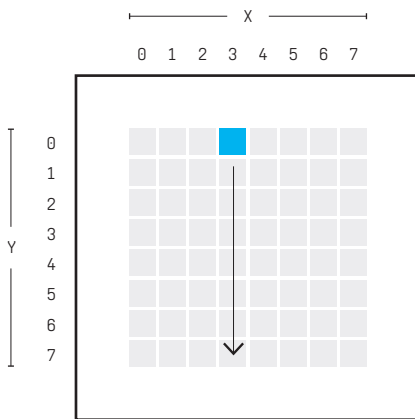


# FICHE A008 CONDITIONS

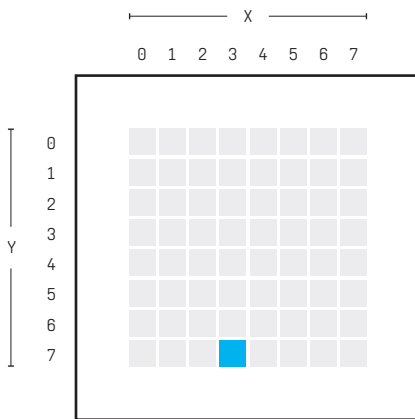
Il est aussi possible de créer des animations à l'aide d'algorithmes, ce qui permet de créer des effets intéressants et d'économiser et d'économiser de nombreux blocs.

## CONSIGNE

Programmer un pixel en train de tomber.



Dessine le pixel à l'aide d'une variable «y». Si la valeur de la variable est inférieure à 7, alors elle peut être incrémentée (augmentée) de 1.

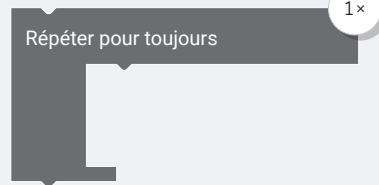


Le pixel a maintenant atteint le bas de l'écran et la valeur de la variable n'est donc plus inférieure à 7. La variable doit maintenant être remise à 0 pour que le pixel soit à nouveau dessiné en haut de l'écran.

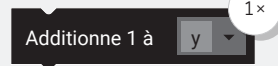
## CONSEIL

Ne dessine le pixel qu'une fois à chaque passage de la boucle. La variable «y» est mise à jour dans le bloc «Si/alors/sinon». Observe pour cela l'illustration ci-dessus.

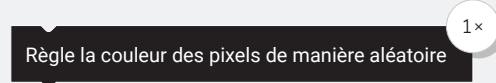
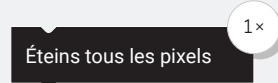
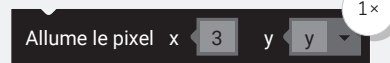
### LOGIQUE



### VARIABLES



### MATRICE LED



### TEMPS



### LISTE DES PIÈCES LISTE DES BLOCS À UTILISER



NIVEAU FACILE

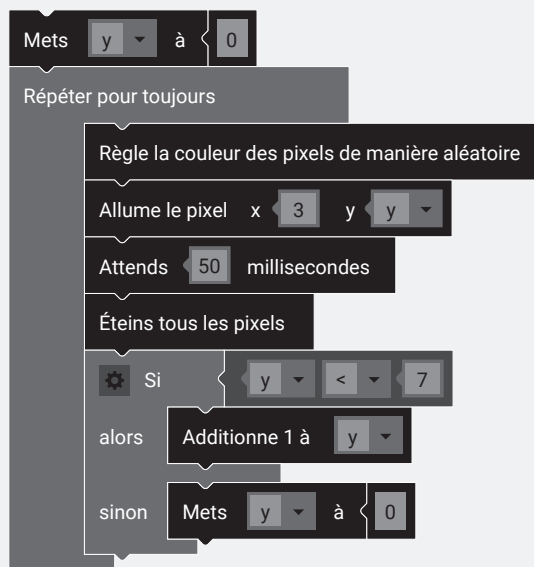
## DÉFI SUPPLÉMENTAIRE:

Fais en sorte que le pixel ne se déplace pas seulement de haut et bas mais encore de gauche à droite. Pour ce faire, incrémente également de 1 la position horizontale du pixel à chaque passage dans la boucle.

# FICHE A008 CONDITIONS

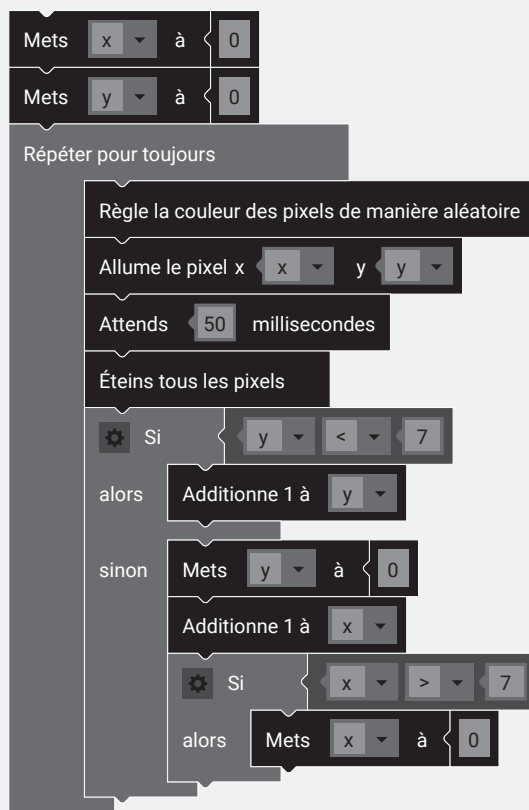
## SOLUTION

PROPOSITION DE SOLUTION



## DÉFI SUPPLÉMENTAIRE

PROPOSITION DE SOLUTION



# FICHE A008

## CONDITIONS

### Objectifs d'apprentissage:

Les élèves apprennent à mettre en place une instruction conditionnelle étendue.

## STRATÉGIE DE RÉSOLUTION

1. On commence par définir la variable « y » et l'initialiser à zéro. On crée ensuite une boucle infinie avec le bloc « Répète pour toujours ». Tous les autres blocs doivent être insérés dans cette boucle.
2. On choisit ensuite une couleur de pixel aléatoire et on dessine le pixel avec une coordonnée « x » constante et une coordonnée « y » variable. Le dessin du pixel nécessite aussi l'utilisation d'un bloc « attends » et d'un bloc « Efface tous les pixels ».
3. On peut ensuite mettre en place une condition permettant de tester si le pixel doit continuer à se déplacer ou s'il doit être remis à sa position de départ.
4. Tant que le pixel n'a pas atteint le bas de l'écran, il faut incrémenter la valeur de la variable. Dès que le bas de l'écran est atteint ou dépassé, on remet la variable à zéro. Étant donné que le pixel est redessiné à chaque passage dans la boucle, il donne l'impression de tomber ligne après ligne.

## NOUVELLES COMMANDES

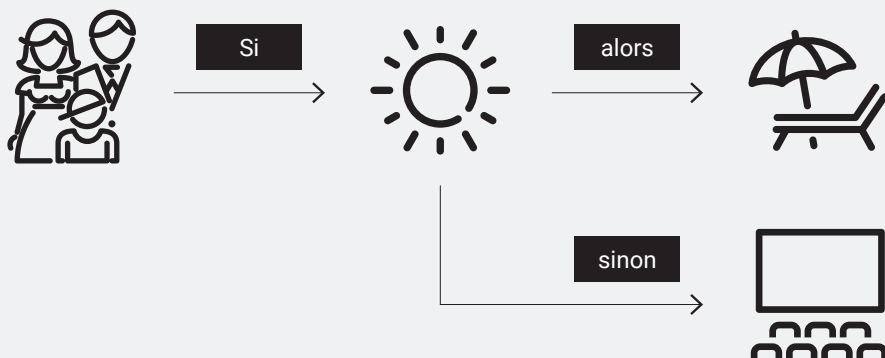


Ce bloc permet de comparer des variables, des nombres ou des calculs à l'aide des opérateurs de comparaison pour créer des conditions.



Ce bloc permet de créer branchements conditionnels. La partie « alors » est exécutée lorsque la condition est vraie et c'est la partie « sinon » qui est exécutée si la condition est fausse.

## COMPLÉMENT DE THÉORIE : SI/ALORS/SINON



**On reprend l'exemple de la sortie en famille :** lorsque le soleil brille, la famille va se baigner. Dans le cas contraire (sinon) elle va au cinéma.