TP OpenGL 4
Animation

Licence Informatique 3ème année

Année 2016-2017

La bibliothèque de fonctions Glut offre une fonction particulière, nommée `glutIdleFunc`, qui permet de spécifier une fonction à déclencher lorsque le programme est **en attente d'un événement**. En effet, l'application que vous avez développée jusqu'à présent, passe plus de 90% de son temps à attendre que vous déclenchiez un événement pour réagir et traiter cet événement. Ce temps inutilisé peut alors être utilisé par une fonction pour effectuer des traitements internes à l'application.

Dans cette partie, nous allons profiter de cette fonctionnalité pour animer simplement des objets. Cette animation se fera par l'intermédiaire de transformations géométriques qui seront appliquées automatiquement par une fonction tournant « en fond de tâche » , cette fonction étant précisée par l'intermédiaire d'un appel à `glutIdleFunc`.

1 La fonction `glutIdleFunc`

Le prototype de cette fonction est le suivant :

```
void glutIdleFunc(void (*fonc)(void))
```

Son paramètre représente l'adresse de la fonction à déclencher lorsque l'application est inactive. A noter que dès que votre application sera en état d'attente, cette fonction sera appelée automatiquement ; dès que son exécution sera terminée, et si l'application n'a toujours aucune action à effectuer, cette fonction sera à nouveau appelée, et ainsi de suite durant toute la durée de l'application.

Il est cependant possible, en cours d'exécution de l'application, de désactiver la fonction s'exécutant en fond de tâche, en faisant un nouvel appel à `glutIdleFunc`, avec le paramètre `NULL`.

2 Animer l'hélice

Dans cet exercice, vous allez être amenés à écrire une fonction permettant d'animer l'hélice de votre « avion » . Cette animation se fera simplement en effectuant une rotation de l'hélice autour de son axe de symétrie, l'angle de la rotation variant successivement et cycliquement de 0° à 360°.

1. définir, dans le module `graphique.c` une variable réelle globale au module, nommée `angle_helice`, qui mémorisera l'angle de rotation courant de l'hélice. Cette variable sera initialisée à 0.0 ;
2. écrire, dans le module `graphique.c` la fonction suivante :

   ```
   void animer(void)
   ```

   Cette fonction se contentera pour le moment d'incrémenter la valeur de la variable `angle_helice` (en gérant le dépassement de la valeur 360) et de rappeler la fonction d'affichage ;
3. mettre à jour le fichier `graphique.h` ;
4. modifier la fonction d'affichage de sorte à y rajouter la rotation de l'hélice (prise en compte de la valeur de la variable `angle_helice`) ;
5. ajouter l'appel de la fonction `glutIdleFunc` dans la fonction `main` de votre application, son paramètre étant bien évidemment la fonction `animer` (à déclarer en `extern` ...).
6. compiler et tester cette nouvelle fonctionnalité.

3 Contrôle de l'hélice

Vous allez à présent ajouter la possibilité de mettre en marche ou d'arrêter l'hélice, via l'appui sur la touche `'h'`.

1. déclarer, dans un fichier nommé `defines.h` les deux constantes `ON` et `OFF` (de valeur respective 1 et 0 par exemple) ;
2. déclarer, dans le module `graphique.c` une variable nommée `helice_active`, initialisée à la valeur `OFF` ;
3. modifier la fonction `animer` de telle sorte que l'angle de rotation de l'hélice ne soit modifié qu'à la condition que la variable `helice_active` ait pour valeur `ON` ;
4. modifier la fonction de gestion du clavier de telle manière que la valeur de la variable `helice_active` change alternativement entre `ON` et `OFF` lors de l'appui sur la touche `'h'` ;
5. mettre à jour les liens entre les différents modules, compiler et tester l'application.

4 Animation des roues

Reprenez le déroulement des paragraphes **??** et **??** pour ajouter la possibilité d'animer et de contrôler (via la touche `'r'`) les roues de votre « avion » .

5 Animation de l'avion

Effectuer toutes les modifications nécessaires pour que l'avion entier se déplace en suivant une trajectoire circulaire, de rayon 10, autour d'un axe vertical dont l'origine se trouve en $(0, 0, -15)$ par rapport à l'observateur. Les fonctionnalités existantes, attachées aux différents boutons programmés dans ce Tp et les Tps précédents, doivent être conservées [1]. Cette nouvelle fonctionnalité devra pouvoir être activée ou désactivée par l'intermédiaire de la touche `'a'`.

1. Les transformations géométriques à appliquer pour gérer cette trajectoire circulaire sont relativement simples, mais peu intuitives ; il est donc fortement conseillé de faire des schémas papier, situant les positions successives du repère, pour obtenir la succession de transformations à rajouter ...