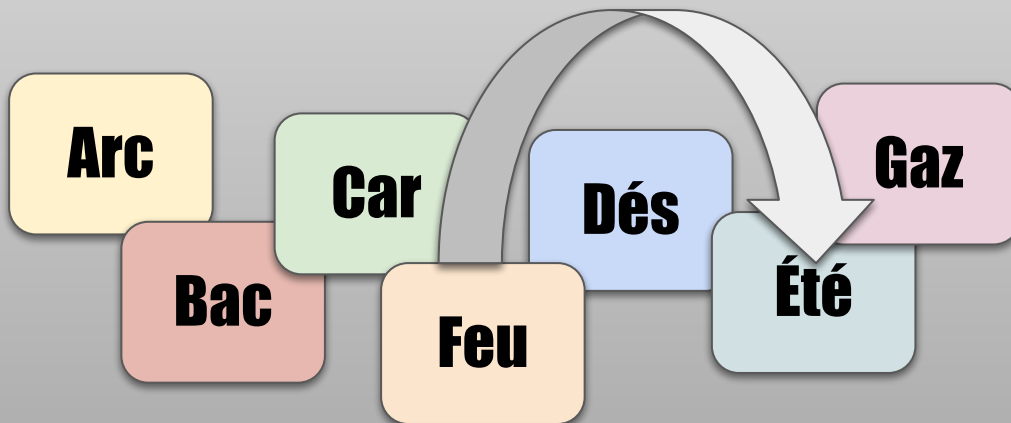


scratch.mit.edu

<https://www.reseau-canope.fr>



Ordonnement

Remettre les étiquettes dans le bon ordre

Février 2019

Pedag O'Scratch



Objectifs

Pour gérer les ordres, nous allons utiliser 2 structures de liste : *liste posée* et *liste correcte*. Créez ces listes.

Liste posée va d'abord servir à mélanger les cartes au début de la partie (Drapeau vert). Videz à nouveau cette liste pour la remplir de 0 (0 indique qu'aucune carte n'est posée dans cette case, par la suite elle contiendra les numéros des cartes posées). Vous entrerez alors à la main les bons nombres dans *liste correcte*, en cliquant l'affichage de la liste.

Évènements

Contrôle

Opérateurs

Variables

The image shows a Scratch script editor with the following code blocks:

- ajouter 0 à liste posée** (orange block)
- supprimer tous les éléments de la liste liste posée** (orange block)
- répéter 7 fois** (orange block) containing:
 - insérer longueur de liste posée + 1 en position nombre aléatoire entre 1 et longueur de liste posée + 1 de liste posée** (orange block)
- longueur de liste correcte** (orange block)
- quand drapeau vert est cliqué** (yellow block) containing:
 - envoyer à tous place et attendre** (yellow block)
- supprimer tous les éléments de la liste liste posée** (orange block)
- répéter fois** (orange block)

A small image of a green card is visible in the top right corner, labeled "Scène".

Le script déjà tout prêt est une méthode très courte mais pas intuitive pour mélanger une liste de 7 nombres dans une liste. Le «Envoyer à tous *place* et attendre» est l'instruction qui envoie le message à toutes les cartes de se placer à l'endroit attribué par leur numéro dans la liste de mélange.



Scène

```
when green flag clicked
  delete all elements of list [liste posée v]
  repeat (7) times
    insert (length of [liste posée v] + 1) at position (random number between 1 and length of [liste posée v] + 1) of [liste posée v]
  send all [place v] and wait
  delete all elements of list [liste posée v]
  repeat (length of [liste correcte v]) times
    add (0) to [liste posée v]
```

Événements

Contrôle

Opérateurs

Variables

Objectifs

Initialisation : remettre toutes les cartes à leur apparence normale. Créez une variable *monNuméro* avec l'option «Pour ce lutin uniquement». Donnez lui la valeur du numéro de carte.

Rangement : Quand il reçoit le message *place*, le lutin carte va se placer dans les coordonnées prévues : tous les Y valent 100 mais les X sont en fonction de leur place dans *liste posée*. Entre -195 et 195 avec un écart de 65 entre chaque carte, soit : -195, -130, -65, 0, 65, 130 et 195.

Mouvement

Apparence

Évènements

Opérateurs

Variables

mettre la taille à 100 % de la taille initiale

1

Carte1

annuler les effets graphiques

quand  est cliqué

mettre monNuméro à 1

élément # de dans liste posée

quand je reçois place

* 65

- 4

aller à x: y: 100

monNuméro

La formule mathématique peut sembler un peu barbare mais c'est un grand classique.
«élément # de *monNuméro* dans *liste posée*» veut dire : à quelle place *monNuméro* se trouve dans *liste posée*. Il s'agit en tout cas de la première occurrence.
On obtient 0 si *monNuméro* ne se trouve pas dans la liste.

Ex. :

- é # de 4 dans (3,2,4,1,4) = 3
- é # de 9 dans (3,2,4,1,4) = 0

quand  est cliqué

mettre la taille à 100 % de la taille initiale

annuler les effets graphiques

mettre monNuméro à 1

1

Carte1

quand je reçois place

aller à x: élément # de monNuméro dans liste posée - 4 * 65 y: 100

Mouvement

Apparence

Événements

Opérateurs

Variables

Objectifs

Nous allons ici faire une petite animation pour que les clics sur les cartes soient visibles, sonores et donc ergonomiques. Attention à vous mettre en plein écran pour tester votre programme, sinon vous êtes en vision «programmeur» et cela ne sera pas correct. Quand on clique sur une carte, on commence par l'enlever de *liste posée*, celle-ci se grise un peu, grandit un peu et se place au premier plan. Un son est émis. Ensuite la carte suit le curseur de la souris jusqu'à ce que le bouton de la souris ne soit plus pressée. Un son est émis et la carte reprend alors ses caractéristiques graphiques de départ.

Mouvement

Apparence

Son

Événements

Contrôle

Capteurs

Opérateurs

The image shows a Scratch script for a card animation. The script starts with a 'when this sprite is clicked' event block. The first block is 'replace element # monNuméro in list posée with list posée by 0'. This is followed by a 'when green flag clicked' block. The script then branches into two paths based on a 'mouse button pressed?' sensor. The 'yes' path consists of: 'add to effect' (set to 'ghost'), 'go to front plan', 'play sound Water Drop', and 'play sound pop'. The 'no' path consists of: 'add to size' and 'add to size'. A 'Carte1' label with the number 1 is in the top right corner.

L'ergonomie est une des choses les plus importantes pour l'utilisateur. On ne se rend parfois pas compte qu'il faut de nombreuses instructions pour qu'elle soit intuitive et transparente. Et c'est donc parfois une des choses les plus complexes à programmer. Vous allez vous en rendre compte dans les prochaines fiches. Pour l'instruction "remplacer...", vous remarquerez que de façon astucieuse, elle fonctionne si la carte est bien dans *liste posée* et si elle n'y est pas cela ne provoque pas de bogues non plus.

Mouvement

Apparence

Son

Événements

Contrôle

Capteurs

Opérateurs

1

Carte1

quand ce sprite est cliqué

remplacer l'élément élément # de monNuméro dans liste posée de la liste liste posée par 0

jouer le son pop

ajouter 50 à l'effet fantôme

ajouter 10 à la taille

aller à l'avant plan

répéter jusqu'à ce que non souris pressée ?

aller à pointeur de souris

jouer le son Water Drop

ajouter -10 à la taille

annuler les effets graphiques

ORDO4

Objectifs

Ces éléments vont à la suite du script construit précédemment.

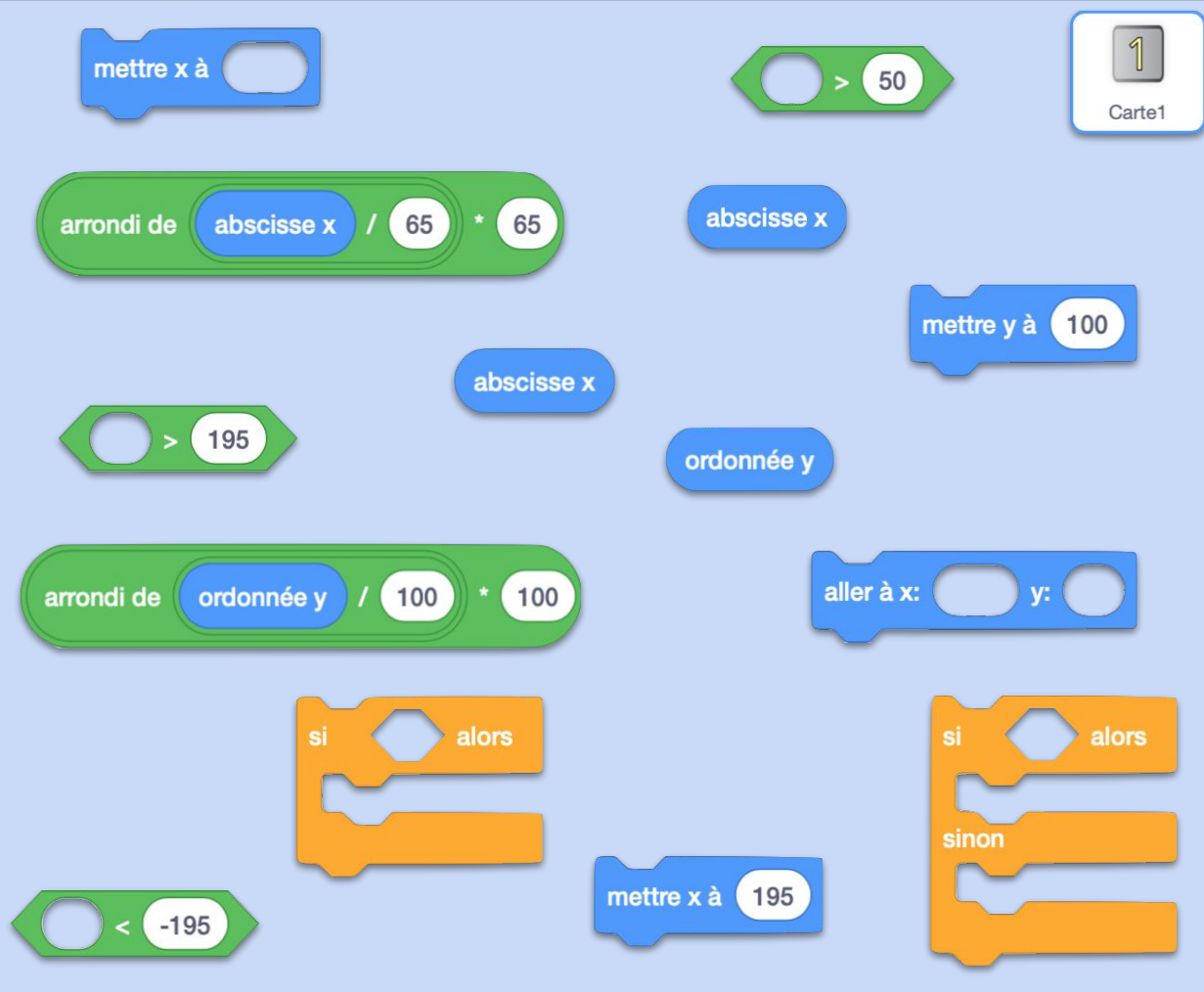
Quand on lâche la carte elle se positionne dans une colonne arrondie à 65 pixels près : de -195 à 195 (voir fiche ORDO2). Une division, un arrondi et une multiplication permettent cela. On force également la ligne à être arrondie à la centaine près de la même manière. On va ensuite vérifier que la colonne n'est pas plus petite que -195 ou plus grande que 195 pour ne pas déborder du décor et remettre la carte dans le cadre si besoin.

On va vérifier aussi que la ligne est plus grande ou plus petite que 50 mais nous finirons de remplir le «si...alors...sinon...» à la fiche suivante.

Mouvement

Contrôle

Opérateurs



The image shows a Scratch script on a light blue background. The script starts with a 'Carte1' block in the top right corner. The main script consists of several blocks:

- A blue 'mettre x à' block with a small empty input field.
- A green 'arrondi de' block containing a blue 'abscisse x' block, a green '/' block with '65', and a green '*' block with '65'.
- A green 'arrondi de' block containing a blue 'ordonnée y' block, a green '/' block with '100', and a green '*' block with '100'.
- A green 'if' block with a green '>' block and '195'.
- A green 'if' block with a green '<' block and '-195'.
- A blue 'mettre x à' block with '195'.
- A blue 'mettre y à' block with '100'.
- A blue 'abscisse x' block.
- A blue 'ordonnée y' block.
- A blue 'aller à x:' and 'y:' block with empty input fields.
- Two orange 'if-then-else' blocks. The first one has a blue 'abscisse x' block in the 'if' section. The second one has a blue 'ordonnée y' block in the 'if' section.
- A green '>' block with '50'.

Nous finirons de remplir ce script sur la prochaine fiche :



Laissez vide pour l'instant

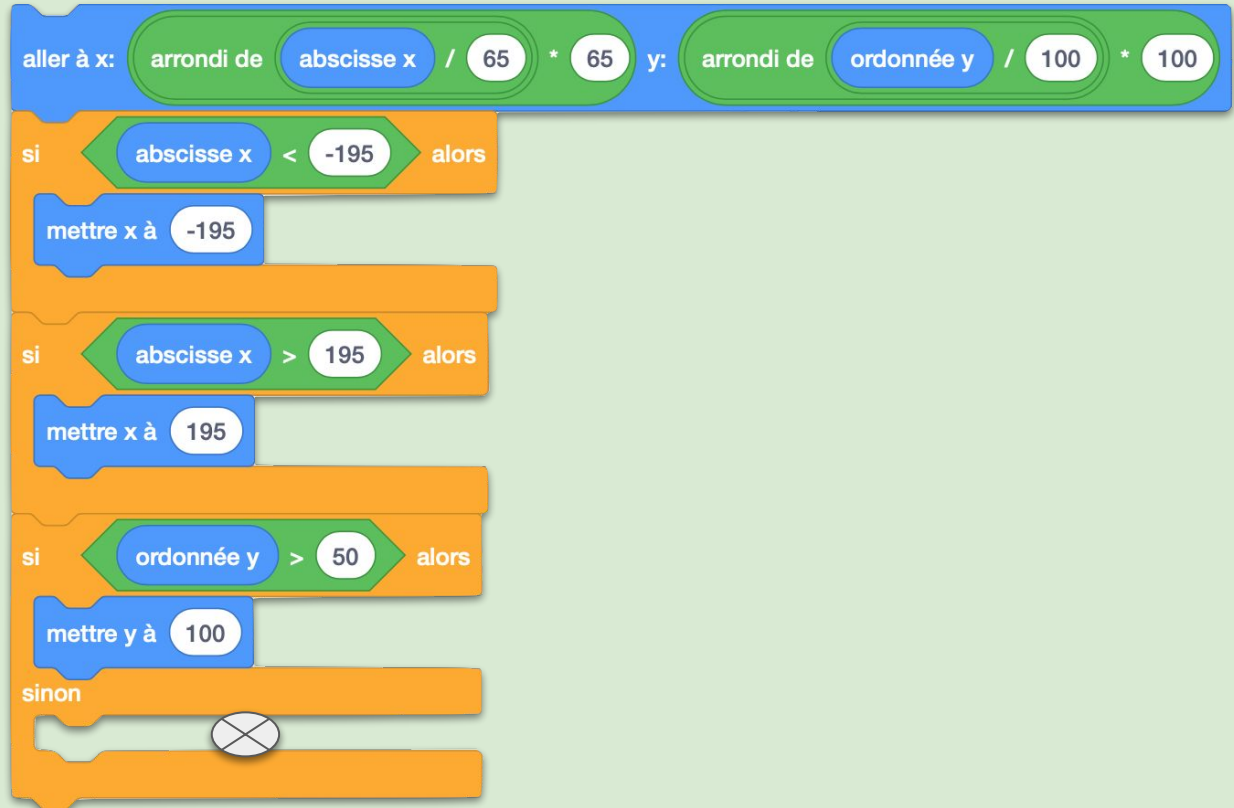
Mouvement

Contrôle


Opérateurs

1

Carte1



Objectifs

Remplissons l'espace  : nous sommes dans le cas où une carte est relâchée en $y < 50$. Nous devons donc d'abord vérifier qu'il y a une case de libre.

Si la case est libre (on trouve 0 dans *liste posée*), on la pose en $y=0$ et la *liste posée* est mise à jour. On lance alors le message "vérifie" pour contrôler si toutes les cartes ont été bien posées (voir fiche suivante).

Si ce n'est pas le cas, un son d'alerte et la carte remonte dans à $y=100$.

Mouvement

Son

Événements

Contrôle

Opérateurs

Variables

remplacer l'élément de la liste liste posée ▼ par monNuméro

1

Carte1

élément de liste posée ▼

envoyer à tous vérifie ▼

arrondi de abscisse x / 65 + 4

si  alors

jouer le son Slide Whistle ▼

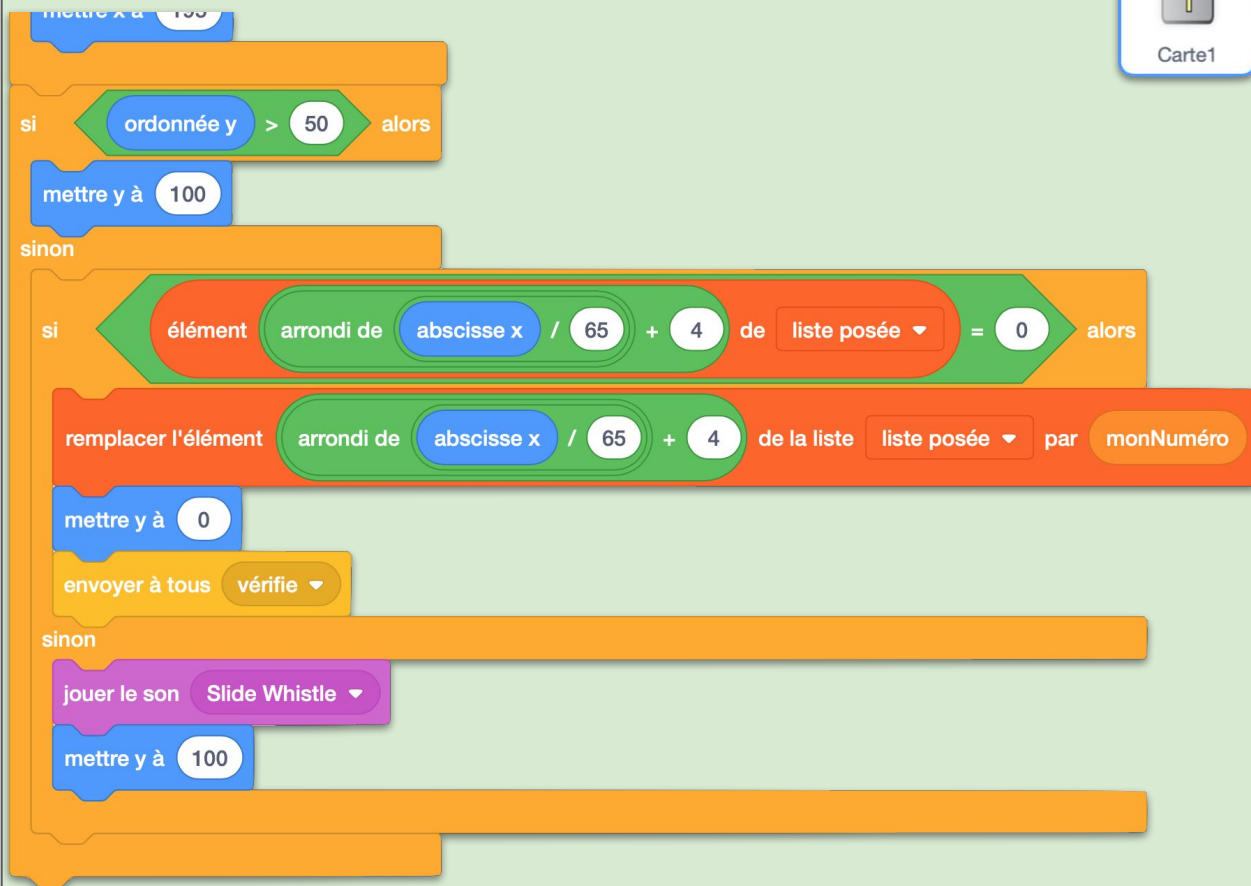
sinon

mettre y à 100

arrondi de abscisse x / 65 + 4

mettre y à 0

On remarquera le calcul inverse de la Fiche 2 : trouver un entier en fonction de l'*abscisse* x au lieu de trouver l'*abscisse* x en fonction d'un entier.



Objectifs

Quand le lutin reçoit le message “vérifie”, il vérifie si la liste posée est la même que la liste correcte. Si c’est le cas, il donne ses félicitations sous les hurras du public en délire. Nous allons comparer chaque élément de ces listes deux à deux. Une nouvelle variable *gagné* est initialisée à 1 et va passer définitivement à 0 si deux éléments placés au même rang des listes ne sont pas égaux (par la multiplication). Dans d’autres langages, on utiliserait des véritables variables booléennes. La variable *i* sert de compteur de rang dans la boucle. *gagné* et *i* sont des variables locales.

Apparence

Son

Événements

Contrôle

Opérateurs

Variables

jouer le son Goal Cheer ▼
 mettre gagné ▼ à
 Lutin
 longueur de liste correcte ▼
 quand je reçois vérifie ▼
 mettre gagné ▼ à 1
 si alors gagné = 1 répéter fois
 mettre i ▼ à 1
 dire Bravo ! C'est le bon ordre pendant 2 secondes
 gagné * élément i de liste posée ▼ = élément i de liste correcte ▼

L'opérateur "=" pour vérifier si deux listes sont identiques n'est pas fiable dans Scratch.
L'astuce non intuitive c'est qu'on peut mettre un bloc booléen (hexagonal) dans une case de variable (ronde). Si un booléen est vrai ("true" en texte), il est considéré comme 1 pour un calcul et comme 0 s'il est faux ("false" en texte).

Pour terminer cette activité, il ne vous reste plus maintenant qu'à dupliquer le lutin Carte 1 six fois et à bien changer la variable *monNuméro* de chacun.



Lutin

```
when received verify
  set gagné to 1
  set i to 1
  repeat (length of list correct) times
    set gagné to gagné * (element i of list correct = element i of list posed)
    add 1 to i
  loop
  if gagné = 1 then
    play sound Goal Cheer
    say Bravo ! C'est le bon ordre for 2 seconds
```

Apparence

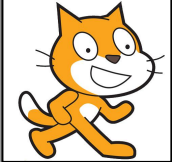
Son

Événements

Contrôle

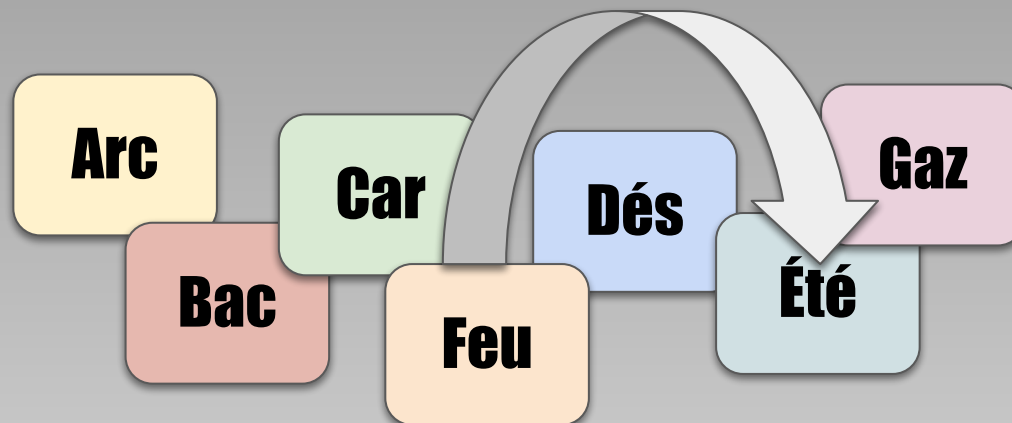
Opérateurs

Variables



scratch.mit.edu

<https://www.reseau-canope.fr>



Pedag O'Scratch

