

# Tutoriel Virtual-Vertigo

## 1 CONSTRUCTION DES GOOGLE CARDBOARD

---

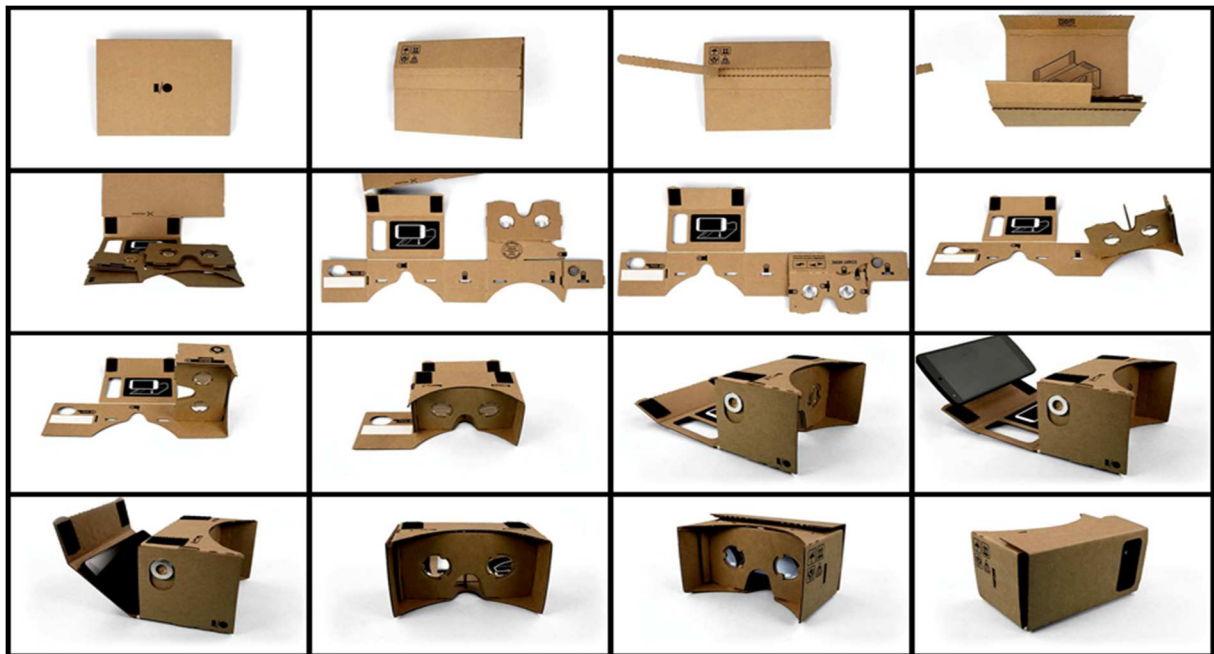
Pour construire des *Google CardBoard* il faut :

- 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



Le matériel de construction se trouve facilement sur *Amazon*. Lien<sup>1</sup> pour l'achat d'un kit complet.

Les patrons de conception et instructions de montage sont fournis par Google sur leur page<sup>2</sup>. Il suffit simplement de télécharger les documents, les imprimer, les coller sur le carton, découper et assembler pour obtenir les *Google CardBoard*. Une vidéo<sup>3</sup> expliquant les étapes pour construire les *Google CardBoard*.



---

<sup>1</sup> Matériel pour la construction : [http://www.amazon.com/AM-CARDBOARD%C2%AE-Complete-Cardboard-Project/dp/BooLM36DUK/ref=pd\\_sim\\_cps\\_14?ie=UTF8&refRID=1SNC11VTMPQND1MF5ZTE](http://www.amazon.com/AM-CARDBOARD%C2%AE-Complete-Cardboard-Project/dp/BooLM36DUK/ref=pd_sim_cps_14?ie=UTF8&refRID=1SNC11VTMPQND1MF5ZTE)

<sup>2</sup> Page des Google CardBoard : <https://www.google.com/get/cardboard/get-cardboard.html>

<sup>3</sup> Construction : <https://www.youtube.com/watch?v=3YopUPZEerwI>

## 2 INSTALLATION ET EXEMPLE D'UTILISATION DE NODEJS

---

Allez sur le site : <http://nodejs.org/download/>

*Infos : Lors de son installation, NodeJS installe également npm qui nous permet de télécharger et installer des modules à NodeJS.*

### Windows

Téléchargez le fichier **msi** pour Windows 32 ou 64 bits disponible sur le site internet et exécutez-le. Laissez les paramètres par défaut.

### Linux

#### *Via terminal*

Ouvrez un terminal et tapez.

```
sudo apt-get install nodejs npm
sudo apt-get update
```

#### *Via internet*

Téléchargez le fichier .tar.gz pour linux 32 ou 64 bits et exécutez-le. Laissez les paramètres par défaut.

### Utilisation de *NodeJS*

*NodeJS* est maintenant prêt à être utilisé. Voici un exemple de serveur web afin que vous puissiez tester :

```
// Load the http module to create an http server.
var http = require('http');

// Configure our HTTP server to respond with Hello World to all requests.
var server = http.createServer(function (request, response) {
  response.writeHead(200, {"Content-Type": "text/plain"});
  response.end("Hello World\n");
});

// Listen on port 8000, IP defaults to 127.0.0.1
server.listen(8000);

// Put a friendly message on the terminal
console.log("Server running at http://127.0.0.1:8000/");
```

Il suffit de lancer un terminal, de vous placer dans le répertoire du serveur et de taper.

*Infos : le .js n'a pas besoin d'être spécifié.*

```
Node <nomserver.js>
```

Après avoir tapé cette ligne de commande, ouvrez un navigateur Web et tapez afin d'afficher ***Hello World***

```
Localhost :8000
```

## Npm

C'est un package manager permettant d'installer des packages sur *NodeJS* permettant de compléter NodeJS en fonction de ce que vous voulez faire avec.

Vous pouvez chercher des packages sur <https://www.npmjs.com/#explicit>.

Voici les packages les plus utilisés :

- 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

Pour installer un module, ouvrez un terminal, placez-vous dans le dossier où se trouve votre serveur node et tapez.

```
npm install <nomdumodule>
```

## 3 INSTALLATION ET TEST DU KINECT V1

---

**Attention : Vous devez être sur Windows 7 ou plus pour pouvoir utiliser le *Kinect v1***

### ***SDK***

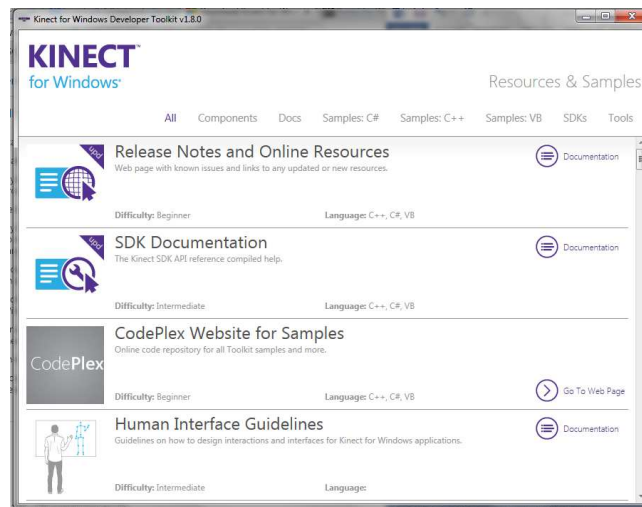
Allez sur <http://www.microsoft.com/en-us/download/details.aspx?id=40278> et téléchargez le SDK 1.8. Exécutez KinectSDK-v1.8-Setup.exe et laissez tous les paramètres par défaut.

### ***Developer Toolkit***

Allez sur <http://go.microsoft.com/fwlink/?LinkID=323589> et téléchargez le *Toolkit*. Exécutez KinectDeveloperToolkit-v1.8.0-Setup.exe et laissez tous les paramètres par défaut.

## Test du Kinect

Branchez votre Kinect sur votre ordinateur et ouvrez le *Toolkit*.



Choisissez un exemple sous *Sample C#* ou *Sample C++* ou *Sample VB* et cliquez sur play.

## 4 UTILISATION DE L'APPLICATION VIRTUAL-VERTIGO

---

### 4.1 RÉCUPÉRATION DES SOURCES DU PROJET

Récupérez le zip contenant l'application *Virtual-Vertigo* se trouvant sur le CD annexé au rapport. Dézippez-le.

### 4.2 LANCEMENT DU SERVEUR

Connectez le Kinect v1 à votre ordinateur et placez la planche sur le sol face au Kinect.

Puis allez dans le dossier Application de l'archive dézippée auparavant et lancez le fichier *launcher.bat*.

*Infos : Ce fichier bat va lancer l'application du Kinect et le serveur Web NodeJS Virtual-Vertigo.*

Lorsque le *serveur Web Virtual-Vertigo* est lancé, il fournit l'adresse ip et le port du PC. Veuillez **retenir** ou noter **cette adresse ip et port**.

### 4.3 LANCEMENT DE LA SIMULATION

Sur le PC, ouvrez un navigateur et allez sur `<adresse ip :port/index.html ?mode=2>` (adresse ip et port noté auparavant) .

Sur votre smartphone, ouvrez un navigateur et allez sur `<adresse ip :port/index.html ?mode=1>` (adresse ip et port noté auparavant) et cliquez sur *fullscreen*

Placez votre smartphone dans les Google Cardboard de cette manière :



Pour finir, placez-vous à l'extrémité de la planche et accrochez les *Google CardBoard* sur votre tête en utilisant le velcro.

Profitez de votre simulation !!