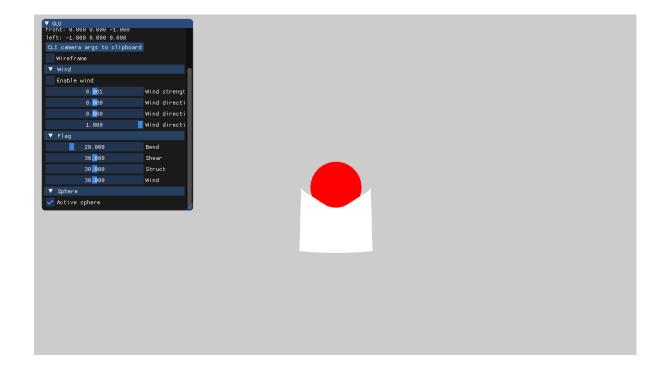
# Projet de moteur physique

Rapport : explication du viewer et difficultés rencontrées

https://github.com/LafLaurine/flag\_viewer

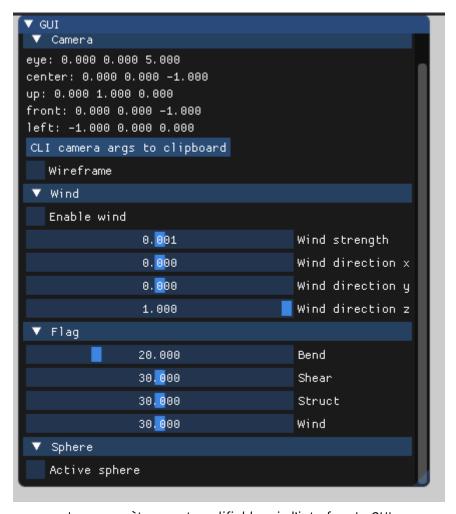


## Sujets choisi

Prolongation - thème 2 : collision avec sphère en calculant la distance entre la sphère et le drapeau. Si on atteint une certaine distance, alors il y a collision. L'état des particules est ensuite mis à jour.

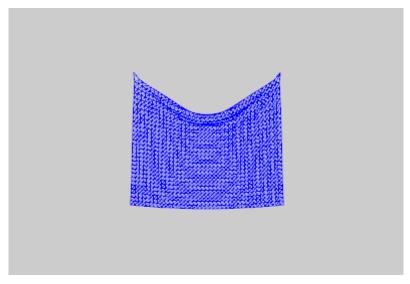
Malheureusement, il semblerait que l'étape de mise à jour des particules n'est pas fonctionnelle.

### Paramètres modifiable



Les paramètres sont modifiables via l'interface ImGUI

Activation du mode Wireframe : checkbox à cocher pour activer ou non le mode



Drapeau en mode wireframe

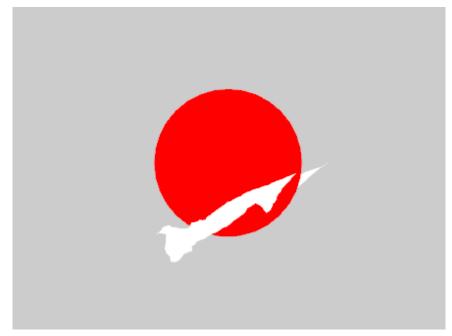
Activation du vent : checkbox à cocher pour activer ou non le vent

Puissance du vent : 0-0.02, slider pour gérer la puissance du vent

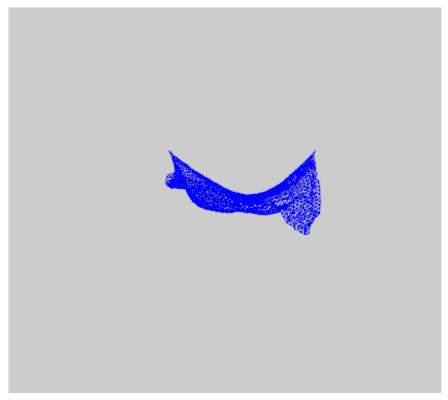
Direction du vent : 3 sliders différents pour la direction x,y et z

Force (contrainte) appliquée au drapeau : Cisaillement, Pliure, Vent et torsion. Chaque contrainte possède un slider pour faire varier les paramètres.

# Captures



Capture d'écran du drapeau qui ne réagit pas à la sphère



Capture d'écran du drapeau en train de réagir au vent

Vous pouvez retrouver une capture vidéo à l'adresse suivante : <a href="https://drive.google.com/file/d/10cApWNMI3nHV-FSI1SlejEkwVPzwi-nJ/view?usp=sharing">https://drive.google.com/file/d/10cApWNMI3nHV-FSI1SlejEkwVPzwi-nJ/view?usp=sharing</a>

### Difficultés

Après avoir essayé beaucoup d'approches différentes et différents calculs pour effectuer la collision de la sphère, j'ai dû finalement abandonner cette idée.

Il me semble que les forces s'appliquent bien, mais que la mise à jour des particules du drapeau ne s'effectue pas.

### Ressources

Site	Description	Illustration
An intro to modern OpenGL. Chapter 4: Rendering a Dynamic 3D Scene with Phong Shading	Explication sur comment construire un drapeau flottant en OpenGL 3.3	Mesh construction  The muchasac file contains code that generates the vertex and element arrays, collectively called a mesh, for the flag flaguole, ground, and wall objects that well be rendering. Most objects in the real world, including real flaguoles and flags, have smooth enviring surfaces, but graphics earlied and but harmages. To render these objects, we have to approximate their surfaces as a collection of tringles. We doubt by filing a vertex array with vertices placed along it surfaces, storing attributes of the surface with each vertex, and connecting the samples into tringless using the element array to give an approximation of the original surface.  The fundamental properties a mesh stores for each vertex are its position in world space and its normal, a vector perpendicular to the original surface. The normal is fundamental to shading calculations, as well see shortly. Normals should be unit vectors, but its, vectors whose length is one. Each vertex also has material parameters that indicate how the surface is shaded. The material can costist of as of the prevettex values, texture coordinates that sample material information from a texture, or some combination of both.
Cloth Simulation Coding Tutorial	Jesper Mosegaard explique différentes étapes comment gérer les particules du drapeau et gérer le vent appliqué (OpenGL)	Structural constraint Shear constraint Bending constraint
Andrei Toterman Github	Implémentation de l'article du dessus en OpenGL moderne	reimplementation of this https://viscomp.alexandra.dk/?p=147 article, using the programmable OpenGi. pipeline dependencies  • OpenGi. • OpenGi. • of the state of