

GDWFSDVSWEBAJAVAEXAIII1A

Prénom : Mickaël  
NOM : PARDAL

Nom du projet : Jeu de Numéro  
Lien GitHub du projet : [https://github.com/mpardal/Jeu\\_Des](https://github.com/mpardal/Jeu_Des)  
URL du site : <https://jeu-des.vercel.app/>

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions. Dans cette rubrique, le jury cherche à voir comment vous procédez : comment vous organisez votre travail, comment vous réalisez concrètement la tâche ou l'opération pas à pas. Utiliser un langage professionnel. Employez le « je », car vous parlez en votre nom. Vous pouvez écrire au temps présent.

Tout d'abord, j'ai démarré par une petite partie de configuration HTML en impliquant seulement un joueur, afin de trouver le bon fonctionnement du jeu. Puis j'ai mis mes lignes de codes link, script et meta dans mon head.

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta charset="UTF-8" />
  <link rel="stylesheet" href="style.css" />
  <link
    rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.3/font/bootstrap-icons.css"
  />
  <link rel="preconnect" href="https://fonts.googleapis.com" />
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
  <link
    href="https://fonts.googleapis.com/css2?family=Lato:wght@100;400&display=swap"
    rel="stylesheet"
  />
  <script src="script.js" defer></script>
  <title>Jeu de dés</title>
</head>
```

Ensuite, j'ai fait mes premières fonctions JavaScript. Tout d'abord, j'ai déclaré les variables de mon HTML dans mon fichier « script.js ».

```
//Déclaration des boutons
const reset = document.querySelector("#reset");
const replay = document.querySelector("#replay");
const hold = document.querySelector("#hold");

//Déclaration des écritures
const labelTourP1 = document.querySelector("#labelTourP1");
const labelGlobalP1 = document.querySelector("#labelGlobalP1");
const labelTourP2 = document.querySelector("#labelTourP2");
const labelGlobalP2 = document.querySelector("#labelGlobalP2");
```

Puis j'ai commencé à faire des événements, afin d'agir sur le jeu en cliquant sur les éléments correspondant à « reset », « replay » et « hold ».


```
replay.addEventListener("click", () => {
  let number = generateNumber();

  if (currentPlayer === player1) {
    styleP1();
    dice(number);
    tirage(number);
    updateHtml(player1);
  } else {
    styleP2();
    dice(number);
    tirage(number);
    updateHtml(player2);
  }
});
```

Ensuite, j'ai rajouté le résultat des différentes actions faites dans les éléments à leurs labels respectifs, à l'aide des propriétés « innerHTML » ou « innerText ». Une fois que cette partie fonctionnait, j'ai ajouté le deuxième joueur.

```
function name() {  
  player1.name = prompt("Quel est le nom du joueur 1 ?");  
  player2.name = prompt("Quel est le nom du joueur 2 ?");  
  if (player1.name === "") {  
    player1.name = "PLAYER 1";  
    p1.innerHTML = player1.name;  
  } else {  
    p1.innerHTML = player1.name;  
  }  
  if (player2.name === "") {  
    player2.name = "PLAYER 2";  
    p2.innerHTML = player2.name;  
  } else {  
    p2.innerHTML = player2.name;  
  }  
}
```

Puis j'ai créé des fonctions, pour permettre la réutilisation des éléments à plusieurs endroits du code.



```
// Gestion du bouton pour jouer
replay.addEventListener("click", () => {
  let number = generateNumber();

  if (currentPlayer === player1) {
    styleP1();
    dice(number);
    tirage(number);
    updateHtml(player1);
  } else {
    styleP2();
    dice(number);
    tirage(number);
    updateHtml(player2);
  }
});
```

Par exemple, ici j'ai créé une fonction « generateNumber » afin que l'on comprenne bien que la variable « number », présente dans l'évènement lié à « replay », permet de générer le numéro aléatoire de à chaque tour.

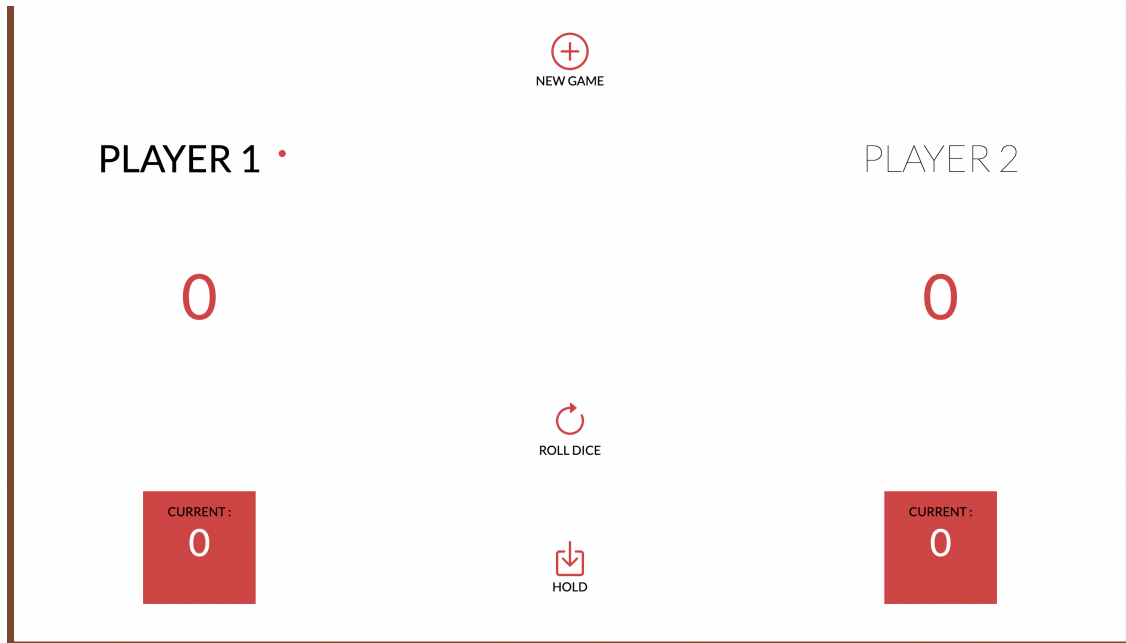
```
/**
 * Permet de définir un nombre aléatoire
 *
 * @return {number}
 */
function generateNumber() {
    return Math.floor(Math.random() * (7 - 1)) +
};
```

La fonction « updateHtml », quand à elle, permet de stocker les changements des résultats à chaque changement de valeurs de « round » ou « global ».

```
/**
 * Permet de mettre à jour le html avec les scores en paramètres
 *
 * @param {Player} p
 */
function updateHtml(p) {
    if (p === player1) {
        labelTourP1.innerHTML = p.round.toString();
        labelGlobalP1.innerHTML = p.global.toString();
    } else {
        labelTourP2.innerHTML = p.round.toString();
        labelGlobalP2.innerHTML = p.global.toString();
    }
}
```

Ensuite, je me suis mis à faire le graphisme, je n'ai pas pris de framework puisque il n'y a pas beaucoup de détails graphiques utilisés dans mon jeu. Afin de faire la disposition de chaque élément, j'ai utilisé le « display : grid ». Et ensuite j'ai alimenté mon fichier « style.css » pour mettre tout en place les points graphiques.

Les icônes ont été pris auprès de Bootstrap, en injectant le CDN des icônes de Bootstrap.



2. Précisez les moyens utilisés. Expliquez tout ce dont vous avez eu besoin pour réaliser vos tâches : langages de programmation, frameworks, outils, logiciels, documentations techniques, etc...

Pour ce programme, je n'ai pas utilisé de framework à part pour les icônes, où j'ai utilisé le CDN des icônes de Bootstrap.

Dans ce programme, il y a du langage CSS, JavaScript et HTML.

J'ai utilisé l'IDE PHPStorm de JetBrains.

La documentation utilisée est principalement MDN.

3. Contexte. Les noms des organismes, entreprises ou associations, dans lesquels vous avez exercé vos pratiques.

J'ai écrit et développé ce site Web seul.

## **CORRECTION**

Bonjour Mickaël ,

Merci pour votre travail.

Le résultat est excellent. Fonctionnement OK. La présentation et les effets visuels sont réussis.

Il manque un petit effet audio, qui dynamiserait davantage l'appli, et rendrait le jeu plus immersif.

GitHub : Bien, le code est bien organisé, les variables sont déclarées en début de script.

bonne utilisation. commits réguliers, avec commentaires.

Scripts HTML et JS commentés, pas le CSS.

Algorithme : fonctionnel , le jeu répond bien aux conditions proposées dans l'énoncé.

L'affichage du vainqueur apparaît trop rapidement, le résultat numérique , lui , n'est pas affiché.

Appli Responsive : fonctionnel, le redimensionnement reste ergonomique.

Description du projet : Bien. Ce sera utile par la suite, pour votre préparation du dossier professionnel.

Mise en ligne : OK

Charte graphique : bien, mais il manque un peu de graphisme pour améliorer la présentation.

Bonne préparation, l'appli est tout à fait fonctionnelle, et répond bien aux conditions du cahier des charges. La description du projet est bien détaillée.

Pour aller plus loin : Vous avez la possibilité de rendre le jeu plus ludique en animant le dé :

<https://fr.wikipedia.org/wiki/Three.js>

<https://threejs.org/>

<https://icodemag.com/3d-rolling-dice-css-javascript/>

[https://developer.mozilla.org/fr/docs/Web/CSS/transform-function/rotate3d\(\)](https://developer.mozilla.org/fr/docs/Web/CSS/transform-function/rotate3d())

1. Connaître et utiliser un environnement de développement (2 points). 2
2. Écrire un algorithme et l'intégrer dans une page web avec des scripts événementiels et avec un langage de script côté client (5 points). 4
3. Utilisation des normes ECMAScript (JS) et du DOM (5 points). 4
4. Connaissance et utilisation d'un framework de présentation de type adaptatif (5 points). 3
5. Déploiement de la réalisation en ligne (3 points). 3