

Objectif: Révisions interactions. Nouvelle notion: navigation virtuelle.

Éléments fournis:

- Vidéo du programme à réaliser
- On utilisera la documentation en ligne pour réaliser le travail

L'objectif est d'étendre le programme précédent (sujet n°1) : îlot de 4 bâtiments voisins du centre du repère OpenGL, placés sur un sol en damier (plan x,z). La caméra est placée à la verticale du repère ($z>0$ et $y>0$, $x=0$) et dirigée vers son centre (fonction `gluLookAt()`). L'objectif de ce TD est le déplacement la caméra.

- 1) Déplacement caméra: permettre de déplacer la position de la caméra selon la direction (0,x) à l'aide des flèches gauche/droite du clavier (Rappel : utiliser `glutSpecialFunc()`).
- 2) Modifier le déplacement précédent pour décrire un cercle autour du centre (y constant). Indication: inspirez vous d'une représentation en coordonnées polaires (dans un plan parallèle à (0,x,z)). Remarque : pour accéder aux fonctions mathématiques, on utilisera le module `math` (« `from math import *` »), fournissant notamment les fonctions « `cos` », « `sin` », « `sqrt` ». Le carré d'un nombre « `n` » se calcule avec la syntaxe « `n**2` ».
- 3) Zoom: les flèches haut-bas doivent permettre de se rapprocher/s'éloigner du centre du repère. Le zoom doit pouvoir se combiner à la rotation.
- 4) Permettre de modifier l'altitude de la caméra (i.e. selon l'axe (0,y)), en utilisant la combinaison de touches `SHIFT+UP` et `SHIFT+DOWN`. On utilisera la fonction `glutGetModifiers()` qui retourne un entier permettant de savoir si une touche de contrôle est enfoncée (i.e. `Ctrl`, `Alt`, `Shift`), en plus de la touche flèche (`UP` ou `DOWN`).
- 5) Au moyen du menu, on doit pouvoir déclencher une rotation automatique de la caméra autour du centre du repère (un tour complet). Vérifier que l'on peut zoomer durant une rotation complète. Le tableau suivante donne un exemple d'animation.
- 6) La fonctionnalité précédente nécessite de connaître à l'avance le nombre d'appel (même si cela peut être contourné par des astuces de programmation). Si l'on souhaite avoir un mouvement de rotation en continue, il est préférable d'utiliser la fonction `glutIdleFunc()` de la librairie GLUT : « `glutIdleFunc(maFonction)` » pour exécuter « `maFonction` » en tâche de fond, « `glutIdleFunc(None)` » pour stopper une tâche de fond. Pour éviter un mouvement de rotation trop rapide (la fonction définie est exécutée en boucle dès que possible), garantisiez un temps écoulé suffisant entre deux rafraîchissements consécutifs. Pour cela, on utilisera un temps de référence écoulé (instruction du type « `time = glutGet(GLUT_ELAPSED_TIME)` ») depuis le lancement de l'application (i.e. depuis « `glutInit()` »).

```
#MaFonctionAnimation est invoquée en tâche de fond (non bloquant pour manipulation application), de manière réursive.
#"nbCall" fois avec une pause de 1000 ms entre chaque invocation dans cet exemple
def MaFonctionAnimation(nbCall):
    print "Animation fonction: il reste ", nbCall, " invocations\n"
    #Gère la répétition : re-invocation tant nbCall > 0
    if nbCall > 0:
        #On déclare que la fonction MaFonctionAnimation (de signature void fct(int value)) sera invoquée dans 1000 millisecondes.
        #Le paramètre passé sera nbCall-1
        glutTimerFunc(1000, MaFonctionAnimation, nbCall-1)
```

Exemple d'animation : déclenchement par appel de *MaFonctionAnimation(9)*