



École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Département Informatique
5ème année
2013 - 2014

Rapport projet d'option RV

Version 3D webgl de l'emploi du temps

Encadrants

Sébastien Aupetit
sebastien.aupetit@univ-tours.fr

École polytechnique de l'université de Tours
Tours - FRANCE

Étudiants

Ping XIANG
ping.xiang@etu.univ-tours.fr
Lei SHANG
lei.shang@etu.univ-tours.fr

DI5 2013 - 2014

Version du 4 mai 2014

Table des matières

1	Introduction	5
2	Travaux réalisés	6
2.1	Modélisation 3D du bâtiment	6
2.1.1	three.js	6
2.1.2	Mesure et collection des données de la modélisation	6
2.1.3	Conception de la scène	8
2.1.4	Difficultés rencontrées	8
2.2	Traitement des données de l'emploi du temps	9
2.2.1	Récupération de l'emploi du temps	9
2.2.2	Analyse et extraction des cours	12
2.2.3	Difficultés rencontrées	12
2.3	Plus court chemin	12
2.3.1	Implémentation de l'algorithme Dijkstra	12
2.3.2	Affichage et animation du chemin	13
2.4	Application créée	13
2.4.1	IHM	13
2.4.2	Publication de l'application	15
3	Conclusion	17

Table des figures

2.1	La mesure des coordonnées	7
2.2	Une salle trapézoïdale	7
2.3	Les salles en différents états	8
2.4	ENT	9
2.5	La requête envoyée depuis le navigateur	10
2.6	La réponse reçue qui contient les données de l'emploi du temps	10
2.7	Exportation du calendrier en URL	11
2.8	URL d'exportation générée	11
2.9	Le plus court chemin de la salle OC à Windows A : affichage	13
2.10	Le plus court chemin de la salle OC à Windows A : animation de la caméra	14
2.11	L'IHM de l'application	14
2.12	Plugin Chrome	15

Introduction

Ce rapport est développé à la suite du projet d'option Réalité Virtuelle : Version 3D WebGL de l'emploi du temps. L'objectif du projet est de visualiser en 3D notre emploi du temps des cours en l'affichant directement sur le modèle 3D de notre département, pour rendre ces informations plus représentatives.

Plus précisément, les fonctionnalités à réaliser sont :

1. Modélisation du bâtiment DI en modèle 3D
2. Importation de l'emploi du temps à partir d'un fichier ICS
3. Importation de l'emploi du temps directement depuis l'ENT
4. Analyse et extraction des données de l'emploi du temps
5. Visualisation des cours sur la scène 3D
6. Affichage du plus court chemin entre 2 salles

Ce projet est réalisé avec les technologies web notamment WebGL de manière à être utilisable depuis un navigateur. Toutes les fonctionnalités ci-dessus ont été réalisées sauf l'importation de l'emploi du temps depuis l'ENT. Ceci est prouvé infaisable à cause de la manque de l'information sur le format des données échangées. Par contre l'utilisateur peut très bien exporter soi même son emploi du temps et fournir l'URL exportée à l'application. Nous avons donc réalisé cette fonctionnalité de substitution.

En plus, nous avons réalisé des fonctionnalités supplémentaires pour rendre notre application encore plus conviviale, voici une liste non-exhaustive :

1. Recherche de la position des salles
2. Calcule et affichage du plus court chemin entre 2 salles choisies
3. Animation de la caméra le long le chemin pour une navigation guidée
4. Affichage des informations des cours en texte
5. Mise en ligne de notre site web
6. Création d'un plugin Chrome de l'application dans *Google Web Store*

Dans la partie suivante, nous allons expliquer en détail ce que nous avons réaslié ainsi que les difficultés rencontrées.

Travaux réalisés

2.1 Modélisation 3D du bâtiment

Dans cette section, nous présentons les travaux sur la mesure et la modélisation des composants du bâtiment comme les salles, les étages, les escaliers, et les points de contrôle pour calculer le plus court chemin.

2.1.1 three.js

Three.js est une bibliothèque JavaScript graphique 3D, elle est basée sur WebGL qui fournit `<canvas>`, `<svg>`, CSS3D et WebGL rendus. Nous l'utilisons pour modéliser notre bâtiment.

2.1.2 Mesure et collection des données de la modélisation

Mesure les coordonnées

Pour modéliser les composants, tout d'abord, on a besoin de les mesurer précisément. Nous avons trouvé les plans de chaque étage du bâtiment de l'école, puis nous avons choisi le coin en bas à gauche comme l'origine. Nous avons utilisé une règle pour mesurer la coordonnée de chaque composant en centimètres. Ensuite, nous les avons converties en pixels et calculé la longueur et la largeur de chaque salle.

La figure 2.1 est pour donner une idée sur notre travail est n'est pas difficile mais qui est coûteux au niveau du temps dépensé.

Stockage des coordonnées

Nous définissons une structure de données pour représenter chaque composant. Les différentes structures qui contiennent des informations nécessaires pour modéliser un composant sont les suivantes :

Salle

- id, identifiant de salles, de 0 à 61.
- numéro, un nombre qui représente le numéro de la salle
- nom, le nom des salles définies par l'école. Il va être apparu sur les murs des salles, et être utilisé pour trouver une salle.
- position, la coordonnée du point du centre de salle.
- taille, un vecteur contient la longueur, la largeur, et la hauteur. La hauteur est définie par nous.
- format, les salles de l'école disposent de deux formes, carré et trapézoïdale. Pour les salles carrées, nous pouvons dessiner directement une géométrie de cube ; et pour les salles trapézoïdales, selon la position et la taille, nous pouvons calculer les coordonnées de chaque vertex, et dessiner 2 triangles par face, au total 12 triangles constituant une trapézoïdale.
- direction, les salles de l'école ont 3 directions : horizontale, verticale et oblique. Après la création des salles, nous faisons la rotation pour les faire correspondre la direction donnée.

Etage et escalier Les étages et escaliers sont construits par des faces. Nous mesurons et fixons les vertex des faces, et choisissons le matériel.



FIGURE 2.1 – La mesure des coordonnées

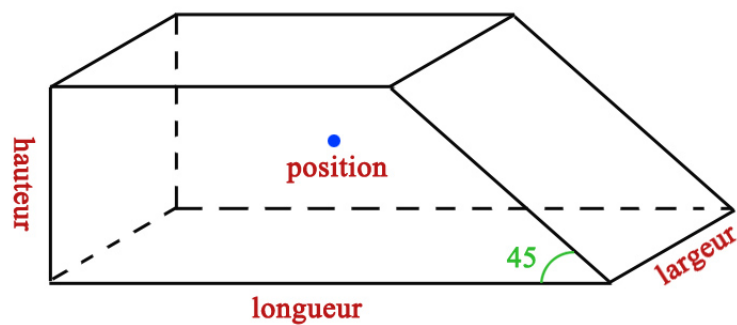


FIGURE 2.2 – Une salle trapézoïdale

Points clés pour calculer le plus court chemin

- id, un nombre unique utilisé pour identifier les points. De 0 à 61, les points sont positionnés dans les couloirs devant les salles ; de 70 à 103, les points supplémentaires placés sur les escaliers, tournant, entrée, etc, utilisés pour garantir la naturalité du chemin. Par exemple, on ne doit pas laisser le chemin traverser un mur.
- position, les coordonnées des points.
- direction, le même sens que pour les salles.

2.1.3 Conception de la scène

Lorsque ce projet a été proposé, nous espérons que nous pouvons montrer les locations des cours plus clairement, pour guider les étudiants à trouver rapidement l'emplacement de la classe. Par conséquent, nous avons décidé les choses suivantes afin de répondre à notre concept :

- utiliser une couleur claire pour les salles ;
- utiliser les couleurs suivantes pour distinguer les salles en les différents états (figure 2.3) :
 - rouge, représente la salle recherchée
 - vert, représente la salle du prochain cours
 - bleu, représente la salle en cours en ce moment
 - gris foncé, représente la salle dont le cours a déjà fini
- afficher le nom des salles sur le mur de chacune pour montrer plus clairement les positions ;
- ajouter le sol du premier étage pour éviter le chevauchement excessif de salle ;
- L'élargissement de la position des escaliers pour fournir une meilleure perspective.

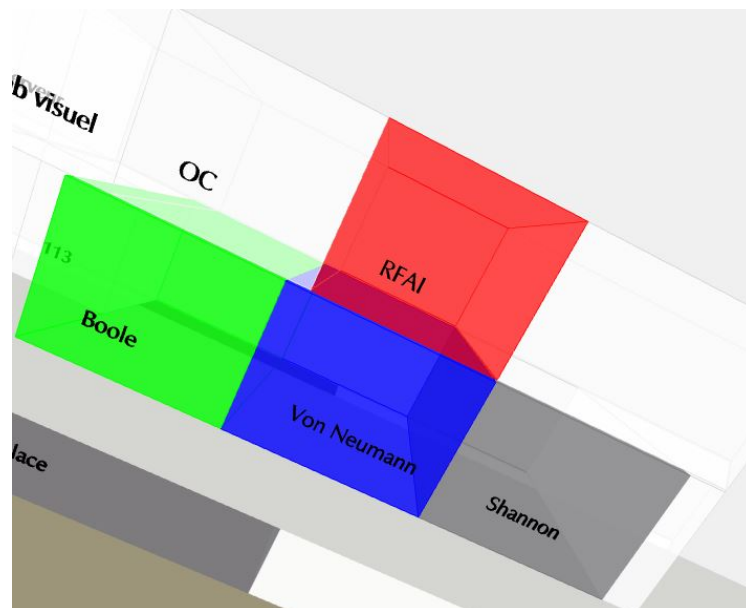


FIGURE 2.3 – Les salles en différents états

2.1.4 Difficultés rencontrées

Le premier problème rencontré est que la création du modèle de l'école est très chronophage. Nous devons mesurer chaque salle précisément afin de modéliser plus réellement le modèle 3D, nous avons dépensé beaucoup de temps sur celui-ci. En même temps, la documentation de three.js n'est pas très précise. Pour de nombreuses classes et fonctions, il n'y a pas de documentation, ni d'exemple, nous étions souvent obligés d'entrer dans le code source pour nous débrouiller.

2.2 Traitement des données de l'emploi du temps

Dans cette partie nous présentons le travail sur la récupération et l'analyse des données de l'emploi du temps. Les informations extraites des cours seront intégrées dans le modèle 3D de notre bâtiment pour enfin montrer à l'utilisateur.

2.2.1 Récupération de l'emploi du temps

Les données des cours de l'école se présentent sur le site ENT¹ de l'université.



FIGURE 2.4 – ENT

Récupération directe à partir de l'ENT

Le site ENT est une application web assez lourde qui regroupe les services informatiques de tous les étudiants de l'université. La récupération de l'emploi du temps à partir de cette application n'est pas évidente.

Dans un premier temps nous avons analysé le flux transféré entre le site ENT et le navigateur pour trouver comment les données de l'emploi de temps sont transférées. Nous avons enfin trouvé des morceaux de ces informations dans plusieurs tableaux json qui contiennent chacun énormément d'enregistrements. En fait, ce qui est transmis n'est pas seulement les données de l'emploi du temps, mais aussi un grand volume de données d'ailleurs, puisque c'est une page web riche.

La figure 2.5 est la requête que nous avons captée, qui est envoyée par le navigateur pour demander les données. La méthode d'envoi est *POST*, le charge de la requête contient beaucoup de chiffres dont nous ne comprenons pas le sens.

La figure 2.6 est la réponse du serveur. Il y a plein de chiffres encodés. Même si nous avons trouvé des morceaux de données qui nous intéressent, nous n'avons pas pu trouver un moyen pour extraire ces données

1. Environnement Numérique de Travail, <http://edt.univ-tours.fr/direct/myplanning.jsp?login=ade-etudiant&password=test>

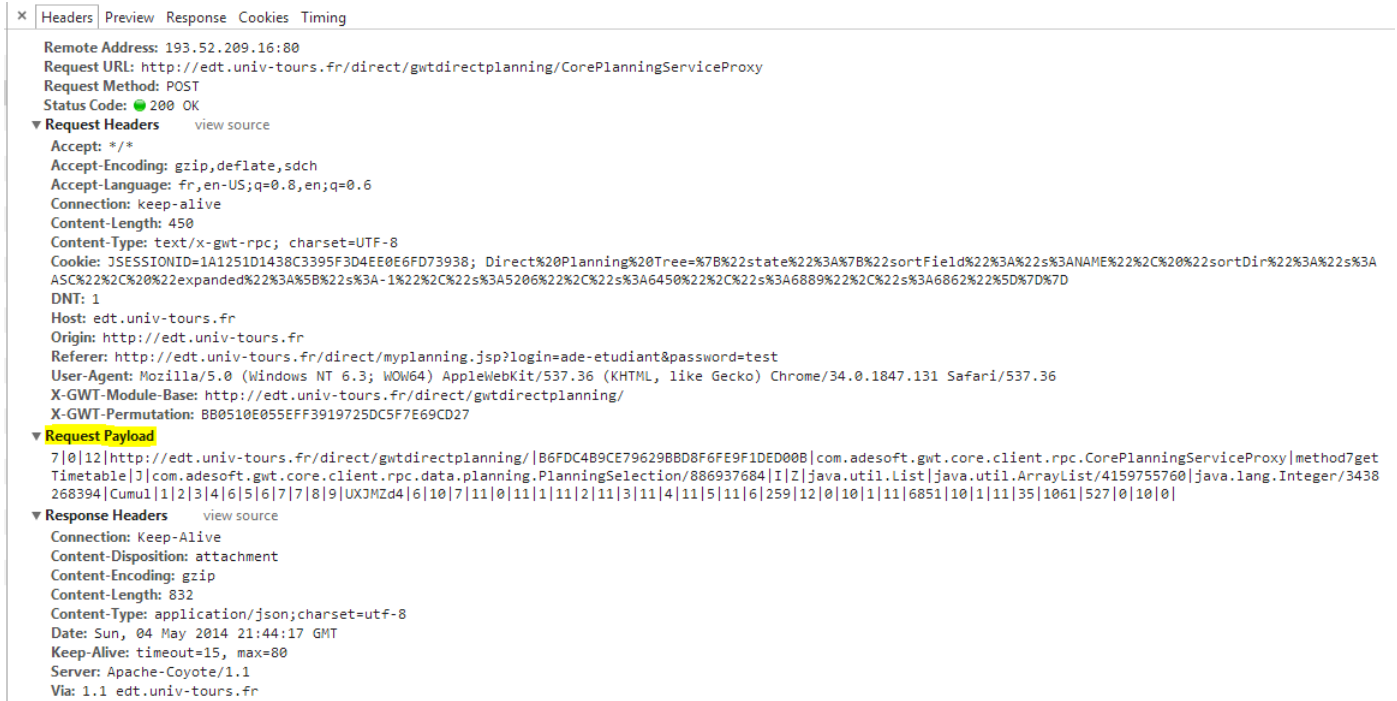


FIGURE 2.5 – La requête envoyée depuis le navigateur

en analysant la réponse du serveur car il n'y a pas assez d'information. L'extraction directe de l'emploi du temps à partir de l'ENT devient donc infaisable pour nous. Nous avons ensuite basculé à d'autres méthodes de récupération.

```
1 //OK[347,605,152,84,0,4,4,0,4,4,0,4,5,0,34,5,0,33,5,0,17,5,0,32,5,10,1,1,0,20,118033,3,1,1,255,255,255,255,102,102,4,13461,3,1,1,2,252,605,152,84,0,4,4,0,4,5,0,31,5,0,29,5,0,14,5,0,30,5,10,1,1,0,4,3
2 0259,3,1,1,255,255,255,255,153,153,4,13517,3,1,1,2,10,605,152,84,0,4,4,0,4,5,0,22,5,0,29,5,0,7,5
3 0,28,5,10,1,1,0,4,29844,3,1,1,255,255,255,255,204,204,4,23701,3,1,1,2,252,302,152,84,0,4,4,0,4,
4 5,0,25,5,0,27,5,0,14,5,0,26,5,10,1,1,0,4,118003,3,1,1,255,255,255,255,153,153,4,7293,3,1,1,2,105,3
5 02,152,84,0,4,4,0,4,5,0,9,5,0,8,5,0,11,5,0,6,5,10,1,1,0,4,118271,3,1,1,255,255,255,255,153,153,4
6 13446,3,1,1,2,10,302,152,84,0,4,4,0,4,5,0,25,5,0,15,5,0,7,5,0,13,5,10,1,1,0,4,114950,3,1,1,255
7 255,255,255,153,153,4,13422,3,1,1,2,347,151,151,84,0,4,4,0,4,5,0,24,5,0,21,5,0,17,5,0,23,5,10,1
8 1,0,20,118751,3,1,1,255,255,255,255,102,102,4,32697,3,1,1,2,252,151,151,84,0,4,4,0,4,5,0,19,5,0
9 18,5,0,14,5,0,16,5,10,1,1,0,20,114812,3,1,1,255,255,255,255,102,102,4,13423,3,1,1,2,105,151,151,84
10 0,4,4,0,4,5,0,22,5,0,21,5,0,11,5,0,20,5,10,1,1,0,20,118749,3,1,1,255,255,255,255,204,204,4,1348
11 1,3,1,1,2,347,0,151,84,0,4,4,0,4,5,0,19,5,0,18,5,0,17,5,0,16,5,10,1,1,0,20,114798,3,1,1,255,255
12 255,255,102,102,4,13423,3,1,1,2,252,0,151,84,0,4,4,0,4,5,0,9,5,0,15,5,0,14,5,0,13,5,10,1,1,0,4,1
13 14948,3,1,1,255,255,255,255,153,153,4,13422,3,1,1,2,105,0,151,84,0,4,4,0,4,5,0,9,5,0,12,5,0,11,5
14 0,10,5,10,1,1,0,20,16337,3,1,1,255,255,255,102,255,102,4,30870,3,1,1,2,10,0,151,84,0,4,4,0,4,5,
15 0,9,5,0,8,5,0,7,5,0,6,5,10,1,1,0,4,118299,3,1,1,255,255,255,255,153,153,4,13446,3,1,1,2,13,1,
16 ["java.util.ArrayList/4159755760",
17 "com.adesoft.gwt.core.client.rpc.data.planning.SquareEvent/3694954416",
18 "java.lang.Integer/3438268394","", "java.lang.String/2004016611",
19 "Prog. Math. TD_G1", "08:15 - 10:15", "BOUQUARD JEAN-LOUIS", "Von Neumann",
20 "Statistique Inférentielle_TD_G1_JCB", "10:30 - 12:30", "BOURGOIN JEAN-CHRISTOPHE",
21 "Microcontrôleur_TD_G1_PMa", "14:00 - 16:00", "MAKRIS PASCAL",
22 "Microcontrôleur_TP_G1_CF", "16:15 - 18:15", "FAUCHEUX CYRILLE", "TP Systèmes",
23 "Programmation WEB_CM", "GIRARD NATHALIE", "Pascal",
24 "Programmation WEB_TP_G1", "Windows B",
25 "S 228-229", "Anglais_G1_JS", "SEN JUPITER",
26 "Statistique Inférentielle_CM_MS", "SLIMANE MOHAMED",
27 "Statistique Inférentielle_TD_G1_MS", "Shannon",
28 "Princip. & Normes Réseaux_TP_G1_PB", "BOURQUIN PASCAL", "Chaîne Production"]
29 ,0,7]
```

FIGURE 2.6 – La réponse reçue qui contient les données de l'emploi du temps

Récupération des données par URL d'exportation

Nous avons constaté que l'ENT fournit l'exportation de l'emploi du temps en URL. On doit d'abord choisir sa spécialité puis la période à exporter. En sortie on obtiendra une URL à partir duquel un fichier ICS² qui contient les informations des cours peut être téléchargé.

Nous avons utilisé ce moyen pour exporter l'emploi du temps dans notre propre calendrier comme Google Agenda, donc nous pensons que c'est possible que nous fassions la même chose, c'est-à-dire récupérer les données depuis une URL d'exportation.

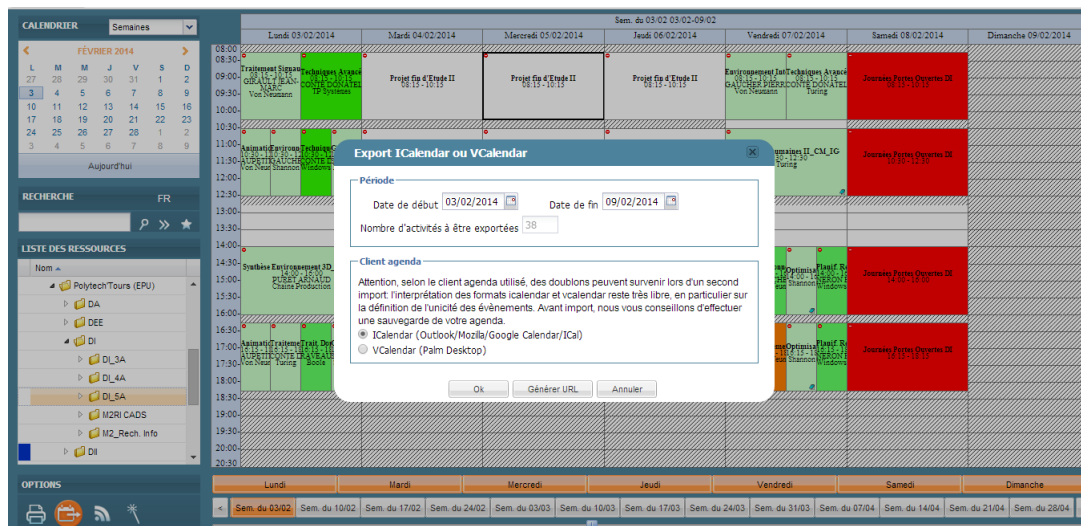


FIGURE 2.7 – Exportation du calendrier en URL

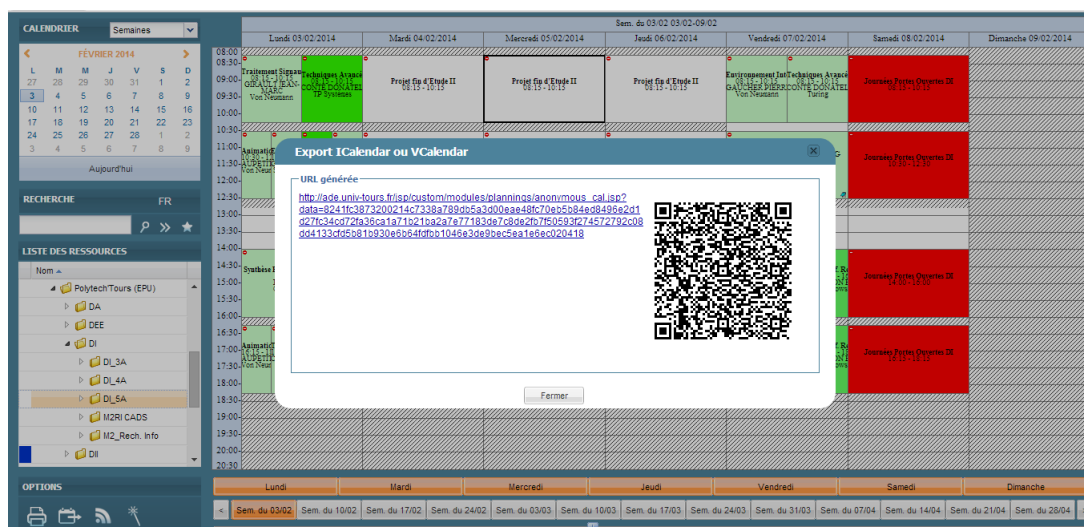


FIGURE 2.8 – URL d'exportation générée

Pour le faire, nous demandons l'utilisateur de générer d'abord cette URL sur l'ENT et puis le fournir à notre application pour la première utilisation. Nous allons ensuite enregistrer cette URL et synchroniser l'emploi du temps à chaque ouverture de l'application.

2. ICalendar, standard RFC 5545

Nous avons rencontré des difficultés pour le téléchargement du fichier ICS. Ce sera expliqué dans la section difficultés rencontrées 2.2.3.

2.2.2 Analyse et extraction des cours

Après l'obtention des données de format ICS, il nous reste à analyser ces données et d'extraire les cours selon la date choisit. Nous avons d'abord étudié le format ICS, ensuite nous avons créé des expressions régulières qui nous permettent de trouver et extraire les informations qui nous intéressent.

Ce travail a été réalisé avec JavaScript, en sortie de cette analyse on a un tableau des événements (cours) qui sont des objets avec les attributs comme : l'horaire du cours, le nom et le lieu du cours, etc. Ces objets seront intégrés après dans le modèle 3D.

2.2.3 Difficultés rencontrées

La principale difficulté dans cette partie de travail concerne la récupération directe des données depuis l'ENT. Nous avons mis beaucoup de temps pour chercher un moyen, mais finalement nous n'avons pas réussi.

La récupération depuis URL d'exportation nous a posé aussi des problèmes. En fait nous avons voulu au tout début, de rendre notre application web purement locale, c'est-à-dire tout le travail de notre application serait réalisé par JavaScript donc pas besoin de mettre en place un serveur. Malheureusement le comportement du téléchargement *cross-domain* de données n'est pas autorisé par défaut par des navigateurs (Chrome pour nous) à cause des soucis de sécurité. Souvent nous pouvons demander le navigateur à l'autoriser en modifiant l'option concernée, mais ça rend notre application beaucoup moins conviviale et surtout ça va exposer nos chers utilisateurs sous danger.

Finalement nous étions obligés de créer la côte serveur en PHP de l'application, pour éviter un téléchargement *cross-domain*. En plus, pour donner plus de choix et de meilleures expériences aux clients, sans jamais oublier notre rêve de *localisation*, nous avons aussi créé un plugin Chrome³ de l'application qui est déjà mis en ligne dans *Chrome Web Store*.

2.3 Plus court chemin

La recherche du plus court chemin est un problème très classique, mais toujours intéressant. Dans ce projet nous avons pu appliquer l'algorithme *Dijkstra* pour chercher le plus court chemin entre 2 salles de l'école.

Notre idée est de positionner dans le modèle des points clés qui sont des noeuds du graphe, pour afficher le chemin, nous pouvons donc relier les points qui composent le chemin.

2.3.1 Implémentation de l'algorithme Dijkstra

Pour appliquer l'algorithme de la recherche du plus court chemin, il faut d'abord modéliser notre problème en une structure de graphe en JavaScript. Nous avons alors créé une class *Graphe* qui peut représenter un graphe et qui contient des méthodes nécessaires pour la création du graphe.

Après étudier la particularité de notre problème et l'algorithme Dijkstra, nous avons choisi de stocker les noeuds dans une liste et pour chaque noeud nous maintenons une liste des noeuds voisins. De cette manière, l'implémentation de l'algorithme devient plus simple et sa performance peut être garantie.

Finalement le graphe de notre bâtiment contient centaine sommets, ce qui est un graphe plus grand que prévu.

3. Les applications Chrome peuvent avoir des droits supplémentaires comme la requête cross-domain dans notre cas

2.3.2 Affichage et animation du chemin

Affichage

A chaque fois l'emploi du temps se charge, l'application va chercher, selon l'horaire en ce moment, le cours suivant à assister. Ensuite elle applique l'algorithme pour trouver le plus court chemin entre la salle du cours en cours (ou le point d'entrée s'il n'y a pas de cours en ce moment) et la salle du cours suivant. Ce chemin est représenté d'un tableau des indices des points clés. Nous créons ensuite à l'aide de *three.js*, un *SplineCurve* basé sur ces points clés et puis un *TubeGeometry* qui est le chemin final.

Pour bien afficher le chemin, il y a énormément de points à positionner : à part des points devant chaque salle, il faut aussi ajouter des points intermédiaires pour assurer que le chemin est bien *naturel*, par exemple il faut mettre des points sur les tournants pour que le chemin ne traverse pas le mur.



FIGURE 2.9 – Le plus court chemin de la salle OC à Windows A : affichage

Animation

Puisque notre bâtiment est de forme d'un triangle, ce qui est un peu fermé pour avoir une bonne vue du chemin affiché. Nous pensons donc de "*guider*" l'utilisateur le long du chemin, c'est-à-dire d'animer la caméra pour qu'il se déplace le long du chemin. Nous avons aussi affiché en 3D le numéro des salles à côté du chemin, donc pendant le parcours on peut toujours se percevoir sa position dans le bâtiment. (cf Figure 2.10)

2.4 Application créée

Afin de rendre mieux l'IHM du site, nous le divisons en deux parties avec l'étiquette `iframe` : la partie gauche est le panel de configuration, réalisé en utilisant `bootstrap` ; la partie droite affiche notre modèle 3D.

2.4.1 IHM

L'IHM de notre application contient deux parties : un panel de configuration à gauche et la vue 3D du bâtiment à droite.

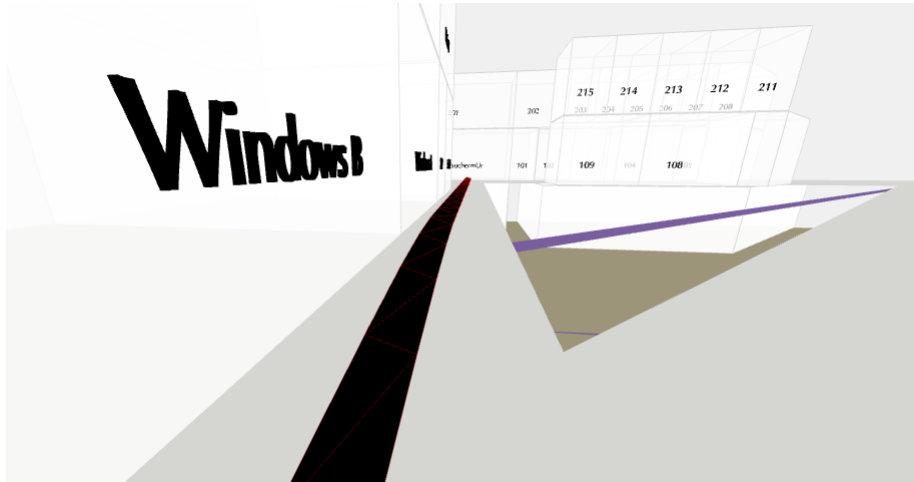


FIGURE 2.10 – Le plus court chemin de la salle OC à Windows A : animation de la caméra

EDT3D Emplois du temps 3D

Ongoing Next Finished

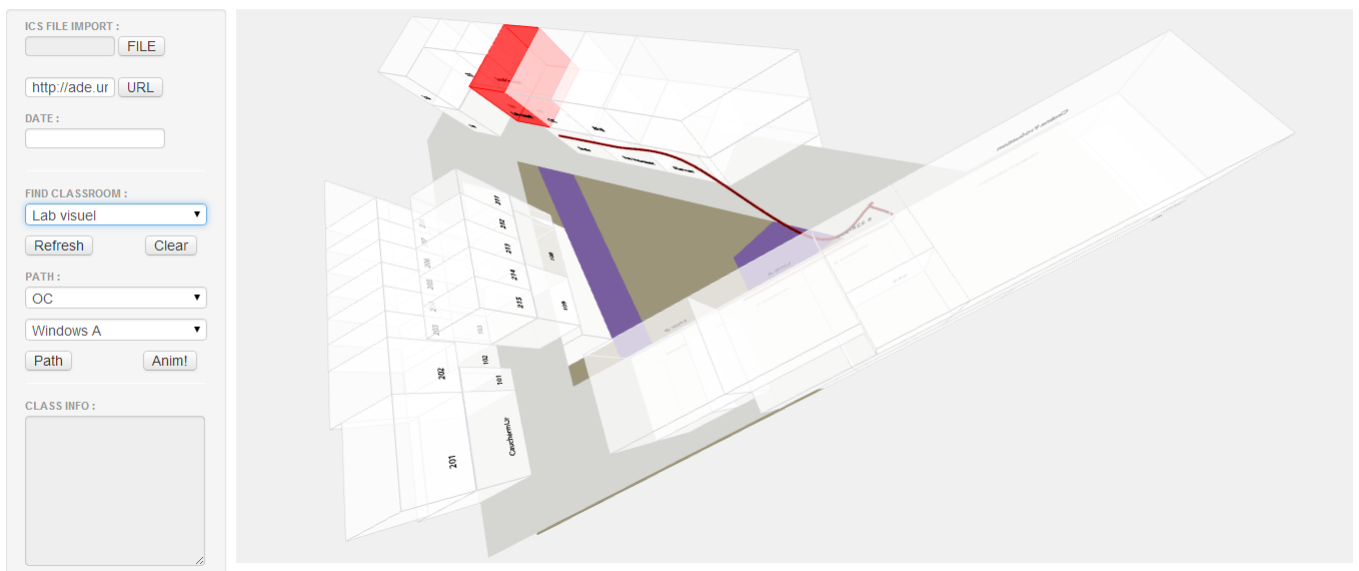


FIGURE 2.11 – L'IHM de l'application

Panel de configuration

Les fonctionnalités dans le panel à gauche sont :

- **Choix du fichier ICS** : importation de l'emploi du temps depuis un fichier en format ics.
- **Saisie de l'URL ICS** : importation de l'emploi du temps depuis une URL.
- **Date** : après l'importation, ce champ sera activé qui permet de choisir une date dont les cours seront affichés dans la vue.
- **Recherche de salles** : pour trouver la position d'une salle, on peut choisir la salle dans la liste, elle sera indiquée dans le modèle 3D.
- **Path** : afficher le plus court chemin entre 2 salles choisies. Le bouton *Path* permet de calculer et d'afficher le chemin, le bouton *Anim !* va lancer l'animation de la caméra selon le chemin précalculé.
- **Classinfo** : afficher en texte les cours de la date choisie. La mise en place de ce champ est pour traiter le problème que pour certains cours la salle n'est pas indiquée dans l'emploi du temps, par exemple pour le *Projet d'Option*. Dans ce cas-là, nous ne pouvons bien sûr pas afficher ce cours dans la vue 3D.

L'apparence de cette partie est créée à l'aide de Bootstrap qui est un cadre frontal élégant, intuitif et puissant pour le développement web rapide et plus facile. Il est créé par Mark Otto et Jacob Thornton, et maintenu par l'équipe de base avec le soutien massif et la participation de la communauté.

Navigation dans la vue 3D

Pour naviguer dans la vue 3D :

- **Rotation** : trainer avec le bouton gauche de la souris.
- **Translation** : trainer avec le bouton droit de la souris.
- **Zoom** : la molette de la souris.

2.4.2 Publication de l'application

Grâce à la difficulté rencontrée 2.2.3 pendant la récupération des données, nous avons enfin créé trois versions de notre application.

Application web serveur

Cette version est une application web en PHP. Il faut la mettre sur un serveur pour être accessible par les utilisateurs. Pour rendre notre projet *complet*, nous avons mis en ligne notre application. Elle est accessible depuis cette URL : <http://edt3d.site40.net/>, mais l'accessibilité ne sera pas toujours garantie.

Application Chrome

Cette version est un plugin Chrome qui peut être installé directement dans le navigateur Chrome par les utilisateurs. Ce plugin est déjà mis dans le *Chrome Web Store* d'où tout le monde peut le télécharger <https://chrome.google.com/webstore/detail/edt3d/deidljgkpcpmknbcempgmnljjmdodlb?hl=fr>. Une icône de l'application apparaîtra dans la barre des plugins.



FIGURE 2.12 – Plugin Chrome

Application web locale

Cette version est la version initiale qui a posé le problème de sécurité. Mais à part de ce risque, c'est quand même une version utilisable donc tout dépend le choix de clients.

Conclusion

Ce projet nous a permis de pratiquer les techniques de Réalité Virtuelle sur un problème réel. L'application que nous avons créée peut aider les nouveaux étudiants à facilement trouver la position des salles et même le plus court chemin pour assister un cours. C'est très pratique et très proche de notre vie, donc ça nous fait sentir que notre travail et les techniques 3D que nous avons appris ont de la valeur.

Au niveau technique, nous avons pu bien connaître un outil JavaScript de développement 3D : *three.js*, nous avons aussi maîtrisé le développement de plugin Chrome.

En un mot, ce projet nous a beaucoup plu et nous a renforcé les connaissances sur la réalité virtuelle.

Version 3D webgl de l'emploi du temps

Département Informatique
5ème année
2013 - 2014

Rapport projet d'option RV

Résumé : Ce rapport est rédigé à la suite d'un projet d'option Réalité Virtuelle. Dans le projet nous avons réalisé une application qui peut afficher l'emploi du temps d'un étudiant DI sur le modèle 3D du bâtiment. L'application peut aider les nouveaux étudiants à trouver une salle et même le plus court chemin entre deux salles. Une animation de navigation peut aussi lancer le long du chemin pour guider l'utilisateur.

Mots clefs : Réalité virtuelle, Emploi du temps 3D, Plus court chemin, Three.js, Plugin Chrome

Abstract: This report is edited for a project in the option Réalité Virtuelle. In this project we have implemented an application which can show students' class schedule on the 3D model of our teaching building DI. It helps new students to find easily the position of classrooms and even the shortest path from one room to another. A guiding tour animation along the path can be launched as well.

Keywords: Virtual reality, 3D class timetable, Shortest path, Three.js, Chrome plugin

Encadrants

Sébastien Aupetit
sebastien.aupetit@univ-tours.fr

École polytechnique de l'université de Tours
Tours - FRANCE

Étudiants

Ping XIANG
ping.xiang@etu.univ-tours.fr
Lei SHANG
lei.shang@etu.univ-tours.fr

DI5 2013 - 2014