

# Jeu de Tir

## 1. Le premier monstre

Complète [jeu-de-tir.html](#) pour que Le joueur ait 2s pour cliquer sur le monstre. S'il clique à temps, le monstre disparaît, sinon « Fin du jeu » s'affiche.

- *Inspire-toi de l'exemple du cours : 5 secondes pour cliquer sur le bouton (slide 30)*
- *Tu pourras utiliser la méthode `remove()`.*
- *Que vaut `this` dans le gestionnaire d'événement ?*

## 2. Positionnement aléatoire

Le monstre doit apparaître à une position aléatoire dans tout le terrain de jeu.

- *`Math.random()` sera utile.*
- *Utilise les propriétés CSS `position : 'absolute'`, `top : '50px'`, `right : '100px'`*
- *Pour utiliser toute la place disponible et être précis, tu pourras utiliser `offsetHeight` et `offsetWidth`*
- *N'y passe pas trop de temps. N'hésite pas à t'aider de la correction.*

## 3. Le constructeur de monstres

Pour pouvoir créer facilement plusieurs monstres, écris un constructeur de monstres (une classe `Monstre`). Les monstres ont ces attributs :

- `element` : une référence vers l'objet du DOM correspondant à ce monstre
- `top` : le nombre de pixels entre le monstre et le bord du haut
- `right` : le nombre de pixels entre le monstre et le bord droit
- `timer` : le compte à rebours qui déclenchera la fin du jeu

+ les attributs et méthodes que tu utilises pour placer le monstre aléatoirement.

- *Attention de ne pas confondre une instance de `Monstre` et un élément du DOM que jusqu'ici on appelait monstre.*
- *Clone le monstre initial avec `cloneNode()`. Puis cache le monstre initial (quand faut-il faire ceci ?). Pour le cacher sans gêner les futurs clonages, utilise `removeChild()`.*
- *Si on ne clique pas à temps, le gestionnaire d'événements est exécuté. A l'intérieur, `this` pointe vers l'élément DOM. Comment récupérer le timer de l'objet `Monstre` correspondant ? Ajoute l'attribut `element.toMonstre`.*

## 4. Lâchez les monstres !

Crée 5 monstres au lieu d'un seul (avec une boucle).

## 5. L'horloge du jeu

Fais apparaître un nouveau monstre à chaque seconde.

- *On pense d'abord à exécuter un `setTimeout()` pour chaque monstre à créer. Mais combien faut-il en lancer ? Et quand lancer les suivants ?*
- *Nous allons plutôt utiliser un timer global qu'on appellera horloge. A chaque fois que passent 500ms il doit exécuter une fonction `action()`. Que va faire `action` ?*
- *`action()` va créer un monstre, et incrémenter temps (compteur de temps qui nous servira plus tard)*

## 6. Score

Affiche un score. Ajoute un point au score à chaque fois qu'un monstre disparaît.

## 7. Fin du jeu

Fais en sorte que le jeu s'arrête proprement : les monstres et les pop-ups doivent cesser d'apparaître.

## 8. Les monstres s'énervent

Fais en sorte que chaque monstre change de couleur au cours de sa vie. Lorsqu'il apparaît il est orange, au bout de 3,5 secondes il doit devenir rouge.