

## ▀ ▀ ▀ 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 :**Mathieu

**Nom :** Henrissat

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

Nom du projet : jeu Evil Duel

Lien Github du projet : https://github.com/Henrissat/jeux\_studi

Lien Drive du projet (si nécessaire) : ………

URL du site : https://tiffanymenu-gestion.fr/STUDI/jeu\_studi/index.html

**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 dans un premier temps mis sur papier mes différents éléments que j'allais devoir créer et les différents liens les uns avec les autres. j'ai donc mis à la base une partie qu'on initialise avec 2 joueurs, pour chacun un score global et son score temporaire. J'ai mis à part le dé de 6 qui agit sur le score temporaire et les Rounds.

Une fois ce schéma créer j'ai commencé à construire mes différentes pages dont j'allais avoir besoin, à savoir : une page HTML que j'ai lié à ma page script et style. J'ai commencé à créer de manière simple la base graphique de mon jeu d'abord en HTML et CSS pour placer mes différentes DIV et les nommer avec des ID que j'utiliserais dans mon JS.

Une fois tous mes fichiers enregistrés avec la base pour ma programmation j'ai init mon projet dans la commande de mon terminal Bash , add mes fichiers et j'ai commit mes fichiers. J'ai créer mon repository sur GitHub et j'ai push mon local sur le serveur distant.

J'ai pu créer un nouvelle branch en local pour commencer à travailler cette fois sur le script. J'ai voulu commencer par créer mes players 1 et 2, mais j'ai rapidement basculé sur la problématique du lancé de dé qui me paraissait assez évident avec un Math.random. Une fois le dé le random en place et le chiffre arrondi je ne tombais pas sur la bonne suite de nombre car il me manquait un "+1" à mon calcul pour aller de 1 à 6 et non de 0 à 5. il me fallait maintenant lui apporter une apparence pour chaque nombre j'ai pensé immédiatement aux conditions if ou While puis au Switch case. La condition if me paraissait une bonne solution si j'ajoutais à ma class (show-) le nombre aléatoire (randNum) cela pourrais me faire apparaitre une face en fonction. Je voulais chercher une solution pour rendre mon dés plus vivant et le faire rouler. En fouillant sur internet, j'ai trouvé le moyen de faire tourner le dés en fonction de ma class + randNum. Pour le moment je n'ai qu'un nombre qui apparait sur les faces, je compte adapter le graphisme plus tard avec le Canvas peut-être avec un Switch Case.

à partir de la j'ai sauvegardé mon projet sur Git et GitHub et j'ai attaqué les conditions du score pour additionner les résultat si 2 à 5 ou game Over si 1 et joueur suivant. j'ai d'abord initialisé le score à 0 et testé sur mon joueur 1 si cela fonctionnais (roundScore-1). Il fallait trouver maintenant le moyen de switché d'un joueur à l'autre pour attribué le score au bon joueur roundScore-1 ou -2 et changer de joueur si 1. Pour que se soit plus simple au niveau de mon code, j'ai modifié tous les noms des class et ID des players comme étant player 0 et player 1 plutot que player 1 et 2.

J'ai alors attaqué le concept de joueur actif (playerActive) et joueur suivant (nextPlayer) et initialiser le joueur qui commençais à 0 et étant le playerActive. mon code fonctionnait mais lorsque je tombais sur un dés de 1 et qu'on passait au joueur suivant il ne repartait pas avec un round Score à 0 mais il gardait celui de son ancien tours, j'ai donc remis le roundScore à 0 dans la function nextPlayer.

Il me restait plus qu'à stocker mon score en appuyant sur le bouton Hold. j'ai commencé par créer mon tableau pour le score des mes deux joueurs et lorsque l'on clic le joueur actif reçoit le score du round et on change de joueur en relançant le funtion nextPlayer.

Après plusieurs test le jeu tourne correctement, il me fallait intégrer le but du jeu c'est à dire que l'un des scores atteignent 100 et la possibilité de relancer une nouvelle partie. J'ai créer ma function winner si l'un des scores[ ] est égal ou supérieur à 100. Le problème c'est que ma fonction cherchait ce résultat au lancement de la page et non à chaque clic sur le bouton hold. j'ai donc mis l'appel dans la function du bouton hold. J'ai testé avec un message d'alert à 100, ça roule !

J'ai ensuite cherché à dynamisé un peu le site pour le rendre plus ludique, j'ai donc ajouté dans un premier temps des bruitages pour les dés ainsi que lorsque la personne tombe sur un "1" en mettant directement dans mon html une balise <audio> que j'ai lié à mon clic et à ma function. Mais le résultat était un peu répétitif, j'ai donc prit plusieurs sons et créer une function random pour les sons et j'ai ajouter de la même façon un message qui apparait lorsque l'on perd.

J'ai ensuite Merge ma branch concernant le moteur du jeu et créer une nouvelle branch pour m'attaquer au style.

Je me suis donc attaqué à l'apparence du jeu en plaçant de manière optimisé mes différents blocs. Pour éviter de répéter plusieurs fois mon code pour les player 1 et 2 j'ai créer des balise h1 h2 h3 dans mes div mais elles n'étaient pas prises en compte, j'ai donc transformé mes div en balise h directement et cela permettait d'optimiser mon code par la même occasion. Il me fallait ensuite pouvoir basculer l'apparence du joueur en cour d'un côté à l'autre pour faire apparaitre mon graphisme. J'ai donc utilisé les sélecteurs |= et ^= pour ne prendre qu'une partie de mes noms de class et ID pour leur attribuer une apparence. Il me restait plus qu'à trouver le moyen de switcher d'un joueur à l'autre l'apparence en utilisant le activPlayer. J'ai donc utiliser le DOM et créer une class bg-active-player sur laquelle j'ai fais la bascule avec un Toggle que j'ai intégré à ma function nextPlayer.

Je souhaitais également personnaliser l'apparence de mon dés avec le canvas, mais après la création de 2 faces du dés, le code nécessaire pour la création de 6 faces et leurs intégrations demandait beaucoup plus de ressources que l'implantation d'image. J'ai donc créer mes images sur Photoshop et les aient ajoutés en css. Pour apporter une ambiance générale dynamique à mon visuel, je voulais donner une impression de brazier. J'ai donc créer des éléments avec Canvas dans une div à part sur mon fond en absolute que j'ai dupliqué et fait apparaitre de manière aléatoire. j'ai ensuite voulu tourné mon canvas avec la méthode rotate sur mon ctx pour accentuer la dynamique, mais sans succès, pour faire plus simple j'ai tourné directement ma div canvas à l'aide du css en ajoutant une div Overflow.

Pour finir, j'ai créer une version mobile à l'aide des media Query.

J'ai commit ma branch Design et je l'ai merge avec la branch principale pour finaliser mon projet.

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...

html  
css pour le visuel et Media Query pour le responsive

Javascript pour la partie dynamique

Canvas pour l'animation du fond

Grid pour le placement joueurs

Media Query pour Responsive Design

Photoshop pour le maquettage et le travail des images

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.

Studi

1. Informations complémentaires (*facultatif*)