RECETTE DU PBRADIUM

Matthias Roves, Hugo Rens et Jehan-Antoine Vayssade

22 février 2018

Université Paul Sabatier (Toulouse III)

Résultats

Divergences

Commentaires techniques

Rétrospective

RÉSULTATS

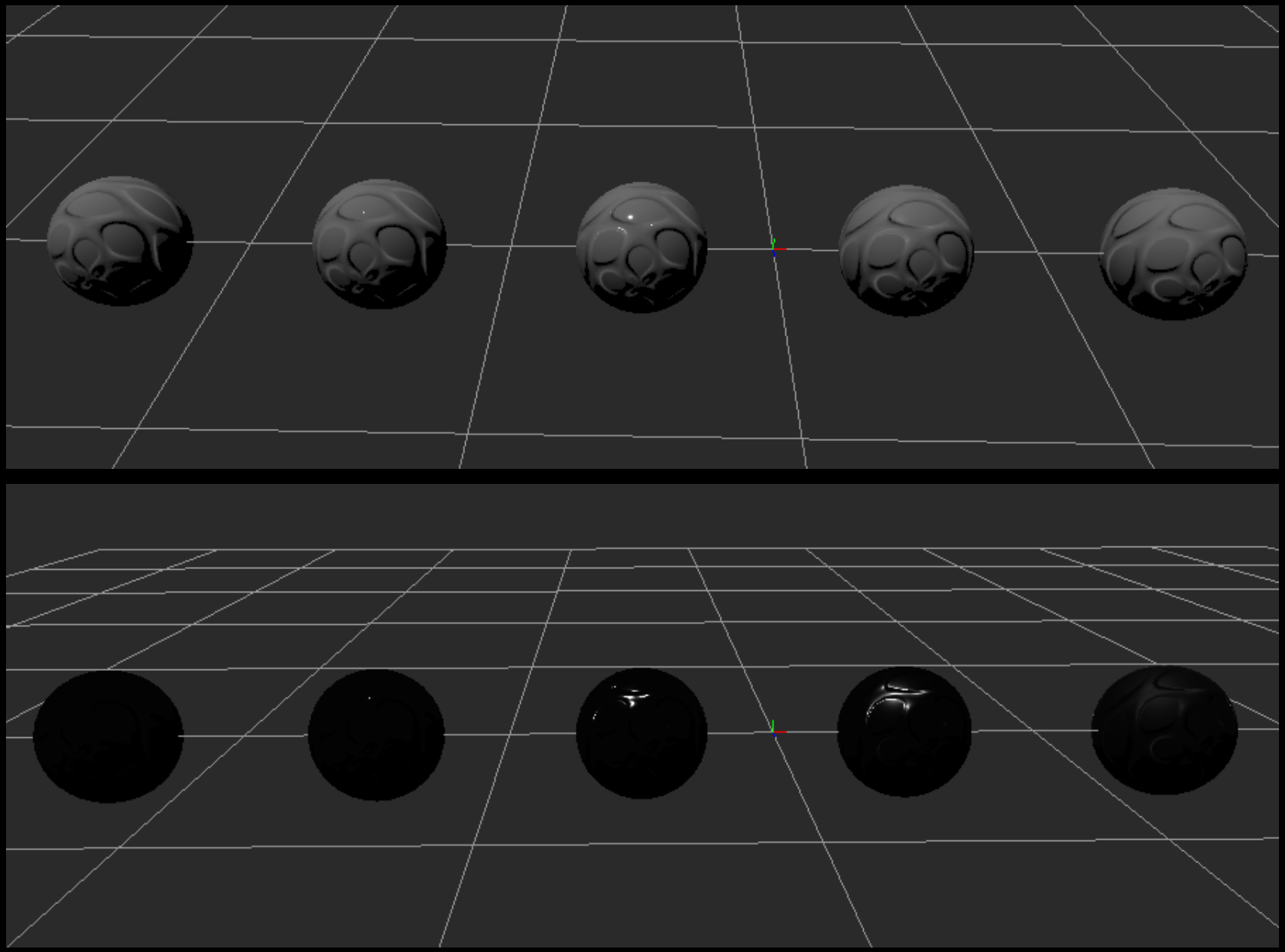


Figure 1 – Plastique (en haut) et Métal (en bas).

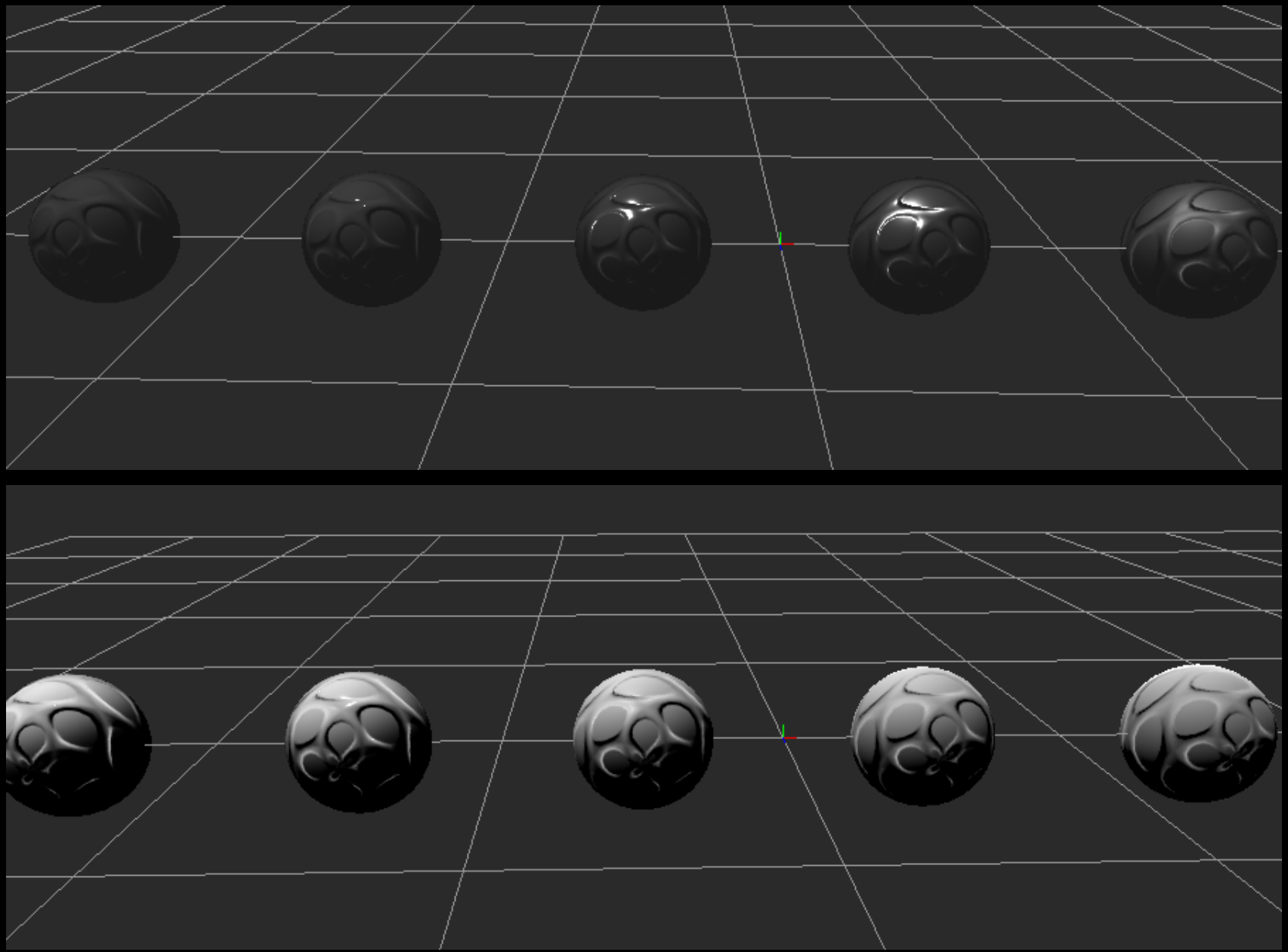


FIGURE 2 – Substrate (en haut) et Principled Disney (en bas).

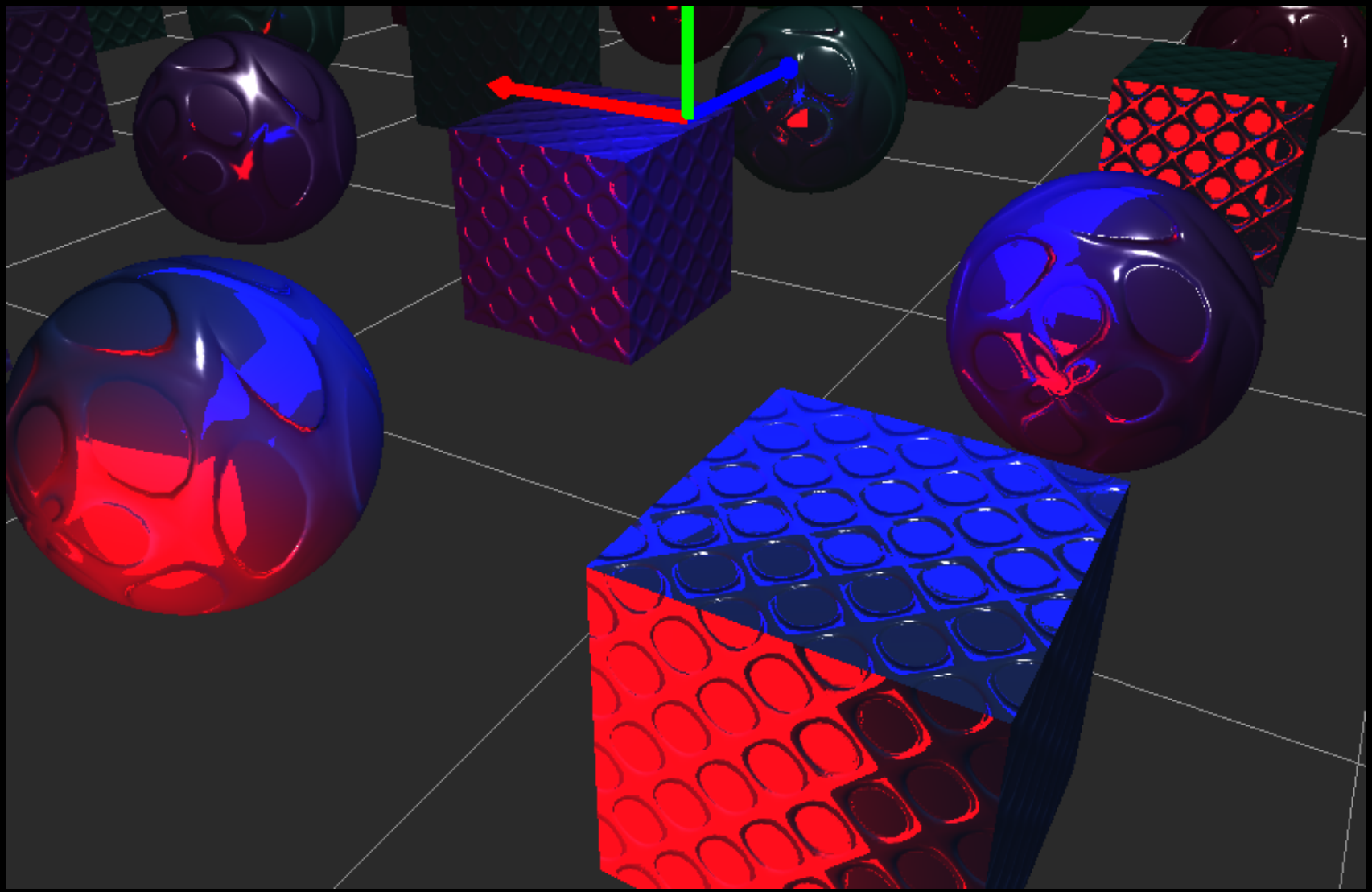


FIGURE 3 – Sources polygonales. Un carré rouge et une étoile bleue sur des matériaux de type Substrate. L'étoile bleue est transformée par le gizmo.

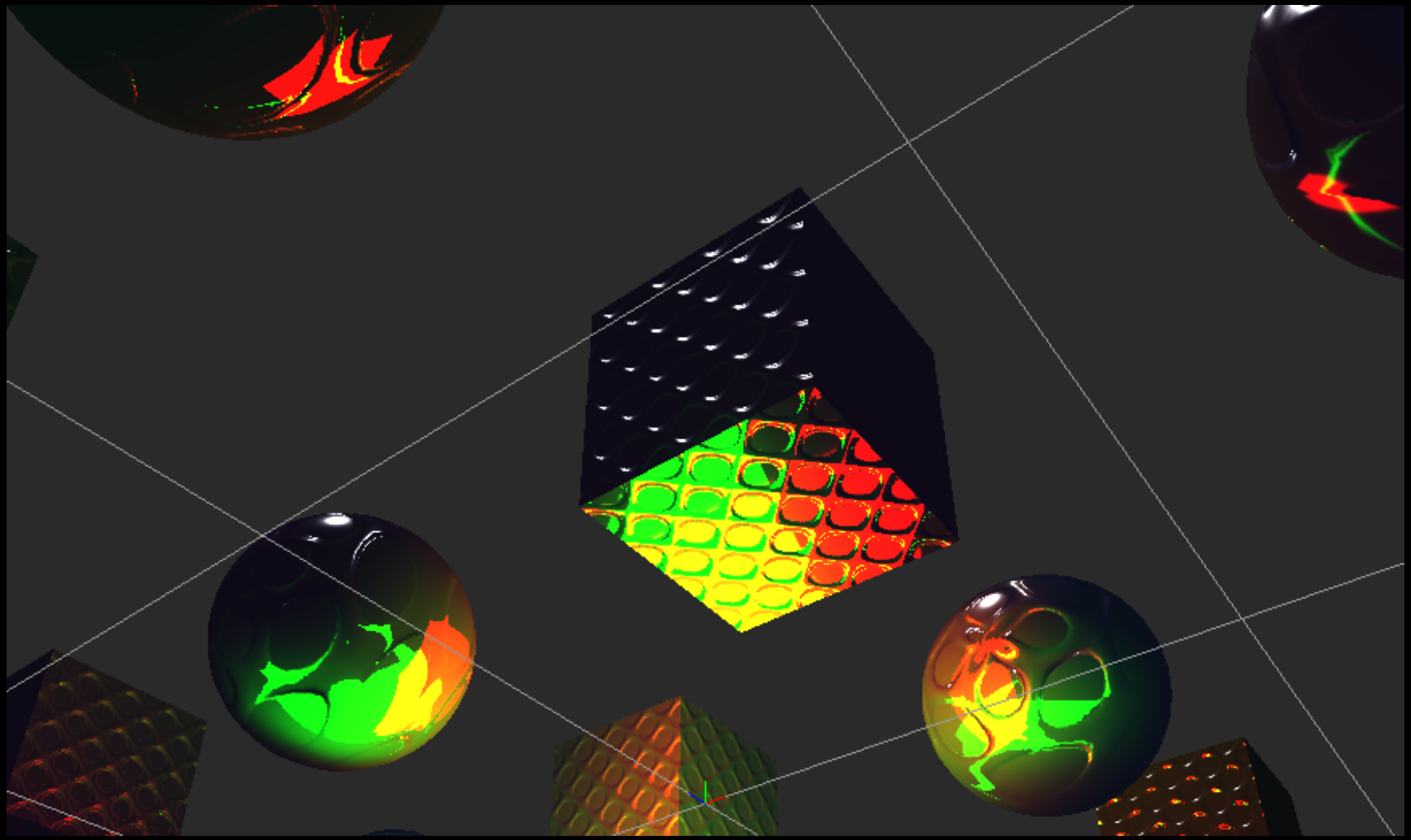


Figure 4 – Superposition de 2 polygones : effet additif différent de la réalité, mais totalement lié à la technique.

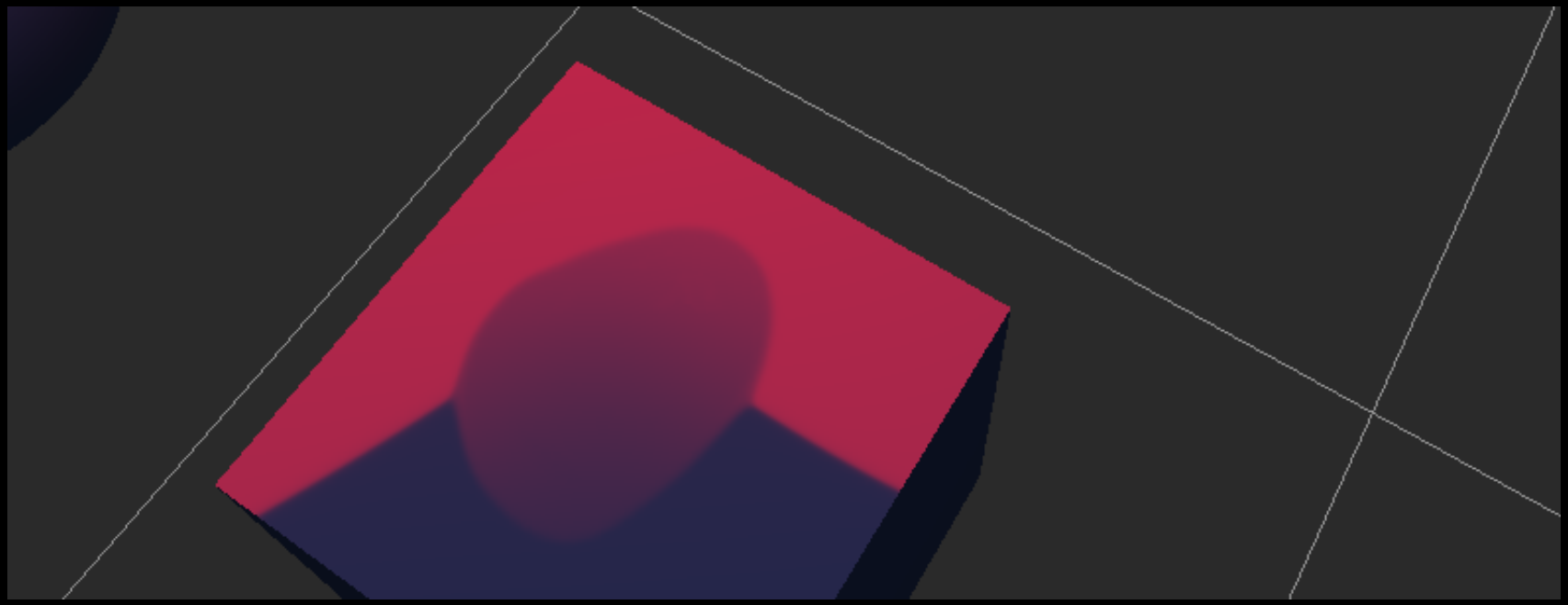


Figure 5 – Changement de rugosité du matériau éclairé par une source en forme d'étoile.

FIGURE 6 – Définition des accès dans la grille de clusters.

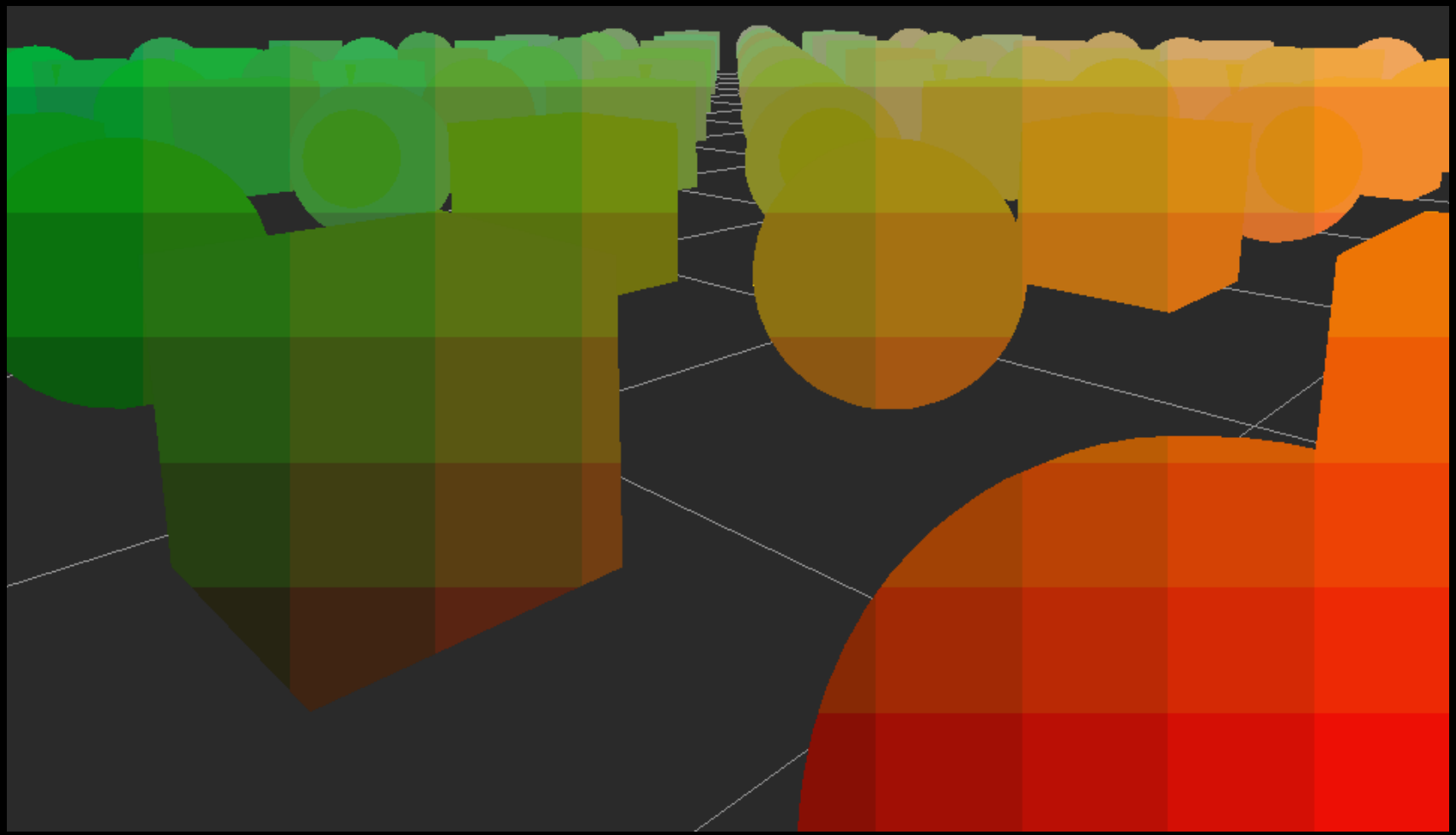


Figure 7 – Nombre de lights visible par clusters en fonction de la position dans l'espace écrans. Bleu : 0, Rouge : 100 sur cette image.

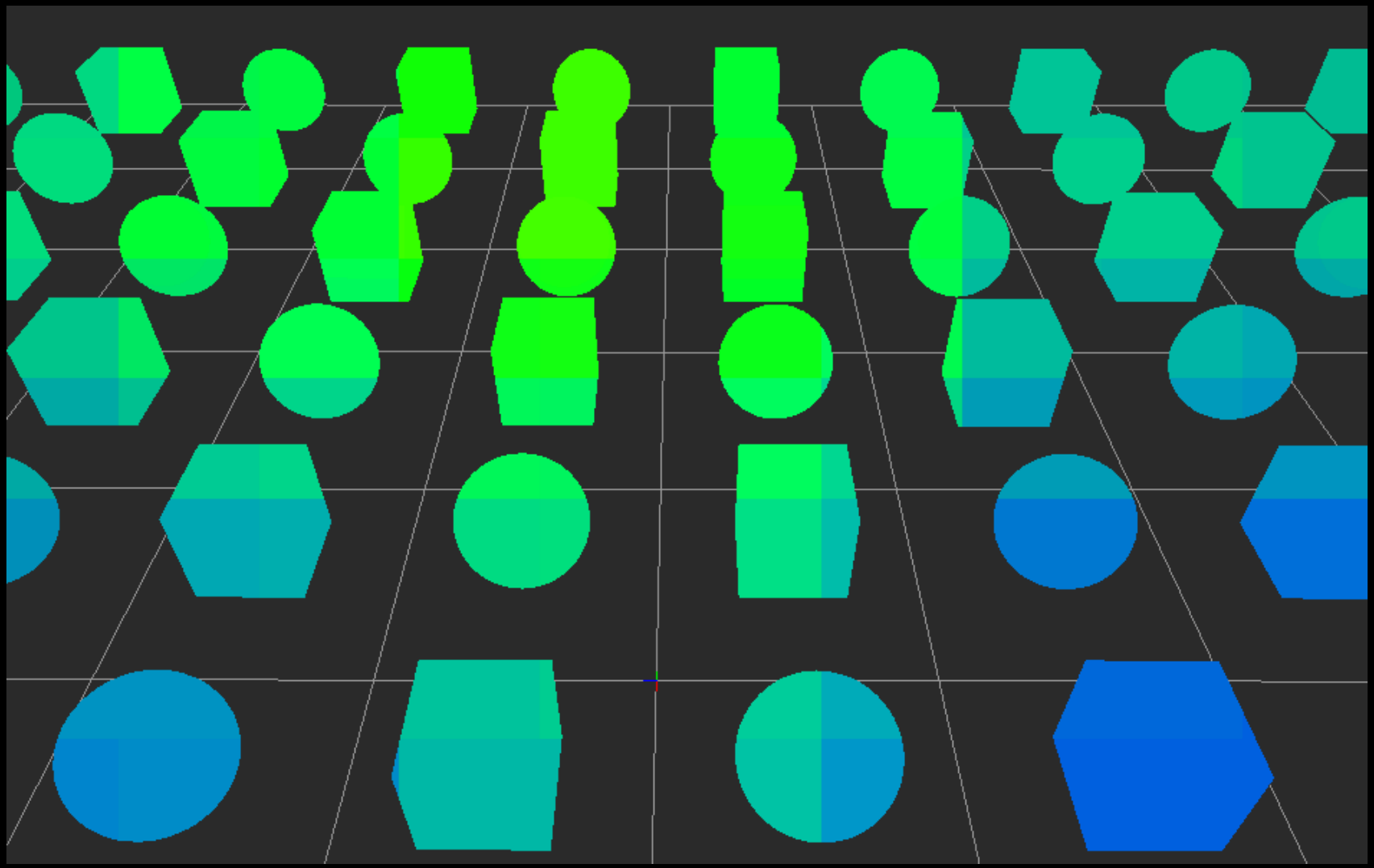


Figure 7 – Nombre de lights visible par clusters en fonction de la position dans l'espace écrans. Bleu : 0, Rouge : 100 sur cette image.

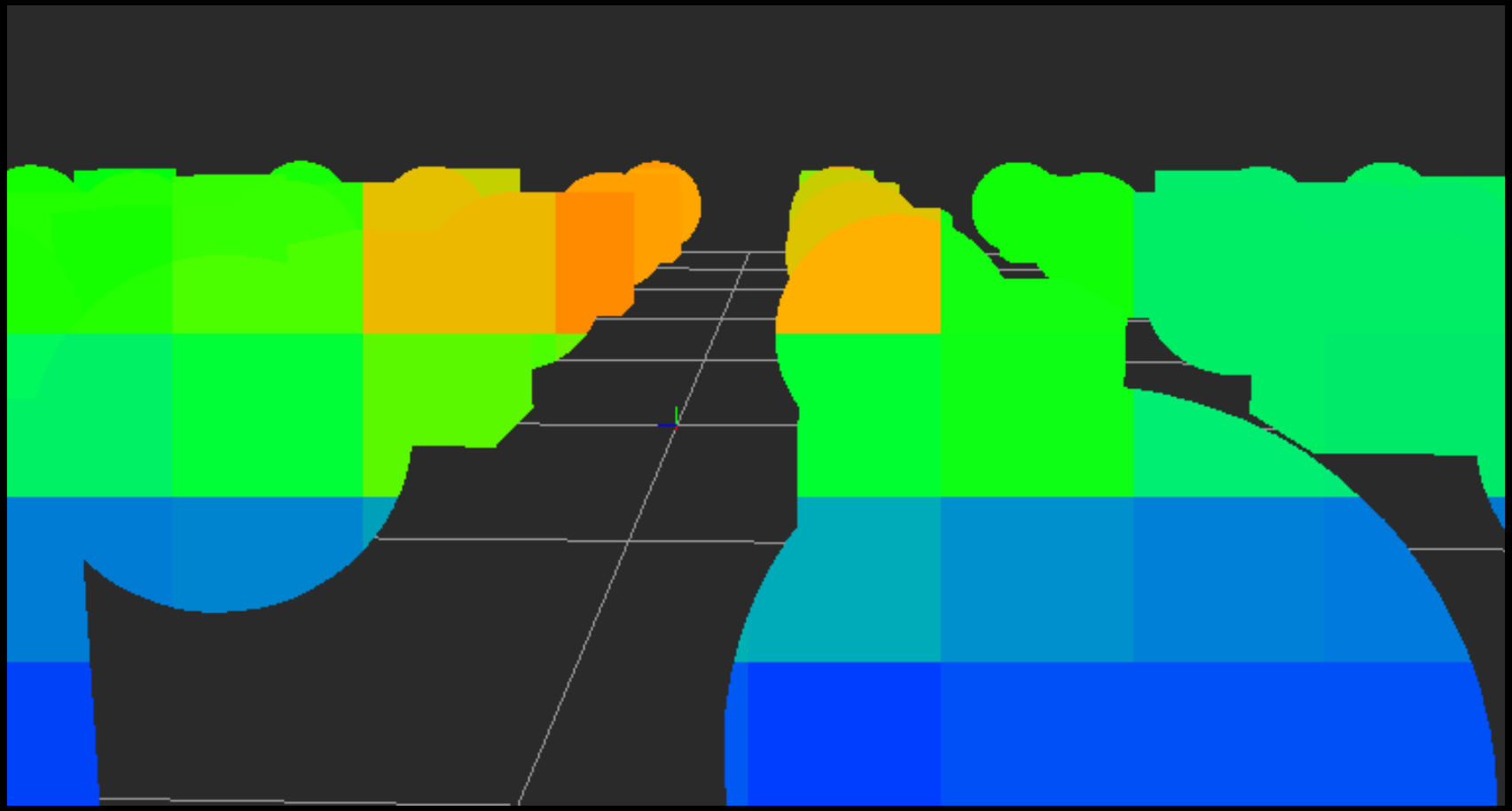
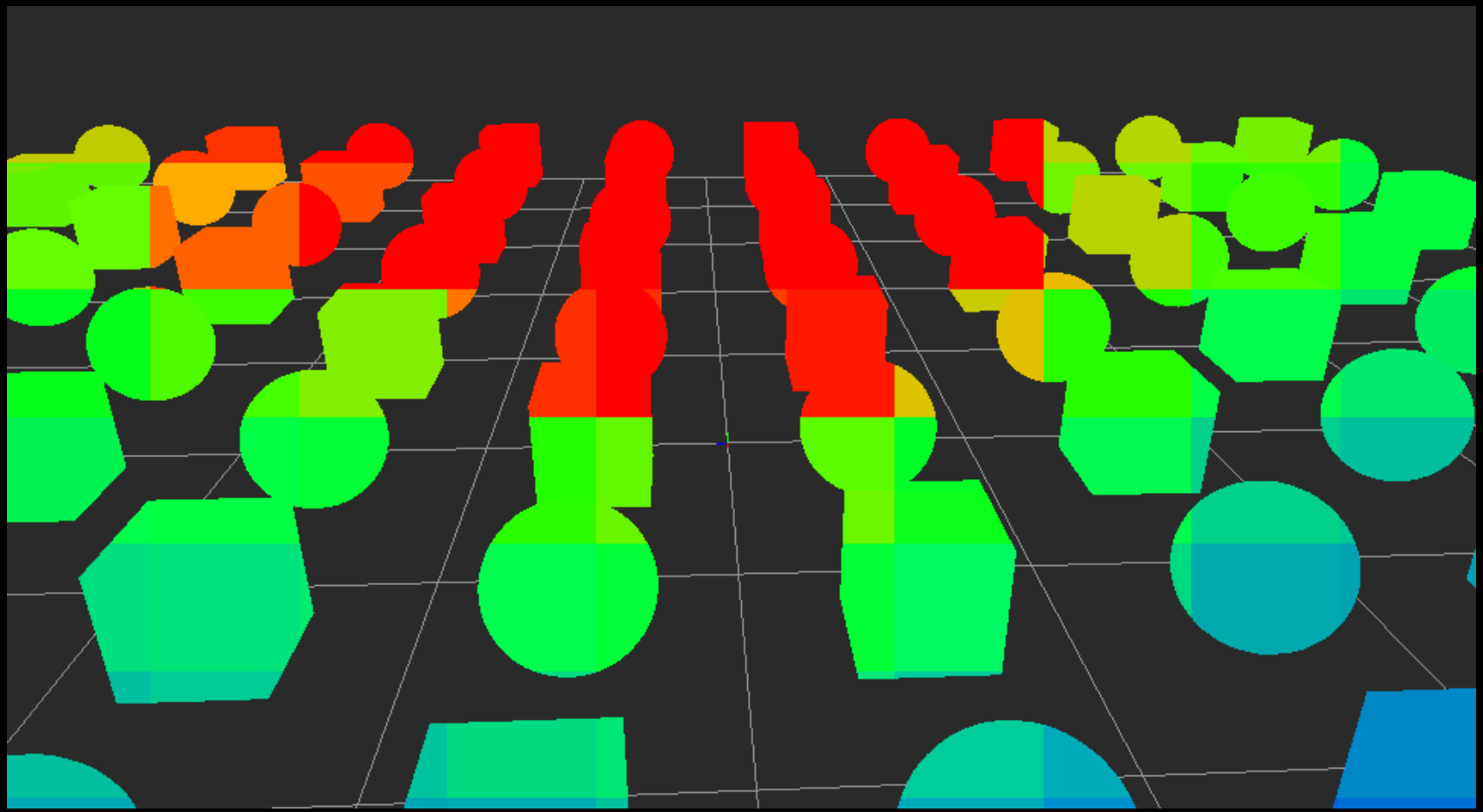


FIGURE 7 – Nombre de lights visible par clusters en fonction de la position dans l'espace écrans. Bleu : 0, Rouge : 100 sur cette image.



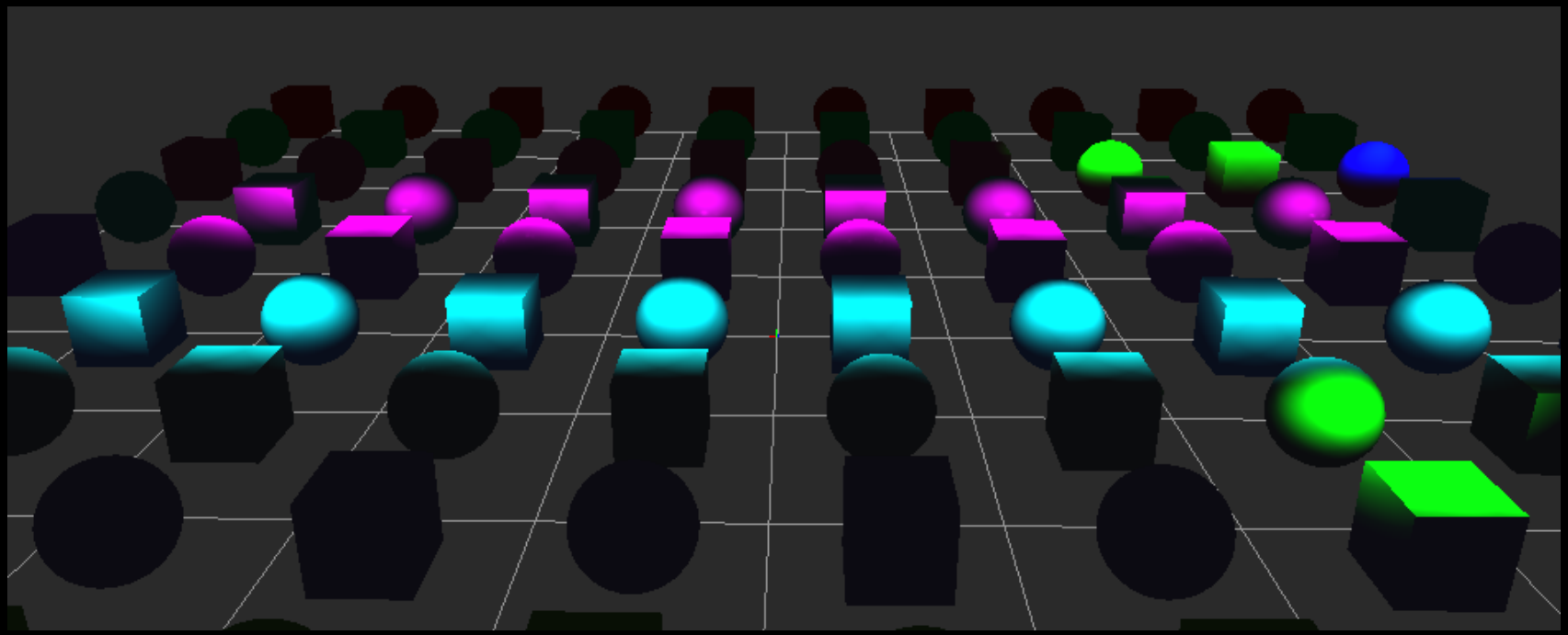


Figure 8 – Rendu avec un nombre de lumières de 100.

DIVERGENCES

- Utilisation de SSBO à la place d'une Texture Array.
  - **Cause** Manque de temps
  - **Effet** Légèrement moins performant
  - **Effet** Taille des données limitée à 16Kb
  - **Effet** Nombre de lumière(s) visible(s) limité à 1365
  - **Effet** Taille de la grille limitée à $11 \times 11 \times 11$
- Ajout d'une nouvelle structure de Light pour le stockage
  - **Cause** Limitation de la taille des données
  - **Effet** Structure minimal moins lourde (384b)
  - **Effet** Plus efficient sur les `updates`
  - **Effet** Uniquement les lumières de type PointLight

COMMENTAIRES TECHNIQUES

Modification du radium

- Les structures des lumières sont devenues des `Component`
- Création d'une classe `LightManager` pour gérer les structures de lumières
- Possibilité d'ajouter des chemins d'`#include` à une `ShaderConfiguration`
- Ajout de la version GLSL en tant que paramètre
- Possibilité de rendre les objets avec une technique différente
- Correction de bugs lors du chargement

PBR RENDERER

- Création d'un nouveau `Renderer` qui différencie l'éclairage direct et indirect à l'aide de deux `LightManager`

Mise en place d'une classe composant des shaders selon la lumière et le type de matériau :

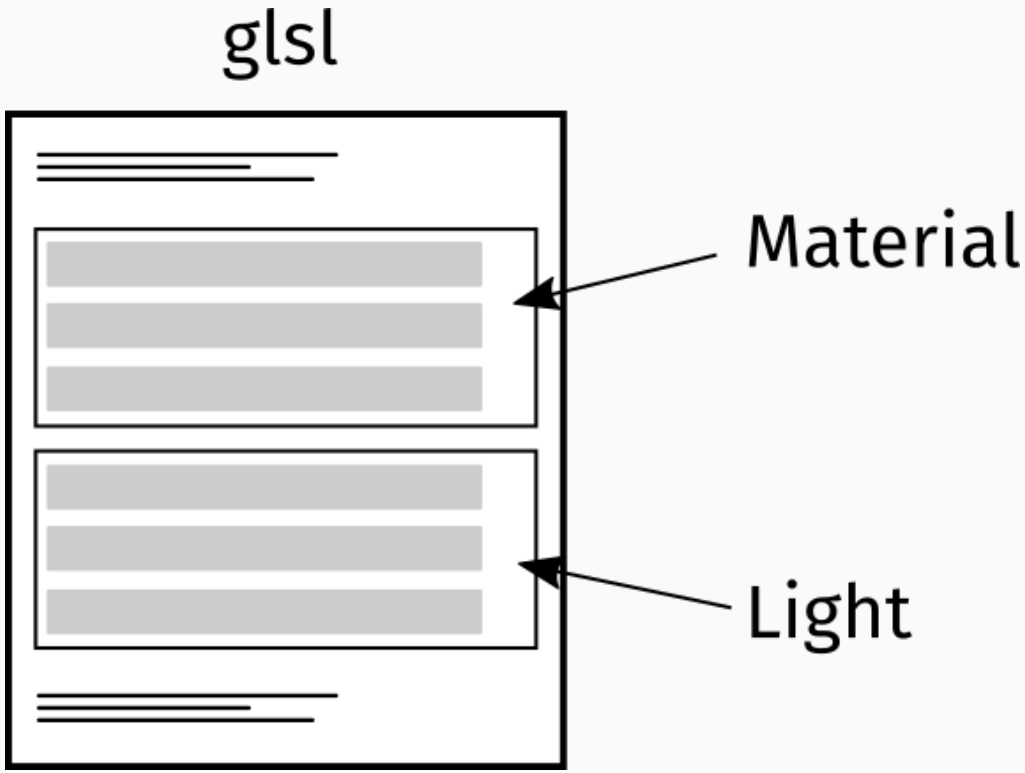


FIGURE 9 – Schéma d'un shader template.

MATÉRIAU

- Implémentation des nouveaux matériaux selon l'interface `Ra:Engine::Material`
- Mise en place de shaders communs pour le rendu à base de micro-facettes et spécifiques selon la technique

Sources ponctuelles nombreuses - calcul

Implémentation de trois étapes de calcul pour permettre le rendu temps-réel de nombreuses lumières :

`LightFiltering` Premier tri des lumières pour ne conserver que celles dans le Frustum

`LightAssignation` Création des clusters et assignation des lumières

`IndirectLightning` Calcul de l'éclairage de chaque fragment

`Count`	Un compteur atomique pour compter le nombre de lumières visibles.
`LightDataBuffer []`	Les données des lumières.
`TileLightIndexesBuffer [NUM_TILES]`	le nombre de lumières par cluster.
`TileLightsBuffer [NUM_TILES][NUM_LIGHTS]`	Les indices des lumières par clusters.
`CubePlanesBuffer [NUM_TILES]`	Les plans de chaque face des clusters.
`ScreenIndexesDataBuffer [NUM_LIGHTS]`	Les indices des lumières visibles.

TABLE 1 – SSBO assurant la communication GPU-CPU.

Côté CPU

- `PolygonalLight` dérivée de `Light`
- Ensemble de points coplanaires
- Deux textures pré-calculées stockées en statique

Côté GPU

- Couleur
- Tableaux `uniform` de `vec3` (N fixé à 16 [1])
  - Le polygone original en espace monde (N)
  - Le polygone transformé (N)
  - Le polygone clippé ($2N$)

1. Mais peut être augmenté en fonction du nombre max. d'uniforms.

1. Récupération des paramètres de M en fonction de α et θ [2]
2. Transformation par M^{-1} et clipping du polygone
3. Intégration de la source

2. Avec θ l'angle d'incidence et α la rugosité.

- Dépendance de l'organisation des points définissant la surface d'où une précaution particulière si l'on veut construire un `Polygon` depuis un `Mesh`
- Pas d'ombres
- Et :

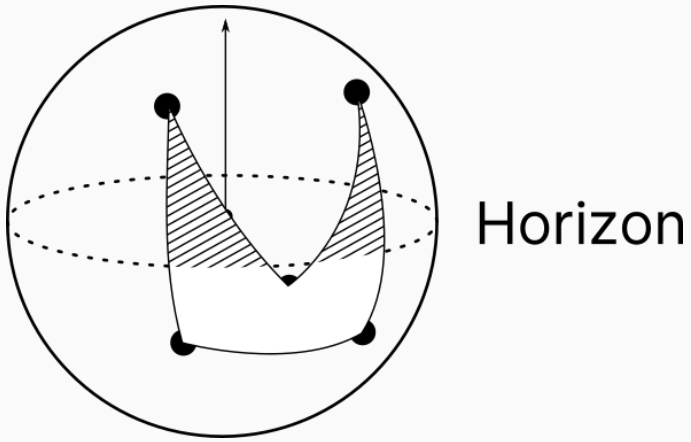


FIGURE 10 – Limitation attendue, mais en pratique aucun problème.

TESTS

- Framework de tests partiellement implémenté dans `Applications/PBRRadiumTest/Tests/`
- Classe paramétrique appelant un shader program spécifique
- Shader fait pour les sources polygonales en tant que démo
- **Code non appelé**

RÉTROSPECTIVE

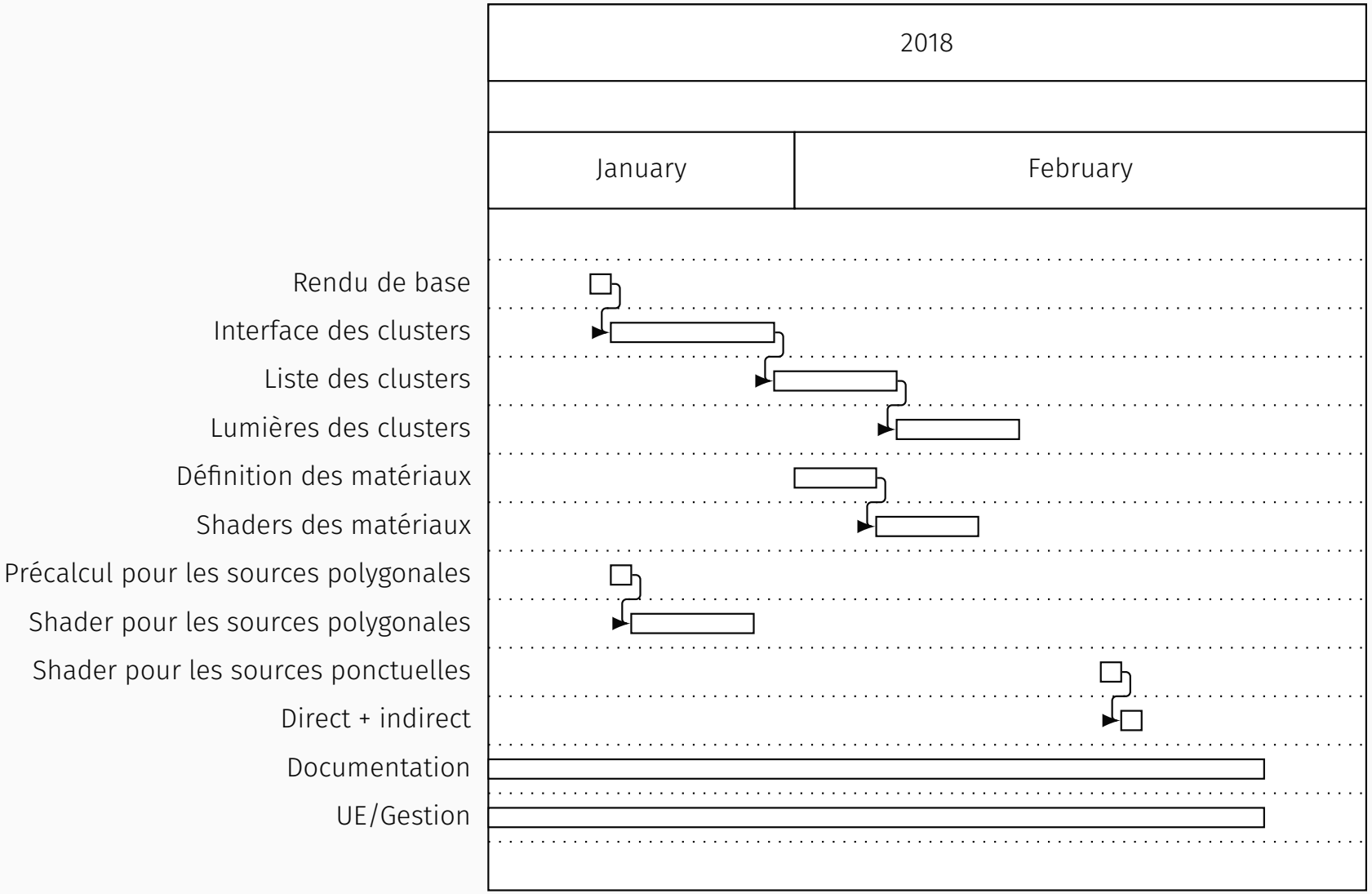
Risque	Survenu	Impact
Indisponibilité d'un membre	Non	∅
Mauvais choix de conception	Non	∅
Mauvais choix d'implémentation	Non	∅
Difficulté théorique	Non	∅
Dépassement des délais	Non	Fatigue
Clustering trop lent	Non	∅
Intégration en plugin	Non	∅
Divergence entre l'API prévue et réelle	Oui	Moyen

TABLE 2 – État des risques.

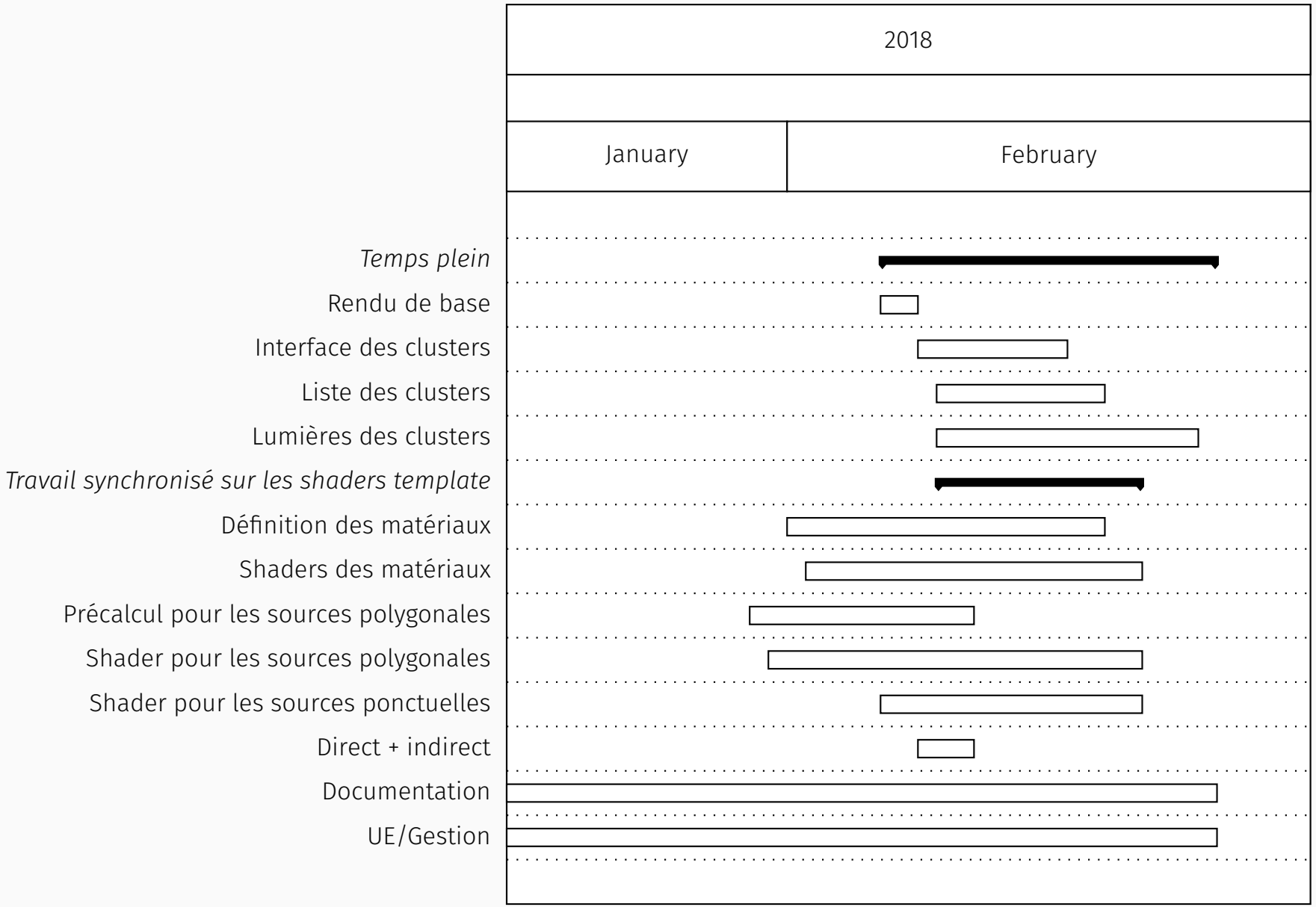
GANTT

- Deux semaines de temps plein
- Relativement respecté dans l'ordonnancement
- Pour les durées, analogie à la compression des plaques tectoniques

GANTT (AVANT)


2018
January
February
Rendu de base
Interface des clusters
Liste des clusters
Lumières des clusters
Définition des matériaux
Shaders des matériaux
Précalcul pour les sources polygonales
Shader pour les sources polygonales
Shader pour les sources ponctuelles
Direct + indirect
Documentation
UE/Gestion

GANTT (APRÈS)


2018
January
February
Temps plein
Rendu de base
Interface des clusters
Liste des clusters
Lumières des clusters
Travail synchronisé sur les shaders template
Définition des matériaux
Shaders des matériaux
Précalcul pour les sources polygonales
Shader pour les sources polygonales
Shader pour les sources ponctuelles
Direct + indirect
Documentation
UE/Gestion

Répartition des tâches

Tâche	Jours. Hommes	Dévelo-ppeurs
Rendu de base	2 1	2 1
Unification en shaders composites	6 ∅	2 1
Direct + indirect	3 3	3 3
Interface des clusters	8 20	1 2
Liste des clusters	6 12	1 2
Lumières des clusters	6 12	1 2
Shader pour les sources indirectes ponctuelles	1 3	1 3
Définition des matériaux	4 4	1 1
Shaders des matériaux	6 12	1 2
Précalcul pour les sources polygonales	1 4	1 1
Shader pour les sources polygonales	12 18	1 3

CONCLUSION

- Fonctionnel (au moins au niveau preuve de concept)
- Pull Request créée sur les branches `master` des dépôts
- Beaucoup de petites améliorations possibles/nécessaires

Perspectives de développement

- Général
  - Affichage des sources
  - Utiliser les tests
  - Création de lumières à la volée

PERSPECTIVES DE DÉVELOPPEMENT

- Matériaux
  - Gérer l'anisotropie dans le terme géometrique sur le matériau Metal

PERSPECTIVES DE DÉVELOPPEMENT

- Sources ponctuelles nombreuses
  - Utilisation d'une structure creuse pour l'association
  - Utilisation d'une `Texture Array` pour stocker les lumières
  - Utilisation d'une `Sparse Texture` pour stocker les indices des lumières
    - Permet de calculer les indices des lumières visibles
    - Pas d'itération nécéssaire sur l'enssemble des Lights
  - Utilisation d'une `Texture3D` pour stocker les clusters
  - Utilisation de la ZPrePass pour l'association

- Sources polygonales
  - Utiliser un stockage de plus grande capacité pour les vertices (SSBO, par exemple)
    - Un seul draw call en stockant plusieurs polygones simplement
    - Permettre des effets comme un polygone troué en conservant le double-face [3]
  - Trouver un algorithme plus rapide (peut-être moins général) pour le clipping du polygone
  - Générer automatiquement un `Polygon` depuis un `Mesh`

3. La valeur intégrée par le théorème de Stokes dépend du sens des arrêtes, donc par défaut un polygone n'éclaire que d'un côté. Pour corriger cela nous faison un `abs()`, mais on perd la possibilité de faire mathématiquement des trous en inversant le sens.

REMERCIMENTS

- L'équipe de développement du Radium Engine
- M.Paulin pour le code de base, les discussions et son aide précieuse
- L'équipe enseignante, pour la transmission du savoir nécessaire

Merci ! Des Questions ?