

Wojskowa Akademia Techniczna im. Jarosława Dąbrowskiego



Wydział Cybernetyki, kierunek informatyka - inżynieria systemów

Laboratorium z przedmiotu:
Grafika komputerowa

Sprawozdanie z ćwiczenia laboratoryjnego nr 3: **Przekształcanie geometryczne**

Prowadzący:
dr inż. Marek Salamon

Wykonał: Radosław Relidzyński, **Grupa:** WCY20IY4S1, nr 17

Data laboratoriów: 7.12.2022 r.

Termin oddania ćwiczenia: 20.12.2022 r.

Spis treści

A. Treść zadania	2
B. Sposób rozwiązania zadań.....	3
C. Prezentacja działania programu – układ planet.....	4
D. Prezentacja działania programu – BWP	7
E. Wnioski	9

A. Treść zadania

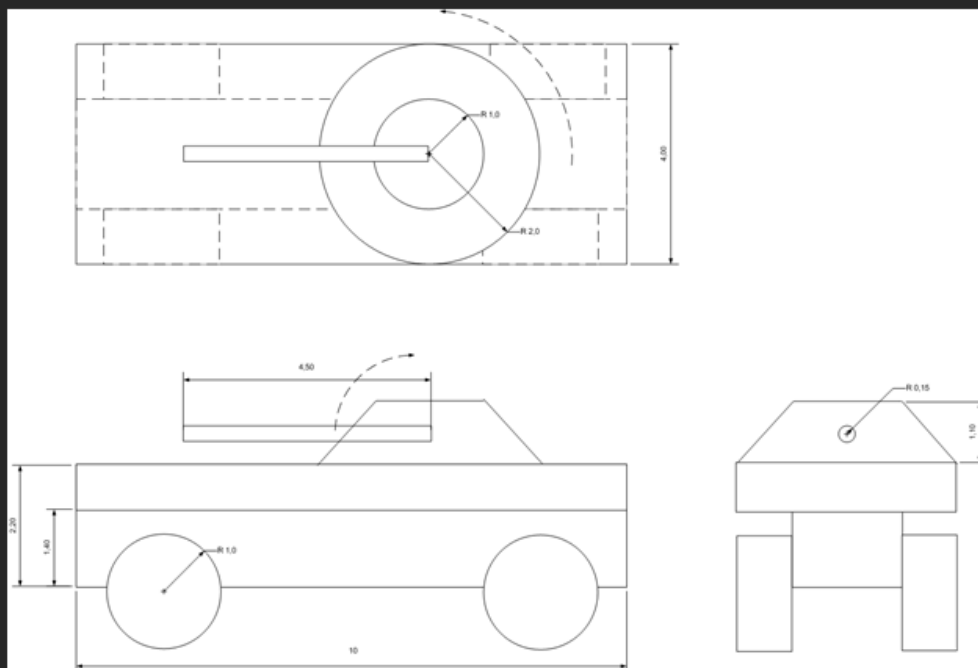
Zadanie laboratoryjne – zestaw 5b

1. Wykorzystać projekt „robot” napisać fragment programu odpowiedzialny za:
 - a) zmianę odległości obserwatora od obiektu w zakresie ($odlmin$, $odlmax$); 0.5 pkt.
 - b) zmianę orientacji obserwatora (bez ograniczeń) względem osi OX , OY i OZ . 0.5 pkt.
 2. Przesunąć obiekt z projektu „robot” w położenie (0.0, 60.0, 0.0) i wykorzystując funkcję **glutSolidCube** oraz funkcje działające na stosie macierzy modelowania napisać program odpowiedzialny za utworzenie obiektu z Zad.1 (ćw.). 1 pkt.
 3. Przesunąć obiekt z Zad.1 w położenie (0.0, 80.0, 0.0) i wykorzystując funkcję **glutWireSphere** napisać program przedstawiający układ planetarny zgodnie z parametrami przedstawionymi w Zad.2 (ćw.). 2 pkt.
Układ uzupełnić o:
 - a) planetę o promieniu 6 poruszającą się wokół słońca po orbicie o promieniu 60 z prędkością kątową 0.15 stopnia/klatkę animacji w kierunku CCW w płaszczyźnie nachylonej do osi OX pod kątem 30 stopni
 - b) planetę o promieniu 4 poruszającą się wokół słońca po orbicie o promieniu 80 z prędkością kątową 0.25 stopnia/klatkę animacji w kierunku CW w płaszczyźnie nachylonej do osi OX pod kątem -15 stopni;
a) 0.5 pkt., b) 0.5pkt.
- Zaznaczyć orbity poruszających się obiektów. Bieguny generowanych sfer powinny leżeć w osi pionowej. 1 pkt.
4. Napisać program przedstawiający obiekt zbudowany z prymitywów przestrzennych udostępnianych przez biblioteki **GLU** i **GLUT**. /zadanie indywidualne oceniane na podstawie sprawozdania/ 4 pkt.

Zadanie 1

Napisać program przedstawiający obiekt zbudowany z prymitywów przestrzennych udostępnianych przez biblioteki *GLU* i *GLUT*. Użytkownik za pomocą klawiatury powinien mieć możliwość wprowadzania zmian następujących parametrów:

1. Kąta obrotu wieży,
2. Kąta podniesienia armaty,



W programie uwzględnić możliwość interakcyjnej zmiany położenia obserwatora poprzez podanie następujących parametrów:

1. Odległości obserwatora od obiektu.
2. Kątą obrotu wokół obiektu w zakresie $[0, 360]$.

UWAGA: Obserwator jest zawsze zwrócony przodem w kierunku obiektu.

B. Sposób rozwiązania zadań

1. Program wykrywa czy naciśnięty jest przycisk „-” lub „+” i odpowiednio zmniejsza lub zwiększa wartość zmiennej „odl”, a następnie stosuje ją do programu poprzez polecenie „glTranslatef(0, 0, odl);”. Zmiana orientacji działa tak samo, przy pomocy funkcji „glRotatef()”. Dla konkretnych osi będzie to wartość rotacji oraz jeden z 3 pozostałych argumentów na wartość prawda (po kolei x, y, z).

Manipulacja odległością:

```
case '+': // przybliżanie
    odl = (odl < odlmin) ? odl + 1.0 : odl;
    break;
case '-': // oddalanie
    odl = (odl > odlmax) ? odl - 1.0 : odl;
    break;
```

Wprowadzanie odległości do matrycy:

```
glTranslatef(0, 0, odl);
```

2. Przesunięcie wykonane jest przy pomocy funkcji „glTranslatef(0.0, 80.0, 0.0);”.

```
// Przesuwanie wiezy
glTranslatef(0.0, 60.0, 0);
// Rysowanie podstawy ramienia (cylinder bez dolnej podstawy)
glPushMatrix();
// - sciany boczne
glRotatef(-90.0, 1, 0, 0);
gluCylinder(podstawaSciany, 3.0, 3.0, 1.0, 20, 4);
```

3. Planetę utworzyłem przy pomocy funkcji „glutWireSphere(1.0, 20.0, 20.0);”, jej rozmiar przy pomocy „glScalef(4.0, 4.0, 4.0);”, a jej położenie przez obrót względem słońca. Księżyc tak samo, z tym że wokół krążącej planety. Obrót dzieje się w pętli,

Zmiany odbywają się przy pomocy tych zmian wartości (wartość dodawana/odejmowana warunkuje prędkość poruszania się obiektu):

```
//korekcja zmiennych w pętli RysujScene
rotPlan1 -= 0.25;
rotPlan2 += 0.15;
rotPlan3 -= 0.25;
obrPlan += 1;
rotKsiezyc += 0.5;
```

Rysowanie jednej z planet:

```
glColor3f(0.0, 1.0, 1.0);
glRotatef(rotPlan1 * spowolnienie, 0, 1, 0);
glTranslatef(0.0, 0.0, 20.0);

glPushMatrix();
glRotatef(obrPlan, 0, 1, 0);
glRotatef(-90.0, 1, 0, 0);
glScalef(2.0, 2.0, 2.0);
glutWireSphere(1.0, 20.0, 20.0);
gluDisk(orbita, 2.5, 2.5, 40, 1);
glPopMatrix();
```

4. Wyznaczone zadanie: BWP. Na początku narysowałem każdy obiekt (kolejno ustawiłem kolor, położenie, obrót każdej składowej wozu, aktualizacja matrycy). Następnie wprowadziłem koordynaty oraz możliwość przybliżania/oddalania obiektu oraz konfigurację obrotu wieży i kąta podniesienia armaty (ten sam mechanizm co w przypadku obrotu widokiem w układzie planet, tym razem odbywa się przez przyciski: armata – w, s, wieża: a, d).

Zmiana położenia wieży i armaty:

```
switch (klawisz)
{
case '+':
    Range -= 1;
    break;

case '-':
    Range += 1;
    break;

case 'd':
    obrotWiezyczki += dObrotWiezyczki;
    break;

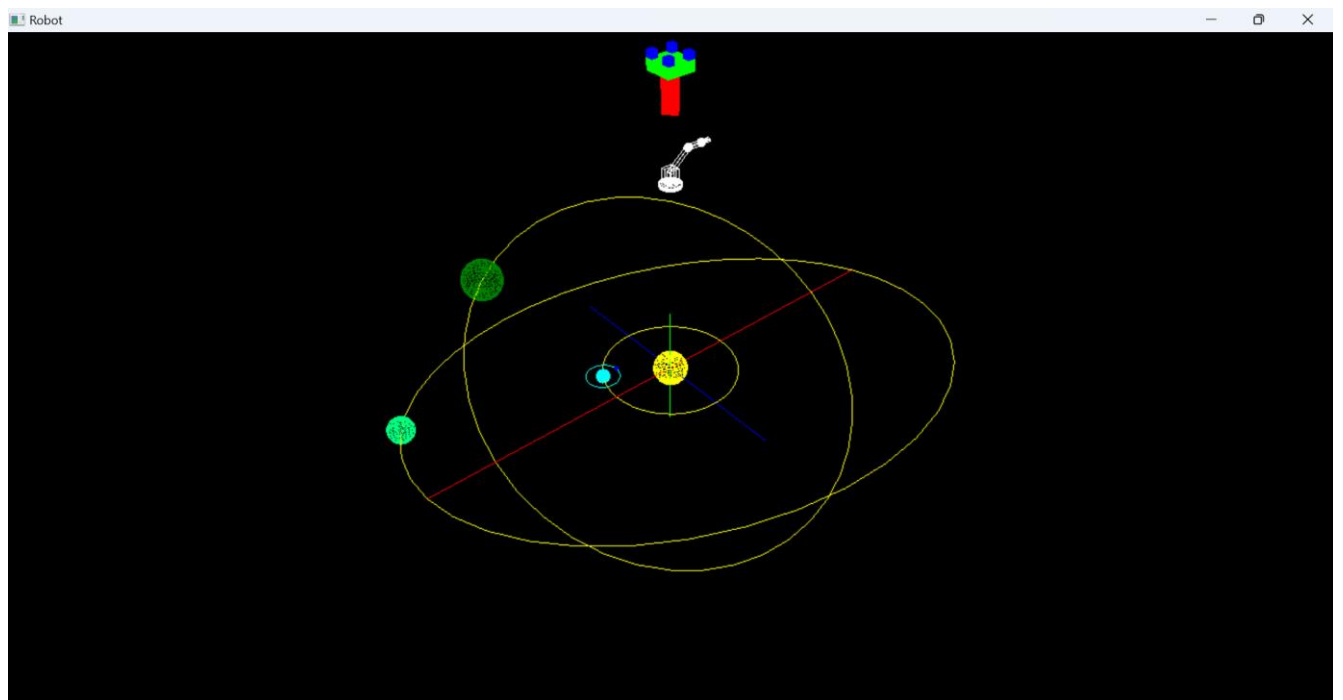
case 'a':
    obrotWiezyczki -= dObrotWiezyczki;
    break;
}
```

Tworzenie wieży i lufy, zaznaczone jest miejsce wprowadzania obrotu wieży i kąta nachylenia lufy

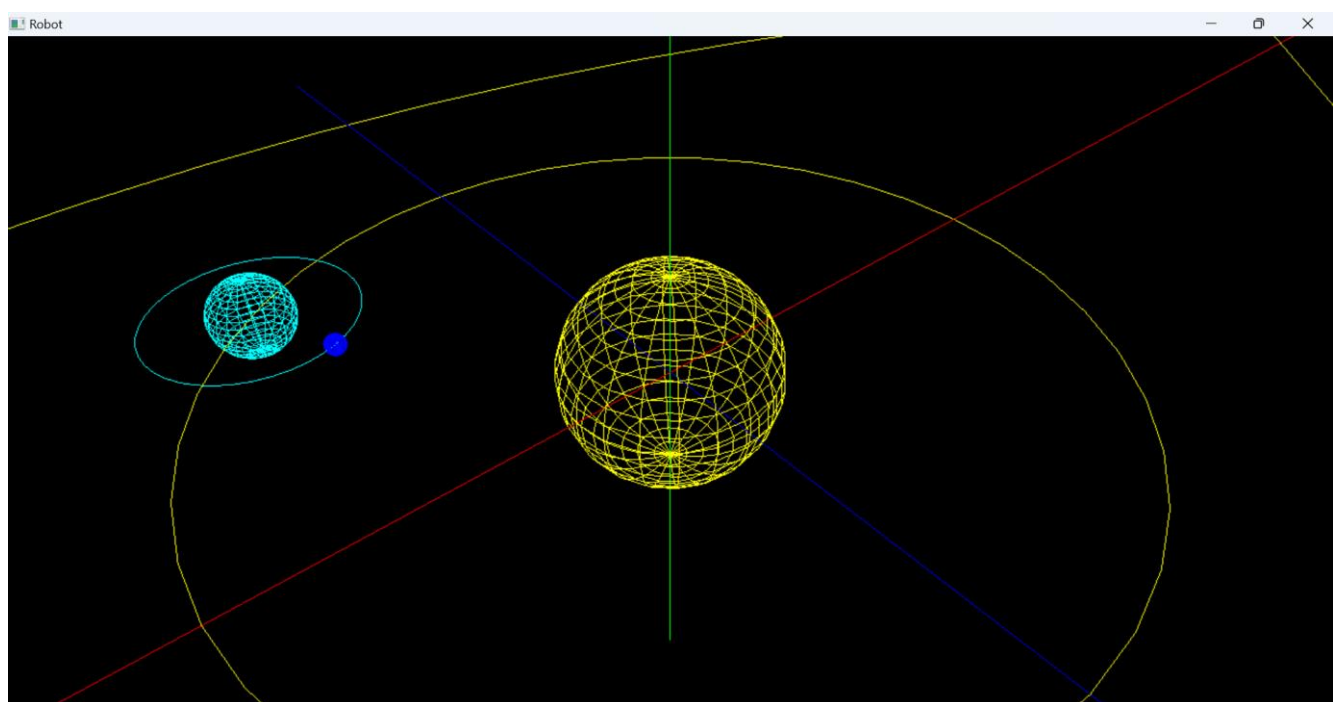
```
//wiezyczka i lufa
kwadryka = gluNewQuadric();
glPushMatrix();
glColor3f(0.1, 0.4, 0.2);
glTranslatef(1.5, 1.5, 0);
glRotatef(obrotWiezyczki, 0, 1, 0); // Obrot wieży
glRotatef(-90, 1, 0, 0);
gluCylinder(kwadryka, 2.0, 1.0, 1.1, 20, 5);
glColor3f(0.4, 0.1, 0.2);
glDeleteQuadric(kwadryka);
kwadryka = gluNewQuadric();
glPushMatrix();
glColor3f(0.6, 0.1, 0.4);
glTranslatef(0, 0, 0.55);
glRotatef(katLufy, 0, 1, 0); // Zmiana kąta obrotu lufy
glRotatef(-90, 0, 1, 0);
glColor3f(0.7, 0.1, 0.5);
gluCylinder(kwadryka, 0.15, 0.15, 4.5, 8, 8);
glPopMatrix();
glDeleteQuadric(kwadryka);
glPopMatrix();
```

C. Prezentacja działania programu – układ planet

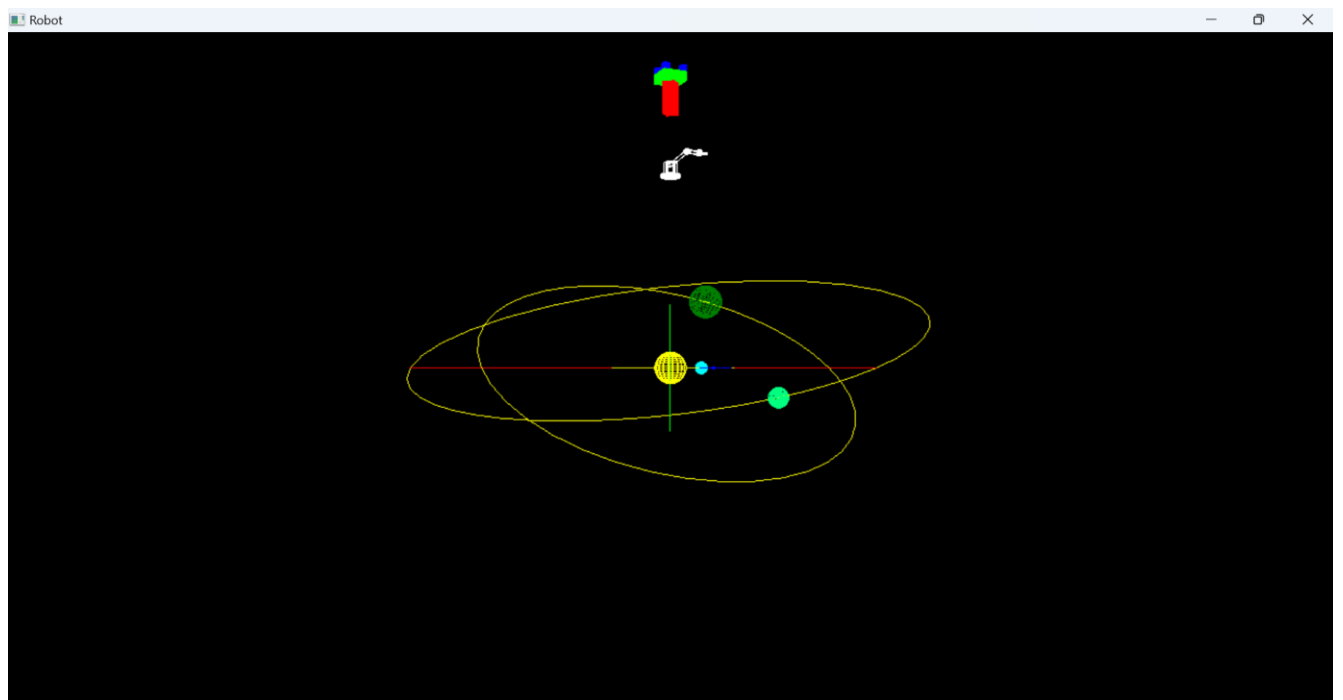
Widok całego układu:



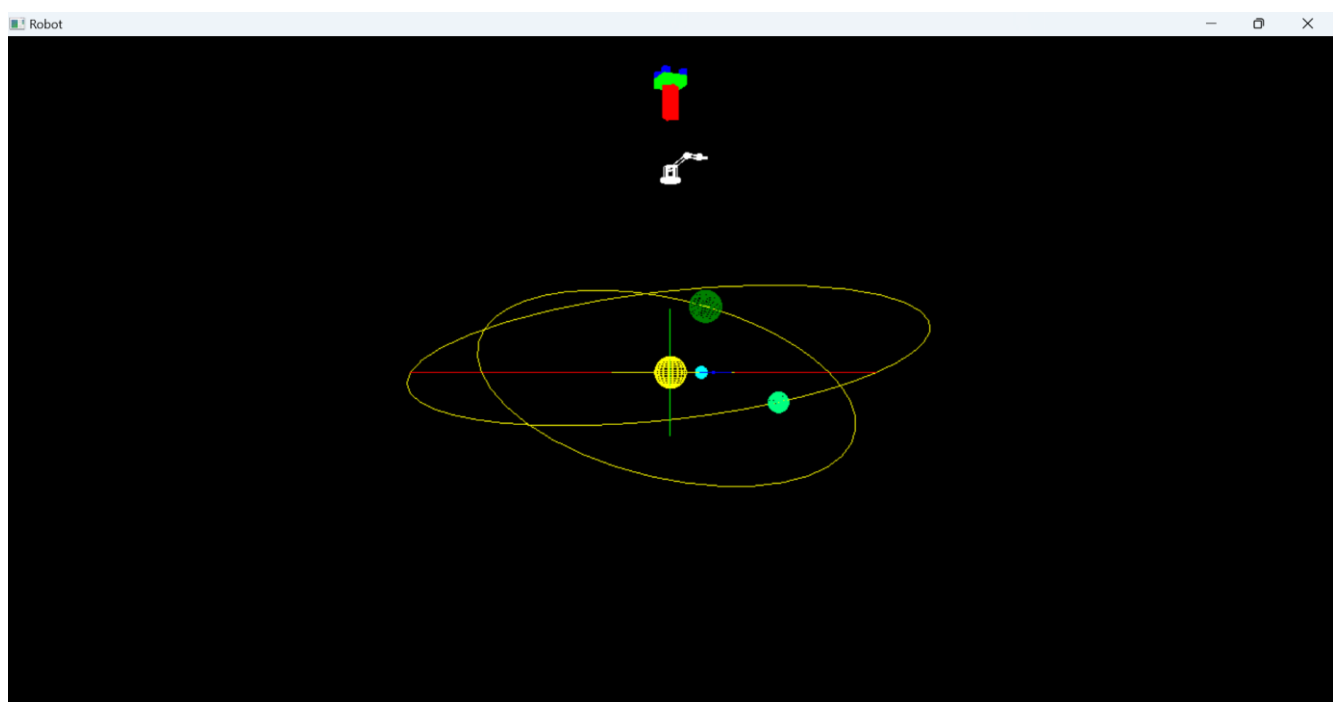
Widok z bliska:



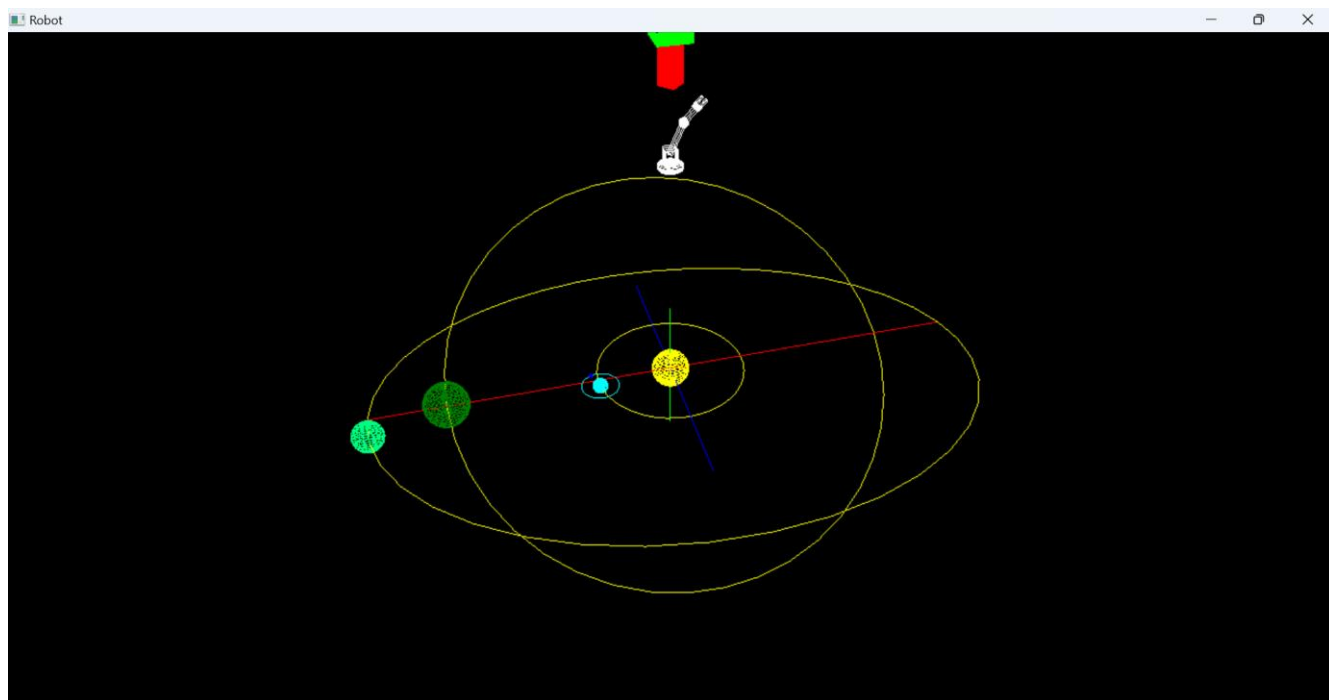
Widok z osi poziomej:



Ustawienie robota 1

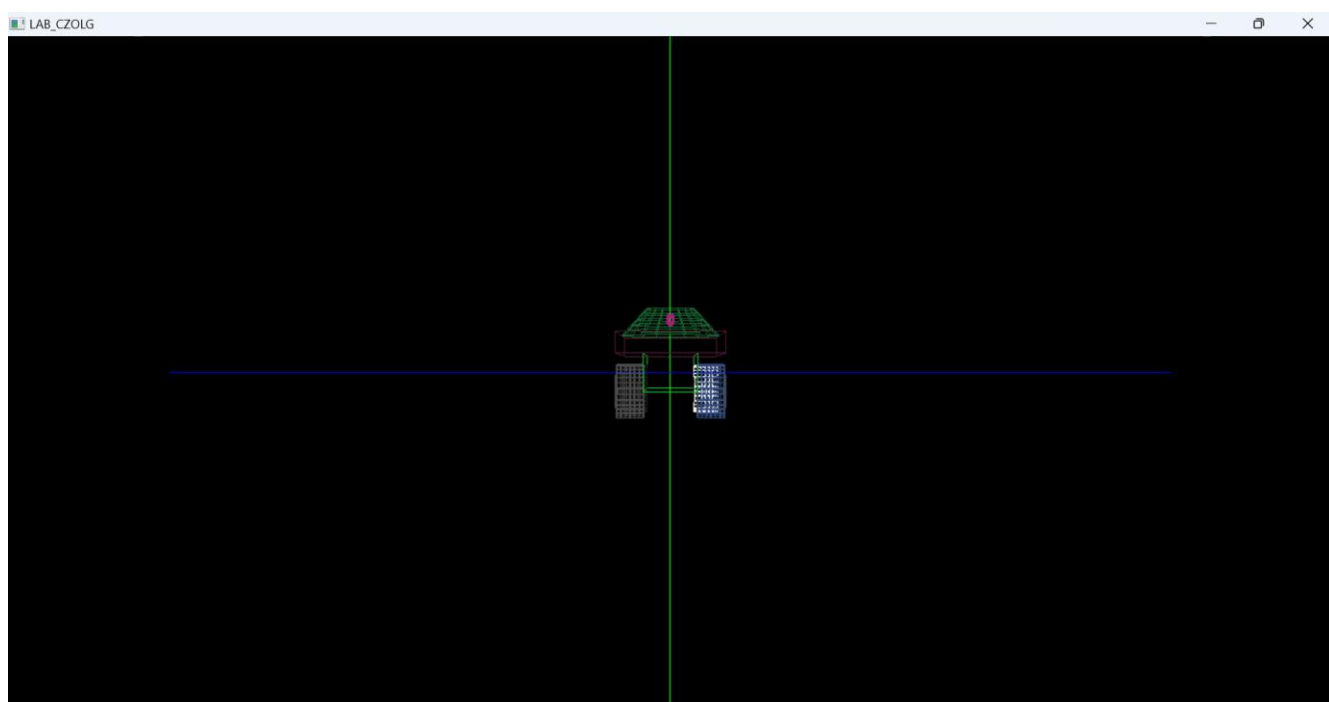


Ustawienie robota 2

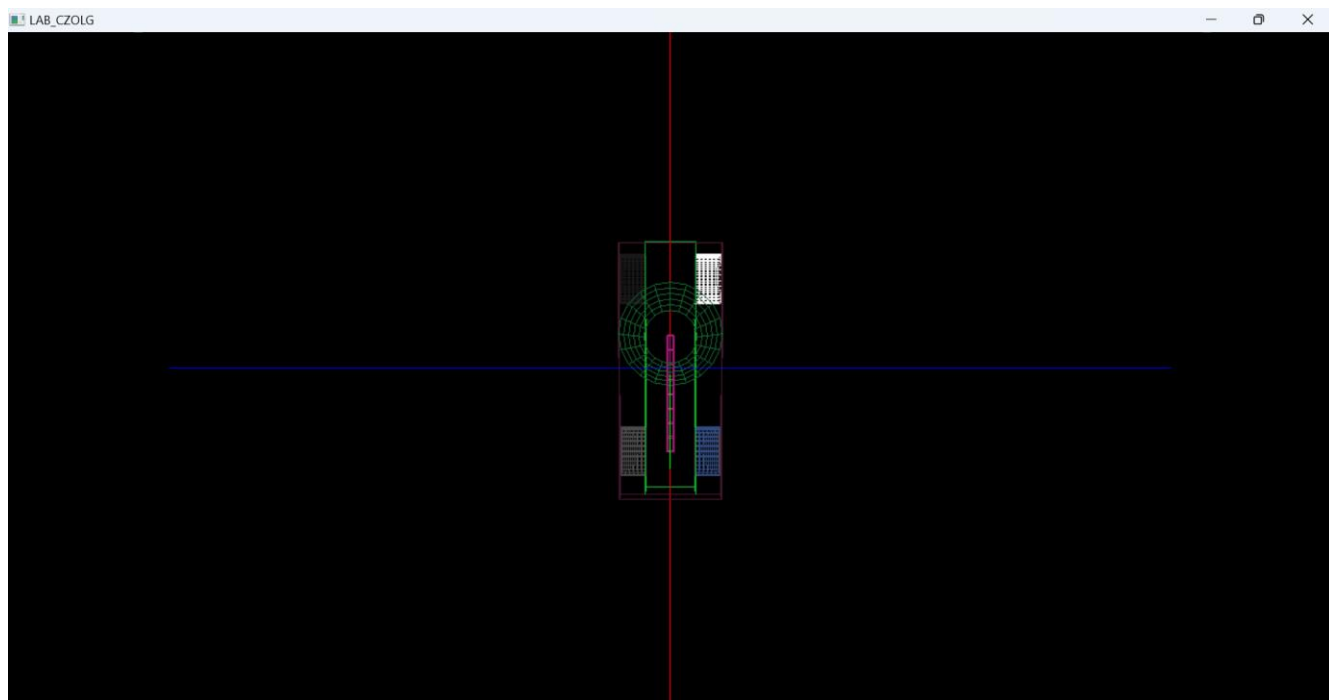


Widok wozu od przodu:

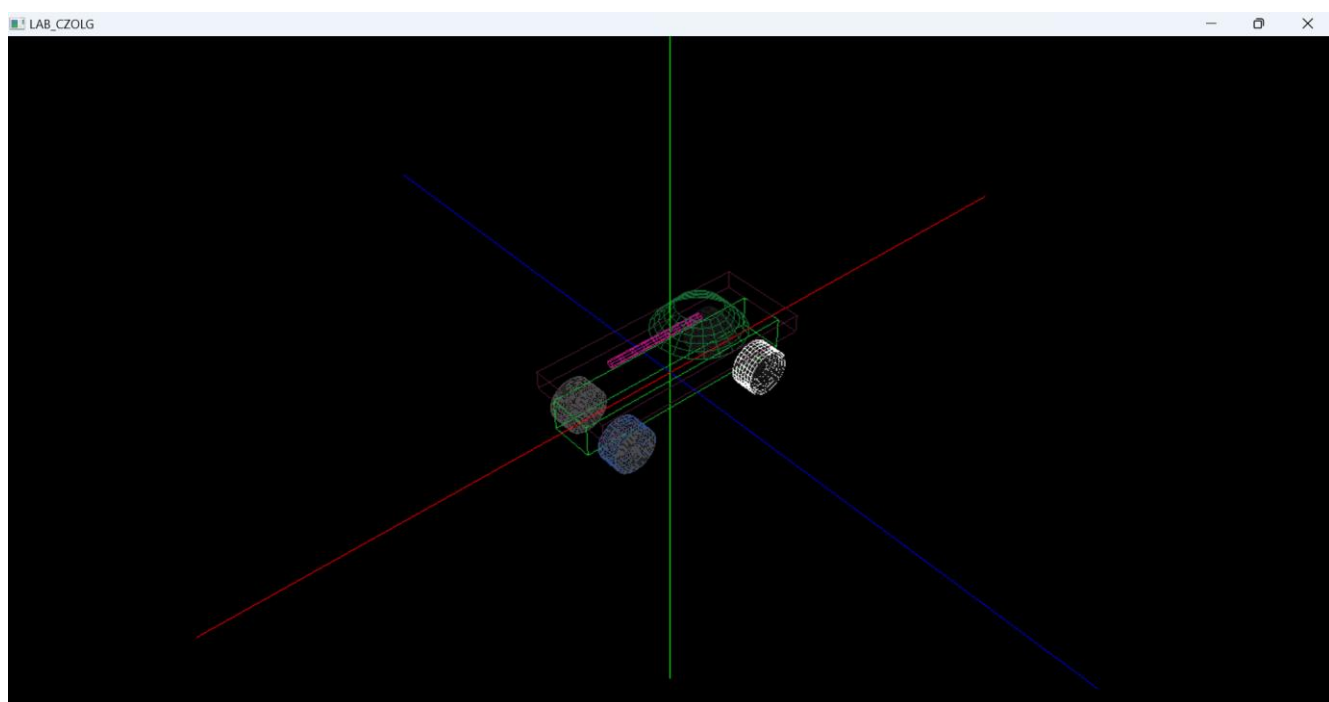
D. Prezentacja działania programu – BWP



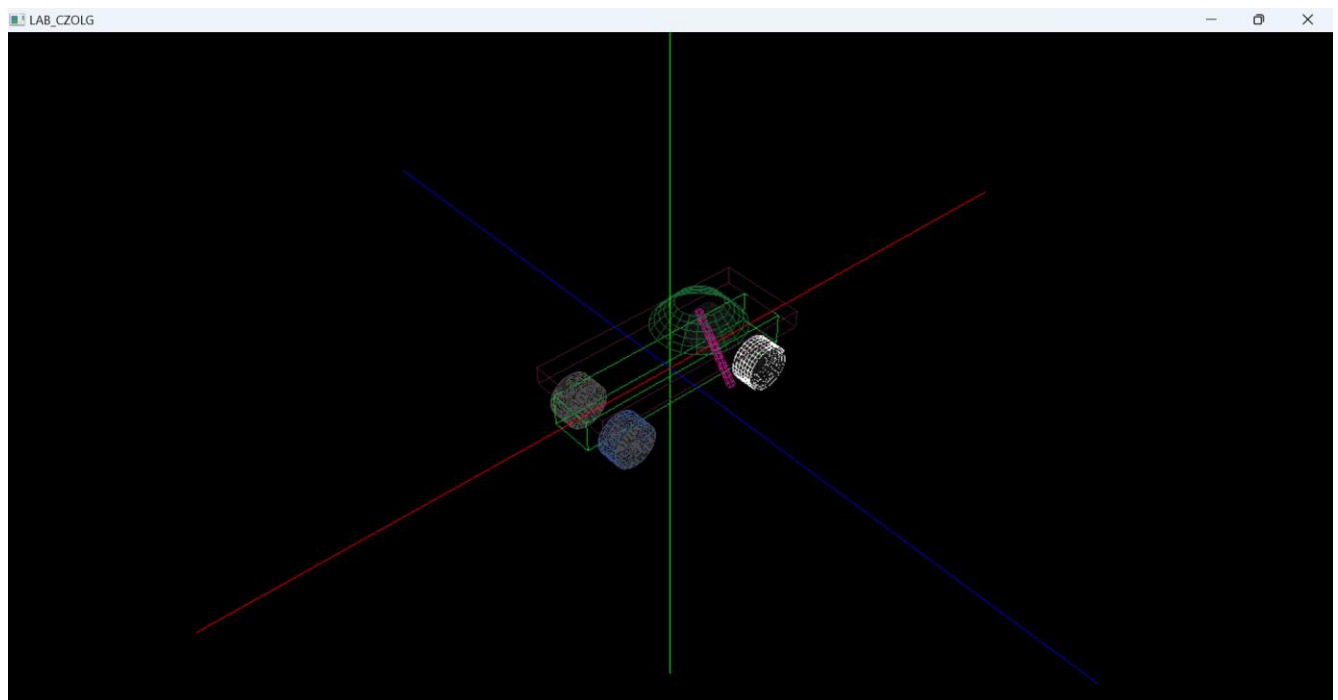
Widok wozu od boku:



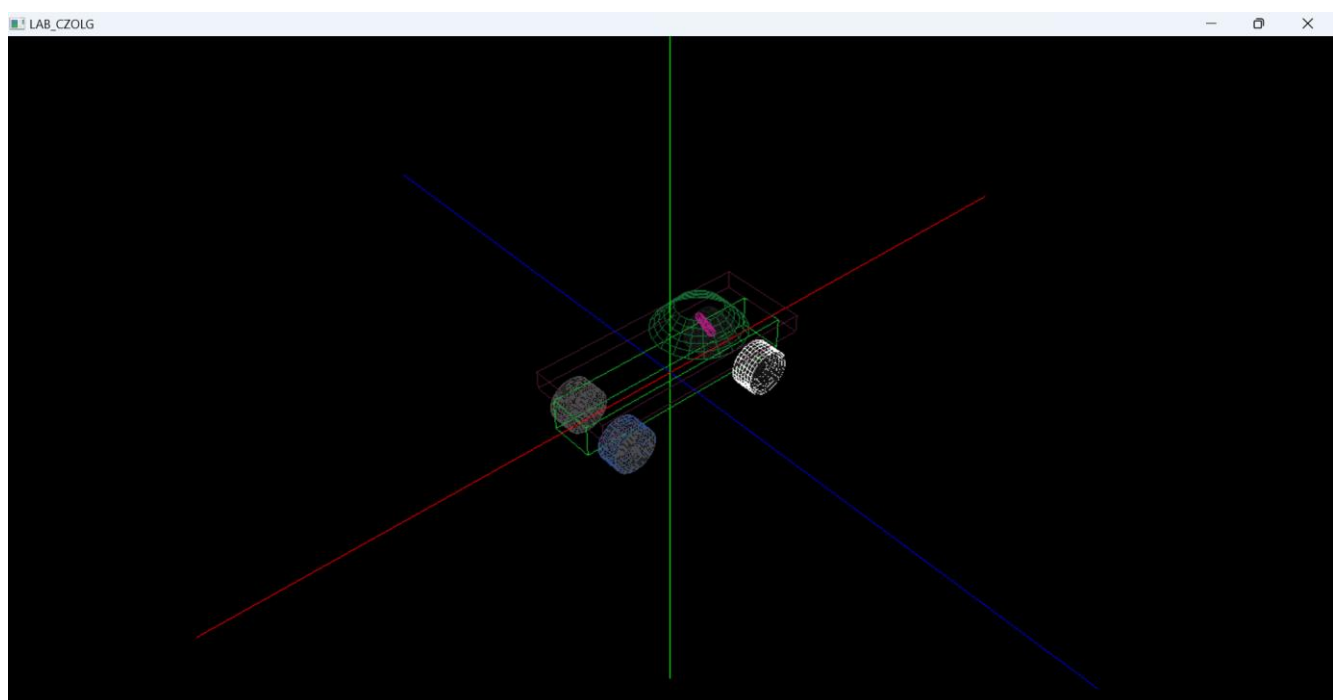
Widok na działo – stan początkowy



Ruch wieżą



Ruch armatą



E. Wnioski

Biblioteka opengl to bardzo przydatne urządzenie w projektowaniu grafiki 3D. Dzięki niej możemy łatwo wprowadzać obiekty i manipulować ich położeniem oraz rozmiarem.

Animacje w projekcie udało się osiągnąć przez cykliczne wprowadzanie drobnych zmian oraz ich wyświetlanie, co przy odpowiedniej ilości klatek na sekundę daje wrażenie ruchu obiektu (mimo, że od strony programistycznej jest to tworzenie nowego obiektu, o takich samych rozmiarach, ale innym położeniu).