

# Spécifications

## Dynamic Sampling and Rendering of Algebraic Point Set Surfaces

William Caisson

Xavier Chalut

Christophe Claustre

Thibault Lejembre

*Client : Nicolas Mellado*

# CONTEXTE

Objectif

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

Plugin dans Radium-Engine implémentant

***Dynamic Sampling and Rendering of Algebraic Point Set Surfaces***

Gaël Guennebaud

Marcel Germann

Markus Gross

# Exigences principales

P1	Plugin Radium
P2	Fonctionnel sur Linux avec carte graphique NVIDIA (support CUDA)
P3	Afficher un nuage de points
P4	Extraire un nuage de points des fichiers .PLY donnés en entrée
P5	1 millions de points / 50 images par seconde

P6	Code source du plugin
P7	Documentation utilisateur

# Exigences secondaires

S1	2 millions de points / 50 images par seconde
S2	Aucune ouverture indésirable dans les surfaces
S3	Afficher une surface lisse
S4	Fonctionnel sur Windows et MacOS

# SOMMAIRE

## Système

Vue générale

Les modules

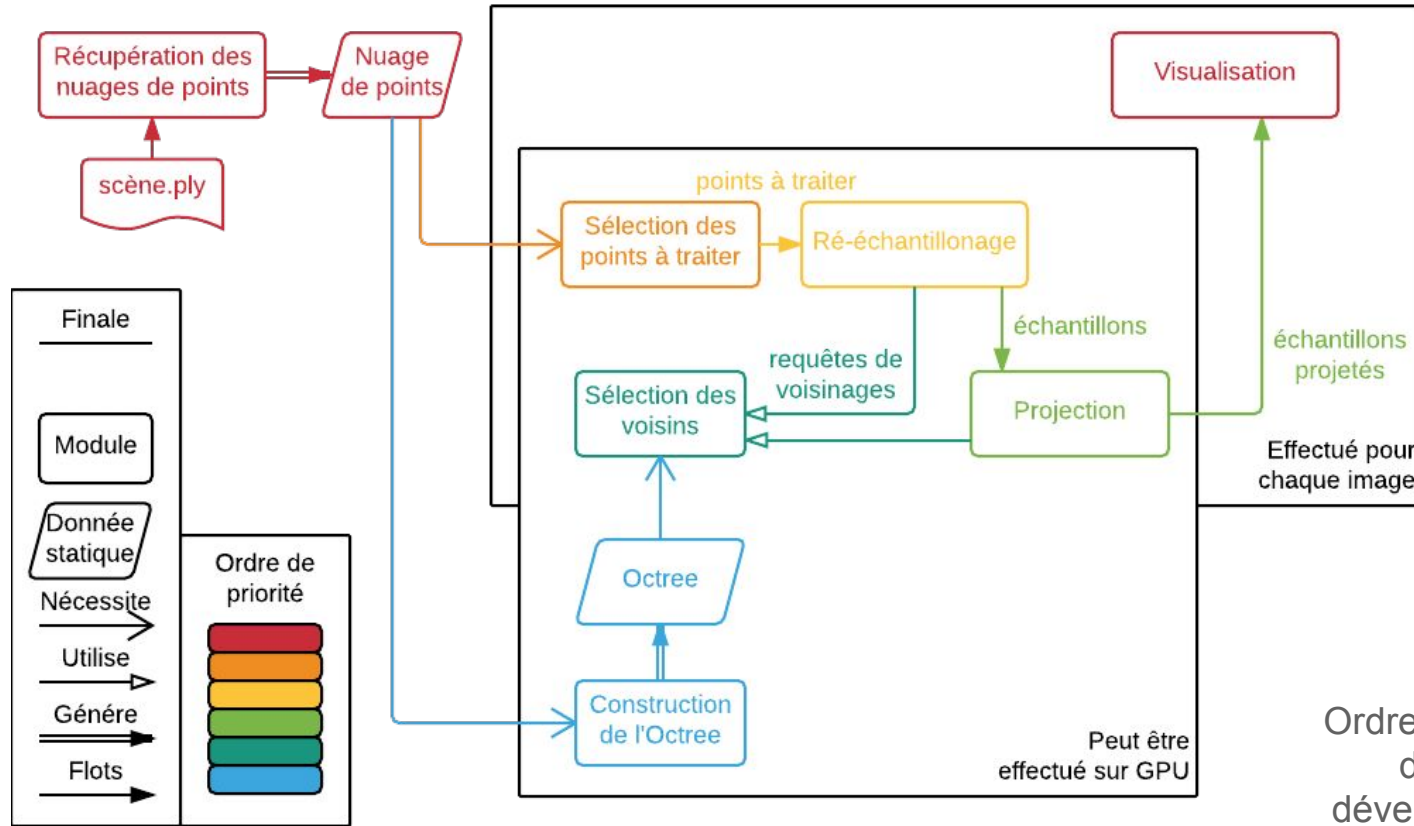
## ORGANISATION

Cycle de développement

Planning prévisionnel

Gestion des risques

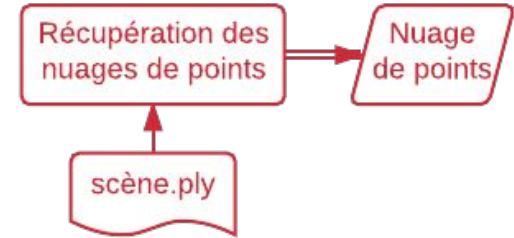
# Vue générale - Le diagramme des modules



Ordre de priorité  
dans le  
développement

# Les modules - Récupérations des nuages de points

<b>Entrée</b>	Un fichier .PLY
<b>Sortie</b>	Un nuage de points
<b>Objectif</b>	Extraire du fichier .PLY un nuage de points ➡ Utilisation de <u>ASSIMP</u>



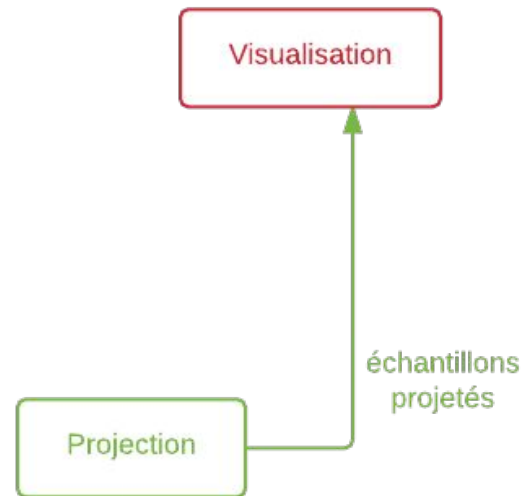
Bonus : Afin de pouvoir tester la lecture ➡ écriture

# Les modules - Visualisation

**Entrée** Des points

**Sortie** Une visualisation correcte de ces points

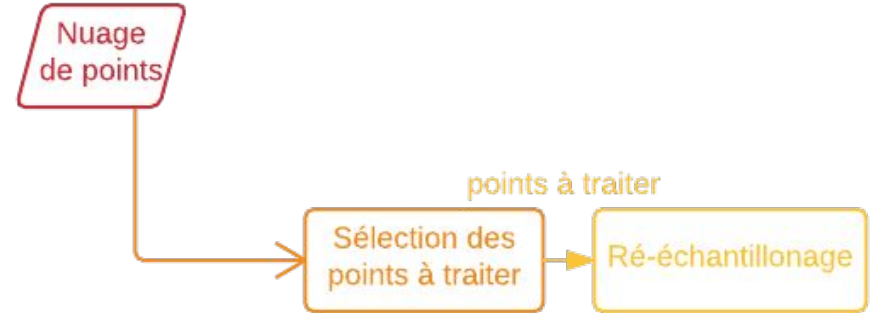
**Objectif** Afficher correctement des points dans un espace 3D du point de vue de la caméra





# Les modules - Sélection des points à traiter

- Entrée** Un nuage de points
- Sortie** Des points à traiter
- Objectif** Sélectionner uniquement les points qui ont besoin d'être traités/affichés

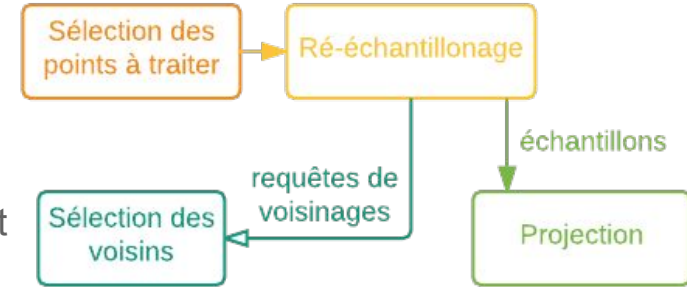


# Les modules - Ré-échantillonnage

**Entrée** Un point, ses voisins

**Sortie** Plusieurs échantillons

**Objectif** Générer un nombre correcte d'échantillon correspondant au point reçu en entrée en fonction de ses voisins

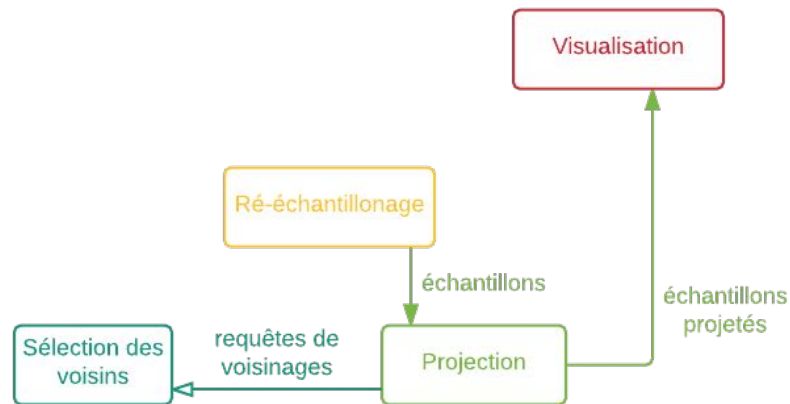


# Les modules - Projection

**Entrée** Un point, ses voisins

**Sortie** Un point projeté

**Objectif** Projeter correctement le point reçu en entrée sur la surface décrite par ses voisins  
➡ Utilisation de Patate

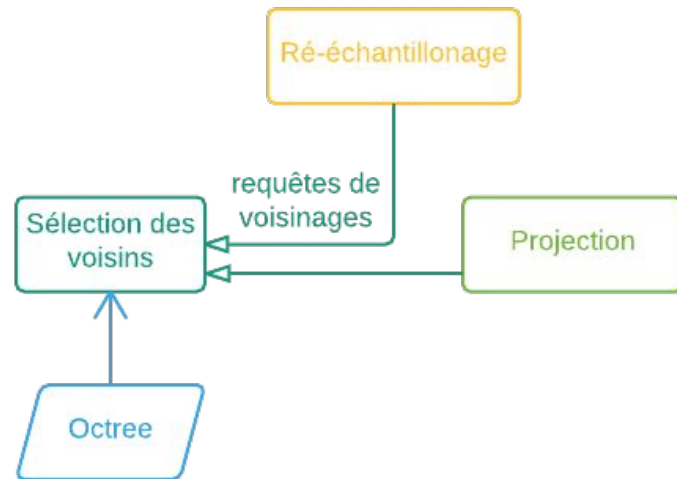


# Les modules - Sélection des voisins

**Entrée** Un point, un nuage de points organisé en octree

**Sortie** Les voisins du point en entrée

**Objectif** Sélectionner efficacement les voisins du point reçu en entrée

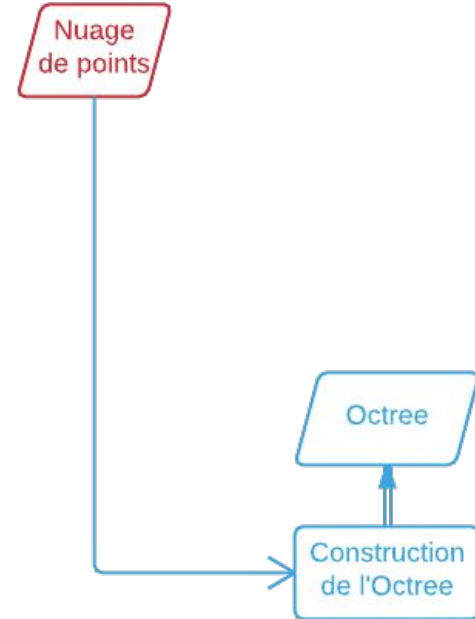


# Les modules - Construction de l'Octree

**Entrée** Un nuage de points

**Sortie** Un Octree

**Objectif** Générer efficacement l'Octree décrivant le nuage de points en entrée





# Cycle de développement

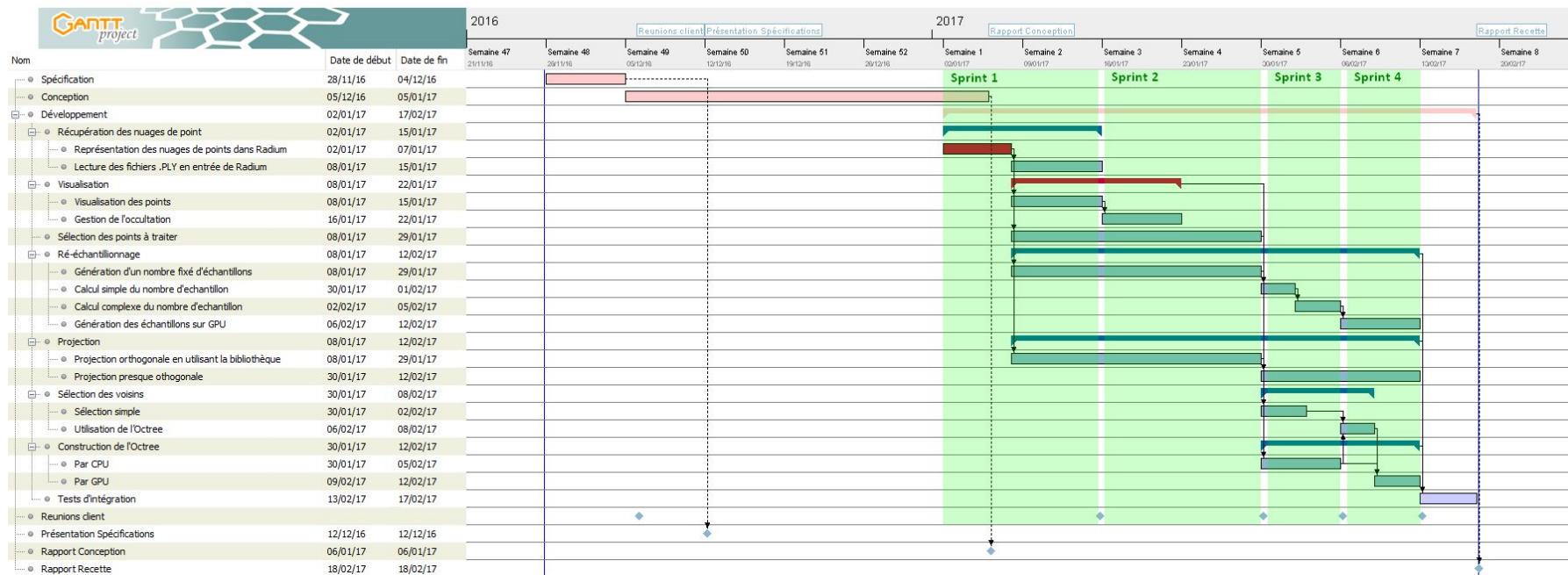
## Méthodes Agiles

- Sprint de 1 à 2 semaines
- 1 réunion par sprint avec le client
- Répartition du travail en tâches

## Objectifs

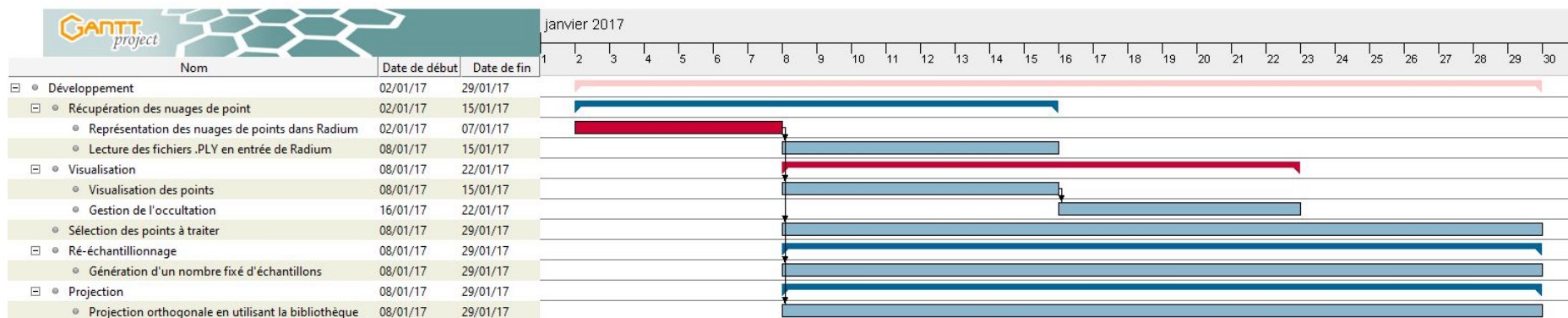
- Obtenir des résultats visibles à chaque sprint
- Garder un contact régulier avec le client/encadrant

# Planning prévisionnel

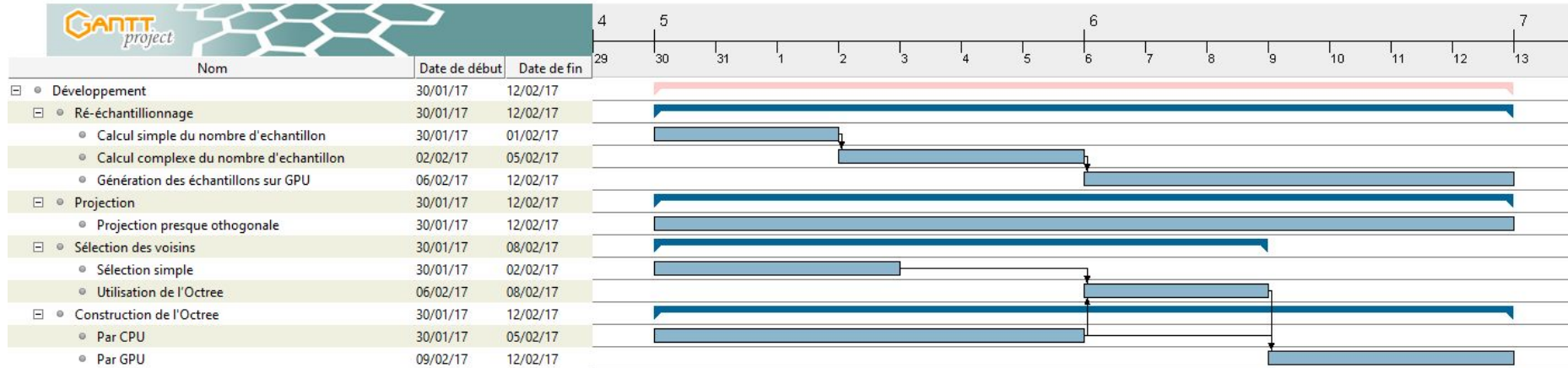




# Planning prévisionnel



# Planning prévisionnel



# Gestion des risques

Objectif prioritaire : Obtenir le rendu d'un nuage de point

Priorités	Tâches
Critiques	<ul style="list-style-type: none"><li>- Représentation du nuage de points</li><li>- Visualisation des points</li></ul>
Très important	<ul style="list-style-type: none"><li>- Sélection des points à traiter</li></ul>
Important	<ul style="list-style-type: none"><li>- Génération d'un nombre fixé d'échantillon</li><li>- Projection en utilisant l'API Patate</li></ul>
Assez important	<ul style="list-style-type: none"><li>- Lecture des fichiers .PLY</li></ul>

Merci de votre attention

Avez-vous des questions ?