

Liste des projets d'application

Projets à réaliser seul et à partager par exemple sur GitHub. Peut faire l'objet d'un projet plus conséquent réalisé à plusieurs (par exemple, faire plusieurs projets "imagerie" pour créer un gros projet d'une lib d'imagerie). Permet d'acquérir de l'expérience sur la maintenance et l'évolution de gros projets (et le travail en équipe si on développe à plusieurs).

Séparé par thématique. Se distingue des exercices, qui sont l'application directe du cours, les projets nécessitent des recherches personnelles pour comprendre le sujet. Le cours aborde que les généralités du langage, les projets permettent d'aborder les problèmes spécifiques de chaque domaine. Exemple de série de exercices/projets à réaliser :

1. exercice sur `iostream` pour enregistrer des données.
2. projet pour lire un fichier complet en mémoire. Un article est proposé à la lecture, pour présenter la problématique de la lecture d'un fichier par petits blocs.
3. projet pour sérialiser les données. Une référence aborde les problèmes de sérialisation (endianess, portage des types, etc)
4. etc.

Le but est :

- prendre l'habitude de lire les références en anglais pour étudier un problème.
- travailler sur des projets que l'on va améliorer progressivement (évolution et maintenance de code)
- travailler en équipe

Références données dans les énoncés des projets, librement accessibles.

Thématiques générales

Utiles pour tous les projets.

- Fichiers
- Design pattern. Référence :
<http://gameprogrammingpatterns.com/>
- algorithmes
- maîtriser la STL

Thématiques spécifiques

- analyse de texte : dictionnaire, parser
- web : génération de pages html, xml, json, web application
- infographie : analyse d'image, rendering, raytracing, géométrie
- modélisation : créer un modèle, simulation
- cryptologie : mot de passe, décodage/encodage
- analyse de données : statistique, data mining, machine learning
- traitement du signal
- Vision par ordinateur (computer vision)

Chapitre précédent	Sommaire principal	Chapitre suivant
Cours, C++		