

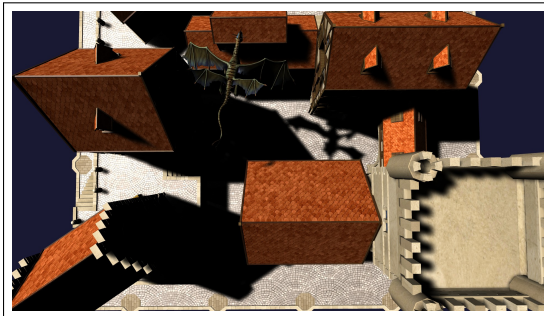
# CHEF-D'ŒUVRE

## Moment Based Rendering

---

# Spécifications

---



Baptiste Delos, Mehdi Djemai, Alban Odot,  
Pierre Mézières et Jean-Baptiste Sarazin

Encadrant : Mathias Paulin

10 janvier 2019

# 1 Introduction

L'objectif de ce projet est d'implanter des méthodes de calcul d'ombre et de transparence basées sur les moments des profondeurs et des indices de transmittance des fragments. L'objectif de ce projet sera de comparer ces techniques avec d'autres méthodes de calcul d'ombrage et de transparence. Les comparaisons seront réalisées selon deux critères : la qualité visuelle et la performance. Le premier, étant subjectif, ne demandera pas de mise en place d'outils ou de module dédié et sera donc orienté utilisateur. Le second, plus quantitatif, sera à l'origine d'un module de mesure de temps et d'analyse de données. Ce projet est donc structuré sur deux axes : l'implantation de méthodes de rendu d'une part et les mesures et comparaisons de celles-ci d'autres part.

Les calculs d'ombres se feront sur la base de l'algorithme du *Shadow Mapping*. La version standard de cette méthode ayant comme principal défaut de produire des ombres crénelées, il existe de nombreuses alternatives pour filtrer ces dernières. Pour cela, nous implanterons trois variantes permettant le filtrage des cartes d'ombres : *Percentage-Closer Filtering* (PCF)[RSC87] qui représentera notre méthode de référence ; *Variance Shadow Mapping* (VSM) [DL06] et *Four Moment Shadow Mapping* (4MSM) (avec ses deux variantes *Hamburger* et *Hausdorff*). Quant au calcul de la transparence indépendant de l'ordre, bien qu'il soit possible d'obtenir de très bons résultats de manière naïve, la difficulté est d'implanter une méthode suffisamment efficace pour des applications temps réel. Nous implanterons la méthode de *Depth Peeling* [NVI01] pour laquelle nous considérerons les images générées comme vérité terrain, les méthodes *Weighted Blended OIT* (WBOIT) [MB13], ainsi que *Moment Based Order Independent Transparency* (MBOIT)[MKKP18] qui se base sur les calculs des moments du 4MSM. Nous détaillerons la spécification de ces méthodes ainsi que des modules de mesures et comparaison.

Ce rapport de spécification définit le cahier des charges, fournit une vue générale du projet accompagnée d'une présentation des différents modules pour finir sur le planning prévisionnel envisagé et la gestion des risques pouvant impacter le déroulement du projet. Celui-ci ne s'étendra pas sur la base de code existant, qui évoluera pour accueillir l'ensemble des modules et faciliter les tests.

## 2 Cahier des charges

### 2.1 Exigences fonctionnelle

#### 2.1.1 Gestion de l'ombrage

Le système devra fournir une implantation des trois méthodes d'ombrage citées plus tôt et la possibilité de passer de l'une à l'autre au cours de l'exécution. Une scène référence illustrant les problèmes des différentes méthodes sera en plus disponible.

**Percentage-Closer Filtering** Priorité : **Haute**.

Hyperparamètres	Type	Information
Rayon du disque	int	Rayon du disque de Poisson pour la comparaison
Diffusion	float	Coefficient de diffusion de l'ombre

**Variance Shadow Mapping** Priorité : **Basse**.

Hyperparamètres	Type	Information
Taille de la Gaussienne	int	Largeur du noyau Gaussien
Variance minimale	float	Permet de clamber la variance

### Four Moment Shadow Mapping Priorité : Moyenne.

Hyper paramètres	Type	Information
Biais du moment	float	Réduit l'erreur de quantification
Biais de profondeur	float	Évite l'auto-ombrage

### 2.1.2 Gestion de la transparence

De la même manière, le logiciel devra fournir une implantation des trois méthodes de transparence citées plus tôt et la possibilité d'alterner leur utilisation au cours de l'exécution. Une scène référence illustrant les problèmes des différentes méthodes sera également disponible.

### Depth Peeling Priorité : Haute.

Hyperparamètres	Type	Information
Nombre de couches	int	Nombre maximal de couleurs mélangées

### Weighted Blended OIT Priorité : Basse.

Hyperparamètres	paramètres	Type	Information
Fonction de poids		-	Fonction de répartition des poids

### Moment Based Order Independent Transparency Priorité : Moyenne.

Hyperparamètres	paramètres	Type	Information
Biais du moment		float	Réduit l'erreur de quantification
Biais du vecteur de moments		vec4	Réduit l'erreur de quantification

### 2.1.3 Capture et mesures

Le système devra contenir un outil permettant de faire une capture de la *frame* courante affichée dans la fenêtre de rendu du moteur. Il sera aussi possible de générer la *frame* courante en combinant les différentes méthodes engendrant la production de 9 images. Cela permettra une manipulation d'images de taille et de format standardisés pour toutes les mesures. Le système devra contenir un outil de mesure du temps de calcul. Afin d'obtenir des mesures précises, chaque unité de calcul sera mesurée pour chaque méthode. Enfin, les outils de mesure d'erreur entre images implantés dans le système permettront d'afficher et de calculer l'erreur commise.

### 2.1.4 Tableau récapitulatif

Item	Priorité	Nombre d'hyperparamètres
<b>Gestion de l'ombrage</b>	<b>Haute</b>	
Percentage-Closer Filtering	Haute	2
Variance Shadow Mapping	Basse	2
Four Moment Shadow Mapping	Moyenne	2
<b>Gestion de la transparence</b>	<b>Haute</b>	
Depth Peeling	Haute	1
Weighted Blended OIT	Basse	1
Moment Based OIT	Moyenne	2
<b>Capture et mesures</b>	<b>Basse</b>	
Capture d'images	Basse	0
Chronomètre	Basse	0

### 3 Vue générale du système

Le système permettra de tester interactivement les différentes méthodes développées et d'effectuer les mesures simplement.

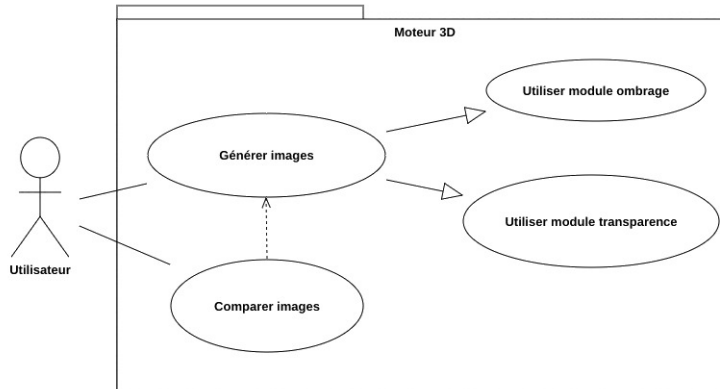


FIGURE 1 – Diagramme de cas d'utilisation général

Le système se divise donc en 2 cas principaux d'utilisation.

— **Génération d'images :**

Le cas d'utilisation de la génération d'images consiste dans le fait que l'utilisateur peut modifier les paramètres des différentes méthodes de rendu. On parle ici des paramètres pour le module d'ombrage et le module de transparence.

— **Comparaison d'images :**

Le cas d'utilisation de la comparaison d'images consiste dans le fait que l'utilisateur peut venir appliquer différentes méthodes de comparaison d'images sur les images générées via le cas d'utilisation de la génération d'images.

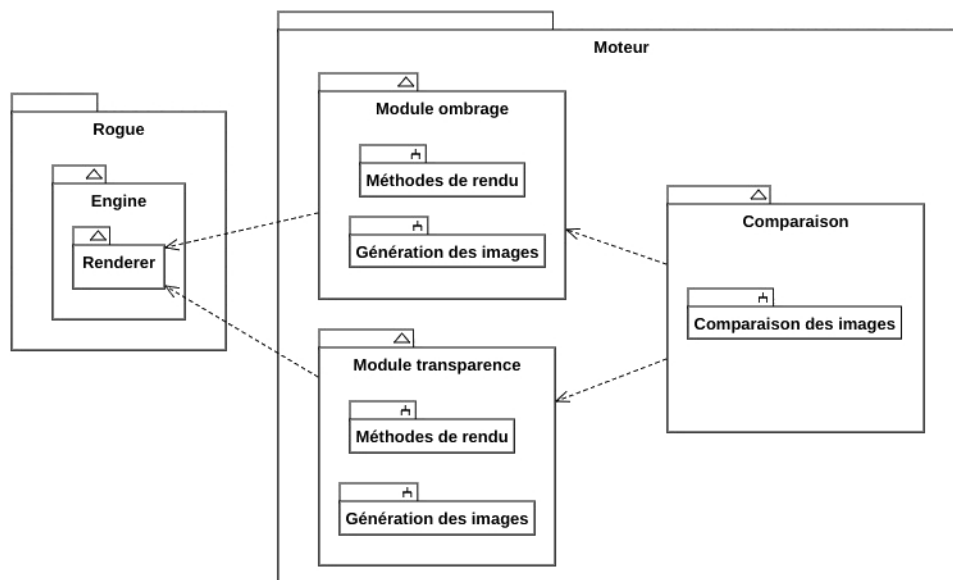


FIGURE 2 – Diagramme de Paquetages

Le projet s'articulera autour de trois modules principaux :

- Module d'ombrage
- Module de transparence
- Module de mesures et comparaisons

Les fonctionnalités de ces différents modules sont explicitées dans la section suivante.

## 4 Modules du système et tests de validation

Cette partie présente les fonctionnalités des différents modules du système<sup>1</sup>, ainsi que les tests de validation associés.

### 4.1 Module d'ombrage

#### 4.1.1 Fonctionnalités

Dans ce module, on vient s'intéresser à toute la partie du calcul d'ombrage (*Shadow Mapping*). Ce module regroupe donc tous les calculs des cartes de profondeur pour les différentes méthodes mises en place, c'est-à-dire la méthode de référence *PCF*, le calcul des moments d'ordre deux avec *VSM* ainsi que les moments d'ordre deux à quatre pour le *4MSM*.

Le module permettra donc à l'utilisateur de choisir la méthode de rendu utilisée pour le calcul de l'ombrage. De facto, l'utilisateur pourra modifier les différents paramètres de la méthode sélectionnée (Les paramètres sont explicités dans le cahier des charges - section 2). Le module permettra aussi de venir sauvegarder la dernière image générée par le moteur, ce de deux manières différentes. La première manière est une sauvegarde classique de l'image au format *png*. La seconde est une sauvegarde de l'image dans une mémoire tampon. De cette manière, l'image sera utilisable dans le module de comparaison (Section 4.3).

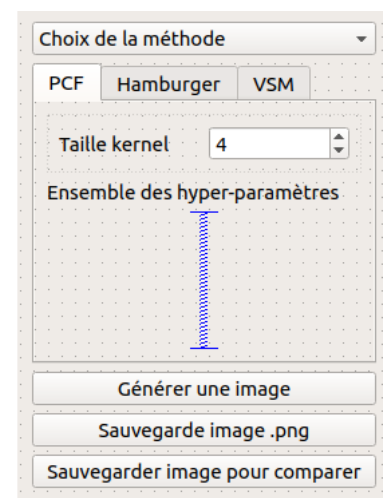


FIGURE 3 – Recherche pour l'interface du module d'ombrage

#### 4.1.2 Tests de validation

Pour valider le fonctionnement de ce module, il s'agit de vérifier si le calcul d'ombrage est visuellement correct, et de vérifier que le résultat obtenu est celui attendu par la méthode de rendu utilisée. On tentera donc de générer des images caractérisées par différentes problématiques liées au calcul d'ombrage. Seront exploitées notamment les scènes présentées dans l'article sur le calcul d'ombres basé sur les moments [PK15], afin de valider, visuellement parlant, la similitude des images générées avec les résultats présentés dans cet article.

1. Les interfaces proposées ici permettent de donner un aperçu des paramètres utilisables, elles ne sont en aucun cas définitives.

## 4.2 Module de Transparence

### 4.2.1 Fonctionnalités

Dans ce module, on s'intéresse à toute la partie du calcul de la transparence indépendante de l'ordre des fragments (*OIT*). Ce module regroupe donc la méthode du *Depth Peeling*, le calcul du *Weighted-Blended OIT* ainsi que le calcul du *MBOIT*.

Les fonctionnalités de ce module coïncident avec celles présentées pour le module d'ombrage, adaptées cette fois-ci au calcul de la transparence. L'utilisateur peut choisir la méthode de rendu de la transparence, et modifier les différents paramètres de la méthode sélectionnée. L'utilisateur pourra également sauvegarder les images directement à partir de ce module.

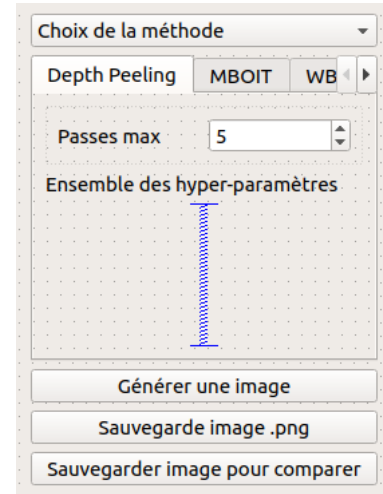


FIGURE 4 – Recherche pour l'interface du module de transparence

### 4.2.2 Tests de validation

De la même manière que pour le module d'ombrage, il s'agit de vérifier si les images générées correspondent aux résultats attendus par la méthode utilisée. Dans un même objectif, on essaiera de générer des images réputées problématiques pour le calcul de la transparence, sur les exemples des scènes de l'article abordant la transparence avec indépendance de l'ordre basée sur les moments [MKKP18].

## 4.3 Module de mesures et comparaisons

### 4.3.1 Fonctionnalités

Dans ce module, on vient s'intéresser à toute la partie correspondant à la comparaison des méthodes. Il y a deux points principaux de comparaisons : Un critère visuel et un critère de performances déterminés respectivement par comparaison d'images au moyen de différents opérateurs de comparaison, et par comparaison du temps de calcul de ces images. Les opérateurs de comparaisons seront typiquement du calcul d'écart de couleurs dans différents espaces de couleur (RGB,  $L^*a^*b^*$  ...). Un opérateur de différences d'images sera aussi implémenté. Pour la comparaison des performances, un outil de mesure du temps de génération des images sera implémenté. Mesure de performance en OpenGL. Cet outil sera facilement attachable à des *renderer*. Pour faciliter la comparaison entre les différentes méthodes de rendu, il pourra aussi disposer d'une méthode pour déclencher la génération d'une seule image. De cette manière, on pourra changer la méthode de rendu en n'ayant à changer aucun autre paramètre de la scène.

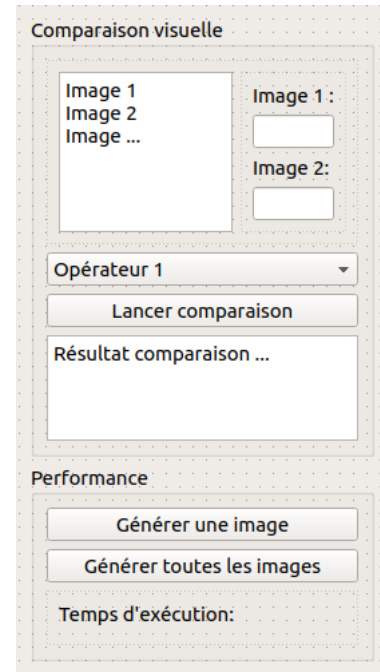


FIGURE 5 – Recherche pour l'interface du module de comparaison

Lors de la sauvegarde des images dans la mémoire tampon, on gardera aussi en mémoire le temps de génération de l'image par OpenGL. Pour simplifier la gestion des images, le nom des images respectera la convention de nommage suivante :

**Nom\_module - Nom\_méthode\_utilisée - Heure\_acquisition**

Le fichier **OIT-DepthPeeling-15-30-24** correspond par exemple à une image générée à 15h30 et 24 secondes dans le module de transparence via la méthode de *Depth Peeling*. Pour la sauvegarde en fichier image classique, on ajoutera le jour d'acquisition, de telle manière que si le même fichier a été généré le 12 Janvier 2019, le nom du fichier sera **OIT-DepthPeeling-12-01-19-15-30-30.png**

### 4.3.2 Tests de validation

Pour ce module, il faut valider principalement le fonctionnement des opérateurs de comparaison, ce qui implique au préalable la génération des images par différentes méthodes de rendu. On pourra ensuite tester les différents opérateurs et vérifier ainsi la cohérence des mesures obtenues par rapport aux images générées.

# 5 Planning prévisionnel et analyse des risques

## 5.1 Planning prévisionnel

### Processus itératif

La planification du projet suit un modèle de développement itératif. La durée de chaque itération s'élève à deux semaines. Pour chacune de ces périodes a donc été élaboré un ensemble de tâches réparties sur les membres du groupe. Cette planification des tâches comporte une phase d'estimation, pour chacune, de la charge de travail qu'elle représente suivant la méthode d'évaluation du *Planning Poker*, ainsi que des livrables à produire avant sa terminaison (rapports, revues, code, etc.). Le niveau de difficulté attribué pourra par ailleurs tenir compte des éventuels renseignements et consignes de développement apportés par le client.

La fin d'une itération donne lieu à une restitution des livrables au client, et au regroupement des membres de l'équipe sur les tâches effectivement réalisées. En outre, deux heures par semaine sur les créneaux réservés pour l'avancement du projet sont consacrées à un bilan sur l'avancement de chacun des membres dans ses tâches.

### Planification des tâches

Le déroulement de ce projet est représenté plus précisément au travers d'un diagramme de Gantt (disponible en annexe 6.1). Le format de ce diagramme permet de représenter le découpage du projet en itération, la liste des tâches envisagées dans chacune et la répartition de ces tâches sur les membres de l'équipe.

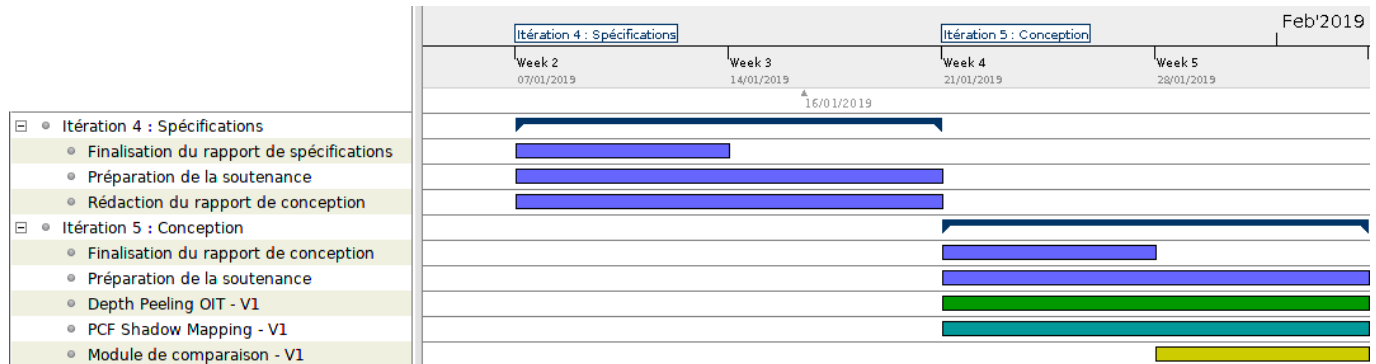
## 5.2 Analyse et gestion des risques

Une part de la gestion du projet est dédiée à la définition des risques susceptibles d'altérer le processus de développement. Il s'agit par conséquent d'estimer les chances de réalisation de chacun de ces risques, leur impact dans la gestion du projet comme dans le développement et proposer des actions de prévention et de résolution pour chacun. La matrice des risques donnée en annexe (Section 6.2) permet d'explorer de ces risques de manière synthétique.

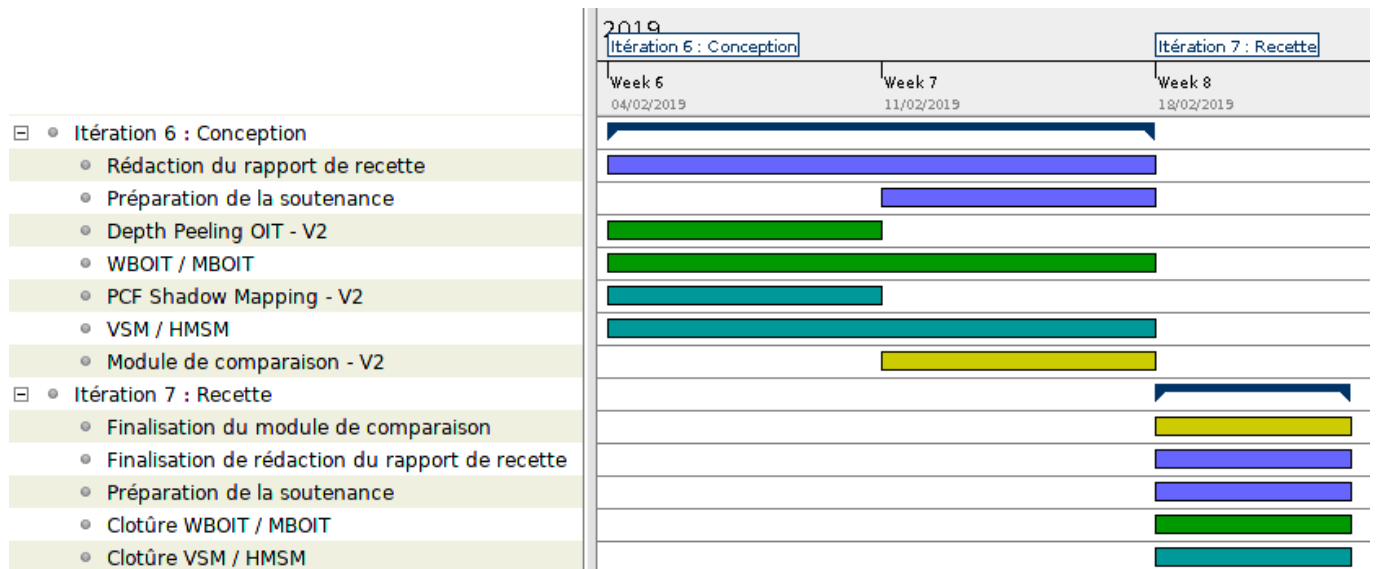


# 6 Annexes

## 6.1 Diagramme de Gantt



(a) Itérations 4 et 5



(b) Itérations 6 et 7

FIGURE 6 – Planning prévisionnel évalué sur les quatre prochaines itérations

## 6.2 Matrice des risques

Risque	Probabilité apparition	Impact	Prévention	Solution
Indisponibilité d'un membre de l'équipe en cours de projet	20%	moyen	Faire en sorte que tous les membres du projet puissent reprendre le travail d'un autre	Répartir la charge de travail efficacement sur les autres membres du groupe
Recommencer le projet en cas de mauvais choix de moteur graphique à utiliser comme base	20%	fort	Vérifier que les fonctionnalités nécessaires de base sont parfaitement fonctionnelles	Développer de manière à ce que le code soient facilement portable sur un autre moteur (éviter la duplication de code, réduire l'utilisation de fonctions intrinsèques au moteur)
Recommencer le projet en cas d'une mauvaise conception de départ	20%	fort	-Passer beaucoup de temps sur la conception initiale et l'implantation utilisera des modèles connus pour leur flexibilité.	Repenser la conception en essayant de modifier le moins possible l'existant en utilisant des design pattern
Compatibilité MAC non fonctionnelle	20%	fort	-Commencer par rendre le projet compatible avant de commencer le développement. -S'assurer de la compatibilité directement avec la machine du client.	Contacteur un expert MAC
Copyright/Copyleft des shaders fournis sur les papiers scientifiques	10%	moyen	Bien se renseigner durant la phase de réflexion	-Réécrire les shaders utilisées -Ne sera un problème qu'en cas de mise sur le marché du produit
Incompréhension et incapacité d'implémenter une méthode présente dans les papiers	20%	moyen	Passer du temps à bien comprendre les papiers	-Demander de l'aide aux autres membres du projet. -Demander de l'aide à M.Paulin
Rendu côté graphique pas assez performant ( lag )	30%	fort	Limiter le nombre de données à envoyer à la carte graphique	-Réduire le nombre de données envoyé sur la carte graphique -Changer ou modifier le moteur si l'optimisation ne suffit pas
Mauvaise rédaction des documents à fournir/diapos	80%	faible	-Faire valider chaque partie par tous les membres du groupes -Envoyer une première version terminée pour correction par M.Paulin	Accepter la critique et ne pas reproduire les mêmes erreurs pour les documents suivants
Pérennité/perte des données	5%	fort	Données stockées à plusieurs endroit (gitlab/sauvegarde en hors ligne)	Pas de solutions. Recommencer le projet.
Dépassement des délais	80%	fort	-Estimer le temps nécessaire à la réalisation des différentes tâches -Répartir correctement les tâches	Faire des sessions de programmation intensives

# Références

- [DL06] William Donnelly and Andrew Lauritzen. Variance shadow maps. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, I3D '06, pages 161–165, New York, NY, USA, 2006. ACM.
- [MB13] Morgan McGuire and Louis Bavoil. Weighted blended order-independent transparency. *Journal of Computer Graphics Techniques (JCGT)*, 2(2) :122–141, December 2013.
- [MKKP18] Cedrick Münstermann, Stefan Krumpfen, Reinhard Klein, and Christoph Peters. Moment-based order-independent transparency. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1) :7 :1–7 :20, May 2018.
- [NVI01] Cass Everitt NVIDIA. Interactive order-independent transparency. 2001.
- [PK15] Christoph Peters and Reinhard Klein. Moment shadow mapping. In *Proceedings of the 19th Meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, i3D '15, pages 7–14, New York, NY, USA, 2015. ACM.
- [RSC87] William T. Reeves, David H. Salesin, and Robert L. Cook. Rendering antialiased shadows with depth maps. *SIGGRAPH Comput. Graph.*, 21(4) :283–291, August 1987.