

Politechnika Warszawska
Wydział Elektryczny
Kierunek Informatyka stosowana

KAMERA WIRTUALNA

Sprawozdanie z pierwszej części projektu
Grafika Komputerowa

Mateusz Ciupa
291062

1 kwietnia 2020

Spis treści

1	Wstęp	2
2	Opis obiektów	2
3	Operacje kamery	2
3.1	Translacja	3
3.2	Rotacja	3
3.3	Zoom	5
4	Odwzorowanie punktu 3D na powierzchni 2D	7
5	Wnioski	7

1 Wstęp

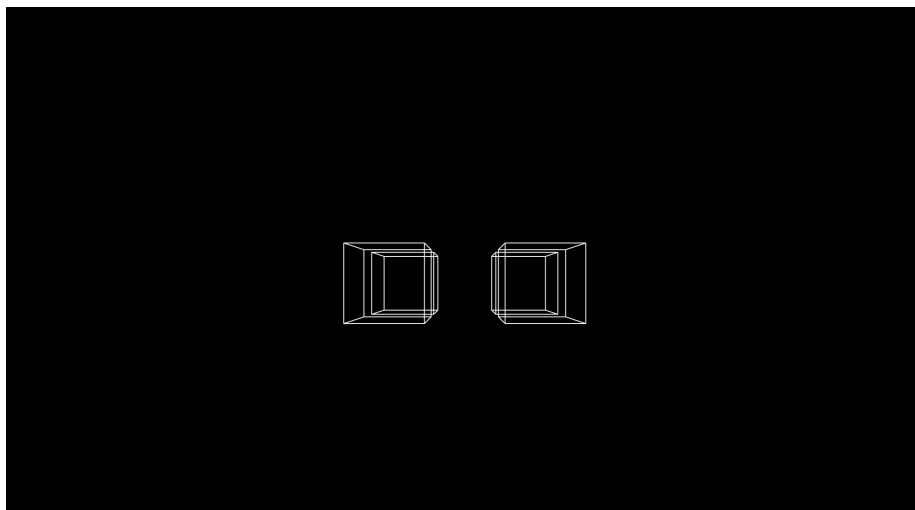
Celem pierwszej części projektu było zaimplementowanie kamery wirtualnej w dowolnym języku. Kamera powinna umożliwiać dokonywania translacji (lewo, prawo, góra i dół), rotacji wokół trzech osi oraz operacji zoom. Ćwiczenie zostało wykonane przy użyciu języka **JavaScript** oraz elementu **Canvas** języka HTML. Materiały (w tym technika wykonania) zostały oparte na książce: *Fundamentals Of Computer Graphics* - Peter, Shirley, Steve Marschner.

2 Opis obiektów

Obiekty, które są rysowane na ekranie, są bryłami, które składają się z krawędzi (dwóch punktów połączonych linią). Wszystkie punkty są określone przez współrzędne jednorodne trójwymiarowej przestrzeni ($x, y, z, 1$).

3 Operacje kamery

Operacje są wykonywane poprzez przemnożenie wszystkich punktów tworzących obiekty z odpowiednimi macierzami (4×4).



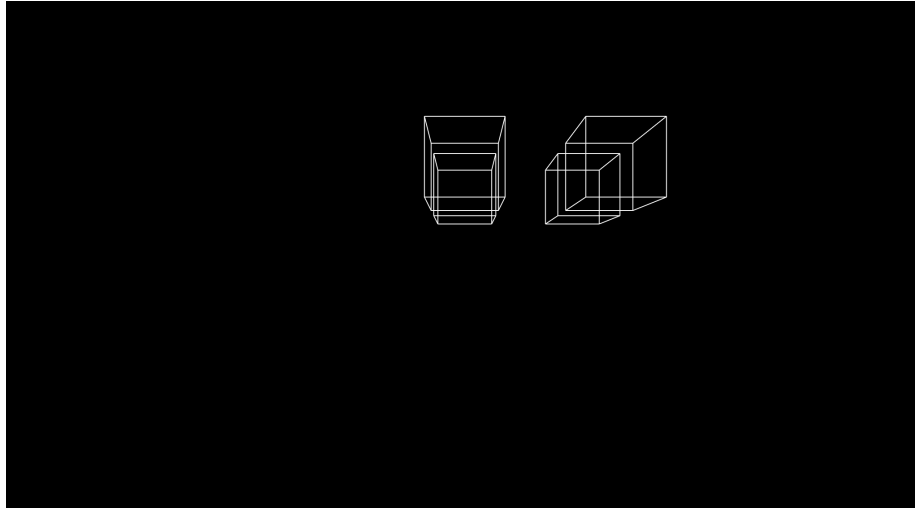
Rysunek 1: Początkowe wyświetlenie obiektów

3.1 Translacja

Macierz translacji:

$$M_t = \begin{bmatrix} 1 & 0 & 0 & x_t \\ 0 & 1 & 0 & y_t \\ 0 & 0 & 1 & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Gdzie x_t, y_t, z_t – kolejne współrzędne translacji obiektu.



Rysunek 2: Obiekty poddane translacji x_t, y_t

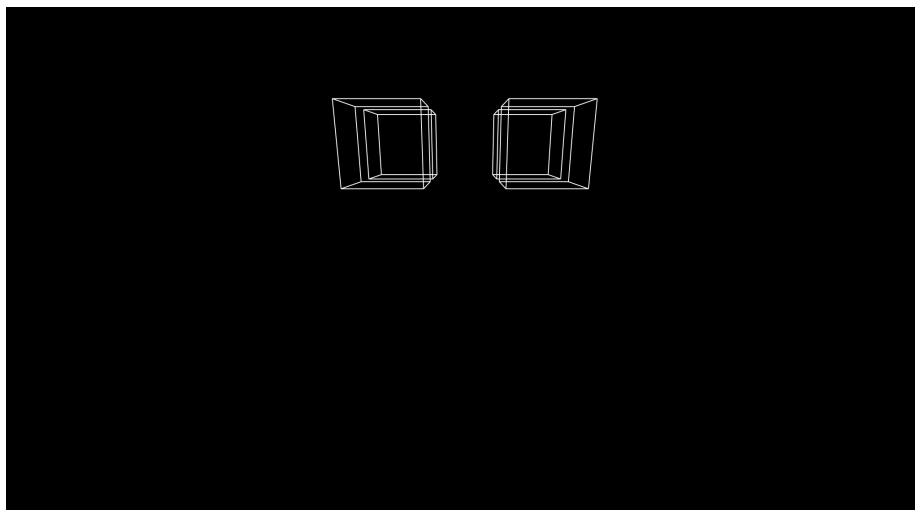
3.2 Rotacja

Macierze rotacji:

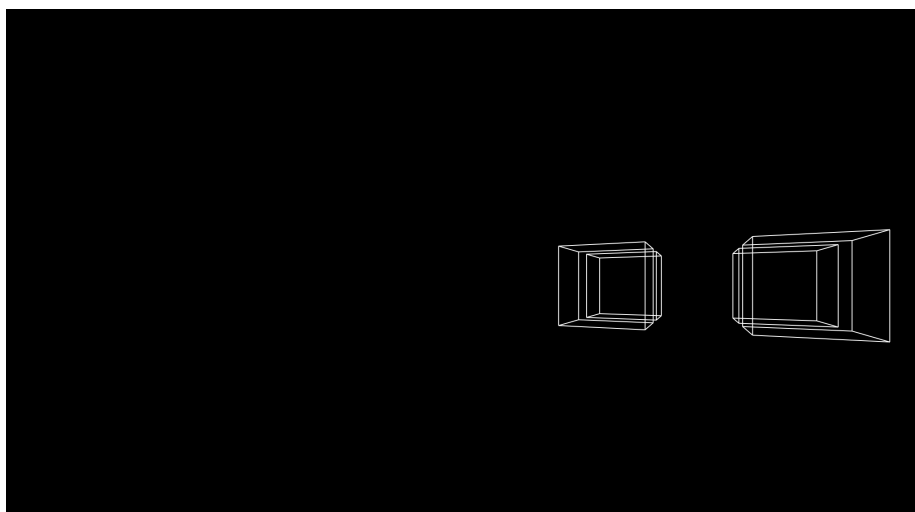
$$M_{rx} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$M_{ry} = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

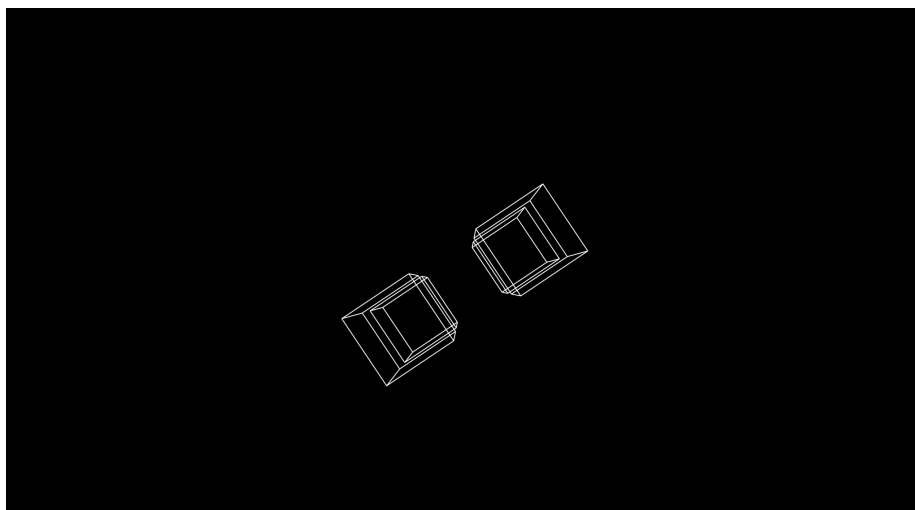
$$M_{rz} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$



Rysunek 3: Obiekty poddane rotacji względem osi x



Rysunek 4: Obiekty poddane rotacji względem osi y



Rysunek 5: Obiekty poddane rotacji względem osi z

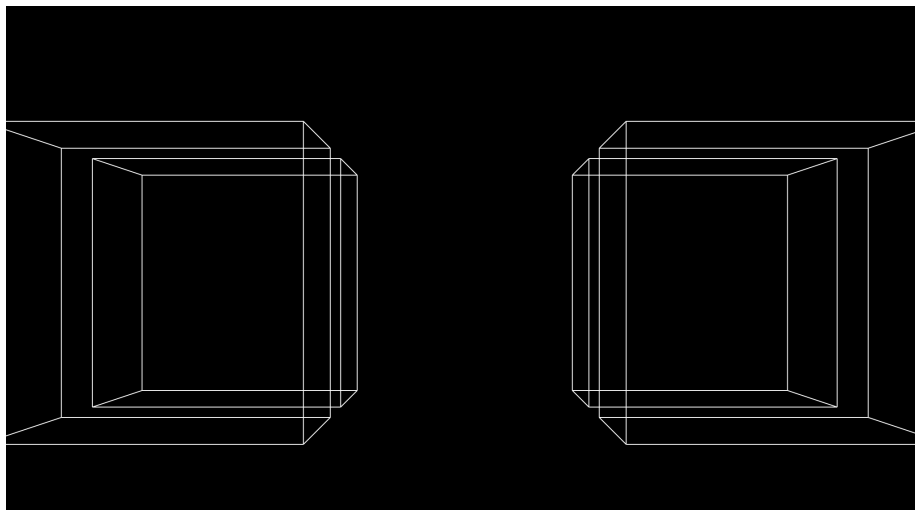
Gdzie x, y, z – osie rotacji.

3.3 Zoom

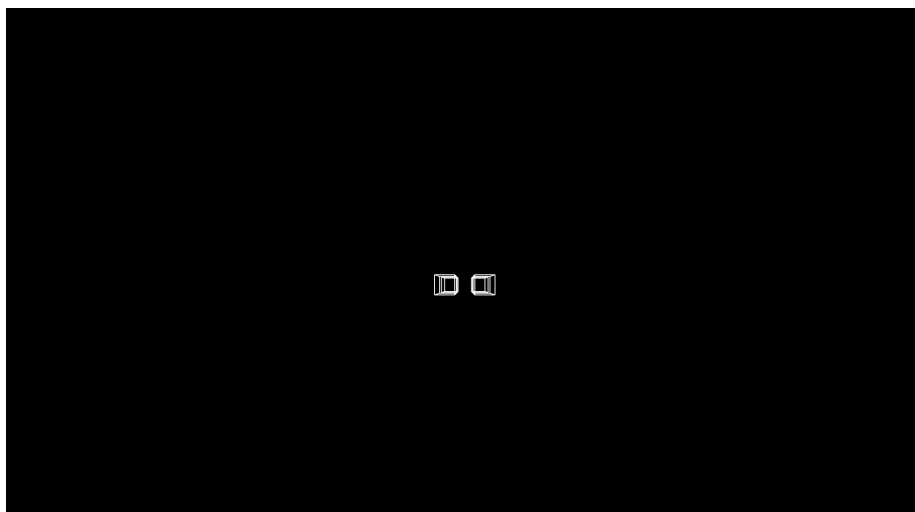
Macierz operacji zoom:

$$M_t = \begin{bmatrix} z & 0 & 0 & 0 \\ 0 & z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Gdzie z – stopień powiększenia.



Rysunek 6: Obiekty poddane operacji zoom ($z > 1$)



Rysunek 7: Obiekty poddane operacji zoom ($z > 0$ i $z < 1$)

4 Odwzorowanie punktu 3D na powierzchni 2D

Odwzorowanie następuje poprzez przemnożenie punktów z kolejnymi macierzami:

1. Przejście z układu współrzędnych kamery do *canonical view volume* wraz z transformacją perspektywistyczną (*orth_to_canonical*perspective_matrix*).

$$M_{per} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{l+r}{l-r} & 0 \\ 0 & \frac{2n}{t-b} & \frac{b+t}{b-t} & 0 \\ 0 & 0 & \frac{f+n}{n-f} & \frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6)$$

Gdzie l, r, t, b, n, f – wymiary *orthographic view volume* (left, right, top, bottom, near, far).

2. *Viewport transformation* – przejście z *canonical view volume* do wymiarów canvasa (bez zmiany z).

$$M_{vp} = \begin{bmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x-1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y-1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Gdzie n_x, n_y – rozmiary canvasa.

5 Wnioski

W wykonaniu ćwiczenia niepotrzebnie wykorzystano transformację z układu współrzędnych kamery do *canonical view volume* – można było zastosować prostsze macierze. Będzie to przydatne w wykonaniu drugiej części projektu do usuwania powierzchni zakrytych.