

Heritage, polymorphisme Test performances

Mehdi Kitane et Thomas Escure (B3424)

Après avoir décidé d'utiliser les structures évoquées dans les spécifications, nous avons voulu tester les performances de notre applications ainsi que les memory leaks. Voici les résultats de l'analyse d'un fichier contenant 1 million d'éléments:

Le protocole suivi est le suivant : Executer un Load, le résultat attendu est une augmentation de la mémoire, ensuite effectuer un clear, puis re-executer un Load puis un Clear, ensuite 4*UNDO pour revenir au point de départ. Finalement, on effectue une commande pour voir si la libération de la mémoire a bien eu lieu.

Memory

Duration: 2 min 7 sec
High: 226,7 MB
Low: 284 KB



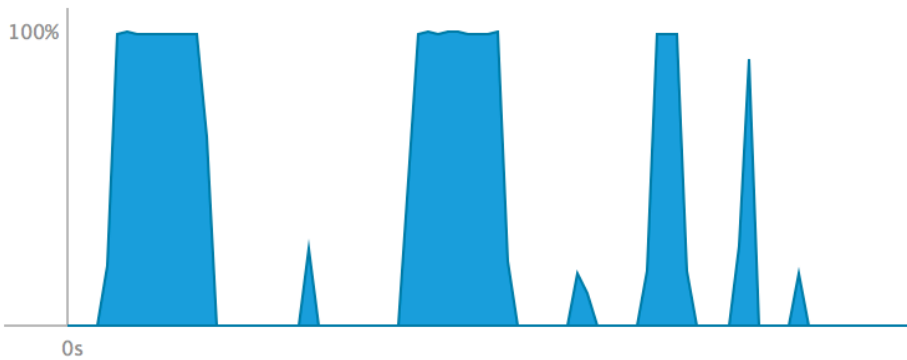
```
LOAD 1millionC.txt
OK
9.08145
CLEAR
OK
0.257636
LOAD 1millionC.txt
OK
9.03938
CLEAR
OK
0.287441
UNDO
OK
3.38989
UNDO
OK
1.20102
UNDO
OK
3.37841
UNDO
OK
1.20112
C c1 1 2 3
OK
# New object : c1
```

Graphe décrivant l'évolution de la mémoire

Commandes effectuées

Utilization over Time

Duration: 2 min 11 sec
High: 100%
Low: 0%



Graphe décrivant l'évolution de l'activité du processeur

De même nous avons effectué un deuxième test afin de vérifier si le REDO est plus efficace qu'exécuter la commande initiale.
 Nous avons donc chargé un fichier de 1million d'éléments puis effectué un UNDO puis REDO

Les résultats sont les suivants :

```
LOAD 1millionC.txt
OK
9.1427
UNDO
OK
1.11995
REDO
2.63103
OK
```

Commandes effectuées avec leur temps d'exécution

Finalement, le dernier test permet de tester les performances de différents éléments géométriques. Le protocole pour tester le load et le save est le suivant : Nous exécutons un script qui permet de créer le fichier à charger à n objets de type demandé. Ensuite nous testons la méthode LOAD et SAVE.

NbElements/Commande	LOAD Cercle	LOAD Poly 2points	Load Rectangle
10 000	0,078s/1,8MB	0,10s/2,1MB	0,09s/1,8MB
100 000	0,7s/14,5MB	1,05s/18MB	1s/14,5MB
1 000 000	9,1 s/140MB	10,5s/170MB	9,5s/140MB

On remarque que la mémoire allouée et le temps d'exécution du programme sont directement proportionnelles au nombre d'éléments.

NbElements/Commande	SAVE Cercle	SAVE Poly 2points	SAVE Rectangle
10 000	0,028s	0,03s	0,03s
100 000	0,28s	0,33s	0,3s
1 000 000	2,73s	3s	2.7s

```
LOAD 1millionC.txt
OK
9.14008
SAVE 1millionC.txt
OK
2.73747
|
```

```
LOAD 10000C.txt
OK
0.078213
SAVE 10000C.txt
OK
0.028976
```

Exemples de tests effectués

NB: Ces tests ont été effectués sur un macbook pro retina Intel Core i5 2,4GHz