

Compte Rendu

Au cours de ce projet, nous avons eu pour objectif de créer une application web permettant de communiquer, à travers des salles privées ou publiques, grâce à la webcam, au transfert de fichiers, et au chat inclut. Ce compte-rendu ne tend pas à être exhaustif, mais à présenter rapidement notre répartition du travail, les technologies utilisées, faire un point ce qui est fini au jour d'aujourd'hui, et finir par ce qui pourrait être fait avec plus de temps. Il est à noter que notre objectif a été tout au long du projet de simplifier au maximum les actions requises pour le client, quitte à complexifier la programmation de certaines fonctionnalités.

Répartition du travail

Une application web peut se découper en trois parties : d'une part le côté serveur (back-end), avec notamment les requêtes sur la base de données, ainsi que la persistance et la récupération des données ; le côté client (front-end), comprenant l'ergonomie, l'accessibilité, et l'adaptabilité du site web face aux différentes fonctionnalités proposées ; et les communications, entre le client et le serveur mais aussi entre plusieurs clients. Etant trois membres dans le groupe, nous nous sommes naturellement répartis sur les points cités plus haut, tout en gardant tout de même une adaptabilité :

Geneviève Cirera - Server side

Morgane Perron - Front side

Benjamin Vella - Constructeur, liaison entre le Front et le Back

Technologies utilisées

Nous avons utilisé les technologies de base du web, HTML, CSS et JavaScript, couplées avec NodeJS permettant de programmer plus facilement. Pour la gestion à la base de données, nous avons choisi MongoDB, vu en cours, car il convient parfaitement à nos besoins et attentes.

Pour le design, nous n'avons pas ressenti le besoin d'utiliser un framework particulier pour ce projet, étant donné qu'il s'agit d'un "one-page website", avec donc peu de cas complexes et de répétitions dans les éléments.

Pour la programmation en elle-même, et notamment la distribution du code, nous avons utilisé le gestionnaire de version git et le dépôt Github "<https://github.com/mperron06/MultiChat.git>".

Après avoir travaillé sur la branche principale (l'équivalent de la production) pendant un moment, nous avons décidé de créer plusieurs branches, notamment "dev" pour la mise en

commun de nos développements. D'autres ont été créées en local afin de faciliter la collaboration.

Objectifs atteints

Tout d'abord, la base des objectifs, une application web fonctionnelle. Cette application gère les salles (de manière persistante) afin de séparer les utilisateurs. Ces salles peuvent être privées, protégées par mot de passe pour accéder à la salle et un mot de passe administrateur qui donne la possibilité de modération par bannissement; ou publiques à "entrée libre". A l'intérieur de chaque salle, nous retrouvons l'historique complet des actions qui ont été effectuées dans la salle depuis sa création, la liste des fichiers (téléchargeables) partagés par tous les utilisateurs aux autres membres de la salle grâce à une zone de drag'n'drop, un chat permettant de communiquer peu importe le nombre de clients connectés, ainsi que la webcam pour les 2 premiers arrivants (voir améliorations). Par ailleurs, une Google maps est présente dans la salle, et contient la position de tous les membres de la salle.

Problèmes rencontrés

Node.js n'ayant été utilisé par personne dans le groupe, la maîtrise de son fonctionnement a été un des plus gros problèmes rencontrés, tant pour le server-side que pour les communications, reposant dessus. Plus précisément, le système de sockets fut compliqué à appréhender et maîtriser, notamment avec mongodb et ses requêtes asynchrones.

Du côté client, la gestion des vidéos des clients a été difficile à mettre en place, et plus précisément la généralisation des connexions p2p à plus de 2 clients.

Enfin, le design et la disposition des éléments pour les rendre aisément utilisables par l'utilisateur a été recommencé plusieurs fois afin d'avoir un rendu satisfaisant. Ce n'est pas une difficulté unique et inhérente à ce projet, mais néanmoins nous a coûté assez de temps et nous n'avons pas ressenti le besoin d'utiliser un framework particulier, ni JS ni CSS (ni HTML d'ailleurs) efforts pour figurer dans cette partie.

Améliorations et évolutions possibles

Un projet web ne se finit pas, il y a toujours des modifications et des améliorations à apporter, et de nouvelles fonctionnalités à inclure.

Par rapport aux demandes brutes, le N-N est fait en grande partie : le chat, le partage de fichiers et de position, sont fonctionnels. Pour le partage vidéo, le CSS est mis en place, et on

peut aussi simuler les connections (en appelant `localCall()` via la console). Cependant, il faudrait terminer la récupération et le partage des vidéos en N-N.

De manière plus générale, on pourrait imaginer permettre à l'utilisateur de déplacer l'historique, le chat, les fichiers, à sa guise sur la page et enregistrer sa disposition préférée (via cookies ou en permanent par l'IP). Il pourrait être aussi intéressant de fusionner des salles, et d'augmenter les possibilités de la modération. C'est la dessus que nous nous serions penché si la durée du projet nous l'avait permis.