

# Zad. 5: Rotacje 3D

## 1 Cel ćwiczenia

Wykształcenie umiejętności modelowania kluczowych dla danego problemu pojęć. Tworzenie diagramu klas. Praktyczne zweryfikowanie wcześniejszej konstrukcji programu. Jeśli nie jest dobrze napisany, to rozszerzenie wymagane w tym zadaniu będzie trudniejsze.

## 2 Program zajęć

- *Ocena realizacji zadania z poprzedniego laboratorium* – ocenie podlega poprawność realizacji zadania, styl pisania programu oraz dokumentacja wygenerowana za pomocą programu *doxygen*.
- *Ocena przygotowania do zajęć* – ocenie podlega diagram klas.
- *Modyfikacja programu wg wskazań osoby prowadzącej* – ocenie będzie podlegała poprawność realizacji modyfikacji. Pracę nad modyfikacją programu (wszystkie operacje należy wykonywać na kopii) należy rozpocząć już w trakcie pierwszej fazy laboratorium, gdyż prowadzący nie będzie w stanie ocenić wcześniejszego programu wszystkim jednocześnie.
- *Realizacja wstępnej fazy prac nad nowym zadaniem* – w ramach wstępnej realizacji zadania należy stworzyć odpowiednio zmodyfikować klasy modelujące pojęcie wektora oraz macierzy.
- *Ocena realizacji wstępnej fazy zadania*

## 3 Opis zadania programowego

Niniejsze zadanie jest kontynuacją wcześniejszego zadania. Obiektem, który ma być poddawany rotacjom jest prostopadłościan. Zasadnicza część menu programu będzie taka sama. Dochodzą jednak nowe pozycje związane z tym, że program powinien być w stanie składać obroty wokół kolejnych osi zgodnie z żądanym przez użytkownika porządkiem.

- obrót bryły o zadane kąty względem wybranych osi z zadaną ilością powtórzeń,
- powtórzenie poprzedniego obrotu,
- wyświetlenie macierzy rotacji,
- przesunięcie bryły o zadany wektor,
- wyświetlenie współrzędnych wierzchołków,
- sprawdzenie długości dwóch przeciwległych boków,
- wyświetlenie menu,
- zakończenie działania programu.

W tym zadaniu użytkownik powinien mieć możliwość zadania kąta obrotu, jak też osi układu współrzędnych (tzn.  $OX$ ,  $OY$  lub  $OZ$ ), względem której ma być dokonany obrót. Program powinien umożliwić wprowadzenie dowolnej sekwencji obrotów względem poszczególnych osi. Wspomnianą sekwencję kończy wprowadzenie znaku kropki zamiast oznaczenia osi obrotu. Natomiast oznaczeniami poszczególnych osi układu współrzędnych są znaki, odpowiednio:  $x$ ,  $y$  oraz  $z$ . Dalsze wyjaśnienia realizacji tej operacji znajdują się w przykładzie działania programu.

## 4 Przygotowanie do zajęć

Należy przygotować diagram klas proponowanych struktur danych.

## 5 Przykład działania programu

Poniżej przedstawiony przykład wyznacza formę komunikatów i ich forma jest obligatoryjny dla programu tworzonego w ramach niniejszego zadania.

```
panamint> ./program_obroty_3D

o - obrot bryly o zadana sekwencje katow
t - powtorzenie poprzedniego obrotu
r - wyswietlenie macierzy rotacji
p - przesuniecie bryly o zadany wektor
w - wyswietlenie wspolrzecznych wierzcholkow
s - sprawdzenie dlugosci przeciwleglych bokow
m - wyswietl menu
k - koniec dzialania programu

Twój wybór? (m - menu) > w

  1.0000000000    1.0000000000    0.0000000000
11.0000000000    1.0000000000    0.0000000000

  1.0000000000    6.0000000000    0.0000000000
11.0000000000    6.0000000000    0.0000000000

  1.0000000000    6.0000000000    7.0000000000
11.0000000000    6.0000000000    7.0000000000

  1.0000000000    1.0000000000    7.0000000000
11.0000000000    1.0000000000    7.0000000000

Twój wybór? (m - menu) > s

:) Wybrane przeciwlegle boki sa sobie rowne.

Twój wybór? (m - menu) > p
```

Wprowadz wspolrzedne wektora translacji w postaci trzech liczb  
tzn. wspolrzednych: x y z  
1 2 3

Twoj wybor? (m - menu) > w

2.0000000000	3.0000000000	3.0000000000
12.0000000000	3.0000000000	3.0000000000

2.0000000000	8.0000000000	3.0000000000
12.0000000000	8.0000000000	3.0000000000

2.0000000000	8.0000000000	10.0000000000
12.0000000000	8.0000000000	10.0000000000

2.0000000000	3.0000000000	10.0000000000
12.0000000000	3.0000000000	10.0000000000

Twoj wybor? (m - menu) > o  
Podaj sekwencje oznaczen osi oraz katy obrotu w stopniach  
x 5  
y 10  
z 35  
a 46

:( Bledne oznaczenie osi. Dopuszczalne znaki to: x y z .  
:( Spróbuj jeszcze raz.

x 45  
.

Ile razy operacja obrotu ma byc powtorzona?  
3

Twoj wybor? (m - menu) > s

:0 Wybrane przeciwlegle boki nie sa rowne!!!

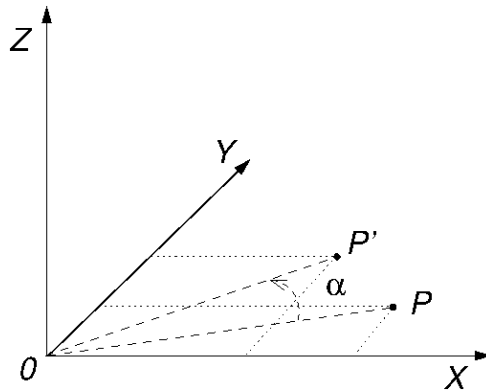
Twoj wybor? (m - menu) > k

Koniec dzialania program

panamint> \_

## 6 Realizacja rotacji

Niech będzie dany punkt  $P = (x, y, z)$ . Rozważmy najpierw rotację tego punktu wokół osi  $OZ$  o kąt  $\alpha$ . Otrzymujemy nowy punkt  $P' = (x', y', z')$  tak jak to jest pokazane na rysunku rys. 1. Łatwo zauważyć, że współrzędna  $z$ -towa tego punktu nie zmienia się. Transformację tę można



Rysunek 1: Rotacja punktu  $P$  o kąt  $\alpha$  wokół osi  $OZ$

potraktować jako rozszerzenie wcześniejszej rotacji w układzie dwuwymiarowym, która opisana została w poprzednim zadaniu. Transformację Współrzędnych punktu  $P$  do współrzędnych punktu  $P'$  realizujemy zgodnie z następującym wzorem:

$$\begin{aligned}x' &= x \cos \alpha - y \sin \alpha, \\y' &= x \sin \alpha + y \cos \alpha, \\z' &= z.\end{aligned}$$

Możemy go zapisać w postaci macierzowej

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Tak więc mając na uwadze, że  $P = (x, y)$  i  $P' = (x', y')$  oraz oznaczając macierz rotacji  $\mathbf{R}_{z,\alpha}$ , powyższą wzór można zapisać w formie

$$P' = \mathbf{R}_{z,\alpha} \cdot P.$$

W programie należy dokonać odpowiednich przeciążeń operatorów, aby tego typu zapis można było stosować bezpośrednio w programie.

Analogicznie obrót wokół osi  $OY$  o kąt  $\beta$  możemy zapisać jako

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Zmiana znaku przy funkcji sinus jest związana z przyjętą konwencją miary kąta. Kierunek dodatni miary jest od osi  $OZ$  do osi  $OX$ . Przedstawiony wyżej typ przekształcenia dalej zapisywać będziemy jako

$$P'' = \mathbf{R}_{y,\beta} \cdot P.$$

Natomiast macierz obrotu wokół osi  $OX$  o kąt  $\gamma$  ma postać

$$\begin{bmatrix} x''' \\ y''' \\ z''' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Analogicznie do wcześniejszej notacji przekształcenie to zapisywać będziemy jako

$$P''' = \mathbf{R}_{x,\gamma} \cdot P.$$

Załóżmy, że mamy przetransformować pewien punkt  $P_a$  poprzez złożenie dwóch obrotów. Tak więc najpierw chcemy dokonać obrotu wokół osi  $OZ$  o kąt  $\alpha$ , a później wokół osi  $OX$  o kąt  $\gamma$ . Tym samym chcemy wykonać następującą operację

$$P'_a = \mathbf{R}_{z,\alpha} P_a,$$

a następnie

$$P''_a = \mathbf{R}_{x,\gamma} P'_a.$$

Jednak te operacje możemy złożyć i zapisać jako

$$P''_a = \mathbf{R}_{x,\gamma}(\mathbf{R}_{z,\alpha} P_a) = (\mathbf{R}_{x,\gamma} \mathbf{R}_{z,\alpha}) P_a.$$

Tak więc składanie obrotów jest równoważne przemnożeniu odpowiednich macierzy obrotów. Tę cechę należy wykorzystać przy oprogramowaniu pozycji menu

- o - obrot bryły o zadana sekwencje katow

Należy zauważyć że obroty wokół różnych osi nie są przemienne, tzn. w ogólnym przypadku mamy

$$\mathbf{R}_{x,\gamma} \mathbf{R}_{z,\alpha} \neq \mathbf{R}_{z,\alpha} \mathbf{R}_{x,\gamma}.$$

## 7 Wymagania co do konstrukcji programu

Oprócz wymagań sformułowanych w opisie zadania należy uwzględnić uwarunkowania przedstawione poniżej.

- Należy zdefiniować klasy `Wektor3D`, `Macierz3x3` na bazie klas `Wektor2D`, `Macierz2x2`. W klasach `Wektor3D`, `Macierz3x3` tych powinny być zdefiniowane analogiczne przeciążenia operatorów jak w zadaniu poprzednim. Natomiast na bazie klasy `Prostokat` należy zdefiniować klasę `Prostopadloscian`. Definiowane klasy muszą mieć **tylko i wyłącznie niezbędne pola reprezentujące atrybuty danego pojęcia**.
- Należy również zwrócić uwagę, że ze względu na konieczność pamiętania wcześniejszego obrotu, niezbędne jest przechowywanie macierzy obrotu. Dobrze jest więc stworzyć dodatkową strukturę danych, której polami są odpowiednio bryła i macierz obrotu. Struktura ta będzie reprezentowała scenę, na której znajduje się obracana bryła.
- Program musi zachować strukturę modułową i odpowiednią strukturę kartotek. O ile będzie to konieczne, należy zmodyfikować plik `Makefile` (np. gdy dodany zostanie nowy moduł).

- Każda z klas powinna zostać zdefiniowana w oddzielnym pliku nagłówkowym. Metody tej klasy powinny być natomiast definiowane w osobnym module związanym z daną klasą, np. definicja klasy `Prostopadloscian` powinna znaleźć się w pliku nagłówkowym `Prostopadloscian.hh`, zaś metody w pliku `Prostopadloscian.cpp`. Proste metody można definiować bezpośrednio w ciele klasy.
- Dla poszczególnych klas należy przeciążyć niezbędne operatory działające na strumieniach. Nie wszystkie przeciążenia są w tym zadaniu potrzebne. Na pewno będą potrzebne przeciążenia operatorów wczytywania i zapisu dla klasy `Wektor3D` oraz operatora wyświetlania dla klasy `Prostopadloscian`.
- Wszystkie metody, które nie zmieniają stanu obiektu, na którym działają, powinny być metodami typu `const`.
- Program powinien umożliwiać graficzną wizualizację wyników działania. Pozwala na to dołączony w załączku moduł łączący do programu `gnuplot`.
- Wszystkie klasy i metody oraz funkcje powinny zostać opisane. Opis ten powinien być zgodny z wymogami systemu `doxygen`. Ponadto należy wygenerować dokumentację w formacie HTML za pomocą programu `doxygen`.

Oprócz tego pozostają w mocy wszystkie wcześniejsze wymagania dotyczące struktury katalogów, pliku `Makefile`, modułowej struktury programu, jak też opisów.

## 8 Materiały pomocnicze

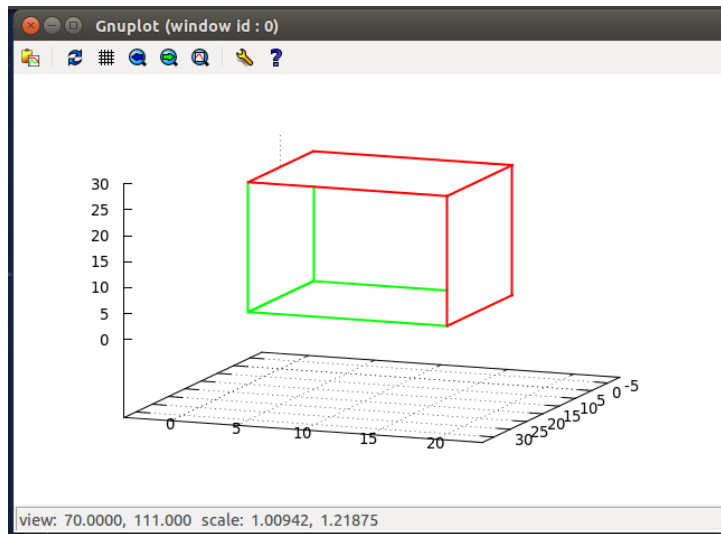
Załączek programu znajduje się w katalogu `~bk/edu/kpo/zad/z5`. Zawiera on przykład wykorzystania modułu łączącego do programu `gnuplot`, który pozwala zwizualizować wyniki obliczeń. Podstawowe objaśnienia znajdują się w kodzie dostarczonego załączka.

We wspomnianym katalogu znajduje się również przykład realizacji programu.

## 9 Zapis danych dla programu `gnuplot`

Program `gnuplot` pozwala na rysowanie powierzchni. Można w ten sposób narysować wszystkie ścianki bryły. Jednak jest to trochę uciążliwe. Dlatego w tym zadaniu proponuje się narysowanie powierzchni prostopadłościanu bez dwóch przeciwległych ścianek (patrz rys. 2). Z tego powodu prostopadłościan przedstawiony na rys. 2 widoczny jest z *prześwitem*. Brzeg tych ścianek, które widoczne są od zewnątrz rysowany jest kolorem czerwonym. Natomiast brzeg ścianek widoczny od wewnątrz rysowany jest kolorem zielonym. Aby otrzymać rysunek w przedstawionej postaci, współrzędne wierzchołków należy zapisać do pliku tekstowego, który będzie przekazany `gnuplotowi`, w odpowiedni sposób.

Chcąc lepiej przedstawić sposób zapisu na rys. 3 przedstawiony jest rysunek prostopadłościanu z oznaczeniem wszystkich wierzchołków. Schemat zapisu współrzędnych przedstawiony jest poniżej. Uwaga: wolne linie są istotne, należy również zwrócić uwagę na to, że na końcu należy powtórzyć zapis współrzędnych pierwszego i ostatniego wierzchołka.



Rysunek 2: Rysunek prostopadłościanu w okienku programu gnuplot

```

x1 y1 z1 # <-- Współrzędne wierzchołka W1
x2 y2 z2 # <-- Współrzędne wierzchołka W2

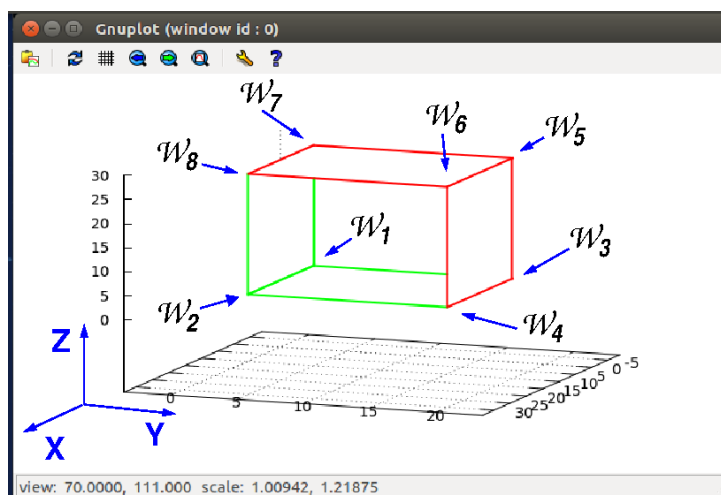
x3 y3 z3 # <-- Współrzędne wierzchołka W3
x4 y4 z4 # <-- Współrzędne wierzchołka W4

x5 y5 z5 # <-- Współrzędne wierzchołka W5
x6 y6 z6 # <-- Współrzędne wierzchołka W6

x7 y7 z7 # <-- Współrzędne wierzchołka W7
x8 y8 z8 # <-- Współrzędne wierzchołka W8

x1 y1 z1 # <-- Współrzędne wierzchołka W1
x2 y2 z2 # <-- Współrzędne wierzchołka W2

```



Rysunek 3: Rysunek prostopadłościanu w okienku z zaznaczonymi kolejnymi wierzchołkami

W dalszej części przedstawiony jest zapis faktycznych współrzędnych wierzchołków prostopadłościanu przedstawionego na obu rysunkach.

```
2.0000000000 3.0000000000 3.0000000000
22.0000000000 3.0000000000 3.0000000000
```

```
2.0000000000 18.0000000000 3.0000000000
22.0000000000 18.0000000000 3.0000000000
```

```
2.0000000000 18.0000000000 28.0000000000
22.0000000000 18.0000000000 28.0000000000
```

```
2.0000000000 3.0000000000 28.0000000000
22.0000000000 3.0000000000 28.0000000000
```

```
2.0000000000 3.0000000000 3.0000000000
22.0000000000 3.0000000000 3.0000000000
```

## 10 Uproszczenie

Jeżeli zadanie sprawia zbyt duże problemy, to można je zrealizować w uproszczonej formie. W dalszej części przedstawione są proponowane zmiany.

### 10.1 Pominięcie sekwencji

W uproszczonej wersji przy realizacji opcji:

- o - obrot bryły o zadana sekwencje katow

można przyjąć, że obrót zadawany jest tylko względem jednej wybranej osi. Tak więc użytkownik zamiast wprowadzać sekwencję oznaczeń: znak osi, kąt obrotu; którą kończy kropką, wprowadzi tylko jedną taką parę. Skorzystanie z tego uproszczenia powoduje obniżenie oceny programu o 1,0 (tzn. jedną ocenę).

### 10.2 Pominięcie powtórzeń

W kolejnym uproszczeniu można zrezygnować z opcji powtórzeń. Powoduje to jednak, że ocena programu będzie obniżona o 0,5.

## 11 Rozszerzenia

Dla osób, które niniejsze zadanie nie sprawi problemu, proponuje się rozszerzenia przedstawione poniżej.

### 11.1 Zbiór brył

Zamiast pojedynczej bryły, należy stworzyć scenę na której jest co najmniej 5 brył i każdą z nich z osobna można obracać niezależnie od innych.



## 11.2 Rysowanie bryły

Należy tak zmodyfikować układ punktów zapisywanych w pliku, aby rysowała się pełna bryła, np. tak jak jest to przedstawione na stronie:

<http://rab.iiar.pwr.wroc.pl/~kreczmer/kpo/zadania/zad-manipulator/pomoc/zasoby/laczedognuplota-doc/index.html>