

Introduction

Nous venons d'étudier le glisser-déplacer d'une occurrence sur la scène. À chaque fois, nous avons effectué un test au moment où le bouton de la souris est relâché pour vérifier la position de l'occurrence par ses coordonnées x et y. Il sera parfois plus simple de vérifier si l'occurrence déplacée touche une autre occurrence.

Une course de coccinelles

Nous avons déjà travaillé sur l'objet `Math`, notamment avec les méthodes `Math.round`, `Math.floor` et `Math.ceil` pour arrondir respectivement à l'entier le plus proche, par défaut et par excès. Cette fois-ci nous utiliserons la constante `Math.random` pour générer des nombres aléatoires.

Math.random()

Fichier de départ : 1001Deb fla

Fichier résultat : 1001Fin fla

Visualisation web : 1001Fin.html

Objectif : générer des nombres aléatoires entre 0 et 1 d'une part et entre deux nombres d'autre part.

Code JavaScript : `Math.random()` et création d'une fonction avec paramètres

- ☞ Ouvrez le fichier **1001Deb fla**.

La scène comporte deux cadres texte.

- ☞ Testez l'animation. Le navigateur affiche une fenêtre avec un titre et un cadre pour les résultats à venir.
- ☞ Dans le scénario, sélectionnez le calque **fond**.
À partir de la Bibliothèque, glissez trois occurrences du symbole **clip_Bt** en bas de la scène. Nommez-les **bt1**, **bt2** et **bt3**.
- ☞ Sélectionnez le calque **action** et affichez la fenêtre **Actions**.

Vous allez commencer par renseigner les étiquettes de chaque bouton :

- le bouton 1 devra afficher un nombre décimal compris entre 0 et 1,
- le bouton 2 un nombre entier supérieur ou égal à zéro et inférieur à 10,
- le bouton 3 un nombre entier supérieur ou égal à 20 et inférieur à 30.

- ☒ Pour cela, à partir de la ligne 3, saisissez le script :

```
this.bt1.etiquette.text="0 < Décimal < 1"
this.bt2.etiquette.text="0 <= Entier < 10"
this.bt3.etiquette.text="20 > Entier <30"
```

- ☒ Ensuite, créez des écouteurs sur chaque bouton en utilisant la méthode `for` :

```
//Ecouteurs
for(i=1; i<4; i++) {
    this["bt"+i].addEventListener("click", choixAlea.bind(this))
}
```

- ☒ Définissez ensuite la fonction `choixAlea()` :

```
//Fonctions
function choixAlea(e) {noAlea=e.currentTarget.name.substr(2,1)
    if(noAlea==1) {this.resultat.text=Math.random()}
    if(noAlea==2) {this.resultat.text=Math.round(10*Math.random())}
    if(noAlea==3) {this.resultat.text=aleatoire(20, 30)}
}
```

Dans cette fonction nous avons mis un paramètre (`e`) qui fait référence au bouton cliqué (`noAlea = e.currentTarget...`).

Si **noAlea** = 1, `Math.random()` retourne un nombre aléatoire compris entre 0 et 1.

Si **noAlea** = 2, `Math.round(10*Math.random())` retourne un nombre entier compris entre 10 et 20.

Si **noAlea** = 3, vous faites appel à une autre fonction **aleatoire** avec deux paramètres (**min** et **max**). Ces paramètres dans notre cas sont 20 pour minimum et 30 pour maximum.

- ☒ Définissez ainsi la fonction **aleatoire** :

```
function aleatoire(min, max) {
    return Math.floor(Math.random() * (max - min + 1) + min);
};
```

☐ Testez l'animation dans le navigateur et vérifiez que tout est ok.

```

action:1
1  this.titre.text="Nombres aléatoires"
2  this.resultat.text="? ? ?"
3  this.bt1.etiquette.text="0 < Décimal < 1"
4  this.bt2.etiquette.text="0 <= Entier < 10"
5  this.bt3.etiquette.text="20 > Entier <30"
6  //Ecouteurs
7  for(i=1; i<4; i++) {
8      this["bt"+i].addEventListener("click", choixAlea.bind(this))
9  }
10 //Fonctions
11 function choixAlea(e) {noAlea=e.currentTarget.name.substr(2,1)
12     if(noAlea==1) {this.resultat.text=Math.random()}
13     if(noAlea==2) {this.resultat.text=Math.round(10*Math.random())}
14     if(noAlea==3) {this.resultat.text=aleatoire(20, 30)}
15 }
16
17 function aleatoire(min, max) {
18     return Math.floor(Math.random() * (max - min + 1) + min);
19 };
20

```

Le script complet

Tester si deux occurrences se touchent

Fichier de départ : 1002Deb fla

Fichier résultat : 1002Fin fla

Visualisation web : 1002Fin.html

Objectif : tester si deux occurrences se touchent.

Code JavaScript : Math.random(), localToLocal, hitTest(), || (ou)

☐ Ouvrez le fichier **1002Deb fla**.

La scène comporte quatre calques. Seul le calque **textes** contient des cadres texte.

Le but de cet exercice est de réaliser un jeu. Comme le montre la figure ci-dessous, deux coccinelles sont en position sur la ligne de départ. Quand vous cliquez sur le bouton **Jouer**, les coccinelles s'élancent. Qui arrivera premier sur la ligne d'arrivée ?

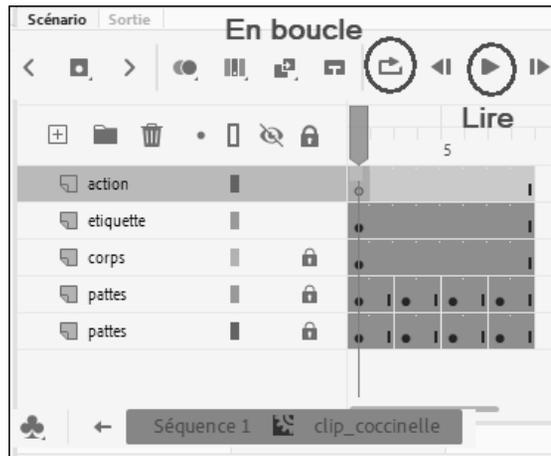


Présentation du jeu

- ☒ Dans le scénario, sélectionnez la première image du calque **lignes**.
- ☒ À partir de la Bibliothèque, glissez deux occurrences du symbole **clip_ligne**. Placez la première à l'abscisse **100** et à l'ordonnée **100**, nommez-la **ligne0**. Placez la seconde à l'abscisse **500** et à l'ordonnée **100**, nommez-la **ligne1**.
- ☒ Toujours sur le calque **lignes**, glissez une occurrence du symbole **clip_Bt**. Nommez-le **bt_Jouer** et placez-le en bas au centre de la scène.
- ☒ Ensuite, sur le calque **coccinelles**, glissez deux occurrences du symbole **clip_coccinelle**. Placez la première à l'abscisse **100** et à l'ordonnée **200**, nommez-la **cocci1**. Placez la seconde à l'abscisse **100** et à l'ordonnée **300**, nommez-la **cocci2**.
- ☒ Testez l'animation dans le navigateur. Vous devez voir les deux coccinelles sur la ligne de gauche en train de bouger leurs pattes.
- ☒ Revenez dans Animate afin d'améliorer cette situation de départ de course.

Nous allons procéder par étapes :

- Au départ, à l'étape 0, les coccinelles doivent être immobiles.
 - Ensuite, quand vous cliquez sur le bouton **Jouer**, c'est l'étape 1, la course est lancée.
 - Quand une des coccinelles arrive sur la ligne d'arrivée, la course s'arrête, c'est l'étape 2.
 - À l'étape 2, quand vous cliquez sur le bouton, vous revenez à l'étape 0.
- ☒ Faites un double clic sur une des coccinelles afin d'entrer dans son scénario. Vous remarquez que ce clip est constitué de 8 images.



Scénario du clip

- ☒ Cliquez sur les boutons **En boucle** puis **Lire** de la barre de navigation du scénario pour visualiser le mouvement des pattes.



Barre de navigation du scénario

Remarquez également que le calque **etiquette** contient un cadre texte destiné à afficher le numéro de la coccinelle.

- ☒ Placez-vous sur le calque **action**.

Vous avez nommé chaque coccinelle par **cocci1** et **cocci2**. Comme nous l'avons vu précédemment, les nombres 1 et 2 sont des caractères d'index 5 (le premier caractère porte l'index 0). Saisissez donc le script :

```
this.etiket.text = this.name.substr(5,1);
```

- ☒ Revenez sur le scénario principal et faites un double clic sur une des lignes (**ligne0** ou **ligne1**) afin d'entrer dans son scénario.

Le calque drapeau contient un cadre texte nommé **etiket** afin d'indiquer si c'est une ligne de départ ou d'arrivée.

- ☒ Revenez sur le scénario principal, sélectionnez le calque **action** et affichez la fenêtre **Actions**. En début du script, écrivez :

```
//Initialisation
this.titre.text="Course de coccinelles"
this.ligne0.etiket.text="Départ"
this.ligne1.etiket.text="Arrivée"
etape=0 ;
```

Comme vous procédez par étapes, vous avez précisé que, au départ, l'étape est 0. Pour la même raison, il va falloir définir un écouteur et une fonction qui actualise le numéro d'étape en fonction des événements.

- ☒ Complétez votre script ainsi :

```
//Ecouteurs
this.addEventListener("tick", actualiser.bind(this))
//Fonctions
function actualiser() {
this.etiqEtape.text=etape ;
    if(etape==0) {
        this.bt_Jouer.visible=true ;
        this.bt_Jouer.etiquette.text="Jouer" ;
        this.cocci1.gotoAndStop(2);
        this.cocci1.x=100 ;
        this.cocci2.gotoAndStop(6);
        this.cocci2.x=100 ;
    }
}
```

Vous avez sans doute remarqué que, en haut à gauche de la scène, il y a un cadre texte nommé **etiqEtape**. Il est là pour afficher le numéro de l'étape. C'est pour cette raison que vous commencez la fonction `actualiser()` par la ligne `this.etiqEtape.text=etape`.

Ensuite vous précisez la situation à l'étape 0 pour `bt_Jouer` (visible et texte) ainsi que pour les coccinelles (abscisse et numéro d'image pour le départ).

Quand vous cliquez sur `bt_Jouer` l'étape passe à 1. Il va donc falloir ajouter un écouteur et une fonction.

- ☒ Complétez le script des écouteurs :

```
//Ecouteurs
this.bt_Jouer.addEventListener("click", lancerJeu.bind(this)) ;
this.addEventListener("tick", actualiser.bind(this))
```

Et ajoutez en fin de script la fonction `lancerJeu()` qui permettra de passer à l'étape 1 ou de revenir à l'étape 0 :

```
function lancerJeu () {
    if(etape==0) {etape=1}
    if(etape==2) {etape=0}
}
```

- ☒ Testez l'animation dans le navigateur et vérifiez, grâce à l'étiquette située en haut à gauche de la fenêtre, le changement d'étape (de 0 à 1) lorsque vous cliquez sur le bouton **Jouer**.

Maintenant, à l'étape 1, il faudrait que chaque coccinelle se déplace vers la droite à des vitesses différentes. Vous allez définir pour chaque coccinelle un nombre aléatoire qui sera renouvelé régulièrement (à chaque événement **tick** de la fonction **actualiser**).

- ☒ Revenez dans `Animate` et complétez le code pour l'étape 1 en dessous de l'étape 0.

```
if(etape==1) {
    alea1=Math.random()
    this.cocci1.play() ;
    this.cocci1.x=this.cocci1.x+alea1
    alea2=Math.random()
    this.cocci2.play() ;
    this.cocci2.x=this.cocci2.x+alea2
    this.bt_Jouer.visible=false
}
```

Collisions

À l'étape 1, à chaque évènement **tick**, un nombre aléatoire **alea1** est généré pour la coccinelle 1 et un autre **alea2** pour la coccinelle 2. L'abscisse de chaque coccinelle augmente de ce nombre et la coccinelle se déplace vers la gauche. Comme les nombres aléatoires sont renouvelés régulièrement, chaque coccinelle peut accélérer ou ralentir. Les nombres aléatoires générés avec l'expression `Math.random()` sont compris entre 0 et 1.

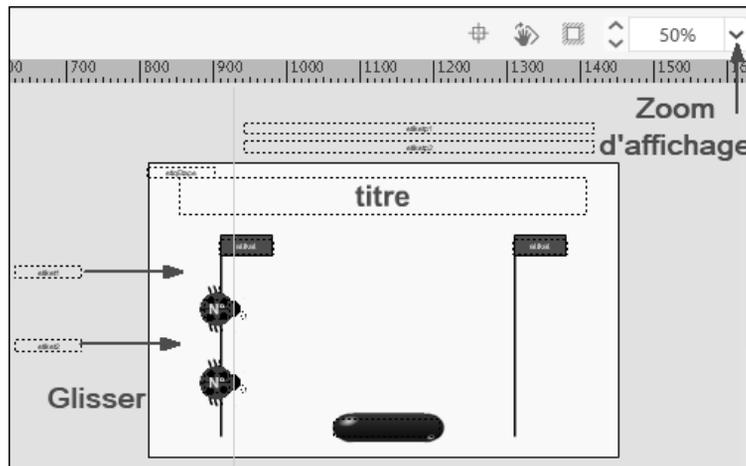
- ☒ Si le mouvement des coccinelles n'est pas assez rapide, vous pouvez par exemple remplacer le code :

```
this.cocci1.x=this.cocci1.x+alea1 ;
this.cocci2.x=this.cocci2.x+alea2 ;
```

par :

```
this.cocci1.x=this.cocci1.x+alea1*2 ; // Pour doubler la vitesse
this.cocci2.x=this.cocci2.x+alea2*2 ;
```

- ☒ Si vous voulez vous rendre compte de la valeur des nombres aléatoires qui sont générés, remarquez à gauche en dehors de la scène deux cadres textes **etiket1** et **etiket2**. Si vous ne les voyez pas, diminuez le zoom d'affichage de la scène.



Réglage du zoom d'affichage

- ☒ Glissez ces deux cadres sur la scène à côté de la ligne de départ et dans la fenêtre **Actions**, écrivez en dessous de `this.bt_Jouer.visible=false :`
`this.etiket1.text=alea1 ; this.etiket2.text=alea2 ;`