

Université de Paris 8

RAPPORT

RAPPORT DU PROJET TUTEURE
Master Informatique première année

sur le sujet

MONDE VIRTUEL EN WEBGL

rédigé par

INE Thierry et Hamidou Touré

Table des matières

1	Introduction	2
2	Etat de l'art	2
2.1	Three.js	2
2.2	Monde Virtuel	2
3	Implémentation	6
3.1	Description de notre application	6
3.2	Aspect Techniques	6
4	Réalisation	7
4.1	Les terrains	7
4.2	Les fonctionnalités	8
5	Déroulement du projet	9
6	Problèmes Rencontrés	10
6.1	Pas assez connaissance en WebGL	10
6.2	Implémentation de la souris et du clavier pour le déplacement	10
6.3	Dessiner les arbres et la forêt	11
6.4	Changement de temps	11
6.5	Gestion des éléments dynamiques	11
6.6	Performance coûteuse	11
6.7	Gestion du temps	11
6.8	GitHub	11
7	Améliorations	12
8	Conclusion	13

Table des figures

1	Jeu 3D en Three.js - Kaiopua	3
2	Jeu 3D - Find Your Way to Oz	4
3	Jeu 3D - 3DChat	4
4	Scène principale	7
5	Désert	8
6	Eau	8
7	BrickFloor	9

1 Introduction

Dans le cadre de notre cursus Master 1 MIME à l'université de Paris VIII, nous devons réaliser un projet tuteuré sur le sujet de "Monde Virtuel". L'objectif est de créer un monde virtuel en WebGL dans lequel l'on peut se promener en utilisant la librairie « three.js ».

Outils informatiques de développement : Les outils informatiques utilisés pour réaliser ce projet sont :

- Eclipse et Sublime Text 3 : pour la programmation en langage Javascript
- Git et GitHub : pour le partage de code et la maintenance de l'historique de l'ensemble des fichiers
- LaTeX : pour la rédaction des documents
- PhotoShop, Paint, Paint.net : pour la modification d'images
- WampSever et Apache sur Windows
- Three.js : librairie permettant l'implémentation des codes WebGL

2 Etat de l'art

Pour réaliser un excellent projet, nous étions obligés d'effectuer des recherches sur le sujet d'un monde virtuel. Nous nous sommes posés quelques questions avant de pouvoir démarrer notre projet, ces questions sont : Qu'est-ce qu'un monde virtuel ? Existe-il des projets similaires ? Quelles sont les méthodes qui ont été utilisées ? Quelles sont les outils informatiques qui peuvent nous aider à l'implémenter ? Three.js ?

2.1 Three.js

WebGL est une API créée pour afficher du graphique et des éléments 3D sur un navigateur Web. Three.js est une librairie qui est sensée faciliter l'utilisation de WebGL. Il existe un grand nombre de démos sur leur site www.threejs.org.

2.2 Monde Virtuel

La définition du terme monde virtuel peut varier de personne en personne, d'époque en époque, de domaines technologiques etc. Notre définition personnelle d'un monde virtuel est la simulation d'un environnement inexistant, un endroit de rêve ou de cauchemar, d'une illusion, d'un monde irréel qui peut aussi être influencé par un monde réel.

Dans le fil de l'histoire des tentatives de création de mondes virtuels ont effectuées par plusieurs artistes avant son introduction dans le domaine informatique.

« Si on se focalise strictement sur le fait que la réalité virtuelle est une manière de créer l'illusion d'être présent à un endroit où ne sommes pas alors les premières tentatives d'une réalité virtuelle sont les 360 degrés ou panoramiques peintures murales du 19ème siècle. Ces peintures étaient destinées à remplir tout le champ de vision du spectateur le faisant se sentir présent à un événement ou une scène historique. » [1]

La création d'un monde virtuel que ce soit dans le domaine de l'art et de la peinture ou dans le domaine informatique demande une connaissance minimum en objets 3D et en perspectives du cerveau humain. L'une des premières tentatives technologiques de création de réalité virtuelle fut réalisée par Monsieur Charles Wheatstone, chercheur scientifique et inventeur du stéréoscope.

« En 1838 les recherches de Charles Wheatstone ont démontrées que le cerveau traite les différentes images 2D en un seul objet de dimension 3D. Le fait de voir deux images stéréoscopiques côte à côte à travers un stéréoscope donne un sens de profondeur et d'immersion au spectateur. Le développement ultérieur du célèbre View-Master stéréoscope était utilisé pour réaliser du tourisme virtuel. Les principes de conception du stéréoscope sont utilisés aujourd'hui pour réaliser le célèbre Google Cardboard et pour le montage d'affichages de réalité virtuelle pour les téléphones mobiles» [1]

La réalité virtuelle ou la création d'un monde virtuel existe depuis des années et même des siècles. On peut la retrouver dans plusieurs domaines comme l'architecture, la construction, l'ingénierie, les sciences de la nature et de l'environnement, l'informatique, la physique, l'art et la peinture, la médecine, la philosophie, etc. On constate qu'au 20e et 21e siècle l'informatique à jouer un très grand rôle, pour ne pas dire le plus grand rôle dans l'histoire de la réalité virtuel en la faisant atteindre un niveau plus avancé. De nos jours le domaine informatique s'intéresse de plus en plus à la réalité virtuelle et aux mondes virtuels. Il existe plusieurs applications et jeux 3D implémentant des mondes virtuels permettant à plusieurs utilisateurs d'y participer. La plupart de ces jeux sont des jeux d'aventure donc ils effectuent des simulations de mondes variant selon le contexte du jeu, ces applications ont moins de contrainte car le rendu graphique n'est pas toujours obligé de respecter les lois de la physique d'un monde réel. Par contre il existe aussi des jeux ou applications qui simulent les différentes activités du monde réel donc ils sont contraints à respecter les lois de la physique.

Voici une liste de quelques applications existantes qui sont dans le même contexte que notre projet :

Kaiopua : est un jeu constructeur / puzzle science-fiction, qui a le joueur adopte l'identité d'un Islander Pacifique, perdu en mer, qui trouve une maison à bord de la plus inattendue de créatures en apprenant les manières de la culture du taro. L'objectif de base de ce jeu est de créer un outil d'éducation de conception universelle qui peut enseigner aux joueurs à mieux comprendre la grille, l'espacement, l'ordre, la modularité et efficacité au sein de la conception. Ciblent principalement les auditoires sont de jeunes étudiants, environ 10 ans - 14, et étudiants de premier cycle de niveau collégial, 18 ans - 25, de tous les genres. [2]



FIGURE 1 – Jeu 3D en Three.js - Kaiopua

Find Your Way to Oz : est une nouvelle expérience Google Chrome apporté à la bande par Disney. Il vous permet de prendre un voyage interactif à travers un cirque Kansas, qui vous mène à la terre d'Oz après que vous avez balayé par une forte tempête. L'objectif était de combiner la richesse du cinéma avec les capacités techniques du navigateur pour créer une expérience amusante, immersive que les utilisateurs peuvent former un lien fort avec. [3]



FIGURE 2 – Jeu 3D - Find Your Way to Oz

3DChat : est un monde virtuel en ligne gratuit où les gens de partout dans le monde peuvent être eux-mêmes. Les joueurs explorent une ville virtuelle où ils peuvent participer à diverses activités, magasiner pour des vêtements, visiter une station ou fête toute la nuit. Semblable à des titres comme IMVU, le principal objectif du jeu est de socialiser plutôt que des quêtes ou des instances exécutées. 3DXChat adopte une approche plus réaliste, puis d'autres titres semblables cependant, la mise en œuvre d'un grand nombre de produits de la vie réelle, les activités et les interactions. Les développeurs ont mis une attention particulière dans la création d'un jeu qui offre aux joueurs tous les outils pour recréer des situations sociales de la vie réelle. Pour les membres premium, ces situations sociales peuvent obtenir assez torride. [4]



FIGURE 3 – Jeu 3D - 3DChat

Nous sommes conscients de l'importance et de l'utilité de notre projet dans le domaine informatique et dans le monde professionnel. Le Mardi 10 Novembre 2015 nous avons reçu la visite de deux anciens étudiants de Master MIME de l'université de Paris 8 qui sont Monsieur Valentin Billotte et Monsieur Cyril Cagnet qui sont devenus maintenant des développeurs et des chefs d'entreprise spécialisés dans le graphique et jeux 3D. Ils ont travaillé sur un projet de réalité virtuelle qui consistait à bloquer le cerveau et le faire croire qu'il est dans un autre monde. Ce projet a tellement bien marché qu'il a intéressé un grand développeur considéré comme le père de la 3D. Ce développeur a réussi à créer un gyroscope qui réagit très rapidement, lorsque l'on porte le casque et bouge la tête la réaction de l'image de la caméra 3D est instantanée ce qui fait que le cerveau est emmené dans un monde virtuelle grâce aux lunettes Oculus. Dans notre cas n'ayant pas ce genre de technologie nous ne pourrions pas atteindre ce niveau. En revanche comme notre application est programmée sur un navigateur web, ceci la rendra facilement accessible à un grand public et gratuitement. Ils nous ont aussi présentés un deuxième projet qui était un jeu 3D qu'ils ont fait pour l'ERDF. Le jeu a été projeté à l'écran pour que nous puissions tous voir son fonctionnement. Ce jeu est une simulation d'un agent d'électricité qui arrive dans une maison où il y a une coupure d'électricité. Ce jeu est constitué de deux étapes, la première étape consiste à regarder par la fenêtre pour voir s'il y a de l'électricité dans la rue pour savoir si c'est une coupure générale ou uniquement dans l'immeuble, et la deuxième étape est de vérifier le tableau électrique pour voir si c'est le local. Ensuite ils nous ont présenté un troisième projet qui consiste à aider les gens à réduire leur facture d'électricité. Le but de ce jeu est de regarder tout ce qui ne va pas dans l'appartement afin réduire la facture d'électricité représentée par une barre rouge affichée sur l'écran. Ce projet a été fait en C Sharp Visual Studio.

On peut dire qu'il existe un très grand nombre de jeux et d'applications qui simulent un monde ou environnement virtuel. Il y a beaucoup d'intérêts dans le développement des jeux ou applications 3D et virtuelles. Elles rapportent beaucoup d'argent à l'entreprise. Cyril et Valentin nous ont présentés un jeu qu'ils ont programmé en une seule semaine et qui a été vendu à 200 000 Euros. La programmation en 3D est la meilleure programmation pour les personnes qui aiment beaucoup le développement car elle permet de toucher à tous les aspects et techniques informatiques.

3 Implémentation

3.1 Description de notre application

L'objectif étant d'implémenter un monde virtuel dans lequel l'on peut se promener, pour atteindre cet objectif nous avons implémenté un monde composé de plusieurs différents terrains (désert, eau, herbe, forêts ...). La caméra qui représente le joueur ou utilisateur avance tout doucement et automatiquement d'une extrémité du monde à l'autre extrémité du monde. Le joueur a la capacité de pivoter et accélérer grâce aux touches du clavier pour observer son environnement. Chaque terrain contient des éléments qui le rendent unique. L'implémentation d'un monde virtuel a moins de contraintes que l'implémentation d'un monde réel donc nous nous sommes basés sur cette idée pour implémenter un monde virtuel simple. Malgré que notre monde ne soit pas un monde réel il se base sur les éléments qui existent déjà dans un monde réel.

3.2 Aspect Techniques

Après nos recherches nous avons démarrés l'implémentation du projet. La première étape a été de créer nos fichiers « HTML et CSS » et notre fichier « Javascript » dans lequel est contenu le code principal.

Les variables : Nous avons déclarés des variables globales importantes et utiles pour la gestion des éléments de l'application comme : la caméra, la scène, le rendu graphique, l'angle de rotation de la caméra, les coordonnées du positionnement de la souris, l'horloge, le temps écoulé, les lumières, les conteneur HTML dans lesquelles on affiche des informations pour le débogage, les objets etc.

Fonctions : Nous avons implémentés quelques fonctions permettant d'initialiser notre scène et de gérer tous les éléments de notre scène.

- **Init()** : Initialise le moteur du rendu graphique, la scène, les terrains (désert, eau, herbe, forêt ...), la lumière et les matériaux. Elle affiche ensuite la première image de l'interface graphique qui est un terrain d'herbes avec un ciel bleu avec du brouillard, elle place la caméra à la première extrémité de la scène. C'est dans cette fonctionne qu'on initialise notre horloge.
- **keyPressed()** : Permet de gérer le déplacement du joueur à l'aide des flèches « haut et bas » du clavier. Elle prend en paramètre un évènement, ensuite vérifie la touche sur laquelle le joueur appuie en récupérant le code de la touche. Si la flèche du haut est appuyée le joueur avance, si la touche du bas est appuyée le joueur recule.
- **setPeriod()** : Cette fonction nous permet de mettre à jour la lumière et le brouillard en fonction du temps écoulé. Elle utilise une liste de couleurs hexadécimales que nous avons définis pour représenter les périodes du jour et de la nuit. Cette fonction peut être améliorée si on réussit à bien coordonner les couleurs des lumières avec les couleurs du brouillard.
- **drawForest()** : Comme son nom l'indique, cette fonction nous permet de dessiner une forêt d'arbres à partir d'un repère. Une forêt est un ensemble d'arbre 3D, un arbre est implémenté ici comme une sphère au-dessus d'un cylindre et c'est sur cette idée que notre

algorithme est basé.

- **updateInfo()** : Cette fonction nous permet d'écrire ou afficher des informations sur l'écran. Elle prend en paramètre un message (chaîne de caractères HTML) qui sera affiché à l'écran. Cette fonction nous a permis d'avoir des informations sur nos éléments et objets en temps réel sans ouvrir la console du navigateur, ceci à faciliter le dessin de notre scène et l'emplacement de nos objets.
- **Animate ()** : Elle fait appel à la fonction `requestAnimationFrame` de Three.js qui dessine les images à chaque instant. Dans cette fonction on calcul aussi le temps écoulé et en fonction de ce temps on mets à jour les lumières, le ciel et le brouillard. Le temps écoulé est remis à zéro après durée définie. Elle est aussi utilisée pour affiché des informations utiles pour pouvoir debugger notre application.
- **moveCamera()** : Permet de pivoter la caméra à l'aide de la souris. Cette fonction est désactivée pour la version finale de l'application.

4 Réalisation

Après avoir collecté assez d'informations utiles concernant le projet, nous avons pu concevoir une scène simple. Cette scène nous présente un terrain d'herbe avec un ciel bleu qui contient du brouillard, voir la Figure4. Le joueur peut pivoter la caméra grâce aux flèches du clavier. La caméra avance toute seule jusqu'à l'extrémité du monde et est censé revenir en arrière. Sur cet terrain on trouve aussi une forêt à gauche et à droite.

4.1 Les terrains



FIGURE 4 – Scène principale

Nous avons ensuite implémenté un deuxième terrain qui est un désert, voir la Figure 5. Ce terrain est de dimension 20000 x 20000. On utilise une texture de type désert qu'on a téléchargé sur le net. Ce terrain est construit en utilisant le matériel "MeshPongMaterial" de ThreeJS, il a aussi la capacité d'être affecté par l'ombre de la lumière.



FIGURE 5 – Désert

Nous avons ensuite implémenté notre troisième terrain qui est de l'eau, voir la Figure 6. Cet élément est de dimension 20000 x 20000, il utilise une texture téléchargée à partir du site de ThreeJS. Il est censé être dynamique mais il ne l'est pas pour le moment.

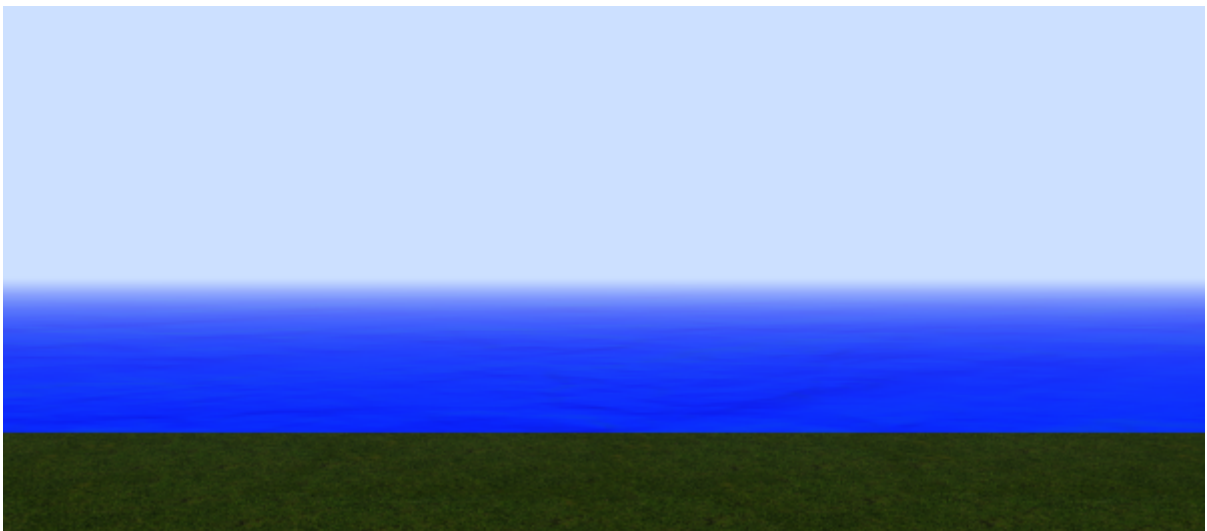


FIGURE 6 – Eau

Ensuite nous avons implémenté notre quatrième terrain qui est un sol en briques téléchargé aussi à partir du site de ThreeJS, voir la Figure 7.

4.2 Les fonctionnalités

En fonction du temps que nous avons, nous avons pu implémenter certaines fonctionnalités supplémentaires à notre projet dans le but de le rendre plus intéressant. Les fonctionnalités que nous avons sont :

L'implémentation de plusieurs terrains et environnements : comme du sable (désert), de l'eau (rivières, mers, océans ...), des forêts. Ces éléments sont statiques dans un premier temps et nous permettent d'agrandir notre monde et de se promener dans plusieurs endroits

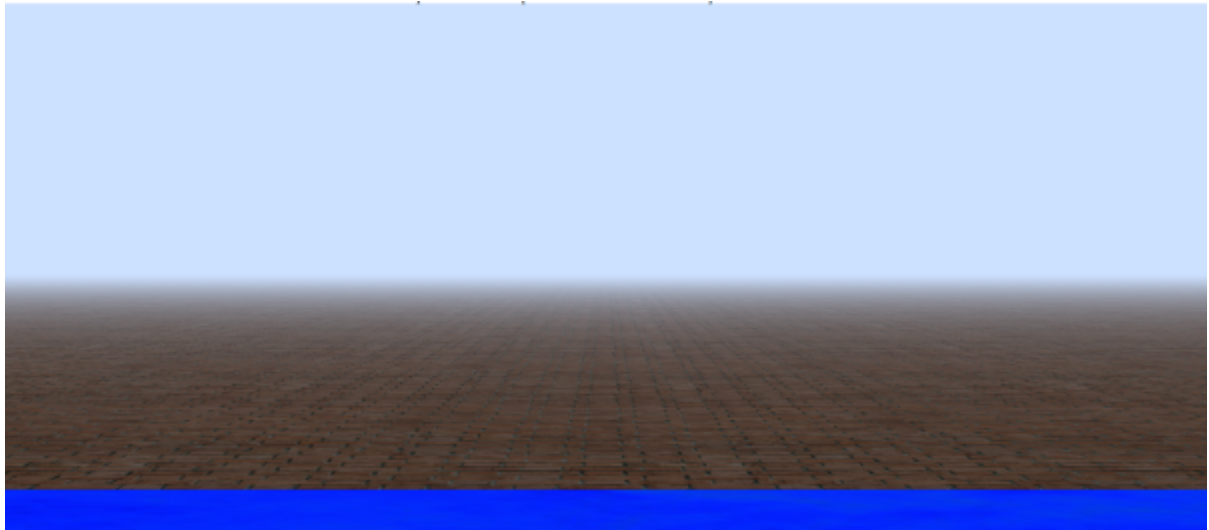


FIGURE 7 – BrickFloor

différents pour éviter que l'utilisateur s'ennuie. Nous comptons ensuite rendre ces éléments dynamiques pour avoir un semblant de rendu visuel en temps réel.

Utilisation du clavier et de la souris : pour gérer les déplacements de l'utilisateur et les mouvements (rotation) de la caméra. Cette idée est inspirée des jeux d'aventures existantes déjà sur les consoles PlayStation et Xbox. Ceci consiste à utiliser la souris comme une caméra pouvant pivoter sur un angle de 360 degrés et les boutons « haut et bas » serviront seulement à avancer et à reculer. Après avoir testé le déplacement avec la souris et le clavier, nous avons préféré le déplacement avec le clavier.

L'évolution du temps : qui consiste à faire évoluer la couleur du ciel et de la scène en fonction du temps écoulé. Ceci nous amène donc à afficher soit une horloge représentant le temps du jeu. Pour atteindre cet objectif nous avons utilisé une technique simple qui est d'insérer 24 ou 48 couleurs de ciels dans un tableau ordonné selon l'intensité de la couleur, ensuite au fur et à mesure que le temps s'écoule, on calcule le pourcentage de temps écoulé et on affecte une couleur au ciel. Comme cet objectif est plus ou moins atteint nous essaieront de modifier la météo en fonction du temps.

5 Déroulement du projet

Dans cette section, nous décrivons comment la réalisation du projet s'est déroulée au sein de l'équipe de projet. La répartition des tâches, la synchronisation du travail et l'utilisation du temps sont abordées.

Nous avons utilisés la méthode agile pour réaliser ce projet. A chaque fois qu'on implémente un nouveau code, on test notre application pour voir si ces nouvelles fonctions ont un effet négatifs sur nos anciens code. Une fois que nous sommes satisfaits du test, nous procédons à la prochaine étape en rajoutant une nouvelle couche.

Pour avoir un bon résultat à la fin du projet, nous avons identifié et défini les tâches utiles et importantes à accomplir, ensuite nous avons établi un planning pour s'organiser selon ces tâches à accomplir, voir la table 1.

Tâches	Personne(s)	Date
Recherche et collecte d'informations	Thierry et Hamidou	Du 07-11-20 au 20-01-2016
Création d'une scène principale basic	Thierry	Du 19-11-2015 au 02-12-2015
Rédaction du rapport	Thierry et Hamidou	Du 04-01-2016 au 20-01-2016
Implémentation du code qui crée la forêt	Thierry	Du 21-01-2016 au 24-01-2016
Implémentation du code qui crée l'eau	Hamidou	Du 21-01-2015 au 24-01-2016
Implémentation du code qui crée le désert	Thierry et Hamidou	Du 25-01-2015 au 27-01-2016
Implémentation du code qui permet de gérer le déplacement à partir du clavier	Hamidou	Du 13-01-2016 au 01-02-2016
Implémentation du code qui permet de gérer le mouvement de la souris	Thierry	Du 13-01-2016 au 01-02-2016
Gestion des obstacles	Thierry et Hamidou	Du 29-01-2016 au 08-02-2016
Implémentation du jour et la nuit	Thierry et Hamidou	Du 29-01-2016 au 08-02-2016
Tests et validations	Thierry et Hamidou	Du 19-11-2015 au 01-03-2016

TABLE 1 – table de planning et organisation

6 Problèmes Rencontrés

6.1 Pas assez connaissance en WebGL

Le fait de n'avoir aucune connaissance ou expérience en WebGL a constitué un blocage important pour nous. Pour résoudre ce problème nous avons lu plusieurs documents fournis sur le site threejs.org. Ensuite nous avons regardé et essayé de comprendre le code et le fonctionnement de certains exemples existant sur le site. Après avoir compris le principe d'utilisation de Three.js, nous nous sommes entraînés en implémentant quelques exemples simples et en effectuant des tests. Cette étape nous a permis de mieux comprendre Three.js, elle nous a aussi aidée à définir les fonctionnalités que nous comptons inclure dans notre projet.

6.2 Implémentation de la souris et du clavier pour le déplacement

Nous avons réussi à un implémenter du code qui permet d'effectuer une rotation de la caméra lorsqu'on bouge la souris et déplacer l'utilisateur lorsqu'il appuie sur les flèches du clavier. Après avoir implémenté cette fonctionnalité nous nous sommes rendu compte qu'elle n'était pas assez fluide et aussi difficile à manipuler. Ensuite nous avons essayé d'utiliser une classe de ThreeJS du nom de « FirstPersonControls » pour gérer les déplacements et mouvements à l'aide de la souris et du clavier, sauf qu'elle ne nous donnait pas le résultat qu'on voulait car elle était parfois trop lente on trop rapide malgré les changements de paramètres. Notre solution finale

a été d'annuler la souris et utiliser les flèches du clavier au lieu du clavier. Ceci a bien marché sauf qu'on n'arrivait pas à bien gérer les angles de rotation en même temps que le déplacement.

6.3 Dessiner les arbres et la forêt

Nous avons rencontré un problème lors du dessin de la forêt. On a voulu utiliser des fractales pour dessiner nos arbres mais le code que nous avons obtenu était difficile à implémenter et nous donnait un rendu 2D, nous sommes passés par plusieurs méthodes pour essayer de transformer notre rendu 2D en 3D mais sans aucun succès. La solution finale pour laquelle nous avons opté a été de représenter nos arbres par des cylindres et des sphères. Ensuite nous avons amélioré ce code pour écrire une fonction qui dessine la forêt (ensemble d'arbres) à n'importe quel endroit de la scène en lui donnant simplement un point initial.

6.4 Changement de temps

Nous avons prévu le changement de la couleur du ciel et de certains éléments de notre scène pour avoir un effet de jour et nuit. Le problème que nous avons rencontré à cette étape est qu'on avait bien défini une liste de couleurs représentant le jour et la nuit sauf lorsqu'on changeait les couleurs du brouillard et du ciel en fonction du temps elle nous affichait une scène obscure qui ne représentait pas la couleur indiquée. En fait le problème est que pour avoir un effet de nuit et jours, il fallait qu'on arrive à bien coordonner les couleurs des lumières avec la couleur du brouillard.

6.5 Gestion des éléments dynamiques

Nous avons essayé d'implémenter des éléments dynamiques (animaux, eau, pluie ...) pour donner plus de vie à notre monde mais ceci fut difficile et le code était beaucoup coûteux en termes de performance. On a implémenté de l'eau (mer, océan ...) qui est censé être en mouvement grâce aux fonctions Sinus et Cosinus, mais pour certaines raisons que nous cherchons toujours, cette eau reste statique.

6.6 Performance coûteuse

Notre code consommait beaucoup en performance et il arrivait que notre programme se crash après une certaine période. Nous avons essayé de régler ce problème en nettoyant notre code et en supprimant les fonctions ou variables inutiles.

6.7 Gestion du temps

Nous n'avons pas pu suivre exactement le planning que nous nous étions fixé à cause de plusieurs changements de situations. Ceci a ralenti l'avancement du projet et on était obligé d'annuler certaines fonctionnalités. Certaines tâches prenaient plus de temps que prévu.

6.8 GitHub

On a utilisé GitHub pour se partager le projet et suivre le déroulement et les avancements du projet. Au cours du projet il est arrivé qu'un jour on ne puisse plus faire des « commit » ou « push » car notre projet était sécurisé selon le message affiché sur la console. Ceci nous a pris 2 semaines pour éventuellement trouver la solution. Le problème avait été causé par un autre projet qu'on avait sécurisé en utilisant BitBucket, en sécurisant ce projet tous nos autres projets

qui utilisent GitHub ont été sécurisés. Ce problème a été résolu par une ligne de commande qui permet de configurer GitHub.

7 Améliorations

Notre application peut être améliorée pour la rendre plus effective. Voici quelques améliorations qui peuvent être apportées à notre projet :

Implémenter plusieurs terrains : Pour agrandir notre scène. Notre application est actuellement composée de 6 terrains ce qui rend le temps de parcours de toute la scène très petit. Des nouveaux terrains comme des montagnes et de la neige peuvent être intéressants à inclure.

Du son en temps réel : qui consistera à télécharger et implémenter du son selon l'environnement dans lequel l'on se trouve ou selon le matériel sur lequel l'utilisateur se déplace. Par exemple intégrer le son de l'eau ou de la mer selon sa distance par rapport à l'utilisateur.

Permettre à l'utilisateur de choisir le mode déplacement automatique ou manuel : On peut inclure une interface qui permettra à l'utilisateur de choisir s'il veut que le déplacement soit fait automatiquement ou manuellement.

Implémenter plus d'éléments dynamiques : Inclure des éléments comme des animaux qui se déplacent dans la scène, ceci pourra donner plus de vie à notre scène et évitera que l'utilisateur s'ennuie.

Utiliser des fractales : Nous avons utilisé des cylindres et sphères pour représenter nos arbres parce que c'était l'option la plus facile et rapide à implémenter en fonction du temps qu'il nous reste pour finir le projet mais l'utilisation des fractales aurait été la meilleure solution pour avoir des arbres qui se rapproche plus de la réalité.

Utiliser des personnages : On peut aussi créer un personnage qui va représenter le joueur qui se déplace dans notre monde au lieu d'utiliser la caméra. L'utilisation d'un personnage au lieu d'une caméra implique aussi la gestion des mouvements des membres du corps, ceci est la raison pour laquelle nous n'avons pas opté pour cette option vu le temps qu'il nous restait pour finir le projet, mais l'implication sera vraiment un bonus pour notre application

Plateforme multi-joueurs : On peut aussi améliorer notre monde virtuel en la rendant accessible par plusieurs joueurs en même temps pour qu'ils puissent interagir entre eux.

Menu de configuration : qui permettra à l'utilisateur de pouvoir paramétrer son monde comme il veut. On peut inclure des boutons qui pourront permettre à l'utilisateur de faire des Play, Pause et Stop et aussi paramétrer la vitesse du temps.

La gestion des obstacles : consistera à mettre en place un algorithme qui permet d'éviter des obstacles pour ne pas que le joueur les traverse. On devra donc définir les éléments qui sont traversables et non-traversables.

8 Conclusion

La réalisation de ce projet n'a pas été une tâche facile et nous en sommes conscient. Ce projet nous a demandé beaucoup de recherches, et beaucoup de connaissances en WebGL et en traitement d'images 2D et 3D. Ce projet nous a permis de tester notre niveau technique actuel et d'acquérir de nouvelles techniques et méthodes de programmation surtout en JavaScript.

Le fait de ne pas avoir d'expérience en WebGL nous a rendu la tâche plus difficile mais étant des étudiants sérieux que nous sommes, nous avons pu surmonter un grand nombre d'obstacle. Notre connaissance approfondie en JavaScript a rendu la réalisation de ce projet un peu plus facile. Nous pouvons toujours faire mieux si nous approfondissons nos connaissances sur les objets, classes et fonctions fournies par ThreeJS.

Malgré que nous n'avons pas assez bien gérer notre temps, nous avons au moins appris comment améliorer la gestion de temps ce qui nous servira pour les projets à venir.

Ce fut un plaisir pour nous d'avoir eu ce sujet comme projet et d'avoir pu réaliser ce projet. Nous comptons continuer de travailler dessus pour apporter des améliorations et ensuite le rendre accessible au publique

Références

- [1] <http://www.vrs.org.uk/virtual-reality/history.html>.
- [2] <http://collinhover.github.io/kaiopua/>.
- [3] <http://www.findyourwaytooz.com/>.
- [4] <http://www.mmogames.com/gamenews/18-online-virtual-world-3dchat-preview-nsfw-contain>