**Sprawozdanie z projektu**

Przedmiot: Programowanie obiektowe i grafika komputerowa

Autorzy: Antoni Sulkowski (197564), Dorian Neumann (198415)

1. **Wstęp**

Celem projektu było zrealizowanie sterowalnej przez użytkownika wizualizacji trójwymiarowego ramienia robota typu articulated arm. Projekt został wykonany w języki Python z pomocą takich bibliotek jak OpenGL, FreeSimpleGUI oraz innych bibliotek wspomagających obliczenia i łączność między oknami GUI. Program wykorzystuje paradygmat programowania obiektowego – każda część ramienia została zaimplementowana jako osobna metoda w obrębie klasy reprezentującej całe ramię robota. Projekt umożliwia wizualizację ruchów ramienia w czasie rzeczywistym, a jego struktura ułatwia dalszy rozwój i rozbudowę o dodatkowe funkcjonalności.

1. **Struktura programu i funkcjonalności**

Sam program składa się z 6 plików:

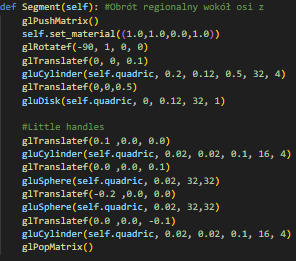
* Main.py – jest to główny plik wykonawczy. Sprzęża wszystkie pozostałe pliki oraz inicjalizuje działanie całego programu.
* Articulated\_arm.py – plik zawiera funkcje klasowe. W tym pliku realizowane jest rysowanie odpowiednich elementów ramienia robota.
* Coordinates.py – plik, w którym wykonywane są wszystkie obliczenia wymagane do określenia obecnej pozycji chwytaka, do przemieszczenia robota na konkretne koordynaty.
* Controls.py – plik, który obsługuje sterowanie klawiaturą/interfejsem.
* Controls\_window.py –
* Surroundings.py – plik zawierający implementacje funkcji, które rysują obiekty dodatkowe, takie jak światło, podstawa robota, pomieszczenie oraz dodatkowy obiekt.
  1. **Articulated\_arm.py**

Głównym elementem tego pliku jest klasa *Robot*, która odpowiada za całe ramię robota.

Metoda set\_material ustawia właściwości materiału obiektu 3D, definiując sposób, w jaki jego powierzchnia reaguje na oświetlenie w scenie. Funkcja określa kolor i intensywność odbicia światła rozproszonego i zwierciadlanego oraz stopień połyskliwości, co pozwala uzyskać realistyczny wygląd ramienia robota.



Funkcja o nazwie *Segment* generuje pierwszy segment robota będący segmentem obrotowym wokół globalnej osi Z (pionowej).



* 1. **Coordinates.py**
  2. **Controls.py**
  3. **Controls\_window.py**
  4. **Surroundings.py**
  5. **Main.py**