

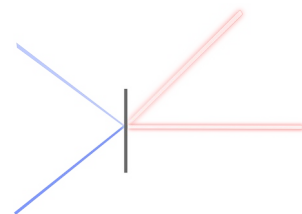
Mirror Verse

(MATH-INFO_01)

Jérôme Bastien

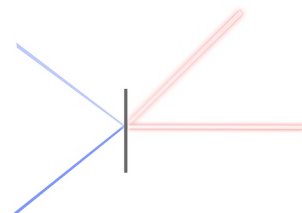
Guillaume Calderon
Mohammad Ali
Eymeric Déchelette

Rappel des objectifs du projet



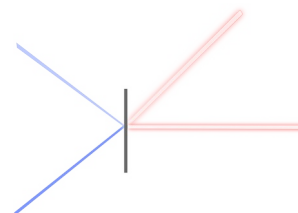
- Simuler la réflexion de rayons lumineux dans des miroirs
- Utiliser la simulation afin d'étudier le chaos du trajet d'un rayon lumineux dans un grand ensemble de miroirs

Grandes parties/Milestones

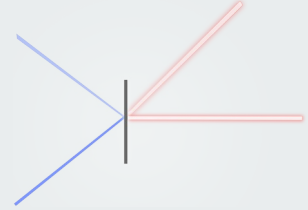


- Création d'une version simplifiée en 2D avec des miroirs plans
- Amélioration du simulateur avec des miroirs plus complexes
- Complexification du simulateur en 3D ou plus
- Analyse des résultats avec potentiel ajout d'outils automatiques selon les besoins (détection des boucles par exemple)

Exigences à atteindre

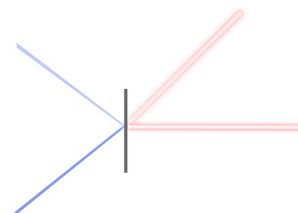


- Les simulations devront coller au maximum à la réalité
- Le simulateur devra supporter un grand nombre de miroirs (au moins 50) et un grand nombre de réflexion (au moins 50)
- Le programme devra permettre une visualisation simple de la simulation
- Les miroirs et rayons devront facilement pouvoir être modifiés depuis un fichier



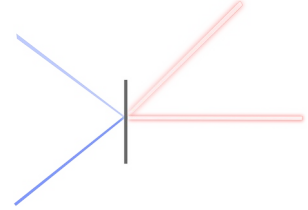
Résultats et bilan

Méthodes utilisés



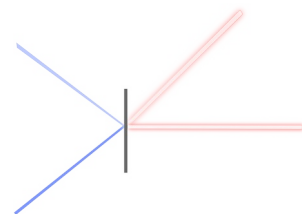
- Utilisation de la loi de Snell-Descartes pour la réflexion
- Utilisation d'algèbre linéaire afin d'optimiser les calculs et de supporter la 3d (et plus) facilement
- Utilisation du langage Rust car compilé donc très rapide et minimise le risque d'erreur
- Utilisation de json pour créer des ensembles de miroirs facilement

Résultat

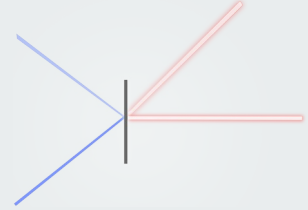


- simulation des réflexions sur des hyperplans ainsi que sur des n-sphère
- affichage en 3d fonctionel (utilisé pour la 2d également)
- Génération aléatoire d'ensemble de miroir
- Sélection simple via interface graphique des différentes simulations
- Détection de boucle infinie de réflexions

Points d'attention particulier

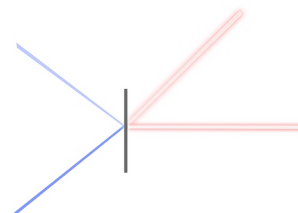


- Découpage du code particulièrement fin et propre (3 parties distinctes + librairie)
- Développement des miroir de manière générique pour n dimensions
- Affichage en 3d optimisé avec gpu. (pas sur de celle la mais ca peu ce tenter)
- Documentation du code très complete



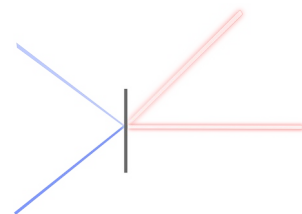
Suite et Apports

Suite



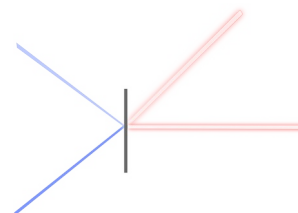
- Implementer plus de forme géométrique comme les paraboloids ou surfaces de bezier
- Créer un éditeur de miroir
- Améliorer le systeme de changement de dimension

Apports du projet



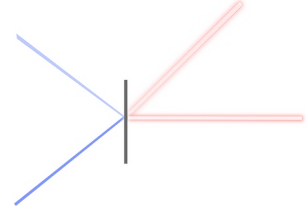
- Techniquement :
 - Eymeric : connaissance en Rust
 - Guillaume : connaissance en rendu 3d
 - Mohammad : connaissance en algèbre linéaire

Apports du projet



- Techniquement :
 - Eymeric : connaissance en Rust
 - Guillaume : connaissance en rendu 3d
 - Mohammad : connaissance en algèbre linéaire

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 taches/organisation en temps
 - Explication, présentation et mis en valeur de notre travail