

Sujet de test - ProHacktive

MODULE FRONTEND_DESKTOP
Navigateur : Google Chrome Desktop

TABLE DES MATIERES

1.	EXPLICATION DU MODULE	3
2.	INSTRUCTIONS AU COMPÉTITEUR	5
3.	CONTENU DU MODULE	5

1. EXPLICATION DU MODULE

DURÉE ESTIMÉE DU MODULE

4 heures

Dans le cadre d'un travail demandé par une agence digitale spécialisée dans les mini-jeux, votre rôle est de concevoir un mini-jeu très simple, le Tic-Tac-Toe.

Le Tic-Tac-Toe est un jeu de réflexion se pratiquant à deux joueurs au tour par tour et dont le but est de créer le premier un alignement. Le jeu se joue généralement avec papier et crayon. Il est aussi appelé « morpion ».

Exemple : <https://www.google.fr/search?q=tic%20tac%20toe>

Description du projet et tâches à suivre

Votre tâche est de construire le moteur du jeu et de relier votre algorithme aux interfaces. Les interfaces sont au nombre de deux : la première est destinée à lancer une partie entre deux joueurs ou contre une IA. La deuxième interface permet de jouer tour par tour, ainsi que de recommencer une partie. Un compteur permet de voir le temps écoulé durant une partie.

Les tâches à suivre sont les suivantes :

- Recréer le jeu Tic-Tac-Toe avec deux modes de partie : joueur vs joueur ou joueur vs IA.

Joueur 1 vs Joueur 2

- Lors de la création d'une partie, les 2 noms doivent être enregistrés le temps d'une partie.
- Tous les noms des joueurs doivent être complétés sinon un message d'erreur (*alert*) est affiché avec le message « Champ manquants ». Les 2 noms sont obligatoires pour commencer une partie.
- Quand un joueur clique sur « Jouer », on accède à l'interface de jeu (après vérification de la présence des noms).
- Dans le cas d'un joueur 1 vs joueur 2, le joueur 1 (couleur bleue – symbole croix) doit toujours commencer.
- Au premier symbole posé, le compteur de temps écoulé se lance.
- Quand le joueur 1 clique sur une case vide, une croix bleue doit remplir cette case.
- Quand le joueur 1 clique sur une case déjà remplie par un symbole, un message d'erreur (*alert*) doit être affiché avec le message « Case déjà prise. Veuillez réessayer ».
- Quand le joueur 2 clique sur une case vide, un cercle jaune doit remplir cette case.
- Quand le joueur 2 clique sur une case déjà remplie par un symbole, un message (pop-in blanche) doit afficher une erreur.
- A chaque tour dans une partie joueur 1 vs joueur 1, le nom de chaque joueur doit être renseigné après le texte « A ton tour, *PRENOM* ».
- A la fin d'une partie joueur 1 vs joueur 2, un message (pop-in blanche) félicite le joueur gagnant avec le message « Bravo ! vous avez gagné, *PRENOM* ».
- Au clic sur « Quitter la partie », les joueurs retournent sur l'interface de sélection du mode de jeu joueur 1 vs joueur 1 ou joueur vs AI et l'algorithme est remis à zéro.
- Au clic sur « Refaire une partie », les joueurs retournent sur l'interface de sélection du mode de jeu joueur 1 vs joueur 1 ou joueur vs AI et l'algorithme est remis à zéro.

Joueur vs IA

- Lors de la création d'une partie, seul le nom du joueur 1 est enregistré le temps d'une partie.
- Le nom du joueur doit être complété sinon un message d'erreur (*alert*) est affiché avec le message « **Champ manquants** ». Le nom du joueur est obligatoire pour commencer une partie. (L'IA n'a pas de nom).
- Dans le cas d'un joueur vs AI, le joueur (couleur bleue – symbole croix) doit toujours commencer.
- Au premier symbole posé, le compteur de temps écoulé se lance.
- Quand le joueur 1 clique sur une case vide, une croix bleue doit remplir cette case.
- Quand le joueur 1 clique sur une case déjà remplie par un symbole , un message d'erreur (*alert*) doit être affiché avec le message « **Case déjà prise. Veuillez réessayer** ».
- Quand l'IA joue une case parmi les cases disponibles, un cercle jaune doit remplir cette case.
- A chaque tour dans une partie joueur vs IA, « **A ton tour, PRENOM** » est renseigné quand c'est au tour du joueur.
- A chaque tour dans une partie joueur vs IA, « **A mon tour** » est renseigné quand c'est au tour de l'IA.
- A chaque tour dans une partie joueur vs IA, un délai de 1 seconde doit être mise en place avant que l'IA ne joue pour simuler la réflexion de l'IA.
- A la fin d'une partie joueur vs IA, un message (pop-in blanche) félicite le joueur avec le message « **Bravo ! vous avez gagné** » sinon l'IA avec le message « **Pas de chance, j'ai encore gagné** ».
- Au clic sur « Quitter la partie », le joueur retourne sur l'interface de sélection du mode de jeu joueur 1 vs joueur 1 ou joueur vs IA et l'algorithme est remis à zéro.
- Au clic sur « Refaire une partie », le joueur retourne sur l'interface de sélection du mode de jeu joueur 1 vs joueur 1 ou joueur vs IA et l'algorithme est remis à zéro.

Intelligence artificielle

Dans ce module, il n'est pas demandé de concevoir une intelligence artificielle capable de rivaliser avec un joueur. Le mécanisme de l'IA se doit de choisir une case aléatoire parmi les cases disponibles dans la grille.

Templates

Un projet de départ est fourni dans ce module. Il contient :

- un fichier index.html qui contient les 2 interfaces de jeu ainsi que la pop-in blanche de fin de partie.
- un fichier style.css qui contient l'ensemble du design du site.
- un répertoire Javascript qui contient un fichier index.js vide.

Javascript

Ce module teste vos compétences en Javascript et votre maîtrise du framework VueJS. Vous êtes autorisé à effectuer des changements sur les fichiers HTML et CSS. Tout le contenu actuel du fichier HTML doit être décomposé en composants VueJS réutilisables, tout en préservant la sémantique initiale du DOM.

Il est autorisé de créer plusieurs fichiers JavaScript mais l'ensemble des classes doivent être instanciées dans le fichier **index.js**.

2. INSTRUCTIONS AU COMPETITEUR

Le livrable doit être fourni dans un dépôt Git accessible en ligne. Un lien de démo en ligne est facultatif.

Vous devrez fournir un fichier **README.md** contenant toute la documentation nécessaire aux différentes étapes d'installation, de configuration, de phase de développement et de mise en production ainsi qu'à la compréhension globale du projet et de son architecture. Si vous souhaitez organiser votre documentation, vous pouvez ajouter un répertoire **docs** à la racine du projet.

Vous êtes libre du choix de l'architecture finale des répertoires et fichiers du projet.

Le projet doit pouvoir être compilé pour être déployé sur un serveur de production. Tous les fichiers seront enregistrés dans un répertoire **build**.

La documentation doit être écrite en Markdown.

3. CONTENU DU MODULE

- Projet de départ