



DÉVELOPPEMENT JAVASCRIPT PROJET D'ÉQUIPE

Noms :

Quelques règles transverses :

- Le projet se réalise en binôme ou en trinôme. Chaque groupe choisit un seul sujet.
- Les points avec des astérisques rouges * sont obligatoires, les autres sont du bonus.
- La qualité du design et l'ergonomie n'ont aucune importance.
- Un soin particulier doit être apporté à la structure du code pour qu'il soit lisible et bien organisé. Éviter les fichiers immenses et tirer profit des exports. Utiliser des classes là où c'est opportun.
- L'utilisation de dépendances n'est pas autorisée sauf dérogation.
- Le code doit-être sous GitHub.
- Chacun des développeurs doit maîtriser et comprendre l'intégralité du code écrit.
- Demandez de l'aide si vous en avez besoin !

SUJET 1 : Le démineur

Réalisez votre implémentation du jeu du démineur en JS.

- * Le joueur peut choisir un niveau de difficulté (facile, moyen, difficile) correspondant à une taille de grille et un nombre de bombes.
- * La grille se remplit aléatoirement en début de partie.
- * Le joueur peut cliquer sur une case pour la déminer.
 - Le clic sur une bombe fait perdre la partie.
 - Le clic sur une zone vide la révèle ainsi que toutes les zones vides adjacentes.
 - Le clic sur une zone portant un numéro révèle ce numéro.
- Le joueur peut utiliser un drapeau pour identifier une bombe (clic droit par exemple). Une fois une zone marquée par un drapeau elle ne peut plus être révélée à moins d'enlever le drapeau.
- * Une fois toutes les cases n'étant pas des bombes révélées, la partie est gagnée.

SUJET 2 : Motus

Réalisez votre implémentation du jeu Motus en JS.

- * Le joueur peut choisir un niveau de difficulté (facile, moyen, difficile) correspondant à des mots plus ou moins longs.
- * Le joueur dispose de 6 essais pour découvrir le mot dont il n'a que la première lettre.
- * A chaque tentative, les lettres bien placées sont mises en évidence sur fond rouge et les lettres mal placées sur fond jaune.
- * La validité des mots devra être vérifiée au moyen d'une API (par exemple <https://api.dictionaryapi.dev/api/v2/entries/en/<mot>>, en anglais).
- De petits bips viennent accompagner la découverte des lettres avec des variations selon qu'elles sont absentes, bien placées ou mal placées.
- * La découverte du mot implique la victoire, un échec à la 6ème tentative implique la défaite du joueur.

SUJET 3 : Jeu de mémoire

Réalisez votre implémentation d'un jeu de mémoire en JS.

- * Le jeu affiche à l'écran des zones de couleur.
- * Lorsque la partie commence, une zone de couleur devient brièvement plus brillante puis reprend son aspect initial. Le joueur est invité à cliquer sur cette même zone afin de reproduire la séquence.
- * A la seconde étape, le jeu ajoute une nouvelle forme à la séquence (ce peut être une seconde fois la même). De nouveau le joueur doit reproduire la séquence en cliquant successivement sur les deux zones.
- * Le jeu se poursuit, à chaque fois que l'utilisateur reproduit la séquence avec succès une nouvelle zone est ajoutée.
- * Le clic sur une zone s'accompagne d'un son distinct.
- * Lorsque le joueur se trompe dans une séquence, son score est affiché.
- Le meilleur score est toujours affiché en haut.
- L'utilisateur peut augmenter ou réduire le nombre de zones.

SUJET 4 : votre idée !

Choisissez un sujet qui vous intéresse et dans ce cas se mettre d'accord avec moi sur les « spécifications ».

SUJET 5 : Le quiz !

Implémentation d'un simple jeu de quiz. Réalisez votre implémentation d'un jeu de mémoire en JS.

- Il n'est pas nécessaire que le quiz fonctionne sur tablette ou mobile
- * Il faut pouvoir naviguer d'une question à l'autre
- * A l'issue du quiz le joueur reçoit sa note et on affiche toutes les réponses
- * Le quizz doit être alimenté via un tableau d'objets contenant les questions et les réponses.