

Rapport séance 1

Après avoir effectué la séance d'introduction de l'option « Nouveaux modes de développement », une liste de tâches a déjà été réalisée et c'est pour cela qu'il est important d'en laisser une trace écrite. Je vais donc résumer de manière brève les actions réalisées et celles que je compte réaliser dans les prochains temps.

Tout d'abord, pour pouvoir se familiariser avec le langage enseigné : Le NodeJS, il est important de mettre son environnement de travail en place. C'est-à-dire, d'installer tous les outils nécessaires afin de développer avec ce langage. Pour ma part, j'ai d'abord installé une machine virtuelle Linux sur mon ordinateur personnel car ce système paraissait mieux adapté pour ce type de travail. Une fois la VM installée, j'ai téléchargé les logiciels demandés à l'aide de l'invite de commandes pour mener à bien le projet. La commande en question est « `sudo apt-get install` ». Les logiciels installés sont les suivants : Emacs, un éditeur de texte utile pour écrire mes codes sources en NodeJS. Node, pour pouvoir exécuter les applications développées sur ma machine et tester mes codes. Git, un logiciel de versionning et enfin Cairo, un outil permettant de réaliser des applications graphiques. Après une heure d'installation de l'environnement, j'ai enfin pu me lancer dans le vif du sujet qui est le projet à réaliser.

Ensuite, j'ai commencé le projet par les premières tâches demandées. La première a été de créer un serveur HTTP en NodeJS. Après avoir effectué quelques recherches, j'ai réussi à réaliser le code pour mettre en place le serveur HTTP, qui permettra par la suite, aux joueurs de s'y connecter pour jouer au jeu. Pour suivre les étapes, j'ai effectué une petite page HTML très simple à afficher quand on se connecte au serveur via un navigateur. J'estime avoir passé une heure sur ces dernières actions. Une fois le serveur http déployé, il a été demandé de le changer en HTTPS, un serveur sécurisé. Pour ce faire, il a fallu créer une clé privée et un certificat SSL pour pouvoir utiliser le serveur HTTPS. Toutefois, après avoir effectué quelques recherches sur internet, le code source et les commandes permettant de créer la clé et le certificat n'ont pas été difficile à trouver. J'ai donc tapé la liste de commandes permettant de créer les fichiers voulus et il s'avère qu'ils sont de formats « `.key` » et « `.crt` » et également « `.csr` ». Par contre, l'utilisation de ces derniers fichiers a été plus délicate. Malgré les exemples rencontrés sur internet, mon serveur HTTPS ne fonctionnait pas, quand je voulais m'y connecter, le navigateur bloquait à l'étape de la connexion sans finir par afficher la page HTML. J'ai donc continué mes recherches et j'ai effectivement trouvé qu'il fallait utiliser des fichiers de format « `.pem` ». Après avoir fait quelques commandes sur Linux, j'ai réussi à changer mes fichiers en format « `.pem` », mais même après cette modification, mon serveur ne fonctionnait pas. Je pense avoir consacré deux heures et demie à avoir créé les fichiers, utiliser ce dernier, et tenter de trouver une solution à mon problème.

Pour la prochaine séance, l'étape du serveur HTTPS sera donc ma priorité car malgré mes recherches, je n'ai pu régler le problème de certificat et de clé privée. J'espère ne pas passer plus d'une heure et demie sur ce même problème, afin de mettre mon serveur en place.

Il sera alors temps de se lancer dans la partie la plus importante du projet : L'élaboration des dessins de disques, de serpents et de leurs mouvements.