



FDF

42 Staff pedago@staff.42.fr

Résumé: Ce projet consiste à créer graphiquement la représentation schématique d'un terrain en relief.

Table des matières

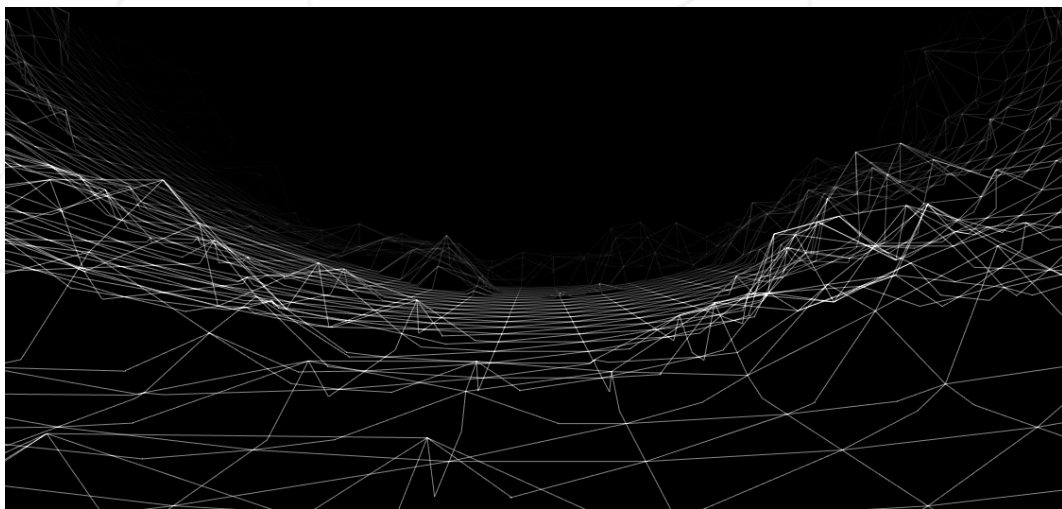
I	Préambule	2
II	Introduction	5
III	Objectifs	6
IV	Consignes générales	7
V	Partie obligatoire	9
VI	Partie bonus	11
VII	Rendu et peer-évaluation	12

Chapitre II

Introduction

La représentation en relief d'un terrain est une pratique clef de la cartographie moderne. Par exemple, en cette ère d'exploration spatiale, avoir une reproduction en trois dimensions de la surface de Mars est un prérequis indispensable à la conquête de cette planète. Autre exemple, comparer des représentations en trois dimensions d'une zone où l'activité tectonique est importante permet de mieux comprendre ces phénomènes et leur évolution, permettant ainsi d'être mieux préparés.

A vous aujourd'hui de vous essayer à cette pratique et de modéliser de magnifiques terrains en trois dimensions, imaginaires ou non...



Chapitre III

Objectifs

Vous découvrirez dans ce projet les bases de la programmation graphique, et en particulier, le placement de points dans l'espace, comment les relier avec des segments et surtout comment observer la scène depuis un certain point de vue.

Vous découvrirez également votre première bibliothèque graphique : La `miniLibX`. Cette bibliothèque développée en interne rassemble le minimum nécessaire pour ouvrir une fenêtre, allumer un pixel et gérer les événements liés à cette fenêtre : le clavier et la souris. Ce sera pour vous l'occasion de vous initier à la programmation dite "événementielle". N'oubliez pas de regarder les vidéos sur l'e-learning !

Chapitre IV

Consignes générales

- Ce projet ne sera corrigé que par des humains. Vous êtes donc libres d'organiser et de nommer vos fichiers comme vous le désirez, en respectant néanmoins les contraintes listées ici.
- L'exécutable doit s'appeler **fdf**.
- Vous devez rendre un **Makefile**.
- Votre **Makefile** devra compiler le projet, et doit contenir les règles habituelles. Il ne doit recompiler le programme qu'en cas de nécessité.
- Si vous êtes malin et que vous utilisez votre bibliothèque **libft** pour votre **fdf**, vous devez en copier les sources et le **Makefile** associé dans un dossier nommé **libft** qui devra être à la racine de votre dépôt de rendu. Votre **Makefile** devra compiler la librairie, en appelant son **Makefile**, puis compiler votre projet.
- Vous ne devez pas utiliser de variables globales.
- Votre projet doit être à la **Norme**.
- Vous devez gérer les erreurs de façon raisonnée. En aucun cas votre programme ne doit quitter de façon inattendue (segmentation fault, bus error, floating point exception, etc...).
- Vous devez rendre, à la racine de votre dépôt de rendu, un fichier **auteur** contenant votre login suivi d'un '\n' :

```
$>cat -e auteur
xlogin$
```

- Vous devez **obligatoirement** utiliser la **miniLibX**. Soit dans sa version présente sur les dumps, soit à partir de ses sources. Si vous choisissez de travailler à partir de ses sources, vous devez appliquer les mêmes règles que pour votre **libft** telles que décrites au dessus.

- Dans le cadre de votre partie obligatoire, vous avez le droit d'utiliser les fonctions suivantes :
 - `open`, `read`, `write`, `close`
 - `malloc`, `free`
 - `perror`, `strerror`
 - `exit`
 - Toutes les fonctions de la lib `math` (`-lm` et `man 3 math`)
 - Toutes les fonctions de la `miniLibX`.
- Vous avez l'autorisation d'utiliser d'autres fonctions dans le cadre de vos bonus, à condition que leur utilisation soit dûment justifiée lors de votre évaluation. Soyez malins.
- Vous pouvez poser vos questions sur le forum, Slack, etc.

Chapitre V

Partie obligatoire

Ce projet consiste à créer graphiquement la representation schématique (en “fils de fer” ou “wireframe” en anglais) d’un terrain en relief en reliant différents points (x , y , z) par des segments. Les coordonnées du terrain seront stockées dans un fichier passé en paramètre, dont voici un exemple :

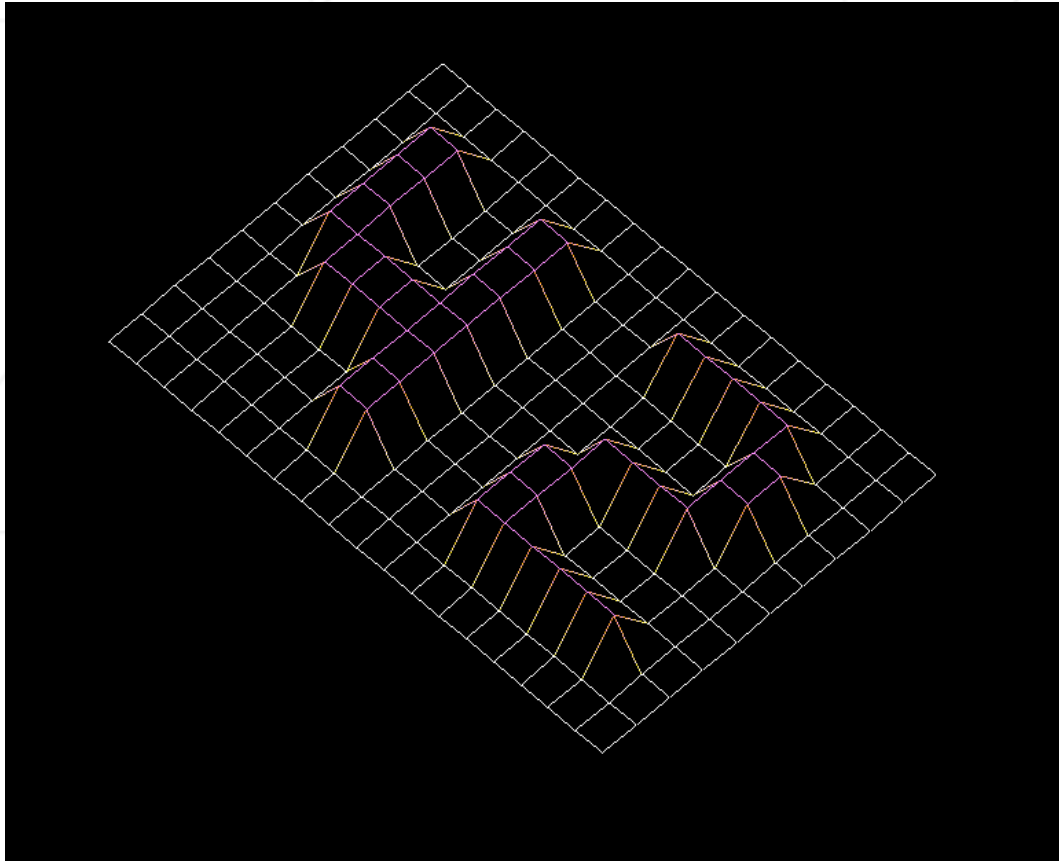
```
$>cat 42.fdf
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 10 10 0 0 10 10 0 0 0 10 10 10 10 0 0 0
0 0 10 10 0 0 10 10 0 0 0 0 0 0 10 10 0 0
0 0 10 10 0 0 10 10 0 0 0 0 0 0 10 10 0 0
0 0 10 10 10 10 10 10 0 0 0 0 10 10 10 10 0 0
0 0 0 10 10 10 10 10 0 0 0 10 10 0 0 0 0 0
0 0 0 0 0 0 10 10 0 0 0 10 10 0 0 0 0 0
0 0 0 0 0 0 10 10 0 0 0 10 10 10 10 10 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$>
```

Chaque nombre correspond à un point dans l’espace :

- La position horizontale correspond à son abscisse.
- La position verticale correspond à son ordonnée.
- La valeur correspond à son altitude.

Si on exécute votre programme `fdf` sur ce fichier, on devra voir quelque chose similaire à :

```
$>./fdf 42.fdf
```



Pensez à exploiter votre `libft` ! L'utilisation de `get_next_line`, `ft_split` et `ft_getnbr` vous permettra de faire une lecture rapide et simple des données du fichier.

En ce qui concerne la représentation graphique :

- Vous avez le choix du type de projection : parallèle, iso, conique.
- Il doit être possible de quitter le programme en appuyant sur la touche `'esc'`.
- L'utilisation des `images` de la `minilibX` est fortement conseillée.
- Vous trouverez dans les fichiers associés au sujet sur l'intranet un binaire de test (`fdf` dans `fdf.zip`) et le fichier d'exemple `42.fdf`



`man mlx`

Chapitre VI

Partie bonus

Voici quelques idées de bonus intéressants à réaliser, voire même utiles. Vous pouvez évidemment ajouter des bonus de votre invention, qui seront évalués à la discrétion de vos correcteurs.

- Remplissage des cases avec une couleur en fonction de l'altitude (vert en bas, puis marron, puis blanc en haut par ex.)
- Pouvoir spécifier en paramètre une palette de couleur.
- Gestion correcte des faces cachées.
- Possibilité de changer de type de projection.

Chapitre VII

Rendu et peer-évaluation

Rendez-votre travail sur votre dépôt GiT comme d'habitude. Seul le travail présent sur votre dépôt sera évalué en soutenance.