

## IFT3355 : Infographie

### Travail Pratique #1 : 15%

Disponible : Vendredi 8 février 13 :30  
Remise : Jeudi 21 février 13 :30 au début de la démo  
Equipes : Deux étudiant(e)s

## 1 Description

Pour ce premier travail pratique, on vous demande d'implémenter une grue mécanique dont la tâche est de transporter une voiture dans un compacteur. La grue est composée de différentes articulations pour lui permettre d'atteindre son objectif. Une fois la voiture atteinte par l'aimant situé à l'extrémité du bras articulé, elle s'accrochera à cet aimant. À l'aide des contrôles du clavier, la grue devra transporter la voiture vers le compacteur. Au contact du compacteur, celui-ci sera activé et compressera la voiture. Le but de ce travail est de vous familiariser avec les diverses transformations matricielles qui sont fréquemment utilisées en infographie.

## 2 Détails

### 2.1 Assemblage

Les objets sont construits à partir de primitives que vous avez à assembler dans votre programme afin de constituer une scène similaire à celle du prototype. Chaque primitive est de taille fixe et centrée à l'origine. Plus précisément, on a

- Cube :  $[-1, 1]^3$
- Cylindre :  $\{(x, y, z) \mid x^2 + z^2 = 1, y \in [-1, 1]\}$
- Plan :  $\{(x, 0, z) \mid (x, z) \in [-1, 1]\}$
- Sphère :  $\{(x, y, z) \mid x^2 + y^2 + z^2 = 1\}$

#### 2.1.1 Grue

Quoiqu'il ne soit pas requis de recréer fidèlement le prototype, voici quelques indications pouvant diriger la création de la grue. La grue, telle que présentée dans le prototype, est constituée de quinze parties distinctes dont les dimensions sont les suivantes

1. Un cube pour le tracteur de hauteur 1, et de largeur et longueur 2,
2. Un cube pour la remorque de hauteur 1, de largeur 2 et de longueur 4,
3. Un cube pour la cabine de dimensions 1.5,
4. Un cylindre pour chacune des six roues de largeur 0.2 et de rayon 0.5,
5. Un cube pour la base rotative de hauteur 0.5, et de largeur et longueur 1.5,
6. Un cylindre pour chacun des deux bras de hauteur 4 et de rayon 0.2,
7. Un cylindre pour le câble de hauteur 4 et de rayon 0.05,
8. Un cylindre pour l'aimant de hauteur 0.2 et de rayon 0.5,

9. Une lumière de type « spot » et d'angle d'ouverture de  $45^\circ$  située sur la cabine éclairant devant la grue.

La hiérarchie de la grue est relativement simple. Celle-ci commence par le tracteur qui constitue sa base. À ce dernier sont attachées la cabine, deux roues et la remorque. Les quatre autres roues sont attachées à la remorque, ainsi que la base rotative. Le premier bras est attaché à la base rotative, puis le deuxième bras au premier. On attache ensuite le câble au deuxième bras pour finalement attacher l'aimant au câble.

Si vous décidez de ne pas recréer fidèlement la grue du prototype, vous devez néanmoins construire une grue de complexité égale (ou supérieure) afin d'obtenir tous vos points.

Les détails de construction de la grue se trouvent dans le constructeur de la classe *Scene*. Quoiqu'il soit possible de créer tous vos objets directement dans la classe *Scene*, il est conseillé de créer plusieurs classes pour chacun de vos objets.

### 2.1.2 Voiture

La voiture est constituée des six parties suivantes

1. Un cube pour la carrosserie de hauteur 0.6, de largeur 2 et de longueur 4,
2. Un cube pour le toit de hauteur 0.6, de largeur 1.6 et de longueur 2,
3. Un cylindre pour chacune des quatre roues de largeur 0.2 et de rayon 0.3.

Sa hiérarchie est formée de la carrosserie à sa base, à laquelle sont attachés les quatre roues et le toit.

### 2.1.3 Compacteur

Le compacteur est constitué des six parties suivantes

1. Un cube pour l'ancrage de dimensions 2,
2. Un cylindre pour chacun des deux bras de longueur 2 et de rayon 0.1,
3. Un cube pour chacune des plaques de compactage d'épaisseur 0.4 et, de largeur et hauteur 2.

Sa hiérarchie est composée de deux parties symétriques. Chaque partie est composée d'un bloc à sa base, auquel est attaché un bras, puis la plaque de compactage est attachée au bras.

## 2.2 Contrôle de la grue

Pour contrôler la grue, utilisez la touche *w* pour avancer et les touches *a* et *d* pour tourner à gauche et à droite respectivement. Les roues avant de la grue devront s'orienter en fonction des touches du clavier et la grue devra se déplacée en fonction de l'orientation de ses roues avant. La remorque étant connectée par un attelage au tracteur, son orientation doit suivre celle du tracteur avec un léger retard de façon à rendre le mouvement de la grue plus réaliste.

La rotation de la base de la grue est contrôlée par les touches *q* et *e*. La rotation du premier bras articulé est contrôlée par les touches *f* et *r*. La rotation du deuxième bras articulé est contrôlée par les touches *g* et *t*. Le câble doit toujours rester perpendiculaire au sol.

Cette animation doit être faite dans la méthode *animate* de la classe *Scene* ou de votre classe *Grue*, le cas échéant.

## 2.3 Collision avec l'aimant

La première partie de l'animation consiste à déplacer la grue afin d'aller prendre la voiture. Pour ce faire, il vous faut détecter si l'aimant et la voiture sont à une distance inférieure à un seuil que vous avez fixé. Dans le cadre de ce travail pratique, cette distance est simplement celle entre deux points sélectionnés sur chacun des objets. Ces points sont le centre du dessous de l'aimant et le centre du dessus du toit de la voiture. Le calcul de la distance doit se faire en espace monde. Le prototype utilise un seuil de 0.4.

Pour convertir un point en espace local (ou objet) en un point en espace global (ou monde), utilisez la matrice de transformation globale qui se trouve par la méthode *getGlobalTransformation()* de l'objet de la hiérarchie.

Lorsque qu'il y a détection de collision, ou plutôt de proximité, vous devez modifier la hiérarchie afin que la voiture soit attachée à l'aimant. Pour ce faire, vous devez utiliser la méthode *setParent* de l'objet concerné qui modifiera la hiérarchie en conséquence. Il n'est pas nécessaire de faire une animation de transition afin de rendre le changement de parents plus fluide.

Cette détection de proximité doit être faite dans la méthode *handleCollisions* de la classe *Scene*.

## 2.4 Collision avec le compacteur

La deuxième partie de l'animation consiste à déplacer la voiture vers le compacteur à l'aide de la grue. De la même façon qu'on a détecté la proximité entre l'aimant et le toit de la voiture, on va maintenant détecter la proximité entre le dessous de la carrosserie et le sol au centre du compacteur. Lorsque la distance entre ces deux points est inférieure à un seuil, e.g., 1 dans le prototype, vous devez effectuer une modification de la hiérarchie afin que le compacteur soit maintenant parent de la voiture.

Il n'est pas nécessaire de faire une animation de transition afin de rendre le changement de parents plus fluide.

Cette détection de proximité doit être faite dans la méthode *handleCollisions* de la classe *Scene*.

## 2.5 Animation du compacteur

La troisième partie de l'animation consiste à compacter la voiture. Les plaques de compactage doivent donc faire un aller-retour vers le centre du compacteur, tout en ajustant la longueur des bras du compacteur. Pendant ce temps, la voiture est compactée, c'est-à-dire qu'elle subit une mise à l'échelle dans le sens de sa longueur. On suppose ici que la voiture est centrée et aligné avec l'orientation du compacteur.

## 3 Bonus

Un point supplémentaire peut être obtenu pour chacune des améliorations suivantes

1. Carte d'élévation. Pour cette partie, il vous faudra créer un nouvel objet gérant une carte d'élévation (ou *heightmap*) qui fera office de terrain accidenté. Votre grue devra alors se déplacer sur ce terrain en respectant l'orientation de ce dernier. À chaque instant, au moins trois roues devront être en contact avec le sol.
2. Compactage++. L'implémentation de base simplifie de façon exagérée l'étape de compactage. Pour plus de réalisme, cette partie consiste à pouvoir déposer la voiture à n'importe quel endroit, puis de faire le compactage de la voiture en conséquence. Au contact d'une seule plaque de compactage, la voiture doit subir une rotation et une translation relative à son centre de masse. Lorsque la deuxième plaque entre en contact avec la voiture, elle est alors compactée dans l'orientation appropriée.

Quoiqu'il soit possible d'obtenir 17 points pour ce travail, le total des points des travaux pratiques de la session est de 50% et ne pourra pas être dépassé.

## 4 Support au développement

Un squelette du programme que vous avez à concevoir est disponible sur le site web du cours. Ce squelette comprend le nécessaire pour créer un fenêtre OpenGL, créer des objets géométriques, des caméras et des sources lumière. Vous aurez à compléter la classe *Scene* en créant les hiérarchies, l'animation et la détection de collisions.

## 5 Outils disponibles

Vous avez à votre disposition un prototype qui démontre le résultat final attendu. Celui-ci peut aussi être utile pour l'assemblage des objets de la scène.

Le code est basé sur la librairie d'interface graphique Qt. Cette librairie fournit la gestion des événements, la création de fenêtres et certains outils mathématiques. Parmi ces classes, vous aurez besoin de *QMatrix4x4* et de *QVector3D* qui vous seront utiles pour la construction de vos matrices de transformation. Entre autres, vous apprécierez les fonctionnalités suivantes :

1. `QMatrix4x4::scale( QVector3D s )` pour une mise à l'échelle non uniforme de facteurs  $s$ ,
2. `QMatrix4x4::translate( QVector3D t )` pour une translation par le vecteur  $t$ ,
3. `QMatrix4x4::rotate(  $\theta$ , QVector3 v )` pour rotation autour de l'axe  $v$ ,
4. `QMatrix4x4::operator*( QMatrix4x4 m )` pour la multiplication matricielle,
5. `QMatrix4x4::operator*( QVector3D v )` pour la multiplication avec le vecteur  $v$  en coordonnées homogènes, en supposant que la 4ème coordonnée est égale à 1.

Pour plus d'informations consultez la documentation de Qt sur lesdites classes. Vous pouvez télécharger gratuitement la librairie *Qt* et l'éditeur *Qt Creator* sur le site web [www.qt-project.org](http://www.qt-project.org). Les versions 4 et 5 sont compatibles avec le code fourni.

## 6 Délivrables et rapport du travail

Nous aurons besoin d'une version électronique de votre programme pour l'évaluer. Vous devez aussi écrire un rapport d'environ 3-4 pages qui contiendra :

1. L'énoncé du problème à résoudre.
2. La description des techniques utilisées. Décrivez les principes théoriques que vous avez appliqués pour obtenir les matrices de transformation. Spécifiez de quelle façon vous avez animé la grille. Décrivez la méthode employée pour traiter les collisions. Nous voulons ici le fondement de vos techniques, pas les détails de la disposition du code.
3. Les problèmes reliés aux techniques implémentées, suggestions pour améliorer la solution au problème, extensions possibles.
4. Références.

La remise se fera par StudiUM.

## 7 Critères d'évaluation

Section	Critères	Pondération
Code	style, clarté, efficacité, résultats	80%
Rapport	esthétique, énoncé, descriptions	10%
Analyse	problèmes, améliorations, extensions	10%

## 8 Bibliographie

P. Shirley et S. Marschner. Fundamentals of Computer Graphics. 3ème édition, AK Peters, 2009.