



HEPIA 2014-2015

Rapport Projet de Semestre

Projet Vertigo : Projet d'immersion dans une réalité virtuelle pour aider à vaincre la peur du vide.

Il s'agit d'utiliser les Google CardBoard et un smartphone pour réaliser un casque de réalité virtuelle. Le monde virtuel en 3 dimensions sera affiché avec la technologie web X3DOM.

Un utilisateur doit pouvoir utiliser le dispositif pour s'immerger dans un monde virtuel dans lequel celui-ci doit se déplacer sur une planche au-dessus du vide reliant deux immeubles entre eux.

La Kinect v1 ou v2 de Windows doit être utilisé pour récupérer les mouvements et les déplacements de la personne sur la planche pour les répercuter sur le personnage représentant l'utilisateur dans le monde virtuel. La Kinect doit être branchée sur un ordinateur qui fera office de serveur pour le smartphone et qui enverra les données récupérées par cette première.

En outre, l'étudiant doit trouver les solutions pour que l'immersion de l'utilisateur dans le monde virtuel soit la plus efficace possible.

Table des matières

Informations	3
Projet	3
Description détaillée	3
Schéma projet	4
Organisation projet	4
Hébergement	4
Modèle de la vue	5
Communication entre les composants.....	5
Google CardBoard	5
Description	5
Utilité dans le projet.....	6
Etudes, recherches & tests.....	6
Montage	6
Tests.....	6
X3DOM	7
Description	7
Utilité dans le projet.....	7
Etudes, recherches & tests.....	7
Bilan.....	7
Capteurs de mouvements	7
Description	7
Utilité dans le projet.....	7
Etudes, recherches & tests.....	7
Types.....	8
Balle qui saute	8
Bilan	9
Github.....	9
Description	9
Utilité dans le projet.....	9
WebHook sur Github.....	9
Description	9
Utilité dans le projet.....	9
Bilan.....	9

WebSocket (NodeJS & MeteorJS).....	10
Description	10
WebSocket.....	10
NodeJS	10
MeteorJS.....	10
Utilité dans le projet.....	10
Etudes, recherches & tests.....	11
NodeJS	11
MeteorJS.....	11
Bilan	11
Kinect.....	11
Description	11
Utilité dans le projet.....	11
Etudes, recherches & tests.....	11
Kinect v1 Prérequis.....	11
Kinect v2 Prérequis.....	12
API.....	12
Détection d'une personne.....	12
Bilan	12
Conclusion	13
Suivi du projet	13
Installation & Utilisation.....	13
Continuité.....	13
Bilan du projet de semestre	13
Annexes	14
Liens utiles	14

Informations

Domingos Ana Sofia : ana-sofia.domingos@etu.hesge.ch

Etudiante ITI travaillant sur le projet Vertigo en travail de semestre

Gobet Jérémy : jeremy.gobet.72@gmail.com

Assistant ITI ayant proposé le sujet en travail de semestre

Malandain Stephane : stephane.malandain@hesge.ch

Professeur ITI responsable de ce travail de semestre

Albuquerque Paul : paul.albuquerque@hesge.ch

Professeur ITI responsable de ce travail de semestre

Projet¹

Discipline / Domaine : Développement Web & 3D

Le but de ce projet est d'utiliser les Google CardBoard et d'utiliser X3DOM pour modéliser une vue en 3D sur une page Web.

Les Google CardBoard sont un boîtier dans lequel on peut disposer un smartphone et s'en servir pour constituer une paire de lunette 3D de réalité virtuelle

Le projet que j'ai décidé de réaliser, avec l'assistant Jérémy Gobet, est l'implémentation d'un exercice qui permet de vaincre la peur du vide.

Je vais pour cela, modéliser la vue d'une planche de 3 mètres reliant 2 immeubles sur laquelle les personnes devront traverser pour arriver à l'immeuble d'en face.

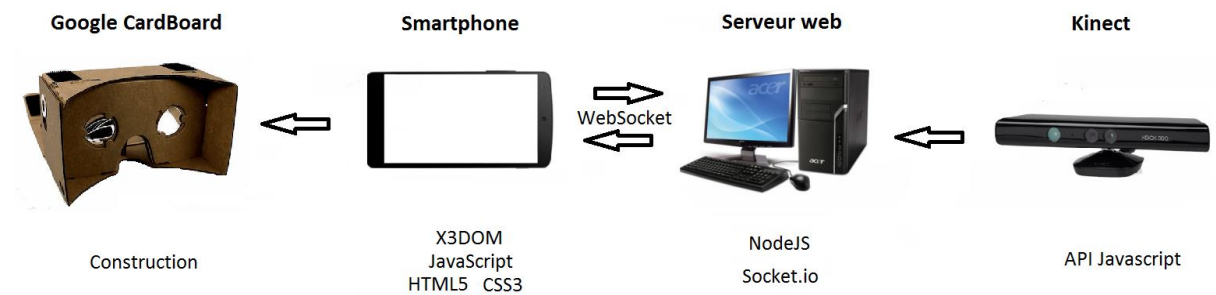
Description détaillée

Pour rendre l'exercice plus réaliste et ressentir les sensations de la planche sous nos pieds et de la peur de tomber, je vais placer une planche sur le sol et les personnes devront marcher dessus en portant les Google CardBoard. Un appareil (Kinect) va récupérer les mouvements et les déplacements de la personne sur la planche pour les répercuter dans le monde virtuel sur le smartphone. On peut aussi imaginer ajouter un ventilateur et des écouteurs pour augmenter l'immersion dans le monde virtuel.

Pour ce projet, je vais travailler sur plusieurs axes (technologies et langages) car les périphériques doivent communiquer entre eux. Le schéma ci-dessous nous montre les axes sur lesquels je vais travailler ainsi que la communication entre les périphériques.

¹ Exemple du projet avec l'Oculus Rift : <https://www.youtube.com/watch?v=7oqazo3ZEnY>

Schéma projet



Ce schéma montre tous les appareils/composants et technologies que je vais utiliser pour la réalisation de ce projet.

Pour cela, j'ai décidé de :

- Construire moi-même les Google CardBoard.
- En plus de la technologie X3DOM, je vais utiliser les basiques du web (HTML/CSS et JavaScript)
- Mettre en place le serveur web en utilisant la technologie NodeJS et Socket.io. (Ces deux technologies se basent sur le JavaScript)
- Utiliser l'API JavaScript du Kinect

Les explications de ses choix sont décrites dans ce rapport.

Organisation projet

Afin de pouvoir avoir un projet complet, je l'ai séparé en 2 étapes.

La première, lors du projet de semestre, durant laquelle je vais étudier, définir et tester toutes les technologies et composants dont j'aurais besoin pour arriver au bout de ce projet.

Durant cette partie, je vais construire les bases du projet. C'est-à-dire, la construction des lunettes, avoir une vue 3D qui s'adapte aux lunettes, aux mouvements et la communication entre les composants/technologies mises en place.

La deuxième, lors du projet de Bachelor, durant laquelle je vais travailler sur la partie de déplacement de la personne en utilisant une Kinect et la mise en place de la vue finale. Je vais donc reporter le déplacement de la personne sur la planche dans la vue sur la CardBoard.

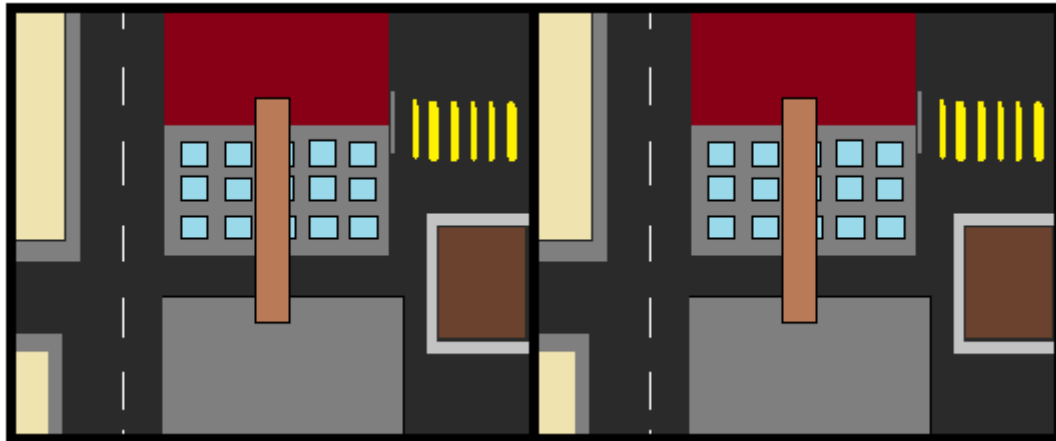
Hébergement

Lien dépôt git (privé) : <https://github.com/PimsJay01/Vertigo>

Lien hébergement : <https://git.hepia.ovh:5001/>

Hôte : 80.80.228.13

Modèle de la vue



Je vais modéliser cette vue ou une vue qui y ressemble en utilisant la technologie X3DOM ainsi que les langages web basiques (HTML, CSS et JavaScript). Comme vous pouvez voir elle sera séparée en deux car c'est comme cela que nous la verrons dans les lunettes.

Je vais essayer de rendre cette vue la plus réaliste possible.

Communication entre les composants

La vue que je vais modéliser s'adaptera en fonction du mouvement de la tête de la personne portant les lunettes. Pour l'adaptation, je vais utiliser les capteurs de déplacement du smartphone. Nous accèderons à ceux-ci depuis le navigateur web.

J'ai également un serveur web qui envoie de manière cyclique des messages au smartphone connecté.

Ce serveur me permettra, lors du travail de Bachelor, de transmettre les données provenant de la Kinect ainsi je pourrais déplacer le personnage et ses membres dans la réalité virtuelle pour un rendu le plus réel possible.

Google CardBoard

Description

Vous pouvez voir les Google CardBoard comme un Oculus Rift pas cher.

En effet, les Google CardBoard se construisent à la maison avec très peu de matériel. Il est donc possible pour n'importe quelle personne d'en construire une.

Les Google CardBoard sont accessibles par tout le monde à un prix très bas. Google permet via ses lunettes à tout le monde de pouvoir s'amuser avec la 3D sans avoir à se ruiner.

Utilité dans le projet

L'un des axes de ce projet de semestre est d'essayer les Google CardBoard et de s'en servir avec un smartphone comme lunettes de réalité virtuelle.

Etudes, recherches & tests

Montage

Pour construire² une Google CardBoard il faut :

- Du carton pas trop épais (1-2mm)
- Des lentilles biconvexes
- Un tag NFC
- Un aimant en céramique
- Un aimant en néodyme
- Du velcro pour tenir le smartphone

J'ai construit une Google CardBoard complète mais dans notre cas, je n'utilise pas tous les composants se trouvant sur les Google CardBoard. Seules les lentilles auraient suffi.

Les patrons et instructions de montage nous sont fournis par Google sur leur page³ parlant des Google CardBoard. Il suffit simplement de télécharger les documents, les imprimer, les coller sur le carton, découper et assembler le tout pour obtenir les Google CardBoard.

Tests

J'ai téléchargé l'application⁴ CardBoard sur Android qui m'a permet de tester si tout marchait bien et de voir le rendu possible avec un smartphone.

J'ai également testé un exemple⁵ de vue 3D d'une salle de classe que l'on trouve sur le site de X3DOM.

² Matériel pour la construction : http://www.amazon.com/AM-CARDBOARD%C2%AE-Complete-Cardboard-Project/dp/B00LM36DUK/ref=pd_sim_cps_14?ie=UTF8&refRID=1SNC11VTMPQND1MF5ZTE
Construction : <https://www.youtube.com/watch?v=3YopUPZErwI>

³ Page de Google sur la CardBoard : <https://www.google.com/get/cardboard/get-cardboard.html>

⁴ Application CardBoard Android : <https://play.google.com/store/apps/details?id=com.google.samples.apps.cardboarddemo>

⁵ Exemple X3DOM salle de classe : <http://examples.x3dom.org/Demos/ClassroomVR/classroom-cardboard.html>

X3DOM

Description

C'est un Framework open-source qui permet d'exécuter des scènes 3D sur un page WEB. Pour cela, il utilise l'HTML5, la 3D est donc créée par des balises.

Cela permet d'avoir une scène de X3D en direct sur un code HTML DOM, nous pouvons donc manipuler le contenu 3D en utilisant les éléments DOM.

En tant que développeur Web, nous pouvons construire des pages Web et applications qui incluent du contenu 3D en utilisant WebGL sans avoir besoin de plugins.

Utilité dans le projet

Je vais utiliser ce Framework pour modéliser la vue en 3D ci-dessus. Dans notre cas, je vais modéliser cette vue sur Blender (juste la partie graphique) puis en utilisant les événements DOM, je vais déplacer la vue en fonction des capteurs du smartphone.

Etudes, recherches & tests

Suivi de quelques tutos se trouvant sur la page⁶ X3DOM pour comprendre le fonctionnement de ce Framework.

Bilan

Pour l'implémentation de la vue 3D, je vais la modéliser la vue sur Blender. Blender est un logiciel 3D qui permet de modéliser une vue et d'exporter celle-ci en un fichier x3d, ce fichier contient toute la scène. Ce fichier peut ensuite être importé dans les balises X3DOM et manipulé par le navigateur.

Capteurs de mouvements

Description

Depuis des années, les smartphones ont des plus en plus de capteurs différents permettant de capter énormément de choses tels que les pas, le déplacement du smartphone et bien d'autres choses.

Utilité dans le projet

Comme j'ai décidé que notre projet se fera entièrement en JavaScript, j'ai dû trouver le moyen de détecter les mouvements de la tête via le navigateur web pour permettre d'adapter la vue lors du mouvement de la tête de la personne portant les Google CardBoard.

Etudes, recherches & tests

Il existe beaucoup de capteurs de mouvements différents intégrés à notre smartphone.

⁶ Tutoriel X3DOM : <http://doc.x3dom.org/tutorials/index.html>

Types

Accelerometer

Il mesure l'accélération appliquée au smartphone en prenant compte de la force de gravité. Il capte les valeurs le long des axes x, y et z.

Linear Accelerometer

Il nous fournit un vecteur tridimensionnel représentant l'accélération le long de chaque axe en excluant la gravité.

Gravity

Il mesure le vecteur à trois dimensions indiquant la direction et l'ampleur de la gravité.

Gyroscope

Il mesure la vitesse ou la rotation en radians par secondes autour des axes x, y et z du smartphone.

Uncalibrated Gyroscope

Similaire au gyroscope sauf qu'aucune compensation gyro-dérivé est appliquée à la vitesse de rotation. Il fournit des résultats plus bruts et peut inclure un certain biais. Il permet également de mesurer la dérive estimée autour de chaque axe.

Il est utilisé pour le post-traitement et des données d'orientation de fusion.

Rotation Vector

Ce vecteur représente l'orientation de l'appareil comme une combinaison d'un angle et un axe dans lequel le smartphone a tourné autour d'un angle delta autour d'un axe.

Significant Motion

Il déclenche un événement spécifique à chaque mouvement détecté puis le désactive. Un événement signifiant est un événement qui peut entraîner un changement dans l'emplacement de l'utilisateur.

Step Counter

Il indique le nombre de mesures prises par l'utilisateur depuis le dernier redémarrage tandis que le capteur a été activé. Il a une latence (jusqu'à 10 secondes) mais est plus précis que le capteur de détection.

Step Detector

Il déclenche un événement à chaque fois que l'utilisateur fait un pas. La latence est inférieure à 2 secondes.

Balle qui saute

Pour tester le sensor que j'ai choisi d'utiliser, j'ai codé un petit « jeu ».

Il s'agit d'une balle dans un carré et en fonction des mouvements du smartphone, la balle bouge et rebondit sur les bords du carré.

Bilan

J'avais d'abord décidé d'utiliser l'accéléromètre mais après des recherches, j'ai trouvé que le vecteur de rotation (Rotation Vector) sera plus adapté à notre cas. Car je dois adapter la vue en fonction de la rotation de la tête.

Github

Description

C'est un service web d'hébergement et de gestion de développement de logiciels basé sur le gestionnaire de version git.

Github permet de créer un projet sur lequel des personnes peuvent y contribuer. Il permet également d'avoir un « versionning » sur un projet, ce qui est primordial pour un projet de développement informatique.

Utilité dans le projet

Permet d'avoir un « versionning » de Vertigo et de partager mon travail avec le corps d'enseignement qui suit ce projet et son avancement.

WebHook sur Github

Description

C'est une méthode qui permet d'étendre, personnaliser et intégrer une application web lorsqu'il se produit un événement. Une notification est envoyée à une URL spécifique via HTTP POST.

Le WebHook, est utilisé généralement de 3 façons :

- Push : réception de données en temps réel
- Plugins : traitement de données et envoi de quelque chose en retour
- Pipes : recevoir des données et les transmettre

Utilité dans le projet

Lors de la modification du projet dans le dépôt git, WebHook permet de modifier sur le serveur hébergeant notre site les fichiers modifiés.

Bilan

Cette méthode permet de ne pas avoir à importer manuellement via une application FTP les fichiers ayant changé sur le serveur hébergeant le site web. Ce qui permet de gagner du temps, il suffit de push les modifications sur Github et nous pouvons directement les tester sur l'hébergeur.

WebSocket (NodeJS & MeteorJS)

Description

WebSocket

C'est un protocole conçu pour être mis en œuvre sur les navigateurs web ainsi que sur les serveurs web mais il peut aussi être utilisé sur n'importe quelle application client/serveur.

Il propose une implémentation native et unifiée dans les navigateurs et serveur web d'un canal bidirectionnel qui permet:

- Au serveur de répondre au client en cas d'une requête de celui-ci
- De notifier au client lors d'un changement d'état du serveur
- D'envoyer des données du serveur vers le client (sans que le client ait à effectuer une requête)

NodeJS

C'est un Framework⁷ logiciel et évènementiel en JavaScript. Il contient une bibliothèque de serveur HTTP, ce qui permet de faire tourner et de mieux contrôler un serveur web sans avoir besoin d'un logiciel externe.

Ce Framework permet d'utiliser plusieurs modules tels que

- Socket.io qui permet d'utiliser les WebSockets et de facilement les manipuler
- ExpressJS qui permet de créer et gérer une application web plus facilement
- AngularJS qui permet d'étendre le langage HTML par de nouvelles balises et attributs
- MongoDB qui permet de gérer des bases de données
- Et bien d'autres

Celle-ci est de plus en plus populaire en tant que plateforme serveur.

MeteorJS

C'est un Framework⁸ simple pour la construction des sites Web modernes, qui se distingue parmi les autres avec son point de vue sur la création d'application web.

Il connecte de façon transparente le client avec Node/MongoDB et promet des applications rapides.

Utilité dans le projet

J'ai besoin d'un serveur pour que nous puissions envoyer les données du Kinect au serveur hébergeant le projet via les WebSockets.

De manière générale, la communication se fait toujours du client vers le serveur. Le serveur se contente de répondre aux requêtes des clients. Pour ce projet, j'ai besoin que le serveur envoie les données provenant du Kinect de manière cyclique à l'utilisateur.

⁷ Site NodeJS : <https://nodejs.org/>

⁸ Site de MeteorJS : <https://www.meteor.com/>

Etudes, recherches & tests

NodeJS

Pour tester NodeJS et comprendre comment il fonctionne, j'ai suivi un tutoriel⁹ qui m'a permis de créer un chat client-serveur. J'ai choisi de faire ce tutoriel parce que le serveur web que je vais mettre en place utilise le même système de communication. Mais à l'inverse, le serveur envoie des informations au client.

MeteorJS

Pour tester MeteorJS et le comprendre, j'ai suivi le tutoriel¹⁰ disponible sur leur site. Il m'a permis de créer un serveur web rapidement sans avoir à coder beaucoup.

Bilan

Lors des tests de ses deux Frameworks, je me suis rendue compte que MeteorJS ne correspondait pas à ce que je voulais faire. En effet, MeteorJS gère tout alors que je veux pouvoir gérer totalement le serveur Web.

Je vais donc utiliser NodeJS qui permet de gérer à plus bas niveau le serveur web. Je vais surtout utiliser le module Socket.io qui me permet de facilement manipuler les WebSockets.

Kinect

Description

C'est un périphérique initialement destiné à la console de jeux Xbox 360 permettant d'agir sur des jeux vidéo sans utiliser de manette.

Depuis quelques années, Microsoft permet d'utiliser ce périphérique pour l'intégrer dans un projet ou une application.

Utilité dans le projet

J'ai besoin d'un outil pour capter les mouvements du porteur des Google CardBoard pour les reporter sur la vue en 3D.

Etudes, recherches & tests

Kinect v1 Prérequis¹¹

OS	Windows 7 ou plus
Processeurs	32 ou 64 bit
Type Processeur	Dual-core, 2.66-GHz ou plus rapide

⁹ Tutoriel NodeJS : <http://socket.io/get-started/chat/>

¹⁰ Tutoriel MeteorJS : <https://www.meteor.com/try>

¹¹ Prérequis Kinect v1 : <https://msdn.microsoft.com/en-us/library/hh855359.aspx>

RAM	2 GB
USB dédié	2.0
Carte graphique	DirectX 9.0c
Software	Virtual Studio 2010 ou 2012

Kinect v2 Prérequis¹²

OS	Windows 8 ou plus haut
Processeurs	64 bits
Type Processeur	I7 3.1 GHz ou plus
RAM	4 GB
USB dédié	3.0
Carte graphique	DX11 capable
Software	Virtual Studio 2012

API

Lors de ma recherche, j'ai trouvé que pour les deux versions du Kinect il y a une API Java et une API JavaScript. Ses deux API sont bien documentées, nous pouvons y trouver beaucoup d'exemples.

Détection d'une personne

Pour tester le Kinect, j'ai codé une petite application qui permet de détecter une personne et d'afficher le squelette des points stratégique du corps sur un navigateur web.

Bilan

Je vais utiliser la Kinect v1 pour plusieurs raisons qui sont les suivantes :

- Kinect v2 trop strict au niveau prérequis
- Un Kinect v1 est disponible donc pas besoin d'acheter la v2
- Je travaille sur Windows 7 et mon ordinateur n'a pas les composants Hardware prérequis pour la Kinect v2

J'ai également décidé d'utiliser d'API JavaScript parce que la communication entre le smartphone et le serveur web de l'application se fait en JavaScript, il est donc plus adapté d'utiliser du JavaScript pour ne pas se perdre dans une multitude de langages.

¹² Prérequis Kinect v2 : <https://msdn.microsoft.com/en-us/library/dn782036.aspx>

Conclusion

Suivi du projet

Durant tout le long du projet, pour le suivi, j'ai tenu un cahier de bord pour chaque entretien. Cela a permis à moi et au corps d'enseignement de suivre le flux du projet et de se rappeler ce qui a été discuté lors des entretiens.

Installation & Utilisation

Pour que l'utilisateur puisse installer et utiliser mon application, j'ai écrit un tutoriel pour l'installation de chaque technologie, de chaque composant et un manuel d'utilisateur pour mon projet.

Continuité

Comme je l'ai mentionné dans la description du projet, je vais continuer ce projet lors du travail de Bachelor.

Le travail à effectuer pour que Vertigo soit opérationnel est le suivant :

- Détecter les mouvements de l'utilisateur lors du déplacement sur la planche via le Kinect
- Mettre en place la scène 3D finale
- Transmettre les déplacements et mouvements via le serveur web sur lequel est connecté le Kinect.
- Reporter les déplacements et mouvements détectés par le Kinect sur la vue 3D du smartphone.
- Chercher des solutions pour augmenter l'immersion totale de l'utilisateur dans le monde virtuel.

Bilan du projet de semestre

Le travail effectué lors de ce projet est le suivant :

- Construction des Google CardBoard
- Mise en place en utilisant X3DOM, sur un navigateur, d'une vue 3D de deux immeubles reliés par une planche (implémentée sur Blender)
- Adaptation de la vue en utilisant les capteurs sur le smartphone
- Mise en place d'un serveur web en NodeJS et Socket.io communiquant de manière cyclique avec la page web contenant la vue. (envoi d'un message)
- Détection en utilisant le Kinect v1 d'une personne et affichage du squelette de celle-ci sur un navigateur

Ce travail me permet donc de pouvoir commencer le travail de Bachelor en ayant toute la base du projet et en ayant déjà travaillé sur chaque technologie. Le travail de Bachelor sera donc l'intégration et la finalisation de cette base mise en place lors de ce projet de semestre.

C'est un projet très intéressant, il m'a permis d'apprendre de nouveaux langages et technologies avec lesquels j'aimerais travailler plus tard. En effet, ce que j'ai appris lors de ce projet est de plus en plus utilisé de nos jours et est donc un sujet qu'il faut maîtriser si je veux évoluer dans mon métier. De plus, le fait que ce projet puisse peut-être aider les gens ayant la peur du vide a rendu ce projet plus motivant.

Annexes

- Cahier de bord
- Tutoriel

Liens utiles

Exemple du projet avec OculusRift : <https://www.youtube.com/watch?v=7oqazo3ZEnY>

Site des Google CardBoard : <https://www.google.com/get/cardboard/get-cardboard.html>

Construire les Google CardBoard: <https://www.youtube.com/watch?v=3YopUPZErwI>

Kit complet AMAZON pour les Google CardBoard: http://www.amazon.com/AM-CARDBOARD%C2%AE-Complete-Cardboard-Project/dp/B00LM36DUK/ref=pd_sim_cps_14?ie=UTF8&refRID=1SNC11VTMPQND1MF5ZTE

Application CardBoard Android :
<https://play.google.com/store/apps/details?id=com.google.samples.apps.cardboarddemo>

Site de X3DOM : <http://doc.x3dom.org/>

Exemple d'une salle de classe en X3DOM :
<http://examples.x3dom.org/Demos/ClassroomVR/classroom-cardboard.html>

Site sur les capteurs de mouvements :
http://developer.android.com/guide/topics/sensors/sensors_motion.html#sensors-motion-grav

Site de NodeJS : <https://nodejs.org/>

Site de Socket.io : <http://socket.io/>

Exemple chat client-serveur NodeJS : <http://socket.io/demos/chat/>

Prérequis Kinect v1 : <https://msdn.microsoft.com/en-us/library/hh855359.aspx>