

Pelmard Bastien  
Oktay Erkan  
Sharif Mohammad

## Rapport Projet Javascript: Developpement Web Licence 3

### But du projet:

Le but de notre projet était de créer un FPS multijoueurs à l'aide de la librairie Three.js.

Pour se faire, nous avons utilisées d'autre librairies à savoir:

- La plateforme de réseau Node.js pour exécuter du JavaScript coté serveur.
- La librairie Socket.io pour une communication en temps réel.
- Express.js pour envoyer les données au serveur.

On à donc du installer:

- Node.js depuis le site officiel.
- Socket.io et Express.js depuis le terminal de Node.js avec les commande "npm install socket.io" et "npm install express".
- La librairie Three.js depuis le site officiel.

### Lancer le site:

Pour lancer le site il faut faire:

- Installer Node.js sur son ordinateur.
- Saisir la commande "node app.js"
- Lancer dans un navigateur le site "<http://localhost:3000>"

### Fonctionnement du site:

Pour naviger dans le jeu il est conseiller de le metre en pleine écran avec la touche F12. Il suffit ensuite de cliquer sur le jeu pour verrouiller son pointeur.

Les touches sont:

- "Z" ou "Fleche du haut" pour avancer avec la touche "Shift" qui sert à sprinter.
- "S" ou "Fleche du bas" pour reculer.
- "Q" ou "Fleche de gauche" pour aller sur la gauche.
- "D" ou "Fleche de droite" pour aller sur la droite.
- "Clique droit de la souris" pour tirer.

## Conception du site:

Le site est composé en plusieurs fichiers différents à savoir un fichier HTML, des fichiers JavaScripts, les différents modules de "socket.io" et "express.js" présent dans le dossier "node\_modules" et les différents ressources du jeu (audio, models, textures, skybox).

Le plus important reste évidemment les fichiers JavaScripts qui sont nombreux dans le projet mais seulement trois sont intéressant à citer car le reste est présent au bon fonctionnement des différentes librairies:

- "app.js" qui va gérer le serveur à savoir récupérer les ressources nécessaires au jeu et les envoyer dans le serveur ou encore d'appeler les fonctions qui vont permettre le multijoueurs du jeu ou l'initialisation du serveur.
- "serveur\_world.js" qui est le fichier qui initialise les fonctions appelées dans "app.js".
- "client\_world.js" qui va être le jeu en lui-même à savoir l'initialisation du monde, le chargement des models, de l'audio, le fonctionnement des touches...

Pour de meilleures explications de ces trois fichiers il est conseillé de lire les commentaires qui sont écrits à l'intérieur même de ces fichiers.

## Bugs et problèmes du site:

Plusieurs problèmes ont été détectés:

- Bug chargement du camion: Il se peut que le chargement du camion ne puisse avoir lieu sur Google Chrome. Ce problème n'est pas présent sur Firefox et Edge.
- Il est actuellement impossible de voir les balles que l'adversaire tire mais celle-ci font quand même des dégâts à nos personnages en cas de collision.
- Selon un certain angle, il se peut que les collisions avec les models présents sur la map ne fonctionnent pas sur les côtés gauche et droite de nos personnages. Il n'y a cependant pas de problèmes avec les collisions à l'avant et l'arrière de nos personnages.
- Pour un problème de Blending, nous sommes obligés de laisser les différentes collisions des models partiellement apparentes. Si nous les mettons transparentes, cela ne fait pas geler le jeu mais produit énormément d'erreurs dans la console du navigateur.