

Projet Discret

Rapport Pédagogique

Thomas BENOIT,
Zied BEN OTHMANE, Adrien BISUTTI, Lydie RICHAUME

Mars 2016

Contents

1	Introduction	2
1.1	Contexte	2
1.2	Problématique	2
1.3	Travail réalisé	2
1.4	Organisation du rapport	2
2	Travail réalisé	3
2.1	HTML et CSS	3
2.2	Fonctionnalités	3
2.2.1	3D	3
2.2.2	2D	3
2.2.3	Gestion de données	3
2.2.4	Traduction	3
3	Conduite de projet	4
3.1	Rôles	4
3.2	Méthode de développement	4
3.3	Planification	4
3.4	Gestion des risques	4
3.5	Plan qualité logicielle	5
3.5.1	La norme Iso 9126	5
3.6	Coûts	6
4	Conclusion	7
5	Annexes	8

1 Introduction

1.1 Contexte

Dans le cadre de notre formation de deuxième année de master d'informatique nous avons eu à réaliser un projet à destination de clients réels. Ce projet s'est déroulé sur six mois : depuis Octobre jusqu'en Mars. Il permet de mettre en pratique les compétences que nous avons acquises durant nos cours de gestion de projet, ainsi que de nous confronter au développement d'un projet de dimension moyenne.

Nos clients, Éric Andres et Gaëlle Largeveau-Skapin, ont développé un algorithme de génération de surfaces de révolution discrètes et souhaitaient pouvoir mettre à disposition un outil qui permettrait d'étayer leur publication.

Une surface de révolution est un objet mathématique en trois dimensions généré à l'aide de deux courbes en deux dimensions : la courbe de révolution et la méridienne. La courbe de révolution (le plus souvent un cercle) définit la transformation à appliquer à la méridienne afin de générer la surface de révolution.

1.2 Problématique

Le but de ce projet est de construire un site web qui permette de créer une surface de révolution discrète à partir d'une courbe dessinée à main levée ou décrite à partir d'une équation. Ce site doit aussi permettre de modifier la courbe de révolution et d'exporter les surfaces obtenues ainsi que de les examiner coupe par coupe.

1.3 Travail réalisé

Afin de répondre à cette problématique, nous avons réalisé une application web se basant sur les technologies HTML5/CSS, Javascript et WebGL qui respecte le cahier des charges¹ signé par les clients.

Screen de l'application

Le travail réalisé comprend également une documentation technique (architecture de l'application sous forme de diagrammes de classes et documentation du code générée avec JSDoc) qui permet de maintenir et de faire évoluer l'application.

1.4 Organisation du rapport

¹Présent en annexe.

2 Travail réalisé

Dans cette section nous décrirons les fonctionnalités développées en lien avec le cahier des charges. Nous détaillerons les difficultés que nous avons pu rencontré ainsi que les solutions que nous avons mises en place.

2.1 HTML et CSS

2.2 Fonctionnalités

2.2.1 3D

2.2.2 2D

2.2.3 Gestion de données

2.2.4 Traduction

3 Conduite de projet

3.1 Rôles

Notre équipe est composée de quatre personnes :

- Thomas Benoît - Chef de projet
- Zied Ben Othmane - Responsable qualité
- Adrien Bisutti - Responsable des risques
- Lydie Richaume - Responsable des tâches

3.2 Méthode de développement

Nous avons choisi un cycle de développement en spirale. En effet, cela nous permettait de travailler de manière incrémentale et de garder un contact constant avec les clients afin d'être certains de leur satisfaction, tout en gardant une certaine souplesse quant à la durée de chaque cycle.

3.3 Planification

3.4 Gestion des risques

Nous avons identifié une vingtaine de risque, les plus importants, dont les fiches sont fournies en annexe, étant :

- la possibilité d'un nouveau client ;
- la possibilité d'une évolution de l'algorithme fourni par le client ;
- la possibilité que le rendu 3D demande trop de ressources ;
-

Nous avons été prévenus dès le début du projet que celui-ci avait été proposé comme projet transdisciplinaire au sein de l'université. Cela pouvait se traduire par un ajout de clients menant possiblement à une modification du cahier des charges. Cela aurait donc pu impacter fortement notre projet, notamment au niveau des délais : en effet, la conception réalisée en début de projet aurait pu ne plus convenir et nous aurions pu avoir à modifier des fonctionnalités déjà développées.

Afin de pouvoir anticiper ces problèmes, il a été décidé que nous demandions un maximum d'informations concernant les changements potentiels à nos clients et que nous nous tenions régulièrement informé des évolutions du dossier.

Ce risque s'est avéré (date): le projet de l'université a été accepté et nos clients ont assisté à une réunion avec les nouveaux clients. Il en est rapidement ressorti que l'université n'aurait pas de demandes supplémentaires. Le degré de gravité du risque a donc diminué.

Il a été annoncé dès le début du projet que nos clients travaillaient encore sur de nouvelles versions de leur algorithme. Il était donc possible que nous ayons à inclure des fonctionnalités supplémentaires à celles spécifiées au départ du projet. Cela pouvait donc impacter nos délais par l'ajout de tâches résultant.

Afin d'éviter que cela nous fasse prendre un retard trop grand, il a été décidé que nous n'accepterions pas de modifications annoncées moins d'un mois avant la fin du projet, et que nous nous informerions régulièrement auprès des clients de l'avancement de leurs recherches afin d'anticiper toute modification de planning.

Ce risque ne s'est finalement pas avéré.

Notre projet reprenait le squelette d'un projet précédent et un problème connu était la lenteur du rendu 3D lors de l'affichage d'une quantité trop importante de voxels. Cela pouvait donc avoir un impact important sur les performances de l'application. Afin de palier ce problème, nous avons prévu une tâche visant à améliorer l'algorithme de génération.

Cette tâche a effectivement permis d'obtenir un temps de rendu acceptable sur la plupart des machines, mais le temps de rendu peut rester long lorsque l'on choisit les dimensions maximales pour l'espace 3D.

Bien que nous développions une application web, nous n'avons pas eu à gérer les risques liés à l'utilisation d'un serveur. En effet, notre projet était développé uniquement en HTML et Javascript, et il était mentionné dans le cahier des charges que l'hébergement de l'application serait laissé à la charge du client.

3.5 Plan qualité logicielle

3.5.1 La norme ISO 9126

Nous avons mesuré la qualité logicielle en se basant sur la norme ISO 9126 qui est une norme standard internationale visant à évaluer la qualité logicielle. Elle normalise et classe un certain nombre de principes qualité. Elle est composée de six caractéristiques générales qui définissent la qualité globale d'une application : la capacité fonctionnelle, la fiabilité, l'efficacité, la maintenabilité, la facilité d'usage, la portabilité. Chacune de ces caractéristiques est décomposée en sous caractéristiques. Cette norme semble être une bonne approche pour déterminer la qualité de notre logiciel dans son ensemble et fournir une vue globale satisfaisante.

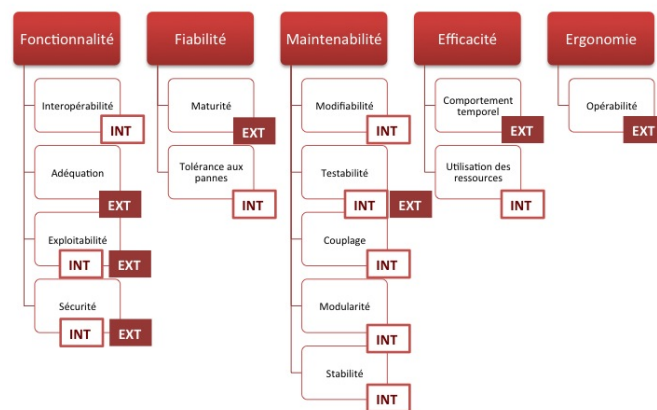


Figure 1: Résumé des attributs qualité ISO 9126

3.5.2 Démarche générale

3.5.3 Les métriques de mesure

3.5.4 Résultats obtenus

3.5.5 Évaluation des résultats

3.6 Coûts

4 Conclusion

5 Annexes