

Projet Informatique – Sections Electricité et Microtechnique

Printemps 2018 : *Decontaminators* © R. Boulic

Rendu2

1. Points principaux

Le rendu2 se concentre d'une part sur la détection d'erreur au moment de la lecture d'un fichier de test (mode **Error** et fichiers de test commençant par 'E') et d'autre part sur le dessin de l'état initial de la simulation (mode **Draw** et fichiers de test commençant par 'D').

Le but des modes est décrit dans la section 7.3.3, en particulier les responsabilités respectives de **main** (Fig 8) qui se contente de vérifier que les arguments sont corrects, et du *modèle* (boite en pointillé dans Fig 8 et 9) qui fait le travail de lecture du fichier, construction des structures de données, etc... Bien lire cette section 7.3.3 concernant le comportement du programme lorsqu'une erreur est détectée.

L'architecture à respecter est décrite dans la figure 9b (mode Error) et 9c (mode Draw). Les modules **simulation**, **robot** et **particule** sont des types **opaques**. Les variable globale au projet entier sont interdites ; par contre il est autorisé d'avoir des variables **static** (= globales à un module sans être visibles à l'extérieur du module) pour mémoriser les structures de données. Seul le module **utilitaire** est un type **concret**.

Le rendu2 rapporte **17 points** sur un total de 52 points pour les 3 rendus. La répartition des points est détaillée dans le document **Bareme** ; elle inclut une composante de style et de lisibilité du code source selon les conventions de programmation (et 87 caractères/ligne, 57 lignes/fonction).

2. Erreurs à détecter dans le fichier lu

Nous nous concentrons sur les erreurs que vous risquez de faire le plus souvent. Elles sont décrites dans la Table 1 avec indication de la fonction du module **error** à appeler pour produire un message d'erreur standardisé qu'il ne faut pas modifier. Certaines fonctions demandent un paramètre de numéro de ligne ; il s'agit du numéro de la ligne du fichier qui est en cours d'analyse. Cela correspond au numéro de ligne visible en ouvrant le fichier dans geany.

ATTENTION : lorsqu'une fonction demande un numéro de robot et/ou de particule, la **numérotation** qui apparaît dans l'affichage d'un message d'erreur :

- commence à **1**
- indique l'ordre d'apparition de l'entité dans le fichier lu

Pour votre convenance, certains fichiers tests fournis indiquent les numéros de robot et de particule en commentaire ; ces commentaires ne sont pas obligatoires.

Type d'erreur	Fonction à appeler (détails dans error.c et .h)	Fichiers test (.txt)
Aucune erreur dans fichier	error_no_error_in_this_file	E01, E08
Pas assez de robots	error_fin_liste_robots	E02, E09
Trop de robots (un par ligne)	error_missing_fin_liste_robots	E03
Pas assez de particules	error_fin_liste_particules	E04, E10
Trop de particules (une par ligne)	error_missing_fin_liste_particule	E05
Robot angle not in $[-\pi, \pi]$	error_invalid_robot_angle	E06, E11
Valeur incorrecte d'une particule	error_invalid_particule_value	E07, E12
3 types de collision	Error_collision	E13, E14, E15

Table1: Types d'erreurs qui seront évalués et fonctions à utiliser

3. Mode Draw

Dans ce mode on construit l'interface graphique et on affiche l'état initial correspondant au fichier lu. La seule différence est que, en cas d'erreur, le programme ne s'arrête pas : on doit détruire les structures de données qui ont été construites pendant la lecture du fichier puis attendre que l'utilisateur exploite l'interface graphique. On testera ce mode avec certains fichiers contenant une erreur.

Au stade du rendu2, on demande que le bouton « exit » fonctionne et fasse quitter le programme, et d'avoir au moins des stubs pour les éléments start/step, record et auto/manuel.

Les textbox et boutons pour Open et Save doivent être opérationnels : on doit pouvoir demander de lire n'importe lequel des 3 fichiers fournis (D01, D02 et D03), dans n'importe quel ordre, de sauver l'état initial dans un autre fichier puis de pouvoir le relire.

Le dessin avec OPENGL doit apparaître dans une fenêtre indépendante sans distortion. De même si on change la taille et la proportion de la fenêtre le dessin doit se ré-afficher sans distortion.

4. Fichiers fournis

Vous pourrez utiliser le rendu1 public du module utilitaire dès qu'il sera en libre accès (après la notation du rendu1). Nous fournissons le module **error** (.c et .h) qu'il faudra utiliser pour signaler une erreur au moment de la lecture d'un fichier, les 15 fichiers pour le mode Error (table 1), les 3 fichiers pour le mode Draw et un exécutable demo.x.

3. Forme du rendu2

Pour chaque rendu **UN SEUL membre d'un groupe** (noté **SCIPER1** ci-dessous) doit télécharger un fichier **zip** sur moodle (pas d'email). Le non-respect de cette consigne sera pénalisé. Le nom de ce fichier **zip** a la forme :
SCIPER1_SCIPER2.zip

Compléter le fichier fourni **mysciper.txt** en remplaçant les numéros présents par les numéros SCIPER1 et SCIPER2 des 2 membres du groupe.

Le fichier archive du rendu2 doit contenir (aucun répertoire) :

- Fichier texte édité **mysciper.txt**
- Votre fichier **makefile** produisant un exécutable **projet.x**
- Vos modules source **.c et .h** ; on peut utiliser le source du rendu1 public pour **utilitaire**
- les fichiers source fournis : **error.c, error.h, contantes.h, tolerance.h**.
- les fichiers de test fournis

On doit pouvoir produire l'exécutable **projet.x** à partir du **makefile** après décompression du contenu du fichier **zip**.

Auto-vérification : Après avoir téléchargé le fichier zip de votre rendu sur moodle (upload), récupérez-le (download), décompressez-le et assurez-vous que la commande **make** produit bien l'exécutable et que celui-ci fonctionne correctement.

Exécution sur la VM: votre projet sera évalué sur la VM du second semestre.

Backup : Vous êtes responsable de faire votre copie de sauvegarde du projet. Il y a un backup automatique seulement sur votre compte myNAS. Sur la VM, vous devez activer vous-même le backup (icone « engrenage » en haut à droite, choisir system settings, choisir backup et activer cette fonction en précisant les paramètres). Une alternative est de s'envoyer la dernière version du code source par email.