



# Travail Dirigé JavaScript

JEU DU LANCER DE DE

Joachim Zadi | TD - DWWM | 14 septembre 2020

# Présentation du jeu

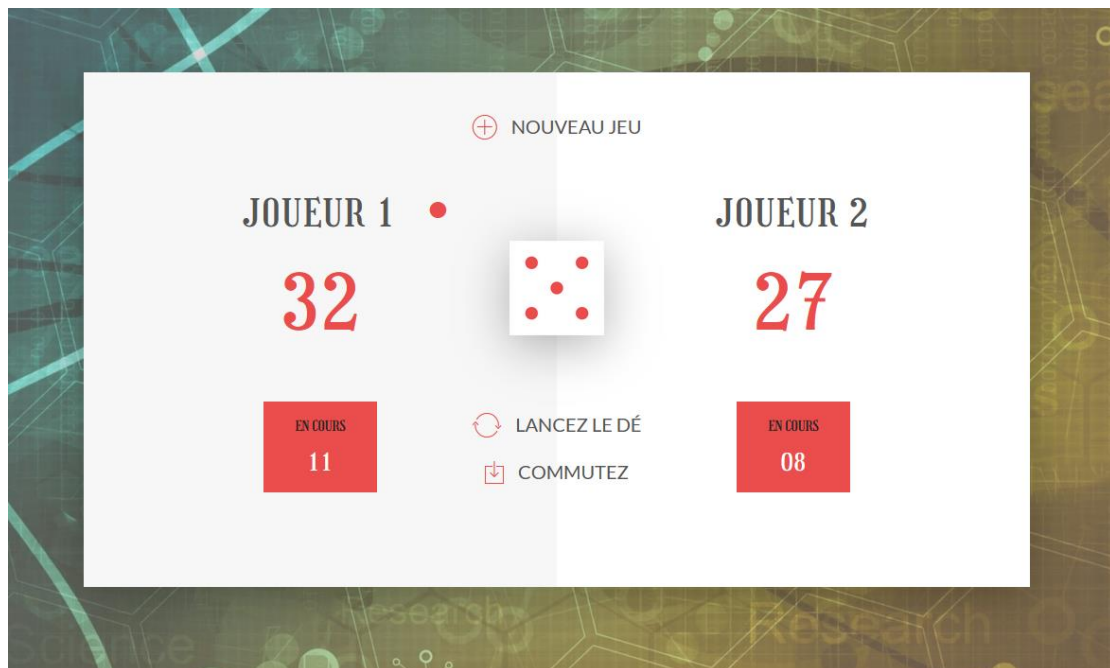


Figure 1: Interface du Jeu

- I. Le jeu est à 2 joueurs, **Joueur 1** et **Joueur 2**, jouant alternativement en séries en cliquant sur le bouton « LANCEZ LE DÉ ».
- II. A chaque tour, un joueur lance un dé autant de fois qu'il le souhaite. Chaque résultat est ajouté à son score EN COURS.
- III. MAIS, si le joueur obtient un 1 au cours d'une série de lancer, tout son score EN COURS est perdu, et doit obligatoirement passer la main au joueur concurrent.
- IV. Le joueur peut choisir de conserver son score EN COURS avec le bouton « COMMUTEZ », ce qui a pour effet de l'ajouter à son score GLOBAL, mais dans ce cas, il doit passer la main à son concurrent.
- V. Le premier joueur à atteindre 100 points sur le score GLOBAL gagne la partie.

# Arborescence du projet

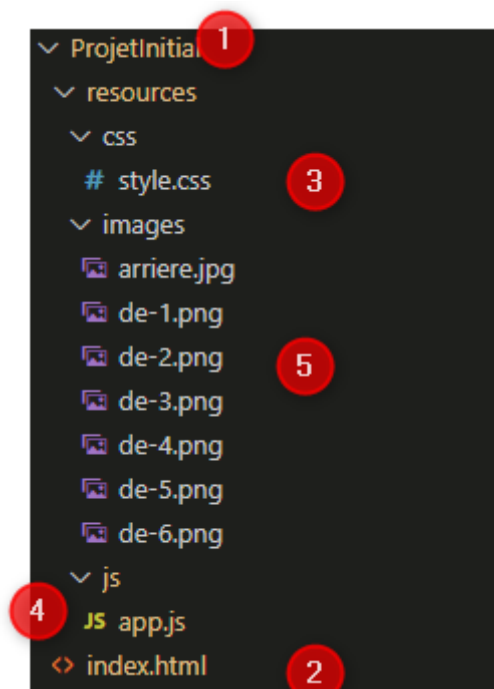


Figure 2: Arborescence du projet

1. La racine du projet qui contient un répertoire « resources » et un fichier « index.html »
2. Le fichier « index.html » permettant d'afficher le jeu dans le navigateur.
3. Le fichier « style.css » de style permettant de mettre en forme la page index.html
4. Le fichier « app.js » de javascript, permettant de rendre notre jeu dynamique.
5. Les images pris en compte dans la réalisation du jeu.

## Code du projet initial

Dans cette section, nous présentons le code initial de notre projet. Remarquons que notre projet n'est pas « RESPONSIVE », car notre but ici est de voir comment mettre en pratique le JavaScript basiquement sur tel projet.

### CONTENU DU FICHIER « index.html »

```
<!DOCTYPE html>
<html lang="fr">

  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="http://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css" />
    <link rel="stylesheet" href="./resources/css/style.css" />
    <title>Dé Virtuel</title>
    <script src="resources/js/app.js" defer></script>
  </head>

  <body>
    <!-- PANNEAU DU JEU -->
    <div class="wrapper clearfix">

      <!-- PANNEAU DU JOUEUR 1 -->
      <div class="joueur-1-panel active">
        <div class="nom-joueur" id="nom-1">Joueur 1</div>
        <div class="score-joueur" id="score-1">32</div>
      </div>
    </div>
  </body>
</html>
```

```

        <div class="joueur-courant-box">
            <div class="joueur-courant-label">En cours</div>
            <div class="joueur-courant-score" id="encours-1">11</div>
        </div>
    </div>

    <!-- PANNEAU DU JOUEUR 2 -->
    <div class="joueur-2-panel">
        <div class="nom-joueur" id="nom-2">Joueur 2</div>
        <div class="score-joueur" id="score-2">27</div>
        <div class="joueur-courant-box">
            <div class="joueur-courant-label">En cours</div>
            <div class="joueur-courant-score" id="encours-2">08</div>
        </div>
    </div>

    <!-- LES BOUTTONS -->
    <button class="btn-nouveau"><i class="ion-ios-plus-outline"></i>Nouveau Jeu</button>
    <button class="btn-lancer"><i class="ion-ios-loop"></i>Lancez le Dé</button>
    <button class="btn-commuter"><i class="ion-ios-download-outline"></i>Commutez</button>
    <!-- L'IMAGE DU DE -->
    
    </div>
</body>
</html>

```

Figure 3: Fichier "index.html"

## CONTENU DU FICHER « style.css »

```

/* ++++++
STYLE GENERAL
+++++ */
@import url('https://fonts.googleapis.com/css?family=Girassol|Lobster|Lato&display=swap');
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    background-image: linear-gradient(rgba(206, 183, 183, 0.4), rgba(206, 183, 183, 0.4)),
    url('../images/arriere.jpg');
    background-size: cover;
    background-position: center;
    font-family: Lato;
    font-weight: 300;
    position: relative;
    height: 100vh;
    color: #555;
}

.wrapper {
    width: 1000px;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    background-color: #fff;
    box-shadow: 0px 10px 50px rgba(0, 0, 0, 0.3);
    overflow: hidden;
}

.clearfix::after {
    content: "";
    display: table;
    clear: both;
}

.joueur-1-panel, .joueur-2-panel {
    width: 50%;
    float: left;
    padding: 100px;
    font-family: 'Girassol', cursive;
}

/* ++++++
STYLE JEU
+++++ */
.nom-joueur {
    font-size: 40px;
    text-align: center;
    text-transform: uppercase;
}

```

```

        letter-spacing: 2px;
        font-weight: 100;
        margin-top: 20px;
        margin-bottom: 10px;
        position: relative;
    }

    .score-joueur {
        text-align: center;
        font-size: 80px;
        font-weight: 100;
        color: #EB4D4D;
        margin-bottom: 50px;
    }

    .active {
        background-color: #f7f7f7;
    }

    .active .nom-joueur {
        font-weight: 300;
    }

    .active .nom-joueur::after {
        font-family: Lato;
        content: '\2022';
        font-size: 47px;
        position: absolute;
        color: #EB4D4D;
        top: -7px;
        right: 10px;
    }

    .joueur-courant-box {
        background-color: #EB4D4D;
        color: #fff;
        width: 40%;
        margin: 0 auto;
        padding: 15px;
        text-align: center;
    }

    .joueur-courant-label {
        text-transform: uppercase;
        margin-bottom: 10px;
        font-size: 15px;
        color: #222;
    }

    .joueur-courant-score {
        font-size: 25px;
    }

    /* ++++++ STYLE BOUTON ++++++ */
    button {
        position: absolute;
        width: 200px;
        left: 50%;
        transform: translateX(-50%);
        color: #555;
        background: none;
        border: none;
        font-family: Lato;
        font-size: 20px;
        text-transform: uppercase;
        cursor: pointer;
        font-weight: 300;
        transition: background-color 0.3s, color 0.3s;
    }

    button:hover {
        font-weight: 600;
    }

    button:hover i {
        margin-right: 20px;
    }

    button:focus {
        outline: none;
    }

    i {
        color: #EB4D4D;
        display: inline-block;
        margin-right: 15px;
    }

```

```

    font-size: 32px;
    line-height: 1;
    vertical-align: text-top;
    margin-top: -4px;
    transition: margin 0.5s;
}

.btn-nouveau {
    top: 45px;
}

.btn-lancer {
    top: 360px;
}

.btn-commuter {
    top: 410px;
}

.de {
    position: absolute;
    left: 50%;
    top: 178px;
    transform: translateX(-50%);
    height: 100px;
    box-shadow: 0px 10px 50px rgba(0, 0, 0, 0.3);
}

.vainqueur {
    background-color: #f7f7f7;
}

.vainqueur .nom-joueur {
    font-weight: 300;
    color: #17a623;
}

```

## CONTENU DU FICHIER « APP.JS »

Remarquons que notre fichier « app.js » est complètement vierge. Notre travail consiste à le remplir de code javascript nécessaire au bon fonctionnement de notre jeu.

# Développement JS du jeu

## VARIABLES ET INITIALISATION DU JEU

```

/* ***** */
/* PARTIE 1 ==> DECLARATION DES VARIABLES DU PROGRAMME */
/* ***** */

let scoresGlobals = [0, 0];
let scoreEnCours = 0;
let joueurActif = 1;

```

Les noms des variables sont assez intuitifs :

- **scoresGlobals** : c'est un tableau destiné à stocker les deux scores globaux des deux joueurs.
- **scoreEnCours** : permet de stocker le score d'une série de lancés pour un joueur donné.
- **joueurActif** : permet de designer le joueur actif au cours du jeu. Cette n'accepte que les valeurs 1 ou 2.

Dans le projet initial, lors du lancement de notre jeu, on a l'interface suivante qui s'affiche.

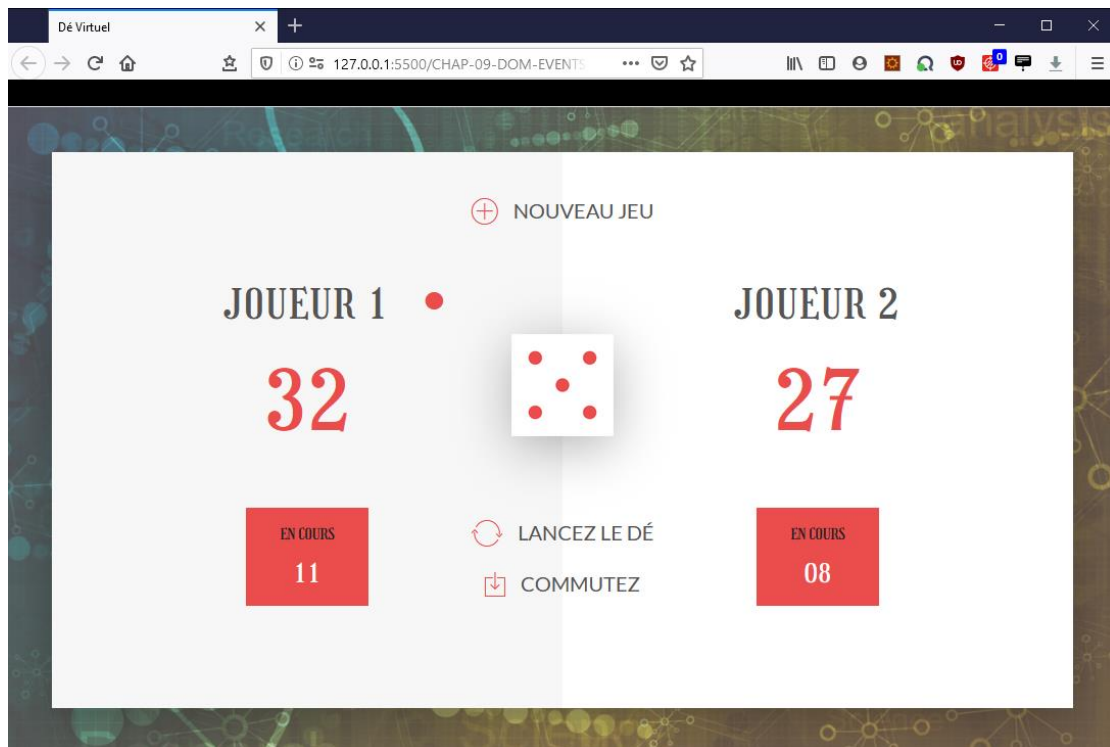


Figure 4 : GUI du projet initial

Ce que nous, nous souhaitons, c'est d'avoir un jeu qui démarre, avec tous les « scores globaux » et « en cours » initialisés à zéro, avec pour joueur actif le « JOUEUR 1 » comme sur la figure suivante :

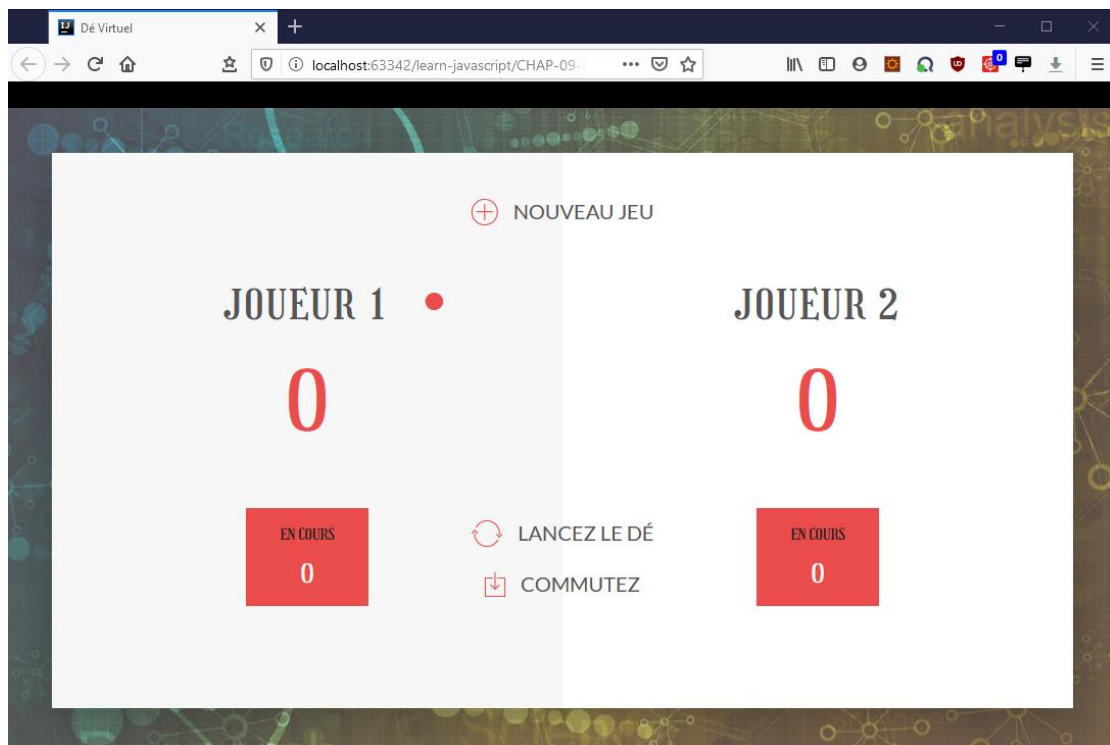


Figure 5: GUI initialisé par JavaScript pour le jeu

Pour pouvoir réaliser ce travail, ajoutons le code suivant à notre fichier « **app.js** » :

```

/* ***** */
/* PARTIE 2 ==> INITIALISATION DU GUI DU JEU */
/* ***** */
document.getElementById('score-1').textContent = '0';
document.getElementById('score-2').textContent = '0';
document.getElementById('encours-1').textContent = '0';
document.getElementById('encours-2').textContent = '0';
document.querySelector('.de').style.display = 'none';

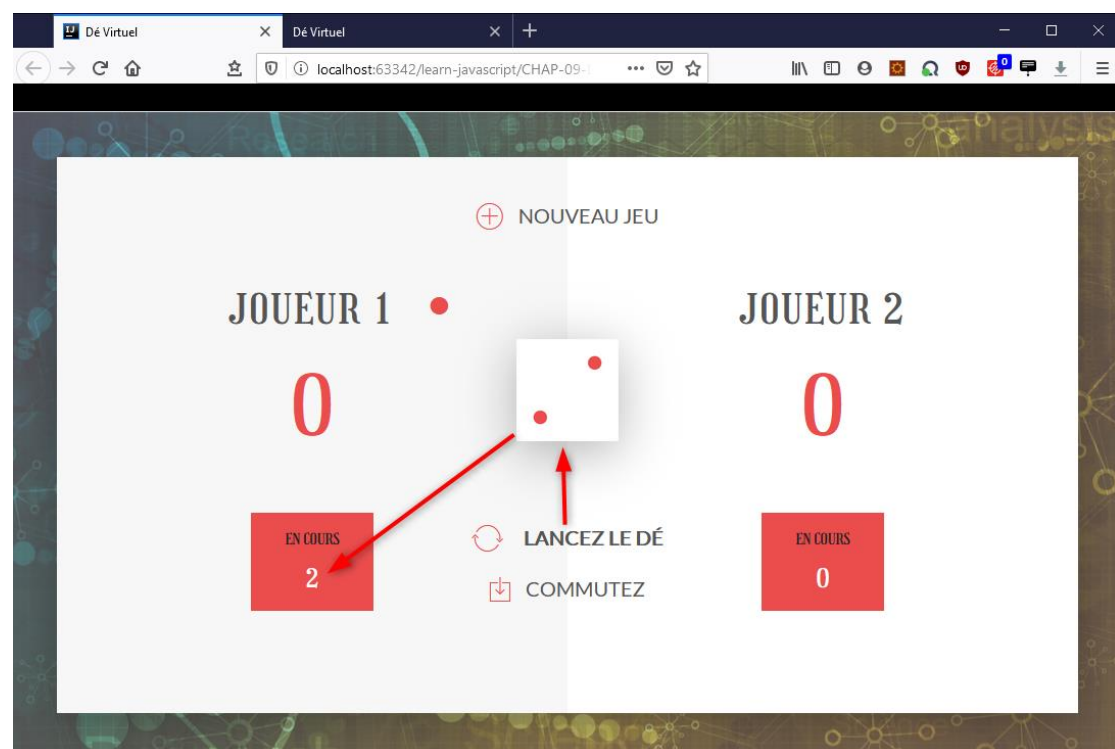
```

Dans le code précédent, les quatre premières lignes ciblent les différents scores à l'aide de leur « id » HTML et initialise leur contenu à « 0 » en JavaScript.

La dernière ligne cible l'image du « Dé » qui a pour attribut de classe « **.de** », puis le masque à l'utilisateur avec la valeur « **none** ».

A ce stade, vous pouvez déjà vérifier que notre GUI réagit très bien, ce qui est sympa pour la suite. Notre code sera appelé à évoluer plus tard.

## GESTION DU CLIC SUR « LANCEZ LE DE »



Selon les règles du jeu, lorsqu'on clique sur le bouton « LANCEZ LE DE », une image du Dé correspondant au tirage du lancer s'affiche, puis met à jour le SCORE EN COURS du joueur actif.

Je vous propose dans un premier temps le codage en JavaScript de cette fonctionnalité.

```

/* ***** */
/* PARTIE 3 ==> GESTION DU CLICK SUR LE BOUTON 'LANCEZ LE DE' */
/* ***** */

//On cible le bouton à l'aide de sa classe
let btnLancezLeDe = document.querySelector('.btn-lancer');

//On attache l'événement de click à notre bouton avec une fonction anonyme lambda.
btnLancezLeDe.addEventListener('click', () => {

    //Je simule le lance le Dé et encapsule le resultat dans la variable de
    let de = Math.floor(Math.random() * 6) + 1;

    //Je met à jour j'image du dé dans le GUI
    let domDe = document.querySelector('.de');
    domDe.style.display = 'block';
    domDe.src = `resources/images/de-${de}.png`;

```



```

//On met à jour le score en cours du joueur actif
scoreEnCours += de;
document.getElementById(`encours-${joueurActif}`).textContent = scoreEnCours;
});

```

Figure 6 : Code du bouton « LANCEZ LE DE »

Avec le code précédent, vous pouvez tester notre jeu, et vous constatez qu'il fonctionne bien. Si vous passez la valeur 2 à notre joueur actif, vous voyez bien que tout est OK ! Bravo si vous êtes dans ce cas.

Pour l'instant, vous constatez que même quand notre lancer nous sort le chiffre « 1 », cela met à jour notre score EN COURS : Ce qui est contraire à la règle du jeu III.

Pour être conforme à la règle III de notre jeu, modifions le code de la PARTIE 3 comme suit :

```

/* ***** */
/* PARTIE 3 ==> GESTION DU CLICK SUR LE BOUTON 'LANCEZ LE DE' */
/* ***** */
//On cible le bouton à l'aide de sa classe
let btnLancezLeDe = document.querySelector(`.btn-lancer`);

//On attache l'événement de click à notre bouton avec une fonction anonyme lambda.
btnLancezLeDe.addEventListener('click', () => {

    //Je simule le lancer du dé et encapsule le résultat dans la variable de
    let de = Math.floor(Math.random() * 6) + 1;

    //Je met à jour l'image du dé dans le GUI
    let domDe = document.querySelector('.de');
    domDe.style.display = 'block';
    domDe.src = `resources/images/de-${de}.png`;

    //On met à jour le score en cours du joueur actif sauf si le lancer vaut 1
    if (de !== 1) {
        //Ici, le lancer a sorti une valeur != 1.
        // On met donc à jour le score en cours du joueur actif
        scoreEnCours += de;
        document.getElementById(`encours-${joueurActif}`).textContent = scoreEnCours;
    } else {
        //Ici, le lancer a sorti la valeur 1.
        // On passe donc la main au joueur suivant

        //J'utilise une expression ternaire pour faire l'alternance entre les deux joueurs
        joueurActif === 1 ? joueurActif = 2 : joueurActif = 1;

        //J'initialise le score en cours du joueur actif à la valeur 0
        scoreEnCours = 0;

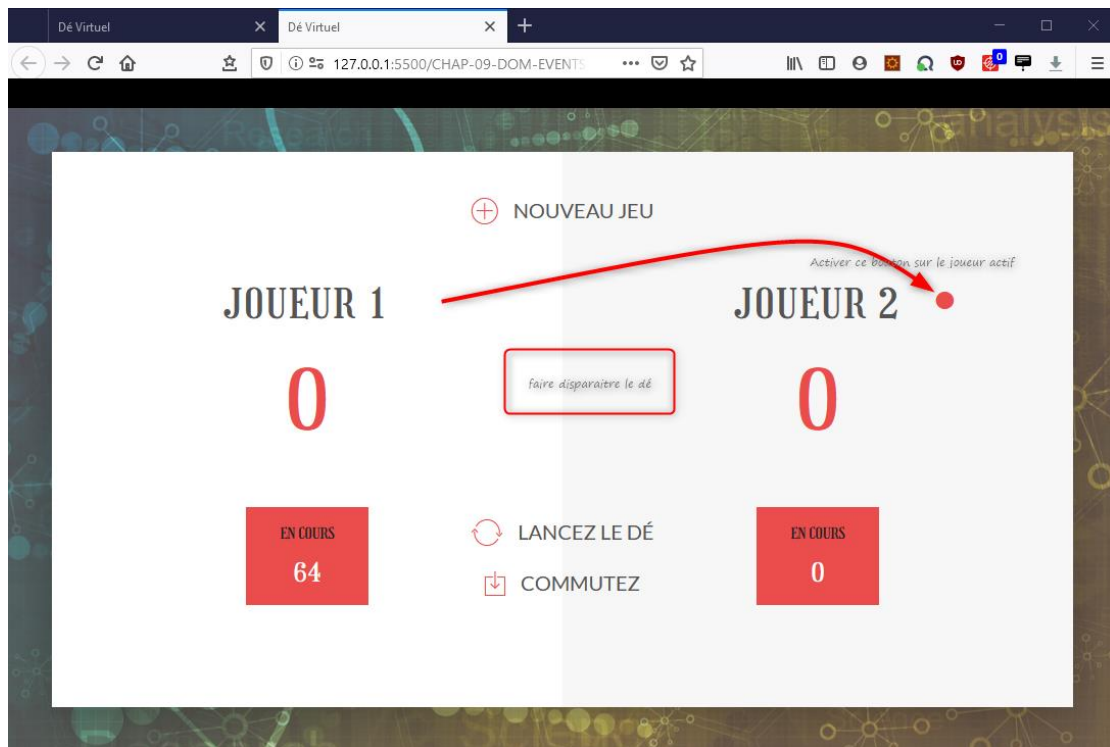
        //Je mets à jour le GUI du score en cours pour le joueur actif
        document.getElementById(`encours-${joueurActif}`).textContent = '0';
    }
});

```

Figure 7 : MAJ du code du bouton « LANCEZ LE DE »

BRAVO !!! Notre code est toujours fonctionnel.

Mais ce que nous souhaitons, c'est qu'en cas de passage de main au joueur concurrent, on commute le « voyant actif » au joueur actif et on fait disparaître le « Dé » du GUI. Comme sur la figure suivante :



Pour cela modifions le code de notre bouton « LANCEZ LE DE » comme suit :

```

/* ***** */
/* PARTIE 3 ==> GESTION DU CLICK SUR LE BOUTON 'LANCEZ LE DE' */
/* ***** */
//On cible le bouton à l'aide de sa classe
let btnLancezLeDe = document.querySelector(`.btn-lancer`);

//On attache l'événement de click à notre bouton avec une fonction anonyme lambda.
btnLancezLeDe.addEventListener('click', () => {

    //Je simule le lancer du dé et encapsule le résultat dans la variable de
    let de = Math.floor(Math.random() * 6) + 1;

    //Je mets à jour l'image du dé dans le GUI
    let domDe = document.querySelector('.de');
    domDe.style.display = 'block';
    domDe.src = `resources/images/de-${de}.png`;

    //On met à jour le score en cours du joueur actif sauf si le lancer vaut 1
    if (de !== 1) {
        //Ici, le lancer a sorti une valeur != 1.
        // On met donc à jour le score en cours du joueur actif
        scoreEnCours += de;
        document.getElementById(`encours-${joueurActif}`).textContent = scoreEnCours;
    } else {
        //Ici, le lancer a sorti la valeur 1.
        // On passe donc la main au joueur suivant

        //J'utilise une expression ternaire pour faire l'alternance entre les deux joueurs
        joueurActif === 1 ? joueurActif = 2 : joueurActif = 1;

        //J'initialise le score en cours du joueur actif à la valeur 0
        scoreEnCours = 0;

        //Je mets à jour le GUI du score en cours pour le joueur actif
        document.getElementById(`encours-${joueurActif}`).textContent = '0';

        //Je mets à jour le GUI du voyant actif pour le joueur actif
        document.querySelector('.joueur-1-panel').classList.toggle('active');
        document.querySelector('.joueur-2-panel').classList.toggle('active');

        //Je mets à jour le GUI du dé en le masquant
        document.querySelector('.de').style.display = 'none';
    }
});

```

Et HOP, tout marche conformément à la règle III. BRAVO à vous, vous venez d'implémenter l'événement de click du bouton « LANCEZ LE DE ». Le code évoluera bien sûr dans les prochaines sections.

## GESTION DU CLIC SUR « COMMUTEZ »

Le temps est maintenant venu d'implémenter l'événement « click » du bouton « COMMUTEZ », et cela conformément à la règle IV des règles de jeu.

Rappelons que lorsque vous commutez, vous décidez d'interrompre votre série de lancer, en mettant à jour votre score GLOBAL avec les acquis du score EN COURS.

Ajoutons donc le code suivant pour réaliser cette fonctionnalité.

```
/* ***** */
/* PARTIE 4 ==> GESTION DU CLICK SUR LE BOUTON 'COMMUTEZ' */
/* ***** */
let btnCommutez = document.querySelector('.btn-commuter');
btnCommutez.addEventListener('click', () => {

    //Lors de la commutation, je mets à jour le tableau de scoresGlobals
    //On oubliera pas qu'un tableau debute avec un index 0
    //Le joueurActif valant soit 1, soit 2, on prendra soin d'adpater le tableau
    scoresGlobals[joueurActif - 1] += scoreEnCours;

    //Je mets à jour le GUI du score GLOBAL
    document.getElementById(`score-${joueurActif}`).textContent=scoresGlobals[joueurActif-1];

    //Je passe la main au joueur concurrent en procédant exactement comme dans la PARTIE 3
    joueurActif === 1 ? joueurActif = 2 : joueurActif = 1;
    scoreEnCours = 0;
    document.getElementById(`encours-${joueurActif}`).textContent = '0';
    document.querySelector('.joueur-1-panel').classList.toggle('active');
    document.querySelector('.joueur-2-panel').classList.toggle('active');
    document.querySelector('.de').style.display = 'none';
});
```

Notre bouton fonctionne à merveille... Que c'est cool alors !

Le seul hic ici, est que nous avons du code répétitif dans la PARTIE III et la PARTIE IV, et ce n'est pas intéressant. Comment résoudre ce problème ? Nous allons faire appelle à une fonction que l'on nommera « **joueurSuivant** », dont le code est le suivant :

```
/* ***** */
/* PARTIE 5 ==> IMPLEMENTATION DE LA FONCTION 'joueurSuivant' */
/* ***** */

//Fonction permet de commuter le jeu d'un joueur à l'autre en mettant
//à jour les scores GLOBAUX.
//Son contenu n'est rien d'autre que le code répétitif des PARTIES III & IV
function joueurSuivant() {
    joueurActif === 1 ? joueurActif = 2 : joueurActif = 1;
    scoreEnCours = 0;
    document.getElementById(`encours-${joueurActif}`).textContent = '0';
    document.querySelector('.joueur-1-panel').classList.toggle('active');
    document.querySelector('.joueur-2-panel').classList.toggle('active');
    document.querySelector('.de').style.display = 'none';
}
```

Cette fonction nous permet de mettre à jour les parties III & IV, ce que nous allons faire maintenant.

```
/* ***** */
/* PARTIE 3 ==> GESTION DU CLICK SUR LE BOUTON 'LANCEZ LE DE' */
/* ***** */
//On cible le bouton à l'aide de sa classe
let btnLancezLeDe = document.querySelector('.btn-lancer');

//On attache l'evenement de click à notre bouton avec une fonction anonyme lambda.
btnLancezLeDe.addEventListener('click', () => {

    //Je simule le lance le Dé et encapsule le resultat dans la variable de
    let de = Math.floor(Math.random() * 6) + 1;

    //Je met à jour j'image du dé dans le GUI
    let domDe = document.querySelector('.de');
    domDe.style.display = 'block';
    domDe.src = `resources/images/de-${de}.png`;

    //On met à jour le score en cours du joueur actif sauf si le lancer vaut 1
    if (de !== 1) {
        //Ici, le lancer à sorti une valeur != 1.
    }
});
```

```

        // On met donc à jour le score en cours du joueur actif
        scoreEnCours += de;
        document.getElementById(`encours-${joueurActif}`).textContent = scoreEnCours;
    } else {
        //J'appelle la fonction joueurSuivant
        joueurSuivant();
    }
});

/* ***** */
/* PARTIE 4 ==> GESTION DU CLICK SUR LE BOUTON 'COMMUTEZ' */
/* ***** */
let btnCommutez = document.querySelector('.btn-commuter');
btnCommutez.addEventListener('click', () => {
    scoresGlobals[joueurActif - 1] += scoreEnCours;
    document.getElementById(`score-${joueurActif}`).textContent = scoresGlobals[joueurActif - 1];
    //J'appelle la fonction joueurSuivant
    joueurSuivant();
});

```

Notre projet fonctionne toujours. SYMPA NON ?

## GESTION DU CLIC SUR « NOUVEAU JEU »

Le bouton « NOUVEAU JEU » a pour but de relancer, il permet de débiter une nouvelle partie.

```

/* ***** */
/* PARTIE 6 ==> GESTION DU CLICK SUR LE BOUTON 'NOUVEAU JEU' */
/* ***** */
let btnNouveauJeu = document.querySelector('.btn-nouveau');
btnNouveauJeu.addEventListener('click', () => {
    //On initialise tous les composants du GUI conformément
    //à un jeu qui démarre nouvellement
    scoresGlobals = [0, 0];
    scoreEnCours = 0;
    joueurActif = 1;

    document.querySelector('.de').style.display = 'none';
    document.getElementById('score-1').textContent = '0';
    document.getElementById('score-2').textContent = '0';
    document.getElementById('encours-1').textContent = '0';
    document.getElementById('encours-2').textContent = '0';
});

```

En regardant de près le code précédent, on se rend compte qu'il est la reprise du code de la PARTIE 2 qui initialise notre jeu au lancement de l'application. Factorisons-le donc dans une fonction comme suit :

```

/* ***** */
/* PARTIE 7 ==> IMPLEMENTATION DE LA FONCTION 'initiliseLeJeu' */
/* ***** */
function initiliseLeJeu() {
    //Code d'initialisation de la PARTIE 2
    scoresGlobals = [0, 0];
    scoreEnCours = 0;
    joueurActif = 1;

    document.querySelector('.de').style.display = 'none';
    document.getElementById('score-1').textContent = '0';
    document.getElementById('score-2').textContent = '0';
    document.getElementById('encours-1').textContent = '0';
    document.getElementById('encours-2').textContent = '0';
}

```

Ce qui nous permet de réécrire la PARTIE 2 & 6 comme suit :

```

/* ***** */
/* PARTIE 2 ==> INITIALISATION DU GUI DU JEU */
/* ***** */
//On appelle la fonction d'initilisation du jeu
initiliseLeJeu();

```

```

/* ***** */
/* PARTIE 6 ==> GESTION DU CLICK SUR LE BOUTON 'NOUVEAU JEU' */
/* ***** */
let btnNouveauJeu = document.querySelector('.btn-nouveau');
btnNouveauJeu.addEventListener('click', () => {
    //On appelle la fonction d'initilisation du jeu

```

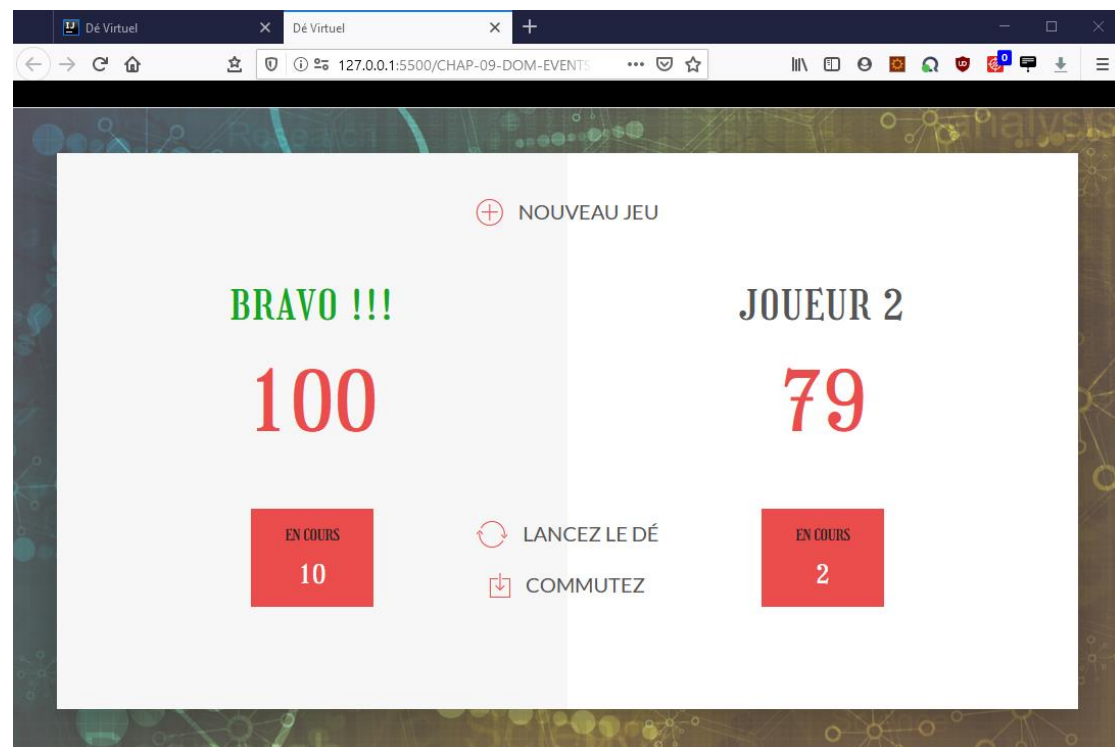
```
initiliseLeJeu();
});
```

Regardons maintenant la règle V du règlement. Pour rappel, elle nous dit que le premier à avoir un score GLOBAL supérieur ou égal à 100 gagne la partie. Qu'à cela ne tienne, nous ne manipulons le score GLOBAL qu'avec le bouton « COMMUTEZ », modifions donc le code de la PARTIE IV pour prendre en compte cette fonctionnalité.

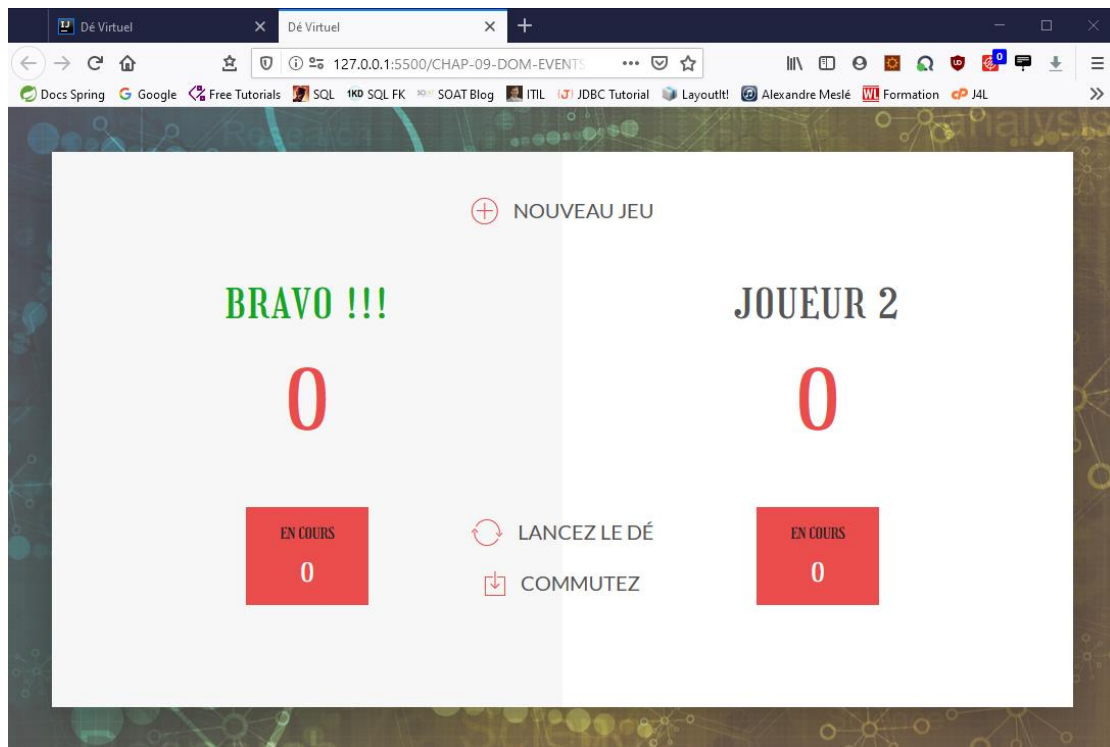
```
/* ***** */
/* PARTIE 4 ==> GESTION DU CLICK SUR LE BOUTON 'COMMUTEZ' */
/* ***** */
let btnCommutez = document.querySelector('.btn-commuter');
btnCommutez.addEventListener('click', () => {
  scoresGlobals[joueurActif - 1] += scoreEnCours;
  document.getElementById(`score-${joueurActif}`).textContent = scoresGlobals[joueurActif - 1];

  //On verifie si le joueur a atteint le score gagnant
  if (scoresGlobals[joueurActif - 1] >= 100) {
    document.getElementById(`nom-${joueurActif}`).textContent = 'Bravo !!!';
    document.querySelector('.de').style.display = 'none';
    document.querySelector(`.joueur-${joueurActif}-panel`).classList.add('vainqueur');
    document.querySelector(`.joueur-${joueurActif}-panel`).classList.remove('active');
  } else {
    //J'appelle la fonction joueurSuivant
    joueurSuivant();
  }
});
```

Et Ouuii, le résultat est le suivant !



Bien à ce niveau, nous nous rendons compte que lorsque nous cliquons sur le bouton « NOUVEAU JEU », on obtient la figure ci-dessous :



Oups. Nous devons donc revoir notre fonction d'initialisation du jeu. Je vous invite à la modifier dans la PARTIE 7 comme suit :

```

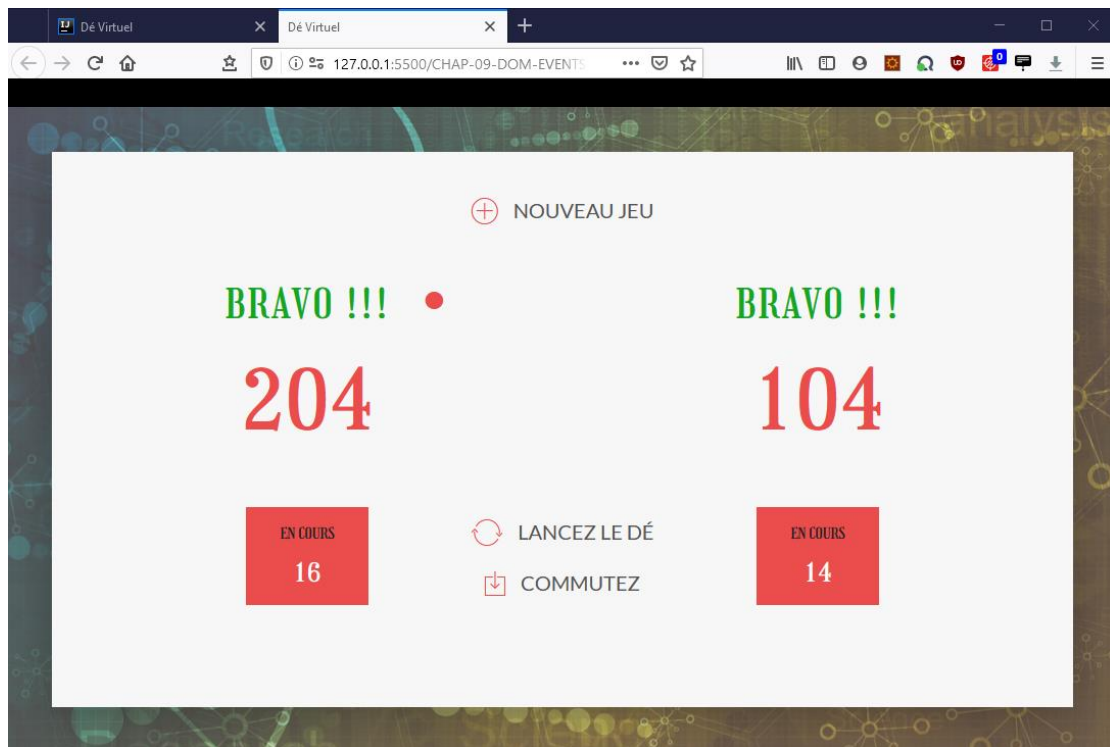
/* ***** */
/* PARTIE 7 ==> IMPLEMENTATION DE LA FONCTION 'initiliseLeJeu' */
/* ***** */
function initiliseLeJeu() {
    //Code d'initialisation de la PARTIE 2
    scoresGlobals = [0, 0];
    scoreEnCours = 0;
    joueurActif = 1;

    document.querySelector('.de').style.display = 'none';
    document.getElementById('score-1').textContent = '0';
    document.getElementById('score-2').textContent = '0';
    document.getElementById('encours-1').textContent = '0';
    document.getElementById('encours-2').textContent = '0';
    document.getElementById('nom-1').textContent = 'Joueur 1';
    document.getElementById('nom-2').textContent = 'Joueur 2';
    document.querySelector('.joueur-1-panel').classList.remove('vainqueur');
    document.querySelector('.joueur-2-panel').classList.remove('vainqueur');
    document.querySelector('.joueur-2-panel').classList.remove('active');
    document.querySelector('.joueur-1-panel').classList.add('active');
}

```

Notre jeu est désormais presque complet.

Seul hic est qu'à ce stade, même quand on a un gagnant, les boutons « LANCEZ LE DE » et « COMMUTEZ » sont toujours actifs « voir figure ci-dessous » :



Pallions rapidement à ce problème en déclarant une nouvelle variable nommée « **jouer** » que nous positionnons à la valeur « **true** » dans la PARTIE 1. Tant que cette dernière a la valeur « **true** », le jeu peu continuer.

```
/* ***** */
/* PARTIE 1 ==> DECLARATION DES VARIABLES DU PROGRAMME */
/* ***** */
let scoresGlobals = [0, 0];
let scoreEnCours = 0;
let joueurActif = 1;
let jouer = true;
```

On peut maintenant mettre les PARTIES 3 & 4 à jour comme suit :

```
/* ***** */
/* PARTIE 3 ==> GESTION DU CLICK SUR LE BOUTON 'LANCEZ LE DE' */
/* ***** */
//On cible le bouton à l'aide de sa classe
let btnLancezLeDe = document.querySelector('.btn-lancer');

//On attache l'événement de click à notre bouton avec une fonction anonyme lambda.
btnLancezLeDe.addEventListener('click', () => {
  if (jouer) {
    //Je simule le lance le Dé et encapsule le resultat dans la variable de
    let de = Math.floor(Math.random() * 6) + 1;

    //Je met à jour j'image du dé dans le GUI
    let domDe = document.querySelector('.de');
    domDe.style.display = 'block';
    domDe.src = `resources/images/de-${de}.png`;

    //On met à jour le score en cours du joueur actif sauf si le lancer vaut 1
    if (de !== 1) {
      //Ici, le lancer à sorti une valeur != 1.
      // On met donc à jour le score en cours du joueur actif
      scoreEnCours += de;
      document.getElementById(`encours-${joueurActif}`).textContent = scoreEnCours;
    } else {
      //J'appelle la fonction joueurSuivant
      joueurSuivant();
    }
  }
});

/* ***** */
/* PARTIE 4 ==> GESTION DU CLICK SUR LE BOUTON 'COMMUTEZ' */
/* ***** */
let btnCommutez = document.querySelector('.btn-commuter');
btnCommutez.addEventListener('click', () => {
  if(jouer){
```

```

scoresGlobals[joueurActif - 1] += scoreEnCours;
document.getElementById('score-${joueurActif}').textContent = scoresGlobals[joueurActif - 1];

//On verifie si le joueur a atteint le score gagnant
if (scoresGlobals[joueurActif - 1] >= 100) {
    document.getElementById('nom-${joueurActif}').textContent = 'Bravo !!!';
    document.querySelector('.de').style.display = 'none';
    document.querySelector('.joueur-${joueurActif}-panel').classList.add('vainqueur');
    document.querySelector('.joueur-${joueurActif}-panel').classList.remove('active');
    jouer = false;
} else {
    //J'appelle la fonction joueurSuivant
    joueurSuivant();
}
});

```

Ce qui achève notre exercice. BRAVO.