Clément FERRERE  
Victor MOMMALIER  
Clara PONCET  
Lucile VELUT



**RAPPORT DE PROJET TUTORÉ**





VISITE VIRTUELLE DE L’IUT  
GÉNÉRATEUR DE PANORAMA

DUT Informatique Clermont-Ferrand – 2ème année

AUTORISATION À DIFFUSER SUR L’INTRANET DE L’IUT

Nous autorisons la diffusion de notre rapport sur l’intranet de l’IUT.

REMERCIEMENTS

Nous remercions notre tuteur de projet M. SALVA pour son accompagnement et le prêt de matériel tout au long du projet ainsi que l’équipe enseignante du DUT Informatique.

SOMMAIRE

A- Introduction5

B- Développement4

I. Présentation A-Frame5

INTRODUCTION

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquementII. Gestion du projet  
  
A. Présentation du WBS  
  
Vous pouvez retrouver celui-ci en entier à la fin du rapport en annexes. Notre WBS se scinde en trois parties distinctes : une première sur le panorama original de l’iut, une seconde sur notre générateur et la dernière partie correspond à notre documentions tel que le rapport et la soutenance  
  
  
Une image contenant texte

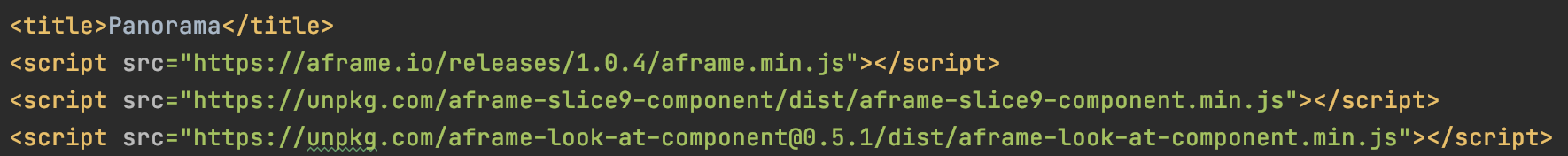
Description générée automatiquementDans la première partie, il y a tout d’abord l’analyse du panorama pour pouvoir commencer au bon endroit et ne pas se lancer dans l’inconnu. On a ensuite programmé la base du panorama, puis la création de la carte et finalement nous avons dû vérifier si notre panorama était totalement fonctionnel sur les trois plateformes demandées : un ordinateur, un smartphone et un casque de réalité virtuelle (type Oculus rift)  
  
  
  
  
Pour le générateur, nous avons aussi commencé par l’analyse pour définir les cas d’utilisations ainsi que les différents packages dans deux diagrammes. Nous avons ensuite codé le générateur avec les différentes étapes que sont le formulaire pour rentrer les photos, l’ajout des points de navigation ainsi que des panneaux informatifs et au final nous avons intégré la possibilité que l’utilisateur rajoute une carte correspondante à son panorama. Pour finir le générateur, il a fallu rajouter une sauvegarde et une génération pour que le résultat soit consultable sur les mêmes plates-formes que le panorama de l’iut.  
  
  
  
Concernant la partie sur le rapport et la soutenance de notre projet, nous avons décidé que dans un premier temps il nous fallait rédiger les documents prévisionnels comme ce WBS ou un Gantt qui a été séparé en deux parties : une première pour la période 2 de notre année et une deuxième pour la période 3. Ensuite nous avons la préparation des soutenances, la deuxième étant la plus longue il a fallu y consacrer plus de temps de préparation. Et finalement ce WBS ainsi que la rédaction de ce même rapport de projet.

Une image contenant table

Description générée automatiquementUne image contenant table

Description générée automatiquement  
B. Gantt prévisionnel  
  
  
Suite au cours Gestion de projet qui se déroulait en parallèle du projet tuteuré, nous avons fait un Gantt prévisionnel afin de nous organiser et de répartir le travail plus efficacement. Comme dit avant, le Gantt a été séparé en deux parties que vous pouvez retrouver en annexe.  
  
  
Nous avons commencé par mettre en place les tâches les plus élémentaires comme faire un exemple d’utilisation du framework A-Frame pour pouvoir le prendre en main et comprendre son fonctionnement. Puis de faire l’analyse du panorama. Ensuite nous avons commencé la programmation des points de navigations ainsi que des panneaux d’informations avec une équipe et une seconde qui s’occupait de la création d’une carte pour faciliter la navigation au sein du panorama. Cette première partie nous l’avons fini en janvier.  
  
  
Après la fin de notre deuxième période nous avons repris notre Gantt afin de le compléter pour la partie sur le générateur. Dans un premier temps il a fallu faire une analyse des besoins pour celui-ci. Nous avons ensuite décidé de créer deux équipes une qui se chargera de la sauvegarde en JSON ainsi que de la génération en .HTML, puis la seconde équipe réalisera le formulaire de soumission des photos ainsi que le JavaScript pour pouvoir rentrer de nouveaux points de navigation et panneaux informatifs. Ces étapes devaient se finir le 14 mars.  
  
  
Nous avons fini par les étapes 3 du WBS, telle que la rédaction du rapport ainsi que la préparation à la soutenance finale de 25 minutes.

C. Gantt réel et analyse des écarts  
  
Nous allons maintenant passer à l’analyse de de notre Gantt réel. Pour la première partie, nous avons fait une analyse poussée grâce à nos cours de gestion de projet. Vous retrouverez tous ces documents en annexe. Ce que nous avons pu conclure de notre première partie de projet c’est que l’on avait fait de bonne prédiction car nous avons suivi le Gantt parfaitement et n’avons pas eu de retard.  
  
À l’inverse, pour la seconde partie, nous savions moins comment nous y prendre. Nous avons tenté de faire un Gantt (cf. ci-dessus) afin de prévoir quelle tâche nous allions faire à quel moment. Mais nous pensions avoir fini le panorama alors qu’il restait des erreurs sur celui-ci. Nous avons donc décidé de nous séparer en deux équipes une qui allait se concentrer sur la création du générateur (Lucile, Clara, Victor) et une autre sur les erreurs à corriger sur le Panorama de l’iut (Clément et Enzo). C’est à ce moment-là que nous avons appris la démission d’Enzo. Clément s’est donc retrouvé seul sur le panorama ce qui nous a ralenti car nous pensions que cette équipe qui travaillait sur le panorama pourrait par la suite nous rejoindre pour nous aider sur le générateur. Nous avons donc pris une semaine de retard sur le Gantt.   
  
Les étapes qui étaient prévues à la fin du projet étaient la sauvegarde en cours du panorama ainsi que la génération du fichier HTML. Nous avons donc choisi de nous concentrer sur la génération car cela nous semblait plus important à finir que la sauvegarde. Comme dit plus haut nous devions finir le projet le 14 mars, or nous avons pris une semaine de retard donc nous n’avons pas encore fini notre projet. Il nous reste la génération du fichier HTML. Nous pensons par ailleurs ne pas avoir assez de temps pour mettre en place la sauvegarde en cours d’édition.   
  
Nous pouvons en conclure que la gestion et l’appréhension du projet se sont mieux passées pour le Panorama de l’Iut que pour le générateur. Cela peut être dû à plusieurs facteurs : en cette troisième période nous n’avons pas eu de cours de gestion de projet nous avons donc peut être délaissé, sans prendre en compte les conséquences, la gestion pour la technique pure afin de finir le projet. De plus, le contexte actuel, nous n’avons jamais eu l’occasion de travailler en présentiel et malgré les moyens mis en place tel que les réunions le lundi matin ainsi qu’une conversation de groupe sur une plate-forme de messagerie il y a eu un manque de communication qui a pu mener à un retard.

III. Présentation A-Frame  
  
A-frame est un framework qui fonctionne avec le langage de programmation JavaScript mais dont l’utilisation se fait principalement en HTML. La première version est rendue disponible en décembre 2015. Il est open-source. Cela veut dire que l’auteur autorise toute personne à l’utiliser gratuitement et à y apporter des modifications.  
Conçu à l'origine au sein de Mozilla et maintenant maintenu par les co-créateurs de A-Frame au sein de Supermedium, A-Frame a été développé pour être un moyen simple mais puissant de développer du contenu VR. A-Frame est devenu l'une des plus grandes communautés de réalité virtuelle.  
  
Basé sur de l’OpenGL il permet de faire facilement des rendus 3D sur des plates-formes Web. Il est plus particulièrement fait pour de la réalité virtuelle que l’on nomme WebVr. Il est plus facile d’utilisation qu’OpenGL. La documentation est accessible sur le site du framework. De plus A-Frame prend en charge la plupart des casques VR tels que Vive, Rift, Windows Mixed Reality, Daydream, GearVR, Cardboard, Oculus Go et peut même être utilisé pour la réalité augmentée.  
  
L’utilisation de ce framework est très simple. En effet, il suffit de partir d’une page HTML vierge et de la compléter comme une page normale. Pour pouvoir ajouter les fonctionnalités du framework. il suffit de l’insérer comme un script JavaScript :  
  
  
  
  
 Nous l’avons utilisé pour développer un panorama et un générateur mais ce framework permet de créer différentes solutions tel que des jeux-vidéos sur serveur web. Ainsi que des applications web.

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquementV. Générateur  
  
D. Sauvegarde et génération html  
1. Modèle persistant  
  
  
Ce que nous avons décidé de rendre persistant est la liste des images dans une classe Panorama qui applique le pattern *Singleton* puisque que l’on veut travailler sur un seul et même panorama à la fois. Puis la liste de photos est une liste d’objet. Chacun d’entre eux ont pour attribut une liste de panneaux et une liste de points de navigation. Les photos disposent de leur chemin en attributs pour pouvoir les recharger.  
  
  
Nous avons seulement besoin de ces trois éléments pour pouvoir générer un fichier en .HTML. Un panorama est en soit qu’un assemblage de photo entre elles. Il faut donc savoir quelle photo mène à quelle autre photo ce qui correspond aux points de navigation. De plus, on peut rajouter des informations sur un panneau pour pouvoir décrire la photo sur laquelle on se trouve. C’est ce qui nous a décidé dans l’analyse à ne rendre sérializable que ces trois éléments.  
  
  
C’est au moment de la création de la session utilisateur qu’est construit le nouveau panorama, il est initialisé avec un nom prédéfini qui est changé au moment de l’ajout des photos par l’utilisateur. Après ce formulaire et l'ajout des photos on fait une simple lecture du dossier d’upload pour pouvoir compter le nombre de photos et les rajouter dans la liste de l’objet Panorama. Par la suite, une boucle est effectuée pour parcourir chacune des photos, et à chaque fois que l’utilisateur a fini avec une photo on fait une sauvegarde depuis le JavaScript au PHP avec un formulaire et ainsi on peut stocker les différents éléments de l’image rajoutée par l’utilisateur tel que les points de navigation ou bien les panneaux d’information.

2. Sauvegarde au milieu de la création  
  
Comme précisé dans notre WBS nous avions prévu de pouvoir faire une sauvegarde au milieu de l’édition d’un panorama pour pouvoir permettre à un utilisateur de sauvegarder sa progression et de revenir dessus plus tard.   
  
Pour commencer, il a fallu décider de la méthode à adapter et du type de fichier que nous allions utiliser. Nous avons choisi la sauvegarde en JSON. L’utilisateur pourra donc récupérer un fichier JSON avec le modèle sauvegardé à l’intérieur. En attendant, on laisse les photos en sauvegarde sur le serveur. Pour reprendre l’édition d’un panorama, il suffira de rentrer dans un formulaire le fichier JSON et après un simple parcours on pourra récupérer la liste de photo ainsi que tous les attributs qui leur étaient associées. Ensuite l’édition reprend comme s’il n’y avait pas eu de coupures.   
  
Malheureusement nous n’avons pas trouvé le temps de programmer et de mettre en place cette fonctionnalité dans le générateur. Nous avons préféré nous concentrer sur la partie génération à la fin du panorama comme indiqué dans l’analyse des écarts.  
  
  
3. Génération du HTML développement  
  
Pour la génération du fichier HTML nous avons choisi de la faire à la toute fin après la sauvegarde de la carte. Pour ce faire, nous avons analysé la structure de notre fichier HTML du projet Panorama IUT pour comprendre quels paramétre changeait selon le panorama et lesquels étaient présents dans tous les cas. Nous avons remarqué que la caméra ne possédait pas d’attributs changeant ainsi que l’entête du fichier. Nous avons donc placé ces éléments dans une page HTML et pour les données changeantes nous avons inséré des boucles de parcours en PHP. Dans un premier temps pour indiquer au framework toutes les ressources à pré charger telles que les images.  
  
Une image contenant texte

Description générée automatiquement  
  
Ensuite, nous avons créé des boucles imbriquées pour parcourir chaque photo et ainsi y associé un « <a-entity id="group » qui est l’entité associée à une image. À l’intérieur de cette boucle, on parcourt la liste des points de navigation de la photo ainsi que les panneaux d’information présents. Le fichier est ensuite téléchargeable et les photos sont détruites du serveur. Si l'utilisateur veut utiliser son fichier, il lui faudra mettre en place son serveur avec les images mais il aura le fichier html prêt et utilisable.

ANNEXES  
  
