haute école
neuchâtel berne jura
arc ingénierie
saint-imier le locle delémont

ARC3D

Travail de Bachelor - 16dlm-tb-219

Thomas Roulin

encadrement pédagogique par
Stéphane Gobron

July 12, 2016

Résumé

Le Campus Arc 2 de la Haute-École Arc, (HE-Arc), Neuchâtel, HES-SO, est un bâtiment de grande envergure dont certaines zones ont dû être sécurisées. Des open-spaces, dont l'accès est réglementé et limité, engendrent des problèmes tant pour les visiteurs que pour les étudiants. Il est régulièrement difficile de savoir, selon le type d'utilisateur, quel chemin emprunter. Un outil facilitant les déplacements, en permettant la visualisation et le tracé du chemin pour se rendre en tout lieu et en toute salle du campus, pourrait résoudre ce problème. Ce rapport décrit le développement dudit outil, lequel facilite les déplacements à l'intérieur du bâtiment mais englobe également les déplacements depuis la gare au campus.

La solution apportée se présente sous la forme d'une page Internet accessible depuis un smartphone ou un ordinateur. La technologie utilisée est celle du WebGL, ce qui évite tous les problèmes de librairies externes ou de plugins.

Abstract

Karim halp me pls.

Table des matières

Chapitre 1

Introduction

1.1 Problématique générale

Au XXIe siècle, tout va toujours plus vite, l'être humain n'entend pas perdre de temps inutilement et, notamment, pas pour chercher son chemin. Pour se déplacer sur un site d'une certaine complexité et d'une certaine envergure, il convient d'offrir une aide au déplacement, sous la forme d'une solution rapide et facile d'accès. L'utilisation de la 3D en navigateur n'est que très peu répandue pour l'instant, mais elle pourrait tout à fait répondre à ce besoin. Proposer un tel outil, innovant et performant, aurait également l'avantage de répondre à une certaine ambition du domaine Ingéniérie de l'HE-Arc de Neuchâtel, en matière de visualisation en temps réel.

1.2 Contextualisation

Dans le cadre du Campus Arc 2, le deuxième étage est considéré comme un open space. La particularité est que différents secteurs sont fermés aux visiteurs et élèves. Outre sa dimension, c'est une des causes principales de problèmes de déplacements à l'intérieur du bâtiment. Il est ainsi fondamentalement intéressant d'offrir un outil pour faciliter les trajets à l'intérieur du bâtiment et depuis la gare.

Chapitre 2

Analyse

2.1 État de l'art

Avant de démarrer le développement il est important de se renseigner sur les travaux qui ont déjà été effectués concernant différents objectifs du projet. Le but est de voir si des recherches ou outils déjà créés pourraient nous aider à développer notre projet.

3D temps réel en navigateur

Dans le cadre ce projet il n'est pas possible de pré-rendre les différents chemins imaginables. C'est pourquoi nous devons nous orienter vers la 3D en temps réel. Le but de cette dernière est de pouvoir rendre des images assez vite pour que l'œil humain ait l'impression que ce soit fluide. Ceci implique de rendre au minimum 30 images par seconde, voir 60 au mieux, plus n'est pas utile.

La technologie propice WebGL est maintenant supportée par la majorité des navigateurs, mobile compris [1]. Elle offre un pont entre notre page internet et la carte graphique de l'utilisateur, c'est grâce à cela qu'il est possible d'offrir ce nombre d'images par seconde.

Cependant le langage a utiliser est très primitif, c'est pourquoi il est intéressant d'avoir une architecture permettant de gérer notre scène. Plusieurs bibliothèques WebGL offrent déjà des infrastructures ainsi que différents outils qui simplifient des tâches qui seraient redondantes et n'apporteraient rien au projet.

Le choix de la bibliothèque s'est porté sur *three.js*[2] car c'est celle qui est la plus développée et mise à jour tout en offrant un grand nombre de fonctionnalités. Elle gère notamment le chargement de modèles et permet aussi d'effectuer par exemple du *Back-face culling* et du *Frustum culling*[3].

Navigation dans un bâtiment

La navigation en intérieur alimente bon nombre de recherche, c'est un sujet sur lequel il n'y a pas de solution parfaite. Il y a plusieurs manières d'offrir des résultats plus ou moins précis pour répondre à ce problème. La section 2.2 parle plus en profondeur de ce thème et décrit les différentes approches expérimentées.

Visualisation architecturale

Beaucoup de visualisations architecturale existantes se contentent de pré-générer une vidéo dans un bâtiment. De ce fait, les technologies utilisées ne sont pas les mêmes, ce n'est pas du temps réel. D'autres optent pour une série de photographies parmi lesquelles il est possible de naviguer.

Il existe malgré tout plusieurs produits de visualisation en temps réel. Cependant la plupart de ces derniers demandent un logiciel installé sur le client et de plus ne sont pas utilisable sur mobile. Dans les rares qui utilisent WebGL il y a par exemple Shapespark [4], qui n'est encore qu'au stade de *pre-release* et a plutôt l'air d'un service qui offre un produit fini utilisable en navigateur, donc rien d'utilisable pour le développement. PlayCanvas [5] permet aussi d'avoir une application en navigateur, cependant l'outil est un moteur 3D qui propose bien plus de fonctionnalités que nous avons besoin et surtout est payant.

2.2 Localisation en intérieur

Le plus grand défi de ce projet est la localisation intérieure de l'utilisateur, par là le degré de précision de cette localisation. De plus, il est intéressant et agréable pour l'utilisateur de disposer d'un suivi régulier de sa position durant la visualisation 3D. Cette section explique les diverses pistes empruntées ainsi que leurs résultats.

GPS

Qui dit localisation pense GPS (*Global Positioning System*),système qui pourrait, à première vue, présenter une bonne solution. Cependant, cette technologie manque de précision, principalement en intérieur. La précision des GPS mobiles se situerait à peu près entre 4 et 50 mètres, dépendant des conditions et du smartphone utilisé. En général, l'erreur est due aux éléments entre l'utilisateur et le satellite, c'est pourquoi la localisation en intérieur ne peut pas se fier au GPS.

Triangulation Wifi

Une méthode utilisée pour la localisation intérieure est la triangulation / trilatération, en utilisant les *Access Points* WiFi. Le concept consiste à retenir à l'avance les coordonnées de tous les AP. Chaque AP possède une adresse MAC qui les différencie les uns des autres. Ensuite, il faut scanner les AP depuis l'objet que l'ont veut localiser et, en fonction des forces des AP, il est possible de retrouver la position.

L'avantage de cette solution est l'absence d'infrastructures à mettre en place, étant donné que le Campus est déjà équipé de bon nombre d'AP. Cependant le but de ce projet est d'offrir une application dans un navigateur Internet. Ainsi, cette contrainte empêche l'accès à toutes les informations de l'appareil concerné par son utilisation. Malgré les technologies qu'offre ce mode, il est encore impossible de récupérer les informations nécessaires à la triangulation / trilatération depuis une page web.

Triangulation Bluetooth

Une autre manière d'utiliser la triangulation est d'installer des beacons bluetooth. Il s'agit de petits émetteurs placés dans les espaces dans lesquels une localisation est nécessaire. Le fonctionnement de la triangulation est identique au WiFi.

Malheureusement, la mise en place de cette infrastructure est un désavantage à cette solution et, de plus, ces accès bluetooth ne sont pas suffisamment supportés depuis une page web. Le tableau ci-dessous 2.1, tiré du site www.caniuse.com, montre que seul Chrome peut le supporter, tout en précisant qu'il faut activer le drapeau nécessaire. Cela démontre que l'API Web Bluetooth n'est pas utilisable pour une application destinée au grand public.

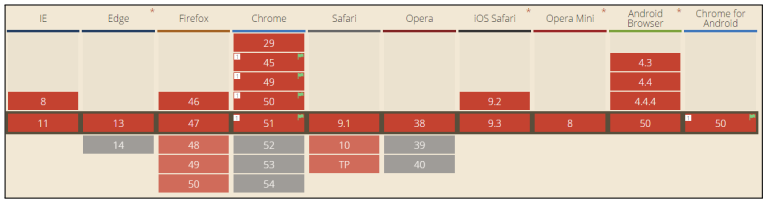

FIGURE 2.1 : Support de l'API Web Bluetooth par les différents navigateurs.

Accéléromètre et Gyroscope

L'approche par accéléromètre et gyroscope est différente, car elle implique d'utiliser les capteurs internes du smartphone pour calculer une position. HTML5 fournit une API pour détecter l'orientation et les déplacements du dispositif. Autrement dit, l'accès à l'accéléromètre et au gyroscope est possible depuis un navigateur Internet.

A l'aide de l'accélération, il est théoriquement possible de calculer une position. Cependant, les valeurs de ces capteurs comportent un bruit et, pour calculer une position, il faut double-intégrer l'accélération. Cela implique qu'il faut double-intégrer le bruit et cela va créer un *drift*. En quelques secondes, l'erreur peut s'élever à une vingtaine de centimètres. En outre, le gyroscope peut avoir une valeur erronée, due à la suppression de la gravité. En d'autres termes, une erreur d'un degré sur la valeur du gyroscope représente plusieurs mètres d'erreur en quelques secondes.

Citer https ://youtu.be/C7JQ7Rpwn2k ?t=23m22s

Le but de cette approche était fondamentalement d'estimer la position de l'utilisateur, sans pour autant connaître la valeur exacte. Les démarches et réflexions prouvent qu'une approximation n'est pas envisageable avec ces capteurs.

Chapitre 3

Développement

3.1 Modèle 3D

Explication des différents problèmes rencontrés avec le modèle, son texturing, son exportation et importation.

Modélisation géométrique

Citer Kevin ? Seulement les plans de sols et pas de coupe. Approximation par ratio des hauteurs.

Texturisation du modèle

Pourquoi ? Contextualisation. se rendre compte qu'on est dans l'école Application des textures. Erreur export de fichier, modification du script d'exportation.

Exportation et importation

Type d'objet utilisé.'Contraintes/avantages.

3.2 Rendu graphique

Matériaux de Lambert Éclairage.

3.3 Recherche de chemin

Nœuds et Graphe

Format Placement dans l'espace

Algorithme de recherche de chemin

A-Star

3.4 Contrôles de la caméra

Ordinateur et mobile

Les différences à gérer

Orientation mobile

Utilisation du gyroscope

3.5 Suivi caméra

Les méthodes que nous avons décidé d'implémenter, à savoir *Live Mode* et *Simulation mode*.

Lissage du chemin

Catmull Rom

Regard de la caméra

Comment la caméra sait où regarder.

3.6 Interface graphique

Toucher un mot sur l'interface graphique.

3.7 Rendu temps réel

je l'ai testé sur 3 trucs. minimum 30 fps

Chapitre 4

Résultats

Lister les objectifs, lesquels sont réussi, pourquoi les autres le sont pas...

4.1 Temps réel

4.2 Modélisation géométrique

4.3 Réalisation des objectifs

Chapitre 5

Discussion

5.1 Conclusion

5.2 Perspectives

Objectifs pas réussis : Comment les réussir.
Améliorations possibles.
Pas de perspectives faciles.

Chapitre 6

Bibliographie

Bibliography

[1] *Can I Use - WebGL*. URL: http://caniuse.com/#feat=webgl (visited on 07/12/2016).

[2] *three.js*. URL: http://threejs.org/ (visited on 07/12/2016).

[3] *Hidden surface determination*. URL: https://en.wikipedia.org/wiki/Hidden_surface_determination (visited on 07/12/2016).

[4] *Shapespark*. URL: http://www.architectureanddesign.com.au/news/industry-news/architects-gear-up-for-real-time-visualisation (visited on 07/12/2016).

[5] *Playcanvas*. URL: https://playcanvas.com/industries/architecture (visited on 04/12/2016).