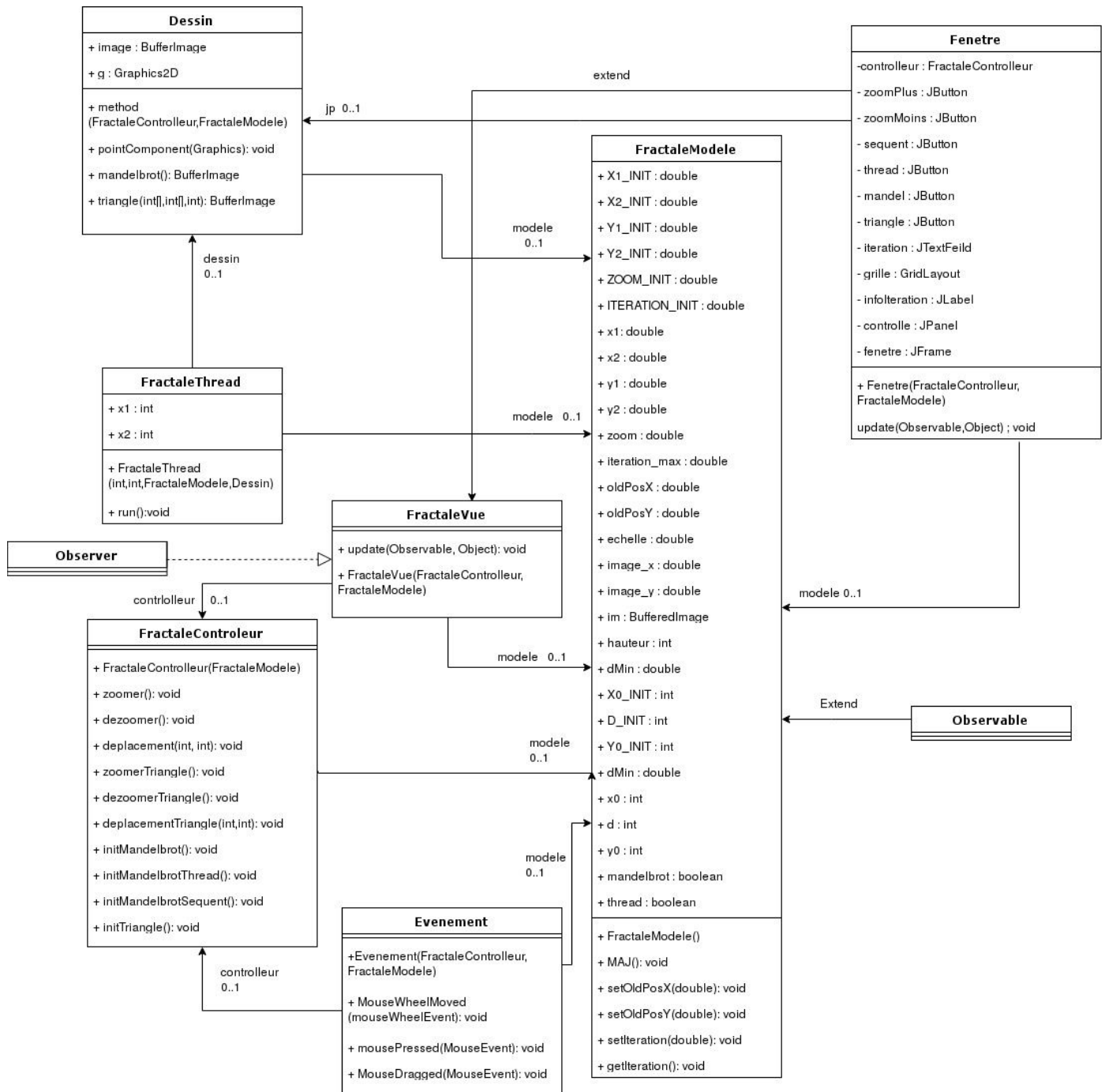


Projet : visualisation de fractales

BETTON Maxime TRIJEAN Julien

I/ Diagramme de Classe



II/ Description des principales méthodes

A/ méthodes de contrôle du fractales

Les méthodes responsables du déplacement de la fractale Mandelbrot sont `zoomer()` et `dezoomer()` du contrôleur qui permettent de zoomer et de dézoomer dans la fractale. De ce fait, on incrémente notre variable `zoom` du modèle, ce qui permet de calculer et d'afficher une plus petite partie de la fractale. La nouvelle valeur de la limite de l'image à droite a donc été modifiée, on modifie donc la limite de l'image à gauche en fonction de la limite de l'image à droite. On incrémente également la valeur `iteration_max` afin que notre fonction de calcul calcule plus de pixels.

Ensuite, pour le déplacement en abscisse et en ordonnée, on utilise une fonction de déplacement qui prend en mémoire une ancienne position de la souris, qui est soustraite à la position de la souris. Et cette différence est ajoutée à la valeur de la position de maximum de l'image à gauche.

De la même manière la fractale Sierpinski est gérée par les méthodes `zoomTriangle()`, `dezoomTriangle()` et `deplacementTriangle()`

B/ méthodes de calcul du fractales

La classe `Dessin` est responsable de toutes les méthodes qui permettent de retourner une `buffered image`. On a `paintComponent()` responsable de définir la taille ainsi que les paramètres de l'image. Cette méthode appelle les méthodes principales (`mandelbrot` ou `triangle` en fonction du choix de l'utilisateur) pour construire la `BufferedImage`. Dans le cas de `mandelbrot`, la classe `FractaleThread` est instanciée pour lancer une double boucle permettant l'affichage de tous les points dans un repère `x,y`.

III/ Liste des fonctionnalités

Le projet java du fractale que nous avons implémenté comprend l'affichage de deux fractales, à savoir Mandelbrot et Sierpinski dans une fenêtre à dimension modulable.

Le zoom et dézoom de la fractale se fait au centre de l'image (illustration par un pointeur) et il est activé via les boutons "+" et "-" ou l'utilisation de la molette. On dispose de deux boutons permettant de sélectionner le fractal à afficher (lors de l'appui du bouton on réinitialise le fractal). On a aussi disposé un `textBox` pour modifier manuellement le nombre d'itération. L'utilisateur a aussi la possibilité de mouvoir le fractal pour effectuer un zoom sur le point de son choix. Enfin la dernière fonctionnalité est la possibilité d'utiliser un calcul séquentiel ou un calcul avec des threads (en parallèle).

La méthode `triangle` suit le même fonctionnement avec comme particularité d'être récursif (plutôt qu'une double boucle).

IV/ Liste des problèmes rencontrés

Un des premier problèmes a été l'appropriation de l'algorithme de mandelbrot en java. Le deuxième gros problème qui n'a jamais été résolu et qui nous as amené à un choix de conception différent est le zoom sur le pointeur de la souris avec le molette. Au vu de la difficulté rencontré et du temps perdu nous avons préféré zoomé sur le milieu du dessin . Le troisième était bête car il s'agissait d'une erreure d'inattention mais après le déplacement du fractal, quand on relâche la souris, le dessin se repositionné à sa position de base.