

# Web avancé – L2 MIASHS

## TD 03 - La citrouille explosée grâce à Javascript !

**Intervenants :** Jonathan Alcuta ([jonathan.alcuta@loria.fr](mailto:jonathan.alcuta@loria.fr)), Mira Bou Saleh ([mira.bou-saleh@univ-lorraine.fr](mailto:mira.bou-saleh@univ-lorraine.fr)), Gaël Guibon ([gael.guibon@univ-lorraine.fr](mailto:gael.guibon@univ-lorraine.fr))



Ce TD a pour objet de vous faire une mise à jour de la citrouille afin de finir le jeu à l'aide du... Javascript !

### Exercice 1 – Destruction de citrouilles 🎃

Cet exercice utilise le travail de l'exercice précédent pour faire apparaître des citrouilles à l'écran et les détruire en utilisant JavaScript et CSS.

1. Créez un fichier JavaScript permettant d'ajouter une citrouille à la page. Le script doit créer les mêmes éléments HTML que celui de l'exercice précédent.
2. Liez ce script avec votre HTML à l'aide de

```
<script type="text/javascript" src="halloween.js"></script>
```

où "halloween.js" est le chemin relatif pointant vers ce script javascript

3. Utilisez la propriété CSS « animation » de manière à faire apparaître la citrouille avec un redimensionnement allant de 0 (citrouille invisible) à 1 (taille initiale de la citrouille). Pour cela vous pouvez utiliser les keyframes CSS, voir par exemple : <https://developer.mozilla.org/fr/docs/Web/CSS/@keyframes>
4. Incorporez votre code JavaScript à une fonction **createPumpkin()** de manière à pouvoir faire apparaître plusieurs citrouilles à des positions aléatoires.
5. Utilisez la fonction JavaScript [setInterval\(\)](#) pour appeler la fonction **createPumpkin()** toutes les 500 ms (millisecondes), et faites en sorte que ces appels récurrents s'arrêtent si plus de 10 citrouilles sont apparues à l'écran.

6. Lors de la création de chaque citrouille (dans la fonction **addPumpkin()**), ajoutez un écouteur d'événement pour écouter [l'événement click](#) de la citrouille qui appelle une nouvelle fonction **destroy(pumpkin)** (à définir). Dans un premier temps, la fonction **destroy(pumpkin)** doit simplement supprimer la citrouille de l'arbre DOM de manière à ce qu'elle disparaisse de la page.

Sachez que l'événement click est le plus utilisé et le plus courant. Comme dit en CM vous avez plusieurs approches : l'attribut [onclick](#) ou l'ajout d'un écouteur d'événement par [addEventListener\(\)](#). Pour ce cas d'usage, **addEventListener()** est préférable. 😊

## Exercice 2 – Explosion de citrouilles 🎃💥

Challenge : avant de supprimer la citrouille de l'arbre DOM, ajoutez une animation d'explosion. Pour faire exploser la citrouille, une solution simple est de créer une vingtaine de nouvelles boîtes oranges réparties aléatoirement autour du centre de la citrouille :



Une fois ces boîtes créées, l'idée est d'ajouter une animation CSS à chacune de ces boîtes, de manière à ce qu'elles se déplacent depuis le centre de la citrouille vers leur position finale.

La principale difficulté est de déterminer ces positions finales. Pour cela, on peut jouer sur deux dimensions : (1) l'angle de rotation et (2) la position en abscisse.

La première utilise la propriété CSS « transform » et la fonction CSS « rotate » pour faire tourner le repère dans lequel sera placée la particule, voir par exemple :

[https://developer.mozilla.org/fr/docs/Web/CSS/transform-function/rotate\(\)](https://developer.mozilla.org/fr/docs/Web/CSS/transform-function/rotate())

On peut par exemple ajouter 36 degrés de rotation chaque fois qu'on passe d'une particule à l'autre. La seconde place la particule aléatoirement dans le repère ainsi pivoté, en tirant au sort une position sur x entre 0 et 300 pixels.

Une vidéo du résultat attendu est disponible sur Arche.

Et voilà ! Le TD est fini. 🙌

Voici un petit cadeau approprié : <https://youtube.com/shorts/s2Y3Zxm0X5I?feature=shared>