DIW JS - Atelier Back

Présentation

Cet atelier consistera dans la réalisation d'un **jeu multi joueur en temps réel**.

Pour ce faire, vous utiliserez les notions acquises en cours, tant du côté client que du côté serveur.

Description

Principe

Vous devez réaliser un jeu en JavaScript. Ce jeu sera multi-joueur, c'est-à-dire qu'il sera possible à, au-moins, deux joueurs de s'affronter dans la **même partie en même temps**. Certaines données devront être **conservées** en base de données.

Thématique du jeu

Le thème du jeu est **entièrement libre**. Il est possible, éventuellement, de conserver le jeu réalisé précédemment si celui-ci est adapté à la participation simultanée de plusieurs joueurs.

Obligations techniques

Hébergement

Le site sera hébergé sur un serveur en ligne accessible à partir d'une connexion internet.

Ergonomie et graphisme

Le jeu doit être intégré dans un site.

Le jeu doit proposer aux joueurs d'**interagir avec l'affichage** du navigateur en fonction d'actions effectuées à l'aide du clavier et/ou de la souris.

Chaque joueur est représenté **sous la forme d'un avatar**. Un avatar est un élément graphique représentant le joueur.

Avant de commencer une partie du jeu, un joueur devra saisir, au moins, son **pseudonyme** de joueur. Vous prévoirez donc un **formulaire de saisie**. Après

la **validation de la saisie**, le joueur pourra accéder à la partie. **Attention**, un joueur qui :

ne saisit pas de pseudonyme ;

saisit un **pseudonyme déjà pris** par un autre joueur ;

ne doit pas pouvoir accéder à une partie et se voit proposer une nouvelle fois de saisir son pseudonyme.

Une partie du jeu doit se dérouler sans qu'**aucun rechargement de page** n'ait lieu. Les modifications de l'état du jeu liées aux actions de chaque joueur doivent se répercuter (presque) instantanément sur les affichages de chacun des joueurs.

Les pseudonymes de chaque joueur participant à une partie **doivent s'afficher** à côté de leurs avatars.

Le jeu contient un **mécanisme de score**. Le score de chaque joueur est enregistré en cours ou en fin de partie. Avant ou pendant une partie, un joueur doit pouvoir consulter la liste des joueurs qui ont participé au jeu, leur durée de jeu et leur score.

Les parties de jeu se déroulent en **temps réel.**

Gestion du temps réel

Pour cette mise en œuvre, deux techniques sont possibles :

Méthode Ajax

Chaque fois qu'un joueur réalise une action, son navigateur envoie les informations relatives à cette action au serveur qui l'envoie à son tour au système de gestion de base de données où elle est enregistrée.

A intervalle de temps régulier, chaque navigateur de chaque joueur interroge le serveur qui interroge le système de gestion de base de données pour obtenir la liste des dernières actions réalisées par les autres joueurs. Le serveur renvoie ensuite ces informations aux navigateurs de chaque joueur qui procèdent à la mise à jour de leur affichage.

Utilisation des Web Sockets

Une connexion persistante est démarrée entre le navigateur d'un joueur et le serveur et ce pour chaque joueur. Lorsqu'un joueur réalise une action, les informations associées sont envoyées au serveur. Lorsque le serveur reçoit des informations de la part d'un joueur, il peut les envoyer au système de gestion de base de données pour les enregistrer et/ou avertir immédiatement les navigateurs des autres joueurs qui procèdent à la mise à jour de leurs affichages.

Conservation des données

Le serveur conservera au moins l'identité et le score de chaque joueur. Un

joueur doit pouvoir se connecter au jeu depuis n'importe quel ordinateur et afficher la liste des scores de tous les joueurs.

Outils de développement

Les technologies préconisées pour la réalisation de l'application sont le HTML5, CSS3 et le JavaScript. Les pages seront valides W3C.

La partie serveur de l'application sera obligatoirement réalisée à l'aide de Node JS. Vous pourrez vous aider :

- si vous optez pour la méthode AJAX, d'Express JS.
- si vous optez pour les Web Sockets, de MeteorJS, Socket.io, ou autre.

La partie client de l'application sera obligatoirement réalisée en HTML5, CSS3 et JavaScript. Vous pourrez vous aider :

- si vous optez pour la méthode AJAX, de jQuery, AngularJS, ou autre.
- si vous optez pour les Web Sockets, de MeteorJS, de Socket.io ou autre.

Même si la gestion des collisions, dans le jeu, n'est pas obligatoire vous devrez gérer les limites de l'espace de jeu.

Compatibilité

Le jeu sera entièrement fonctionnel sur tous les navigateurs compatibles HTML5, c'est-à-dire à partir des versions suivantes des navigateurs : Internet Explorer 9, Firefox 5, Chrome 10, Safari 5 et Opera 9.

La compatibilité avec les appareils mobiles n'est pas requise.

Recommandations

Vous veillerez à bien organiser votre code pour améliorer sa présentation (indentation, sauts de ligne, choix de noms explicites). Pour ce faire, vous soignerez particulièrement les commentaires.

Vous n'hésiterez pas à scinder votre code en plusieurs fichiers.

Préalablement à la réalisation du site, vous élaborerez l'organisation de votre code.

En JavaScript, privilégiez l'usage d'objets plutôt que de nombreuses variables. Vous limiterez l'usage des variables globales en usant préférentiellement de variables locales.

Vous déboguerez votre code par des tests unitaires au cours du développement. Vous envisagerez, également, toutes les possibilités qui pourront se présenter durant le déroulement du jeu.