

# BATTLE ARENA

## Introduction

Battle Arena est une petite arène où s'affrontent 2 players.

Toute l'arène est déjà créée, ainsi que les animations et actions. Vous devez rendre le jeu fonctionnel. *Si possible, activez le son.*

## Contenu du sujet

- Fichier HTML & CSS du jeu,
- Assets (images, audio...)
- Fichiers JS :
  - general.js : contient toutes les fonctions du jeu,
  - game.js : contiendra votre code.

## Consignes

Vous avez la liberté de toucher à tous les fichiers JS.

## Initialisation du jeu

- Remplir les informations sur la page d'initialisation du jeu :
  - Le bouton pour lancer le jeu doit avoir la classe **is-disabled** tant que tous les champs ne sont pas remplis et valides (nombre de PV supérieur ou égal à 1).
  - Si tous les champs sont remplis et valides, le bouton pour lancer le jeu doit être cliquable (il ne doit plus avoir la classe **is-disabled**).
- Lors du lancement du jeu :
  - Le nom de chaque player est actualisé sur la page à la place de "Player 1" et "Player 2",

- Les barres de progressions deviennent vertes (elles perdent leur classe **is-pattern**) et sont définies sur les valeurs renseignées (cf. Barre de PV ci-dessous).
- Lancer la musique d'ambiance (cf. Fonctions ci-dessous).

## Barre de PV

Pour définir le total de PV de la barre de vie : modifier son attribut **max**.

Pour définir le nombre de PV actuel de la barre de vie : modifier son attribut **value**.

## Fonctions

### **attack**(*player, attackName*)

Cette fonction permet d'animer une attaque. Elle prend 2 paramètres :

1. Le player (1 ou 2),
2. Le nom de l'attaque (sword, spear, punch, moew, claw, crocroc).

Exemple : **attack(1, "moew")** pour animer l'attaque meoww sur le player 1.

---

### **damage**(*player, amount*)

Cette fonction permet d'afficher l'animation de perte de PV. Elle prend 2 paramètres :

3. Le player (1 ou 2),
4. Le nombre de PV perdus.

Exemple : **damage(2, 5)** pour animer une perte de 5 PV sur le player 2.

---

## **victory(player)**

Cette fonction permet d'afficher la victoire d'un player. Elle prend un seul paramètre :

1. Le player (1 ou 2).

Exemple : **victory(1)** pour animer la victoire du player 1.

---

## **LaunchMusic()**

Cette fonction lance la musique d'ambiance.

---

## Fonctionnement

- Pour chaque attaque subie :
  - Réduire le nombre de PV du player à chaque attaque subie,
  - Animer la perte de PV avec la fonction **damage()**.
- Faire en sorte que les boutons d'attaque lancent les attaques (*le nombre de dégâts infligés par chaque attaque doit correspondre au nombre dans le bouton de l'attaque*),
- Si une attaque inflige plus de dégâts qu'il ne reste de PV, utiliser la fonction **victory()** pour afficher la victoire du player victorieux.

## Les bonus incroyables

- Mettre une musique victorieuse lors de la victoire,
- Mettre en place un système de tour par tour :
  - Lorsqu'un player joue, l'autre ne peut pas jouer (ses boutons d'actions sont disabled).
  - Créer un sort de régénération pour les 2 players (le gif et le son sont dans les assets).

# Conseils du sage

- Bien lire les consignes,
- Ne pas perdre de temps à rester bloqué·e sur quelque chose,
- Procéder étape par étape en suivant la grille de notation,
- Écouter de la bonne musique en travaillant,
- Ne pas s'échanger du code sinon j'appelle la police,
- Ne pas penser que ça ne se voit pas si le code partagé est un peu modifié.