

Fractal

Bruno Malenfant

28 juin 2014

- date de remise : 28 juillet 2014 avant 23:55
- peut être fait seul ou en équipe de deux.

1 Description

Pour le dernier tp vous allez faire des modifications sur un logiciel permettant de construire des fractals à l'aide de fonction linéaire et non-linéaire.

2 Mise en place

Sur le site il y a le fichier **tp2.tar.gz**. Vous devez télécharger ce fichier et le désarchiver avec les commandes suivantes :

```
gunzip tp2.tar.gz
tar xvf tp2.tar
```

Ensuite vous pouvez travailler sur le projet. Pour compiler le projet la commande suivante est suffisante :

```
make
```

Pour construire la documentation utilisez la commande suivante :

```
make doc
```

Cette commande va construire un sous répertoire du nom de **documentation** contenant une page html (**index.html**) avec la documentation du projet.

3 Maintenance

Voici les modifications que vous devez apporter au projet :

3.1 Noms

Ajoutez vos noms dans le fichier `main.c` comme personne ayant faites les modifications (n'effacez pas celui de l'auteur).

3.2 Interpolation entre deux descriptions

L'utilisateur va pouvoir donner deux fichiers de description de fonction IFS et demander une transition linéaire entre les deux descripteurs. La commande accepte déjà un fichier de description (f_0), il suffit de permettre la mention d'un second (f_{n-1}). Pour cela, l'utilisateur va utiliser le `flag -f` suivi du nom du second fichier sur la ligne de commande. Le `flag -a` suivit d'une valeur numérique entière (n) va indiquer le nombre d'images que le logiciel devra générer. Cette valeur numérique doit être dans l'intervalle $[2..250]$. C'est une erreur de l'utilisateur d'utiliser `-a` ou `-f` sans utiliser l'autre. Les fichiers de description doivent contenir le même nombre de lignes.

Une description f_i contient k ligne. Chaque ligne représente une fonction à l'aide de 8 valeurs :

- les 6 premières valeurs représentent les coefficients a, b, c, d, e et f .
- La valeur suivante représente un numéro de couleur.
- La dernière valeur est une chaîne de caractères représentant une fonction non linéaire. Les fichiers en entrées doivent avoir les mêmes fonctions non linéaire.

Pour calculer une image intermédiaire, vous devez calculer le système d'équations intermédiaire équivalent. Le calcul de f_i (le système d'équation pour l'image i) est construit en calculant l'interpolation linéaire entre chaque coefficient.

$$v_i = v_0 + i \left(\frac{v_{n-1} - v_0}{n - 1} \right)$$

Exemple :

soit $n = 101$, $f_0 =$:

a	b	c	d	e	f	Couleur	fnl
0.1	0.2	0.3	0.4	0.5	0.6	0	lineaire
1.1	1.2	1.3	1.3	1.2	1.1	100	spherique
-0.1	-0.2	-0.1	0.2	0.1	0.2	200	polaire

et $f_{100} =$:

a	b	c	d	e	f	Couleur	fnl
1.1	1.2	0.3	0.4	1.5	1.6	200	lineaire
0.1	0.2	0.3	0.3	1.2	1.1	0	spherique
0.1	0.2	0.1	0.2	0.1	0.2	100	polaire

alors f_{20} donnera :

a	b	c	d	e	f	Couleur	fnl
0.3	0.4	0.3	0.4	0.7	0.8	40	lineaire
0.9	1.0	1.1	1.1	1.2	1.1	80	spherique
-0.06	-0.12	-0.1	0.2	0.1	0.2	180	polaire

Pour chaque élément, vous générez l'image équivalente et la sauvegardez sous le nom déjà utilisé par le logiciel, auquel vous ajoutez quatre chiffres indiquant la valeur du i . Dans notre exemple, cela nous donnerait `a0020.ppm`.

3.3 Couleur unique

Ajoutez un *flag* `-c` qui permet de choisir une couleur de tracé unique plutôt que d'utiliser la palette. Ce *flag* sera suivi de trois valeurs entières entre 0 et 255. Ces valeurs indiquent les trois composantes : rouge, vert et bleu de la couleur de tracé.

4 Remise

Pour remettre le tp utilisez la commande suivante pour archiver (effacez l'ancienne archive `.tar` avant) :

```
make archive
```

Ensuite remettez le fichier résultant (`tp2.tar.gz`) sur moodle.

5 Évaluation

- Fonctionnalités (10 pts)
 - Ancienne fonctionnalité encore valide.
 - Modifications fonctionnelles.
- Lisibilité (5 pts)
 - Les modifications suivent le style de l'auteur original.
 - Le code reste lisible.
 - Les modifications gardent une bonne modularité.
 - Les commentaires du code reste correct.