

## 3D трансформации

**Задача 2:** Написване на код, който да представлява симулация на малка слънчева система, съставена от слънце, една планета, обикаляща около слънцето, заедно със своята луна, която пък се върти около нея.

Използвайте готови обекти от библиотеката GLUT.

Понеже кода няма да поддържа динамична светлина, за да не изглеждат триизмерните сфери като кръгове, използвайте линейно (Wire) изобразяване на обектите.

```
#include<windows.h> //Хедърният файл на Windows
#include<gl/glut.h> // Хедърният файл на GLUT
#include<gl/gl.h> // Хедърният файл на OpenGL

void Render(void);
void Resize(int width, int height);
void Menu(int value);

int MenuID; //променлива, в която ще съхраним иден. номер на нашето меню

//дефинираме променливите, които ще регулират завъртането на планетите
float Planet1_Angle;
float Planet2_Angle;
float Moon_Angle;

int main(int argc, char* argv [ ])
{
    glutInit(&argv, argv); //инициализираме библиотеката GLUT
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA); //определя режима на
    рендериране
    glutInitWindowSize(600, 600); //определя големината на прозореца
    glutInitWindowPosition(100, 100); //определя позицията на прозореца
    glutCreateWindow("Created with GLUT (OPENGL Utility Toolkit"); //създаваме
    прозореца
    glutDisplayFunc(Render); //определяме рендериращата ни функция

    //определяме нашата оразмеряваща функция, която с извиква при промяна размерите
    на прозореца
    glutReshapeFunc(Resize);

    glEnable(GL_DEPTH_TEST); //включваме проверка на разстоянията

    //създаваме нашето меню и указваме функция, която ще се извиква при избиране на
    елемент от менюто
    MenuID = glutCreateMenu(Menu);

    glutSetMenu(MenuID); //определяме менюто като текущо меню
```

```

glutAddMenuEntry("About", 1); //първи елемент от менюто

glutAddMenuEntry("Exit", 2); //втори елемент от менюто

glutAttachMenu(GLUT_RIGHT_BUTTON); //определяме бутона, чието натискане
определя менюто

glMatrixMode(GL_PROJECTION);

    gluPerspective(45, 1, 1, 200); //определяме 3D изгледа

glMatrixMode(GL_MODELVIEW); //превключваме обратно на MODELVIEW матрица

glutMainLoop( );

return 0;

}

void Render ( )

{

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); //изчистваме
буферите

    glLoadIdentity( ); //зареждаме първоначалната матрица

    //увеличаваме ъгъла на завъртане
    Planet1_Angle += 0.5;
    Planet2_Angle += 0.3;
    Moon_Angle +=0.5;

    //определяме позиция на камерата
    gluLookAt(0.0, 10.0, -50.0, 0.0, 0.0, 0.0, 1.0, 0.0);

    glColor3f(1.0, 1.0, 0.0); glutWireSphere(4.0, 10, 10); //създаваме слънцето

    glPushMatrix( ); //поставяме матрица в стека

    glRotatef(Planet1_Angle, 0.0, 1.0, 0.0); //завъртаме сцената

    glTranslatef(10.0, 0.0, 0.0); //преместване сцената с десет единици

    glColor3f(1.0, 0.0, 0.0); glutWireSphere(1.5, 10, 10); //изрисуваме първата планета

    glPopMatrix( ); //изваждаме матрицата от стека

    //за изрисуване на втората планета повтаряме същата процедура

```

```

glPushMatrix( );

glRotatef(Planet2_Angle, 0.0, 1.0, 0.0);

glTranslatef(-20.0, 0.0, 0.0);

glColor3f(0.0, 1.0, 0.0); glutWireSphere(2.0, 10, 10);

//сега идва ред на луната, която трябва да обикаля около планетата

glRotatef(Moon_Angle, 0.0, 1.0, 0.0);

glTranslatef(5.0, 0.0, 0.0);

glColor3f(0.0, 0.0, 1.0); glutWireSphere(1.0, 10, 10);

glPopMatrix( );

/*Забележете, че изрисуваме луната в матрицата на втората планета, защото искаме
първите две трансформации, преди изрисуването на втората планета, да се отразят и на
луната. Само така ще постигнем желанния резултат – планета, въртяща се около
слънцето и луна, въртяща се около планетата. */

glutSwapBuffers( ); //разменяме буферите

glutPostRedisplay( ); // изискваме следващо извикване на рендериращата ни функция
}

void Resize(int width, int height)

{

    if (height==0) height=1; //проверяваме стойността на height, за да избегнем деление
на 0

    //указваме на OpenGL, че искаме изрисуване върху целия прозорец
glViewport(0, 0, width, height);

glMatrixMode(GL_PROJECTION); //включваме Projection матрицата

glLoadIdentity( ); //зареждаме първоначалната Projection матрица

gluPerspective(45, (GLdouble)width/height, 1, 200); //определяме новия 3D изглед

glMatrixMode(GL_Modelview); //превключваме отново към Modelview матрица

glLoadIdentity( );

}

```

```
void Menu(int value)

{

    int result;

    //обработваме подаденото на нашата функция съобщение (номера на избрания
    елемент на менюто)
    switch(value)

    {

        // ако е избран първия елемент от менюто се показва прозорец със съобщение
        case 1: MessageBox(NULL, "Created with GLUT (OpenGL Utility Toolkit)", "About",
        MB_OK);
            break;

        //ако е избран втория елемент от менюто (Exit) се показва прозорец с въпрос, дали
        искаме да затворим програмата
        case 2: result = MessageBox(NULL, "Do you really want to quit?", "Quit?",
        MB_YESNO);
            if (result == IDYES) exit (0);
            break;

    }

}
```