

Nom 1 : TEKPA

Prenom : Max

TP : 8

Nom 2 : Tran

Prenom : Thien-Loc

TP : 8

Devoir n°2 : Segmentation par QuadTree

1.Modularisation

Nous avons décidé de découper le projet en deux modules.

- **Quadtree.h** : Ce module contient des structures et des fonctions pour manipuler un quadtree. Les structures incluent Point, qui représente un point avec ses coordonnées 'x' et 'y', ListePoint qui est une liste chaînée de Point, et Quadtree qui représente un quadtree avec ses données telles que le nombre d'éléments, la position, la taille et ses quatre fils. Les points sont stockés dans les nœuds de l'arbre, et chaque nœud peut contenir jusqu'à un certain nombre maximal de points. Si ce nombre maximal est dépassé, le nœud est divisé en quatre quartiers et les points sont répartis dans les quartiers appropriés. Ce module fournit également des fonctions pour ajouter des points à l'arbre, afficher tous les points stockés dans l'arbre, réinitialiser les particules, ainsi que pour libérer la mémoire allouée pour l'arbre et les points.
- **Interface.h** : Ce module définit une interface utilisateur pour interagir avec un quadtree. Il fournit des fonctions pour récupérer les paramètres d'initialisation du quadtree à partir de l'utilisateur, attendre les clics de la souris pour ajouter des points au quadtree, afficher les points stockés dans le quadtree sur l'interface, dessiner le quadtree sur l'interface et fermer la fenêtre de l'interface.

2. Choix d'implémentations

Après avoir fait le choix des structures et la modularisation, nous avons commencé par les fonctions allocations des points et du quadtree. Dans la fonction "allouePoint", il a été décidé d'initialiser les points avec la valeur -1. Cette décision a été prise pour garantir que les points sont initialisés même si leur valeur doit être modifiée plus tard lorsque l'utilisateur commence à insérer des points dans l'interface. Ensuite, nous avons implémenté la fonction "alloueQuadtree" qui initialise de manière récursive l'ensemble du Quadtree. Cette fonction nous permet de ne pas avoir à allouer de nouveaux nœuds ou points après l'initialisation. En ce qui concerne la taille minimale qui peut contenir autant de points que possible, nous avons réservé un grand espace pour la liste des particules en multipliant le nombre maximal de points par quadtree par 100, car nous n'avons pas le droit d'allouer ou de faire des reallocs après l'initialisation.

Ensuite, nous avons implémenté les fonctions d'insertion en commençant par "insererPoint". Cette fonction permet de chaîner les points qui se trouvent dans le même quadtree. Ensuite, nous avons créé les fonctions d'insertion dans le quadtree en commençant par "ajouteTree". Cette fonction vérifie si le quadtree est bien initialisé, puis elle appelle la fonction "ajoutFils" qui sert à insérer les points dans le quadtree en utilisant la fonction "insererPoint" pour les chaîner. Si le nombre maximal de points est atteint dans un quadtree, la fonction "ajoutFils" appelle à son tour les fonctions "purge_un" et "reinitialise_particule". La fonction "purge_un" fragmente le nœud père en ses quatre fils en fonction de leur position dans l'espace, tandis que la fonction "reinitialise_particule" libère les points se trouvant dans le nœud père en les mettant à NULL et met le "nb_elm_in_part" (qui est le nombre de points dans le nœud) à -1 pour montrer qu'il a été purgé.

Avant de passer à la partie graphique, nous avons implémenté la fonction "afficheQuadtree" qui permet d'afficher le contenu de chaque nœud du quadtree à la fin de chaque exécution. Ensuite, nous avons commencé la partie graphique avec les fonctions permettant d'afficher les points sur l'interface et également les carrés qui contiennent les points. La fonction "donnees" récupère les paramètres nécessaires pour initialiser le quadtree. Elle permet de modifier la valeur de ces paramètres en passant les pointeurs par référence et renvoie un entier indiquant si la saisie des paramètres a été réussie ou non. La fonction "get_point_on_clic" permet d'attendre le clic de la souris pour récupérer les coordonnées du point cliqué et renvoie les coordonnées du point sous la forme d'un objet Point.

Difficultés rencontrées: Nous avons rencontré quelques difficultés en ce qui concerne l'affichage graphique, car la fenêtre effaçait tous les points avant de les réafficher à chaque itération.

3. Mode d'emploi

Voici les instructions pour exécuter le programme :

1. Ouvrez le terminal et exécutez la commande "make" pour compiler le programme, puis exécutez-le en tapant "./quad".
2. Suivez les instructions fournies par l'interface pour saisir les paramètres d'initialisation, tels que le nombre total de particules et le nombre maximum de particules par feuille du quadtree. Notez que "nb total de particules" correspond au nombre maximal de points qui peuvent être affichés sur l'interface, tandis que "nb max par feuille" correspond au nombre maximal de points que chaque feuille du quadtree peut contenir.