

Projet Technologies du Web

Jeu Memory



Antoine ZAGO & Camilia TURKI

Sommaire :

Introduction.....page 3

Fonctionnalités implémentées.....page 4

Difficultés rencontrées.....page 7

Commentaires et suggestions.....page 8

Auto-évaluation.....page 9

GitHub.....page 9

Introduction:

Ce compte-rendu porte sur le projet *Jeu Memory* développé en HTML, CSS, Javascript et PHP. L'objectif de ce projet est de créer un site web dynamique proposant un jeu de mémoire dans lequel le joueur doit retrouver des paires de cartes identiques. Le jeu intègre également un système de score basé sur le temps écoulé et le nombre de coups effectués.

Au lancement, le programme affiche un menu composé de plusieurs boutons correspondant aux différents niveaux de difficulté, ainsi que deux autres boutons dédiés respectivement au classement et aux statistiques.

Les trois premiers boutons redirigent l'utilisateur vers une page de jeu correspondant au niveau de difficulté sélectionné. Les deux autres permettent quant à eux d'accéder à l'historique des parties jouées (statistiques) et au classement des scores enregistrés (classement).

Fonctionnalités implémentées :

-Fonctionnalité du chronomètre:

Cette partie est programmée en Javascript et son affichage est mis en place en HTML et CSS. La fonctionnalité chronomètre met en place un décompte en temps réel dans le jeu. Elle commence à deux minutes (2:00) et diminue chaque seconde jusqu'à atteindre zéro. Un bouton de démarrage est désactivé une fois le chronomètre lancé pour éviter les déclenchements multiples.

```
function displayTimer(){
    document.getElementById("timer").textContent = `Temps : ${minute.toString().padStart(2,'0')}:${seconde.toString().padStart(2,'0')}`;
}

/* Indente le temps */
function Timer(){
    if (minute === 0 && seconde === 0) {
        stopTimer();
        return;
    }
    if (seconde == 0) {
        seconde = 60;
        minute--;
    }
    seconde--;
    displayTimer();
}

/* En combinaison de Timer() et setInterval() permet de mettre en place un décompte toute les secondes */
function ongoingTimer(){
    time = setInterval(Timer,1000); /* (function,delay) */
}

/* Commence le timer et désactive le bouton pour éviter le multiple lancement de ongoingTimer() */
function startTimer(){
    ongoingTimer();
    document.getElementById("start").disabled = true;
}

/* Arrête le timer et réactive le bouton */
function stopTimer(){
    clearInterval(time);
    document.getElementById("start").disabled = false;
}
```

-Fonctionnalité compteur de coups:

La fonctionnalité compteur de coups permet de suivre le nombre de tentatives effectuées par le joueur pour retrouver les paires de cartes. À chaque fois que le joueur retourne deux cartes, le compteur s'incrémente de un, qu'il s'agisse d'une paire correcte ou non. Ce compteur est mis à jour en temps réel sur la page. Il est réinitialisé à chaque nouvelle partie.

```
/* Fonction vérifiant si c'est une paire / identifie le nombre de coups */
function is_paire(){
    if (compare[0].dataset.valeur == compare[1].dataset.valeur) { // Vérifie si les deux éléments du tableau compare son similaire
        for(var i = 0; i <= 1; i++){
            compare[i].classList.add("paire"); // ajoute à la carte la classe paire pour éviter de pouvoir la retourner une fois la paire correcte
        }
        paire--;
    } else {
        for(var i = 0; i <= 1; i++){
            compare[i].querySelector(".carte").classList.toggle("retourne"); // On retourne les deux cartes qui ne sont pas une paire
        }
    }
    compare = []; // On réinitialise le tableau
    isClicable = true; // On redonne la capacité de cliquer à l'utilisateur
    coups++; // On ajoute un coup quelque soit le résultat
    document.getElementById("score").textContent = `Nombre de coup : ${coups}`; // On actualise le compteur de coups
}
```

-Fonctionnalité de mélange aléatoire:

La fonctionnalité de mélange aléatoire utilise l'algorithme de Fisher-Yates pour mélanger les cartes de manière équitable et imprévisible à chaque partie. Cela garantit que chaque nouvelle partie commence avec une disposition unique des cartes, rendant le jeu plus intéressant et moins prévisible.

```
// Fonction pour mélanger un tableau (algorithme de Fisher-Yates)
function melangerCartes(cartes) {
    for (let i = cartes.length - 1; i > 0; i--) {
        let j = Math.floor(Math.random() * (i + 1));
        [cartes[i], cartes[j]] = [cartes[j], cartes[i]]; // Échange des éléments
    }
    return cartes;
}
```

```
function shuffle(carteX){
    carteX = melangerCartes(carteX);
    const terrain = document.querySelector(".terrain");
    carteX.forEach(({ valeur, src }) => {
        const carte = `
            <div class="cartes" data-valeur="${valeur}">
                <div class="carte">
                    <div class="carte-dos face-arriere">
                        
                    </div>
                    <div class="carte-face face-devant">
                        
                    </div>
                </div>
            </div>
        `;
        terrain.innerHTML += carte;
    });
}
```

-Fonctionnalité d'affichage du classement:

Cette fonctionnalité, développée en PHP, permet d'afficher un tableau récapitulatif des résultats des joueurs au sein d'une structure HTML. Les données affichées sont réparties selon trois niveaux de difficulté (facile, moyen et difficile), chacun étant représenté par un tableau distinct. Pour chaque niveau, les résultats incluent le pseudo du joueur, le score, le temps restant, la difficulté sélectionnée et la date de la partie. L'ensemble est présenté de manière claire et structurée afin de faciliter la lecture et la comparaison des performances.

-Fonctionnalité d'affichage des statistiques:

Cette fonctionnalité, développée en PHP, permet d'afficher l'historique des parties jouées sous forme d'un tableau HTML. La page affiche l'ensemble des parties enregistrées, en ajoutant également une colonne pour le pseudo du joueur.

Le joueur peut également renseigner son pseudo et ainsi récupérer le résultat (victoire ou défaite) de toutes les parties jouées, le score, le temps restant, le niveau de difficulté et la date de la partie.

Difficultés Rencontrées et Solutions Apportées

Pendant la phase de conception du jeu, un problème est survenu concernant l'organisation des cartes sur le terrain de jeu. La disposition n'était pas claire, surtout lorsque le nombre de cartes variait selon le niveau de difficulté.

Pour résoudre ce problème, une grille CSS (grid) a été mise en place afin de répartir les cartes en colonnes et en lignes. Cette solution a permis d'assurer un affichage équilibré, en s'adaptant aux niveaux et donc aux nombre de cartes à afficher.

Le problème majeur qui a été rencontré est le système de sauvegarde dans une base de données SQL en utilisant le langage PHP.

En effet on cherche à sauvegarder le pseudo du joueur, le score, le temps restant, la difficulté sélectionnée et la date de la partie à chaque fin d'exercice dans une base de données et l'afficher dans la page classement et statistiques.

La difficulté résidait dans la transmission correcte de ces données du JavaScript vers le PHP, leur traitement sécurisé, puis leur insertion fiable dans la base. Ce système était essentiel pour permettre l'affichage des classements et des statistiques.

Pour résoudre ce problème, les données comme le pseudo, le score, le temps restant, la difficulté choisie et la date de la partie sont envoyées en POST depuis le front-end, puis traitées côté PHP avant d'être insérées dans la base de données. Par la suite, d'autres requêtes PHP permettent de récupérer ces informations, triées selon le niveau de difficulté ou le joueur, afin de les afficher clairement dans les pages de classement et de statistiques. Cette méthode garantit une présentation lisible et organisée pour l'utilisateur.

Commentaires et Suggestions

Remarques générales :

Les consignes du projet étaient bien structurées et permettaient une répartition efficace entre les différents membres du binôme.

Suggestions d'amélioration :

Puisque le cours n'englobe pas l'intégralité des éléments du projet, ce dernier demande de s'aider de la documentation internet pour pouvoir être réalisé.

Focus sur les Apprentissages :

Le projet nous a appris les bases en HTML, CSS et en Javascript ainsi que la gestion de données en PHP. C'est une bonne première approche du développement des langages de technologies du web.

Auto-évaluation

Les consignes de base du projet ont été respectées et le projet répond aux attentes.

Chaque membre du binôme a réalisé sa partie du code.

Code et GitHub

Lien vers le dépôt GitHub : https://github.com/ZagoAnt/Memory_web