

TP : Création d'un Jeu de Morpion en JavaScript

Contexte et objectifs

Dans ce TP, vous allez développer un jeu du morpion (tic-tac-toe) en JavaScript afin de mettre en pratique vos connaissances en manipulation du DOM et en gestion des événements. L'objectif est de réaliser, à partir d'un squelette minimaliste (HTML et CSS déjà fournis), la logique complète du jeu en JavaScript, puis d'enrichir l'application avec des fonctionnalités bonus pour améliorer l'expérience utilisateur.

Partie A : La version de base

Fichiers fournis

- **index.html**
Contient la structure de base de la page, ainsi que la grille du jeu sous forme de div ou de tableau (squelette minimaliste).
- **styles.css**
Comprend la mise en forme de la grille (couleurs, dimensions, alignements, etc.) pour que le jeu soit visuellement attractif.
- **script.js**
Fichier vide (ou contenant quelques commentaires de démarrage) dans lequel vous devez implémenter la logique JavaScript.

Objectifs fonctionnels

1. **Interaction avec la grille :**
 - Lorsque l'utilisateur clique sur une cellule, le script doit y afficher le symbole correspondant au joueur courant (X ou O).
 - Assurez-vous qu'une cellule déjà jouée ne puisse plus être modifiée.
2. **Gestion des tours :**
 - Alternner les tours entre deux joueurs.
 - Indiquer, par un message sur la page, le joueur actif.
3. **Détection des conditions de victoire et d'égalité :**
 - Vérifier, après chaque coup, si l'un des joueurs a gagné (alignement horizontal, vertical ou diagonal).

- Déclarer une égalité si la grille est pleine et qu'aucun alignement gagnant n'est détecté.

4. Réinitialisation du jeu :

- Mettre en place un bouton « **Recommencer** » qui permet de réinitialiser la grille et de lancer une nouvelle partie.

Contraintes techniques

- Utilisez les méthodes de manipulation du DOM telles que `getElementById`, `getElementsByClassName` ou `querySelector`.
 - Exploitez les écouteurs d'événements pour gérer les clics sur les cellules.
 - Organisez et commentez votre code de manière claire pour faciliter la lecture et la maintenance.
-

Partie B : Axes d'amélioration

Vous êtes encouragés à aller au-delà de la version de base en implémentant une ou plusieurs des options suivantes (vous pouvez aussi ajouter vos propres améliorations) :

1. Option 1 : Compteur de temps et score

- Implémentez un chronomètre qui mesure la durée de la partie et affiche le temps écoulé.
- Enregistrez, en utilisant par exemple le `localStorage`, le meilleur temps réalisé et affichez-le.

2. Option 2 : Animations et transitions

- Ajoutez des animations pour l'apparition des symboles dans la grille (par exemple, une transition de couleur ou un effet d'agrandissement).
- Animez l'affichage du message de victoire ou d'égalité pour dynamiser l'interface.

3. Option 3 : Mode joueur contre ordinateur

- Proposez un mode où le joueur affronte l'ordinateur.
- Implémentez une logique simple (par exemple, des coups aléatoires) ou une IA plus élaborée pour rendre le jeu intéressant.
- Permettez à l'utilisateur de choisir le mode de jeu au démarrage (PvP vs PvIA).

4. **Option 4 : Interface responsive et personnalisée**

- Rendez l'interface du jeu responsive pour une utilisation sur mobile et tablette.
 - Ajoutez un menu de configuration pour permettre, par exemple, de choisir qui commence ou de modifier certains aspects visuels.
-

Critères d'évaluation

La notation tiendra compte des aspects suivants :

1. Fonctionnalité (10 points)

- Le jeu est fonctionnel : alternance correcte des tours, détection précise des victoires et des égalités, et bouton de réinitialisation opérationnel.
- Les fonctionnalités bonus, le cas échéant, sont correctement implémentées et s'intègrent bien à la version de base.

2. Qualité du code (3 points)

- Le code est bien structuré, lisible et correctement commenté.
- Vous utilisez de manière pertinente les méthodes de manipulation du DOM et les écouteurs d'événements.
- Bonne organisation modulaire et réutilisation de fonctions.

3. Interface utilisateur (2 points)

- La mise en page est soignée et conforme au CSS fourni.
- L'ergonomie du jeu (affichage des messages, boutons, transitions) est appréciée.
- Les améliorations visuelles dans les bonus sont cohérentes et fluides.

4. Originalité et enrichissement (5 points)

- La qualité et l'originalité des axes d'amélioration implémentés.
 - La pertinence des choix techniques et créatifs apportés pour améliorer l'expérience utilisateur.
-

Livrables attendus

- **Dossier du projet**

Un dossier compressé (.zip) contenant tous les fichiers du projet : index.html, styles.css, script.js, ainsi que tous fichiers additionnels utilisés à envoyer sur Teams avant la fin de la journée de cours.

- **README**

Un fichier README (document Word, ou fichier TXT) décrivant :

- Les fonctionnalités de base réalisées.
- Les axes d'amélioration implémentés.

- **Rapport technique (1 à 2 pages)**

Un bref document expliquant :

- Votre démarche de conception.
- Les choix techniques effectués et les éventuels problèmes rencontrés.
- Une réflexion sur les axes d'amélioration et les perspectives d'évolution.

Contraintes de réalisation

- Le TP doit être réalisé en une journée. Il est important de planifier votre temps et de valider régulièrement votre code pour éviter les bugs.
- Vous pouvez travailler en binôme si vous le souhaitez.
- Respectez les consignes de codage (indentation, commentaires, bonnes pratiques).

Remarques supplémentaires

- **Encadrement** : La gestion d'événements et la manipulation du DOM sont des compétences clés à maîtriser.
- **Extensions** : Si vous avez terminé la version de base avant la fin de la journée, travaillez sur les options bonus pour maximiser votre note.
- **Originalité** : L'originalité dans les axes d'amélioration sera particulièrement valorisée (par exemple, une IA bien pensée ou des animations fluides, ou encore l'ajout de niveaux de difficulté, des améliorations graphiques, etc ...).