

Ćwiczenie IV - Biblioteka OpenGL - transformacje przestrzenne obiektów

Przygotowanie do ćwiczenia:

1. Zapoznać się z transformacjami przestrzennymi (obrót, przesunięcie, skalowanie),
2. Zapoznać się z opisem jednorodnym transformacji przestrzennych,
3. Zapoznać się z zestawem komend OpenGL umożliwiającym dokonywanie transformacji przestrzennych obiektów na scenie wraz z podstawowymi zasadami posługiwania się tymi komendami.

Przebieg ćwiczenia:

1. Założenia:
 - a. Celem prac na zajęciach laboratoryjnych jest wykonanie ruchomego modelu siatki manipulatora Puma.
 - b. Wynikiem prac na dzisiejszych zajęciach ma być program zbliżony w działaniu do programu „**puma_siatka.exe**” dostarczonego do materiałów laboratoryjnych.
 - c. Realizacja ćwiczenia polega na uzupełnieniu kodu programu „**gl_template**” modyfikowanego na ostatnich zajęciach.
 - d. W realizacji prac wzorować się należy na rozwiązaniach przyjętych w programie „**robot1**” również dołączonym do materiałów laboratoryjnych.
2. Uwagi do sposobu realizacji celu dzisiejszych zajęć laboratoryjnych:
 - a. Do poruszania ramionami robota można wykorzystać klawiaturę. Uzupełnienie obsługi komunikatu WM_KEYDOWN może wyglądać następująco:

```
case WM_KEYDOWN:
{
//...
    if(wParam == '1')
        d1-= 1.0f;

    if(wParam == '2')
        d1+= 1.0f;
//...
}
```

- b. Podczas umieszczania obiektów na scenie należy pamiętać, że znajdują się one w tym miejscu sceny i w takiej orientacji jak ustalony lokalny układ współrzędnych.
- c. Przykład wykorzystania złożenia transformacji do konstruowania prostego modelu manipulatora zawarto w programie „robot1”. Sekwencja instrukcji tworzących ruchomy model robota wygląda w tym programie następująco:

```
glPushMatrix();
glTranslatef (0.0, -0.6, 0.0);
auxWireBox(1.0, 0.2, 1.0);

glTranslatef (0.0, 0.6, 0.0);
```

```
glRotatef ((GLfloat) base, 0.0, 1.0, 0.0);  
auxWireBox(0.4, 1.0, 0.4);
```

```
glTranslatef (0.0, 0.5, 0.0);  
glRotatef ((GLfloat) elbow, 0.0, 0.0, 1.0);  
glTranslatef (0.0, 0.5, 0.0);  
auxWireBox(0.4, 1.0, 0.4);
```

```
glTranslatef (0.0, 0.5, 0.0);  
glRotatef ((GLfloat) wrist, 0.0, 0.0, 1.0);  
glTranslatef (0.0, 0.5, 0.0);  
auxWireBox(0.4, 1.0, 0.4);
```

```
glPopMatrix();
```

- d. Model robota należy zamknąć w jednej funkcji przyjmującej jako parametry wywołania kąty obrotu dla poszczególnych przegubów np.:

```
void puma(double d1, double d2, double d3)
```