Documentation

Le code est organisé en plusieurs classes représentant les différents composants du jeu :

- **GameHome** est le composant d'accueil du jeu. Il affiche le titre du jeu et propose deux boutons **Jouer contre l'ordi** et **Guide de Jeu**
- **GameRules** affiche les règles du jeu. Il permet au joueur de revenir à l'accueil du jeu.
- **GameSettings** permet au joueur de configurer les paramètres de la partie avant de commencer à jouer contre l'ordinateur.
- **GameCell** représente une cellule individuelle du plateau de jeu. Chaque cellule peut être vide ou contenir le symbole d'un joueur ('X' ou 'O').
- **GameBoard** gère le plateau de jeu et la logique du Morpion. Il initialise la partie, gère le tour des joueurs, vérifie les conditions de victoire et gère le chronomètre.

GameHome

Structure HTML

Le rendu HTML du composant est le suivant :
- Un titre <h1>.
- Une image représentant le jeu.
- Deux boutons pour naviguer vers les paramètres du jeu ou le guide des règles.

Méthodes

connectedCallback(): Méthode appelée lorsque le composant est inséré dans le document DOM. Elle attache les gestionnaires d'événements aux boutons.
openRules(): Remplace le composant actuel par le composant **GameRules**.
openGameSettings(): Remplace le composant actuel par le composant **GameSettings**.
render(): Génère le contenu HTML du composant.

Événements

click sur le bouton "Guide de jeu" : appelle openRules().
click sur le bouton "Jouer contre l'Ordi" : appelle openGameSettings().

GameRules

Structure HTML
Un bouton "Retour à l'accueil".
Des sections décrivant les règles du jeu, les objectifs, le matériel nécessaire, le déroulement, les conditions de victoire et la fin de la partie.

Méthodes
connectedCallback(): Attache le gestionnaire d'événement au bouton de retour.
backToHome(): Remplace le composant actuel par le composant GameHome.
render(): Génère le contenu HTML des règles du jeu.

Événements
click sur le bouton "Retour à l'accueil" : appelle backToHome().

GameSettings

Structure HTML

Un popup contenant :
- Un bouton pour fermer les paramètres.
- Un champ de saisie pour le pseudo du joueur.
- Un menu déroulant pour le temps par tour (15 ou 30 secondes).
- Un menu déroulant pour choisir qui joue en premier (Ordi ou Joueur).
- Un bouton "Valider" pour commencer la partie.

Méthodes

connectedCallback(): Attache les gestionnaires d'événements aux boutons.
startGame(): Vérifie les entrées utilisateur, crée le composant **GameBoard** avec les attributs appropriés et remplace le composant actuel.
closeGameSettings(): Remplace le composant actuel par **GameHome**.
render(): Génère le contenu HTML du composant.

Événements

click sur le bouton "X" : appelle closeGameSettings().
click sur le bouton "Valider" : appelle startGame().

GameBoard

Attributs

pseudo: Le pseudo du joueur.
time: Le temps par tour (en secondes).
first-player: Indique qui joue en premier ('joueur' ou ‘ordi').

Propriétés

board: Tableau représentant l'état actuel du plateau (9 cellules).
currentPlayer: Le symbole du joueur actuel ('X' ou 'O').
players: Objet contenant les informations des deux joueurs (nom, symbole, temps restant).
timer: Référence au chronomètre pour le décompte du temps par tour.

Méthodes

connectedCallback(): Initialise le jeu.
initGame(): Initialise les joueurs, détermine qui commence, et lance le rendu initial.
startTimer(): Démarre le chronomètre pour le joueur actuel.
stopTimer(): Arrête le chronomètre.
editTimeForPlayers(): Met à jour l'affichage du temps restant pour chaque joueur.
handleCellClick(index): Gère le clic sur une cellule du plateau.
updateCell(index): Met à jour l'affichage de la cellule après un coup.
checkingWinner(): Vérifie si un joueur a gagné.
switchPlayer(): Change le joueur actuel et redémarre le chronomètre.
getOtherPlayer(): Retourne le symbole de l'autre joueur.
computerPlay(): Logique pour le coup de l'ordinateur.
endGame(): Termine la partie, affiche le résultat, et retourne à l'accueil après un délai.
render(): Génère le contenu HTML du composant.
renderBoard(): Génère le plateau de jeu en créant des **GameCell**.

Événements

Gestion des clics sur les cellules via **GameCell**.

GameCell

Attributs

number: Le numéro (indice) de la cellule sur le plateau (0 à 8).
value: Le symbole actuel de la cellule ('' par défaut, 'X' ou ‘O').

Méthodes

connectedCallback(): Attache le gestionnaire d'événement au clic sur la cellule.

handleClick(): Appelle la méthode handleCellClick() du parent **GameBoard** pour gérer le clic.
render(): Génère le contenu HTML de la cellule.

Propriétés

parentBoard: Référence au composant parent GameBoard pour accéder aux méthodes du plateau.

Événements

click sur la cellule : appelle **handleClick**().