

RAPPORT DE PROJET

Projet de Cproj 2015/2016

Etudiants:

- Charles Rouillard
- Bacquart Julien
- Zeglache Adel



UFR d'Informatique

Enseignants:

- M. Marsault Victor
- M. Zacchioli stefano

PLAN

[Introduction](#)

[Fonctionnalités traitées](#)

[Outils utilisés](#)

[Vue globale du projet](#)

[Organisation du code](#)

[Difficultés rencontrées](#)

[Conclusion](#)

I. Introduction

Dans le cadre de la Licence nous avons été amenés à suivre un cours permettant de voir les différents aspects pour la conduite d'un projet. Le cours nous a présenté des outils que nous avons utilisés pour pouvoir travailler à plusieurs sur un seul et même projet.

Notre projet consiste à réaliser un logiciel qui permet de prendre des fichiers au format osm pour afficher la carte correspondante. Vous trouverez dans ce document une description de l'organisation générale de notre projet, les outils que nous avons utilisés et les fonctionnalités que nous avons codés.

II. Fonctionnalités traitées

Après avoir fini le renderer minimal, nous nous sommes répartis les tâches pour améliorer l'affichage, vous trouverez ci-dessous la liste des améliorations :

- Parsing des nodes, ways, relations et affichage correcte des routes, bâtiments, rivières etc...
- Ratio de la fenêtre adapté au ratio de la carte.
- Affichage de la largeur des routes adaptés a l'échelle de la carte.
- 'Lissage' des angles formés par les routes.
- Gestion correcte de l'ordre d'affichage (ponts au dessus des rivières, etc.).
- Affichage des cours intérieurs des bâtiments grâce a la gestion des relations.
- Affichage correcte de la mer et de la côte.
- Affichage des chemins et pistes cyclables avec des styles propres (lignes pointillées, lignes de tirets).
- Affichage des noms des rues et bâtiments.

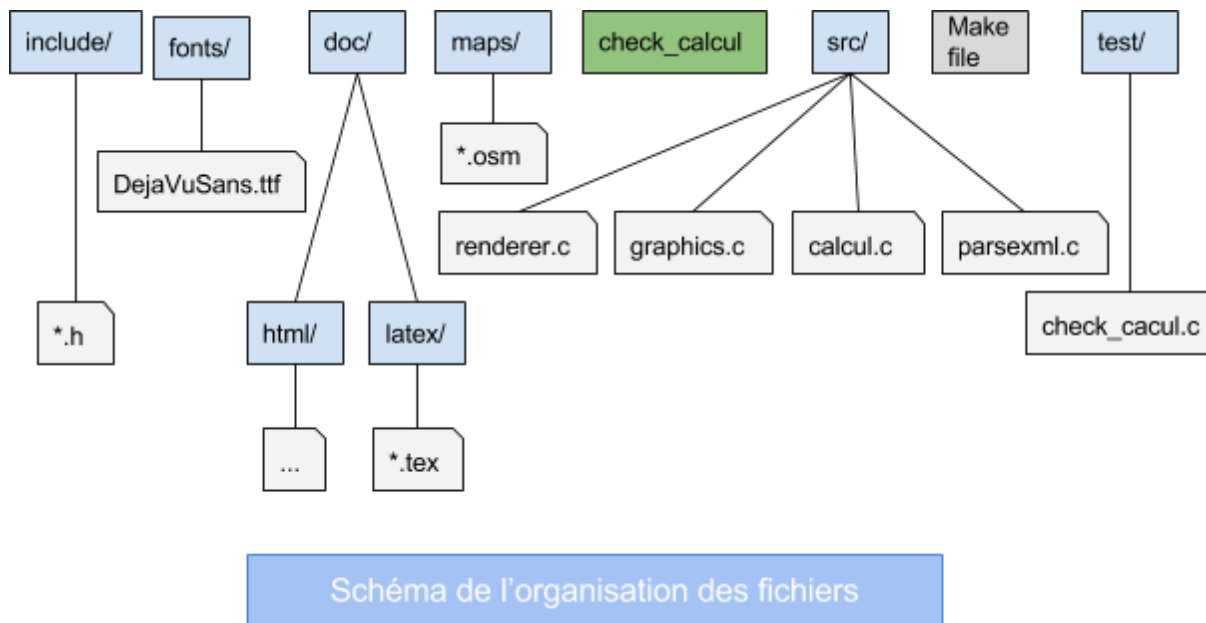
III. Outils utilisés

- [Github](#): Gestionnaire de version
- [LibXML2](#): Parser XML
- [Utaash](#): Implémentation d'une hashtable en C
- [SDL2](#): Bibliothèque graphique

- [SDL2_gfx](#): Primitives de dessin pour SDL2
- [SDL_ttf](#): Primitives de texte pour SDL2

IV. Vue globale du projet

A. Organisation du code



- **src/**:
 - **parsexml.c** : Parse le fichier osm et renvoie une structure Map contenant toutes les données (nodes, ways, relations ...) nécessaires à l'affichage.
 - **calcul.c** : Contient les méthodes de calcul pour la projection de coordonnées GPS sur des coordonnées à l'écran.
 - **graphics.c** : Les méthodes pour dessiner les différents éléments (routes, bâtiments...).
 - **renderer.c** : Contient la fonction main et la boucle principale, fait appel tour à tour aux fonctions des autres fichiers.
- **include/**:
 - **utash.h**: Implémentation d'une hashtable en C.
 - **graphics.h**: Contient toutes les valeurs pour l'affichage (épaisseur, couleur, style, priorité d'affichage...) pour tous les éléments affichés à l'écran. Contient les signatures des méthodes de graphics.c.
 - ***.h** : Déclaration des signatures des méthodes du fichier source correspondant.
- **docs/**: La documentation générée par Doxygen

- **maps/**: Contient les fichiers .osm de test.
- **tests/**: Les tests utilisés par check.
- **fonts/**: Les fonts nécessaire pour l’affichage du texte.
- **renderer**: L'exécutable principal.

B. Difficultés rencontrées

Organisation:

En début de projet, nous avons suivi l’organisation suggérée en TD, à savoir se fixer un ou deux objectifs pour la semaine suivante et les réaliser.

Cette méthode s’est révélée peu efficace dans notre cas:

- Plusieurs personnes travaillent séparément sur une même fonctionnalité, ce qui génère très souvent des conflits au sein du dépôt git
- Il est très dur d’estimer a l’avance la complexité de certaines tâches, et donc nos prévisions se retrouvent soit trop optimistes soit trop pessimistes.

Dans notre cas, il a été plus efficace de passer par la suite a une “TODO List”, on inscrit sur cette liste les fonctionnalités que l’on souhaite voir dans le projet a plus ou moins terme, et des que chacun a du temps disponible, il essaye d’implémenter une de ces fonctionnalité.

Cette méthode a l’avantage, de laisser à chaque membre plus d’autonomie et d’être plus souple, chacun peut travailler sur une fonctionnalité suivant ce qu’il l’intéresse et le temps qu’il a de disponible.

En multipliant les différentes tâches, nous pouvons ainsi réduire le risque de conflits au sein du code.

Elle permet aussi de planifier a plus long terme que juste la semaine suivante.

Librairies:

C’est le point qui nous a posé le plus de problèmes, il est relativement difficile d’estimer si une librairie va couvrir nos besoins, est correctement documentée et ne possède pas de bugs, avant de l’avoir utilisé intensivement.

Nous avons ainsi rencontré de nombreux problèmes avec les librairies choisies.

Bugs:

- *SDL2_gfx* a un sérieux problème de gestion des couleurs sous forme d'*Uint32* ([voir forum](#)). Ce qui nous a obligé à réécrire nos fonctions pour utiliser 4 valeurs séparées *R, G, B, A* dans toutes les signatures.
- La fonction ***gfxPrimitivesSetFontRotation*** de *SDL2_gfx* ne semble pas fonctionner avec des angles autre que 0° .

Absence de documentation:

- La ***TTF_RenderUTF8_Blended_Wrapped*** de *SDL2* n'est documentée nulle part.

Fonctionnalités manquantes:

- De manière générale *SDL* est vraiment trop bas niveau pour nos besoins, *SDL2_gfx* était censé combler ce besoin.
Malheureusement de nombreuses fonctions sont absentes dans *gfx*, par exemple il est impossible de dessiner un cercle rempli avec de l'antialiasing, nous avons donc été obligés d'utiliser des cercles non-remplis sous peine d'avoir des résultats affreux pour de petites échelles.

La bonne surprise a été la librairie *utash*, qui s'est révélée très bien documentée, très facile à utiliser et à intégrer au projet.

IV. Conclusion

Ce projet nous a permis de mieux comprendre certains aspects du cours de Cproj en les mettant en pratique. En plus de nous apporter de nouvelles compétences techniques ce projet nous a également permis de gagner en autonomie (car nous avons utilisés des outils que nous ne maîtrisions pas) et d'apprendre à travailler en groupe.