



---

# Chef d'œuvre - Conception

## Approximation d'éclairage indirect en temps réel

---

Blaise Cardonne    Gauthier Bouyjou    Valentin Camus  
Rihab Elrifai      Sylvain Durand

Encadrant/Client : François Desrichard

Janvier  
2020

## Contents

1	Introduction	3
2	Vue d'ensemble détaillée du système	4
3	Décomposition en classes	4
4	Tests unitaires	5
5	Planification mise à jour	5
6	Risques mis à jour	7

# 1 Introduction

Dans le cadre de notre chef d'œuvre sur l'approximation d'éclairage indirect en temps réel, nous allons cette fois vous présenter les différents choix de conception que nous avons effectués pour répondre aux exigences fonctionnelles et aux diverses contraintes qui ont été définies précédemment lors de la spécification.

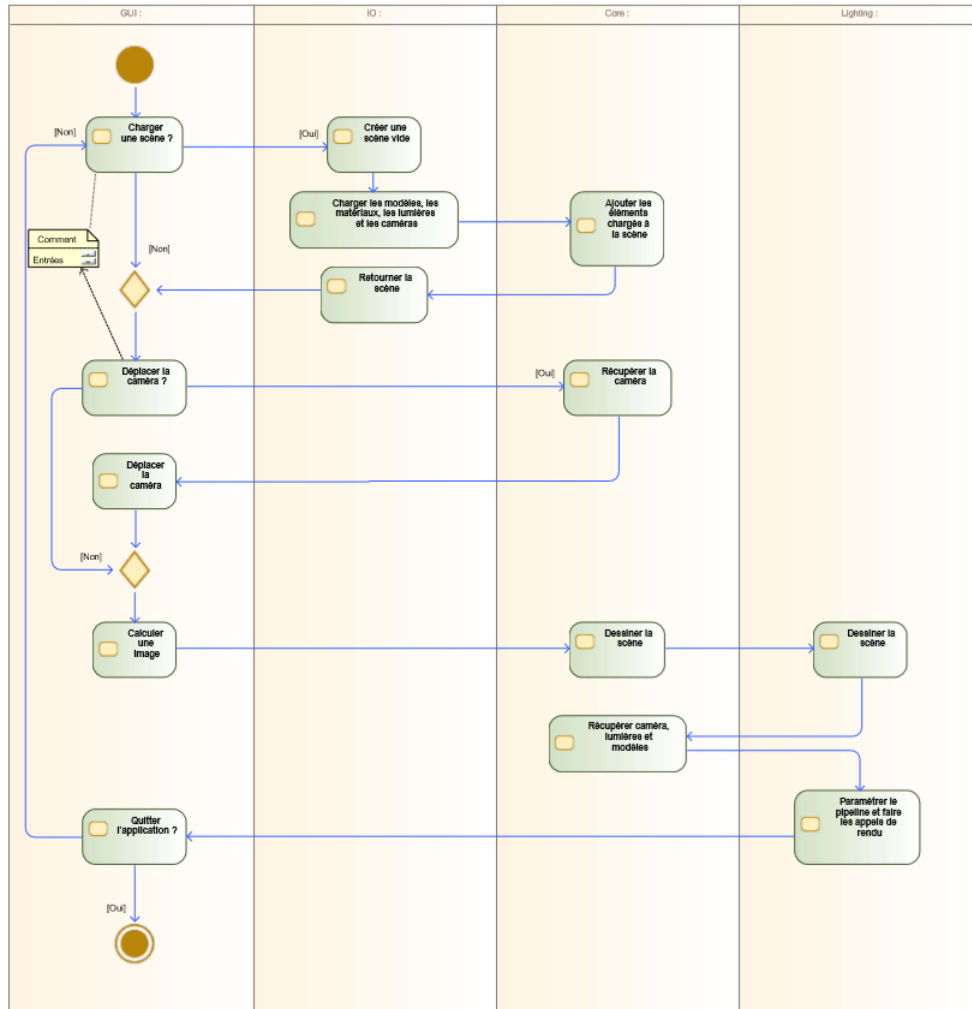


Figure 1: Diagramme d'activité global du système

## 2 Vue d'ensemble détaillée du système

Pour obtenir une vue détaillée des interactions inter-modules du système, nous avons produit le diagramme d'activité ci-dessus (voir figure 1). Nous ne détaillons pas dans ce diagramme comment chacune des tâches est accomplie intra-module, notamment parce qu'ici l'objectif est simplement d'avoir une idée globale du fonctionnement du système.

## 3 Décomposition en classes

Nous avons produit le diagramme de classe suivant (voir figure 2, l'image est également annexée à ce document) à partir de la décomposition en module précédemment effectuée. La séparation par les couleurs permet d'identifier quelles classes relèvent de quel module. Cette décomposition va nous permettre de mieux estimer le temps nécessaire pour implanter chacun des modules et de bien répartir le travail au sein des binômes. Bien évidemment ce diagramme ne contient pas la totalité des méthodes et attributs de la démo finale: ce serait trop complexe à effectuer et nous ne pouvons de toute façon pas anticiper les choix de factorisation du code qui seront effectués dans chacun des modules.

On y trouve cependant la totalité des attributs importants (pour la compréhension du rôle de chaque classe) et les méthodes qui seront utilisées ailleurs qu'en intra-module. Autrement dit, les interfaces entre les modules sont clairement définies.

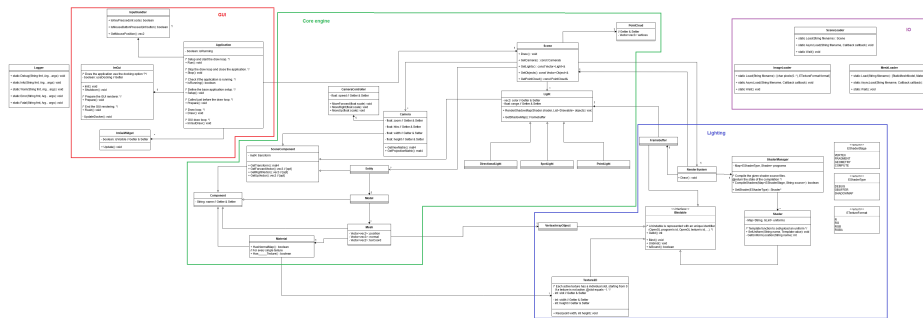


Figure 2: Diagramme de classe du système

Ce diagramme de classe permet justement de bien mettre en avant les dépendances entre les modules qui étaient difficile à extraire lors de la phase de spécifications uniquement à partir de la décomposition en module.

## 4 Tests unitaires

Comme pour toute application, nous avons prévu un ensemble de jeux de tests pour garantir autant que possible le fonctionnement des points critiques du système.

Nous avons par exemple prévu des tests pour:

- Assurer la bonne orientation de la caméra après qu’une transformation ait été appliquée
- Vérifier l’état de la scène après un chargement par le “SceneLoader”
- Vérifier que la compilation de shaders fonctionne via le booléen de retour

Dans cette application, il y a en réalité assez peu de code réellement testable unitairement qu’on peut considérer comme “à risque”. Cela est dû au fait que la majorité du travail “difficile” (comme identifié lors de la spécification) sera développée dans des shaders que nous ne pourrons pas tester comme du code traditionnel.

## 5 Planification mise à jour

Comme lors de la planification initiale (produite au moment de la spécification), nous avons utilisé un diagramme de Gantt pour positionner l’ensemble des tâches à effectuer dans le temps (voir figure 3, l’image est également annexé à ce document). La modification majeure est dû aux examens qui ont été positionnés lors des deux dernières semaines de Janvier et qui vont venir décaler le travail initialement prévu pour la première itération et réduire la marge prévue pour les fonctionnalités optionnelles.

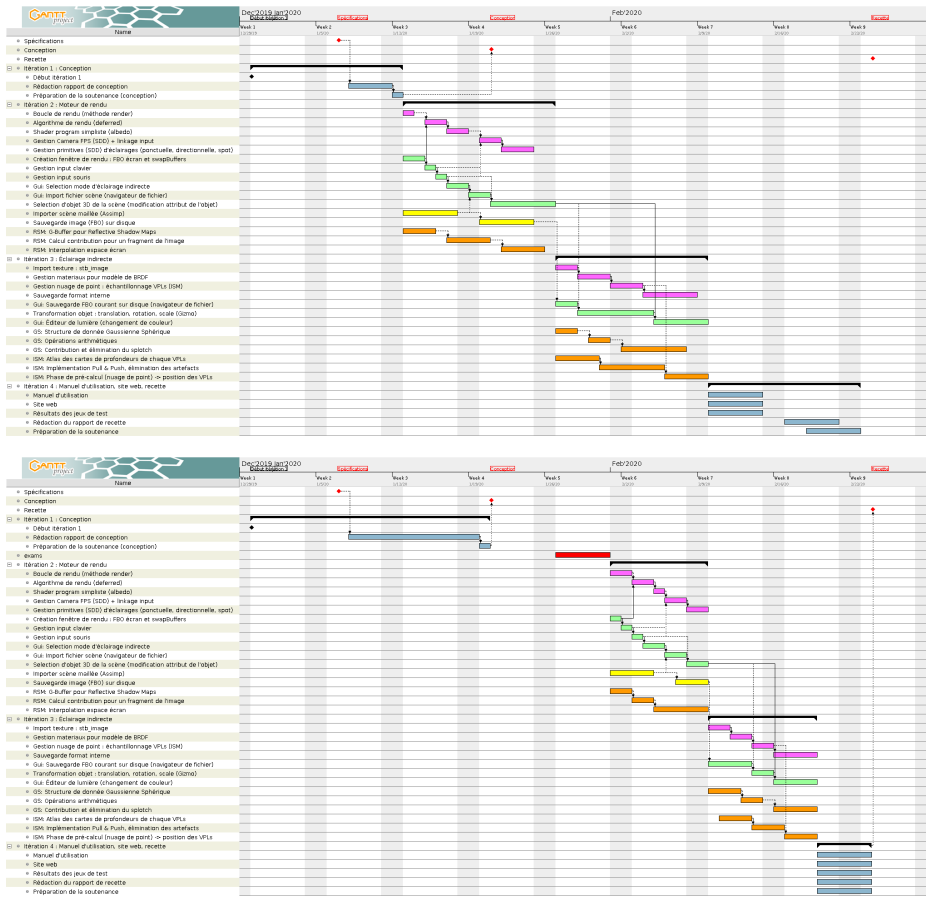


Figure 3: Diagramme de Gantt prévisionnel initial (en haut) et à jour (en bas)

Nous avons également comme prévu créé le Trello (voir figure 4) qui reprend les fonctionnalités listées dans le Gantt.

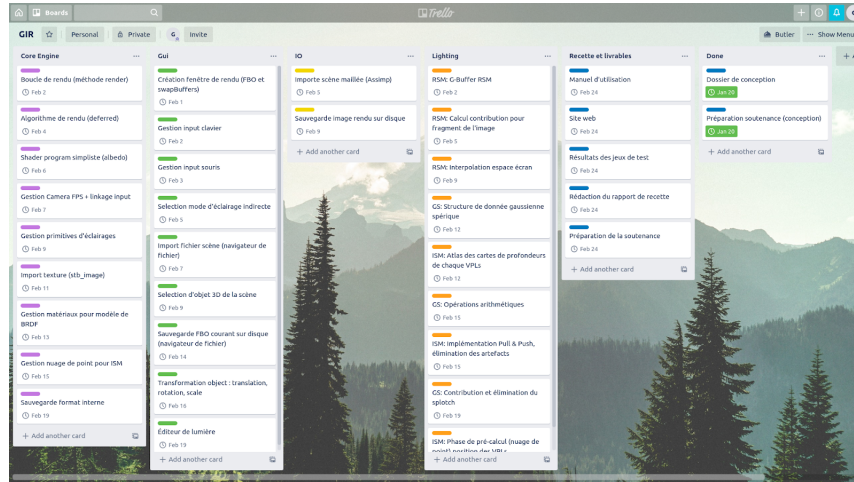


Figure 4: Tâches à effectuer et planning prévisionnel via Trello

## 6 Risques mis à jour

Du côté de la gestion des risques (voir figure 5), on observe assez peu de changement. Nous avons simplement modifié les solutions qui avait été critiquées lors de la présentation sur la spécification et retiré le risque qui concernait les dates de partiels étant donné que celui-ci s'est déclenché (d'où les modifications qui ont été apportées au diagramme de Gantt).

Risque	Probabilité d'apparition	Impact	Prévention	Solution
Indisponibilité/abandon d'un membre de l'équipe au cours du projet	Faible	Fort	Faire en sorte que sur le planning des tâches il y ait des tâches faites en binôme.	Les tâches en binôme deviendront des tâches pour une seule personne.
Base moteur instable	Modérée	Moyen	Prendre exemple sur moteur existant (code source libre) et tester rigoureusement le code produit.	On n'aura d'autres choix que de corriger les problèmes qui apparaissent à la volée.
Perte de temps sur le moteur alors que le coeur du sujet du chef d'oeuvre est l'éclairage indirecte	Modérée	Moyen	L'objectif est de faire une base moteur minimaliste afin de mieux se concentrer sur la partie "éclairage".	Faire une très bonne prévision des tâches et une conception détaillée précise évitant les questions lors des sprints.
Performance décevantes	Forte	Faible	Utiliser des techniques d'optimisations.	Se référer aux implémentations existantes et utiliser des outils de mesures de performances et de "profiling".
Compatibilité multi OS	Forte	Faible	Il faut que tous les membres de l'équipe puissent compiler sur son OS pour développer sereinement.	Remplacer les morceaux de code non portable. Dans le pire des cas, abandonner la compatibilité pour un OS.
Dépassement des délais	Modérée	Fort	Effectuer une planification temporelle assez large.	Abandonner des fonctionnalités optionnelles.
Mauvaise estimation des délais (trop court)	Faible	Moyen	Prévoir une marge dans l'estimation des délais dès le départ.	Reprendre le planification des délais. Impossibilité de prévoir des marges car nous devons tout développer en peu d'itérations.
Manque de communication dans le groupe	Faible	Moyen	Utiliser un bon outil de communication dès le départ.	Utiliser un autre logiciel de communication.

Figure 5: Tableau des risques à jour