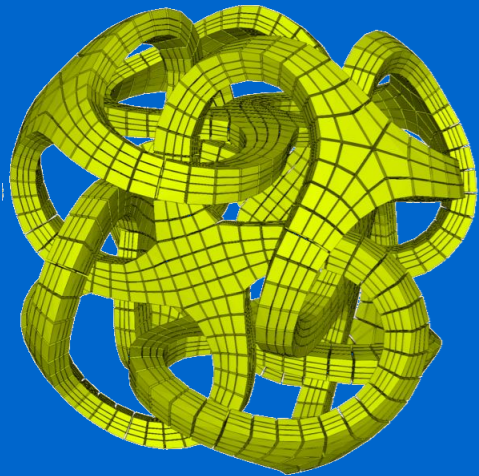
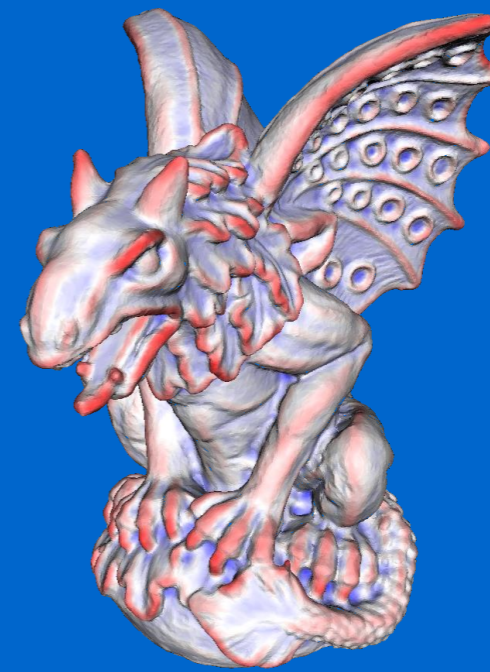




# CGoGN



Sylvain Théry

[thery@unistra.fr](mailto:thery@unistra.fr)

ICube

Love Data Week 2023  
MeetUp logiciels libres de recherche



- IUT informatique Lille
- Licence, maîtrise, DEA, Thèse (D. Bechmann) à l'Université Louis Pasteur Strasbourg, LSIIT 2000
- Post-doc à l'IRIT Université Paul Sabatier Toulouse 2001
- Ingénieur de recherche à l'IRCAD 2006
- Ingénieur de recherche dans l'équipe IGG du LSIIT (ICube)



Ingénieur de recherche développeur  
dans une équipe d'informatique





Un processus itératif (éternel recommencement)

- 2007 conception de CGoGN
- 2014 CGoGN 2
- 2019 CGoGN 3
- 2022 intégration de CGoGN dans la plateforme GAIA



informatique **G**raphique, **A**nalyse de données et **I**ntelligence **A**rtificielle



- Numérisation
- Gestion des données
- Traitement et analyse
- Modélisation et simulation
- Visualisation et interactions

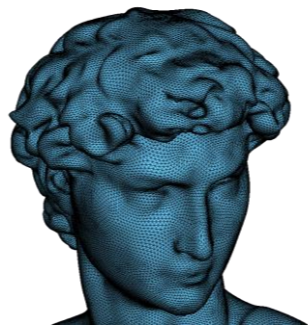
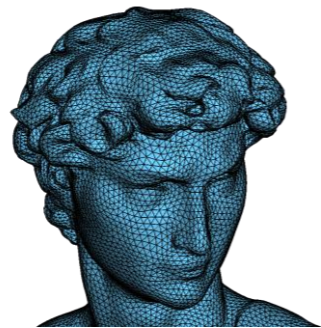
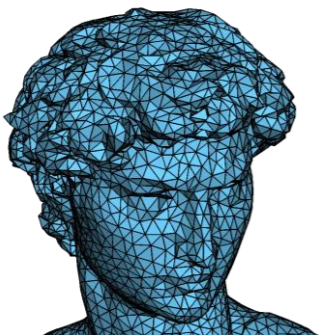
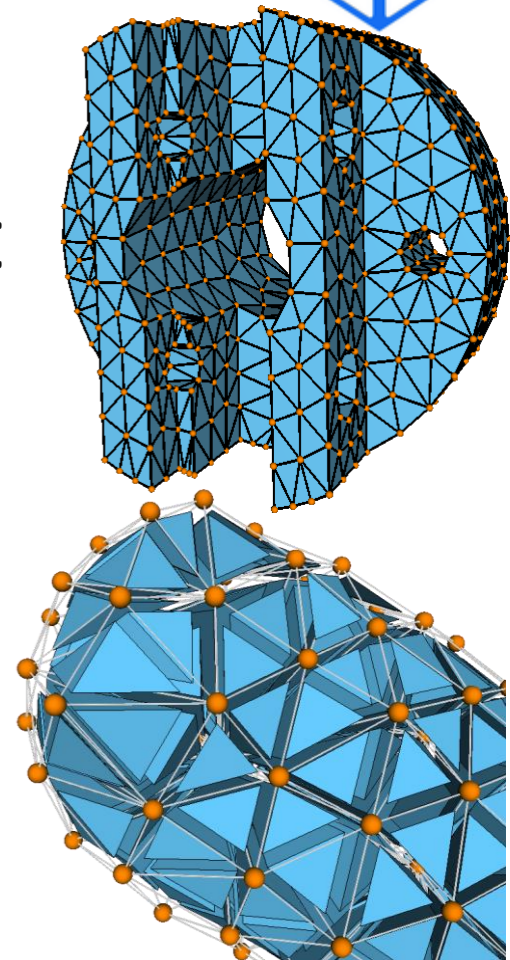


# Contexte: les cartes combinatoires

Les cartes combinatoires : un modèle permettant de représenter la topologie d'un maillage 2D/3D

Celui-ci permet de définir sûrement et optimalement

- les parcours globaux
- les parcours de voisinage locaux
- les opérations de bases: couper une arête, une face, basculer une arête, contracter une arête ...



# Contexte: les cartes combinatoires

CGoGN est une librairie C++ (17) qui propose une implémentation optimale des cartes (duales) surfacique et volumique.



Elle nécessite un minimum de connaissance du modèle sous-jacent pour être utilisée.

Elle doit être à la base du développement de l'application. Très difficile de l'ajouter à une application existante.



Principalement (localement) par et pour des thèses:

Encadrants:

- D. Bechmann, David Cazier, **Pierre Kraemer**

Doctorants:

- Thomas Jund, **Pierre Kraemer**, Lionel Untereiner, Kenneth Vanhoey, Paul Viville, Quentin Wendling, ...





- Versioning avec git
- Hébergement sur github.com
- les contributions se font uniquement à travers un système de pull-request
- Utilisation de branches spécifiques (master / develop / perso /...)
- Utilisation de l'intégration continue:
  - chaque PR est compilée et testée sur Linux, Windows et MacOS.





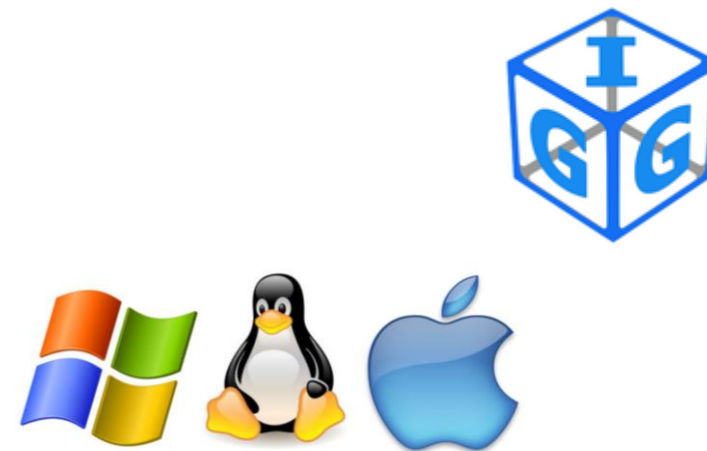


- C++
- Utilisation des spécificités du C++ 11/14/17 20?
- actuellement nécessite le support du C++17
- But: le code d'un algo utilisant CGoGN doit être le plus simple et le plus lisible possible.
- Parallélisation transparente & automatique disponibles à l'aide fonction très simples:

```
parallel_foreachCell(m, [&](Vertex v)  
{  
    ...  
});
```



le multi-plateforme (Linux Windows Mac)  
peut être très chronophage



Support d'OpenGL arrêté à 4.1 sur mac:  
empêche l'utilisation des compute shaders  
et de nombreuses possibilités introduites à partir du 4.3



CMake permet de gérer *presque* proprement la compilation sur tous les systèmes.



Gestionnaire de paquets:

Utilisation de brew (MacOS) et de vcpkg (Windows) permet de simplifier la gestion de dépendance.

Remarque: minimizer les dépendances, attention à leur péreinité. Attention aux licences.

*(CGoGN LGPL)*



# Evolutions & développement futur

En cours :

- passage du système de rendu de OpenGL à Vulkan



A l'étude:

- introduction de modèles topologiques plus exotiques (cartes généralisées, chaînes de cartes)
- Utilisation du C++20, concepts, modules ?

Dans un plus lointain futur:

- passage au langage *Rust*
- vraie topologie sur le GPU





# Questions ?

