OpenGL - TD 05

Exercices Avancés

Ce TD vous propose des exercices avancés, qui vous demanderont de faire une synthèse de ce que vous avez appris durant les précédentes séances de TD

Ces exercices sont indépendants et peuvent être fait dans l'ordre que vous souhaitez.

Notez que les consignes dans ce TD seront moins directives où détaillées que dans les précédents TDs, et que différentes approches pourraient répondre à une même consigne. Ce TD est le moment idéal pour expérimenter un peu plus le fonctionnement d'OpenGL et de faire parler votre créativité une première fois avant d'aborder le projet final.

Prérequis : Avoir complètement terminé les TD 01, 02, 03 et 04.

Notice – Heure courante

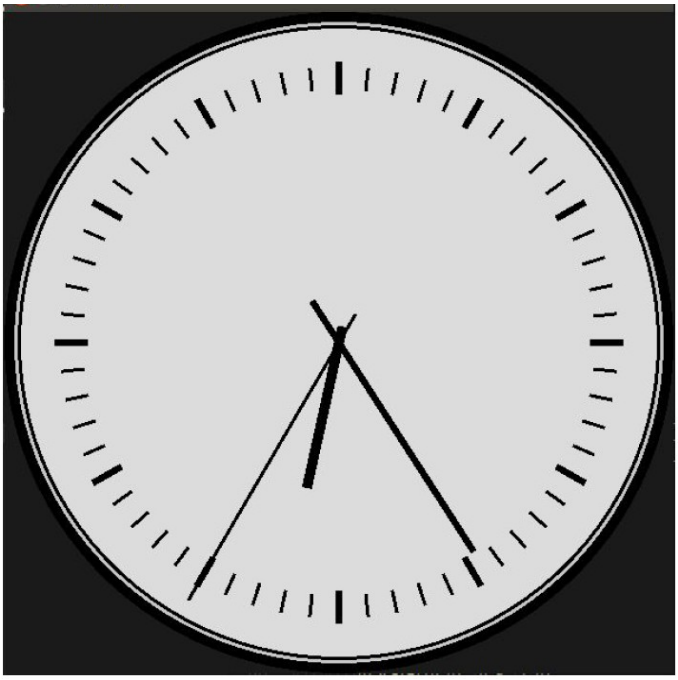
Pour récupérer l'heure système actuelle, vous pouvez utilisez les fonctions de `<time.h>` :

```
#include <time.h>
// ...
time_t rawtime;
struct tm* timeinfo;

time(&rawtime);
timeinfo = localtime(&rawtime);

printf(“hours %d\n”, timeinfo->tm_hour);
printf(“minutes %d\n”, timeinfo->tm_min);
printf(“seconds %d\n”, timeinfo→tm_sec);
```

Exercice 01a – Horloge mécanique



A faire :

01. Dessinez une horloge mécanique avec ses subdivisions et ses 3 aiguilles. Pensez à réutiliser les fonctions de dessin canoniques du TD 02.

02. Faites en sorte que les 3 aiguilles bougent et donnent l'heure du système.

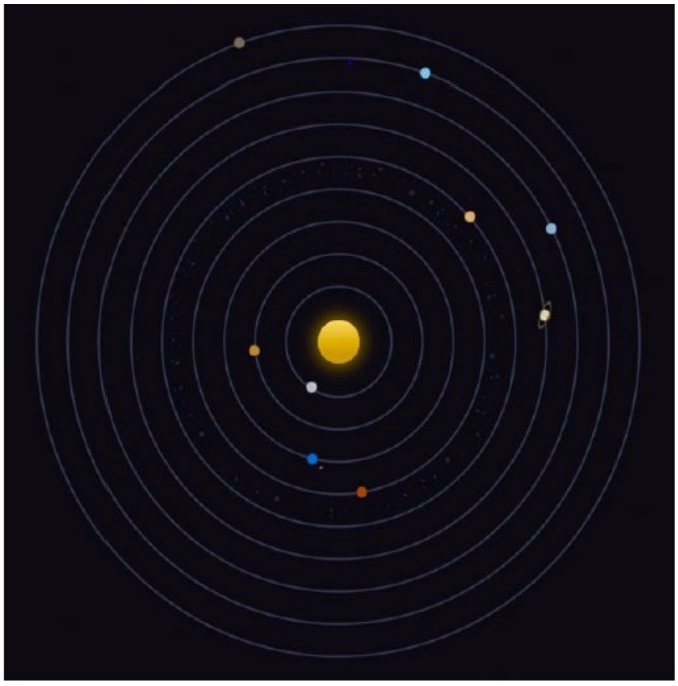
Exercice 01b – Horloge digitale

A faire :

01. Affichez une horloge numérique, en vous servant des images dans *doc/numbers.zip*

02. Faites en sorte que les chiffres affichés correspondent à l'heure du système.

Exercice 02 – Système solaire



A faire :

Réalisez une représentation de notre système solaire. Essayez de représenter les planètes avec des proportions cohérentes pour leur tailles, vitesses, et distance par rapport au soleil.

Vous pouvez récupérer les données de chaque planète sur la page Wikipédia suivante : https://en.wikipedia.org/wiki/Solar_System#Distances_and_scales

01a. Représentez le soleil et les planètes par des sphères de couleur.

01b. Faites en sorte que les planètes tournent autour du soleil avec des vitesses cohérentes.

02. Représentez les planètes et le soleil par des textures. Vous pouvez récupérer des textures depuis : https://www.solarsystemscope.com/textures/

03. Ajoutez quelques satellites (Lune, Titan, Callisto, …) autour de leur planètes respectives. Vous pourrez également récupérer les données depuis les pages Wikipédia adéquates.

04. Faites en sorte de pouvoir déplacer la caméra dans le système solaire :
- lorsque l'utilisateur appuie sur une touche du clavier, la caméra suive une planète
- lorsque la molette est tournée, la caméra zoome / dezoom par rapport à la planète suivie
- un drag&drop permet de décaler la position de la caméra

Pour arriver à ce résultat, vous devrez déplacer la fenêtre virtuelle d'OpenGL, en faisant varier les paramètres envoyés à `gluOrtho2D`.

05. Représentez la ceinture d'astéroïdes situé entre Mars et Jupiter (a.k.a. ceinture de Kepler), en créant aléatoirement des points situés entre ces deux astéroïdes.

Cette ceinture recouvre un anneau située entre 1.8 et 4.2 Unités Astronomiques du soleil.

Pensez à faire varier la taille des point via la fonction `glPointSize`. Idéalement, vous ferez aussi varier la couleur des ces points dans des teintes gris/marron.

06. Représentez maintenant le système en donnant des orbites elliptiques aux planètes, plus proches de celles qu'elles décrivent dans la réalité.

Vous aurez pour cela besoin des valeurs d'Aphélie et de Périhélie pour chaque planète, que vous pourrez trouver sur leur pages Wikipédia respectives.

Pour tracer une ellipse, vous pouvez au choix :
- dessiner un cercle en appliquant un facteur de distance ou d'échelle sur chacun des axes
- utiliser des fonctions mathématiques plus complexes de dessins d’ellipses comme celle expliquée à cette adresse : https://www.mathcurve.com/courbes2d/ellipse/ellipse.shtml