

Programmer un algorithme permettant de tracer un rectangle de longueur 100 et de largeur 80, puis enregistrer ce programme.

Modifier le programme en faisant dire au chat : « Je vais tracer un rectangle... » avant qu'il ne commence.

Tu peux ici mettre une temporisation : attendre 1 secondes pour voir le lutin tracer chaque segment.

**Aide**

Utiliser l'instruction dire Hello! pendant 2 secondes

Améliorer le programme en masquant le chat une fois que le rectangle est tracé.

**Aide**

Utiliser les instructions cacher et montrer

Améliorer le programme en traçant chaque côté du rectangle d'une couleur différente.

**Aide**

Utiliser l'instruction choisir la couleur pour le stylo ou mettre la couleur du stylo à 0

## Les signes opératoires Niveau 2

Programmer un algorithme permettant de dessiner le signe d'addition ci-contre, puis enregistrer ce programme.

**Aide**

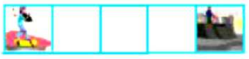
Pour que le lutin démarre toujours au même endroit, on peut utiliser l'instruction aller à x: 0 y: 0 et s'orienter à 90

Programmer un algorithme permettant de dessiner un signe de multiplication, puis enregistrer ce programme.

Améliorer les deux programmes précédents pour que les quatre branches

Dans un algorithme, une **boucle** consiste à faire répéter un certain nombre de fois (connu à l'avance ou non) une même séquence d'instructions. Il existe principalement deux types de boucles : la boucle « Répéter *x* fois » et la boucle « Répéter jusqu'à ».

**Exemple :** Pour aider Julie à rejoindre le skate park, on peut donner comme instructions ▶ ▶ ▶ ▶.



On peut aussi utiliser des boucles :

« Répéter 4 fois ▶ »

On utilise la boucle « Répéter *x* fois » quand on sait déjà combien de fois on doit faire répéter les instructions.

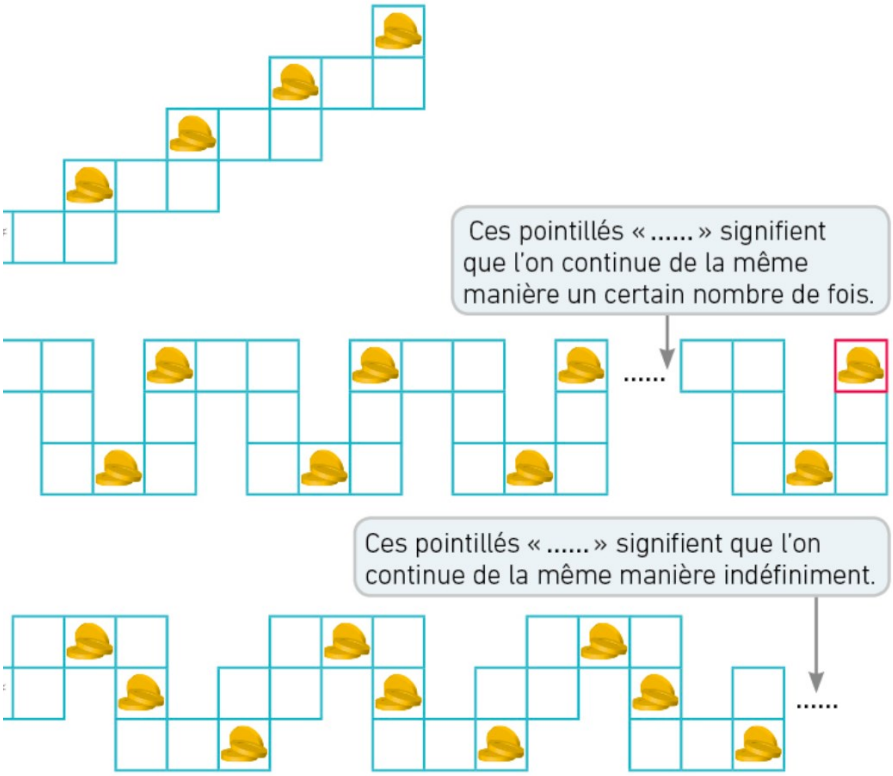
« Répéter ▶ jusqu'à *"Le skate park est atteint"* »

On utilise la boucle « Répéter jusqu'à » quand on ne sait pas combien de fois on doit répéter les instructions mais quand on sait à quel moment on doit s'arrêter.

## 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.




**Question 1.**

.....

.....

.....

**Question 2.**

.....

.....

.....

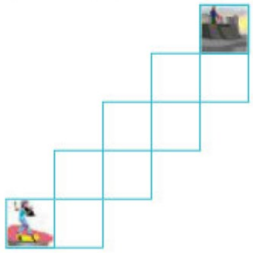
**Question 3.**

.....

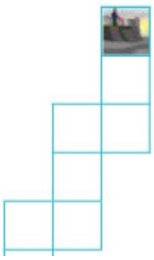
.....

.....

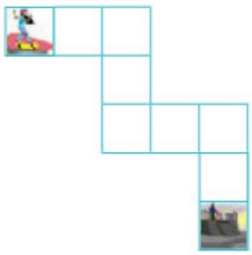
Dans chacun des cas ci-dessous, écrire un algorithme permettant à Julie d'atteindre le skate park à l'aide des quatre instructions : ◀ ▶ ▲ ▼ et un autre algorithme utilisant une boucle.



2.



3.



## cice 2 Les caisses automatiques Difficulté ★★

les supermarchés actuels, on trouve de plus en plus de caisses automatiques qui remplacent p
sivement les hôte(sse)s de caisse. Pour rendre la monnaie aux clients, ces automates en libre-s
suivent et produisent des algorithmes.

l'aide de boucles et des deux
uctions : **Donner une pièce de 1 DT** et
**er une pièce de 20 cents**, à n'utiliser
ne seule fois chacune, écrire un
ithme qui permette de rendre 3,80 DT
ient.

distributeur peut donner des pièces
T, 1 DT, 50 cents, 20 cents, 10 cents,
nts, 2 cents ou encore 1 cent.
e un algorithme contenant des
les qui permette de rendre 9,48
onnant le moins de pièces possible
ient.

**Question 1.**

.....

.....

.....

.....

**Question 2.**

.....

.....

.....

.....

.....

.....

.....

.....

## cice 3 Compétition de natation Niveau 3

Dans chaque cas, en utilisant une boucle « ***Répéter x fois*** » et l'instruction « ***Traverser le bassin*** », écrire un algorithme décrivant le parcours d'une nageuse effectuant en grand bassin (50 m de longueur) :

**a.** un 100 m nage libre ; **b.** un 800 m nage libre.

## cice 4 Un motif coloré

e dans l'emplacement ci-contre un
ramme qui permette de réaliser le
in ci-dessous où les 5 motifs ont des
eurs différentes.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Les différents types de **boucles** de Scratch sont dans le menu Contrôle .

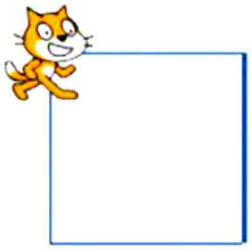

• On peut indiquer le nombre de répétitions souhaitées : ce nombre peut être une variable.


• La répétition est ici effectuée jusqu'à ce qu'un test soit validé. Ces tests sont dans le menu Opérateurs . On a, par exemple :  ,  ou encore  .


• Cette boucle est utilisée par exemple quand on attend une réponse au clavier. Cela nécessitera généralement l'usage d'une instruction conditionnelle : elle **sera placée dans ce bloc**.

**Exemple**

Ce programme va permettre de faire tracer un carré de côté 100 à chaque fois que l'utilisateur appuie sur  .

Entre ce programme dans Scratch et exécute-le afin de vérifier qu'il produit bien le résultat attendu.

## Construire des polygones Niveau 1


Simplifier le programme ci-contre à l'aide d'une boucle pour construire un pentagone (polygone à 5 côtés), puis enregistrer ce programme.

Faire évoluer ce programme afin d'obtenir un hexagone.

Faire évoluer ce programme afin d'obtenir un triangle équilatéral.

## Les décompteurs Niveau 2

Programmer un compteur qui va de 1 jusqu'à 20 et s'arrête dès que la valeur 20 est atteinte.

**Aide**

Utiliser l'instruction  pour voir le décompte défiler.

Améliorer le programme en faisant compter le chat au fur et à mesure.

Modifier le programme en faisant un décompte de 20 jusqu'à 0.

Programmer un décompte de 7 en 7 à partir de 343 qui s'arrête quand 0 est atteint.

## Le marquage au sol Niveau 3

ogrammer un algorithme qui reproduit le dessin contre. Chaque segment mesure 20 et chaque pace mesure 20 également.


## Un arc-en-ciel Niveau 4

faisant évoluer la valeur de la couleur de 0 à 200, réaliser ce nuancier :

**Aide**

On peut réduire la taille du chat et

Une **instruction conditionnelle** est de la forme :

**Si « Condition »**

**Alors « Instruction(s) »**

Dans ce cas, si la **« Condition »** est réalisée, alors les « Instruction(s) » seront effectuées.

**Exemple**
**Si le soleil brille cet après-midi, alors** j'irai à la piscine.

**ou**

**Si « Condition »**

**Alors « Instruction(s) 1 »**

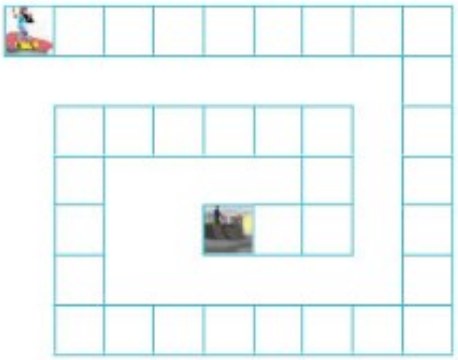
**Sinon « Instruction(s) 2 »**

Dans ce cas, si la **« Condition »** est réalisée, alors les « Instruction(s) 1» seront effectuées, sinon ce sont les « Instruction(s) 2 » qui seront effectuées.

**Exemple**
**Si le soleil brille cet après-midi, alors** j'irai à la piscine, **sinon** j'irai au cinéma.

## Les instructions de Julie Niveau 1

agit ici d'écrire des algorithmes permettant à Julie d'atteindre le skate park utilisant le
ructions avancer d'une case , tourner à droite , des boucles et des instructior
ditionnelles avec la condition Pas de case devant .



crire un algorithme en utilisant une instruction conditionnelle de type « **Si** ... **Alors** ... . »

crire un algorithme en utilisant une instruction conditionnelle de type « **Si** ... **Alor**
. **Sinon** ... . »

## Les tirs au but Niveau 2

urant la Coupe du monde de football, à la fin du
emps réglementaire d'un match, quand les équipes
ont à égalité, le match continue durant une période
ue l'on appelle prolongations.

crire un algorithme utilisant une instruction
onditionnelle « **Si** ... **Alors** ... » pour décrire cette
ituation.

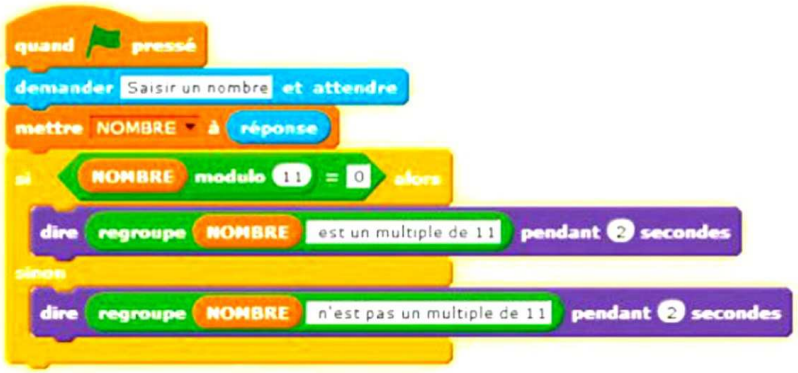

la fin des prolongations, si les équipes n'ont pas réussi à se départager, elles entame
ne séance de tirs aux but. Chaque équipe tire cinq fois.
i à la fin de cette séance, les équipes sont toujours à égalité, vient la « mort subite ».
haque équipe tire au but une fois jusqu'à ce que l'une marque et l'autre non.

Les différents types d'instructions conditionnelles de Scratch sont dans le menu Contrôle.


Les conditions sont à construire à partir des éléments du menu Opérateurs comme < , = ou >.

**Exemple**

Ce programme va permettre de dire si un nombre est ou n'est pas un multiple de 11.




**Aide**

L'opérateur modulo donne le reste dans la division euclidienne du premier nombre par le second.


Entre ce programme dans Scratch et exécute-le afin de vérifier qu'il produit bien le résultat attendu.

## 3 Un carré pas trop grand Niveau 1

- Programmer un algorithme demandant un nombre entre 0 et 200.
  Si le nombre entré est inférieur à 200, le chat doit dire : « Merci ! »
  Enregistrer le programme.
- Modifier ce programme pour qu'il demande la longueur du côté d'un carré et qu'il le trac uniquement si le côté est plus petit que 200.
- Améliorer le programme pour que dans le cas où il ne trace pas le carré, le chat dise « Désolé ce nombre est trop grand ! »

## 4 Un nombre au hasard Niveau 2

- Programmer un algorithme qui met dans une variable Nombre une valeur aléatoire de 1 à 3, puis qui cache cette variable à l'écran.
- Compléter le programme pour qu'il demande ensuite à l'utilisateur de choisir un nombre entre 1 et 3, puis faire dire au chat si les deux nombres sont identiques ou non.

## 5 La balade du chat Niveau 3

rogrammer un algorithme faisant déplacer le hat vers la droite ou vers la gauche à l'aide des èches du clavier.

**Aide**

Utiliser l'instruction touche flèche droite pressée?

ainsi que la boucle répéter indéfiniment .

Une **variable** est une boite dans laquelle on stocke une information pour l'utiliser plus tard. On désigne une variable par un nom.

**Exemple :** Le score d'un joueur est une variable qui pourra évoluer tout au long du jeu, une nouvelle valeur efface la précédente.



## Le jeu des pièces d'or

Niveau 1

jeu se joue à deux joueurs ou plus.

**atériel** : un dé à six faces numérotées de 1 à 6.

**ègle du jeu** : Au début d'une partie, chaque joueur reçoit 50 pièces d'or.
ut au long du jeu, ce nombre de pièces va évoluer et les joueurs vont devoir gére nombre de pièces qu'ils possèdent. Pour cela, chaque joueur prend une feuille blanche our noter son nouveau nombre de pièces en barrant l'ancien à chaque fois.
joueur qui débute la partie doit aussi comptabiliser, à un endroit de sa feuille, le nombre tours effectués. À tour de rôle, en tournant dans le sens des aiguilles d'une montre aque joueur lance le dé et exécute l'action correspondant au résultat du dé :

| | |
|---|---|
|  | **Échange tes pièces avec un des joueurs qui en a le plus.** |
|  | **Ton nombre de pièces augmente de 10.** |
|  | **Échange tes pièces avec le joueur de ton choix.** |
|  | **Ton nombre de pièces diminue de 10.** |
|  | **Donne 10 pièces au joueur qui joue juste après toi** |
|  | **Ton nombre de pièces est multiplié par 2.** |

partie se joue en 10 tours, le joueur qui a le plus de pièces à la fin de ces 10 tours gagne

re une ou plusieurs parties avec des camarades de la classe en se plaçant par groupe our d'une table. Une fois la partie terminée, répondre aux questions ci-dessous.

Dans ce jeu, il y a plusieurs **variables** :
- une variable qui contient le nombre de tours effectués. On peut l'appeler « TOURS » ;
- des variables qui contiennent les nombres de pièces de chaque joueur, on leur donner comme nom le « prénom » de chaque joueur.

nbien de variables ont été utilisées dans cette partie ?

Avec le logiciel Scratch, on peut créer des variables à partir du menu **Données**.

On commence par choisir le nom de ses variables.

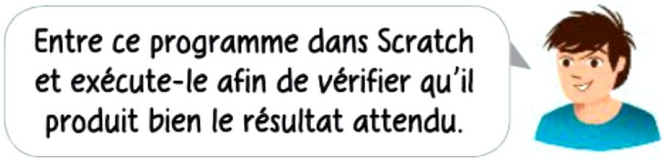


Dès que la variable est créée, plusieurs actions se découvrent.


**Exemple**

Ce programme va demander un nombre et donner son double à chaque fois que l'utilisateur appuiera sur 




## On compte Niveau 1


Créer une variable « COMPTEUR » et écrire un programme qui fasse augmenter de 1 cette variable à chaque clic sur le drapeau vert. Enregistrer ce programme.

Améliorer le programme en faisant dire au chat : « Le compteur est arrivé à ... »


## Le périmètre d'un rectangle Niveau 2

Programmer un algorithme demandant la longueur du côté d'un carré avant de le tracer. Enregistrer ce programme.


Modifier ce programme en demandant une longueur et une largeur avant de tracer un rectangle aux dimensions saisies.

Faire évoluer ce programme en créant une variable « PERIMETRE » et en faisant dire à Scratch à la fin du tracé : « Le périmètre de ce rectangle est égal à ... »


## Qui es-tu ? Niveau 3


Un **bloc d'instructions** peut parfois être remplacé par un seul intitulé.

**Exemple**

L'envoi d'un courriel par messagerie est constitué de plusieurs actions simples, intégrées dans un même bloc nommé « Envoi d'un courriel ».

| Envoi d'un courriel |
| --- |
| • Ouvrir sa messagerie<br>• écrire à l'aide des touches du clavier de l'ordinateur<br>• Choisir la ligne destinataire marquée « à » ou « pour »<br>• Ecrire l'adresse du destinataire<br>• Cliquer sur « envoyer » |

**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.

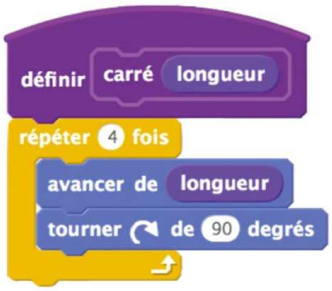

**Exemple ② :** Leïla souhaite tracer un carré de longueur donnée.

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.




## 1 Travaux ménagers Niveau 1

En utilisant les instructions suivantes, écris deux groupements : l'un correspondant à « mettre la table pour une personne » et l'autre correspondant à « préparer une lessive ».

Mettre un verre

Mettre le linge dans le tambour

Préparer un pichet d'eau

Mettre les couverts

Mettre la lessive

Poser une assiette sur la table

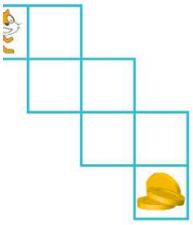
Mettre un adoucisseur si besoin

Choisir le programme de lavage

# Utilisation d'un bloc d'instructions

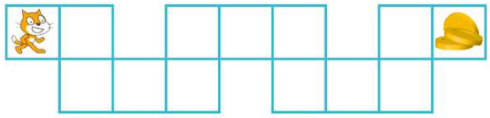
## cice 1 Le labyrinthe Difficulté ★

chaque labyrinthe, aider Scratchy à récupérer toutes les pièces d'or en définissant le bloc Prog1 c
épètera deux ou trois fois. On utilisera les instructions de déplacement : ◀, ▶, ▲ et ▼.



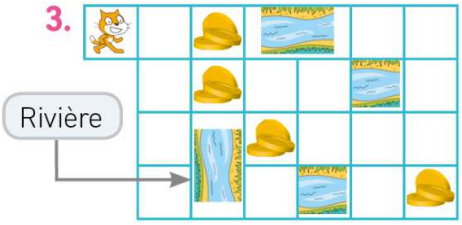
Répéter 3 fois
Prog1
)éfinir Prog1 :
.....

2.



Répéter 2 fois
Prog1
Définir Prog1 :
.....

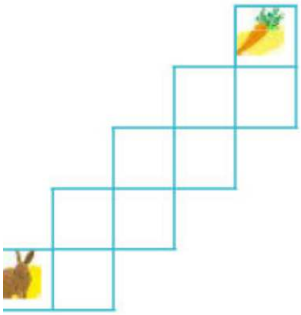
3.




Répéter 3 fois
Prog1
Définir Prog1 :
.....

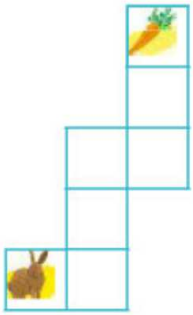
## Le repas du lapin Niveau 2

s chacun des cas ci-dessous, écrire le groupement d'instructions (GI) permettant au
n de retrouver sa carotte à l'aide des quatre instructions ◀ ▶ ▲ ▼



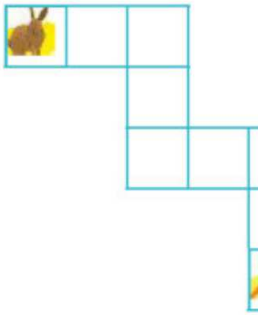
éter 4 fois
oupement
nstructions GI1

2.



Répéter 2 fois
{ Groupement
{ d'instructions GI2
Fin

3.



Répéter 2 fois
{ Groupement
{ d'instructions GI3
Fin

Utiliser un bloc d'instructions ici permet d'éviter d'avoir à mettre une boucle dans une boucle.

Un bloc permet de mémoriser un petit algorithme intervenant dans un algorithme plus élaboré, éventuellement plusieurs fois, ce qui facilite également la compréhension de l'algorithme principal.
On peut créer des blocs à partir du menu **Ajouter blocs**.
Lorsqu'il est créé, le bloc est utilisable comme les autres éléments.

**Exemple**

Le bloc carré permet de tracer, à chaque fois qu'il est appelé, un carré de côté 100 à l'endroit où se trouve le lutin.

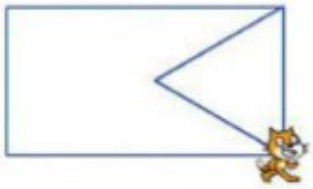

Entre ce programme dans Scratch et exécute-le afin de vérifier qu'il produit bien le résultat attendu.

## Géométries d'une enveloppe — Niveau 1

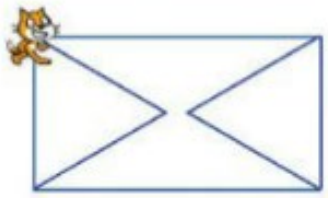
Écrire un algorithme définissant un bloc rectangle de 80 sur 150 et un bloc triangle équilatéral de côté 80.

Utiliser alors ces blocs de façon à obtenir les figures ci-dessous :

a.



b.



## Quel cirque ! Maximus ! — Niveau 2

Écrire un algorithme faisant appel à un bloc « triangle équilatéral » pour construire un hexagone.

**Aide**

Utiliser l'instruction répéter 10 fois

## Silence ! On tourne ! — Niveau 3

Créer un bloc « CARRÉ » qui trace un carré de côté 100 à partir de l'endroit où se trouve le lutin.

Programmer un algorithme qui efface tout, va au centre de l'écran (0 ; 0), pose le stylo et répète 36 fois la même séquence :


Améliorer le programme pour que la couleur du stylo change à chaque carré.

**Aide**

On pourra par exemple utiliser : ajouter 1 à la taille du styl

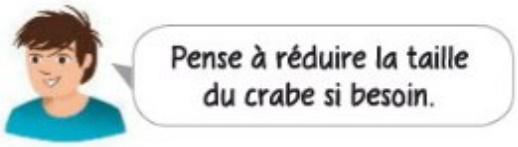
La couleur dans Scratch correspond à des nombres entiers compris entre 0 et 200.

# Le crabe aux pinces magiques

Dans un univers maritime, l'antre d'un crabe est attaqué par des ballons ! Heureusement, ses pinces magiques lui permettent de détruire ces ballons avant qu'ils n'atteignent leur but !

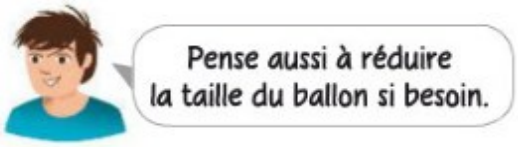
## Étape 1 ■ Positionner et déplacer le crabe

- Prendre comme lutin un crabe et le positionner en bas de l'écran.
- Programmer le déplacement horizontal (gauche et droite) du crabe à l'aide des flèches du clavier.




## Étape 2 ■ Positionner et déplacer un ballon

- Ajouter un ballon qui descend automatiquement en démarrant toujours de la même abscisse –170. Cette abscisse sera modifiée par la suite.




## Étape 3 ■ Programmer les actions

- Modifier le programme pour que lorsque le crabe touche le ballon, celui-ci réapparaisse en haut à son point de départ.
- Prévoir aussi un retour au point de départ si le ballon arrive en bas de la scène, sans être touché par le crabe.

## ›e 4 ■ Ajouter un décor et un compteur

›uter un compteur permettant de savoir ›mbien de fois le crabe a touché le ballon.

›uter aussi un arrière-plan adapté.

Un double-clic sur

compteur 0

permet d'obtenir cet affichage 0. On obtient le même résultat en faisant un clic droit et en choisissant dans le menu déroulant.

## ›e 5 ■ Faire apparaitre le ballon n'importe où

›re en sorte que le ballon puisse surgir n'importe quelle abscisse entre –200 +200.

C'est l'occasion d'utiliser l'élément

nombre aléatoire entre ● et ●

## ›e 6 ■ Créer une fin pour le jeu

›rminer le jeu avec les indications ›vantes :

›sque le compteur atteint 5, augmenter ›vitesse de déplacement du crabe ainsi ›e la vitesse de descente des ballons ;

›sque le compteur atteint 10, augmenter ›core la vitesse de descente des ballons ;

›sque le compteur atteint 20, faire dire crabe : « Gagné ! » et arrêter le jeu.


**›LUTIONS POSSIBLES**

›aire varier la couleur des ballons, le bleu valant plus de points que les autres.

# Le mage et la grenouille

Un mage doit viser une grenouille avec sa baguette magique.
Lorsque la grenouille a reçu assez de magie, elle se transforme en prince ou en princess

## Étape 1 ■ Préparer la grenouille

- Choisir comme lutin la grenouille de Scratch et aller dans l'onglet costumes.
- Dupliquer la grenouille.
- Avec l'outil Gomme, supprimer la langue sur le costume 1.

Pour plus de précision dans l'usage de la gomme, on peut utiliser l'outil loupe en bas à droite de l'écran.

100%

## Étape 2 ■ Le mage et sa magie

- Écrire un algorithme de déplacement de la grenouille dans les quatre directions. On pourra utiliser

touche flèche haut pressée?

- Ajouter le lutin *Wizard* (le nommer *mage*) et créer un lutin *magie* de ce style :

- Créer une variable x mage et une variable y mage.
- Attribuer à x mage une valeur aléatoire entre –210 et 60 et à y mage une valeur aléatoire entre –120 et 120.
- Créer une variable angle qui pourra prendre une valeur aléatoire entre –30 et +30. Faire tourner le *mage* de cette valeur.

## Étape 3 ■ Introduction du hasard dans le jeu

- Créer une variable dé, lui attribuer une valeur aléatoire entre 1 et 10.

nombre aléatoire entre 1 et 10

Si la valeur de dé est inférieure à 8, le *mage* envoie de la magie par sa baguette : à cet effet, un message est envoyé au lutin *magie*.

Prévoir ensuite une temporisation d'une seconde environ.

Sinon, on refait bouger le mage et on relance le dé

## ...e 4 ■ Traitement de l'événement victorieux

...s que la grenouille est touchée ... *magie*, elle tremble en tirant ...rentrant la langue.

...bjectif est d'être touché le plus ...uvent possible par *magie*.

compteur doit donc mémoriser ...ombre de fois où la grenouille ... touchée.


## ...e 5 ■ Traitement de la fin de partie

...eut désormais cacher le ...teur.

...rtir de la 11e fois où la ...ouille a été touchée :
...e dé donne 1, la grenouille se ...sforme en prince (autre lutin), ...jeu s'arrête ;
...e dé donne 2, la grenouille ...ansforme en princesse (autre ... et le jeu s'arrête ;
...s les autre cas, le jeu ...nue.

## ...e 6 ■ Comptabilisation des points

...oisir un arrière-plan.

...rminer le jeu en créant une ...riable « POINTS ».

... démarrage, on a 1 000 ...nts.

mettre points à 1000

...haque fois que la grenouille ...mble, on ajoute 200 points.
...haque fois que la grenouille ...st pas touchée, on perd 100 ...nts.
...rsque le compteur dépasse ... 10, on a 20 points à chaque ...jusqu'à la transformation.


### ...LUTIONS POSSIBLES

...aire varier la taille de la grenouille en fonction de la magie reçue.

# Le chiffre de César

Crypter… décrypter… voilà ce à quoi s'attaque le chiffre de César !
Cette forme simple de cryptage était déjà utilisée dans l'Antiquité.
À ton tour de chiffrer un message comme les Romains !

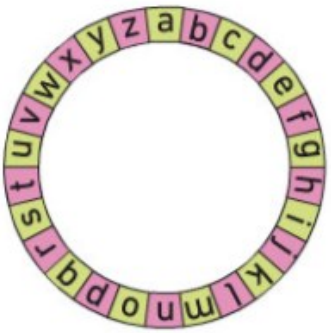
## Étape 1 ■ Création et stockage de l'alphabet

– Le chiffre de César est un procédé qui consiste à décaler les lettres de l'alphabet vers la droite ou vers la gauche d'un nombre de crans déterminés.

On choisit ici de décaler vers la droite.


– Créer une variable **alphabet** contenant les 26 lettres de l'alphabet dans l'ordre.

Il s'agira de reconstituer le principe du cer de lettres ci-dessous à partir des 26 lettres l'alphabet dans l'ordre mises dans la varial




## Étape 2 ■ Saisie du décalage et du message à coder

– Demander le nombre correspondant au **décalage** souhaité.

– Demander le **message** à coder.

**Attention !** Ce message sera tout d'abord un mot simple, c'est-à-dire sans trait d'union, sans apostrophe, etc.
On dispose désormais du message (mot) à coder et du décalage souhaité : il faudra, à l'étape suivante, créer le message codé.


## Étape 3 ■ Création des variables nécessaires au codage

– Créer une variable **message codé** qui sera initialisée par du vide.

– Créer ensuite une variable **lettre** qui prendra successivement comme valeur chacune des lettres du mot à coder.

– Créer une variable **compteur** qui permettra de


## pe 4 ■ Création du bloc permettant le codage

affecte à la variable [lettre] la lettre à coder.

appelle un bloc [CODER] qui va :

déterminer la position de cette lettre dans l'alphabet
t la stocker dans une variable [position], en utilisant
ventuellement l'instruction [lettre 1 de world] ;

ajouter le [décalage] à la variable [position] pour
rouver la position de la lettre codée ;

affecter à la variable [lettre] sa nouvelle valeur.

Par exemple avec la lettre **b** et un décalage de **3**, on aura comme valeur successives des variables :

- lettre b
- CODER
- position 2
- position 5
- lettre e

**tention !** Dans le cas où la somme [position] + [décalage] dépasse 26, il faut trouver un moyen
redémarrer au début de l'alphabet (27 correspond à 1, 28 à 2, etc.)
ur cela, utiliser l'instruction [modulo] qui donne le reste d'une division euclidienne.

## pe 5 ■ Création et affichage du message codé

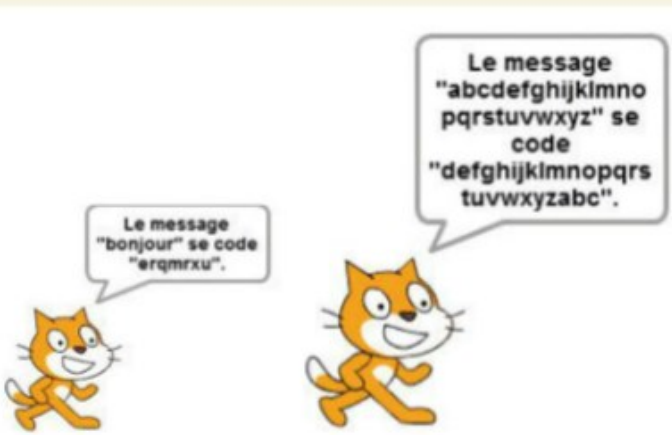
éer une boucle qui va, pour chacune
s lettres du message initial, trouver
lle qui doit la remplacer à l'aide du
oc [CODER] et l'ajouter à la suite du
essage codé.

pourra utiliser pour la boucle :

[ongueur de world] et [répéter fois]

pour construire le message codé :

[regroupe message codé lettre]




rifier que le codage fonctionne bien, par exemple en tapant l'alphabet entier comme message
ec un décalage de 3 pour observer si chaque lettre se code bien.

## pe 6 ■ Amélioration du programme pour coder des phrases entières

odifier le bloc [CODER] pour prendre en compte les
paces, les apostrophes ou les traits d'union dans une
rase.

le programme ne trouve pas la lettre à coder dans
lphabet lors de la recherche de position, il doit
nserver la lettre d'origine.

espace sera donc codé par un espace, une
ostrophe par une apostrophe, etc.

pourra pour cela utiliser [ou] dans la
ucle.


**OLUTIONS POSSIBLES**

ntégrer un codage spécifique pour les lettres accentuées.