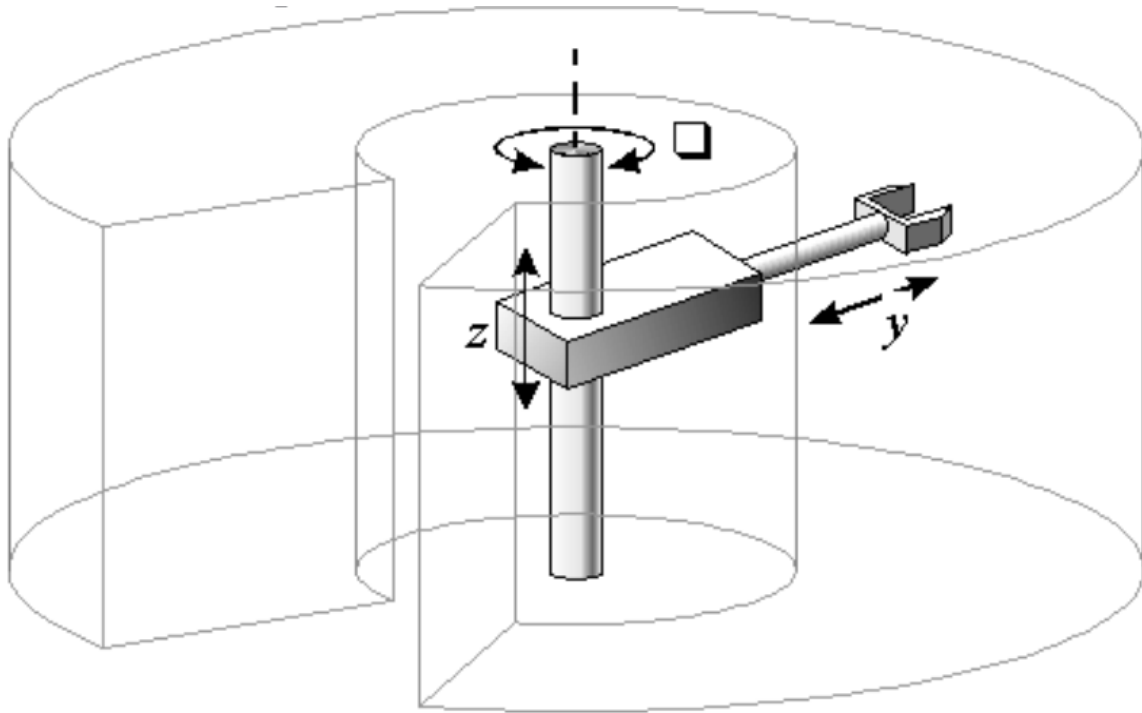


# Sprawozdanie z projektu z przedmiotu “Programowanie Obiektowe i Grafika Komputerowa”

Oskar Nowak 184289 | Adrian Nowogrodzki 184332

Temat Projektu: “3. Zrealizować animowaną wizualizację ramienia robota typu „cylindrical arm””

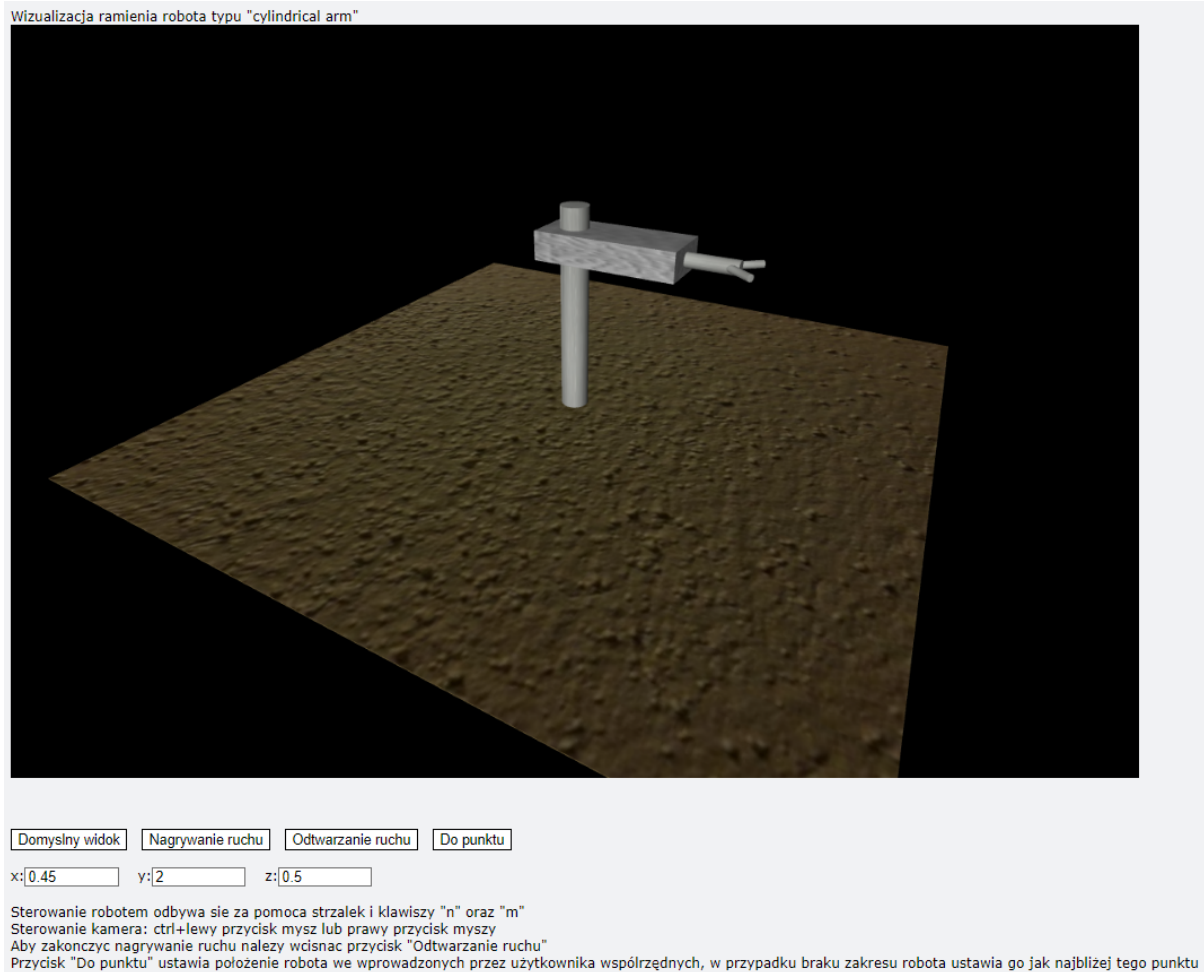


Program został napisany w języku Python z użyciem biblioteki VPython.

Symulator posiada następujące funkcjonalności:

- Sterowanie manipulatorem w przestrzeni trójwymiarowej
- Ruch kamerą obserwatora
- Domyślny widok - ustawia kamerę w pozycji startowej
- Nagrywanie ruchu - zapisuje ruchy użytkownika
- Odtwarzanie ruchu - odtwarza zapisane ruchy użytkownika
- Kinematyka odwrotna - wykonuje ruch autonomiczny do wprowadzonego przez użytkownika położenia

## Wynik uruchomienia programu:



## Struktura programu:

Cała wizualizacja wykonywana jest za pomocą instancji klasy Visualization i wykonywana jest w pętli głównej umieszczonej w metodzie run() tej klasy.

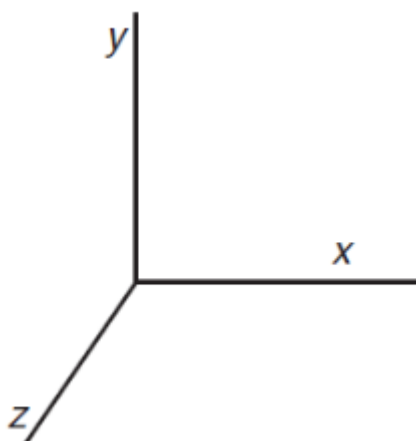
```
30.     def run(self):
31.
32.         while True:
33.             k = keydown()
34.             if self.recording:
35.                 self.record_list.append(k)          #zapisywanie wejsc klawiatury do listy, w trakcie trybu nagrywania; z tej listy
                                                       zostana odtworzone ruchy ramienia robota
36.
37.                 if not self.playing and not self.to_point:
38.                     self.control(k)
```

Wszystkie ruchy roboty, sekwencyjne lub “na żywo” z klawiatury, wykonywane są za pomocą metody control(), która jako obowiązkowy argument przyjmuje listę “wejść”, czyli ruchów, które mają zostać wykonane w danej chwili.

Argument rate\_val odpowiada za maksymalne klatkowanie.

```
40.     def control(self, k, rate_val=5000):
41.         i=0
42.         rate(rate_val)
43.         if 'left' in k:
44.             i = 0.001
45.         if 'right' in k:
46.             i = -0.001
47.         self.rotate(i)
48.         i = 0
49.         if 'up' in k:
50.             i = 0.0001
51.         if 'down' in k:
52.             i = -0.0001
53.         self.move(i)
54.         i = 0
55.         if 'm' in k:
56.             i = 0.0001
57.         if 'n' in k:
58.             i = -0.0001
59.         self.extend(i)
```

Układ współrzędnych w programie:



Funkcję kinematyki odwrotnej realizuje metoda kinOdw(), która kolejno ustawia ramię robota na odpowiedniej wysokości - oś 'y' i za pomocą zamiany układu współrzędnych kartezjańskich 'x' i 'z' na współrzędne biegunowe przesuwa ramię do danego kąta i odpowiednio je wysuwa/wsuka.

```
157.     def kinOdw(self):
158.         self.to_point = True
159.         self.makeCaption('W trakcie ruchu do punktu')
160.         self.btn_rec.disabled = True
161.         self.btn_play.disabled = True
162.         self.btn_point.disabled = True
163.
164.         ang_to = atan2(self.number_z, self.number_x)
165.         mag_to = sqrt(self.number_x**2 + self.number_z**2)
166.
167.         #sprawdzenie czy punkt nie jest poza zakresem ramienia
168.         if self.number_y > 2.7999: self.number_y = 2.7999
169.         elif self.number_y < 0.2001: self.number_y = 0.2001
170.         if mag_to > 0.99: mag_to = 0.99
171.         elif mag_to < 0.2: mag_to = 0.2
172.
173.         #ustawienie ramienia robota do docelowego punktu
174.         while self.ext_obj.pos.y < self.number_y:
175.             self.control(['up'])
176.         while self.ext_obj.pos.y > self.number_y:
177.             self.control(['down'])
178.         while atan2(self.ext_obj.pos.z, self.ext_obj.pos.x) > ang_to:
179.             self.control(['left'])
180.         while atan2(self.ext_obj.pos.z, self.ext_obj.pos.x) < ang_to:
181.             self.control(['right'])
182.         while sqrt(self.ext_obj.pos.x**2 + self.ext_obj.pos.z**2) > mag_to:
183.             self.control(['n'])
184.         while sqrt(self.ext_obj.pos.x**2 + self.ext_obj.pos.z**2) < mag_to:
185.             self.control(['m'])
186.
187.         self.number_x = round(self.ext_obj.pos.x, 3)
188.         self.number_z = round(self.ext_obj.pos.z, 3)
189.
190.         self.to_point = False
191.         self.makeCaption()
192.         self.btn_rec.disabled = False
193.         self.btn_play.disabled = False
194.         self.btn_point.disabled = False
195.
```