

Structures de données II - Rapport de Projet.

Dubuc Xavier & Hannecart Aurore

2 mai 2010

Table des matières

1	Introduction	2
2	Algorithme du peintre	2
3	Test des heuristiques	2
4	Reflexion sur le projet	3
5	Utilisation du programme graphique	3
6	Utilisation du programme console	3
7	Remerciements	3

1 Introduction

Notre projet consistait à déterminer ce que voit un oeil d'une scène en deux dimensions sur un segment en une dimension. Pour cela, nous avons dû nous familiariser avec les arbres **BSP** et implémenter leur construction ainsi que l'algorithme du peintre permettant d'interroger un tel arbre et afficher ce qui est vu par un oeil fournit en paramètre. L'énoncé du problème principal est donc : «**Etant donné un arbre BSP représentant une scène dans un plan (ensemble de segments) et un oeil (un point dans ce plan), donner un algorithme qui affiche la représentation de ce que voit l'oeil**». Nous fournissons dès lors 2 programmes :

- le premier, *Test_GUI.java*, est un programme interactif permettant de charger un fichier contenant une scène en 2D, choisir la position du point de vue, choisir une des heuristiques de construction des arbres **BSP** disponibles et finalement afficher la représentation graphique de ce que voit l'oeil en appliquant l'algorithme du peintre.
- le second, *Test_Heuristiques.java*, est un programme *console* qui permet de comparer les différentes heuristiques de construction des arbres **BSP**, mais qui n'affiche pas de représentation graphique de ce que voit l'oeil à l'écran. Ces heuristiques seront comparées en fonction des critères suivants : la **taille** de l'arbre **BSP** généré par l'heuristique, la **hauteur** de cet arbre, le **temps CPU** utilisé pour construire cet arbre ainsi que le **temps CPU** utilisé pour appliquer l'algorithme du peintre sur cet arbre.

2 Algorithme du peintre

Cet algorithme nous permet d'interroger l'arbre **BSP** que l'on a créé afin de répondre à notre problème, il est donc primordial de connaître la complexité de celui-ci dans le pire des cas. Après réflexion nous trouvons que cette complexité est en $O(n)$ avec n qui est la taille de l'arbre **BSP** (en sachant que la taille de cet arbre est en fait le nombre de segments ou fragments de segments contenus dans l'arbre). En effet, afin d'obtenir un ordre ainsi que la projection des segments sur la droite orthogonale à la vue, il faut parcourir chaque segment/fragment de segments que contient l'arbre afin de lui appliquer la projection et pour chaque segment, on applique de simples instructions en $O(1)$ d'où la complexité totale en $O(n)$.

3 Test des heuristiques

Voici un tableau regroupant les temps d'exécution (en secondes) pour l'algorithme de création des arbres appliqué à un fichier de chaque type :

Scène	Heuristique Linéaire	Heuristique Aléatoire	Heuristique de Teller
Octangle	0	0	0
RectanglesHuge	0.12	0.04	343.08
EllipsesMedium	0.06	0.02	11.92
RandomSmall	0.04	0.08	34.87

Voici un tableau regroupant les hauteurs des arbres BSP créés par l'algorithme de création des arbres appliqué à un fichier de chaque type :

Scène	Heuristique Linéaire	Heuristique Aléatoire	Heuristique de Teller
Octangle	11	10	11
RectanglesHuge	46	15	43
EllipsesMedium	720	130	720
RandomSmall	54	28	121

Voici un tableau regroupant les tailles des arbres BSP créés par l'algorithme de création des arbres appliqué à un fichier de chaque type :

Scène	Heuristique Linéaire	Heuristique Aléatoire	Heuristique de Teller
Octangle	14	15	14
RectanglesHuge	16400	16413	16402
EllipsesMedium	720	738	720
RandomSmall	4541	3922	3537

4 Reflexion sur le projet

Nous avons eu le plus de problèmes lors de la projection des segments sur l'horizontale afin d'afficher ce que voit la vue dans notre rectangle dans le programme graphique. Nous avons finalement trouvé un moyen qui fonctionne pas trop mal qui consiste à utiliser la droite horizontale passant par le point d'ordonnée minimale des segments projetés sur la droite perpendiculaire à la vue et de projeter tous ces segments sur la nouvelle droite. Pour projeter ceux-ci, nous calculons la taille (distance euclidienne entre les 2 extrémités) des segments et nous la reportons sur la droite horizontale. Une fois que nous avons cette droite, nous calculons le rapport entre la taille du segment et la taille de la droite totale et nous l'adaptions à la taille du rectangle où l'on veut afficher, tout en gardant les proportions.

Un autre problème récurrent était que nous avions des difficultés à prendre en compte tous les cas possibles, nous plaçant toujours dans un cas trop spécifique et en occultant les autres possibles.

5 Utilisation du programme graphique

L'application se lance et ne peut être utilisée que lorsqu'un fichier de scène valable a été ouvert (via Fichier > Ouvrir ...), ensuite l'utilisation est assez instinctive, mais il y a quelques subtilités que nous détaillons ici :

- vous pouvez tracer une vue sur la scène avec un «*drag & drop*»,
- vous pouvez (dé)zoomer avec la molette de votre souris en maintenant la touche «*Control*»,
- vous pouvez placer la scène en zoom 1x en maintenant la touche «*Control*» et en cliquant avec le bouton gauche de la souris,
- vous pouvez afficher la scène entièrement en maintenant la touche «*Alt*» et en cliquant avec le bouton gauche de la souris,
- vous pouvez entrer les coordonnées de la vue et ensuite cliquer sur le pinceau afin de dessiner celle-ci (la vue doit être absolument dessinée avant de cliquer sur le «*V*» vert pour valider et afficher ce que voit la vue), (Les coordonnées sont et doivent être exprimées dans le repère JAVA)

6 Utilisation du programme console

Il suffit de suivre les instructions, cependant quelque chose d'important, pour lancer les gros fichiers, il est nécessaire d'augmenter la taille de la pile **JAVA** sinon le programme ne pourra pas fonctionner. Il suffit pour cela d'ajouter l'option «-Xss128m» lors de l'exécution du programme.

7 Remerciements

Nous avons tenu à glisser un petit mot à votre rencontre afin de vous remercier pour la compréhension dont vous avez fait preuve en repoussant maintes fois la date de remise du projet. Un énorme merci à vous, car cela nous a énormément aidé.