Interpreter opisu działań obiektów – Etap 2.

1 Wprowadzenie

Niniejszy dokument opisuje zakres prac, które należy wykonać w ramach 2. etapu. Etap ten nominalnie trwa od najbliższych zajęć do zajęć następnych. Jednak do najbliższych zajęć należy wykonać wstępne prace, które wymienione są w dalszej części niniejszego opisu.

2 Zakres prac

W ramach pracy nad programem należy zrealizować następujące podzadania:

- Wczytywanie konfiguracji w XML.
- Łączenie z serwerem graficznym.
- Przesyłanie do serwera poleceń wizualizacji i rysowania.
- (Rozszerzenie) Sekwencyjne wykonywanie poleceń.

3 Zakres prac wstępnych

Wymienione poniżej podzadania należy zrealizować do najbliższych lub bezpośrednio na zajęciach. Stanowią one część zadań wymienionych we wcześniejszym rozdziale. Wspomniane podzadania to:

- Należy wykorzystać dostarczony zalążek parsera pliku XML i należy go zintegrować z własnym programem.

4 Założenia

Zakłada się, że na scenie mogą istnieć niezależne prostopadłościany, jak też powiązane ich łańcuchy kinematyczne. Nie ma odgórnego ograniczenia co do ilości tego typu obiektów geometrycznych. Domyślnie prostopadłościany są jednostkowymi sześcianami, których lokalny układ współrzędnych umieszczony jest w ich geometrycznym środku. Jego lokalizację można zmieniać poprzez parametr T_{shift}. Natomiast rozmiar poprzez skalę S. Orientacja obiektu zadawana jest poprzez kąty *Roll-Pitch-Yaw*, tzn. rotacje kolejno względem osi OX, OY i OZ. Finalna pozycja obiektu zadawana jest poprzez wektor translacji T_{trans}. Wymienione współczynniki uwzględnione są w wyliczaniu współrzędnych punktu zgodnie z następującym wzorem

$$\mathbf{p}' = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi) \cdot (S \cdot (\mathbf{p} + T_{shift})) + T_{trans}, \tag{1}$$

gdzie $\mathbf{p}$ oznacza współrzędne jednego z wierzchołków jednostkowego sześcianu wyrażonego w jego lokalnym układzie współrzędnych znajdującym się w geometrycznym środku tegoż sześcianu. Strukturę łańcucha kinematycznego zadaje się poprzez odpowiednią konstrukcję nazwy obiektu. Przyjmuje się, że znak kropki nie jest częścią nazwy. Pełni on rolę separatora między kolejnymi ogniwami łańcucha. Tak więc nazwa

```
Podstawa
```

jest nazwą obiektu, który znajduje się bezpośrednio na scenie. Zadając jego parametry, takie jak wektor translacji, zadajemy jego położenie na scenie. Natomiast nazwa

```
Podstawa.Ramie1
```

Jest nazwą złożoną, która oznacza, że obiekt o nazwie własnej `Ramie1` jest elementem łańcucha kinematycznego i jest w tym łańcuchu następnikiem obiektu `Podstawa`. Zadając parametry dla obiektu `Ramie1` zadajemy jego położenie i rotację względem obiektu `Podstawa`. W komunikacji z serwerem należy posługiwać się tylko i wyłącznie pełnymi nazwami złożonymi danego obiektu.

5 Składnia pliku konfiguracyjnego XML

W pliku konfiguracyjnym znaleźć się muszą dwa główne zbiory informacji:

- Jakie wtyczki mają zostać załadowane. Wtyczki identyfikowane są poprzez pełną nazwę biblioteki. Zbiór wtyczek zawarty jest w treści elementu `<Plugins>` (patrz przykład pliku w dalszej części opisu). Specyfikacja pojedynczej wtyczki zawarta jest w atrybucie `Name` elementu `<Lib>`. Zawiera on nazwę biblioteki dzielonej.

- Jakie prostopadłościany zostaną wykorzystane wraz z podaniem ich parametrów i struktury kinematycznej poprzez odpowiednią konstrukcję nazwy. Zbiór tych obiektów opisany jest w treści elementu `Objects`. Pojedynczy prostopadłościan opisany jest przez atrybuty elementu `Cube`. Atrybuty te odpowiadają parametrom użytym we wzorze (1). Ich postać podana jest poniżej. Nie wszystkie one muszą występować w liście atrybutów danego elementu. Wówczas przyjmowane są wartości domyślne.

  `Name` – pełna nazwa obiektu. Atrybut ten jest obowiązkowy.

  `Shift` – odpowiada parametrowi T_{shift}. Jego wartością są trzy współrzędne (x, y, z), np.

  ```
  Shift="0 -0.3 -0.4"
  ```

  Domyślna wartość atrybutu to `"0 0 0"`.

  `Scale` – odpowiada parametrowi S. Zawiera współczynniki skali względem poszczególnych osi. Wartości te nie mogą być ujemne. Dopuszczalny jest przypadek, gdy wszystkie współczynniki skali będą równe 0. Zapis analogiczny jak w przypadku atrybutu `Shift`. Domyślna wartość atrybutu to `"1 1 1"`.

  `RotXYZ_deg` – zawiera wartości kątów ϕ, θ oraz ψ. Są one wyrażone w stopniach. Zapis analogiczny jak w przypadku atrybutu `Shift`. Domyślna wartość atrybutu to `"0 0 0"`.

  `Trans_m` – odpowiada parametrowi T_{trans}. Zakłada się, że wartości są wyrażone w metrach. Zapis analogiczny jak w przypadku atrybutu `Shift`. Domyślna wartość atrybutu to `"0 0 0"`.

  `RGB` – definiuje kolor obiektu we współrzędnych RGB. Domyślna wartość atrybutu to `"128 128 128"`.

Przykładowa postać pliku:

```
<Config>
 <Plugins>
    <Lib Name="libInter4Move.so"/>
    <Lib Name="libInter4Rotate.so"/>
    <Lib Name="libInter4Pause.so"/>
 </Plugins>
 <Objects>
    <Cube Name="Podstawa" Shift="0 -0.5 0" Scale="0.7 0.7 0.2"
                          RotXYZ_deg="10 0 0" Trans_m="0.1 0 0" RGB="0 0 255"/>
    <Cube Name="Podstawa.Ramie1"  Shift="0 -0.5 0"  Scale="0.1 0.1 0.6" Trans_m="0 0.7 0"/>
    <Cube Name="Podstawa.Ramie1.Ramie2" Shift="0 -0.5 0"  Scale="0.1 0.1 0.6"
                                        Trans_m="0 0.7 0"/>
 </Objects>
</Config>
```

6 Protokół komunikacji z serwerem

Domyślnie serwer nasłuchuje połączeń na porcie 6217. Do serwera można przesłać cztery typy poleceń, które mają postać tekstową i muszą kończyć się znakiem przejścia do nowej linii (’\n’). Polecenia akceptowane przez serwer to: `AddObj` (dodanie obiektu w postaci prostopadłościanu), `UpdateObj` (aktualizacja parametrów obiektu), `Clear` (usunięcie wszystkich elementów sceny), `Close` (zakończenie połączenia). Ogólna idea polega na tym, że do serwera wysyła się informację o lokalizacji poszczególnych obiektów prostopadłościennych aktualizują tym samym wcześniejszy zapis (polecenie `Update`). W przypadku, gdy dany obiekt jest na początku listy łańcucha kinematycznego lub występuje samodzielnie na scenie, przesłane parametry określają jego położenie bezpośrednio na scenie. W innym przypadku określają one położenie względem elementu go poprzedzającego w danym łańcuchu. W dalszej części przedstawiona jest składnia poszczególnych poleceń.

6.1 Polecenie AddObj

Polecenie to pozwala w scenie wstawić nowy obiekt prostopadłościenny i zadać jego parametry. Może on zostać dodany do sceny jako samodzielny obiekt lub jako element istniejącego już łańcucha kinematycznego. Składnia polecenia ma postać:

`AddObj Name=`*Kwalifikowana_Nazwa_Obiektu* {*Parametry_i_Wartości*} **`\n`**

Znak ’\n’ symbolizuje sekwencję znaków przejścia do nowej linii. Spacje między ostatnim parametrem, a znakiem przejście do nowej linii nie są wymagane w żadnym z poleceń. Kolejność parametrów jest dowolna. Kwalifikowana nazwa obiektu jest obligatoryjna. Składa się on z ciągu znaków *niebiałych*. Nazwa pojedynczego obiektu nie może zawierać znaku kropki (’.’), który pełni rolę separatora. Pragnąc lepiej zobrazować postać i znaczenie tego typu nazwy załóżmy, że na scenie znajdują się trzy obiekty o nazwach własnych: `Podstawa`, `Ramie1` oraz `Ramie2`. Załóżmy dalej, że obiekty te tworzą łańcuch kinematyczny `Podstawa-Ramie1-Ramie2`. Pełne kwalifikowane nazwy tych obiektów to:

```
Podstawa
Podstawa.Ramie1
```

`Podstawa.Ramie1.Ramie2`

Parametry opisujące geometrię obiektu są opcjonalne. Jeżeli nie wystąpią one w liście parametrów, to przyjmowane są wartości domyślne, analogicznie jak w przypadku konfiguracyjnego pliku XML. Parametry mają analogiczne nazwy i znaczenie jak w przypadku pliku XML. Mają też te same wartości domyślne. Jedyną różnicą jest zapis wartości. Ma on postać zapisu wektorowego, np.

`Shift=(0,-0.3,-0.4)`

Poniżej znajduje się przykład trzech poleceń, które *budują* ramię robota. Należy zauważyć, że ich kolejność jest istotna. Chcąc umieścić kolejny element w łańcuchu kinematycznym, wcześniejsze muszą już istnieć.

```
AddObj Name=Podstawa Scala=(0.5,0.5,0.15)\n
AddObj Name=Podstawa.Ramie1 Shift=(-0.5,0,0) Scala=(1,0.3,0.3) Trans_m(0.15,0,0)\n
AddObj Name=Podstawa.Ramie1.Ramie2 Shift=(-0.5,0,0) Scala=(0.8,0.3,0.3) Trans_m(1,0,0)\n
```

6.2 Polecenie UpdateObj

Polecenie to może być wykonane tylko i wyłącznie dla istniejących już obiektów na scenie. Powoduje ono aktualizację parametrów. Polecenia `AddObj` i `Update` mają identyczną składnię, a tym samym listę parametrów. Składnia polecenia przedstawiona jest poniżej.

`UpdateObj Name=`*Kwalifikowana_Nazwa_Obiektu* {*Parametry_i_Wartości*} **`\n`**

6.3 Polecenie Clear

Polecenie to powoduje usunięcie wszystkich obiektów ze sceny. Warto je wysłać jako pierwsze polecenie. Przydaje się to wtedy, gdy testujemy program klienta i uruchamiamy go po raz drugi. Dzięki temu poprzednia zawartość sceny zostanie usunięta i zaczynamy wszystko od nowa. Polecenie składa się tylko z jednego wyrazu, jak niżej.

`Clear \n`

Spacje między słowem, a znakiem przejście do nowej linii nie są wymagane w żadnym z poleceń.

6.4 Polecenie Close

Polecenie to powoduje zamknięcie połączenia z serwerem. Aplikacja przed zakończeniem swojej pracy obligatoryjnie powinna wysłać to polecenie do serwera. Składa się ono tylko z jednego wyrazu, jak niżej.

`Close \n`

Poprawne zakończenie połączenia pozwala zwolnić dany port i powtórnie uruchomić sam program lub serwer. Jeśli tak się nie stanie, to dane gniazdo będzie zajęte przez system przez kilka minut. Po czym zostanie zwolnione. Poprawne zamknięcie połączenia pozwala uniknąć tej zwłoki.