

Projet Amélioration de l' algorithme Boyer- Bourdin'2000 (Auto-Adaptative)

Debbah Rabah Janvier 2021

Introduction :

Voici le rapport du projet Boyer-Bourdin'2000 (Auto-Adaptative).

Travail réalise :

Les objectifs du projet étaient les suivant :

Programmer un algorithme de référence, par exemple Bresenham'65.

Programmer l'algorithme demandé.

Comparer en temps et en utilisation mémoire les deux algorithmes.

Ici pour tester les performances des algorithmes on les relance 10 millions de fois afin d'avoir des stats précises concernant l'amélioration de ceux-ci un peu plus tard.

Cependant pour le moment je n'ai pas réussi à programmer l'algorithme Boyer-Bourdin car j'ai du mal avec le pas auto-adaptative.

Problèmes rencontrés :

Le premier problème rencontré était la gestion de mémoire afin d'avoir un affichage clair que ce soit sur l'affichage de la droite avec openGL, de plus le test mémoire ne donne pas de résultat très concluant.

Le deuxième problème est que je n'ai toujours pas réussi à réaliser une droite avec le deuxième algorithme (Boyer-Bourdin'2000 (Auto-Adaptative) malgré la documentation fourni et recherché sur le web.

Guide d'utilisation

Le programme est basé sur le tp réalisé en cours sur les images mais adapté pour les droites

De ce fait nous avons les 3 modes suivants:

- VIT pour rentrer en mode performance et afficher le temps pris par chaque

Algorithme pour calculer 10 millions de fois la droite avec les coordonnées fournis en arguments..

- DRAW permet de dessiner les droites des différents algorithmes et en les superposant pour montrer qu'elles sont bien semblables/
- M permet de vérifier la consommation de data des différents algorithmes .

Exemple :

8.573309 pour seconds br algo time

3.741293 seconds bresenham upgraded algo time

Bibliographie :

Totalité du cours de M.Bourdin sur les droites discrètes.

<https://home.mis.u-picardie.fr/~choplin/enseignement/dess/sim/t1/Bresenham/Bresenham.html>

https://www.researchgate.net/publication/3208895_Auto-adaptive_step_straight-line_algorithm

<https://ieeexplore.ieee.org/document/865882>

