Ćwiczenie 3. Transformacje w 3D. Rzutowanie. Translacja, skalowanie i obrót. Stos macierzy. Hierarchia obiektów.

Witold Alda

Spis treści

1 Co chcemy zrobić?

Celem tego, dosyć rozbudowanego, ćwiczenia jest przeanalizowanie podstawowych operacji w przestrzeni 3D, obejmujących rzutowanie prspektywiczne, podstawowe transformacje (translację, skalowanie, obrót) oraz przechodzenie pomiędzy układami współrzędnych z wykorzystaniem stosu macierzy przekształceń. W ćwiczeniu opieramy się na przykładach z tutoriali 04, 05 i 06.

2 Macierz rzutowania

Przyjmujemy ją bez wyprowadzania. Podstawowa postać macierzy zaproponowana w poręczniku i w tutorialu jest następująca:

$$\begin{bmatrix} S & 0 & 0 & 0 \\ 0 & S & 0 & 0 \\ 0 & 0 & \frac{F+N}{N-F} & \frac{2NF}{N-F} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

gdzie S jest współczynnikiem skalującym oznaczonym w programie jako `fFrustumScale`, a N i F oznaczją bliższą i dalszą płaszczyznę odcięcia, `fzNear` i `fzFar`. Tak opisana macierz zakłada, że rzutowanie odbywa się na środek ekranu, oraz że obszar rzutowania na płaszczyznę `z= fzNear` jest kwadratem, który sięga od -1 do +1 wzdłuż osi x i y. Macierz ta jest użyta w przykładach tutoriala `Tut 04`. W `MatrixPerspective` macierz jest wyliczona w zasadniczym programie i przesłana do shaderów jako uniform, natomiast w `ShaderPerspective` macierz jest wyliczana wprost w shaderze. To ostatnie rozwiązanie wydaje się nieco gorsze, bo jednak wymusza wielokrotne liczenie macierzy na GPU.

Czasami stosuje się bardziej ogólną postać macierzy rzutowania wykorzystując informację, że obszar rzutowania jest prostokątem ograniczonym czterema wartościami: L, R, T, B (odpowiednio: left, right, top, bottom) Wtedy macierz przyjmuje postać:

$$\begin{bmatrix} \frac{2N}{R-L} & 0 & \frac{R+L}{R-L} & 0 \\ 0 & \frac{2N}{T-B} & \frac{T+B}{T-B} & 0 \\ 0 & 0 & \frac{F+N}{N-F} & \frac{2NF}{N-F} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

, co akceptujemy również bez wyprowadzenia.

2.1 Co do zrobienia na zajęciach?

1. Proszę zmodyfikować przykład z pliku `MatrixPerspective.cpp`, tak żeby można było z klawiatury przesuwać obiekt wzdłuż osi x i y oraz by można było zmieniać wartości macierzy rzutowania `fFrustumScale`, `fzNear` i `fzFar` (trzeba wtedy ustawianie i przesyłanie macierzy rzutowania przepchąć z funkcji InitializeProgram() dalej, np. do `display()`). Można wtedy wygodnie zobaczyć jak zmiany macierzy wpływają na to co widzimy.

2. Proszę rozszerzyć postać macierzy rzutowania, zgodnie z wzorem podanym wyżej i ocenić jakie otrzymujemy dzięki temu możliwości w manipulowaniu rzutowaniem.

3 Bufor głębokości

Przykłady w tutorialu `Tut 05` poświęcone są buforowi głębokości i zasłanianiu dalszych obiektów przez bliższe. Proszę uruchomić i obejrzeć przykłady, jednak na zajęciach nie będziemy ich modyfikować. Zwłaszcza proszę popatrzeć na przykład `DepthBuffering`.

Warto w nim zwrócić uwagę na funkcje związane z buforem głębokości i jego testowaniem:

- `glEnable(GL_DEPTH_TEST)` – uaktywnienie bufora głębokości
- `glDepthFunc()` – określenie warunków zasłaniania obiektów – proszę je sprawdzić w Reference Pages na `www.opengl.org`. Można je tak ustawić, że dalsze obiekty zasłaniają bliższe - choć to raczej wyjątkowa sytuacja.

Użyty jest tu również inny sposób rysowania trójkątów siatki. Zamiast stosowanej wcześniej funkcji `glDrawArray(GL_TRIANGLES,...)`, która rysuje na podstawie na podstawie tablicy wierzchołków (każde kolejne trzy wierzchołki budują nowy trójkąt), proponuje się tu funkcję `glDrawElements(GL_TRIANGLES,...)` która buduje kolejne trójkąty na podstawie listy wierzchołków (tak jest w przykładzie `DepthBuffering`.

4 Transformacje geometryczne. Składanie transformacji

W tutorialu `Tut 06` są dostępne cztery przykłady. Trzy z nich demostrują działanie trzech podstawowych transformacji: przesunięcia, skalowania i obrotu. Na scenie jest kilka obiektów i każdy z nich jest inaczej transformowany. Odbywa się to w ten sposób, że w funkcji `display()`, w pętli jest przesyłana macierz transformacji danego obiektu, a następnie wskazywany jest sam obiekt (tu za pomocą `glDrawElements()`.

W przykładach w `Tut 06` użyto funkcji z biblioteki glm. Są to funkcje, które pozwalają operować na wektorach i macierzach w sposób zbliżony do zastosowanego w GLSL. Opis (niezbyt długi) biblioteki glm jest na jej stronie

`http://glm.g-truc.net/glm-0.9.4.pdf`.

4.1 Co do zrobienia na zajęciach?

W oparciu o przykłady `Translation`, `Scale` i `Rotations` proszę zestawić program, który wykonywałby dla obiektów na scenie złożenie tych trzech transformacji - w dowolnej kolejności.

5 Stos macierzy. Obiekty hierarchiczne

Proszę uruchomić i przeanalizować działanie programu `Hierarchy` z tutoriala `Tut 06`.