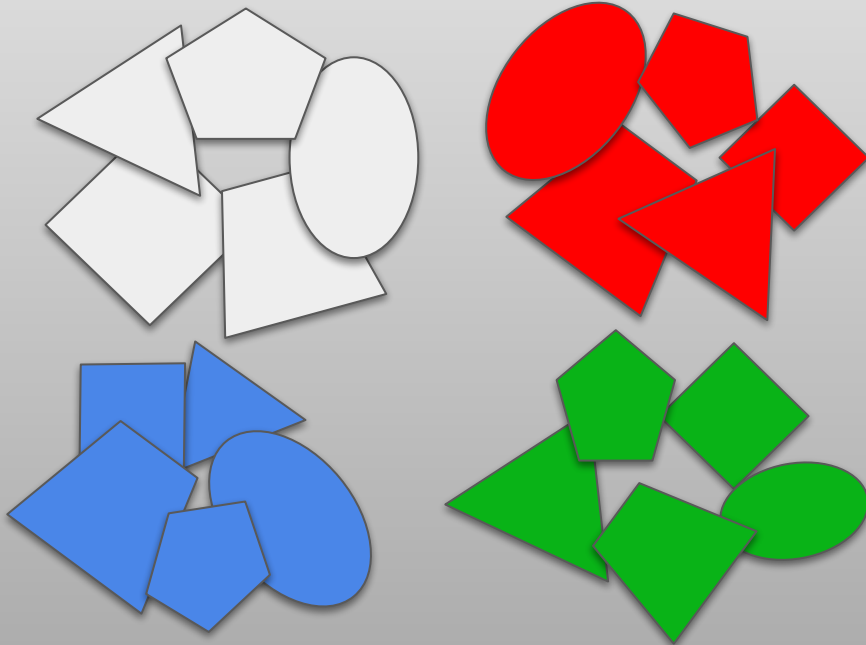


[scratch.mit.edu](https://scratch.mit.edu)



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

# Classement

Mettre les cartes dans la bonne zone

Septembre 2019

*Pedag O'Scratch*





## CLAS01

### Objectifs

Au début du jeu, les cartes doivent se placer en bas au centre de l'écran.

Elles doivent montrer leur costume "jeu".

Créez 2 variables "pour ce sprite uniquement" : *maColonne* et *maCatégorie*.

Initialisez *maColonne* à 0 et *maCatégorie* à 4.

*maColonne* correspond à la colonne dans laquelle la carte se trouve.

*maCatégorie* correspond à la colonne dans laquelle doit être classée la carte

Mouvement

Apparence

Évènements

Variables

mettre maCatégorie à 4

quand est cliqué

aller à x: 0 y: -150

basculer sur le costume jeu

mettre maColonne à 0

porte

porte

Rien de particulier à signaler, l'ordre des blocs après le bloc Drapeau n'est pas important. Toutes les cartes démarreront au même endroit (comme si elles étaient les uns sur les autres). Le bloc "mettre maCatégorie à 4" sera à régler pour chaque carte.

Mouvement

Apparence

Évènements

Variables



## CLAS02

### Objectifs

Lorsque que le joueur clique sur une carte, celle-ci doit se “soulever” (augmenter de taille) et suivre le curseur de la souris jusqu'à ce que le joueur relâche le clic de la souris. Alors elle se “pose” (elle reprend sa taille initiale).

Mouvement

Apparence

Évènements

Contrôle

Capteurs

Opérateurs

ajouter -10 à la taille

porte

porte

quand ce sprite est cliqué

souris pressée ?

ajouter 10 à la taille

répéter jusqu'à ce que

non

aller à pointeur de souris ▼



On répète l'instruction "aller à pointeur de souris" jusqu'à ce que la souris ne soit pas pressée. On verra dans les activités suivantes qu'on peut encore ajouter des effets sonores pour améliorer l'ergonomie et la sensation de soulèvement d'une carte.



quand ce sprite est cliqué

ajouter 10 à la taille

répéter jusqu'à ce que  non  souris pressée ?

aller à pointeur de souris ▼

ajouter -10 à la taille

Mouvement

Apparence

Événements

Contrôle

Capteurs

Opérateurs

## CLAS03

### Objectifs

À la suite du script précédent, vous ajouterez les blocs ci-présents.

Il s'agit d'abord de vérifier si la carte a été posée sur une zone incorrecte. C'est à dire y trop haut (supérieur à 115), y trop bas (inférieur à -80) et x trop à gauche (inférieur à -239).

Si c'est le cas, on la remet à sa place initiale et on met *maColonne* à 0.

Sinon, on calcule *maColonne* par rapport à sa position horizontale et on la recentre bien au milieu dans la colonne.

Mouvement

Contrôle

Opérateurs

Variables

The image shows a Scratch script on a light blue background. In the top right corner, there is a 'porte' (door) block. The script consists of the following blocks:

- An orange 'si... alors... sinon' (if... then... else) control block.
- Inside the 'alors' branch, a blue 'mettre x à' (set x to) block with the expression  $120 * \text{maColonne} - 300$ .
- Inside the 'sinon' branch, a green 'ou' (or) block containing three conditions: 'ordonnée y < -80', 'abscisse x < -239', and 'ordonnée y > 115'.
- Following the 'ou' block, a green 'plafond de' (floor of) block with the expression  $\text{abscisse x} + 240 / 120$ .
- A blue 'aller à x: 0 y: -150' (go to x: 0 y: -150) block.
- Two orange 'mettre maColonne à' (set maColonne to) blocks. The first has an empty input field, and the second has the value '0'.

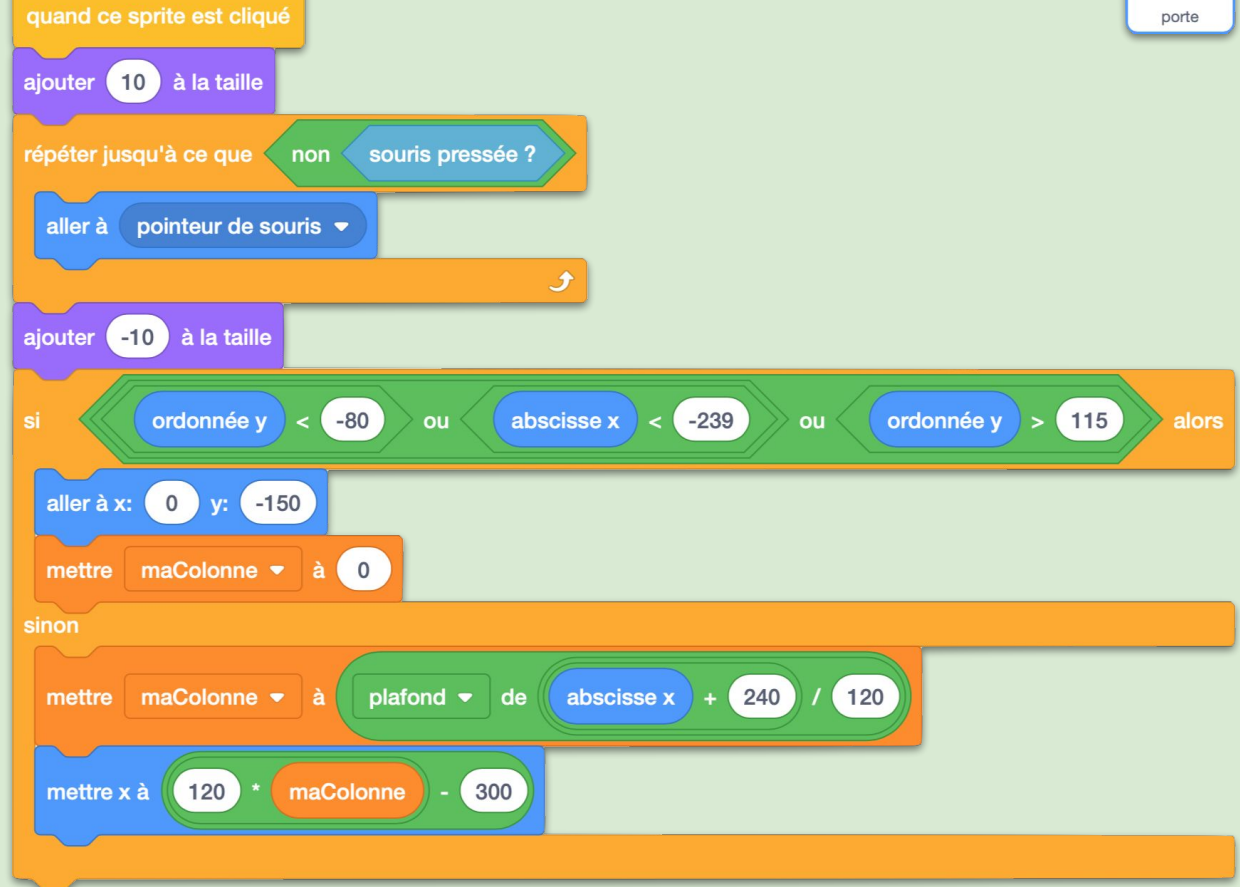
La traduction en langage naturel : on répète l'instruction "aller à pointeur de souris" jusqu'à ce que la souris ne soit pas pressée. On verra dans les activités suivantes qu'on peut encore ajouter des effets sonores pour améliorer l'ergonomie et la sensation de soulèvement d'une carte.

Mouvement

Contrôle

Opérateurs

Variables





Objectifs

D'abord, le sprite Scratch, qui sera le contrôleur du jeu, doit initialiser le nombre d'étiquettes à au moins 1 (suivant le nombre de cartes que vous allez utiliser dans le jeu final) et le score à 0.

L'autre script sert à vérifier si les cartes sont bien placées : quand le joueur clique sur Scratch, on remet *score* à 0. Le script envoie ensuite le message *montreRéussite* et attend les "réponses". Si *score* est égal à *nbÉtiquettes*, la partie est alors gagnée.



mettre nbÉtiquettes à 4

mettre Score à 0

quand est cliqué

dire Bravo, tout est bien classé ! pendant 2 secondes

si alors

quand ce sprite est cliqué

nbÉtiquettes

mettre Score à 0

Score

=

envoyer à tous montreRéussite et attendre

Mouvement

Apparence

Son

Événements

Contrôle

Capteurs

Opérateurs

Variables

Le passage de *score* à 0 est invisible car le calcul est trop rapide pour l'humain.

Il faut bien utiliser "envoyer à tous ... et attendre" sinon l'instruction "si" risque de s'exécuter avant que toute les cartes aient "répondu" au message *montreRéussite*, ce qui fausserait le résultat.



```
quand [drapeau] est cliqué
mettre [Score] à 0
mettre [nbÉtiquettes] à 4
```

```
quand ce sprite est cliqué
mettre [Score] à 0
envoyer à tous [montreRéussite] et attendre
si [Score = nbÉtiquettes] alors
  dire [Bravo, tout est bien classé !] pendant 2 secondes
```

Mouvement

Apparence

Son

Évènements

Contrôle

Capteurs

Opérateurs

Variables

**Objectifs**

Les sprites des cartes ont 3 costumes :

- *jeu* qui est l'apparence normale
- *erreur* pour montrer que la carte est mal placée
- *ok* pour montrer que la carte est placée dans la bonne colonne.

Quand le message *montreRéussite* est reçu, la carte vérifie si *maCatégorie* est égal à *maColonne*. Si c'est le cas elle passe en costume *ok* et ajoute 1 au score, sinon elle passe juste en costume *erreur*. Dans les deux cas, on attend 2 secondes pour que le changement de costume puisse être vu par l'être humain.

Évènements	Apparence
Opérateurs	Contrôle
	Variables

The Scratch script consists of the following blocks:

- Event:** quand je reçois (when I receive) - montreRéussite (montreRéussite)
- Conditional:** si (if) - alors (then) - sinon (else)
- Comparison:** maCatégorie (maCatégorie) = maColonne (maColonne)
- Action 1 (if true):** basculer sur le costume (switch to costume) - ok (ok)
- Action 2 (if true):** ajouter (add) 1 (1) à (to) Score (Score)
- Action 3 (if false):** basculer sur le costume (switch to costume) - erreur (erreur)
- Final Action:** attendre (wait) 2 (2) secondes (seconds)

Toutes les cartes bien placées ajoutent 1 à score quasiment en même temps. Cela peut sembler naturel, mais en informatique c'est parfois dangereux d'agir sur une même ressource en même temps. Dans notre cas, tout se passera bien.

Apparence

Évènements

Contrôle

Opérateurs

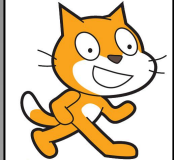
Variables



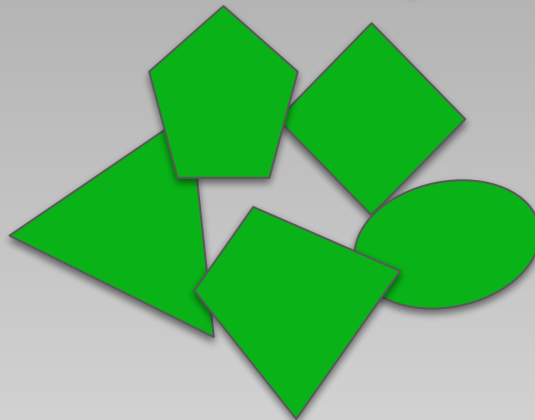
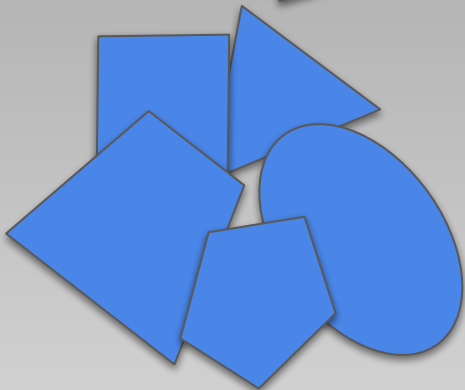
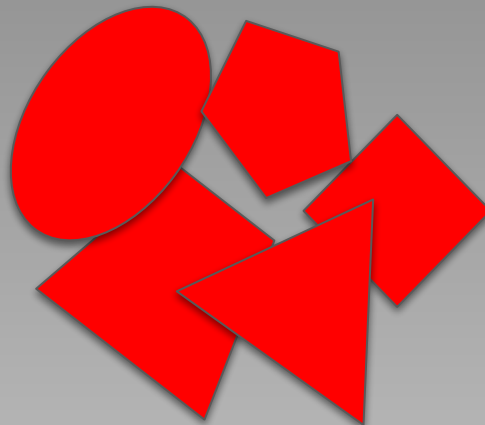
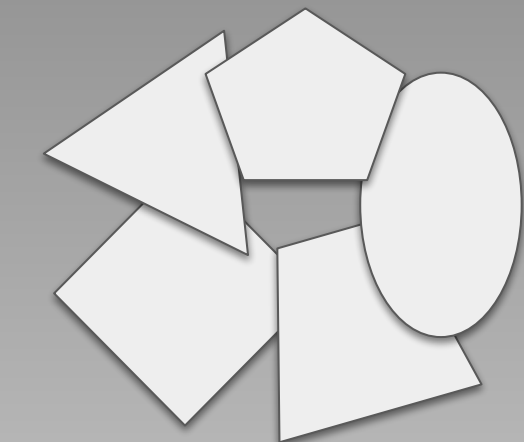
porte

porte





[scratch.mit.edu](https://scratch.mit.edu)



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

*Pedag O'Scratch*

