Conception

Dynamic Sampling and Rendering of Algebraic Point Set Surfaces

William Caisson Xavier Chalut Christophe Claustre Thibault Lejemble

Client : Nicolas Mellado

CONTEXTE

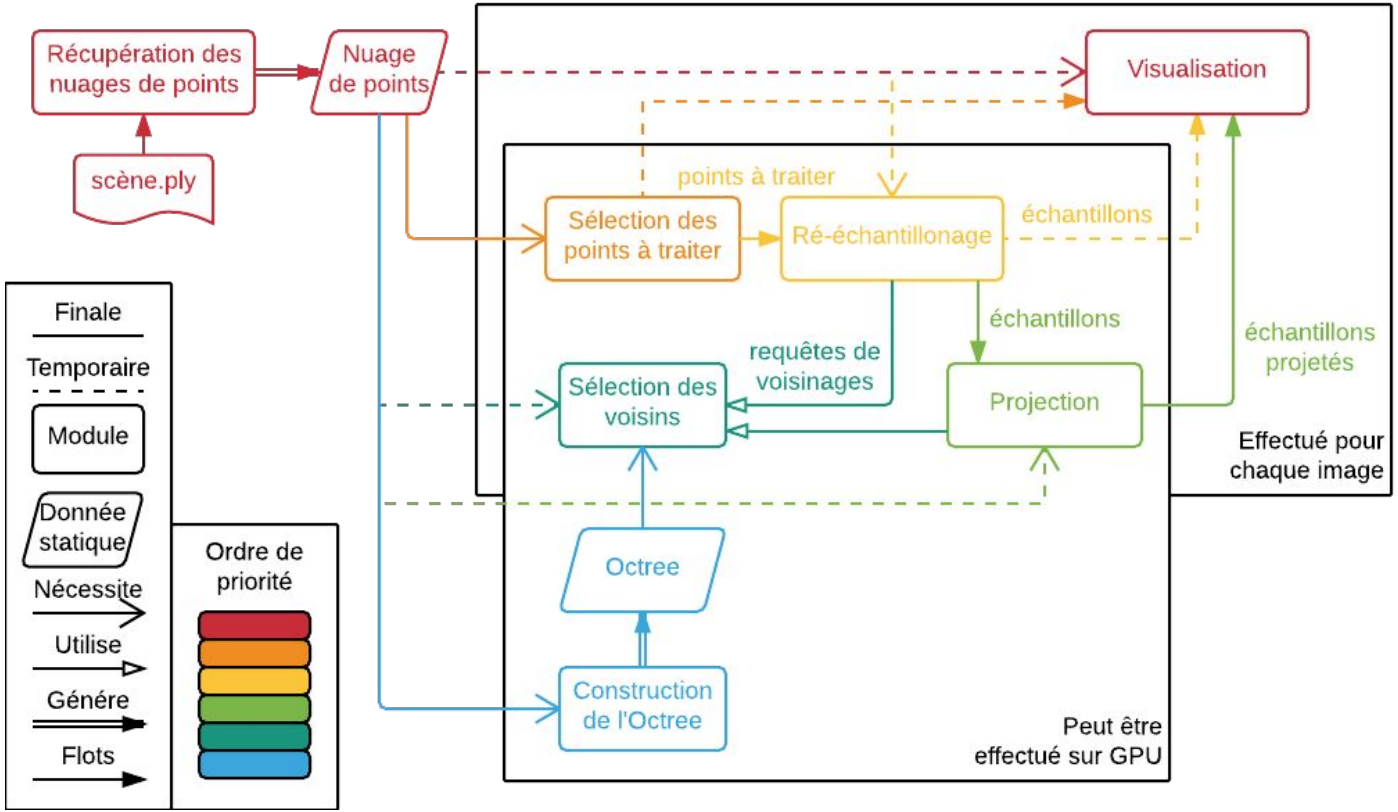
Objectif

Visualiser en temps réel une surface lisse qui approche le nuage de points

Dynamic Sampling and Rendering of Algebraic Point Set Surfaces

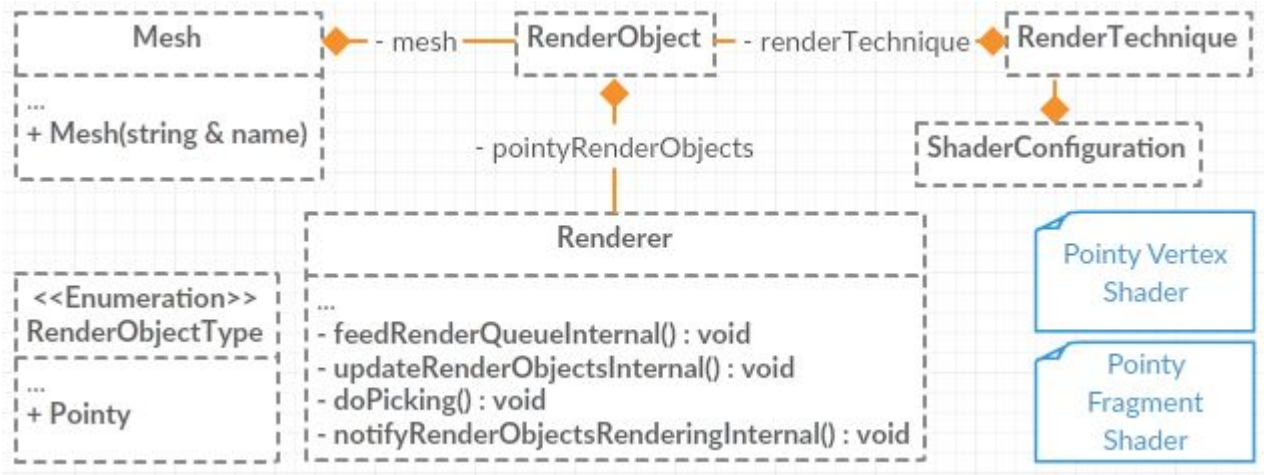
Gaël Guennebaud Marcel Germann Markus Gross

Rappel du système



SOMMAIRE

L’architecture logicielle - Représentation du nuage

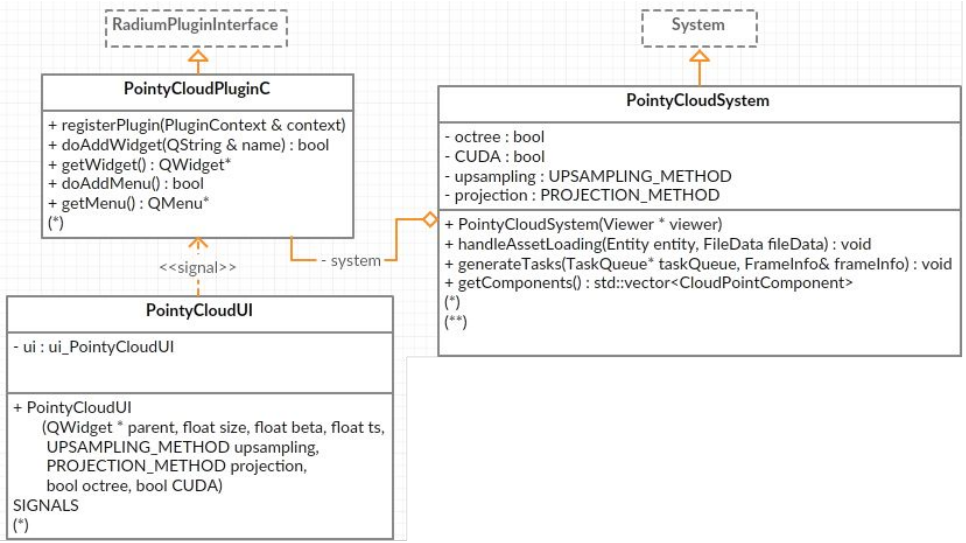


Mesh : conteneur de géométries (points, normales et couleurs) **+** appels OpenGL

RenderObject : wrapper contenant un **Mesh** et une **RenderTechnique**

Renderer : dessine toutes les instances de **RenderObject** existantes

L’architecture logicielle - Plugin Radium

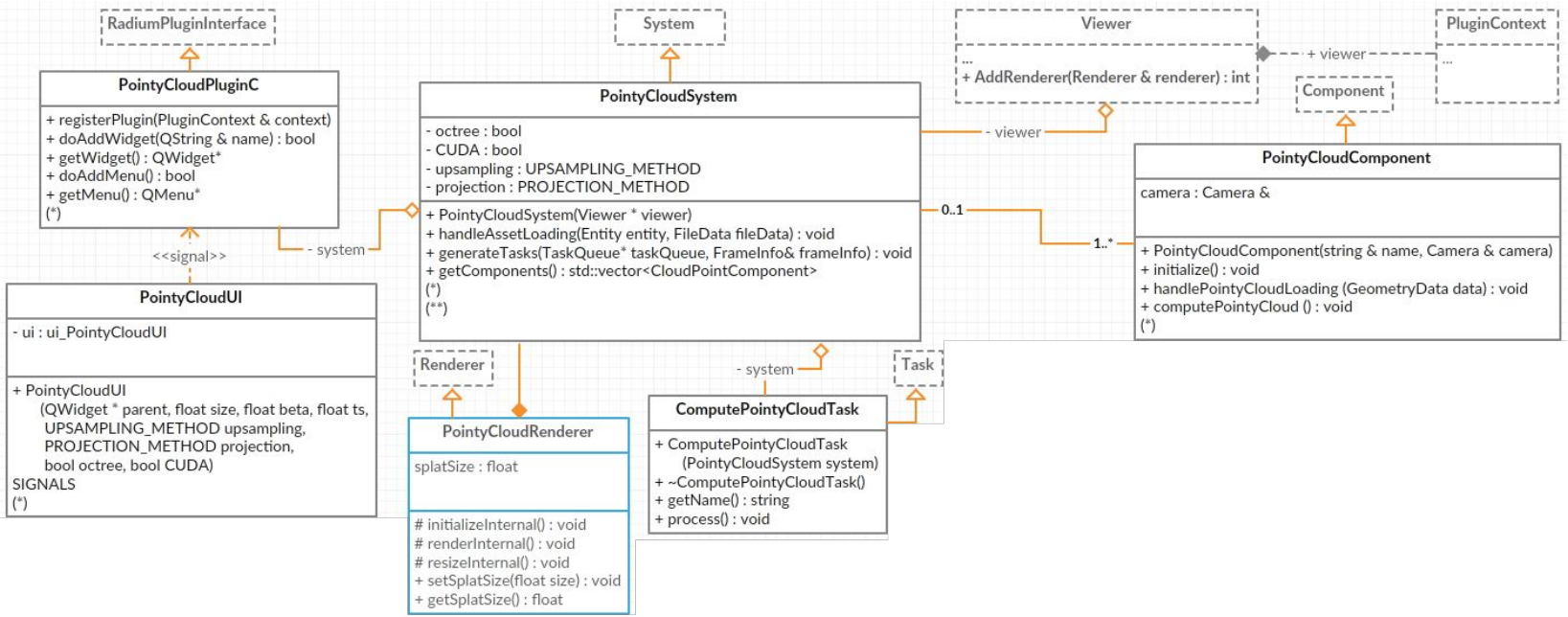


PointyCloudPluginC : interface du plugin avec Radium

PointyCloudUI : interface graphique pour l’utilisateur

PointyCloudSystem : gestion des données du plugin

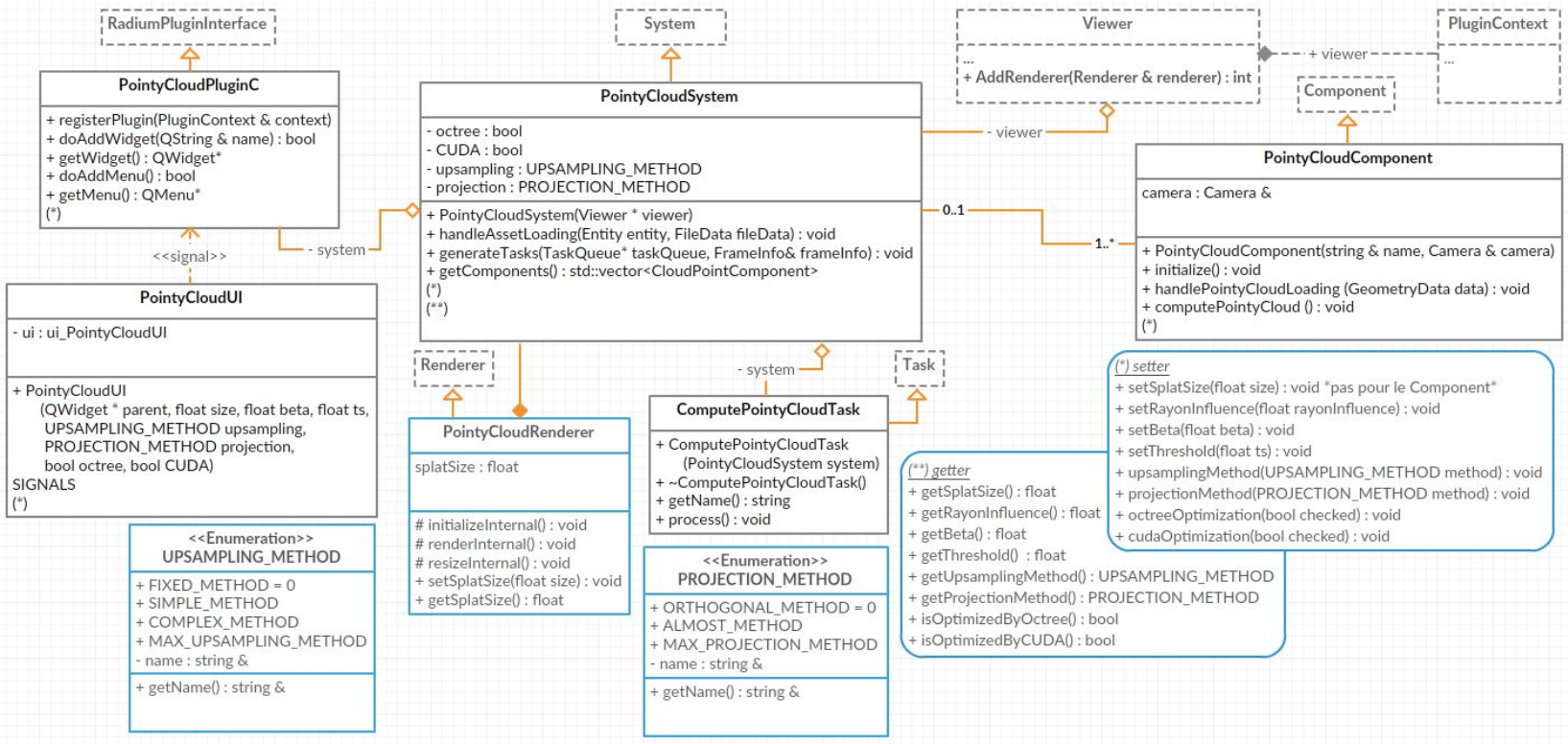
L’architecture logicielle - Plugin Radium



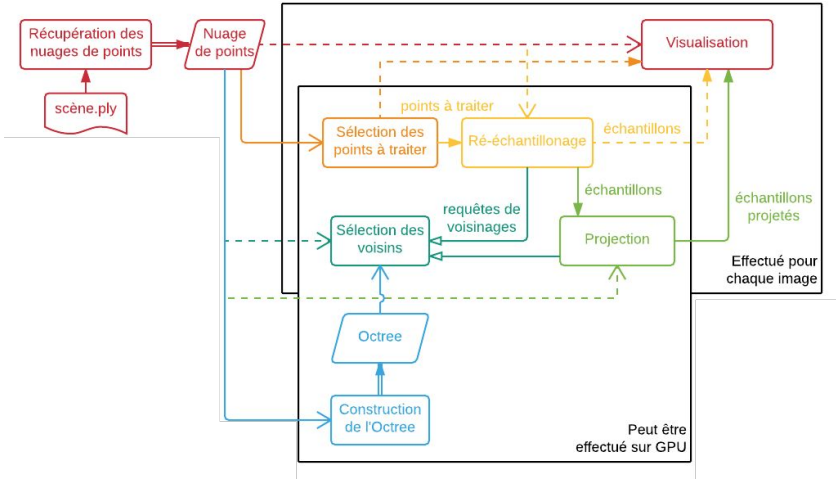
PointyCloudComponent : élément regroupant des **RenderObject** ➜ nuages de points

ComputePointyCloudTask : tâche exécutée à chaque image ➜ calcul de l’**APSS**

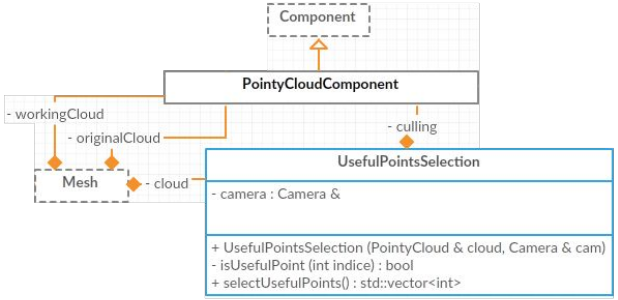
L’architecture logicielle - Plugin Radium



L’architecture logicielle - APSS



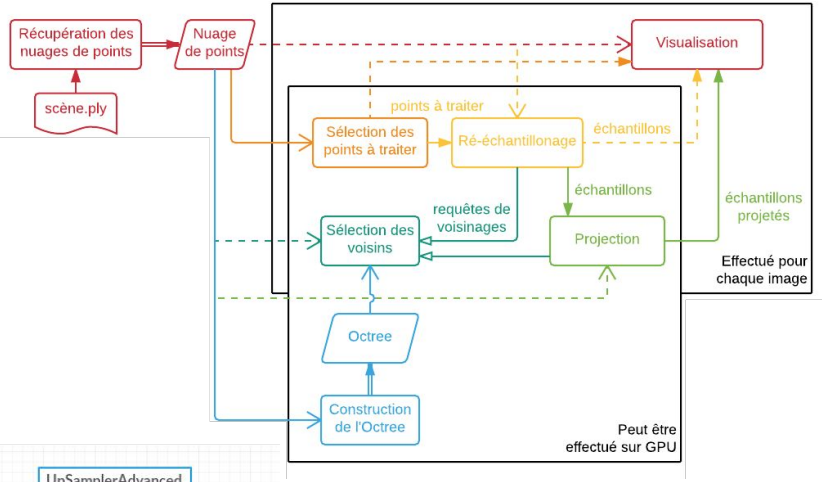
UsefulPointsSelection : sélection des points à traiter

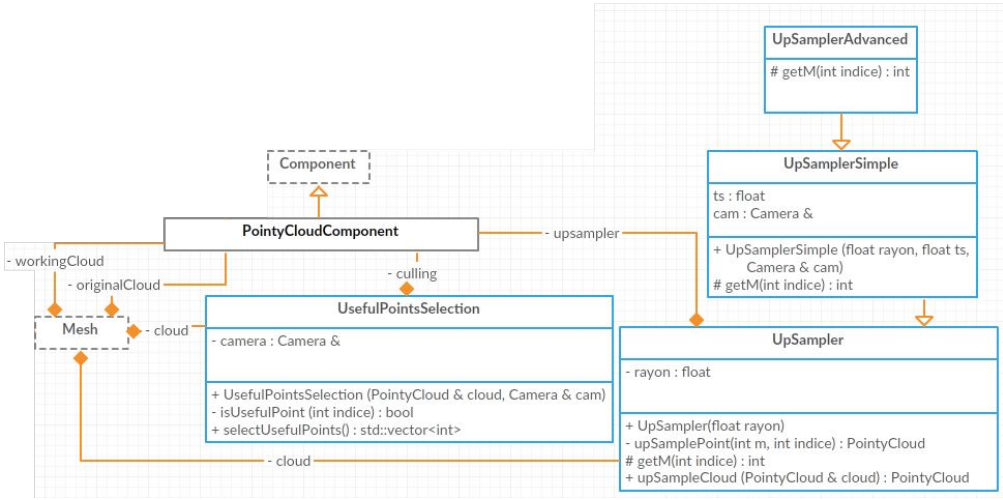


L’architecture logicielle - APSS

UsefulPointsSelection : sélection des points à traiter

UpSampler : ré-échantillonnage



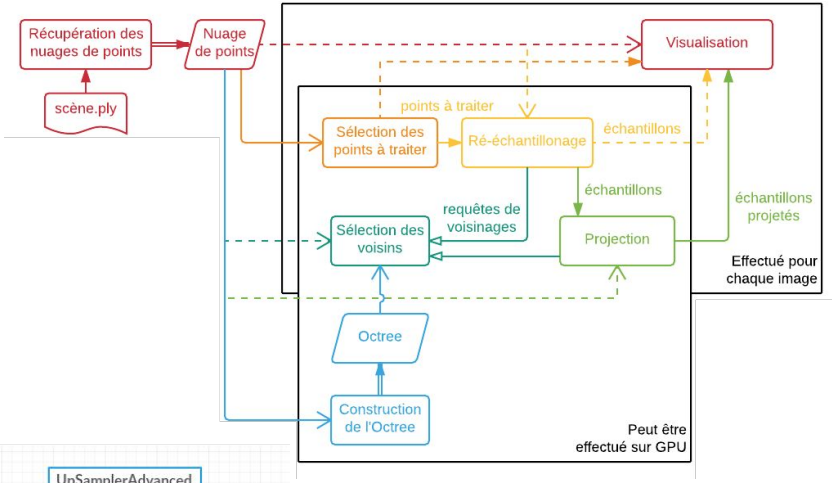


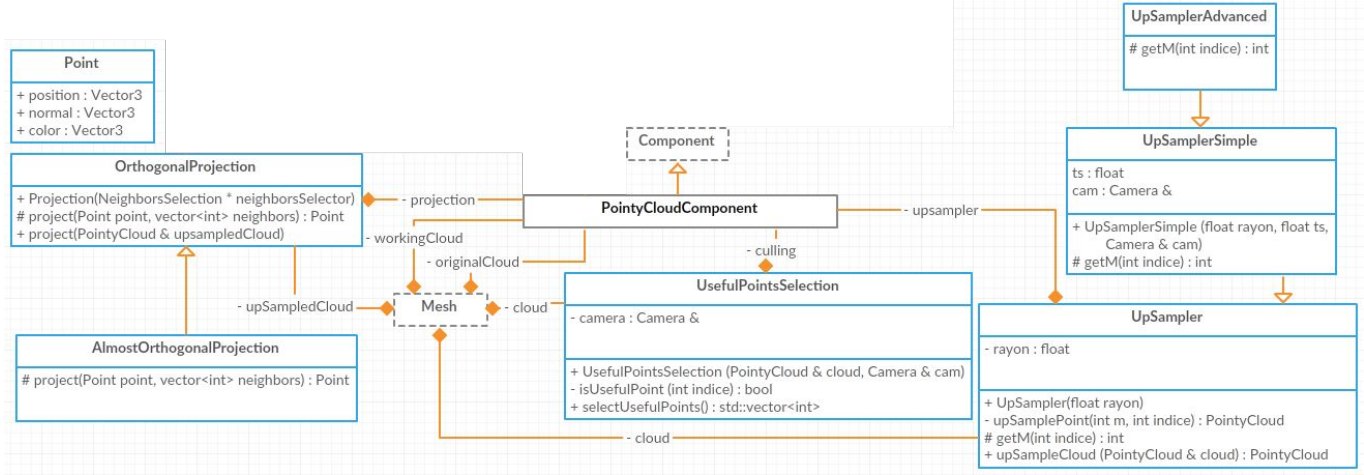
L’architecture logicielle - APSS

UsefulPointsSelection : sélection des points à traiter

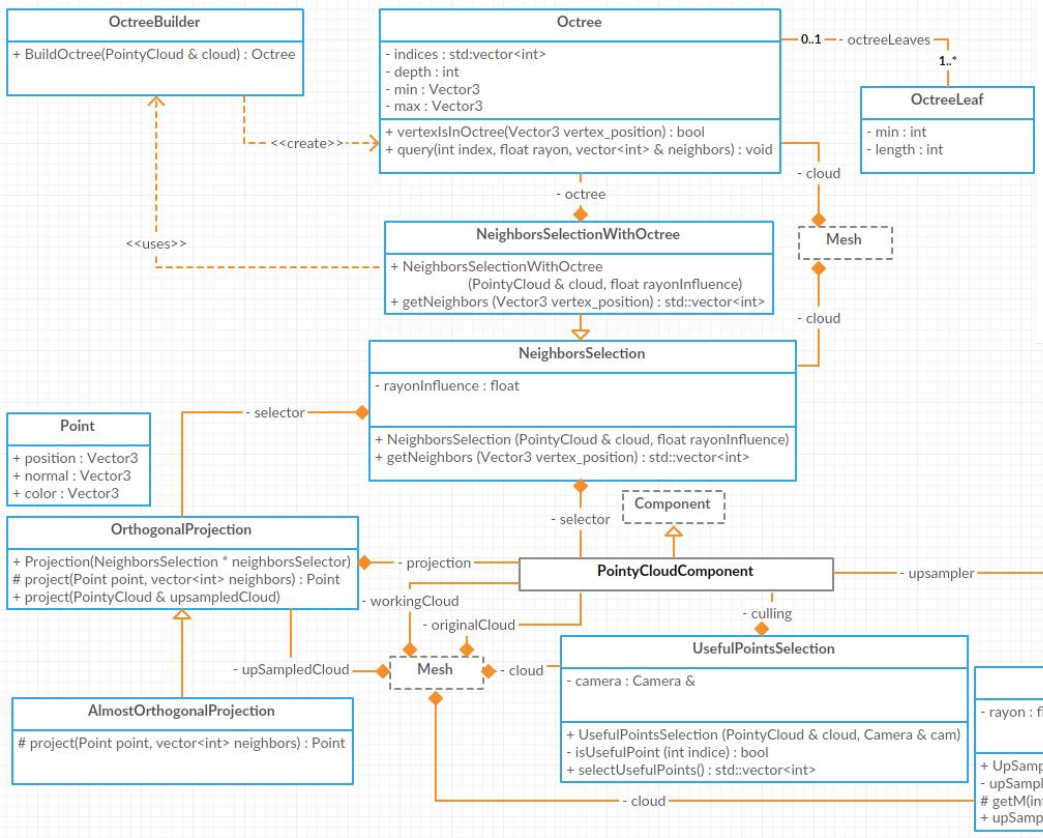
UpSampler : ré-échantillonnage

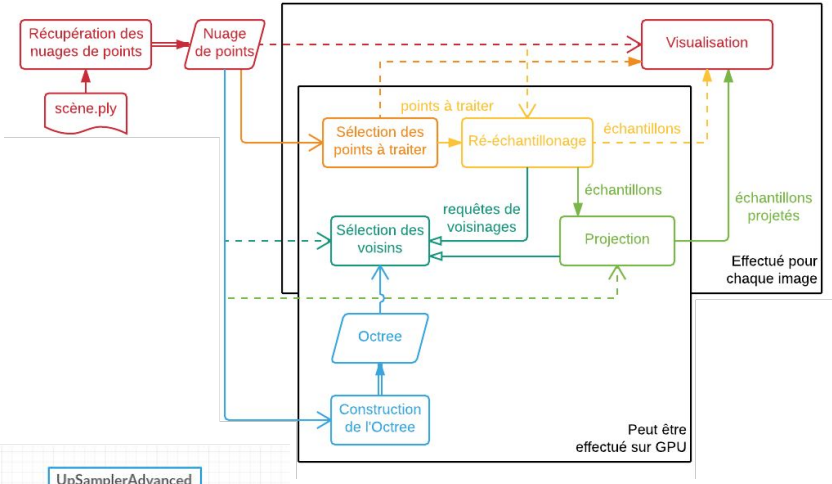
OrthogonalProjection : projection





L'architecture logicielle - APSS





NeighborsSelection :

sélection des voisins

OctreeBuilder :

construction de l'octree

Tests unitaires

Représentation des nuages de points

Id	Classe/Méthode	Données d'entrée	Résultat attendus	Protocole de vérification
1	`PointyCloudSystem::handleAssetLoading()`, `PointyCloudComponent::handlePointyCloudLoading()`	Un fichier contenant un nuage de points avec peu d'éléments	Le nuage de points créé doit contenir les même données que le fichier d'entrée	Affichage de la structure de données et comparaisons avec les données du fichier

Visualisation

2	`PointyCloudRenderer` et les différents *shaders* utilisés	Un nuage de points	Visualisation correcte du nuage	Vérification visuelle ou comparaison avec un logiciel tierce

Tests unitaires

Sélection des points à traiter

Id	Classe/Méthode	Données d'entrée	Résultat attendus	Protocole de vérification
3	`UsefulPointsSelection:: isUsefulPoint()`	Un point et une caméra orientée vers l'arrière du point et alignée sur sa normale	Retourne *false*	Un fichier *.PLY* est rempli manuellement. La caméra est orientée grâce à sa méthode *setDirection()*.
4	`UsefulPointsSelection:: isUsefulPoint()`	Un point et une caméra orientée vers l'avant du point et alignée sur sa normale	Retourne *true*	Un fichier *.PLY* est rempli manuellement. La caméra est orientée grâce à sa méthode *setDirection()*.
5	`UsefulPointsSelection:: isUsefulPoint()`	Un point et une caméra orientée vers l'arrière du point et décalée de 45° par rapport à sa normale	Retourne *false*	Un fichier *.PLY* est rempli manuellement. La caméra est orientée grâce à sa méthode *setDirection()*.
6	`UsefulPointsSelection:: isUsefulPoint()`	Un point et une caméra orientée vers l'avant du point et décalée de 45° par rapport à sa normale	Retourne *false*	Un fichier *.PLY* est rempli manuellement. La caméra est orientée grâce à sa méthode *setDirection()*.
7	`UsefulPointsSelection:: selectUsefulPoints()`	Un nuage de points dont les 10 premiers points sont alignés selon l'axe `+X` et les 10 derniers selon l'axe `-X`, et une caméra orientée dans la direction `-X`	Un vecteur contenant des indices allant de 1 à 10.	Un fichier *.PLY* est rempli manuellement. La caméra est orientée grâce à sa méthode *setDirection()*. Le vecteur retourné est finalement affiché

Tests unitaires

Ré-échantillonnage

Id	Classe/Méthode	Données d'entrée	Résultat attendus	Protocole de vérification
8	`UpSampler:: upSampleCloud()`	Un nuages de points de courbure variable	Les zones courbées se distinguent par leur haut niveau d'échantillonnage	Vérification visuelle grâce à un système de couleur

Projection

9	`OrthogonalProjection:: project()`	Un nuage de points contenus dans un plan de plus 1000 points légèrement bruitées	Toute les projections se feront dans le même plan (non bruitées) décrit par le nuage initial	Affichage des projections

Tests unitaires

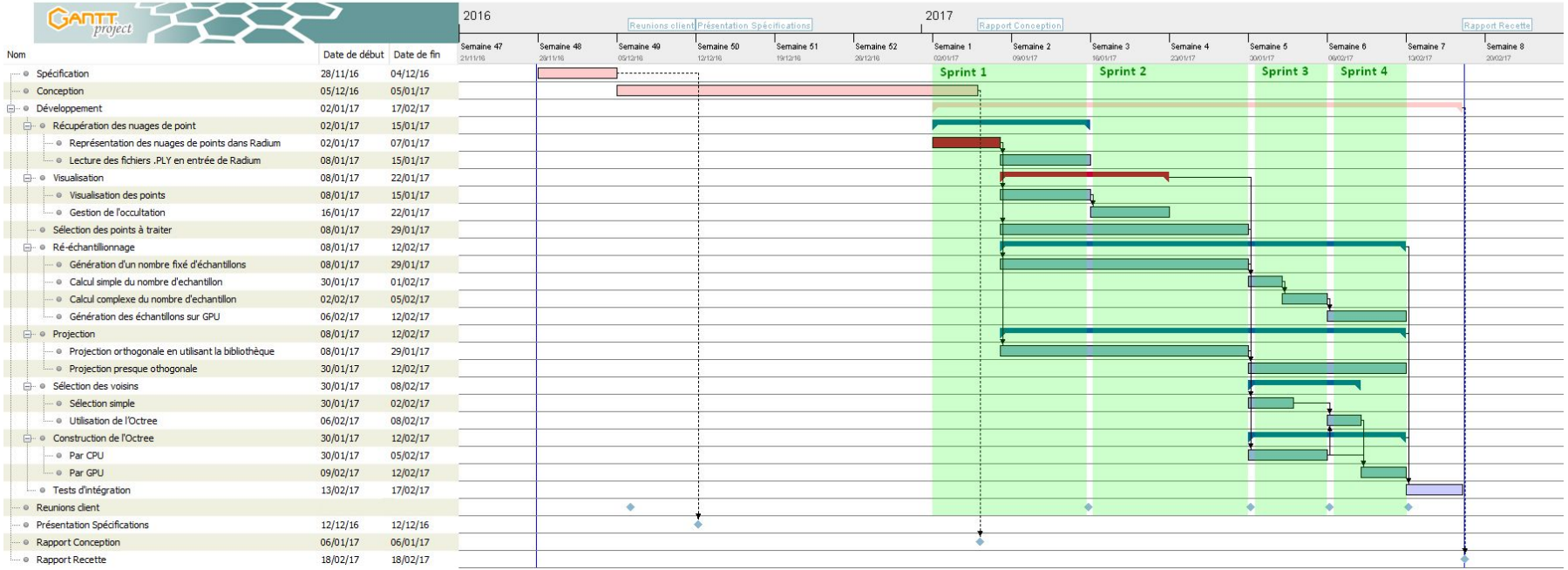
Sélection des voisins

Id	Classe/Méthode	Données d'entrée	Résultat attendus	Protocole de vérification
10	`NeighborsSelection:: getNeighbors`	Un nuage de points dont tous les points sont espacés de plus de 2 unités de longueurs, avec un rayon d'influence de 1 unité de longueur	Un vecteur vide pour toutes requêtes	Le fichier *.PLY* est créé manuellement ou de manière procédurale puis des requêtes de voisinages sont effectuées pour tous les points du nuage
11	`NeighborsSelection:: getNeighbors`	Un nuage de point contenant notamment une sphère échantillonnée de rayon de 1, et un rayon d'influence de 2 unités de longueur	La requête du voisinage du centre de la sphère retourne tous les indices des points de la sphère	Création procédurale du fichier *.PLY* puis comparaison des indices retournés avec ceux des points de la sphère

Construction de l'octree

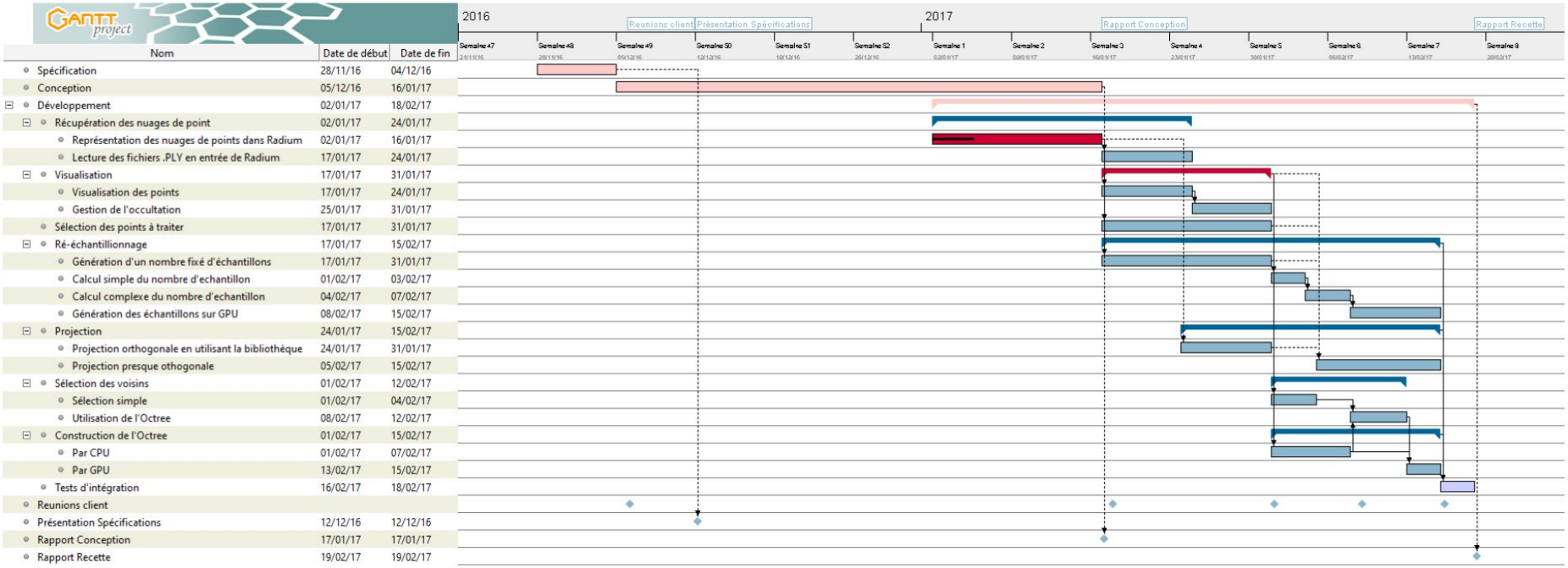
12	`OctreeBuilder:: buildOctree()`	Un nuage vide	Un Octree contenant la racine vide et aucune autre feuille	Chargement d' un fichier *.PLY* vide, et affichage des feuilles de l'Octree créé
13	`OctreeBuilder:: buildOctree()`	Un nuage de plusieurs points dont on connaît les points correspondant aux coins minimal et maximal	Les *min* et *max* de l'Octree doivent correspondre à ceux attendus	Création manuelle d' un fichier *.PLY*, puis contrôle manuel des *min* et *max*
14	`OctreeBuilder:: buildOctree()`	Un nuage de points répartis uniformément dans un cube	Un Octree contenant un point dans chaque feuille	Création procédurale d'un tel fichier d'entrée, puis comptage du nombre de points par feuille de l'Octree

Planning - Avant



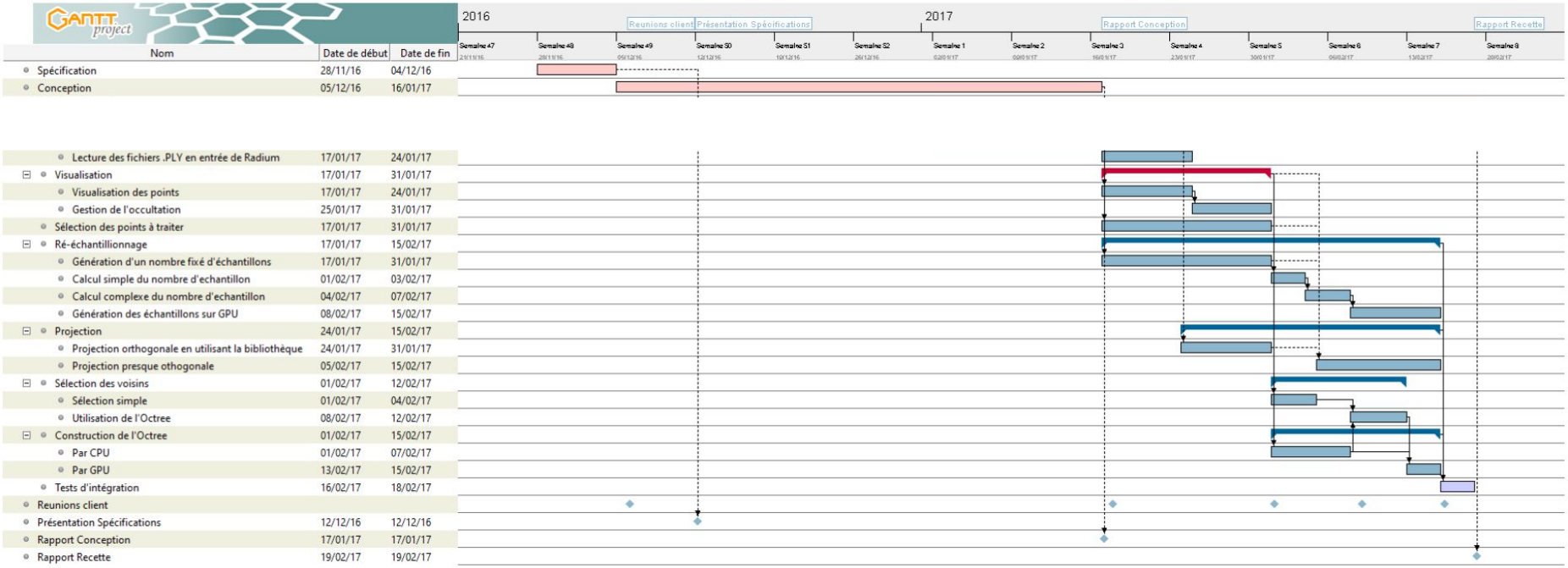
Nom	Date de début	Date de fin
Spécification	28/11/16	04/12/16
Conception	05/12/16	05/01/17
Développement	02/01/17	17/02/17
Récupération des nuages de point	02/01/17	15/01/17
Représentation des nuages de points dans Radium	02/01/17	07/01/17
Lecture des fichiers .PLY en entrée de Radium	08/01/17	15/01/17
Visualisation	08/01/17	22/01/17
Visualisation des points	08/01/17	15/01/17
Gestion de l'occultation	16/01/17	22/01/17
Sélection des points à traiter	08/01/17	29/01/17
Ré-échantillionnage	08/01/17	12/02/17
Génération d'un nombre fixé d'échantillons	08/01/17	29/01/17
Calcul simple du nombre d'echantillon	30/01/17	01/02/17
Calcul complexe du nombre d'echantillon	02/02/17	05/02/17
Génération des échantillons sur GPU	06/02/17	12/02/17
Projection	08/01/17	12/02/17
Projection orthogonale en utilisant la bibliothèque	08/01/17	29/01/17
Projection presque othogonale	30/01/17	12/02/17
Sélection des voisins	30/01/17	08/02/17
Sélection simple	30/01/17	02/02/17
Utilisation de l'Octree	06/02/17	08/02/17
Construction de l'Octree	30/01/17	12/02/17
Par CPU	30/01/17	05/02/17
Par GPU	09/02/17	12/02/17
Tests d'intégration	13/02/17	17/02/17
Reunions client		
Présentation Spécifications	12/12/16	12/12/16
Rapport Conception	06/01/17	06/01/17
Rapport Recette	18/02/17	18/02/17

Planning - Mise à jour



Nom	Date de début	Date de fin
Spécification	28/11/16	04/12/16
Conception	05/12/16	16/01/17
Développement	02/01/17	18/02/17
Récupération des nuages de point	02/01/17	24/01/17
Représentation des nuages de points dans Radium	02/01/17	16/01/17
Lecture des fichiers .PLY en entrée de Radium	17/01/17	24/01/17
Visualisation	17/01/17	31/01/17
Visualisation des points	17/01/17	24/01/17
Gestion de l'occultation	25/01/17	31/01/17
Sélection des points à traiter	17/01/17	31/01/17
Ré-échantillionnage	17/01/17	15/02/17
Génération d'un nombre fixé d'échantillons	17/01/17	31/01/17
Calcul simple du nombre d'echantillon	01/02/17	03/02/17
Calcul complexe du nombre d'echantillon	04/02/17	07/02/17
Génération des échantillons sur GPU	08/02/17	15/02/17
Projection	24/01/17	15/02/17
Projection orthogonale en utilisant la bibliothèque	24/01/17	31/01/17
Projection presque othogonale	05/02/17	15/02/17
Sélection des voisins	01/02/17	12/02/17
Sélection simple	01/02/17	04/02/17
Utilisation de l'Octree	08/02/17	12/02/17
Construction de l'Octree	01/02/17	15/02/17
Par CPU	01/02/17	07/02/17
Par GPU	13/02/17	15/02/17
Tests d'intégration	16/02/17	18/02/17
Reunions client		
Présentation Spécifications	12/12/16	12/12/16
Rapport Conception	17/01/17	17/01/17
Rapport Recette	19/02/17	19/02/17

Planning - Mise à jour



Gestion des risques : Rappel

L'organisation modules :

Représentation des nuages de points

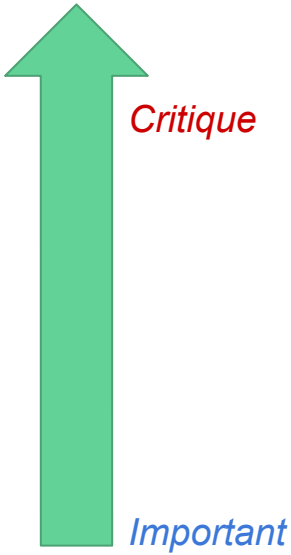
Visualisation des points

Sélection des points à traiter

Génération d'un nombre fixe d'échantillons

Projection

Lecture des fichiers en .PLY



Merci de votre attention

Avez-vous des questions ?