

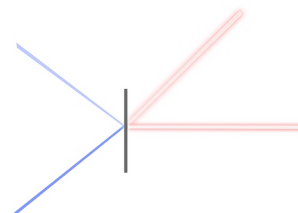
# Mirror Verse

## (MATH-INFO\_01)

Jérôme Bastien

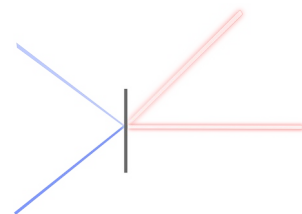
Guillaume Calderon  
Mohammad Ali  
Eymeric Déchelette

# Rappel des objectifs du projet



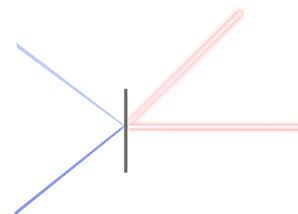
- Simuler les réflexions successives de rayons lumineux dans un ensemble de miroirs.
- Utiliser la simulation afin d'étudier le chaos du trajet d'un rayon lumineux dans un grand ensemble de miroirs.

# Grandes parties/Milestones

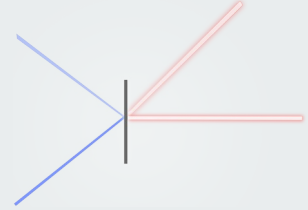


- Création d'une version simplifiée en 2D avec des miroirs plans.
- Amélioration du simulateur avec des miroirs plus complexes.
- Généralisation du simulateur en 3D ou plus.
- Analyse des résultats avec potentiel ajout d'outils d'analyse
- automatique selon les besoins (détection des boucles par exemple).

# Exigences à atteindre

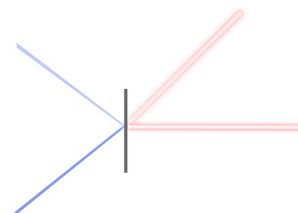


- Respecter au maximum le modèle théorique des rayons établi dans le cahier des charges.
- Supporter un grand nombre de miroirs (50+) et un grand nombre de réflexions (50+), sans souci de performance.
- Visualisation simple et intuitive de la simulation.
- Sauvegarde, création et modification des simulations et de leurs paramètres par le biais d'un format de fichier.



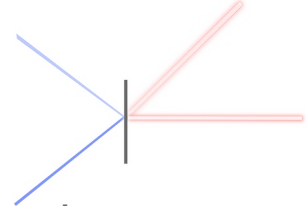
# Résultats et bilan

# Méthodes utilisés

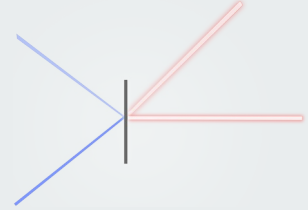


- Utilisation (un cas particulier) de la loi de Snell-Descartes pour la réflexion.
- Utilisation d'algèbre linéaire afin d'optimiser les calculs et de faciliter la généralisation en 3D et plus.
- Utilisation du langage Rust, pour sa vitesse d'exécution, ergonomie et sécurité.
- Utilisation de fichiers JSON pour créer des ensembles de miroirs et les stocker facilement.

# Résultats



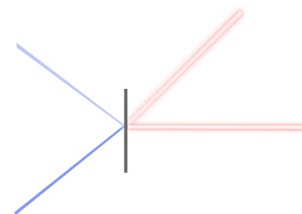
- Simulation des réflexions sur des (hyper)plans ainsi que sur des (hyper)sphère.
- Affichage en 3D fonctionnel (utilisé pour la 2D également).
- Génération aléatoire d'ensembles de miroirs.
- Sélection simple via interface graphique des différentes simulations.
- Détection de boucles infinies de réflexions.



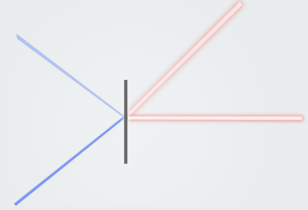
# Démonstration



## Points d'attention particuliers

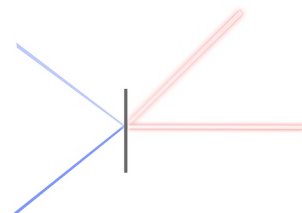


- Découpage du code particulièrement fin et propre (3 parties distinctes + librairie)
- Développement des miroirs de manière générique pour n-dimensions
- Rendu en 3D permettant le déplacement dans le monde pour mieux visualiser la simulation
- Documentation exhaustive du code



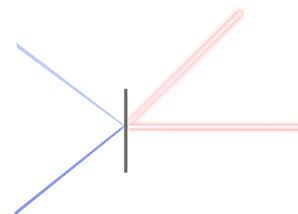
# Suite et Apports

## Suite



- Implémenter plus de miroirs différents comme les paraboloïdes ou surfaces de Bézier.
- Créer un éditeur de simulation.
- Ajouter une manière changer de dimension sans recompilation.
- améliorer les algorithmes de rendu pour augmenter la qualité de l'affichage.

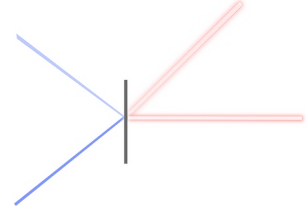
# Apports du projet



- Séparation des tâches :

- Eymeric : Implémentation des miroirs et UI, tests unitaire
- Guillaume : Rendu 2D et 3D de chaque miroir dans un monde 3D
- Mohammad : Algorithmes de réflexions, sauvegarde et chargement des simulations dans des fichiers, génération aléatoire, architecture générale du projet.

# Apports du projet



- Humainement :
  - Partage de connaissance (chacun à appris des autres)
  - Travail en équipe
  - Organisation sans temps prévu pour ce projet
- Gestion de projet:
  - Priorisation des tâches/organisation du temps
  - Explication, présentation et mise en valeur de notre travail