Travaux pratiques
3 La scène

Dans ce TP, vous allez finir la création du logiciel de visualisation d'images 3D.

Classe Scene : Implémente une scène. Cette classe doit hériter de la classe `SceneInterface` et implémenter toutes ses méthodes. La scène contient :

- Un pointeur sur la GUI.
- Une caméra (classe `Camera`).
- Une liste d'objets 3D (classe `Object3D`).

La classe doit, entre autres, définir les méthodes suivantes.

- `draw() const`
  Dessine tous les objets dans le champs de vision de la caméra.
- `void draw_object( Object3D * const ) const`
  Dessine toutes les faces de l'objet passé en argument qui sont tournées vers la caméra.
- `void draw_wire_triangle( const Triangle & ) const`
  Dessine la face passée en argument (les trois arrêtes du triangle).
- `void draw_edge( const Point<float, 4> &, const Point<float, 4> & ) const`
  Dessine le segment passée en argument.
- `Point<float, 4> perspective_projection( const Point<float, 4> & ) const`
  Projette le point passé en argument sur l'écran (« near plane »).

Classe Camera : Implémente la caméra. La caméra contient :

- La hauteur et largeur de l'image (écran).
- Sa vitesse de déplacement.
- Sa vitesse de changement d'orientation.
- Sa vitesse de zoom.
- Sa position actuelle.
- Son orientation actuelle.
- Sa vitesse de déplacement actuelle (une valeur pour chaque axe x, y et z).
- Sa vitesse de changement d'orientation actuelle (également trois valeurs).
- Sa vitesse de zoom actuelle.
- La distance actuelle entre la caméra et le plan de projection.
- Son champ de vision (Tronc).

Cette classe doit définir au moins les méthodes suivantes.

- `move_...()`
- `turn_...()`
- `zoom_...()`
- `stop_move...()`
- `stop_turn...()`
- `stop_zoom...()`
- `Transformation get_transform() const`
  Retourne la transformation qui correspond au point de vue la caméra.
- `bool outside_frustum( const Sphere & ) const`
  Retourne si la sphère passée en argument est en dehors du champ de vision de la caméra.
- `bool sees( Triangle & ) const`
  Retourne si la caméra « voie » la face triangulaire passée en argument.
- `LineSegment visible_part( const LineSegment & ) const`
  Retourne la partie visible du segment passé en argument.
- `void update()`
  Met à jour la position et orientation de la caméra.

Classe Object3D : Implémente un objet 3D. L'objet 3D contient :

- Un nom.
- Une position.
- Un vecteur de sommets.
- Un vecteur de faces (la « mesh »).

Cette classe doit définir au moins les méthodes suivantes.

- `Sphere bsphere() const`
  Retourne la « bounding sphere ».
- `Triangle face( unsigned int ) const`
  Retourne la *n*-ième face de l'objet où *n* est passé en argument.
- `unsigned int num_faces() const`
  Retourne le nombre de faces de l'objet.
- `void add_face( unsigned int, unsigned int, unsigned int )` Ajoute une face à l'objet. Les trois entiers passés en arguments se réfèrent à la liste de sommets.
- `void remove_face( unsigned int )` Supprime une face de l'objet. L'entier passé en argument se réfère à la liste de faces.

Classe Frustum : Implémente le champ de vision de la caméra. L'objet contient 6 plans :

- Plan proche.
- Plan lointain.
- Plan droite.
- Plan gauche.
- Plan supérieur.
- Plan inférieur.

Cette classe doit définir au moins les méthodes suivantes.

- `bool outside( const Point<float, 4> & ) const`
  Retourne si le point passé en argument est en dehors du champ de vision.
- `bool outside( const Sphere & ) const`
  Retourne si la sphère passée en argument est complètement en dehors du champ de vision.
- `LineSegment inter( const LineSegment & ) const`
  Retourne l'intersection du segment avec le champ de vision (la partie visible).
- `void update( float h, float v, float e )`
  Met à jour la position du champ de vision, où *h* est la résolution horizontale, *v* est la résolution verticale et *e* est la distance entre le plan de projection et la caméra.

Module principal : Dans le module principal du programme, vous devez implémenter deux fonctions.

- `void load_geo_file( const * char file, Scene & scene )`
  Ouvre un fichier au format `.geo` et insère l'objet dans la scène.
- `int main( int argc, const char * argv[] )`
  Initialise la GUI, lit le fichier (où les fichiers) au format `.geo` passés en arguments, exécute le `main_loop` et ferme la GUI. La fonction doit aussi capturer les éventuelles exceptions et les traiter si possible.