

scratch.mit.edu



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

Place aux bons endroits

Replacer des éléments sur une image

Mars 2019

Pedag O'Scratch



Objectifs


Commençons par la cible. Elle sera légèrement transparente pour apercevoir le décor derrière (effet fantôme) et l'on réduit sa taille. Elle fera une seule chose indéfiniment : s'agrandir quand la souris passe dessus et se remettre à sa taille normale quand la souris n'est plus dessus. C'est une fonction d'ergonomie pour mieux les repérer lorsque l'on va poser les cartes dessus.

Évènements

Apparence

Contrôle

Capteurs

quand  est cliquési  alors

sinon

répéter indéfiniment mettre la taille à % de la taille initialemettre l'effet à mettre la taille à % de la taille initialemettre la taille à % de la taille initialetouche le ?

Cible1

Le “répéter indéfiniment” est indispensable parce que sinon on ne fait le test qu’une seule fois, alors qu’on veut qu’il le fasse tout le temps du jeu.

Évènements

Apparence

Contrôle

Capteurs

quand  est cliquémettre l'effet **fantôme** à **30**mettre la taille à **40** % de la taille initiale

répéter indéfiniment

si **touche le** **pointeur de souris** ? alorsmettre la taille à **80** % de la taille initiale

sinon

mettre la taille à **40** % de la taille initiale

Cible1

Objectifs

Chaque carte sera désignée par un numéro qui lui sera propre (variable "pour ce lutin uniquement"). Les cibles seront nommées en utilisant ce numéro. La première étape consiste à placer les cartes au début du jeu. Ici en colonne à droite de l'image dans une zone sans cible. On réduit sa taille, on annule tous les effets graphiques que nous allons lui appliquer dans les fiches suivantes. Son emplacement (x,y) est calculé en fonction de son numéro, en haut à droite donc.

Mouvement

Apparence

Évènements

Opérateurs

Variables

1

Carte1

annuler les effets graphiques

quand  est cliquémettre la taille à % de la taille initiale

monNuméro

4 - * mettre à aller à x: y:

Ce lutin ira donc se placer en $y=(4-1) \times 50=150$ et le lutin suivant enlèvera 50 donc en 100 jusqu'au 7e qui sera en $y=(4-7) \times 50=-150$. Ils seront donc répartis uniformément dans la colonne des $x=200$.

quand  est cliqué

annuler les effets graphiques

mettre la taille à % de la taille initiale

mettre à

aller à x: y: - *

Mouvement

Apparence

Évènements

Opérateurs

Variables

Objectifs

Préparons maintenant l'arbitre du jeu : le chat Scratch.

Pour savoir si les réponses sont bonnes, nous allons utiliser une liste. *Liste ok* contiendra tous les numéros des cartes qui sont bien placées. Elle sera donc vide au début et à chaque placement de carte, si c'est le bon emplacement, on ajoutera le numéro de la carte à la liste. Si la liste contient autant de nombres que de cartes, c'est qu'elles sont toutes bien placées et la partie est gagnée.

Évènements	Apparence	Son
Opérateurs	Contrôle	
	Variables	

The Scratch script editor contains the following blocks:

- répéter indéfiniment** (orange loop block)
- dire** Tu peux me cliquer pour vérifier. pendant 2 secondes (purple speech bubble block)
- lutin** Lutin (Scratch cat icon)
- jouer le son** Goal Cheer jusqu'au bout (purple sound block)
- quand** est cliqué (yellow event block)
- si** alors (orange conditional block)
- stop** tout (orange stop block)
- dire** Place les éléments aux bons endroits. pendant 3 secondes (purple speech bubble block)
- supprimer tous les éléments de la liste** liste ok (orange list block)
- longueur de** liste ok (orange list block)
- =** 7 (green operator block)
- dire** Bravo ! pendant 2 secondes (purple speech bubble block)

Vous pourrez remarquer la forme particulière de l’instruction “stop tout”. En effet, comme le programme s’arrête complètement, il ne sert à rien de rajouter une instruction à sa suite.

Évènements

Apparence

Contrôle

Son

Opérateurs

Variables



Lutin



Objectifs

Voyons ce qu'il doit se passer quand on clique sur une carte pour la déplacer :
 comme on la soulève, on enlève son numéro de *liste ok* s'il s'y trouve. On joue un son qui indique que la carte est bien attrapée. On ajoute un effet fantôme, on augmente sa taille puis on la place au premier plan, devant tous les autres lutins. La carte va ensuite suivre le curseur de la souris jusqu'à que son bouton ne soit plus cliqué. On joue alors un son de pose et on remet son apparence normale.

Mouvement	Apparence	Son
Événements	Contrôle	Capteurs
Opérateurs	Variables	

jouer le son Water Drop ▼

jouer le son pop ▼



quand ce sprite est cliqué

ajouter 20 à la taille

supprimer l'élément élément # de monNuméro dans liste ok ▼ de liste ok ▼

répéter jusqu'à ce que

souris pressée ?

non

ajouter 50 à l'effet fantôme ▼

ajouter -20 à la taille

annuler les effets graphiques

aller à l'avant ▼ plan

aller à pointeur de souris ▼

À noter : l'instruction "élément # de X dans liste" donne le numéro de la place où X apparaît dans la liste.

Si on a liste=(4,2,7,8,3) alors

"élément # de 7 dans liste" = 3 (7 est le 3e élément de la liste). Si l'

élément ne se trouve pas dans la liste, alors on obtient 0 : "élément # de 9 dans liste" = 0.

L'instruction "supprimer l'élément Y de liste" supprime le Y-ième élément de la liste. "supprimer l'élément 3 de liste" donne : (4,2,8,3). Si Y=0 (l'élément ne se trouve pas dans la liste dans notre cas) alors on ne supprime rien, ce qui nous convient parfaitement.

Mouvement

Apparence

Son

Événements

Contrôle

Capteurs

Opérateurs

Variables

1

Carte1

quand ce sprite est cliqué

supprimer l'élément élément # de monNuméro dans liste ok de liste ok

jouer le son pop

ajouter 50 à l'effet fantôme

ajouter 20 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 -20 à la taille

annuler les effets graphiques

Objectifs

Une fois la carte posée, nous allons vérifier si elle touche une cible. Hélas, pas de façon plus facile que de vérifier si elle touche les cibles, une par une. Qu'à cela ne tienne, faisons une boucle avec une petite variable locale *i* qui prendra successivement les valeurs de 1 à 7. Même mécanisme que pour l'activité appariement. Si la carte touche bien une cible, on la replace exactement au même endroit que cette cible. Si encore, la cible a le même numéro que la carte, on ajoute alors ce numéro à *liste ok*. Cela veut dire que cette carte est correctement placée. Même si ce n'est pas le cas, on arrête alors le script. En dernier, on envoie le message "vérifie réussite" que nous expliquerons plus tard.

Mouvement

Contrôle

Capteurs

Opérateurs

Variables

The image shows a Scratch script designed to verify the placement of a card. The script is composed of several blocks:

- Carte1**: A variable block containing the number 1.
- aller à**: A "go to" block with a blank field.
- regrouper Cible et i**: A "group together" block with "Cible" and "i" selected.
- i = monNuméro**: An assignment block setting the variable *i* to the value of *monNuméro*.
- regarde le touche le ?**: A "look at" block with a question mark, intended to check if the card touches the target.
- si alors**: A conditional block with a blank "if" field and a blank "then" field.
- stop ce script**: A "stop" block to end the script.
- ajouter 1 à i**: An "add" block adding 1 to the variable *i*.
- ajouter monNuméro à liste ok**: An "add" block adding the value of *monNuméro* to the *liste ok* variable.
- répéter 7 fois**: A "repeat" block set to 7 iterations.
- si alors**: A second conditional block with a blank "if" field and a blank "then" field.
- mettre i à 0**: A "set" block setting the variable *i* to 0.

À signaler qu'il faudra éviter dans la conception de l'activité de mettre des cibles trop proches les unes des autres. À vous d'imaginer ce qu'il peut se passer si une carte touche plusieurs cibles en même temps. Est-ce normal ? Est-ce voulu ? Cela peut-il provoquer un bug ? Ce bug est-il bénin ou bloquant pour l'utilisateur du programme ? À vous de tester cela à la fin des fiches.

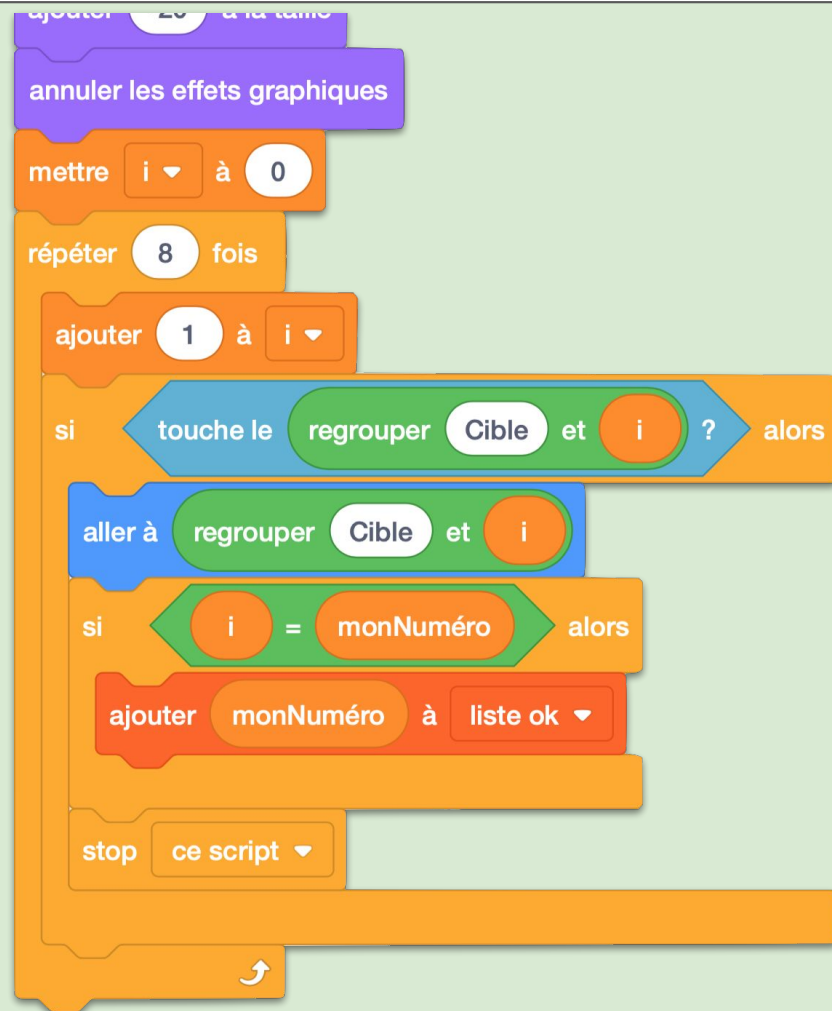
Mouvement

Contrôle

Capteurs

Opérateurs

Variables



PABE6

Objectifs

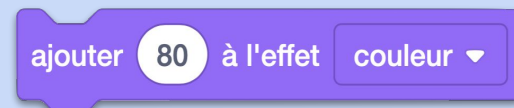
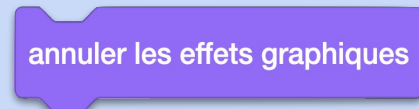
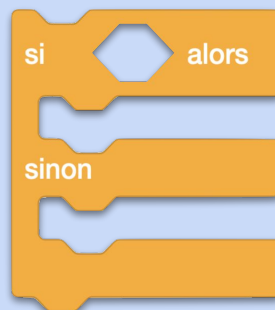
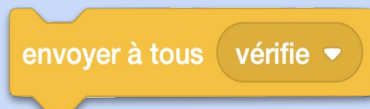
Lorsque toutes les cartes seront posées, si elles sont toutes correctement posées, la partie se termine (la *liste ok* est composée de 7 nombres). Si ce n'est pas le cas, le joueur peut cliquer sur le chat pour savoir quelles cartes sont bien placées. Il envoie un message "vérifie". Les cartes à réception de ce message, vérifient si leur numéro fait partie de *liste ok*. Si c'est le cas, leur couleur passe du bleu au vert, sinon au rouge, pendant 1 seconde.

Évènements

Apparence

Contrôle

Variables



Votre première carte et votre première cible sont prêtes, il suffit maintenant de les dupliquer et de changer pour chacune l'instruction qui leur attribue un numéro (voir fiche PABE2). Personnalisez bien entendu les dessins sur les cartes, trouvez le décor qui convient à votre activité et placez les cibles aux bons endroits.

Évènements

Apparence

Contrôle

Variables



Lutin

1

Carte1

quand ce sprite est cliqué

envoyer à tous vérifie ▼

quand je reçois vérifie ▼

si liste ok ▼ contient monNuméro ? alors

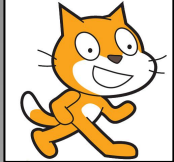
ajouter -50 à l'effet couleur ▼

sinon

ajouter 80 à l'effet couleur ▼

attendre 1 secondes

annuler les effets graphiques



scratch.mit.edu



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

Pedag O'Scratch

