

Exercice : Créer un jeu de mastermind avec React, NextJS et Mantine

Pour cet exercice, vous allez créer un jeu du mastermind.

Pour ceux qui ne connaissent pas :

- Le jeu du mastermind consiste en une suite de 4 couleurs générée aléatoirement. Vous aurez alors 10 essais pour essayer de deviner cette suite de couleur
- Chaque tour, vous sélectionnez une suite, et le système devra vous dire
 - Le nombre de bonnes couleurs bien placées
 - Le nombre de bonnes couleurs mal placées

Prenons un exemple :

- La génération de couleurs est **Rouge - Vert - Jaune - Rouge**
 - Votre essai est **Rouge - Rouge - Bleu - Jaune**
 - L'application devrait vous retourner :
 - 1 bonne couleur bien placée (car le rouge en 1ère position est OK)
 - 2 bonnes couleurs mal placées (car le rouge en 2ème position et le jaune en 4ème position sont bien sur le plateau mais mal placées)

Au bout de 10 essais, si vous n'avez toujours pas trouvé, afficher une notification de défaite

Si vous avez trouvé en moins de 10 essais, afficher une notification de victoire, et enregistrer le nombre d'essais utilisées dans un fichier JSON pour faire un tableau de score

Les couleurs disponible sont : **Rouge, Bleu, Vert, Jaune, Orange et Rose**

Vous devrez donc :

- Au niveau des pages :
 - Avoir une page d'accueil avec deux boutons, un pour commencer un jeu et un bouton pour voir le tableau des scores
 - Avoir une page de jeu
 - Avoir une page de tableau des scores
- Au niveau des fonctions :
 - Avoir une fonction pour générer la suite de 4 couleurs aléatoirement. Si il y a 3 fois la même couleur, relancer la fonction pour une nouvelle suite
 - Avoir une fonction permettant de vérifier la proposition du joueur par rapport à la suite de couleur générée. Cette fonction devra retourner le nombre de bonnes couleurs bien placées, ainsi que le nombre de bonnes couleurs mal placées. Si 4 bonnes couleurs sont bien placées, le joueur à gagné
- Sur la page de jeu:
 - Pour sélectionner les couleurs, utilisez des [Select](#) avec les 6 couleurs possibles.
 - Utilisez un [Formulaire](#) afin de valider le choix des couleurs, votre formulaire aura donc les clés "color1", "color2", "color3" et "color4"

- Pensez, lorsque vous validez votre formulaire, une fois votre essai ajouter à l'accordéon, etc... à faire un **form.reset()**, afin de réinitialiser les valeurs dans vos select
- Pour chacun de vos select, utiliser le paramètre "validate" de votre form, afin de dire que si la valeur de la clé est à **null**, afficher "Sélectionner une couleur" sous votre select
- Affichez les essais sur la droite dans un [Accordéon](#), du coup, à chaque essai, un item s'ajoute à votre accordéon
 - En titre, mettre "Essai 1", "Essai 2", ...
 - Dans l'accordéon, afficher les couleurs utilisées, ainsi que le nombre de bonnes couleurs bien placées et bonnes couleurs mal placées
- Pour votre tableau des scores, enregistrez les scores dans un fichier JSON, et afficher les scores dans un [Tableau](#)