Formation C++ Ubisoft - Module 2

Romain Arcila[1,2]
Charles de Rousiers[1]

16 mars 2009

[1] INRIA Grenoble
[2] Liris -CNRS Lyon

Plan

Exercice 1

Objectifs

- Supprimer du code tous les bugs existants
- Ajouter le code nécessaire pour obtenir les traces attendues

- Procédez méthodiquement
- Lisez bien le code avant de modifier quoi que ce soit !
- Ĩl y a juste du code à rajouter !
- Fonction Dump() pour voir l'existance de fuites mémoire

Indices

On retrouve principalement :

- Fuites mémoire
- Mauvaise initialization / double libération
- Mauvaise concordance new / delete

Exercice 2

Objectifs

Ajouter les opérateurs allocation nécessaires pour obtenir les traces attendues

- Procéder méthodiquement
- Pensez bien à tous les opérateurs d'allocation que l'on a vu

Plan

Exercice 3

Objectifs

Ecrivez un allocateur de mémoire de type ‘memory pool’

- **Test1** : allocateur brute Allocate / Deallocate
- **Test2** : surcharge des opérateurs new / delete pour une intégration transparente du pool de mémoire

Exercice 4

Objectifs

Ecrivez un pointeur intelligent de type shared pointer

- Pensez aux différents constructeurs
- Surcharger les bons opérateurs
- Pensez bien aux cas de multi-références