

Esercitazione n. 6

Realizzare una semplice libreria grafica basata sul polimorfismo (utilizzando OpenGL e la libreria *freeglut*).

Per l'implementazione fare riferimento alle dispense relative alle librerie grafiche basate su OpenGL, all'esempio di libreria grafica basata sul polimorfismo e all'esercitazione 4.

Il visualizzatore dovrà essere implementato mediante una classe `CMyViewer` che mette a disposizione i metodi per inizializzare la libreria GLUT e per effettuare il rendering a video (in questo caso delle forme geometriche). La funzione di rendering (`RenderScene()`) deve essere dichiarata *friend* sia per la classe `CMyViewer` sia per la classe astratta `Shape`. La classe astratta `Shape` dovrà avere, oltre agli attributi necessari alla gestione della lista di forme geometriche, gli attributi comuni a tutte le forme geometriche (per esempio, colore e traslazione).

Dovrà quindi essere possibile utilizzare la libreria grafica nel modo seguente:

```
int main(int argc, char* argv[])
{
    float trans1[3] = {1, 0, 0};
    float trans2[3] = {0, 1, 0};
    float col1[3] = {1, 0, 0};
    float col2[3] = {0, 1, 0};
    MyQuad obj1(col1,trans1,-1,1);
    MyTriangle obj2(col2,trans2,-1,1);
    CMyViewer view;
    view.show(argc,argv);
}
```

Nota: affinché alla chiusura della finestra di visualizzazione la funzione `glutMainLoop()` ritorni al chiamante e' necessario effettuare questa ulteriore inizializzazione:

```
glutSetOption(GLUT_ACTION_ON_WINDOW_CLOSE, GLUT_ACTION_GLUTMAINLOOP_RETURNS)
```

E' necessario settare correttamente le proprietà del progetto:

Configuration Properties -> C/C++ -> Additional Include Directories

Configuration Properties ->Linker->General-> Additional Library Directories

Configuration Properties ->Linker->Input-> Additional Dependencies (freeglut.lib, opengl32.lib, glu32.lib)