

Politechnika Wrocławska	Autor: Piotr Józefek 272311	Wydział Informatyki i Telekomunikacji Rok akademicki: 3
Grafika komputerowa i komunikacja człowiek – komputer		
Data ćwiczenia: 12.11.2024	Temat ćwiczenia laboratoryjnego: Interakcja z użytkownikiem, transformacje wierzchołków	Ocena:
Nr ćwiczenia: 4		Prowadzący: dr inż. arch. Tomasz Zamojski

1. Wstęp

Interakcja użytkownika z wirtualnym światem w aplikacjach 3D opiera się na obsłudze zdarzeń z klawiatury i myszy, które umożliwiają manipulację widokiem, obiektami oraz kamerą. Kluczową rolę w tym procesie odgrywają przekształcenia geometryczne, które są podstawą modelowania, animacji oraz renderowania scen. Przekształcenia dzielą się na afiniczne i nieafiniczne. Przekształcenia afiniczne, takie jak identyczność, skalowanie, translacja i obrót, zachowują liniowość oraz równoległość linii w przestrzeni. Z kolei nieafiniczne, takie jak rzutowanie ortogonalne i perspektywiczne, wprowadzają zmiany perspektywy, które odpowiadają za realistyczne odwzorowanie głębi sceny. Do opisania tych przekształceń stosuje się współrzędne jednorodne oraz macierze przekształceń, które umożliwiają ich reprezentację matematyczną i łatwe składanie. W kontekście wirtualnej sceny istotne są również pojęcia kamery, obserwatora oraz względności ruchu. Kamera w aplikacji 3D definiowana jest przez macierze patrzenia i rzutowania. Macierz patrzenia określa pozycję oraz orientację obserwatora w scenie, natomiast macierz rzutowania opisuje sposób odwzorowania przestrzeni 3D na płaszczyznę 2D, co jest kluczowe dla renderowania. Składanie przekształceń, praca z macierzami oraz konfiguracja kamery w środowisku Legacy OpenGL stanowią podstawowe elementy implementacji scen 3D. Operacje te obejmują manipulację macierzami skalowania, translacji i obrotu oraz ich kompozycję w celu uzyskania pożądanych efektów wizualnych. Zrozumienie rzutowania perspektywicznego i prostopadłego umożliwia precyzyjne dostosowanie parametrów kamery do potrzeb aplikacji i wymagań wizualizacji.

2. Cel ćwiczenia

- Zapoznanie się z mechanizmem obsługi urządzeń peryferyjnych.
- Zrozumienie zasady działania przekształceń wierzchołków.
- Implementacja transformacji w reakcji na działania użytkownika.

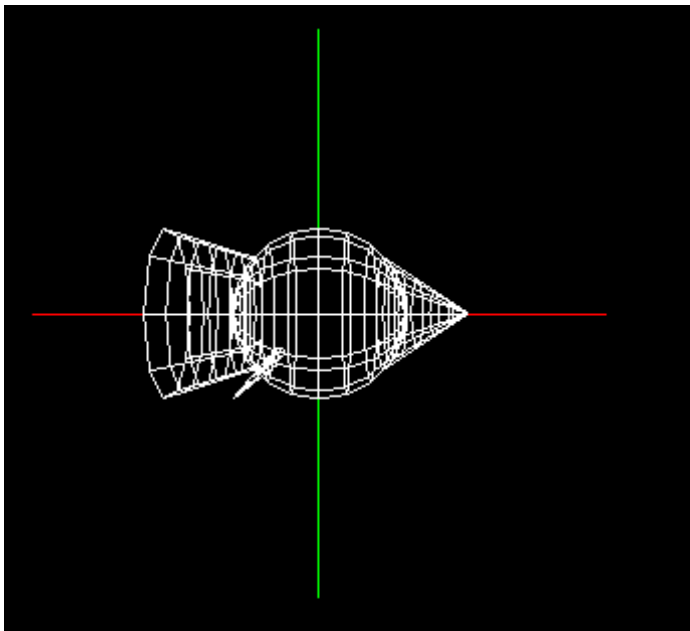
3. Wykonane zadania

Udało się zrealizować wszystkie zadania. W zadaniu na 5.0 połączyłem trzy tryby użytkownika. Domyślnie działa tryb ruchomej kamer, gdzie lewy przycisk obraca kamerę wokół osi X, a prawy wokół Y. Po naciśnięciu spacji przestawiamy się na tryb Obracania Obiektu, gdzie lewym przyciskiem myszy można przybliżyć lub oddalić obiekt, a prawym obrócić obiekt. Po naciśnięciu entera przełączymy się na tryb pierwszoosobowy gdzie poruszamy się przyciskami WASD, a prawym przyciskiem obracamy kamerę z pierwszej osoby.

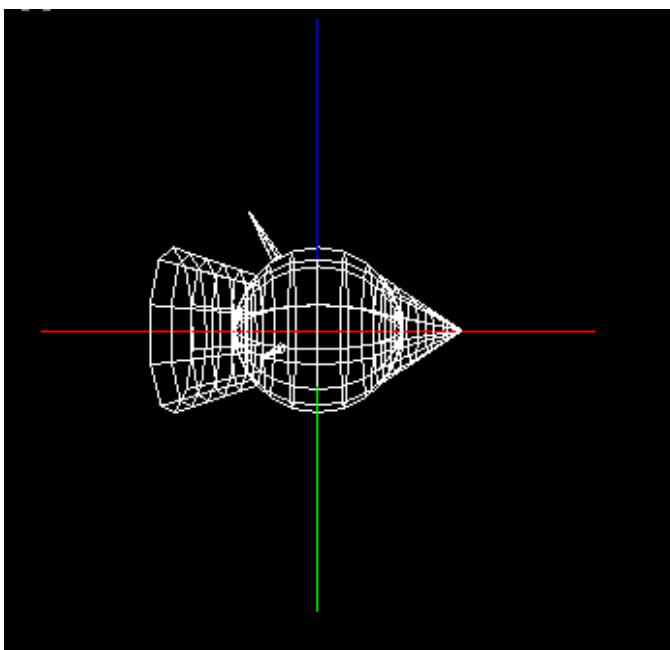
4. Prezentacja i omówienie funkcjonalności

4.1. Obracanie się wokół osi X

Kod został zmieniony, aby umożliwić obrót obiektu w dwóch osiach na podstawie ruchu myszy. Ruch w osi X kontroluje obrót poziomy (θ). Różnice w położeniu myszy (Δx) są śledzone, a obroty obiektu realizowane za pomocą funkcji `glRotatef` w odpowiednich osiach. Zamiana wywołania `glRotatef(theta, 0.0, 1.0, 0.0)` na `glRotatef(phi, 1.0, 0.0, 0.0)` zmienia obrót obiektu z poziomego wokół osi Y na pionowy wokół osi X.



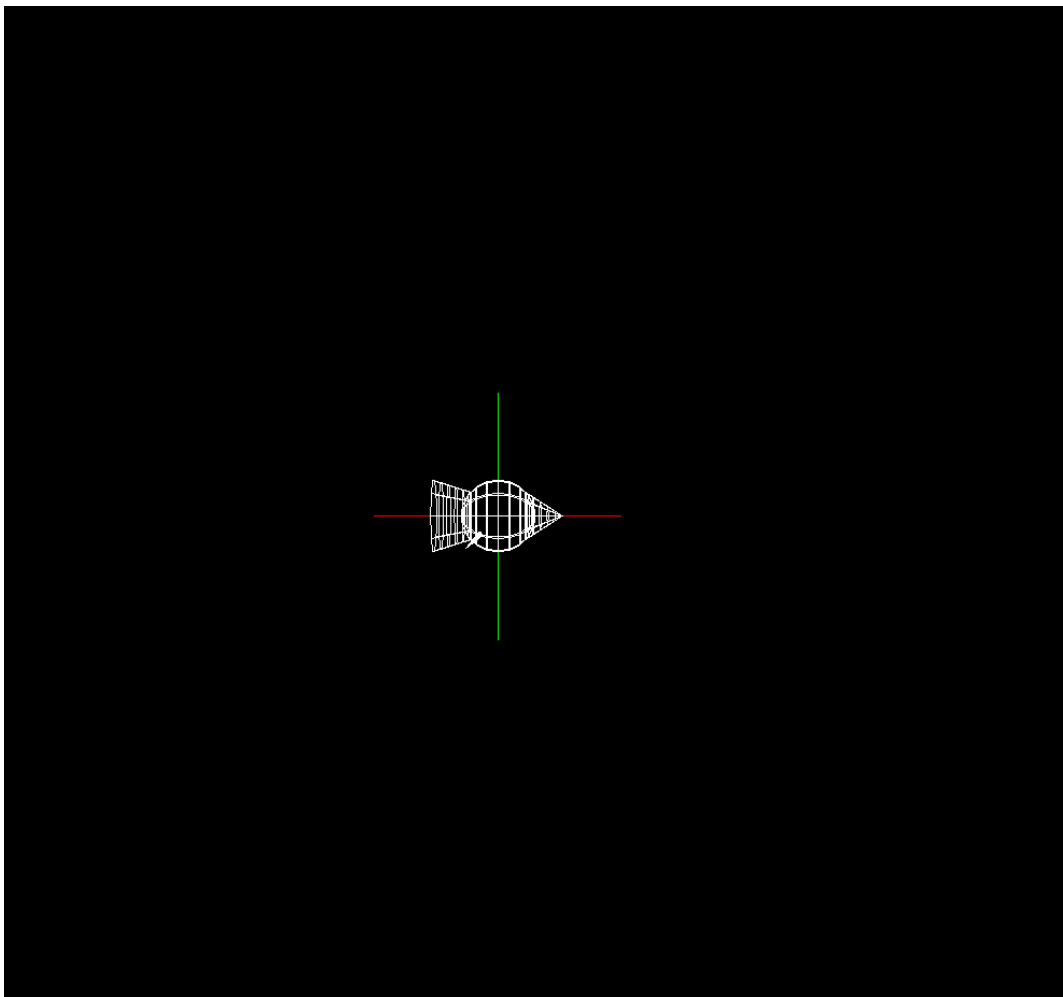
Rys 1 Oryginalny obiekt



Rys 2 Obiekt obrócony w osi X

4.2. Obsługa drugiego przycisku myszy do skalowania

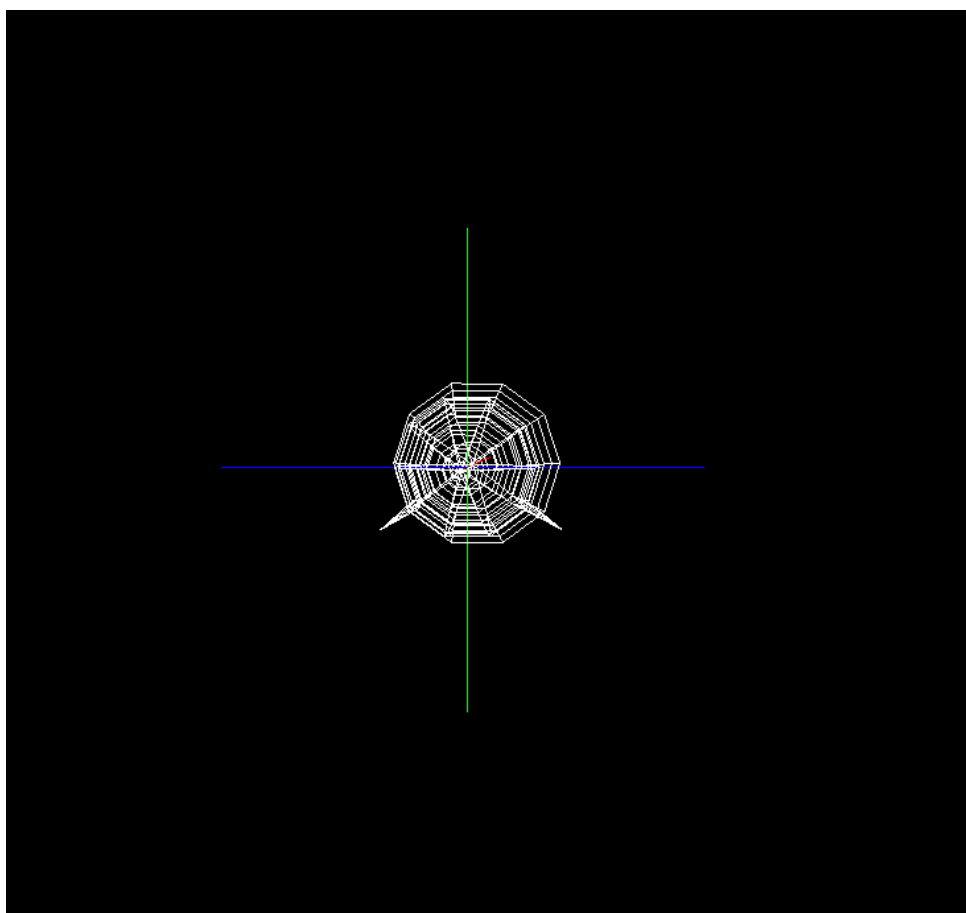
Celem tego zadania było dodanie funkcji skalowania obiektu za pomocą drugiego przycisku myszy. W tym celu wprowadzono zmienną `scale`, która kontroluje stopień skalowania. W przypadku naciśnięcia prawego przycisku myszy wartość zmiennej `scale` jest aktualizowana na podstawie ruchu kursora pionowo (`delta_y`) według wzoru $scale += delta_y * 0.01$. Następnie funkcja `glScalef(scale, scale, scale)` stosuje obliczoną wartość zmiennej `scale` do przeskalowania obiektu w trzech wymiarach.



Rys 3 Oddalony obiekt

4.3. Poruszanie kamerą wokół modelu

Celem zadania było umożliwienie sterowanie kamerą w taki sposób, aby obracała się wokół obiektu znajdującego się w centrum sceny 3D. Początkowo funkcja `gluLookAt` ustawia widok kamery skierowanej na punkt $(0, 0, 0)$, definiując podstawowe parametry układu współrzędnych i kierunek patrzenia. Pozycja kamery w przestrzeni jest następnie obliczana w oparciu o współrzędne sferyczne, gdzie promień R określa odległość od obiektu, a kąty θ i ϕ wyznaczają kąt poziomy i kąt pionowy. Obliczone współrzędne (x_{eye} , y_{eye} , z_{eye}) są wykorzystywane przez kolejne wywołanie `gluLookAt`, aby precyzyjnie określić aktualne położenie kamery w zależności od jej orientacji. W przypadku wciśnięcia prawego przycisku myszy wartości kątów θ i ϕ są dynamicznie zmieniane na podstawie ruchu myszy, co pozwala na płynne obracanie kamery wokół obiektu w czasie rzeczywistym.

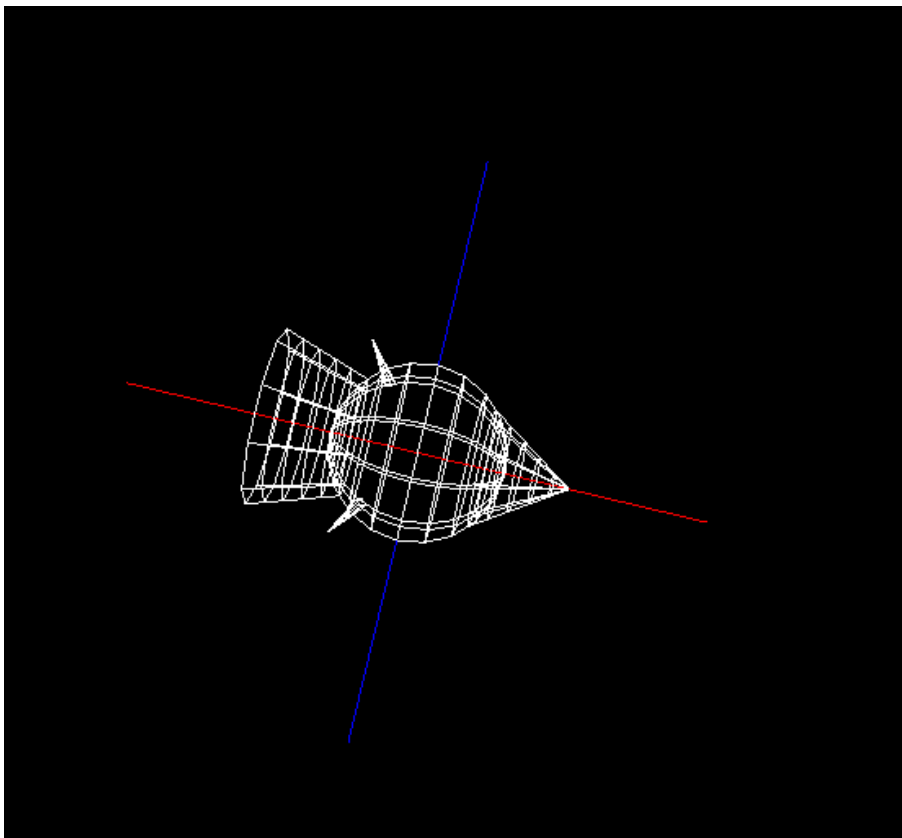


Rys 4 Obracanie kamerą wokół obiektu

4.4. Usprawnione poruszanie kamerą

Ulepszony kod obracania kamery pozwala na bardziej płynne i precyzyjne sterowanie widokiem 3D w scenie. Pozycja kamery jest dynamicznie modyfikowana w odpowiedzi na interakcję użytkownika. W przypadku naciśnięcia lewego przycisku myszy, użytkownik może zmieniać odległość kamery od obiektu poprzez manipulację wartością R . Zmienna R jest aktualizowana w zależności od ruchu myszy w osi pionowej, a ograniczenia zapewniają, że odległość kamery mieści się w zadanym zakresie (od 2.0 do 50.0).

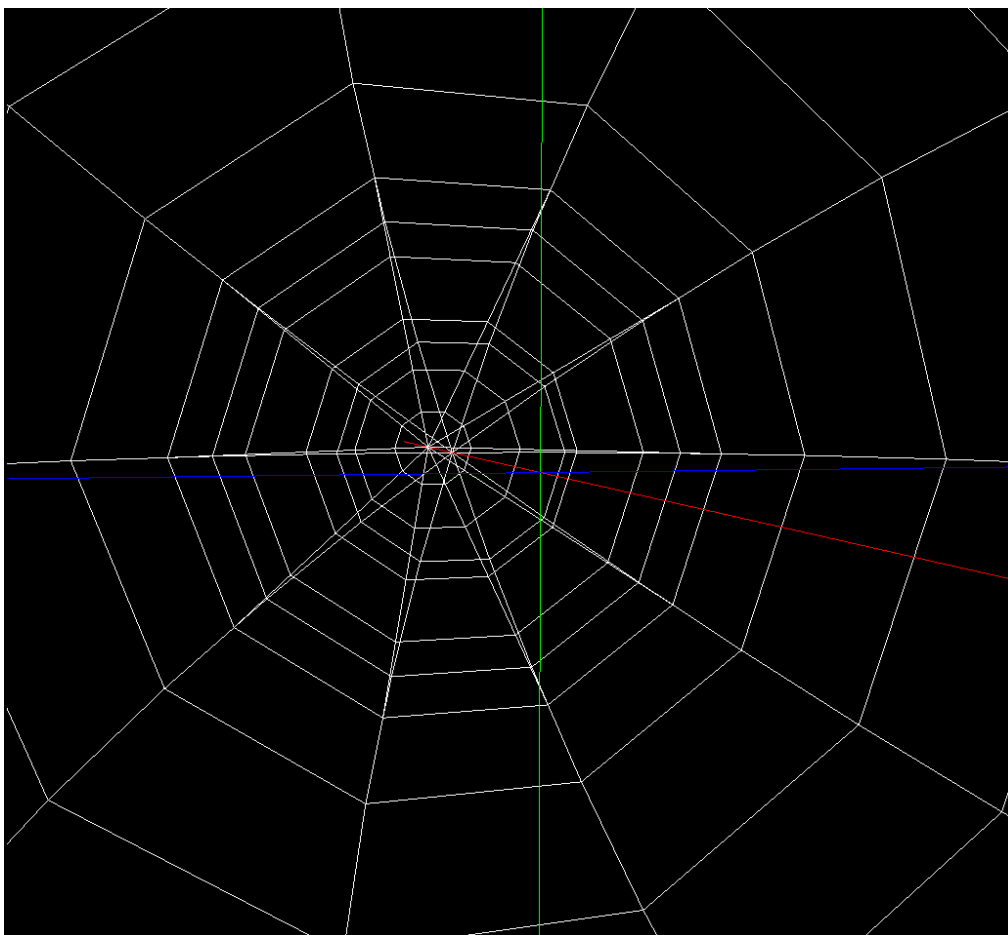
W przypadku wciśnięcia prawego przycisku myszy, użytkownik może obracać kamerę wokół obiektu. Kąt poziomy (θ) i kąt pionowy (ϕ) są zmieniane na podstawie ruchów myszy w poziomie i pionie. Zmienność kąta ϕ jest ograniczona, aby zapobiec przechyleniu kamery poza wartości 90° i -90° , co zapobiega niewłaściwemu ustawieniu widoku.



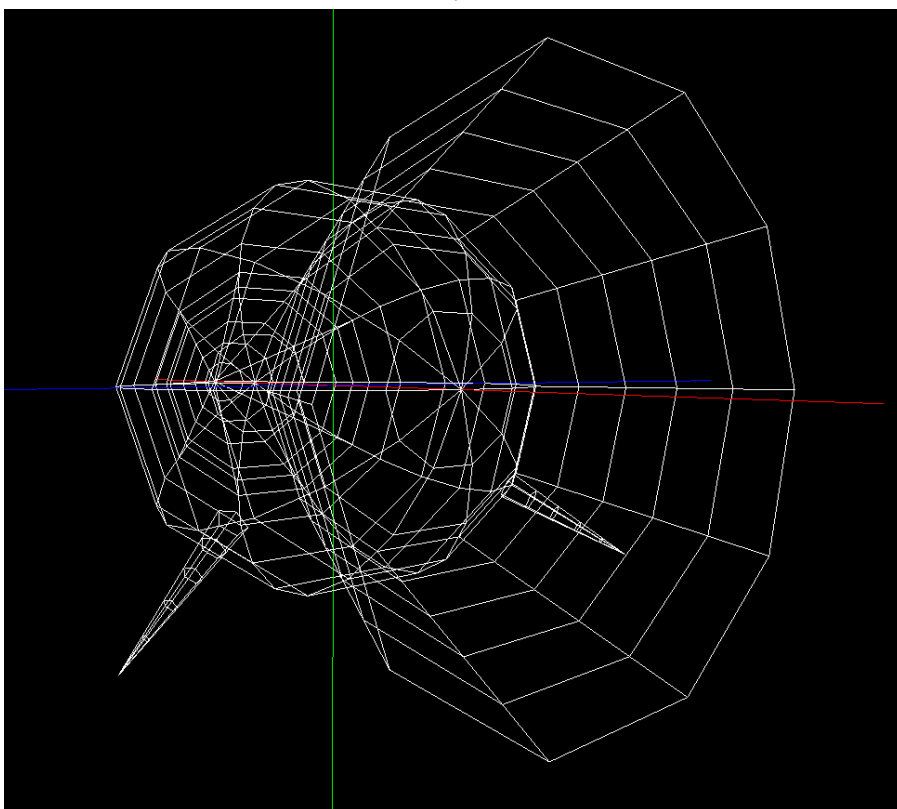
Rys 5 Obracanie kamerą wokół obiektu z ograniczeniem obrotu kamery

4.5. Pierwszoosobowy ruch swobodny z obrotem kamery

Kod implementuje obsługę kamery i interakcji użytkownika, umożliwiając poruszanie się w różnych trybach sterowania widokiem oraz obiektem. W trybie first-person użytkownik może sterować kamerą tak, jakby znajdował się wewnątrz wirtualnego świata. Obrót kamery w tym trybie odbywa się tylko wtedy, gdy wciśnięty jest prawy przycisk myszy. Zmienna theta odpowiada za obrót kamery w poziomie, natomiast zmienna phi kontroluje obrót w pionie. Aby zapobiec skokom kamery, wartość phi została ograniczona do zakresu od -90° do 90° , co uniemożliwia przewracanie widoku. Kierunek patrzenia kamery obliczany jest na podstawie zmiennych theta i phi, tworząc wektor `look_direction`, który opisuje aktualne położenie kamery względem obiektu. Dodatkowo wyliczany jest wektor „prawo-lewo”, który pozwala na przesunięcia w poziomie niezależnie od aktualnego kąta patrzenia. Za pomocą klawiszy W i S użytkownik może poruszać się w przód i w tył wzdłuż kierunku patrzenia kamery, natomiast klawisze A i D umożliwiają przesunięcia w lewo lub w prawo w płaszczyźnie poziomej. W trybie obracania obiektu kamera pozostaje nieruchoma, a użytkownik może manipulować pozycją obiektu w przestrzeni. Jeśli wciśnięty jest lewy przycisk myszy, zmienna phi modyfikuje obrót obiektu wzdłuż osi X. W przypadku prawego przycisku myszy zmienna theta odpowiada za obrót wzdłuż osi Y. Te zmiany są następnie przekazywane do funkcji `glRotatef`, która obraca obiekt wokół odpowiednich osi zgodnie z wartościami zmiennych phi i theta. Gdy użytkownik nie wybierze żadnego z powyższych trybów, aktywowany jest tryb kamery orbitującej. W tym trybie kamera obraca się wokół środka sceny, pozostając w stałej odległości od obiektu. Pozycja kamery jest określana za pomocą współrzędnych sferycznych, które są przekształcane na współrzędne kartezjańskie z użyciem zmiennych theta, phi i promienia R. Użytkownik może zmieniać wartość R (odległość kamery od obiektu) za pomocą ruchu myszy w osi pionowej, przy czym minimalna odległość wynosi 2.0 jednostki, a maksymalna 50.0 jednostek. Dodatkowo, prawy przycisk myszy pozwala obracać kamerę wzdłuż osi Y (theta) oraz osi X (phi), z ograniczeniem kąta phi do zakresu od -90° do 90° . Kod wykorzystuje również funkcję obsługi klawiatury, która pozwala przełączać między trybem obracania obiektu (`move_object_mode`) a trybem first-person (`first_person_mode`) za pomocą klawiszy spacji i Enter. Klawisze W, S, A i D są wykorzystywane do przesuwania kamery w przestrzeni w trybie first-person. Dzięki temu rozwiązaniu użytkownik ma pełną kontrolę zarówno nad widokiem, jak i obiektem, co umożliwia precyzyjną manipulację sceną w aplikacji 3D.



Rys 6 Widok pierwszoosobowy wewnątrz obiektu.



Rys 7 Widok pierwszoosobowy od tyłu obiektu.

5. Wnioski

- Udało się zrealizować wszystkie zadania
- W zadaniu 5 udało się połączyć poprzednie podpunkty dzięki wykorzystaniu obsługi klawiatury i wykorzystaniu jej przycisków do zmiany trybów

6. Literatura

- [Szymon Datko \[GK\] Lab4 :: Transformacje wierzchołków](#)