Projet C++ et synthèse d'images

—IMAC 2—

BOIDS !

L'objectif de ce projet est de réaliser une application de visualisation de Boids originale, permettant d'exploiter vos nouvelles connaissances en algorithmique et OpenGL.

---

1 Contexte général de l'application

Votre application suivra, concernant notamment la partie C++, les consignes présentées dans le github suivant : https://julesfouchy.github.io/Learn–Clean-Code-With-Cpp/sujet/

Dans cette application, les boids sont "enfermés" dans un cube. Si vous avez probablement commencé en 2D, il s'agit maintenant de présenter une application en 3D respectant les consignes suivantes de la prochaine partie

1.1 IHM

Pour la GUI vous pouvez utiliser Dear ImGui qui fonctionne avec le contexte 3D que vous créerez. C'est celle déjà présente. Vous pouvez aussi utiliser NanoGUI. Ces bibliothèques d'IHM sont compatibles avec OpenGL et vous permettront de permettre à l'utilisateur de fixer les paramètres de la simulation.

2 Spécifications pour la partie 3D

Vous trouverez ci dessous la liste des spécifications qui vous permettront d'avoir un résultat correct pour la partie synthèse d'images.

2.1 Boids

Les boids devront présenter une **modélisation 3D originale**. Pour assurer une visualisation correcte qui dépende pas (trop) de l'ordinateur utilisé, vos boids **devront avoir plusieurs niveaux de détails**. Vous devrez donc utiliser ce qu'on appelle des LODs.

C'est l'application, via votre GUI, qui permettra de modifier le niveau de détail de tous les boids.

Les boids seront animés suivant les règles présentées dans la partie C++ avec notamment les comportement de séparation, d'alignement et de cohésion.

2.2 L'environnement

Le design de l'environnement est laissé à votre imagination mais il devra être constitué des éléments suivants :

— Le cube englobant la simulation : il devra être visible (il peut être très légèrement transparent éventuellement). Il devra être texturé.

— Des éléments de décors : leur modélisation devra être originale. Ces décors devront être positionné dans la scène et suffisamment nombreux et/ou complexe pour être d'importance. Ils devront être texturés

2.3 La caméra et le personnage principal : l'arpenteur

En plus des Boids et de l'environnement, vous devrez modeliser un troisième élément : l'arpenteur. L'arpenteur c'est l'utilisateur de l'application. Il est capable de se mouvoir comme il l'entend dans le cube mais ne peut pas en sortir. De plus **il devra être visible à la troisième personne**, à savoir que l'application montre l'arpenteur qui est controlé par l'utilisateur. On appelle cela une vue à la troisième personne (comme dans le jeu Tomb Raider typiquement)

La modélisation de l'arpenteur devra être originale et soignée.

2.4 L'éclairage et l'illumination

La scène devra être éclairée par au moins deux lumières ponctuelles dont une est située sur l'arpenteur.

De plus, une (ou plus) de vos lumières devra permettre de générer des ombres. Il vous faudra donc implémenter un algorithme permettant la génération des ombres. Vous êtes libre de choisir un des algorithmes vu en cours mais je vous déconseille l'algorithme des shadow volumes.

Vous avez le droit de sélectionner des *shadow casters* et des *shadow receivers* mais le groupe des *shadow casters* devra comprendre à minima les boids alors que le groupe des *shadow receivers* devra comprendre à minima les faces du cube de l'environnement.

3 Le programme

Votre programme sera codé en C++ et devra pouvoir compiler et s'exécuter dans vos salles machines de l'université. Les bibliothèques que vous pouvez utiliser son celles présentes dans le github et celles indiquées dans ce document. Si vous souhaitez utiliser d'autres bibliothèques vous devez obtenir l'accord des deux enseignants concernés (Jules Fouchy et Venceslas Biri).

Vous présenterez votre programme lors d'une soutenance. Nous vous indiquerons ultérieurement les modalités de cette soutenance.

4 Le rapport

Vous fournirez **en version électronique** un rapport de 10 pages maximum. **Consacrez suffisament de temps au rapport car il représente une partie de la note finale**. Essayez de respecter au mieux les directives suivantes :

- Le rapport vise à expliciter vos choix algorithmique et fonctionnel. Les éventuelles formules d'interaction nouvelles par exemple. Ou les détails sur la technique d'ombrage utilisée.
- Ne perdez pas de temps à réexpliquer le sujet du projet, l'enseignant le connaît déjà, faites seulement un bref résumé de l'idée générale de votre application. De manière plus générale, ne détaillez pas des méthodes déjà expliquées dans l'énoncé mais plutot celles que vous avez développées.
- Un rapport sert aussi à montrer comment vous avez fait face aux problèmes (d'ordre algorithmique). Certains problèmes sont connus (on en parle dans l'énoncé), d'autres sont imprévus. Montrez que vous les avez remarqués et compris. Donnez la liste des solutions à ce problème et indiquez votre choix. Justifiez votre choix (vous avez le droit de dire que c'est la méthode la plus facile à coder). A savoir : "on a eu un super problème par ce que ça compilait pas ..." ne nous intéresse pas.
- Il ne doit figurer aucune ligne de code dans votre rapport. Un rapport n'est pas un listing de votre programme où vous détaillez chaque fonction. Vous devez par contre détailler vos structures de données et mettre du pseudocode pour expliquer vos choix algorithmiques.
  Il est autorisé d'utiliser des "raccourcis" tels que "`initialiser le tableau tab a 0`" plutôt que de détailler la boucle faisant la même chose.
- n'hésitez pas à mettre des images dans votre rapport pour illustrer vos propos et vos résultats.
- Enfin, il est **très important** de faire la liste de ce que vous avez fait, de ce qui fonctionne correctement et la liste des dysfonctionnements. Précisez quand il s'agit d'options que vous avez rajoutées en plus de ce qui était demandé. N'hésitez pas à montrer des timings et des screenshots.

5 Pour finir

Pour la partie synthèse d'images, votre projet sera évalué sur :

- La qualité de la modélisation notamment si elle est originale
- La qualité et la cohérence de l'illumination
- La qualité de l'animation et du déplacement des éléments de la scène
- L'aisance de l'IHM
- La qualité de l'algorithme d'ombrage

Comme indiqué précédemment le respect "simple" des spécifications vous permettra d'obtenir une note correcte. Pour aller plus loin, concernant la partie Synthèse d'images, vous pouvez libre cours à votre imagination. Vous pouvez par exemple :

- Proposer des modèles d'illumination plus complexe
- Proposer un cycle jour / nuit

— Implémenter des effets spéciaux avec nuage de particules
— Animer l'arpenteur