

Chef d'Oeuvre M2 IGAI

Encadrant : M. PAULIN

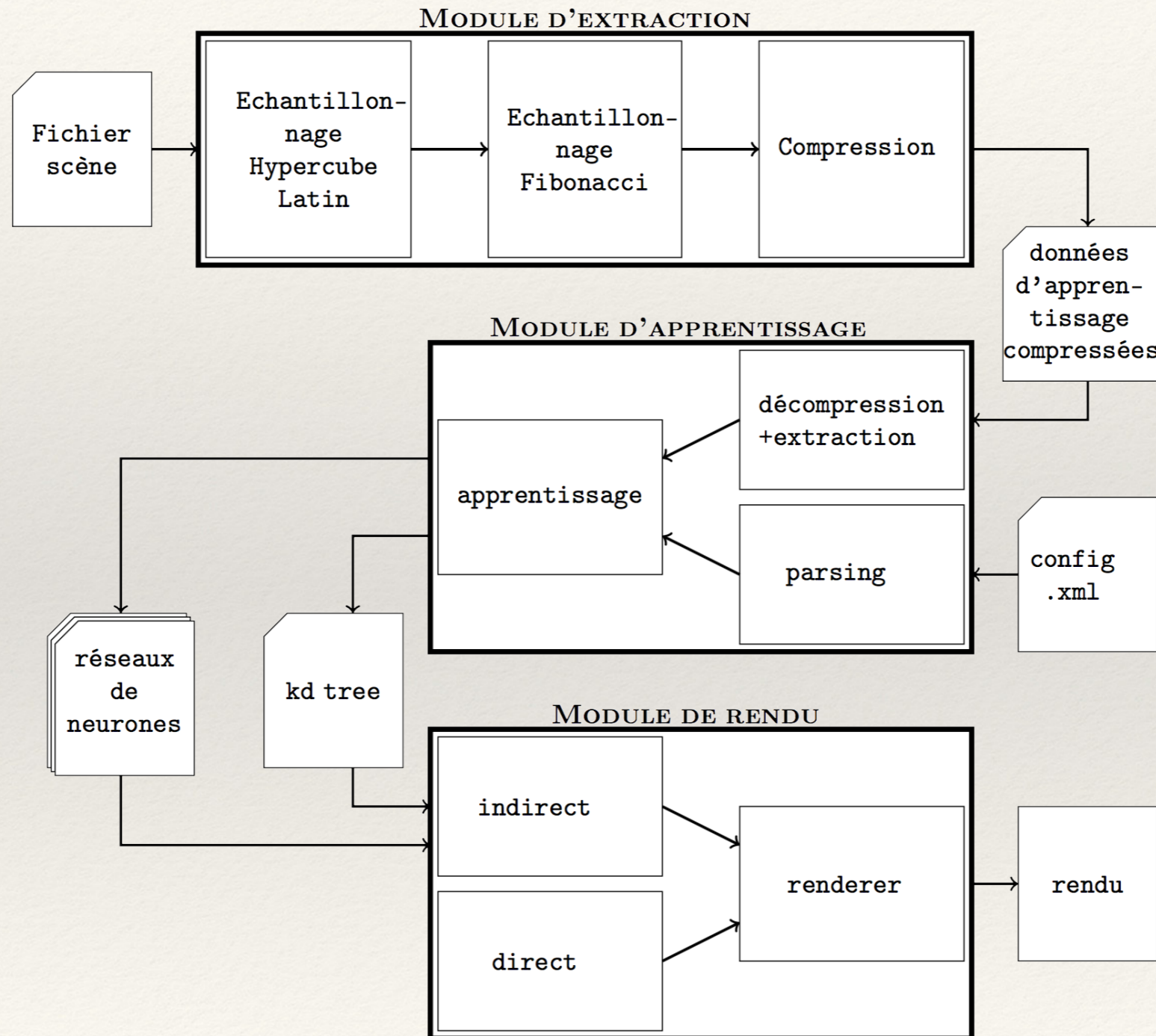
Global Illumination with Radiance Regression Functions

Yannick BERNARD
Pierre GUERINEAU
Kevin MENIEL
Romain MOUTRILLE
Matthias ROVES

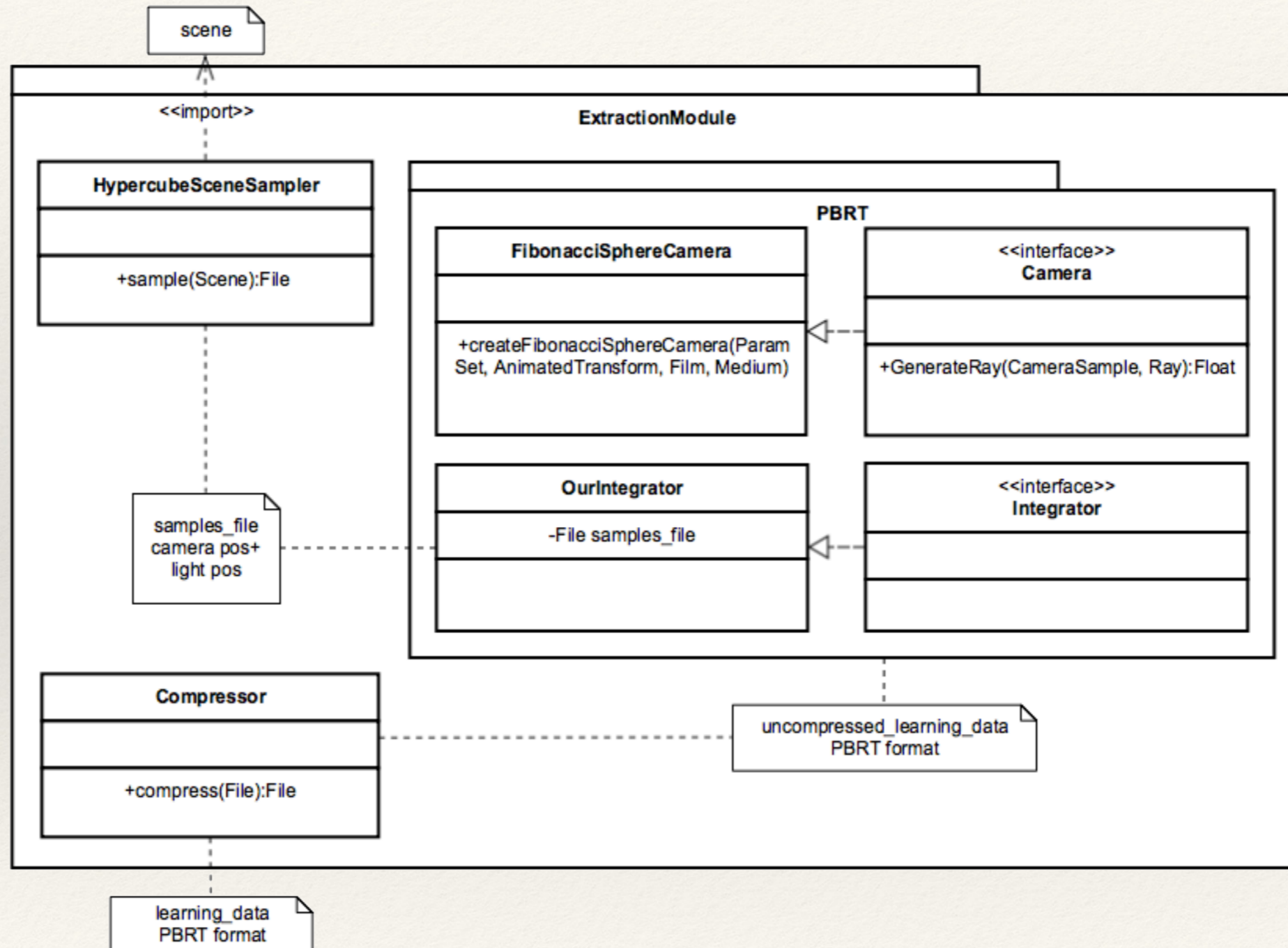
Introduction



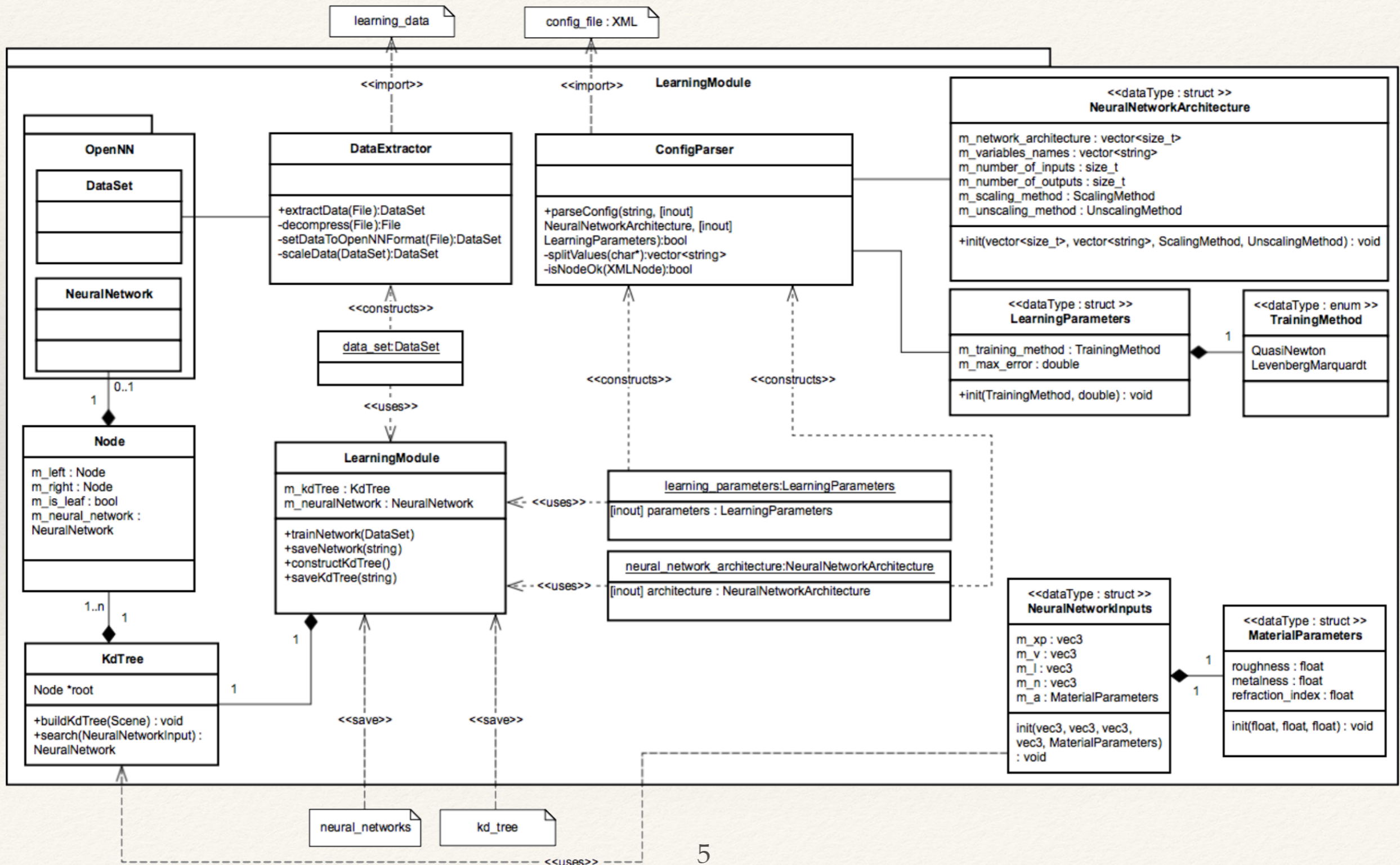
Vue d'ensemble détaillée



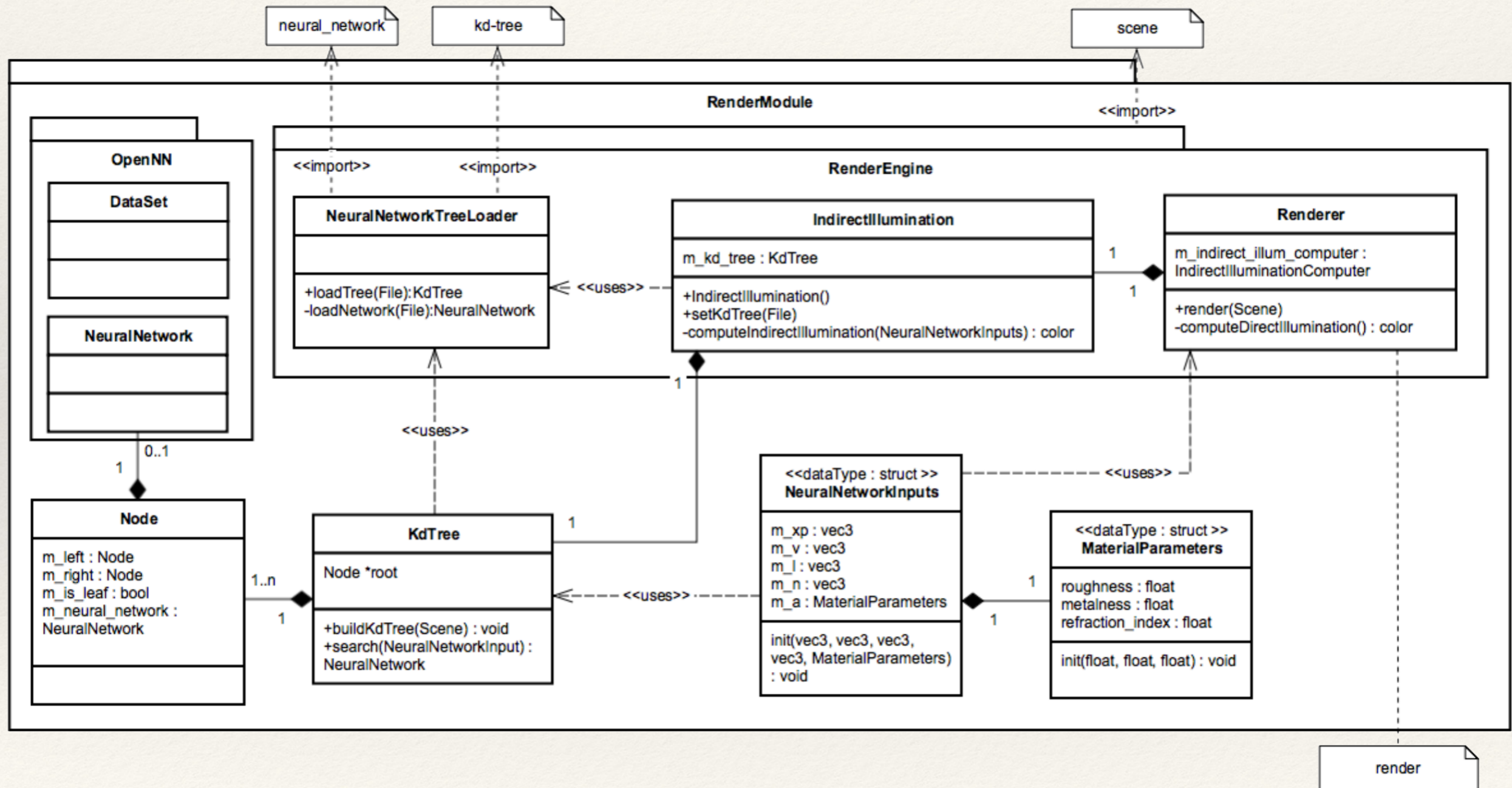
Décomposition en classes



Décomposition en classes



Décomposition en classes



Tests unitaires - Extraction

- ❖ **Classe HypercubeSceneSampler**

- ❖ sample - Affichage des positions des samples dans la scène à l'aide d'OpenGL (affichage de GL_POINT)

- ❖ **Classe Compressor**

- ❖ compress - Vérification de l'intégrité des fichiers dans le module de compression

- ❖ **Classe Camera**

- ❖ generateRay - Verification des bonnes propriétés pour des rayons connus.

- ❖ **Classe FibonacciSphereCamera**

- ❖ createFibonacciSphereCamera - Verification de l'uniformité des directions des rayons pouvant être générés via un affichage en OpenGL.

Tests unitaires - Extraction

- ❖ Verification de l'exactitude des données
 - ❖ Scène simple
 - ❖ Modification paramètre
- ❖ Capacité à traiter n'importe quelle scène
 - ❖ Différentes scènes

Tests unitaires - Apprentissage

- ❖ **Classe LearningModule**

- ❖ `constructKdTree` - affichage de KdTrees avec OpenGL pour vérifier leur bonne construction

- ❖ **Classe DataExtractor**

- ❖ `extractData` - vérification que le nombre d'inputs/outputs du `opennn::DataSet` généré est correct
- ❖ `decompress` - vérification de l'intégrité des données après décompression
- ❖ `setDataToOpenNNFormat` - la construction du `opennn::DataSet` ne doit pas retourner d'erreur

Tests unitaires - Apprentissage

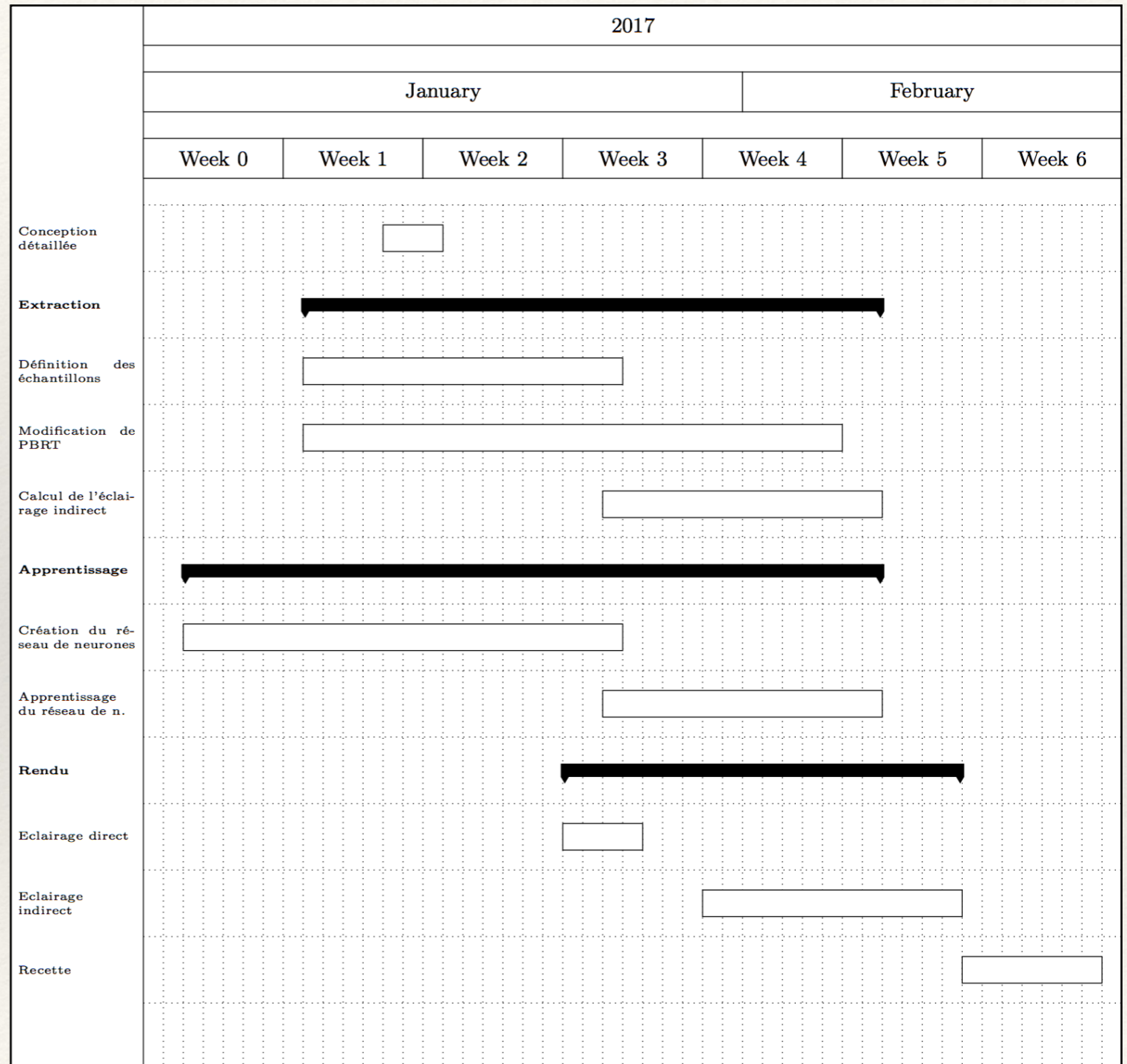
- ❖ **Verification de la déduction**
 - ❖ 3 données n_1, n_2, n_3 $n_1 \approx n_3$
- ❖ **Vérification de l'exactitude des données**
 - ❖ Cross-validation (70% apprentissage, 30% tests)

Tests unitaires - Rendu

- ❖ **Verification de l'exactitude des données**
 - ❖ Comparaison avec/sans composante indirecte
 - ❖ Vérification visuelle
 - ❖ Comparaison avec PBRT

Planning prévisionnel

NOM	DATE DÉBUT	DATE FIN
Conception détaillée	14/01/17	16/01/17
Extraction	10/01/17	07/02/17
Définition des échantillons	10/01/17	25/01/17
Modification de PBRT	10/01/17	05/02/17
Calcul de l'éclairage indirect	25/01/17	07/02/17
Apprentissage	04/01/17	07/02/17
Création du réseau de n.	04/01/17	25/01/17
Apprentissage du réseau de n.	25/01/17	07/02/17
Rendu	23/01/17	11/02/17
Eclairage direct	23/01/17	26/01/17
Eclairage indirect	30/01/17	11/02/17
Recette	12/02/17	18/02/17



Analyse des risques

n°	Type	Description	Probabilité	Impact	Solutions	
					Évité	Accepté
1	–	Impossibilité de compresser les données d'apprentissage	●	●	Implémenter un compresseur avant l'apprentissage	Ne pas compresser les données
2	–	Le KdTree se construit mal	●	●	Implémenter l'algorithme de construction au plus tôt	Plans de coupe aléatoires
3	–	Durée trop grande d'apprentissage	●	●	Commencer l'apprentissage le plus tôt possible et en connaître le temps estimé	On n'exploite que les données valides
4	–	Impossibilité d'utiliser l'algorithme de Levenberg Marquardt	●	●	Faire fonctionner l'algorithme au plus tôt	Utiliser l'algorithme QuasiNewton
5	–	Difficultés d'utilisation de PBRT	●	●	Se documenter et se former au plus tôt sur PBRT	Utiliser Mitsuba
6	–	Ne pas avoir une base fonctionnelle de notre moteur	●	●	Réaliser le module d'évaluation au plus tôt	Utiliser le Radium Engine
7	–	Impossible de paramétrer l'erreur d'apprentissage	●	●	Se documenter au plus tôt sur OpenNN	Modifier les fichiers OpenNN
8	–	Impossible d'utiliser OpenNN pour chaque fragment lors du rendu	●	●	Se documenter au plus tôt sur OpenNN	Modifier les fichiers OpenNN
9	–	Difficultés lors du développement CUDA	●	●	Linker CUDA et vérifier son fonctionnement avant le développement du module de rendu	Traiter les parties CUDA sur CPU

Probabilité & Impact : ● faible ● moyen ● forte ● très fort

Questions ?