

scratch.mit.edu



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

Memory

Retourner des paires de cartes identiques

Mars 2019

Pedag O'Scratch



Objectifs

Tout d'abord Scratch le chat va expliquer le jeu au milieu de la scène c'est à dire en coordonnées (0,0), tourné vers la droite (direction 90) puis se déplacer sur le chemin avec un angle de direction 80. Ensuite sa taille va diminuer progressivement en avançant et en alternant les costumes, donnant l'illusion d'éloignement.

Enfin, il se retourne vers la gauche (direction -90) pour annoncer les cartes.

Mouvement

Apparence

Évènements

Contrôle

The image shows a Scratch script for a memory game introduction. The script is composed of several blocks:

- Initial Setup:**
 - Mouvement:** s'orienter en direction de 80
 - Mouvement:** avancer de 17 pas
 - Apparence:** costume suivant
 - Apparence:** ajouter -5 à la taille
 - Mouvement:** aller à x: 0 y: 0
- Dialogue 1:**
 - Apparence:** dire Cliquez sur une carte pour la retourner puis sur une deuxième. pendant 4 secondes
- Dialogue 2:**
 - Apparence:** dire Je vous propose un petit jeu de memory. pendant 4 secondes
 - Apparence:** dire Bonjour! pendant 2 secondes
- Dialogue 3:**
 - Mouvement:** s'orienter en direction de 90
 - Apparence:** dire Je place les cartes. pendant 2 secondes
- Dialogue 4:**
 - Apparence:** dire Si les deux cartes correspondent alors elles resteront de ce côté. pendant 4 secondes
- Final Setup:**
 - Mouvement:** s'orienter en direction de -90
 - Apparence:** dire Faisons de la place...
 - Apparence:** attendre 0.2 secondes
- Control:**
 - Apparence:** répéter 10 fois (loop)
 - Apparence:** quand est cliqué (event)

The script is set to repeat 10 times, triggered by a click event.

Que se passe-t-il si l'on omet l'instruction "Attendre 0.2 secondes" dans la boucle ?

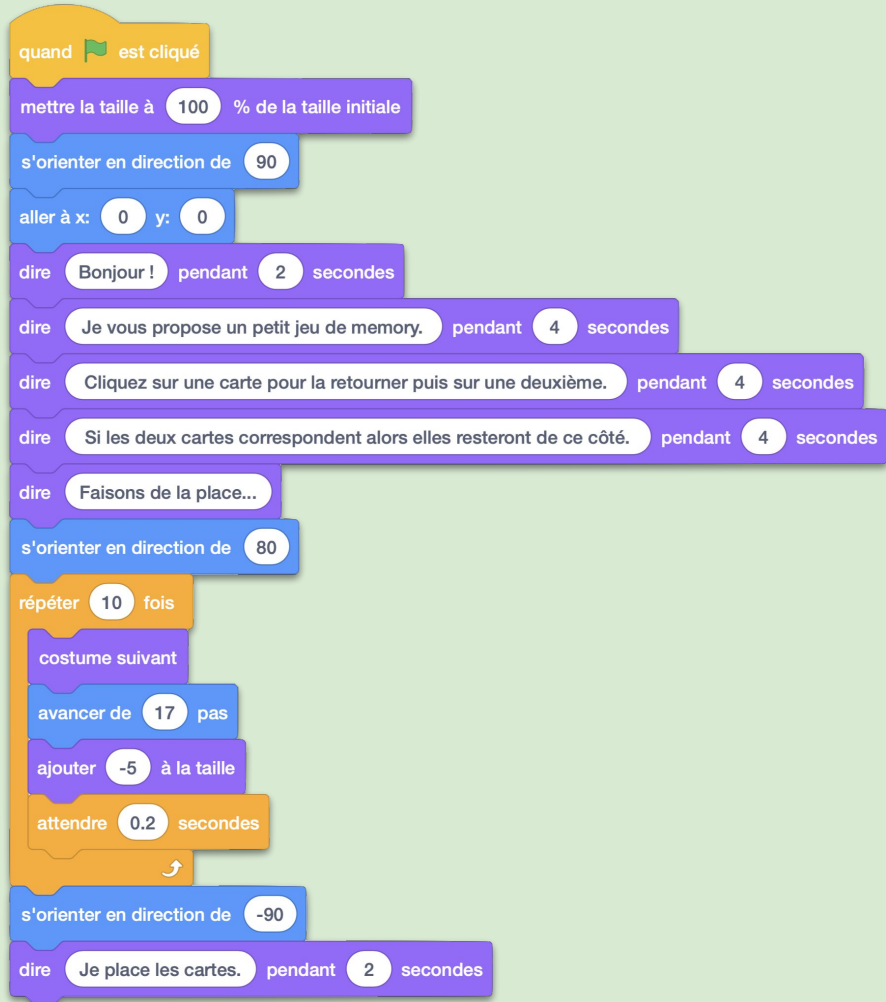
On peut augmenter le nombre de boucles pour rendre le mouvement moins saccadés : c'est le principe du dessin animé, plus le nombre d'images par secondes est grand, plus l'animation est fluide. Il faudrait alors réduire les paramètres à l'intérieur pour que chaque étape corresponde à un changement plus petit.

Mouvement

Apparence

Évènements

Contrôle



Scratch le ...

```
quand le drapeau est cliqué
mettre la taille à 100 % de la taille initiale
s'orienter en direction de 90
aller à x: 0 y: 0
dire Bonjour! pendant 2 secondes
dire Je vous propose un petit jeu de memory. pendant 4 secondes
dire Cliquez sur une carte pour la retourner puis sur une deuxième. pendant 4 secondes
dire Si les deux cartes correspondent alors elles resteront de ce côté. pendant 4 secondes
dire Faisons de la place...
s'orienter en direction de 80
répéter 10 fois
  costume suivant
  avancer de 17 pas
  ajouter -5 à la taille
  attendre 0.2 secondes
s'orienter en direction de -90
dire Je place les cartes. pendant 2 secondes
```

MEMO2

Objectifs

L'autre script du chat qui démarre au drapeau vert va initialiser les variables nécessaires à l'arbitrage du jeu.

D'abord, une variable *nbPaires* qui va servir à suivre le nombre de paires correctes découvertes par le joueur.

Ensuite *carteRetournée* qui contient le nom de la carte qui vient d'être retournée. Cette variable est soit vide, soit contient le nom d'une carte (c'est du texte).

Enfin *attendre*, il s'agit de la variable qui va servir à bloquer les clics sur les autres cartes en attendant qu'on gère les clics précédents. Elle contiendra soit "faux" soit "vrai".

Le script attend ensuite que le nombre de paires trouvées corresponde à une partie gagnée pour féliciter le joueur.

Quel est ce nombre pour 10 cartes ?

Événements

Opérateurs

Apparence

Contrôle

Variables

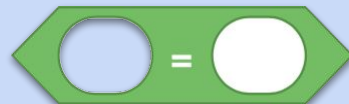
Son

mettre

attendre ▼

à

faux



attendre jusqu'à ce que

jouer le son

Cheer ▼

dire

Bravo, c'est gagné !

pendant

5

secondes

quand



est cliqué

mettre

carteRetournée ▼

à



mettre

nbPaires ▼

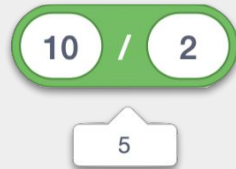
à

0

nbPaires

Attention à bien mettre du vide dans le “mettre *carteRetournée* à”. Même une espace (invisible à votre œil dans cette case blanche) serait un bogue bloquant.

Sinon la bonne réponse était bien entendu :



Événements

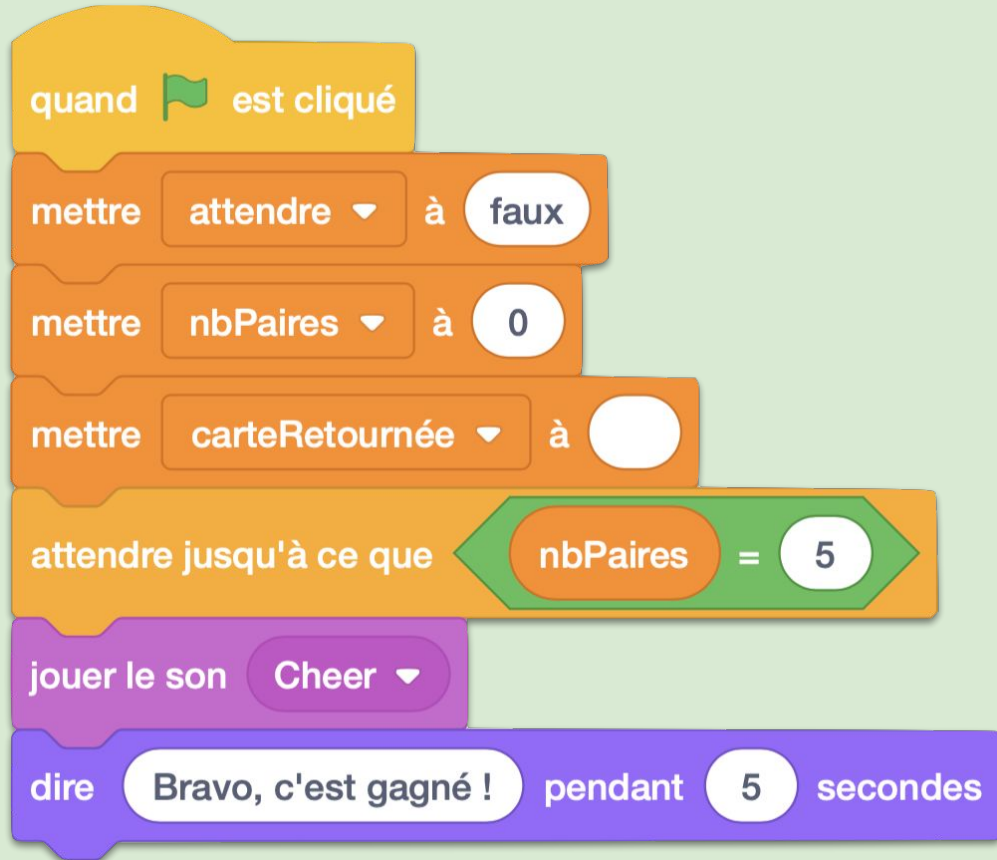
Apparence

Contrôle

Son

Opérateurs

Variables



MEMO3

Objectifs

Pour les cartes, nous allons d'abord initialiser les variables *monNom* et *leNomDeMaPaire*. Chaque carte aura sa propre variable à ces noms, ce sont des variables locales. On initialise leurs tailles et leurs costumes. Enfin, on les cache.

Après les 20 secondes que mettra Scratch le chat à expliquer le jeu, les cartes se placent sur le chat, se montrent et essaient de glisser vers une place sur la scène qui ne soit ni sur une autre carte (couleur noire touchée), ni sur le bord ni sur Scratch le chat. À chaque essai, on entend un petit son de pose de carte.

Mouvement

Apparence

Son

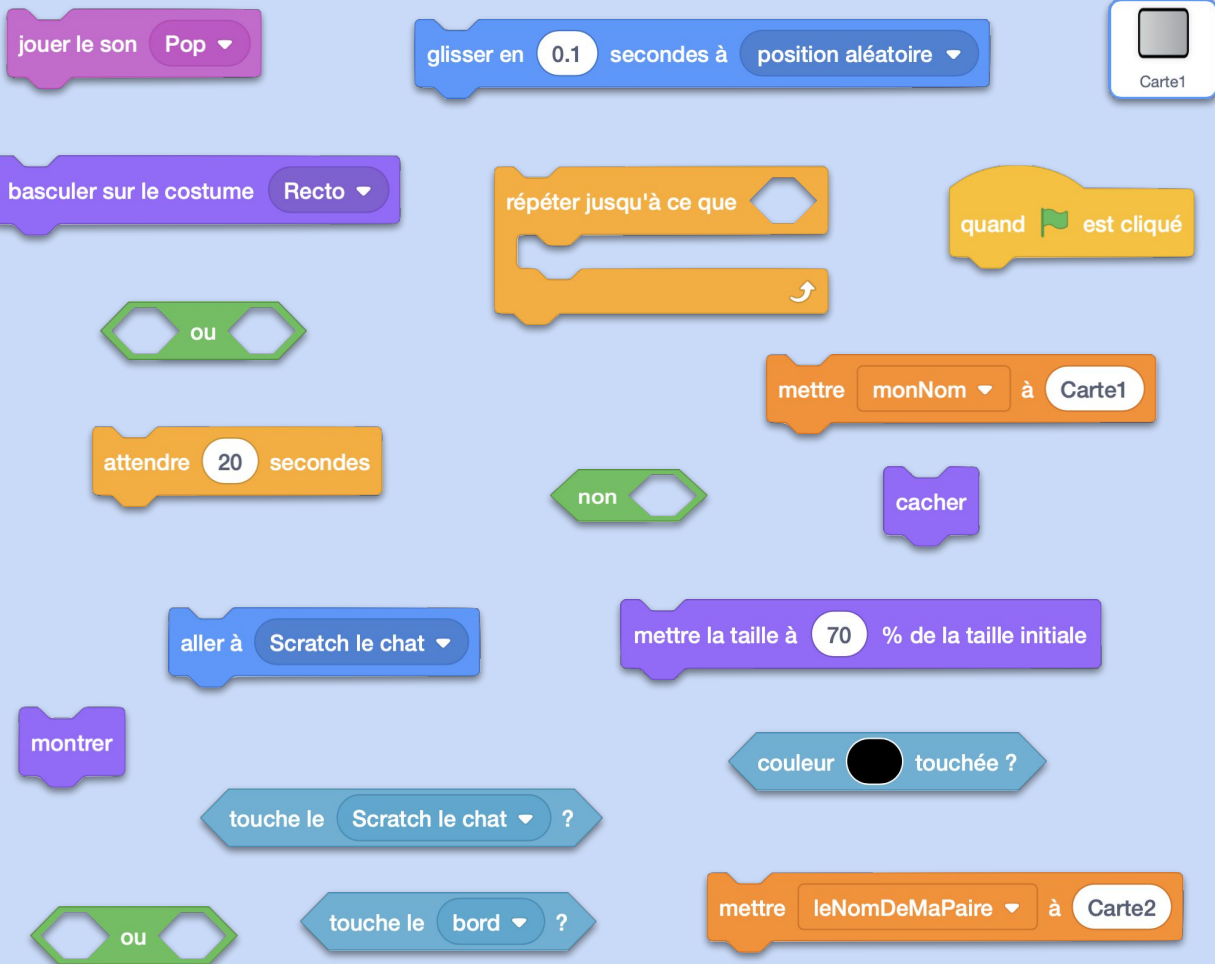
Événements

Contrôle

Capteurs

Opérateurs

Variables



Ici le plus complexe c'est l'imbrication des "ou" et du "non".

La logique booléenne est bien respectée, même si elle ne correspond pas exactement au langage naturel :

- "[non (touche le bord)] et [non (touche Scratch)]"

est équivalent à

- "non [(touche le bord) ou (touche Scratch)]"

De plus :

- "x ou y ou z"

est équivalent à

- "[(x ou y) ou z]"

ou encore à

- "[x ou (y ou z)]"

Mouvement

Apparence

Son

Événements

Contrôle

Capteurs

Opérateurs

Variables



Carte1

```

quand [drapeau vert] est cliqué
mettre monNom à Carte1
mettre leNomDeMaPaire à Carte2
mettre la taille à 70 % de la taille initiale
basculer sur le costume Recto
cacher
attendre 20 secondes
aller à Scratch le chat
montrer

répéter jusqu'à ce que non couleur [noir] touchée ? ou touche le bord ? ou touche le Scratch le chat ?
jouer le son Pop
glisser en 0.1 secondes à position aléatoire
  
```

MEMO4

Objectifs

On attaque la partie la plus complexe d'un point de vue algorithmique : 3 tests imbriqués les uns dans les autres. Du coup, on vous donne des ensembles de blocs déjà tout prêts.

Tout cela doit se dérouler quand on clique sur une carte. D'abord, on vérifie qu'elle est du côté verso, et qu'on attend pas la fin d'un clic sur une autre carte.

Si c'est le cas (on passe sur costume *Verso*), on va vérifier si une première carte était déjà retournée. Et si c'est le cas, on vérifie si celle-ci forme bien une paire ou non.

L'évènement *check* indique à la première carte de se retourner définitivement (la paire est faite : costume *Verso2*).

L'évènement *cache* demande à toutes les cartes non bloquées en *Verso2* de se cacher (la paire n'est pas faite : elles repassent sur le costume *recto*).

Évènements	Apparence	Son
Opérateurs	Contrôle	
	Variables	

The image shows a Scratch script for a card game. It starts with a 'when clicked' event. The main logic is a nested 'if' structure: 'if (nom du costume = Recto) and (attendre = faux) then...'. Inside this, there are two parallel paths: one for 'check' events (switch to costume Verso2, play Collect sound, set attendre to vrai) and one for 'cache' events (play Bonk sound, switch to costume Verso2, set attendre to faux). A second 'if' block checks 'carteRetournée = leNomDeMaPaire' to determine if a pair is formed, which then triggers 'set carteRetournée to true'.

```
when clicked
  if (nom du costume = Recto) and (attendre = faux) then
    if (check) then
      basculer sur le costume Verso2
      envoyer à tous check
      jouer le son Collect jusqu'au bout
      mettre attendre à vrai
    else
      jouer le son Bonk jusqu'au bout
      envoyer à tous cache
      basculer sur le costume Verso2
      mettre attendre à faux
  if (carteRetournée = leNomDeMaPaire) then
    mettre carteRetournée à true
```


La variable *attendre* se met à “vrai” au début de la gestion du clic, et se remet à “faux” à la fin. C’est ce qu’on appelle un flag (drapeau). C’est ce qu’on est obligé de mettre en place pour éviter que le joueur clique plusieurs fois de suite sur plusieurs cartes avant que le reste du programme ait eu le temps de déterminer si les clics sont corrects ou non. C’est une des difficultés de la programmation parallèle où plusieurs scripts peuvent s’exécuter en même temps.

Évènements

Apparence

Son

Contrôle

Opérateurs

Variables

```

quand ce sprite est cliqué
si (nom du costume = Recto) et (attendre = faux) alors
  mettre attendre à vrai
  basculer sur le costume Verso
  si (carteRetournée = [ ]) alors
    mettre carteRetournée à monNom
  sinon
    si (carteRetournée = leNomDeMaPaire) alors
      basculer sur le costume Verso2
      envoyer à tous check
      jouer le son Collect jusqu'au bout
    sinon
      jouer le son Bonk jusqu'au bout
      envoyer à tous cache
  mettre carteRetournée à [ ]
mettre attendre à faux
  
```



Carte1

MEMO5

Objectifs

Deux petits scripts pour terminer :

- évènement *cache* (mauvaise paire de carte) :

Si la carte est sur Verso, elle se tourne sur Recto

- évènement *check* (bonne paire de carte) :

le nom de la carte est celle de *carteRetournée* alors on la passe sur son costume Verso2. On ajoute 1 à *nbPaires*.

The image shows a Scratch workspace with two scripts. The top script is triggered by the 'cache' event and checks if the card is on the 'Verso' side. If so, it switches the costume to 'Recto'. The bottom script is triggered by the 'check' event and checks if the card name matches 'carteRetournée'. If so, it switches the costume to 'Verso2' and increments the 'nbPaires' variable by 1.

Script 1 (Top):

- Event: *cache*
- Control: **si** / **alors**
- Condition: **nom** du costume **=** Verso
- Action: basculer sur le costume **Recto**

Script 2 (Bottom):

- Event: *check*
- Control: **si** / **alors**
- Condition: **monNom** **=** **carteRetournée**
- Action: basculer sur le costume **Verso2**
- Action: ajouter **1** à **nbPaires**

Legend:

- Évènements (Yellow)
- Opérateurs (Green)
- Apparence (Purple)
- Contrôle (Orange)
- Variables (Blue)

Vous n'avez plus maintenant qu'à dupliquer le lutin Carte1 pour obtenir autant de cartes qu'il vous faut.

N'oubliez pas pour chacune d'entre elles, de leur donner un nom et le nom de la carte avec laquelle elle sera appairée. Changez également le nombre de paires à atteindre si vous n'utilisez pas 10 cartes.

D'ailleurs si vous utilisez plus de cartes, il faudra sans doute que vous réduisiez leurs tailles pour qu'elles puissent toutes se placer dans l'espace de la scène.

Apparence

Évènements

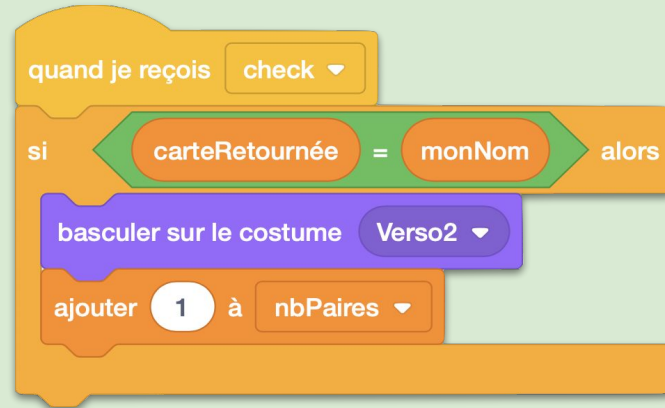
Contrôle

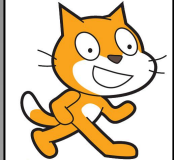
Opérateurs

Variables



Carte1





scratch.mit.edu



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



Pedag O'Scratch

