

## ▀ ▀ ▀ Modèle de copie :

**Dynamiser vos sites web avec Javascript**

**GDWFSDVSWEBAJAVAEXAIII1A**

**Ceci est un modèle de copie. N’oubliez pas de renseigner vos prénom/nom, ainsi que le nom et le lien vers le projet.**

**Vous pouvez bien sûr agrandir les cadres pour répondre aux questions sur la description du projet si nécessaire.**

**Prénom :**jean-Pierre

**Nom : LEDOS**

**ATTENTION ! PENSEZ À RENSEIGNER VOS NOM ET PRÉNOM DANS LE TITRE DE VOS FICHIERS / PROJETS !**

Nom du projet : dicegames

Lien Github du projet : [**jpLedos/dicegames**](https://github.com/jpLedos/dicegames)

Lien Drive du projet (si nécessaire) : na

URL du site (si vous avez mis votre projet en ligne) : <https://jpledos.github.io/dicegames/>

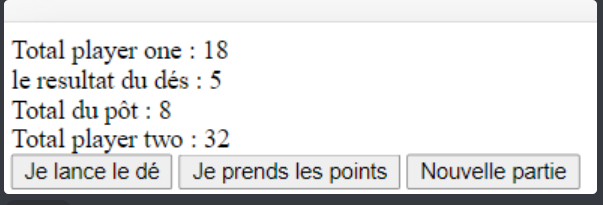
**Description du projet**

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.

J’ai commencé par étudier l’algorithme avant de procéder à l’écriture du javascript.

J’ai créé un fichier index.html a minima, avec seulement les boutons qui m’ont servi à déclencher les fonctions callback. J’ai utilisé le console.log pour afficher des résultats au fur à mesure du process et m’assurer que les réponses étaient correctes

. 

1ere etape : algorithme

2me etape : creation contantes et variable (les éléments HTML, une image pour l’affichage dynamique du dé, nombre de joueurs , round, turn pour compter les tours, result (lancé du dé, current player, et les sons.

3me etape : les fonctions ; newparty(), roolResult(), changeplayer(), refresh()..

4me etape ; creation des evenements pour lancer les fonction de callback.

1. 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 la mise en page, le HTML étant très simple, j’ai juste pris un peu du framework bootstrap pour le flexbox. J’ai également utilisé un media query reglé à (max-width: 992px) pour être en phase avec le paramètre bootstrap (-lg) qui a été utilisé. J’ai consulté régulièrement le MDN pour lever des doutes sur les propriétés ou les méthodes utilisées.

Je souhaitais mettre un peu de son avec des fichiers très léger. J’ai donc récupéré des sons système Windows. Ne sachant pas comment jouer des .wav , j’ai trouvé une solution dans stackoverflow avec plus de 500 réponses positives. J’ai pu convertir les .wav au format « base64 » sur le site <https://base64.guru/converter/encode/audio/wav>.

Ne trouvant de photos libres de droits, j’ai photographié les dés que j’ai détouré sur photoshop.

J’ai créé un reposit sur github et fait des commit régulier.

Les test sous différents navigateur , et le test responsive avec chrome étaient satisfaisant. Cependant, une fois en ligne, les test à partir du mobile ne convenait pas, les boutons étant trop petits.

J’ai donc repris quelques réglages dans le media query du CSS pour obtenir quelque chose de satisfaisant

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

NB: Pour le cas des exercices et évaluations demandées sur la plateforme Studi, il s'agit de...Studi.

Ce sujet est un sujet imposé par Studi à la fin du module javascript.

1. Informations complémentaires (*facultatif*)

RAS