Rapport projet POO2

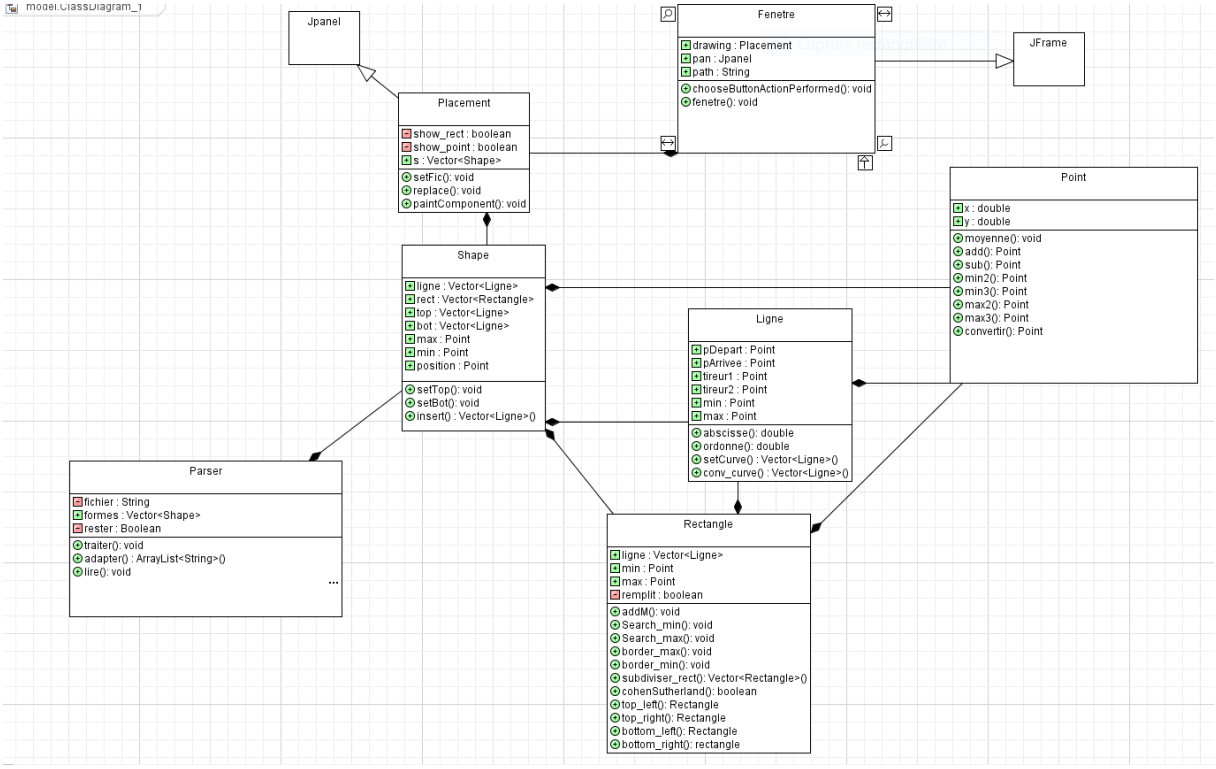
Nicolas Hoerter

Adrien Fleith

Utilisation

L'interface graphique est constituée de différents bouttons. Nous avions beaucoup d'idées en tête pour la réalisation de ce projet. Malheureusement, le temps nous a manqué pour concrétiser certaines d'entres-elles. C'est pourquoi notre interface contient un bouton développement en plus du bouton pour la version stable. Il est possible de charger un fichier ou d'en charger successivement plusieurs (pour améliorer le rendement la machine ne devrait pas être arrêté tant que le rouleau n'est pas remplit). Les autres boutons servent à détailler le fonctionnement interne.

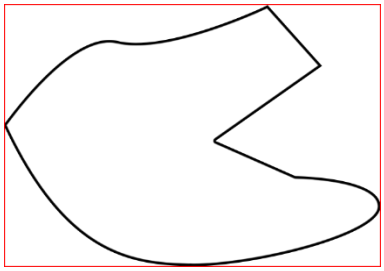
UML

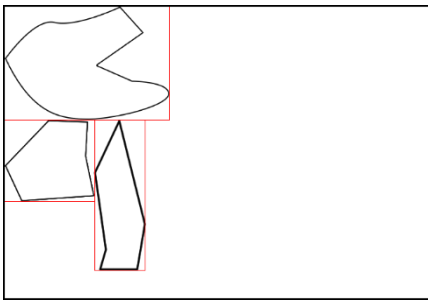


Algorithmes

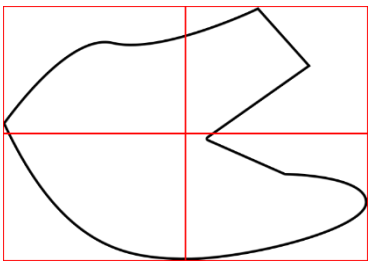
Le parseur utilise un BufferedReader et, au long de sa lecture, ajoute dans un Vector<Shape> (Shape représentant une forme) les points des figures. Ceux-ci sont ajoutés par « Lignes », la classe Ligne étant constituée de quatres points : le point précédent, le point suivant, le premier point tirant la courbe et le second. A la rencontre d'une balise <path>, les positions absolues des points sont stockées par rapport au premier déclaré en (0,0) dans un ArrayList<String>, choisi pour sa méthode addAll(). Cette méthode est utile dans le cas des groupes ou toutes les formes sont ajoutées avant d'être traité à la sortie du groupe.

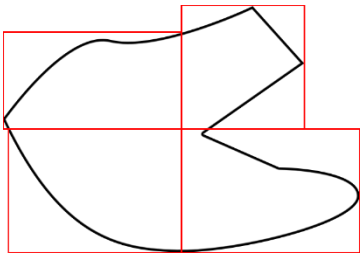
Afin d'agencer les différentes formes entre elles nous avons réalisé 2 algorithmes, un premier très simple où nous plaçons toutes ces formes dans des rectangles englobant totalement chaque forme. A la suite de cette opération nous les plaçons le plus à gauche possible (en naviguant de haut en bas), afin de ne pas superposer les formes nous vérifions qu'aucun rectangle qui entoure une forme ne rentre en collision avec un autre.





Dans le cadre du deuxième algorithme de placement, nous effectuons des subdivisions successives des rectangles contenant les formes. Cette découpe se fait en 2 étapes d'une part nous découpons chaque rectangle en 4 parts égales et ensuite nous réduisons chaque rectangle pour s'approcher au plus près de la courbe, pour effectuer cette deuxième étape j'ai utilisé l'algorithme de cohen-sutherlande afin de déterminer les positions où les lignes coupent chaque rectangle.





Enfin à la suite de cette subdivision par manque de temps nous avons réutiliser l'algorithme de placement utilisé par notre première technique de placement au détriment d'une autre méthode qui serait plus optimisée. Cette dite technique ajoute une étape supplémentaire après la subdivision des rectangles : déterminer les bords supérieur et inférieur (respectivement gauche et droite) de toutes le formes et effectuer un lancer de rayons à partir de chaque changement de hauteur afin de déterminer la distance maximale sans avoir de collision entre les bords.