INTRODUCTION

Notre projet tutoré de deuxième année de DUT Informatique consiste en la réalisation d’une visite virtuelle de l’IUT Informatique de Clermont-Ferrand ainsi que d’un générateur de panorama en ligne. Il a été réalisé sur une durée de 5 mois, du 9 novembre au 29 mars.

Notre équipe est composée de Clément Ferrere, Enzo Mazella, Victor Mommalier, Clara Poncet et Lucile Velut. Ce projet a été proposé et supervisé par notre tuteur M. Salva, professeur de l’IUT.

Le projet est composé de deux parties et mettra en avant les panoramas sous forme d’application web. Un panorama est défini comme un enchainement de scènes composées de photos à 360°. Ces scènes seront accompagnées de panneaux informatifs, de points de navigation permettant de circuler d’une scène à l’autre, et d’une carte interactive facilitant les déplacements et apportant une vue d’ensemble du panorama. Ce projet a été réalisé grâce au Framework A-Frame qui permet de créer du contenu web en réalité virtuelle.

La première partie de notre projet est une visite immersive de l’IUT. Ce panorama apporte des informations sur l’IUT et les points de navigations permettent de se déplacer d’une scène à l’autre afin de visiter l’intégralité du département informatique comme si l’on s’y trouvait.

La seconde partie est une application web de génération de panorama. Ce site internet permet à n’importe qui ayant des photos à 360° de créer son propre panorama en arrangeant les scènes les unes par rapport aux autres et en ajoutant des éléments de navigation et d’information. Le panorama résultat est ensuite téléchargeable et peut être visualisé grâce à un serveur web.

Notre but est donc de créer deux outils ergonomiques, un pour la visite de l’IUT et l’autre pour un public plus large, qui offrent des solutions concrètes de visite à distance tout en offrant une expérience immersive et utilisable sur trois plateformes : casque de réalité virtuelle, navigateur sur ordinateur et sur téléphone et casque de RV pour téléphone.

Dans ce rapport, nous allons détailler notre travail au cours de ces cinq mois. Tout d’abord, nous aborderons la gestion de notre projet, puis nous présenteront l’élément technologique clé de ce projet : le framework A-Frame. Nous détaillerons ensuite le panorama de l’IUT, puis le générateur de panorama avant de conclure.

LEXIQUE :

RV : Réalité virtuelle, technologie qui consiste à simuler la présence de l’utilisateur dans un autre lieu.

Framework

Php et html et css et js

back-end / front-end

PRESENTATION DU PROJET

1. L’environnement technique

Le projet ayant déjà été proposé à un autre groupe d’étudiants il y a quelques années, les photos à 360° du département informatique et du bloc central du site de Clermont-Ferrand de l’IUT nous ont été fournies. Ces photos ont été réalisées grâce à un appareil photo à 360° composé de 2 objectifs *fisheye* qui produit des photos rectangulaires qui, une fois mise sous forme d’une sphère, recréent une vue immersive de l’environnement photographié. Ces photos ont été compressées afin de réduire la taille importante des fichiers qui aurait fortement ralenti les transferts.

Pour mettre en scène ces photos en RV, nous avons utilisé le framework A-Frame qui est un framework HTML permettant de créer des environnements virtuels. La visite virtuelle de l’IUT a ainsi été réalisée majoritairement avec le langage HTML avec des parties en JavaScript, notamment pour les animations.

Pour créer le générateur, nous avons choisi le langage PHP pour réaliser le back-end de notre application web et les langages HTML et CSS pour la partie front-end. En revanche, une partie importante du site repose sur du Javascript, notamment l’édition des scènes.

1. Les objectifs

Notre principal objectif commun aux deux parties de notre projet était de créer des visites virtuelles compatibles sur trois plateformes : casques RV, ordinateurs et téléphones. Le générateur n’est utilisable que sur ordinateur mais c’est son résultat qui doit être compatible sur ces trois plateformes.

Les objectifs spécifiques à la visite de l’IUT sont les suivants :

* Visualiser des scènes de l’IUT en 360°
* Se déplacer d’une scène à l’autre grâce aux points de navigation
* S’informer grâce à des panneaux informatifs
* Accéder à la carte représentant une vue d’ensemble de l’IUT
* Naviguer rapidement en cliquant sur des lieux figurants sur la carte

Ceux pour le générateur de panoramas sont :

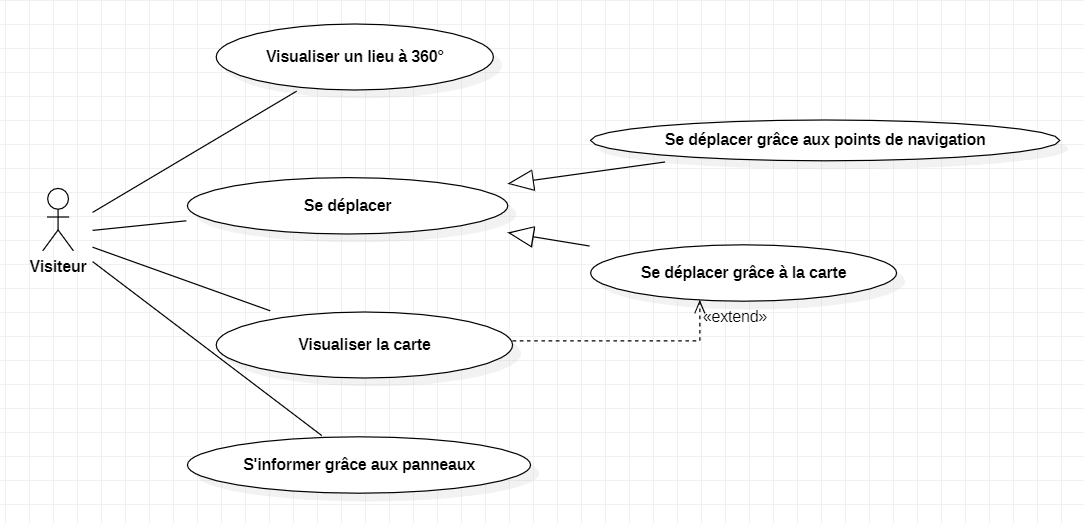
* Uploader des photos 360°
* Editer chaque scène :
* Définir le point de vue, la position de la caméra
* Fixer les points de navigation
* Ajouter des panneaux informatifs
* Créer et éditer la carte
* Sauvegarder le panorama sous forme d’une application web

Tous les objectifs de la visite de l’IUT s’appliquent également au résultat du générateur.

Grâce au panorama de l’IUT, nous avons souhaité créer une visite immersive s’adressant à n’importe qui désirant visiter le département informatique de l’IUT de Clermont-Ferrand. Cela pourrait par exemple être un futur étudiant ne pouvant se rendre sur place. Nous souhaitions pouvoir le proposer comme un outil supplémentaire lors des portes ouvertes qui ont dû se dérouler à distance cette année.

Avec le générateur, nous souhaitions viser un public beaucoup plus large. Nous pourrions imaginer notre site internet utilisé par un particulier souhaitant créer une visite virtuelle d’un bien immobilier qu’il souhaite vendre. La seule condition d’utilisation est de posséder des photos à 360°.

Panorama Use case

Afin de mieux définir les objectifs de la visite virtuelle de l’IUT, nous avons réalisé un diagramme de cas d’utilisation de notre application web.

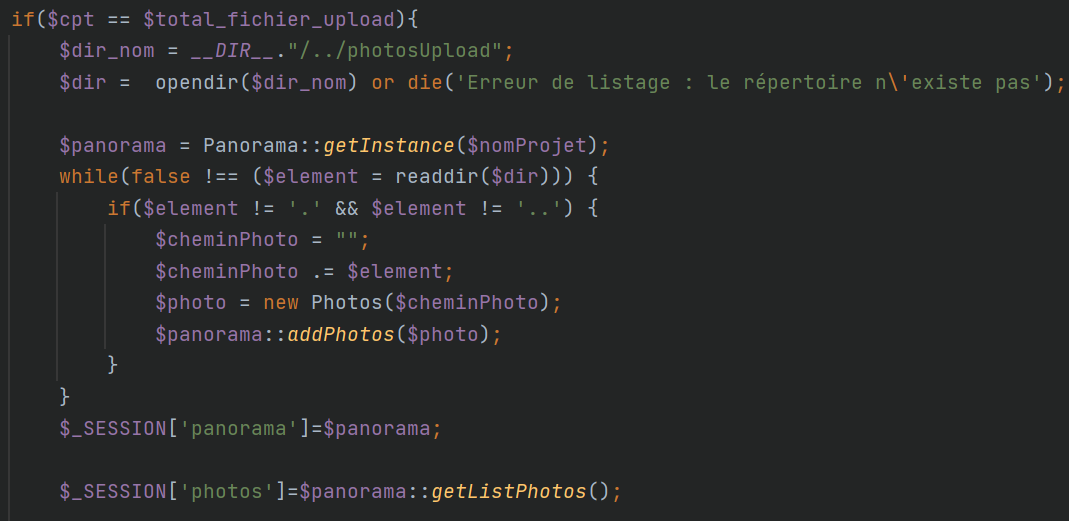
Sur ce diagramme, nous pouvons voir que l’utilisateur peut visualiser un lieu à 360° et s’y déplacer, soit grâce aux points de navigation soit en visualisant la carte qui doit être accessible à partir de chaque scène de la visite.

Le panorama doit également comporter des panneaux informatifs qui indiquent à l’utilisateur où il se trouve et vers où il peut se diriger.

c. EDITION DES SCENES

1. Créer une scène interactive

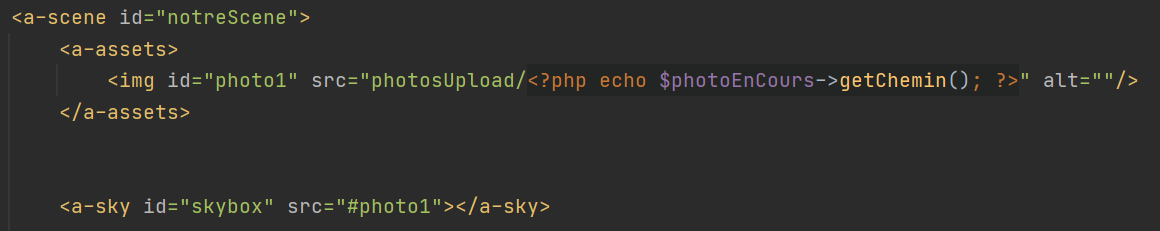
Dans un premier temps, l’utilisateur peut charger ses photos. Grâce à un formulaire, le générateur récupère ces photos et les copie dans un fichier *photosUpload* ainsi que dans une variable de session.



Chaque fichier du dossier *photosUpload* est ainsi transformé en objet Photos avant d’être sauvegardé.

Après avoir validé le formulaire de chargement des photos, l’utilisateur se voit proposer un second formulaire où il choisit quelle sera la première photo traitée. La photo choisie est ensuite déplacée en tête de la liste des photos puis est ouverte grâce à un fichier HTML intégrant des balises A-Frame.

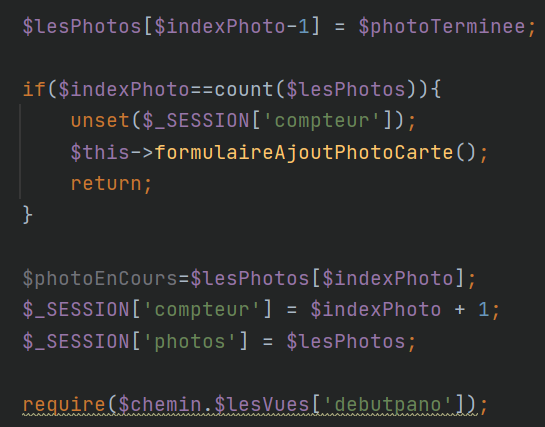
L’utilisateur peut ainsi visualiser sa photo comme dans un panorama.



On envoie le chemin de la photo à la vue, qui l’utilise comme a-sky de la vue, c’est-à-dire comme image de fond de la sphère virtuelle créée par a-frame.

Lorsque l’édition de cette première photo est terminée, il peut passer à la suivante en appuyant sur une touche de son clavier. Cela déclenche l’appel d’une fonction javascript qui va sauvegarder tous les éléments de la scène en cours dans un formulaire qui va ensuite être récupérer en back-end.

Lorsque la sauvegarde est terminée, le générateur affiche la photo suivante en parcourant simplement la liste des photos sauvegardées en session grâce à une variable *index* elle aussi sauvegardée en session.



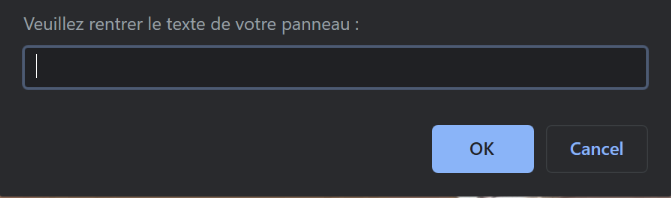
Si la photo qui vient d’être éditée (ici *$photoTerminee*) est la dernière, alors l’utilisateur passe à l’étape suivante, l’ajout de la carte (détaillée dans la partie 6.V.c.3 ci-dessous). Sinon, la même vue est rappelée avec la photo suivante qui peut être éditée à son tour.

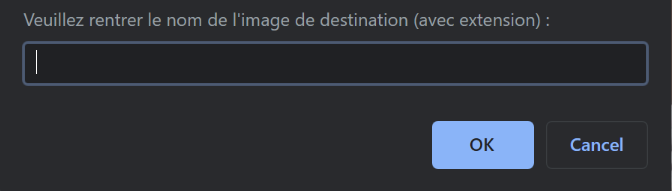
2. Générer des éléments grâce au clavier

Lors de l’édition d’une scène, l’utilisateur peut ajouter sur celle-ci des panneaux informatifs ainsi que des points de navigation. Pour cela, nous avons intégré à la vue d’édition des scènes un fichier javascript qui écoute les événements envoyés par le clavier. Si l’utilisateur appuie sur certaines touches, certaines actions se produisent.



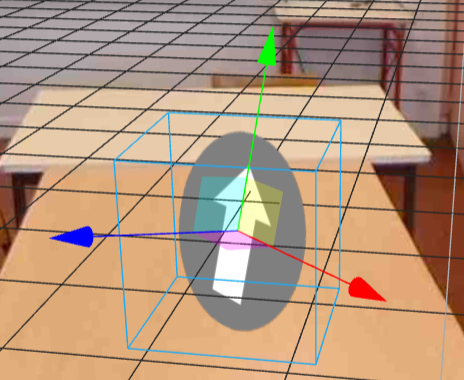
Ainsi, pour ajouter un panneau informatif, l’utilisateur appuie sur la touche **P** et une fenêtre apparait à l’écran en lui demandant d’indiquer le texte qui sera présent sur le panneau. Pour un point de navigation, un appuie sur la touche **N** ouvre une fenêtre permettant d’indiquer le nom de l’image de destination.





Après avoir validé le texte, l’élément apparait à l’écran et l’utilisateur peut alors le déplacer.

Sur les scènes créées avec A-Frame, chaque élément peut être disposé selon trois axes comme le montre la figure suivante :



Afin de simplifier le processus du placement d’un élément, nous avons décidé de ne permettre à l’utilisateur de déplacement l’élément que sur un seul axe à la fois. Avec les flèches de son clavier, l’utilisateur déplace son point ou son panneau selon un axe, puis peut changer d’axe avec la touche **A**. En appuyant plusieurs fois sur cette touche, l’utilisateur peut changer indéfiniment l’axe en cours d’utilisation pour pouvoir perfectionner la position de son élément.

La rotation des éléments n’est pas modifiable puisque chaque panneau ou point de navigation sur le panorama final sera par défaut orienté pour regarder la caméra (c’est-à-dire l’utilisateur).

Une fois que l’utilisateur a terminé d’ajouter et de placer tous les éléments, il peut déclencher la sauvegarde de la scène avec un appui sur la touche **J** de son clavier. Cela va appeler la fonction de sauvegarde du script JS qui lit chaque élément composant la scène et inscrit les éléments importants dans un formulaire. Pour un panneau, le générateur sauvegarde son texte et sa position. Pour un point de navigation, c’est le nom de son image de destination et sa position qui sont sauvegardés.

Pour ce faire, chaque information est ajoutée comme saisie dans un formulaire.

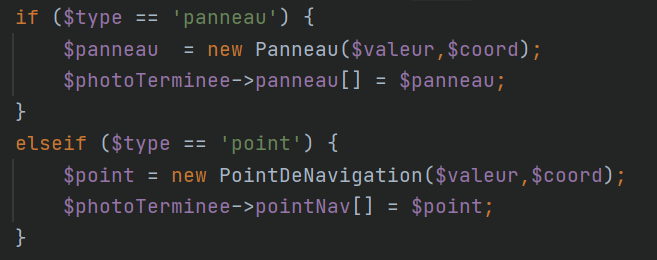


Dans un premier temps, on récupère tous les enfants de la scène étant de classe « panneau » ou « point » puis on les sauvegarde un à un.



Lors de la sauvegarde, on enregistre le texte pour un panneau et le nom de l’image de destination (ici passée en identifiant du point de navigation afin que l’utilisateur ne puisse pas circuler de scène en scène lors de l’édition) ainsi que la position de chaque élément et un nom s’incrémentant pour chaque donnée pour permettre la récupération des éléments par la suite.

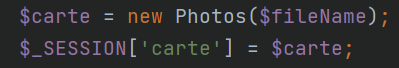
Une fois le formulaire envoyé par le script JS, il est récupéré en *back-end*.



Une boucle permet ensuite de parcourir chaque élément et de créer un objet correspondant au type de l’élément (panneau ou point de navigation) pour les ajouter aux attributs de la photo qui vient d’être éditée. Ainsi, tous les éléments créés lors de l’édition de la scène et envoyés via le formulaire sont sauvegardés en session car contenus dans l’objet photo.

3. Création de la carte

Une fois chaque scène éditée, l’utilisateur est invité à déposée la photo de sa carte dans un nouveau formulaire. Le fichier est à nouveau enregistré dans le dossier *photosUpload* avec les autres photos et un objet Photos est créé dont les attributs permettront de sauvegarder les éléments. Cet objet est également sauvegardé en session mais dans une variable à part afin de pouvoir être traité différemment des autres photos lors de la sauvegarde.



La carte est ensuite affichée comme un plan vertical sur une scène vide afin de ne pas déformer la carte qui n’est pas une photo à 360°. L’utilisateur peut ensuite ajouter des points de navigations comme lors de l’édition des scènes et les placer à sa guise.

Une fois tous les points placés, l’utilisateur peut sauvegarder la carte, dont les points seront sauvegardés de la même manière que pour une scène, et termine ainsi son panorama.

