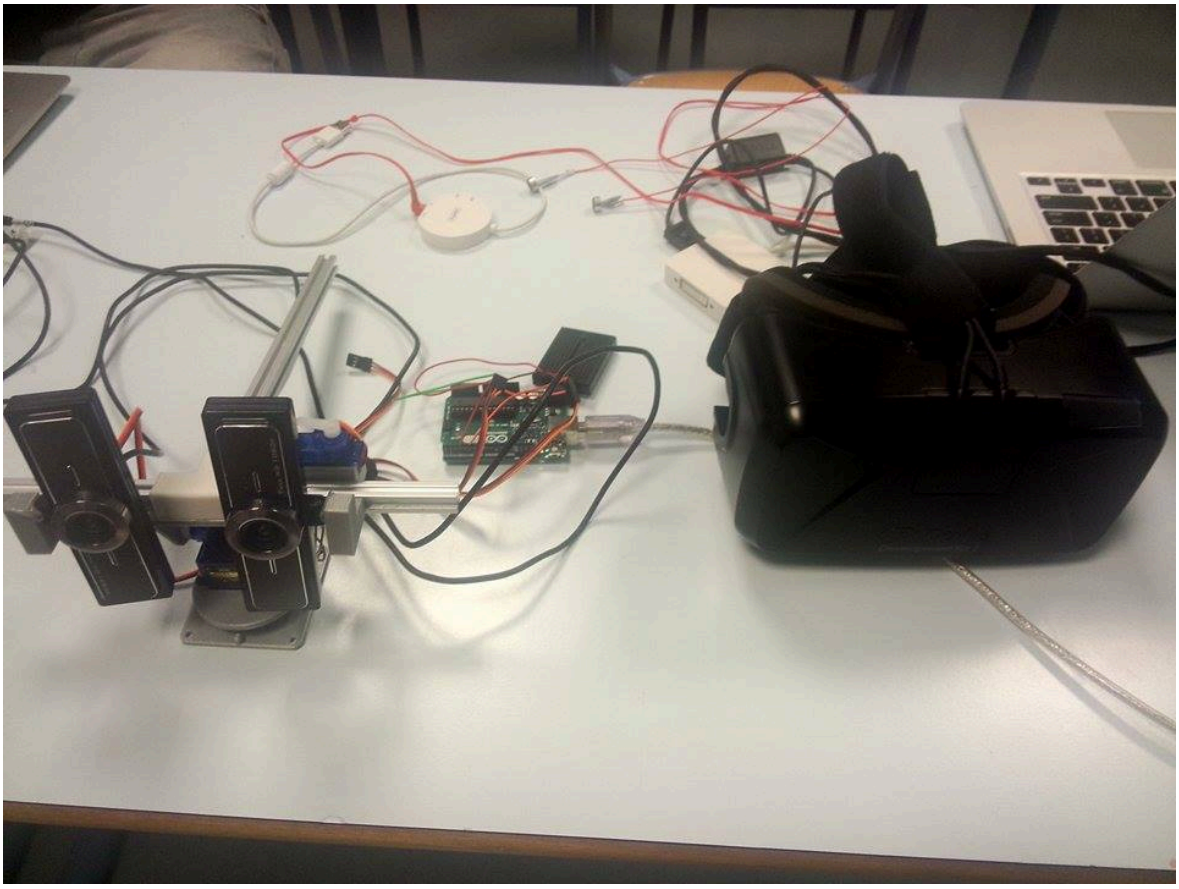
Zilong ZHAO
Guillaume HAMMOUTI

POLYTECH GRENOBLE

Projet RICM
Streaming en stéréoscopie

(Années 2015-2016)



Sommaire

Remerciements :

Nous tenons à remercier dans un premier temps, toute l'équipe pédagogique de Polytech Grenoble responsable de la formation RICM pour leur aide.
Nous tenons à remercier tout particulièrement Jérôme Maisonnasse pour nous avoir donné l'opportunité de travailler sur ce projet et également pour tout le temps et toute l'aide qu'il nous a apportée tout au long de ce projet.
Enfin nous remercions également FabLab pour nous avoir fournis tout le matériel nécessaire à la réalisation du projet.

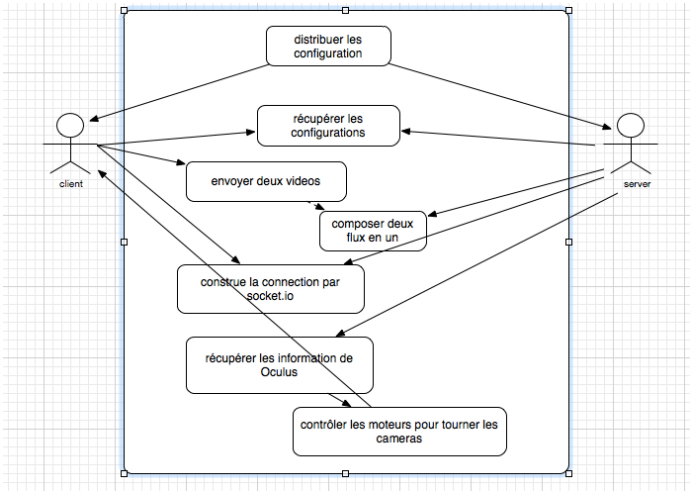
Présentation du projet :

Objectifs

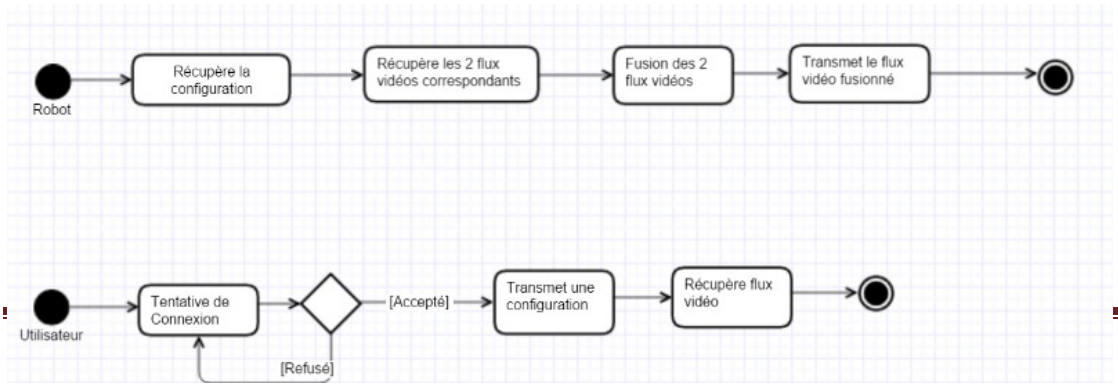
L'objectif principal de notre projet était de pouvoir envoyer un flux contenant le flux de 2 caméras entre 2 navigateurs puis de projeter ce flux dans l'oculus rift afin d'avoir un rendu 3D. En effet notre projet va venir s'incorporer dans un autre projet sur lequel travaillent d'autres étudiants, RobAir. Plus précisément, les 2 caméras représenteront les yeux du robot et l'utilisateur pourra voir en 3D ce que vois le robot.
Notre projet comportait également plusieurs objectifs secondaires. Le premier était de pouvoir contrôler les caméras, uniquement les mouvements, à l'aide de l'oculus rift. Par exemple si l'utilisateur tourne la tête à droite, le mécanisme soutenant les 2 caméras doit en faire de même. Le deuxième objectif fut d'incorporer 2 micros à notre dispositif et de faire en sorte d'obtenir un son binaural sur l'ordinateur de sortie.
Nous avons donc développé une interface web entièrement en JavaScript pour répondre à ces exigences.

UML :

- **Diagramme de cas d'utilisation**



- **Diagramme d'activité**



Matériel et Technologies :

Matériel :

- Oculus Rift DK2
- 2 caméras
- 4 servos
- 1 Arduino uno
- 2 micros

Logiciel

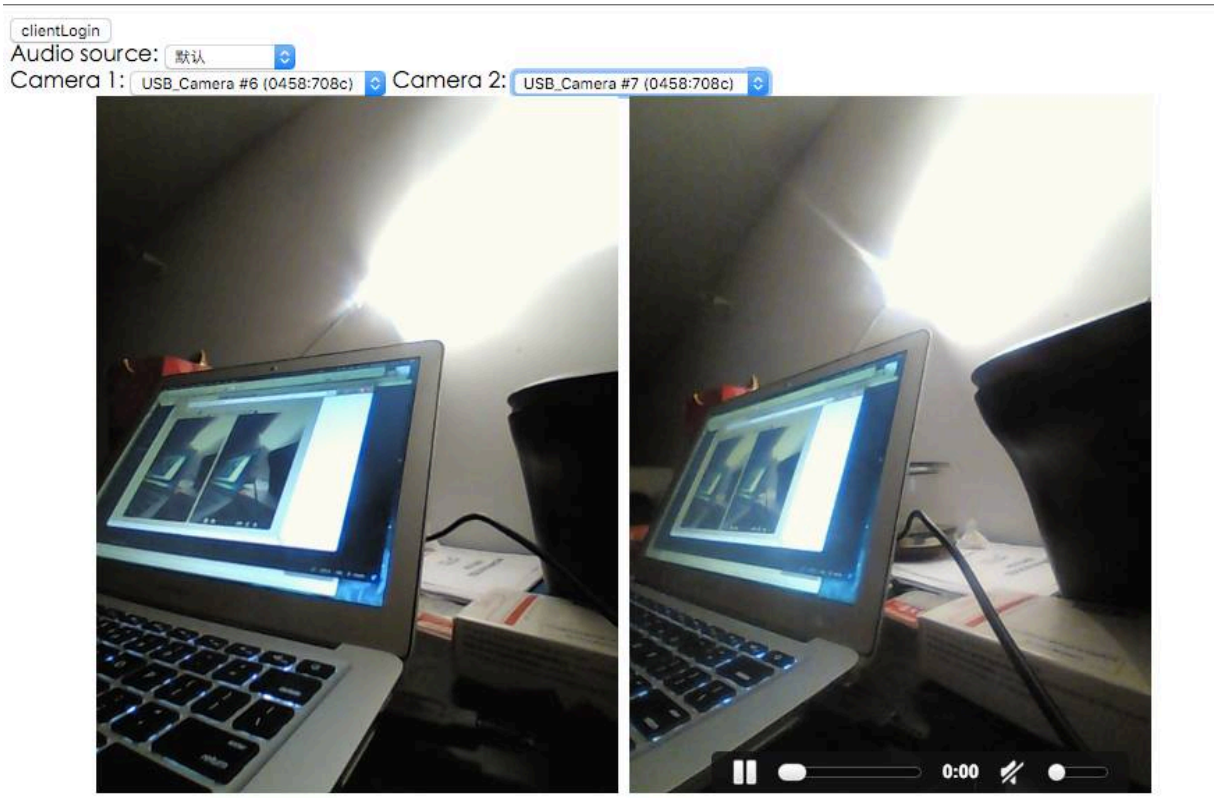
- OS X : Une bonne partie du projet a été développé sur MAC en raison des nombreux problèmes de compatibilité avec entre oculus et win10
- IDE : Eclipse

Technologie

- GitHub : Contrôler la version du programme
- WebRTC : L'envoie des flux vidéo et audio
- NodeJS : Lancement d'un server websocket, au début du projet, avant que les deux côtés construisent un lien P2P, ils ont besoin d'échanger les configurations.
- Socket.io : Pour envoyer les informations de rotations de Oculus à Arduino par internet, l'objet de Cylon.js ne supporte que Socket.io.
- Cylon.js : Une librairie qui fournit les fonctions pour contrôler Arduino
- Arduino : Une carte matériellement avec microcontrôleur qui peut piloter les servos
- Servo : Moteur qui est utilisé pour tourner nos cameras

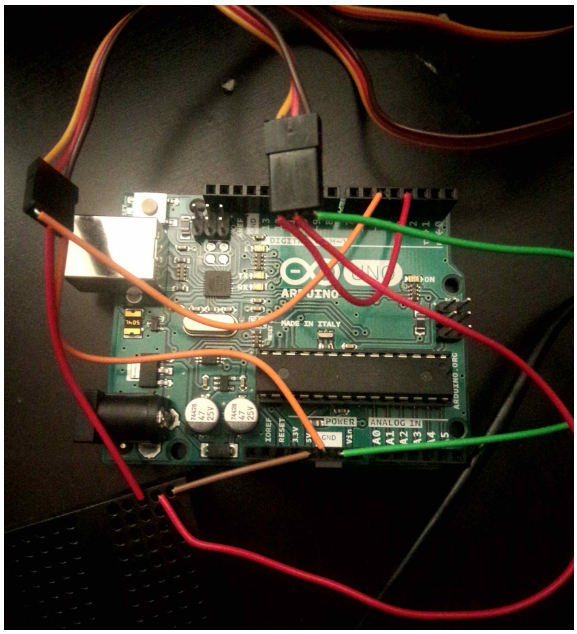
Développement :

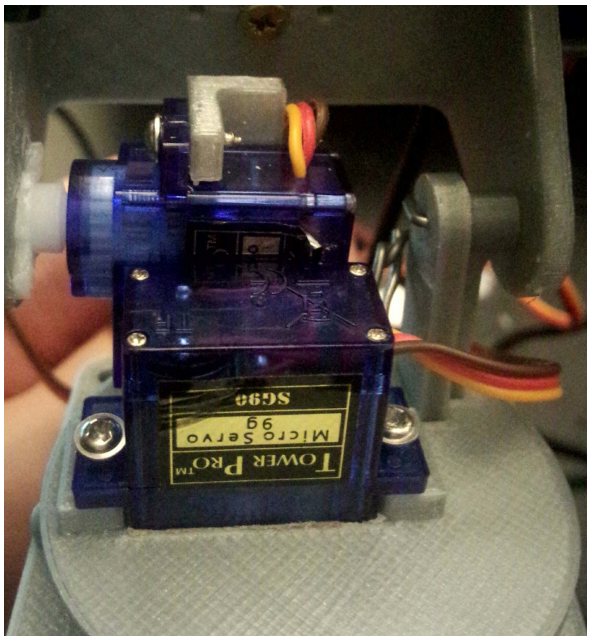
Nous avons commencé par développer une page web qui permet de détecter toutes les caméras et tous les micros branchés sur l'ordinateur. Ensuite à l'aide de plusieurs ComboBox nous avons implémenté une fonctionnalité permettant à l'utilisateur de choisir le flux de quelles caméras et de quel micro il voulait envoyer. Nous envoyons 2 vidéos tout simplement parce que l'oculus rift a besoin de 2 images pour en produire une en 3D. La distance et l'inclinaison entre les caméras est relativement importantes si l'on souhaite avoir quelque chose de bien et non pas avoir une image dans un œil et une autre dans l'autre œil.



Pour avoir un son en stéréo, on n'est obligé d'utiliser Chrome. Pendant que le serveur et le client échangent les configurations, il faut que l'on modifie les sessions pour avoir un vrai audio en stéréo.

Ensuite pour piloter les servos, on a choisi de programmer dans un Arduino uno à l'aide de cylon.js. Pour communiquer entre Oculus et Arduino, on utilise socket.io, il peut écouter dans un port, puis on envoie les données sur ce port par un socket, et Arduino peut utiliser ces données. Cylon.js fourni les interfaces pour accéder Arduino et piloter les servos, donc maintenant, on connecte juste deux servos pour tourner à gauche ou droit, plus haut et plus bas.

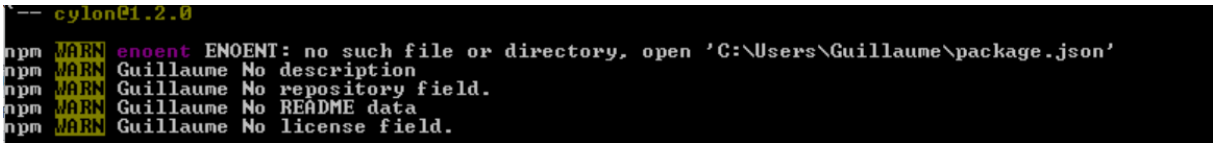




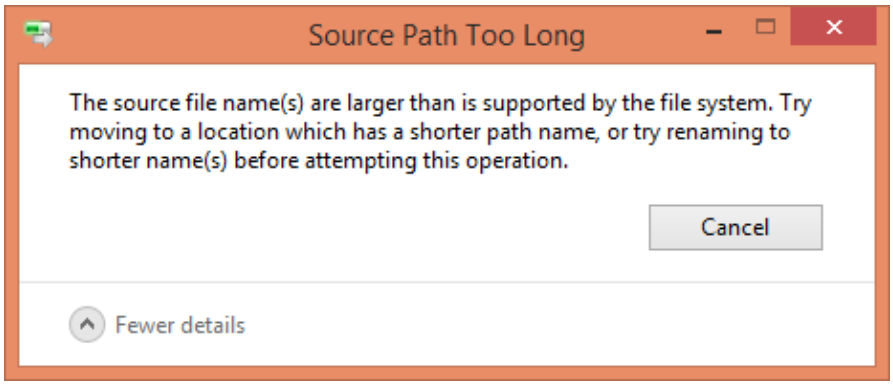
Problèmes rencontrés :

La plupart des problèmes rencontrés, si ce n'est tous, sont tous venus à cause de l'utilisation de Windows et tout particulièrement Windows 10. En effet l'Oculus Rift est très mal supporté par Windows 10. Ce qui nous a pas mal ralenti étant donné qu'une seule personne sur les 2 possédait un Mac, seule cette personne était en mesure de réaliser les tests.

Nous avons rencontré le même type de problème avec Cylon.js qui ne voulait pas s'installer sous windows (et ubuntu).



Mais également avec Node js qui a tendance à créer trop de dossier dans d'autres dossiers ce qui a pour conséquence de faire planter le système de gestion de fichiers de Windows.



Conclusion

En conclusion nous dirons que ce projet n'a pas été évident étant donné que nous n'avons pas eu le moindre cours de JavaScript. Cependant nous sommes relativement satisfaits du travail que l'on a achevé dans les délais qui nous avaient été imparti. Comme nous avions réussi à atteindre l'objectif principal au bout de quelques semaines Jérôme Maisonnasse nous a donné les 2 autres objectifs secondaires afin de pouvoir aller encore plus loin dans ce projet. Ces 2 objectifs ont été plaisants à réaliser étant donné qu'avec Mr Maisonnasse nous avons pu utiliser l'imprimante 3D afin de fabriquer le support pour le maintien des caméras. Nous sommes toutefois un peu déçus car nous ne pourrons pas voir si le prototype que nous avons réalisé sera un jour utilisé sur RobAir.

Concernant les possibilités d'évolution de notre projet, on pourrait imaginer un support qui ressemblerait davantage à une tête afin qu'elle puisse être greffée sur RobAir. Ou encore l'utilisation d'un micro 3D afin d'avoir un rendu sonore meilleur. On pourrait également développer un mécanisme permettant à l'utilisateur de pouvoir contrôler le focus des 2 caméras et pourquoi pas un zoom. Actuellement la communication vocale ne se fait que dans un sens, RobAir vers utilisateurs, donc on pourrait également ajouter des enceintes et développer une application permettant la communication dans l'autre sens pour pouvoir communiquer avec les personnes autour de RobAir.