

Sprawozdanie Lab 3

Wojciech Kosierkiewicz 272926

26 listopada 2024

1 Wstęp

Podczas tych laboratoriów uczyłem się operowania wirtualną kamerą w środowisku 3D. W zadaniach musiałem ustawiać kamerę w różnych miejscach i pod różnymi kątami. Do sterowania jej używałem danych z ruchu myszką i sygnałów klawiatury.

2 Zadania

2.1 zadanie 1

W miałem przestudiować obrót kamerą w poziomie i dodać możliwość obrotu w pionie

```
def render(time):
    global theta
    global phi

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()

    gluLookAt(viewer[0], viewer[1], viewer[2],
              0.0, 0.0, 0.0, 0.0, 1.0, 0.0)

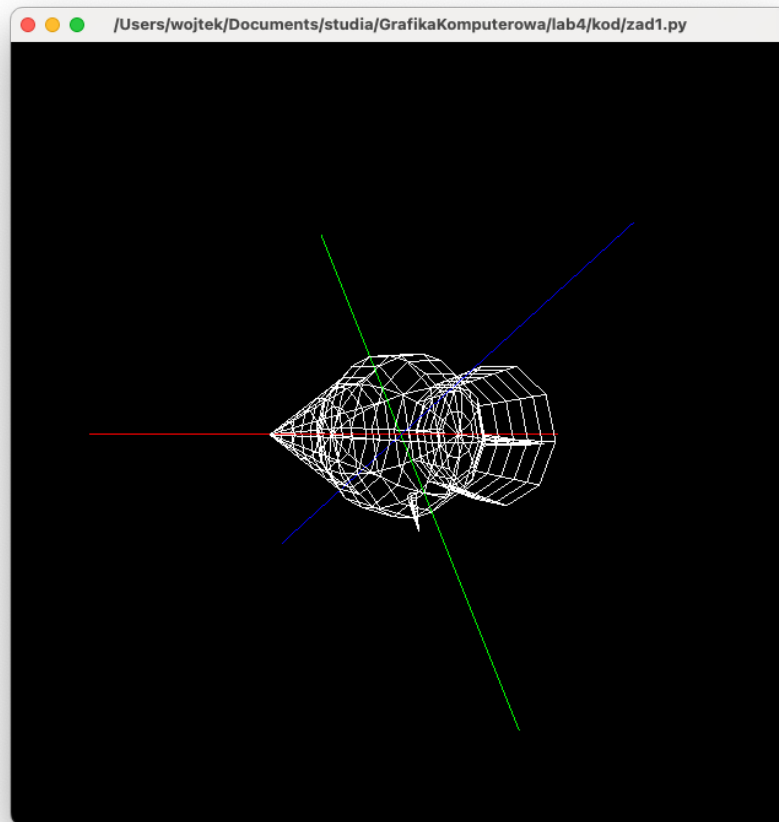
    if left_mouse_button_pressed:
        theta += delta_x * pix2angle
        phi += delta_y * pix2angle

    glRotatef(theta, 0.0, 1.0, 0.0)
    glRotatef(phi, 1.0, 0.0, 0.0)

    axes()
    example_object()

    glFlush()
```

Rysunek 1: Kod obsługujący kamery wokół obiektu obiektu



Rysunek 2: Wynik zadania 1

2.2 zad2

W następnym zadaniu miałem jeszcze dodać opcje skalowania skalowania naszego obiektu po przyciśnięciu guzika myszy. Do wykonania tego zmienialem albo θ i ϕ na podstawie myszki lub $scale$ zależnie od naciśniętego guzika. Dodałem także wywołanie `glScalef` które było odpowiedzialne za odpowiednie przeskalowanie.

```

def render(time):
    global theta
    global phi
    global scale
    global scale

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()

    gluLookAt(viewer[0], viewer[1], viewer[2],
               0.0, 0.0, 0.0, 0.0, 1.0, 0.0)

    if left_mouse_button_pressed:
        theta += delta_x * pix2angle
        phi += delta_y * pix2angle

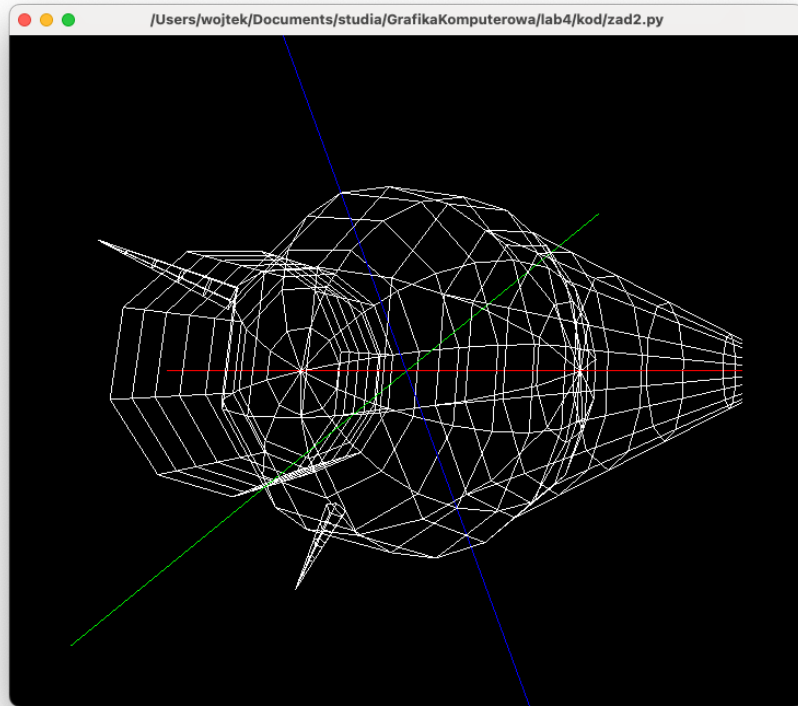
    if right_mouse_button_pressed:
        scale += delta_y * 0.01
        scale += delta_x * 0.01

    glRotatef(theta, 0.0, 1.0, 0.0)
    glRotatef(phi, 1.0, 0.0, 0.0)
    glScalef(scale, scale, scale)

    axes()
    example_object()

    glFlush()

```



Rysunek 3: Wynik zadania 2

2.3 zad3

W zadaniu 3 musiałem wykonać pełny obrót kamery wokół obiektu przy obliczaniu x_{eye} y_{eye} i z_{eye}

```

def render(time):
    global theta
    global phi
    global scale
    global r

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()

    if left_mouse_button_pressed:
        theta += delta_x * pix2angle
        phi += delta_y * pix2angle

    if right_mouse_button_pressed:
        r += delta_y * pix2angle
        r += delta_x * pix2angle

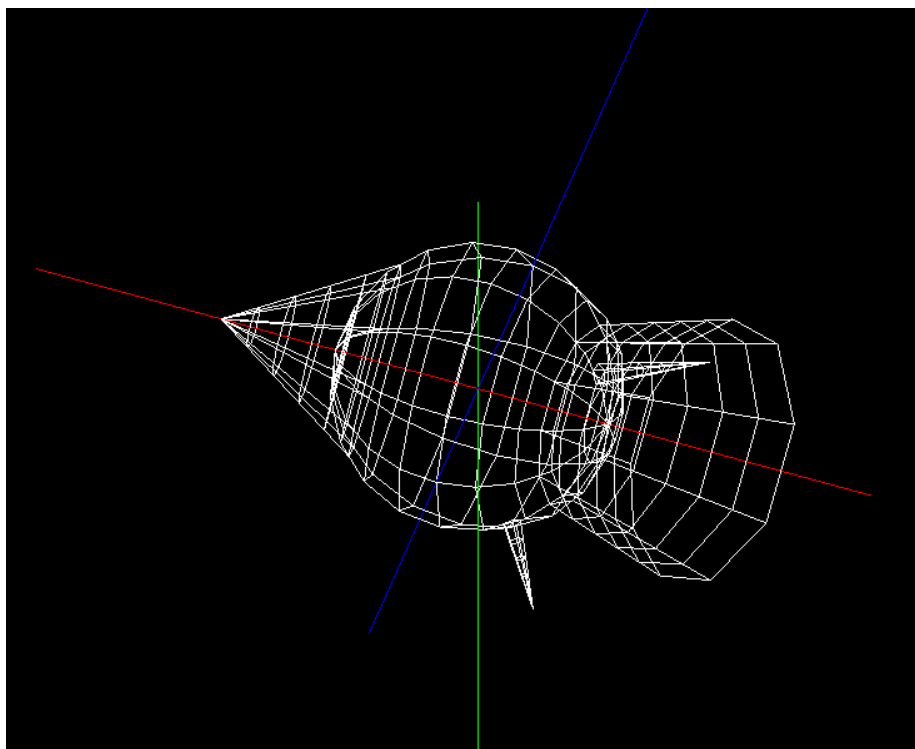
    xeye=r*sin(theta*pi/180)*cos(phi*pi/180)
    yeye=r*sin(phi*pi/180)
    zeye=r*cos(theta*pi/180)*cos(phi*pi/180)

    gluLookAt(xeye, yeye, zeye, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0)

    axes()
    example_object()

    glFlush()

```



Rysunek 4: Wynik zadania 3

2.4 zad4

W zadaniu 4 musiałem dodać limity obrotu dzięki którym nie możliwe było obrócenie przedmiotu ponad zakres logiczny. dlatego dla osi poziomej ustawiłem maksymalny kąt 360 stopni a dla osi pionowej obrót o -90° i 90° .

```

def render(time):
    global theta
    global phi
    global thetas
    global phis
    global scale
    global r
    global xeye, yeye, zeye
    global objclrx, objclry, objclrz

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    if (moving == "camera"):
        objclrx = 1.0
        objclry = 1.0
        objclrz = 1.0
    else:
        objclrx = 0.0
        objclry = 1.0
        objclrz = 1.0
    glLoadIdentity()

    if (moving == "camera"):
        xeye=r*sin(theta*pi/180)*cos(phi*pi/180)
        yeye=r*sin(phi*pi/180)
        zeye=r*cos(theta*pi/180)*cos(phi*pi/180)
        if left_mouse_button_pressed:
            theta += delta_x * pix2angle
            phi += delta_y * pix2angle
            phi = max(-90.0, min(90.0, phi))
            theta = fmod(theta, 360.0)
        if right_mouse_button_pressed:
            r += delta_y * pix2angle
            r += delta_x * pix2angle
            r = max(1.0, abs(r))
            r = min(10.0, abs(r))
    else:
        if left_mouse_button_pressed:
            thetas += delta_xs * pix2angle
            phis += delta_ys * pix2angle

        if right_mouse_button_pressed:
            scale += delta_ys * pix2angle/100
            scale += delta_xs * pix2angle/100

    gluLookAt(xeye, yeye, zeye, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0)
    glRotatef(thetas, 0.0, 1.0, 0.0)
    glRotatef(phis, 1.0, 0.0, 0.0)
    glScalef(scale, scale, scale)
    8

    axes()
    example_object()

    glFlush()

```