

Approximation d' éclairage indirect en temps réel: Conception détaillée

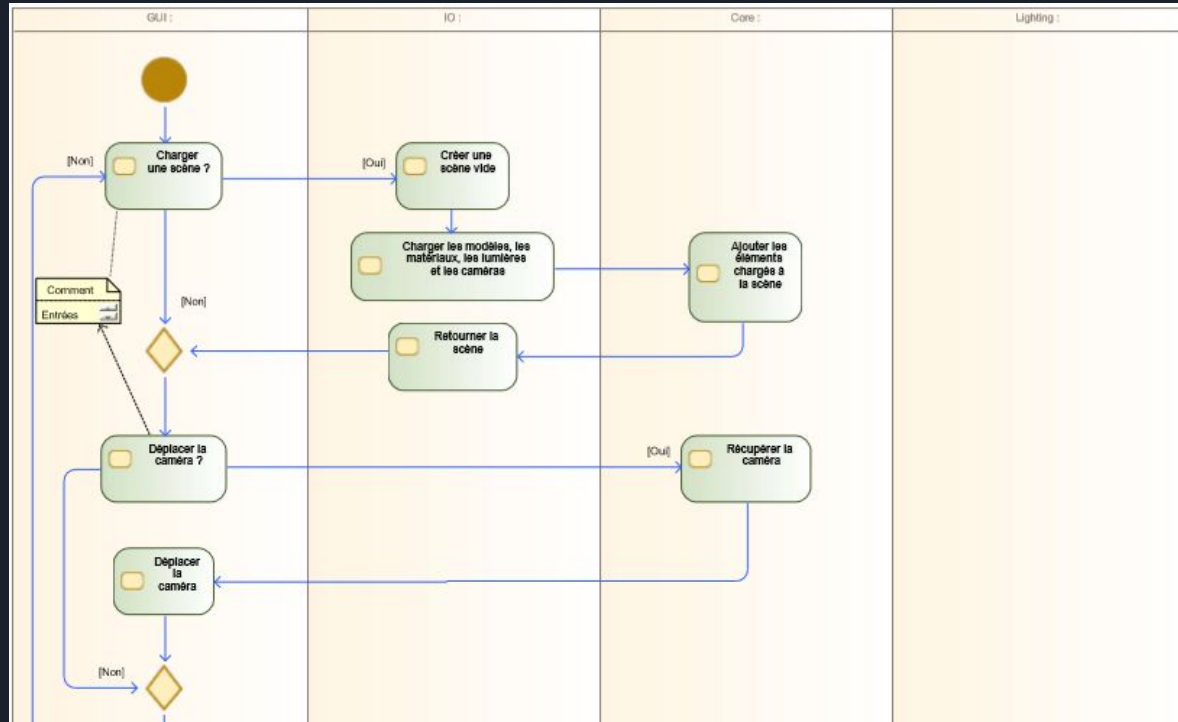
Blaise Cardonne, Gauthier Bouyjou, Valentin Camus, Rihab Elrifai, Sylvain Durand
Encadrant / Client : François Desrichard



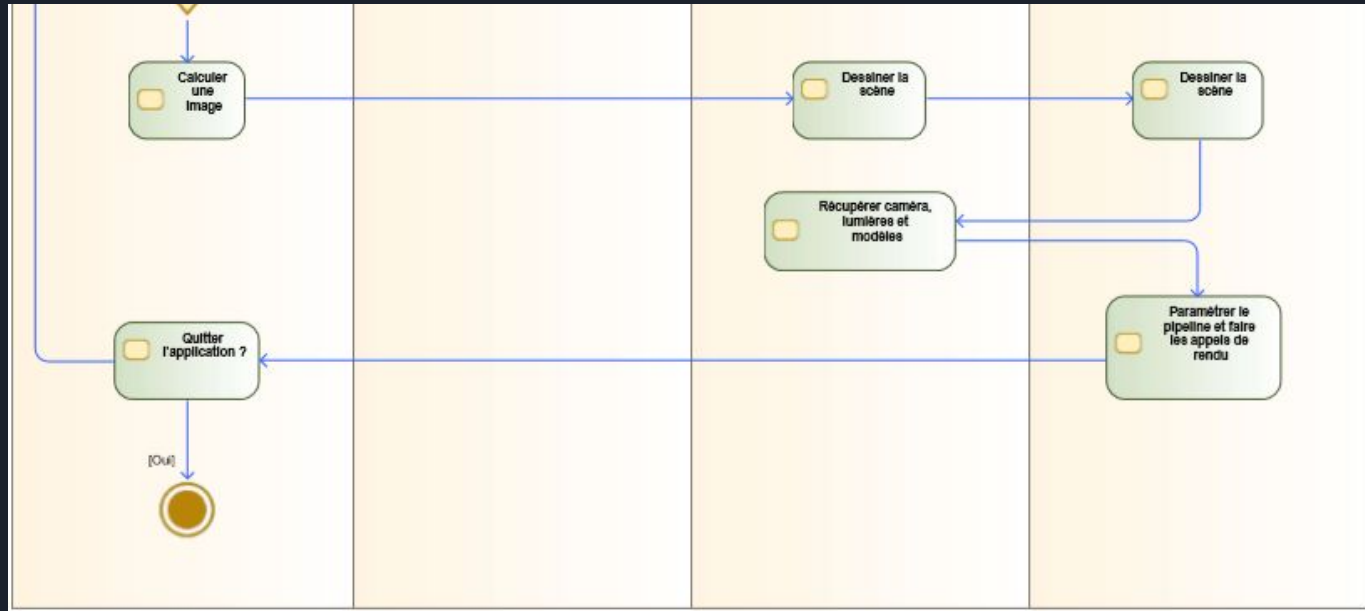
Axes de la présentation

1. Vue d'ensemble
2. Classes et structures de données
3. Tests unitaires
4. Révision du Planning
5. Révision des risques

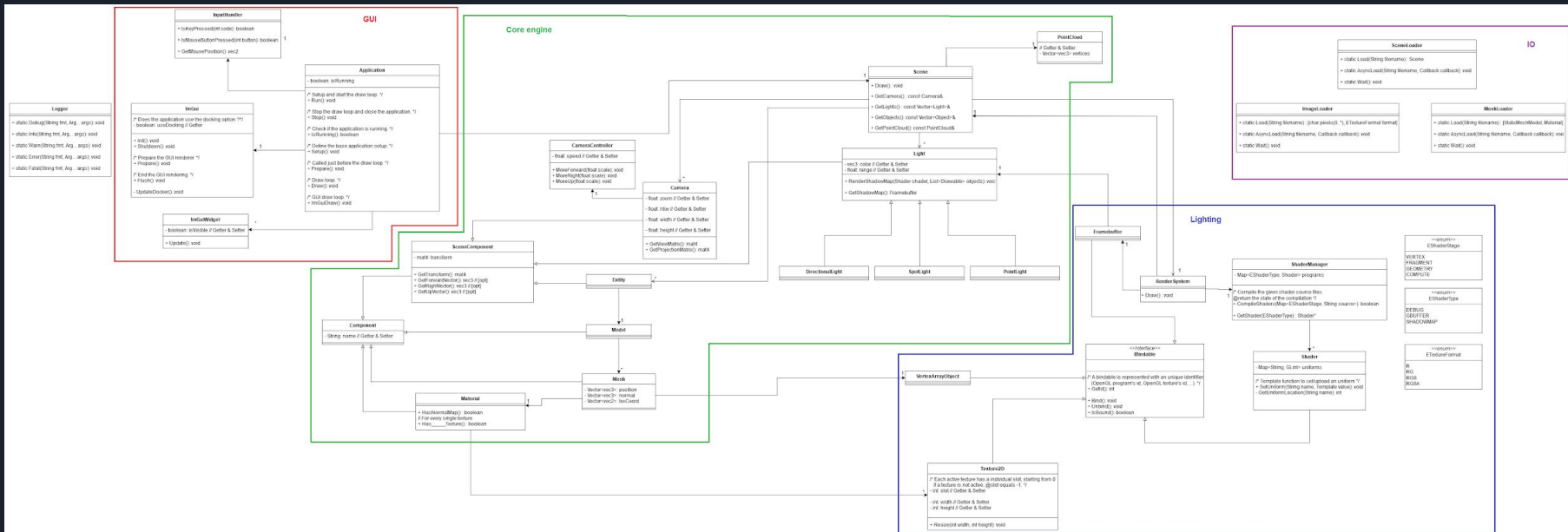
Vue d'ensemble



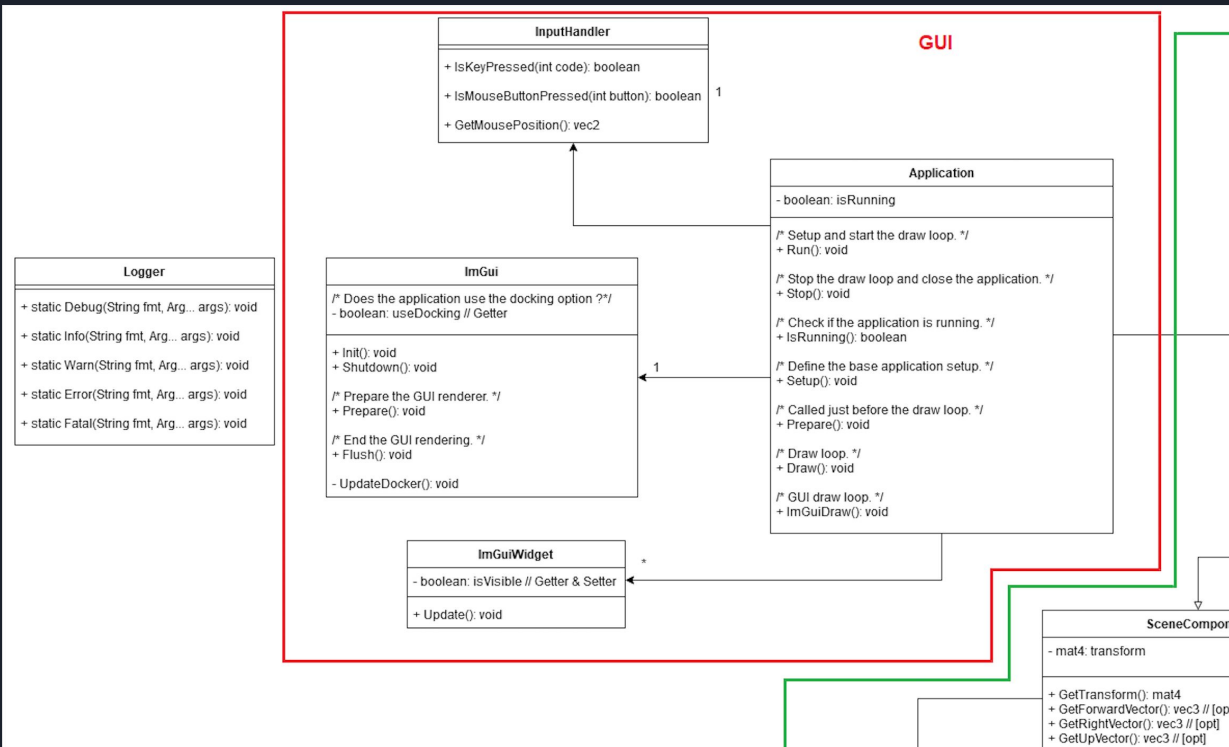
Vue d'ensemble



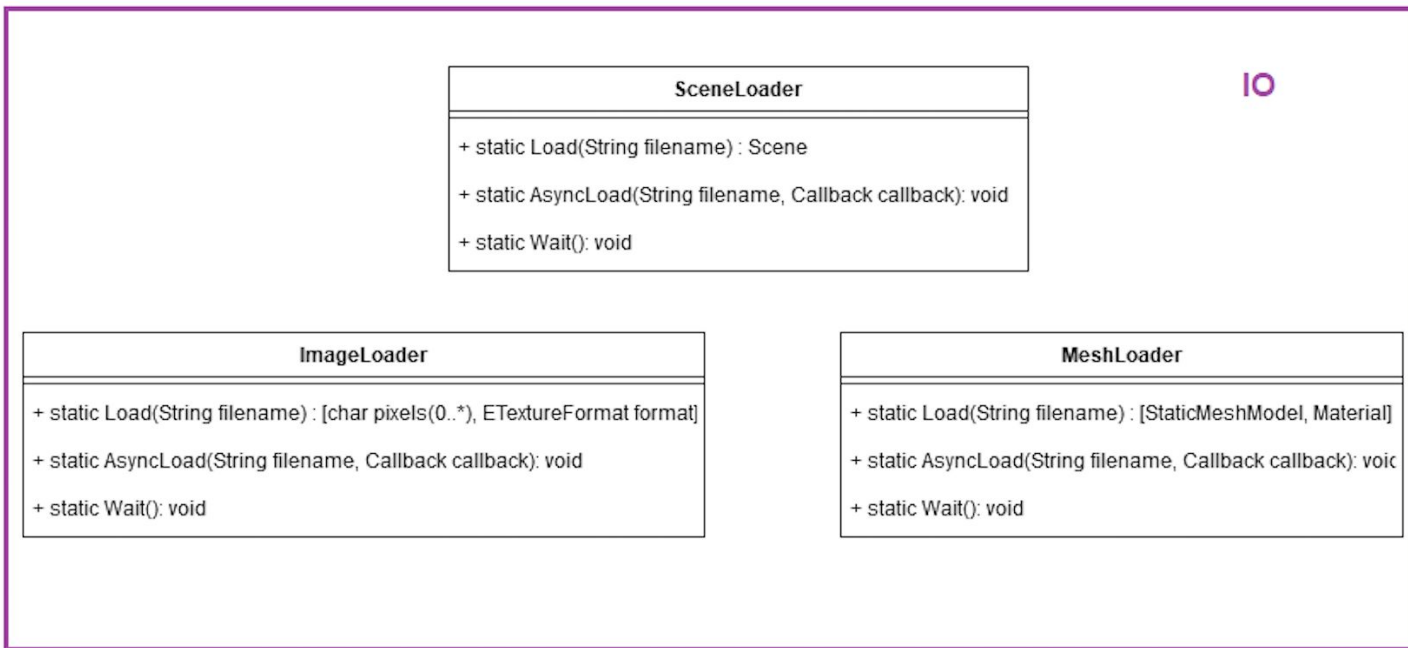
Classes et structures de données



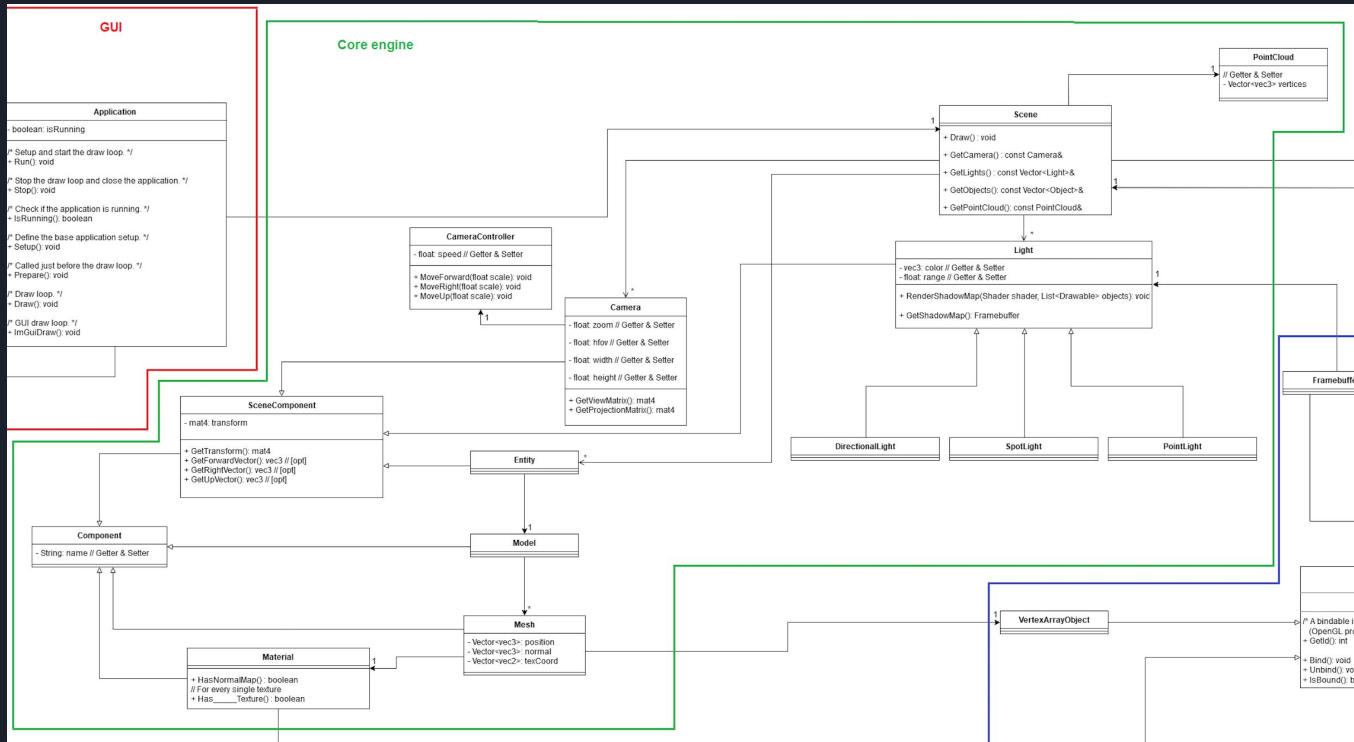
Classes et structures de données



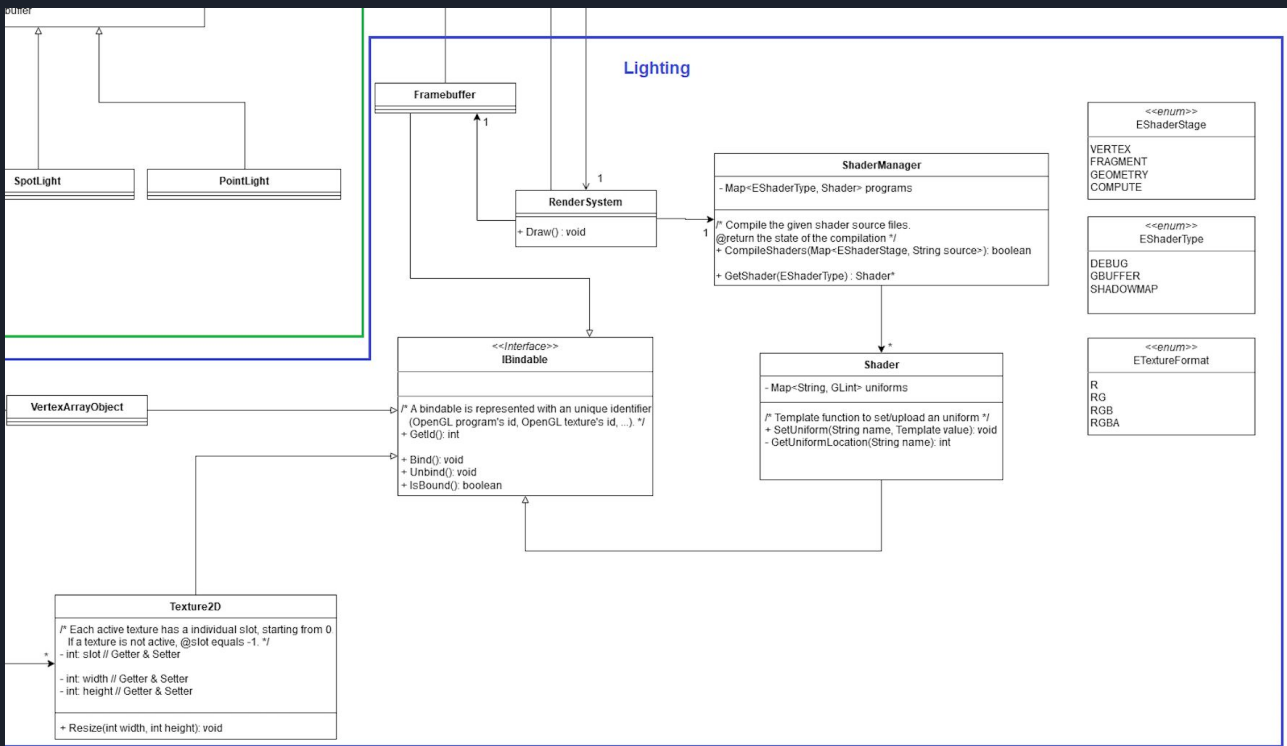
Classes et structures de données



Classes et structures de données



Classes et structures de données





Tests unitaires

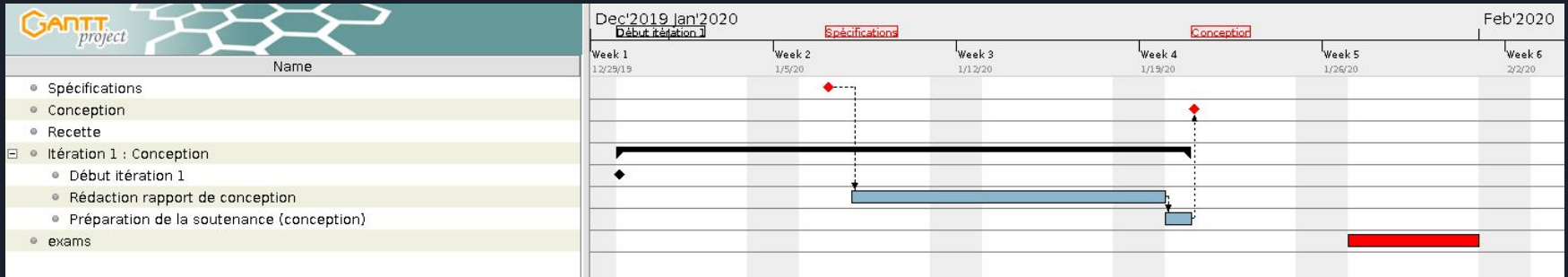
Exemples de tests unitaires à mettre en place :

- Communication CPU-GPU
- Algorithmes non triviaux
- Structure de données

Révision du planning

Équipe

Itération 1 :

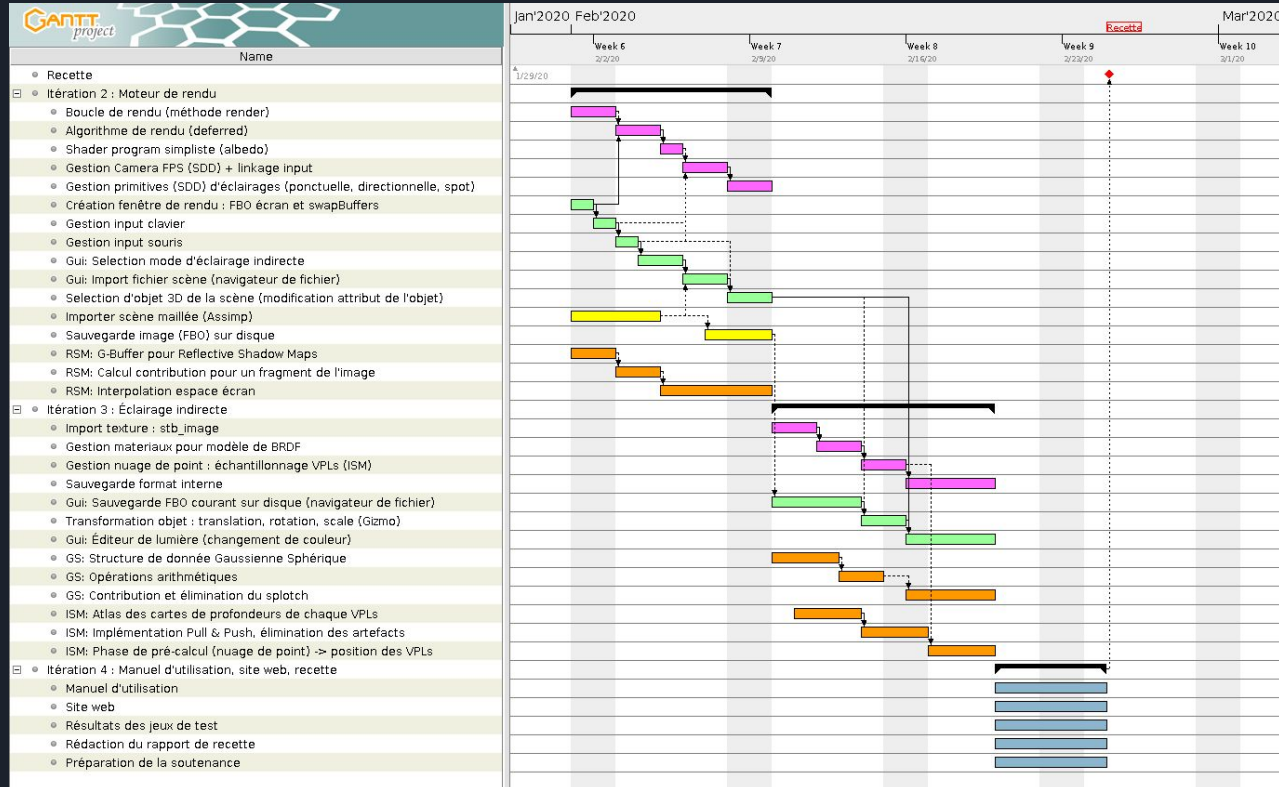


Module	Core Engine	Gui	IO	Lighting
Couleur				

Révision du planning

Équipe


Itération 2-3-4 :



Initialisation du Trello

The screenshot shows a Trello board named "GIR" with a background image of a forest. The board is organized into six columns, each representing a different stage or category of the project. Each column contains a list of tasks, each with a progress bar and a due date.

- Core Engine**
 - Boucle de rendu (méthode render) (Feb 2)
 - Algorithme de rendu (deferred) (Feb 4)
 - Shader program simpliste (albedo) (Feb 6)
 - Gestion Camera FPS + linkage input (Feb 7)
 - Gestion primitives d'éclairages (Feb 9)
 - Import texture (stb_image) (Feb 11)
 - Gestion matériaux pour modèle de BRDF (Feb 13)
 - Gestion nuage de point pour ISM (Feb 15)
 - Sauvegarde format interne (Feb 19)
- Gui**
 - Création Fenêtre de rendu (FBO et swapBuffers) (Feb 1)
 - Gestion input clavier (Feb 2)
 - Gestion input souris (Feb 3)
 - Selection mode d'éclairage indirecte (Feb 5)
 - Import fichier scène (navigateur de fichier) (Feb 7)
 - Selection d'objet 3D de la scène (Feb 9)
 - Sauvegarde FBO courant sur disque (navigateur de fichier) (Feb 14)
 - Transformation object : translation, rotation, scale (Feb 16)
 - Éditeur de lumière (Feb 19)
- IO**
 - Importe scène maillée (Assimp) (Feb 5)
 - Sauvegarde image rendu sur disque (Feb 9)
- Lighting**
 - RSM: G-Buffer RSM (Feb 2)
 - RSM: Calcul contribution pour fragment de l'image (Feb 5)
 - RSM: Interpolation espace écran (Feb 9)
 - GS: Structure de donnée gaussienne sphérique (Feb 12)
 - ISM: Atlas des cartes de profondeurs de chaque VPLS (Feb 12)
 - GS: Opérations arithmétiques (Feb 15)
 - ISM: Implémentation Pull & Push, élimination des artefacts (Feb 15)
 - GS: Contribution et élimination du splotch (Feb 19)
 - ISM: Phase de pré-calcul (nuage de point) notation des VPLS (Feb 19)
- Recette et livrables**
 - Manuel d'utilisation (Feb 24)
 - Site web (Feb 24)
 - Résultats des jeux de test (Feb 24)
 - Rédaction du rapport de recette (Feb 24)
 - Préparation de la soutenance (Feb 24)
- Done**
 - Dossier de conception (Jan 20)
 - Préparation soutenance (conception) (Jan 20)

Révision des risques

Risques impactants



Risque	Probabilité d'apparition	Prévention	Solution
Indisponibilité/abandon d'un membre de l'équipe au cours du projet		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		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		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.

Révision des risques

Risques impactants



Risque	Probabilité d'apparition	Prévention	Solution
Performance décevantes		Utiliser des techniques d'optimisations.	Se référer aux implémentations existantes et utiliser des outils de mesures de performances et de "profiling".
Compatibilité multi OS		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		Effectuer une planification temporelle assez large.	Abandonner des fonctionnalités optionnelles.
Manque de communication dans le groupe		Utiliser un bon outil de communication dès le départ.	Utiliser un autre logiciel de communication.

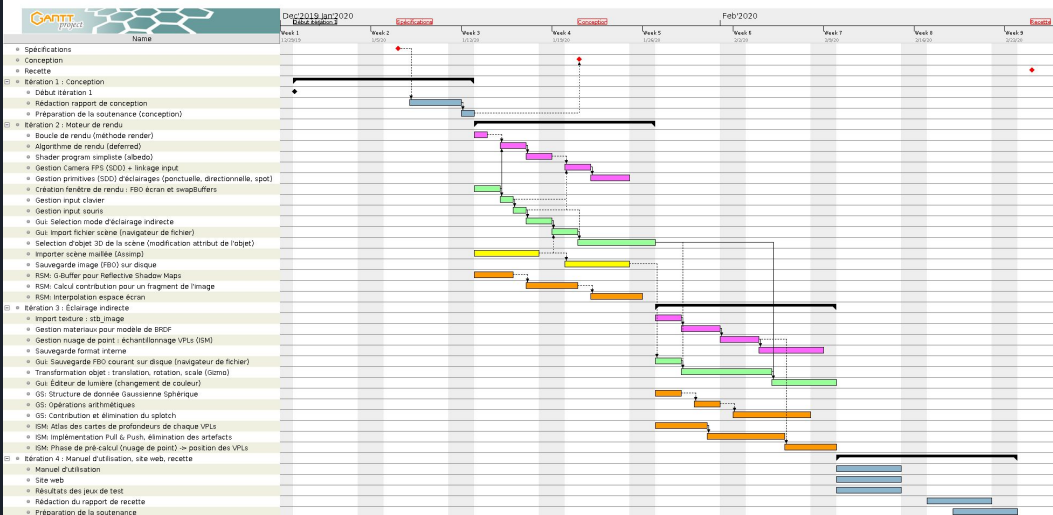


Merci de votre attention.

Des questions ?

Annexe

Planning initial : (spécification)



Planning revu : (conception)

