**1. Wstęp**

* Wykonywane zadanie: **Zad. 3 – Ramię robota typu „cylindrical arm”**
* Cel zadania: Celem projektu jest zaprojektowanie i implementacja robota cylindrycznego z wykorzystaniem środowiska programistycznego Java3D. Robot ten powinien posiadać przynajmniej trzy stopnie swobody, a także mieć możliwość zapamiętywania i odtwarzania zadanych przez użytkownika sekwencji ruchów. Użytkownik musi mieć ponadto możliwość sterowania robotem z poziomu klawiatury w obrębie dostępnych stopni swobody. Program musi udostępniać użytkownikowi interfejs z dostępem do odpowiednich funkcji robota, a także posiadać wbudowany prosty element wchodzący w interakcję z robotem.

**2. Lista zaimplementowanych elementów:**

* szkielet robota cylindrycznego posiadającego trzy stopnie swobody oraz nieruchomy chwytak (stworzone z obiektów Shape3D);
* prymityw wchodzący w interakcję z robotem;
* możliwość sterowania elementami robota za pomocą klawiatury;
* możliwość sterowania prymitywem za pomocą strzałek na klawiaturze;
* możliwość obracania i oddalania kamery za pomocą myszy;
* możliwość uczenia robota i wykonywania zapamiętanego programu;
* możliwość zresetowanie położenia kamery w celu łatwiejszej obsługi interfejsu;
* grawitacja oddziałująca na upuszczony prymityw;
* wykrywanie kolizji robota z prymitywem ;
* nałożenie tekstur na elementy przestrzeni oraz dodanie tła;

**3. Opis sterowania robotem i elementem interaktywnym (walcem) z poziomu klawiatury:**

* Unoszenie robota w osi pionowej - **W**
* Opuszczanie robota w osi pionowej - **S**
* Obrót robota w prawo - **D**
* Obrót robota w lewo - **A**
* Wysuwanie chwytaka - **E**
* Wsuwanie chwytaka - **Q**
* Oddalanie się prymitywu od robota w linii prostej - **Strzałka w górę**
* Przybliżanie się prymitywu do robota w linii prostej - **Strzałka w dół**
* Ruch prymitywu po okręgu w prawo względem osi pionowej robota - **Strzałka w prawo**
* Ruch prymitywu po okręgu w lewo względem osi pionowej robota - **Strzałka w lewo**

**4. Sterowanie kamerą:**

* Obracanie widokiem wokół obiektów - **Lewy przycisk myszy (przytrzymać)**
* Przybliżanie i oddalanie widoku - **Kółko myszy**

**5. Struktura programu:**

TG\_pomocnicza\_przedmiotu

TG\_przedmiot

Przedmiot

TG\_podloga

Chwytak2

Ramie1

Ramie2

wezel\_scena

Pion

Chwytak3

TG\_chwytak3

TG\_chwytak2

TG\_ramie1

TG\_ramie2

TG\_os\_pionowa

TG\_chwytak1

Chwytak1

Podloga

Opis struktury:

- BranchGroup

- TransformGroup

- Kształty 3D

**6. Opis przycisków interfejsu:**

* **Złap -** chwycenie prymitywu przez robota
* **Puść -** puszczenie prymitywu przez robota
* **Uczenie robota -** rozpoczyna tryb uczenia; kończy tryb wykonywania, jeśli był aktywny; od tego momentu wszystkie ruchy robota są zapamiętywane
* **Wykonywanie programu -** rozpoczyna tryb wykonywania zapamiętanej sekwencji ruchów;kończy tryb uczenia, jeśli byłaktywny
* **Swobodna praca -** wyłącza zarówno tryb uczenia, jak i tryb wykonywania; umożliwia swobodne sterowanie robotem przez użytkownika
* **Reset kamery** -ustawia początkowe położenie kamery

**7. Najważniejsze metody i klasy:**

* **run -** metoda klasy Ruchy\_robota; odpowiada za wywoływanie funkcji odpowiedzialnych za poruszanie elementami robota oraz walcem zgodnie z taktem wewnętrznego zegara programu;
* **utworzScene -** wewnątrz tej metody definiowane są elementy robota i jego otoczenie;
* **actionPerformed -** metoda odpowiedzialna za obsługę zdarzeń związanych z przyciskaniem przez użytkownika klawiszy interfejsu;
* **Podnoszenie\_sie\_w\_pionie\_robota, Opuszczanie\_sie\_w\_pionie\_robota, Wysuwanie\_ramienia\_robota, Cofanie\_ramienia\_robota, Obrot\_w\_lewo\_robota, Obrot\_w\_prawo\_robota -** metody dokonujące zmian położenia odpowiadających im elementów robota;
* **Ruch\_przedmiotu\_lewo, Ruch\_przedmiotu\_prawo, Ruch\_przedmiotu\_przod, Ruch\_przedmiotu\_tyl** - metody odpowiedzialne za zmianę położenia walca;
* **keyPressed, keyReleased -** metody obsługujące zdarzenia związane z przyciskaniem i puszczaniem przez użytkownika klawiszy na klawiaturze;
* **Projekt\_JAVA -** główny konstruktor programu; definiowane są tutaj wszystkie najważniejsze parametry przestrzeni 3D; znajduje się tu implementacja sterowania kamerą;
* **Grawitacja –** metoda obsługująca zjawisko grawitacji oddziałujące na prymityw
* **CollisionDetector –** klasa obsługująca kolizję między robotem a prymitywem

**Szczegółowe wytłumaczenie działania metod i klas znajduje się w komentarzach w kodzie programu**