

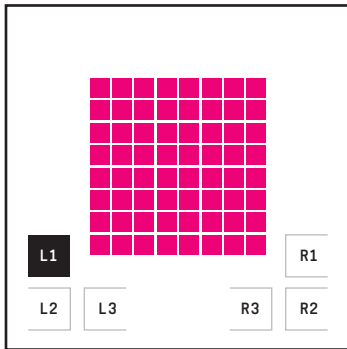
# FICHE A019

## LISTE

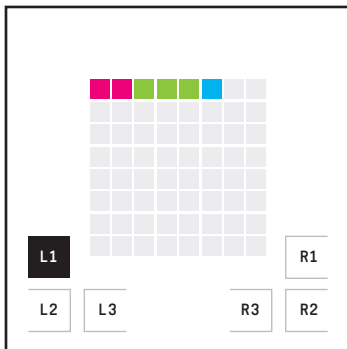
Dans cette fiche, nous allons aborder le fonctionnement de la liste de couleurs.

## CONSIGNE

Attribuer à chaque bouton de gauche (L1, L2 ou L3) une couleur à afficher courtement et à insérer dans la liste de couleurs. Si l'on appuie sur un bouton de droite, par exemple R1, le programme doit afficher le contenu de la liste de couleurs en affichant la suite de pixels colorés correspondante.



Le bouton L1 affiche par exemple courtement du rouge et l'enregistre dans la liste de couleurs. Les boutons L2 et L3 peuvent correspondre au vert, respectivement au bleu.

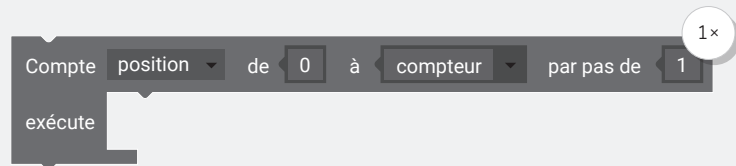
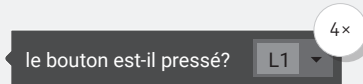
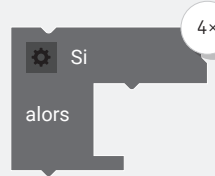


Le bouton R1 montre ensuite les couleurs stockées dans la liste. Dans le cas présent, l'utilisateur a appuyé 2 fois sur L1, 3 fois sur L2 et une fois sur L3.

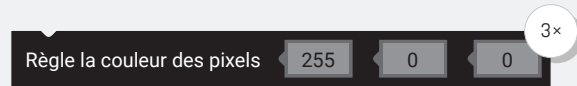
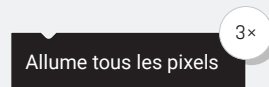
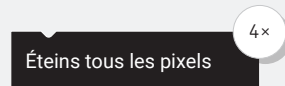
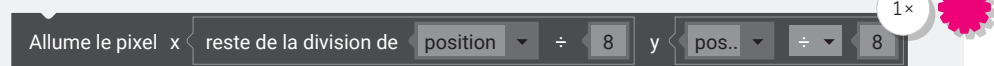
## CONSEIL

Utilise le modèle présenté dans la liste de blocs ci-contre pour dessiner les pixels dans la boucle. Il ne te reste ensuite plus qu'à lire couleur correspondante de la liste de couleurs et à l'afficher.

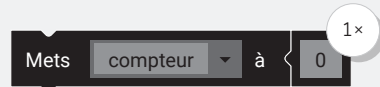
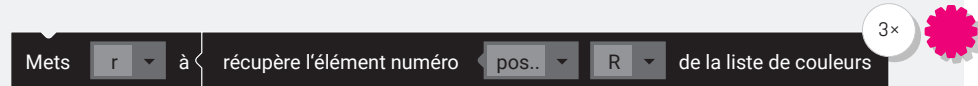
### LOGIQUE



### MATRICE LED



### VARIABLES



### TEMPS



### LISTE DES PIÈCES LISTE DES BLOCS À UTILISER



NIVEAU DIFFICILE

# SOLUTION

PROPOSITION DE SOLUTION

The image shows a Scratch script designed to cycle through four colors (red, green, blue, and grayscale) based on button presses. The script is organized into four main sections, each triggered by a different button (L1, L2, L3, and R1).

- Initial Setup:** A 'Mets' block sets the 'compteur' variable to 0.
- Red Cycle (L1):** When L1 is pressed, it sets RGB to (255, 0, 0), increments the counter, sets the color, turns on pixels, waits 500ms, and turns off pixels.
- Green Cycle (L2):** When L2 is pressed, it sets RGB to (0, 255, 0), increments the counter, sets the color, turns on pixels, waits 500ms, and turns off pixels.
- Blue Cycle (L3):** When L3 is pressed, it sets RGB to (0, 0, 255), increments the counter, sets the color, turns on pixels, waits 500ms, and turns off pixels.
- Grayscale Cycle (R1):** When R1 is pressed, it loops from position 0 to the counter value. For each position, it retrieves the R, G, and B values from a 'liste de couleurs' and sets the pixel color to grayscale based on their average. It then turns on the pixel, waits 50ms, and turns it off. After the loop, it waits 3000ms and turns off all pixels.

```
graph TD
    Start[Mets compteur à 0] --> Loop[Répéter pour toujours]
    Loop --> L1[Si le bouton est-il pressé? L1]
    L1 -- alors --> L1_1[Insère R: 255 G: 0 B: 0 dans la liste de couleurs]
    L1_1 --> L1_2[Additionne 1 à compteur]
    L1_2 --> L1_3[Règle la couleur des pixels 255 0 0]
    L1_3 --> L1_4[Allume tous les pixels]
    L1_4 --> L1_5[Attends 500 millisecondes]
    L1_5 --> L1_6[Éteins tous les pixels]
    L1_6 --> L2[Si le bouton est-il pressé? L2]
    L2 -- alors --> L2_1[Insère R: 0 G: 255 B: 0 dans la liste de couleurs]
    L2_1 --> L2_2[Additionne 1 à compteur]
    L2_2 --> L2_3[Règle la couleur des pixels 0 255 0]
    L2_3 --> L2_4[Allume tous les pixels]
    L2_4 --> L2_5[Attends 500 millisecondes]
    L2_5 --> L2_6[Éteins tous les pixels]
    L2_6 --> L3[Si le bouton est-il pressé? L3]
    L3 -- alors --> L3_1[Insère R: 0 G: 0 B: 255 dans la liste de couleurs]
    L3_1 --> L3_2[Additionne 1 à compteur]
    L3_2 --> L3_3[Règle la couleur des pixels 0 0 255]
    L3_3 --> L3_4[Allume tous les pixels]
    L3_4 --> L3_5[Attends 500 millisecondes]
    L3_5 --> L3_6[Éteins tous les pixels]
    L3_6 --> L4[Si le bouton est-il pressé? R1]
    L4 -- alors --> L4_1[Compte position de 0 à compteur par pas de 1]
    L4_1 -- exécute --> L4_2[Mets r à Récupère l'élément numéro position R de la liste de couleurs]
    L4_2 --> L4_3[Mets g à Récupère l'élément numéro position G de la liste de couleurs]
    L4_3 --> L4_4[Mets b à Récupère l'élément numéro position B de la liste de couleurs]
    L4_4 --> L4_5[Règle la couleur des pixels r g b]
    L4_5 --> L4_6[Allume le pixel x Reste de la division de position ÷ 8 y position ÷ 8]
    L4_6 --> L4_7[Attends 50 millisecondes]
    L4_7 --> L4_8[Attends 3000 millisecondes]
    L4_8 --> L4_9[Éteins tous les pixels]
```

# FICHE A019

## Objectifs d'apprentissage:

Les élèves découvrent le fonctionnement d'une liste.

## STRATÉGIE DE RÉSOLUTION

1.

On commence par initialiser la variable de comptage à zéro.

2.

On introduit ensuite la boucle infinie qui contiendra tous les autres blocs du programme.

3.

On place dans cette boucle tous les blocs « Si/alors » qui se chargent d'interroger l'état des différents boutons.

Dans les trois premières parties « alors », on insère à chaque fois la couleur correspondante dans la liste de couleurs, on incrémente la variable de comptage de 1 et on affiche la couleur correspondante pendant un court instant.

4.

Dans la dernière partie « alors », il faut lire un à un tous les éléments présents dans la liste de couleurs au sein d'une boucle et les afficher. Comme un élément de couleur possède à chaque fois trois composantes de couleur (R, G, B), il faut trois variables différentes (r, g, b). On détermine alors la couleur lue et on affiche le pixel dans cette couleur à la position appropriée. Pour créer un joli effet de remplissage, on peut encore ajouter au sein de cette boucle un bloc pour attendre courtement.

5.

En plus du bloc « compte dans position de 0 à compteur par pas de 1 », on utilise un autre bloc « attends » nécessaire pour que les pixels dessinés puissent être vus pendant un certain temps. Finalement, on peut de manière facultative effacer les pixels.



Ce symbole indique un « bloc expert » qui n'est donc disponible que lorsque l'option « activer le mode expert » est active dans les « Préférences ».

Pour cela, clique sur l'icône « Préférences » en-bas à gauche ...

### Préférences

... et enclenche « activer le mode expert ».

## NOUVELLES COMMANDES

Insère R: 255 G: 0 B: 0 dans la liste de couleurs

Le bloc « Insère RGB à la fin de la liste de couleurs » permet d'insérer une couleur quelconque dans une liste définie au préalable. Chaque appel suivant à ce bloc insère un nouvel élément dans la liste.

Récupère l'élément numéro 0 R de la liste de couleurs

Le bloc « Récupère l'élément numéro x de la liste de couleurs » permet d'accéder à un élément préalablement inséré dans la liste de couleurs. Comme une couleur est composée de trois valeurs, il est nécessaire d'utiliser le bloc à trois reprises pour accéder à chacune des composantes couleur, en particulier pour les assigner chacune à une variable différente.

À l'image de presque tout en programmation, les listes sont indexées à partir de zéro. Le premier élément de la liste est donc placé à la position 0 et non 1.

## COMPLÉMENT DE THÉORIE : LISTES

L'avantage des listes par rapport aux variables individuelles est qu'il est possible d'y insérer ou d'en supprimer un nombre quelconque d'éléments au cours de l'exécution du programme. Elles constituent en particulier un outil très utile lors de la programmation de jeux. Elles permettent de stocker et de retrouver l'état d'un grand nombre d'éléments du jeu sans devoir créer une nouvelle variable pour chaque nouvel élément. Cela permet par exemple d'enregistrer une chronologie ou une file d'attente.

