Groupes de fonctions :

|  |  |  |
| --- | --- | --- |
| **Nom groupe** | **Fonctions** | **Fichier d’implémentation** |
| Tri | sort  compare | mainFunctions.c  mainFunctions.c |
| Réflexion | isReflectedRay  findNormalisedVector | mainFunctions\_PEUL.c  mainFunctions\_PEUL.c |
| Collision | doesCollide\_PEUL  intersectLight\_PEUL  doesRayCollideWithAnyEllipse | mainFunctions\_PEUL.c  mainFunctions\_PEUL.c  mainFunctions\_PEUL.c |
| Ombre | isInTheShadow  calcParamEquaBetweenTwoPos  listingTimes\_PEUL  **Groupe tri**  **Groupe collision** | mainFunctions\_PEUL.c  mainFunctions\_PEUL.c  mainFunctions\_PEUL.c |

Appel des fonctions :

|  |  |
| --- | --- |
| **Fonctions** | **Fichier d’implémentation** |
| main  loadFromFile  nbLine  planEqua  vect  produitVect3d  createImage  newBMP  equaParamLight  listingTimes  **Groupe tri**  doesCollide  intersectLight  doesCollideEllipse  **Groupe réflexion**  mirrorFace (récursif)  listingTimesWithParamEqua  **Groupe tri**  **Groupe collision**  **Groupe réflexion**  **Groupe ombre**  setcolor  **Groupe ombre**  pixelAvg  saveBMP  getcolor  deleteBMP  freeAll | main.c  structure.c  structure.c  structure.c  structure.c  structure.c  mainFunctions.c  bmp.c  mainFunctions.c  mainFunctions.c  mainFunctions.c  mainFunctions.c  mainFunctions\_PEUL.c  mainFunctions.c  mainFunctions\_PEUL.c  bmp.c  mainFunctions.c  bmp.c  bmp.c  bmp.c  structure.c |

Fonctionnement global du module :

1. Lecture du fichier data.txt, récupération des champs de données et stockage dans les structures correspondantes.
2. Suppression du fichier data.txt **(?)**
3. Création du squelette de l'image (tableau de hauteur\*largeur pixels)
4. Boucle pour chaque pixel :
   1. détermination de l'équation paramétrique du faisceau lumineux qui va de l'observateur au pixel du plan image
   2. détermination de toutes les intersections possibles avec les plans(fonctionnel)/sphères (en cours de développement) **(?)**
   3. test de collision avec le polygone de chaque plan en partant du temps le plus petit
   4. si collision, attribution de la valeur rgb de la face au pixel
   5. détection des obstacles entre le point incident et la source lumineuse
   6. si oui : facteur de luminosité réduit pour simuler une ombre
   7. attribution de la valeur rgb du background
5. Création du fichier .bmp
6. Post-traitement : anticrénelage