

# Memory



## Présentation

Ce projet consiste à réaliser un programme où le joueur devra reconstituer par mémorisation les 18 paires d'images affichées dans un carré de 6 x 6.

## Objectifs pédagogiques

Ce projet aura pour but de travailler sur :

- les boucles
- les variables locales et globales
- les structures conditionnelles (si... alors...)
- les listes

## Organisation et évaluation

Vous travaillerez en binôme.

Chaque binôme disposera de trois fichiers documents:

- le présent **guide** explicatif
- un **programme Scratch** de démarrage
- une **fiche d'avancement** à compléter par chaque élève et à chaque séance pour indiquer l'état de progression du travail

L'évaluation de votre travail tiendra compte :

- du bon fonctionnement du programme
- des éléments de développement personnalisés que vous aurez ajoutés
- de l'implication de chacun (= évaluation individuelle, pas en binôme)

Les meilleurs projets pourront être publiés sur le site du collège.

A chaque fin de séance, enregistrez votre travail sur :

- la zone de partage de données de la classe
- la zone personnelle de chaque élève du binôme (important en cas de problème sur la zone de partage)

Pour chaque début de séance, il vous appartient d'apporter votre travail :

- soit sur clé USB
- soit par mail
- soit en travaillant directement sur votre zone personnelle au collège



### **Durée**

Ce projet s'étendra sur 4 séances et le travail devra être rendu à l'issue de la dernière séance (programme Scratch + fiche d'avancement complétée).

### **Méthodologie :**

Vous disposez de 37 lutins (le lutin Abby qui guidera le joueur + les 36 cartes) et de la scène.

Chacune des 36 cartes comporte :

- 2 costumes : un costume face cachée (point d'interrogation) et un autre costume face visible
  - 2 propriétés (variables propres à chaque lutin) :
    -  qui désigne son numéro (1, 2, 3, ..., 36)
    -  qui désigne ce qu'elle représente (« un renard », « un fantôme », etc.)
- ⇒ Ces deux propriétés serviront à identifier chaque lutin par la suite.

### **Scénario :**

Au clic sur le drapeau, les cartes sont mélangées et rangées dans un carré de 6 x 6, face cachée.

Abby apparaît et demande au joueur de trouver les paires : le jeu commence.

Le joueur sélectionne une carte (un clic) puis une autre (un clic) : les cartes sont retournées.

Si les cartes correspondent, le joueur marque un point et les cartes restent visibles, autrement elles sont remises face cachées 2 secondes après avoir montré la deuxième carte.

Le jeu s'arrête lorsque toutes les cartes sont retournées.

Le programme recommence du début quelques secondes après avoir annoncé le nombre d'essais.

### **Exemples d'améliorations:**

- Le jeu peut s'arrêter au terme d'un délai fixé au début du jeu : le joueur a perdu.
- Le jeu peut s'arrêter au bout d'un certain nombre d'essais infructueux : le joueur a perdu.
- Le joueur peut choisir le niveau :
  - facile : 16 cartes (4 x 4)
  - difficile : 36 cartes (6 x 6)
- ...

Vous devrez réaliser successivement les objectifs suivants.  
 Votre méthodologie peut différer de celle proposée ;  
 dans ce cas, c'est de la responsabilité du binôme si la méthodologie choisie n'aboutit pas.

### Objectif n°1 : Placement des cartes

#### Principe :

Prenons une liste de nombres entiers classés de 1 à 36. Quand on prend un élément au hasard dans la liste, qu'on le retire de la liste, puis qu'on renouvelle l'opération jusqu'à ce que la liste soit vide, ceci assure que les nombres retirés dans la liste forment une succession au hasard de tous les nombres entiers compris entre 1 et 36.

Nous utiliserons cette méthode pour disposer les cartes de façon aléatoire, dans un carré de 6 x 6.

#### Application :

**Dans la scène**, faire vider la **Liste** puis lui faire ajouter 36 nombres entiers numérotés de 1 à 36.

Utiliser pour cela une variable **nombre** qu'on initialisera à 1 avant de répéter 36 fois l'ajout de **nombre** dans **Liste** puis l'incrémenter de **nombre** (son augmentation de 1).

Après avoir initialisé une variable **Colonne** à 1 et **Ligne** à 1, on va effectuer une autre répétition 36 fois encore.

Dans cette répétition, **Colonne** désignera la colonne (entre 1 et 6) et **Ligne** la ligne (entre 1 et 6) qui nous intéressent pour placer correctement la carte dont on calculera son abscisse et son ordonnée.

En créant une variable **Abscisse** et une variable **Ordonnée** qui désigneront les coordonnées dans la scène de la carte à placer, on utilisera les formules de calcul suivantes en fonction de **Colonne** et **Ligne** :

```

mettre Abscisse à Colonne * 60 - 270
mettre Ordonnée à Ligne * -60 + 210
    
```

Une fois ce calcul fait, on prend un nombre au hasard compris entre 1 et longueur de **Liste** et on le met dans la variable **ligne de la liste**. On mémorise alors dans une variable nommée **Numéro carte à placer** l'élément **ligne de la liste** de **Liste** avant d'**envoyer à tous Carte, place toi ! et attendre**.

**Pour chacun des 36 lutins**, lorsqu'il reçoit le message envoyé par la scène, si le **Numéro carte à placer** est égal à **Carte**, c'est que c'est à son tour de se placer : on lui demande alors de se positionner en **Abscisse** et **Ordonnée** et de se montrer. Autrement, on ignore le message.

On pourra utiliser un effet de glissement rapide pour donner une impression de distribution des cartes.

**Dans la scène**, on supprime ensuite de **Liste** l'élément **ligne de la liste**, puis on passe à la case suivante.

Pour changer de case, on ajoutera 1 à **Colonne** et, si sa valeur dépasse 6, il faudra la remettre à 1 tout en ajoutant 1 à **Ligne**. Ainsi, **Colonne** et **Ligne** prendront toutes les valeurs répertoriées dans le tableau ci-dessous :

Colonne=1 Ligne=1	Colonne =2 Ligne =1	...	...	...	Colonne =6 Ligne =1
Colonne =1 Ligne =2	...	...	...	...	Colonne =6 Ligne =2
Colonne =1 Ligne =3	...	...	...	...	...
Colonne =1 Ligne =4	...	...	...	...	...
Colonne =1 Ligne =5	...	...	...	...	...
Colonne =1 Ligne =6	...	...	...	...	Colonne =6 Ligne =6

Lorsque la répétition est terminée, on envoie un message à Abby pour qu'elle commence à animer le jeu.

### Objectif n°2 : Déroulement du jeu

**Chez Abby**, elle demande durant 1,5 seconde au joueur de découvrir les paires d'images.

Ensuite, on commence une boucle qui se répétera jusqu'à ce que la variable **Score** soit égale à 18 (18 paires de cartes découvertes).

Chaque fois que la boucle sera répétée, Abby va **attendre jusqu'à** ce que le joueur ait retourné deux cartes.

Si les noms des deux cartes sont les mêmes, alors le coup est gagnant (on augmente le score de 1) et les cartes restent visibles. Autrement, par l'envoi d'un message par Abby, les cartes sont retournées face cachée au bout d'un délai de 2 secondes.

**Chez les cartes**, quand le joueur clique sur le lutin, on mémorise **Figure** de la carte retournée dans la variable **Nom carte 1** à la première carte cliquée, et **Nom carte 2** lorsque le joueur retourne la seconde.

Et quand le lutin reçoit le message d'Abby pour se retourner (en cas de coup perdant), il doit se remettre face cachée s'il fait partie des deux cartes qui viennent d'être retournées.

### Objectif 3 : Gestion des bugs

Durant la distribution des cartes ou durant l'attente des 2 secondes d'un coup perdant, le joueur ne devrait pas pouvoir retourner les cartes. Pour éviter cela, on pourra utiliser une variable **Sélection autorisée** qui pourra valoir « oui » ou « non » aux endroits où le clic sera autorisé ou pas,

et mettre chez chaque lutin une condition **si ... alors** pour agir ou ignorer l'action au moment du clic.

Egalement, le clic sur une carte déjà retournée ne devrait pas être possible. Lorsque **costume #** d'un lutin vaut 1 c'est que la carte est face cachée, et s'il vaut 2 sa face est visible.