

# UE : Dimensionnement et évaluation des architectures (I5AISE51)

## Sujet de TP GPU

Institut National des Sciences Appliquées de Toulouse  
P.-E. Hladik, pehladik@insa-toulouse.fr

—  
Version bêta (14 janvier 2021)

### 1 Travail à réaliser

Au cours des séances de TP vous allez devoir porter des algorithmes classiques sur une cible GPU. Vous travaillerez par binôme sur un sujet donné. Le but est de mettre en œuvre les concepts vus en cours et TD sur un problème concret.

Le travail à rendre se composera d'un document et du code source. Le document sera décomposé en :

- une présentation rapide du problème abordé,
- les grandes lignes de l'algorithme CPU,
- les grandes lignes de l'algorithme GPU,
- les lignes de commande pour compiler et exécuter le code (voire faire un script),
- la comparaison des performances CPU vs GPU,
- d'éventuelles pistes qui auraient pu être explorées.

Le code source devra être compilable et exécutable et comprendra :

- l'implémentation de l'algorithme CPU,
- l'implémentation l'algorithme GPU,
- l'affichage des temps d'exécution de CPU et GPU
- un système de test qui vérifie que les calculs CPU sont bien identiques à ceux sur GPU (par exemple en comparant l'ensemble des valeurs obtenues avec le CPU et celles obtenues avec le GPU).

N'hésitez pas à consulter toutes les sources possibles (internet, livre, etc.) tant que vous êtes capables de l'expliquer et d'avoir une version exécutable.

### 2 Organisation

L'organisation imaginée pour mener à bien ce TP est la suivante :

- séance 1 : choix du sujet, compréhension de l'algorithme CPU, implémentation (ou réutilisation) de l'algorithme CPU
- séance 2 : première version de l'algorithme GPU, fonction de comparaison CPU/GPU
- séance 3 : seconde version de l'algorithme GPU, relever des performance, écriture du document.

### 3 Sujets proposés

Ci-dessous une liste d'idée de sujets :

1. jeu de la vie : [présentation sur wikipedia](#), [version simple CPU](#)
2. algorithme génétique : [présentation sur wikipedia](#), [exemple en C++](#)
3. perceptron : [présentation sur wikipedia](#), [exemple d'implémentation en C++](#), [implémentation en C](#)
4. Transformée de Hough : [explication sur wikipedia](#), [exemple en C](#)
5. Pathfinding : [présentation wikipedia](#), [implémentation de A\\*](#)

Un sujet peut-être pris par deux binômes mais il est attendu au final un travail par binôme.

Idées abandonnées : boids, flocking, ant colony