

1 Informacje wstępne

Projekt tworzony jest w ramach przedmiotu *Grafika komputerowa* realizowanego podczas 6. semestru kierunku *Informatyka Stosowana, wydział Elektryczny, na Politechnice Warszawskiej*. Projekt stanowi praktyczne wykorzystanie poznawanych podczas kursu algorytmów i metod używanych w grafice komputerowej.

2 Cel projektu

Celem projektu jest stworzenie programu (wirtualnej kamery) pozwalającego na poruszanie się w trójwymiarowej przestrzeni, w której znajdują się obiekty zbudowane z odcinków. Kamera ma umożliwiać przemieszczanie się w trzech osiach, obrót wokół trzech osi oraz skorzystanie z funkcji „zoom”. Scena po której porusza się kamera jest predefiniowana i obserwator nie ma na nią wpływu.

3 Stos technologiczny

Projekt został stworzony w języku C# z wykorzystaniem platformy .NET Framework. Rozwiązanie zostało stworzone według struktury WPF (Windows Presentation Foundation), która ułatwiła tworzenie aplikacji okienkowej. Projekt nie korzysta z zewnętrznych bibliotek. Wykorzystane zostały jedynie biblioteki systemowe, dostarczane wraz z platformą .NET Framework.

4 Opis działania programu

Naciśnięcie klawisza funkcyjnego powoduje zlecenie wykonania akcji w najbliższej generowanej klatce. Klatki są generowane co 20ms. Dopóki użytkownik nie puści klawisza funkcyjnego, dana akcja będzie zlecana w każdej kolejnej klatce.

Podczas każdej klatki dzieje się następujący ciąg czynności:

1. Program sprawdza, jakie akcje zostały zlecone (zoom, obrót, ruch);
2. Program wymnaża macierze przekształceń przez macierz kombinacji wszystkich zmian dla kolejnych wybranych akcji;
3. Pozycja punktów w przestrzeni trójwymiarowej zostaje uaktualniona poprzez przemnożenie macierzy kombinacji wszystkich zmian przez każdy punkt;
4. Na podstawie nowych punktów trójwymiarowych w procesie rzutowania perspektywicznego zostają wygenerowane nowe punkty w przestrzeni dwuwymiarowej;
5. Punkty w przestrzeni dwuwymiarowej oraz połączenia między nimi są rysowane na ekranie.

5 Operacje i algorytmy

W celu przeprowadzenia operacji (przesunięcia i obrotu) należy wymnożyć macierz przekształcenia przez każdy punkt w przestrzeni trójwymiarowej zgodnie ze wzorem:

$$\begin{bmatrix} x'_p \\ y'_p \\ z'_p \\ 1 \end{bmatrix} = M_p * \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} \quad (1)$$

Gdzie:

- M_p – macierz przekształcenia;
- (x_p, y_p, z_p) – współrzędne punktu przed przekształceniem;
- (x'_p, y'_p, z'_p) – współrzędne punktu po przekształceniu.

Należy pamiętać o normalizacji macierzy. W przypadku, gdy po wymnożeniu, w ostatnim wierszu wyjściowej macierzy znajduje się wartość inna niż „1”, należy podzielić wszystkie elementy wektora przez tę wartość.

5.1 Składanie przekształceń

W programie zostało zastosowane składanie macierzy przekształceń, dzięki czemu można wykonywać wiele operacji na raz. Skraca to czas w stosunku do wykonywania tych operacji sekwencyjnie. Została zastosowana następująca kolejność składania przekształceń:

1. obrót wokół osi X o kąt ϕ ;
2. obrót wokół osi Y o kąt ϕ ;
3. obrót wokół osi Z o kąt ϕ ;
4. translacja o wektor $[T_x \ T_y \ T_z]$.

Składanie realizowane jest według następującego wzoru:

$$P'' = M_2 * M_1 * P = M_2 * P' = M_{21} * P \quad (2)$$

Gdzie:

- P, P', P'' – współrzędne punktu przed, po pierwszym i po drugim przekształceniu;
- M_1, M_2 – macierz przekształcenia pierwszego i drugiego;
- M_{21} – macierz połączonych przekształceń.

5.2 Translacja o wektor

Translacja jako przekształcenie zostaje dodana do głównego przekształcenia jednocześnie dla wszystkich osi. Macierz translacji zostaje przemnożona przez macierz kombinacji wszystkich zmian. Dla translacji o wektor $[T_x \ T_y \ T_z]$ macierz przekształcenia wygląda następująco:

$$M_{pt} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

5.3 Obrót

Macierz obrotu dla każdej osi jest mnożona przez główną macierz przekształcenia osobno. Poniżej znajdują się macierze przekształceń obrotu dla kątu ϕ .

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

$$M_{poy} = \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$M_{poz} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 & 0 \\ \sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

5.4 Zoom

Opcja „zoom” nie wymaga wykorzystania żadnej macierzy przekształceń. Zmienia ona jedynie parametr rzutowania perspektywicznego FOV (field of view).

5.5 Rzutowanie perspektywiczne

Rzutowanie perspektywiczne wymaga przemnożenia macierzy rzutowania przez każdy punkt w ściętym stożku widzenia, a następnie znormalizowania. Poniższa macierz została zaczerpnięta ze sposobu w jaki radzi sobie z tym zagadnieniem biblioteka [OpenGL](#).

$$M_{rz} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (7)$$

- n – bliższa płaszczyzna odcinania;
- f – dalsza płaszczyzna odcinania;
- $ratio$ – proporcje wyświetlanego ekranu;
- FOV – pole widzenia;
- $t = \tan(\frac{FOV}{2}) * n$;
- $r = t * ratio$;
- $b = -t$;
- $l = -r$.

6 Obsługa programu

Program po uruchomieniu automatycznie ustawia kamerę na wygenerowanej scenie i pozwala na swobodną eksplorację. Obserwatorem sterować można za pomocą następujących klawiszy funkcyjnych:

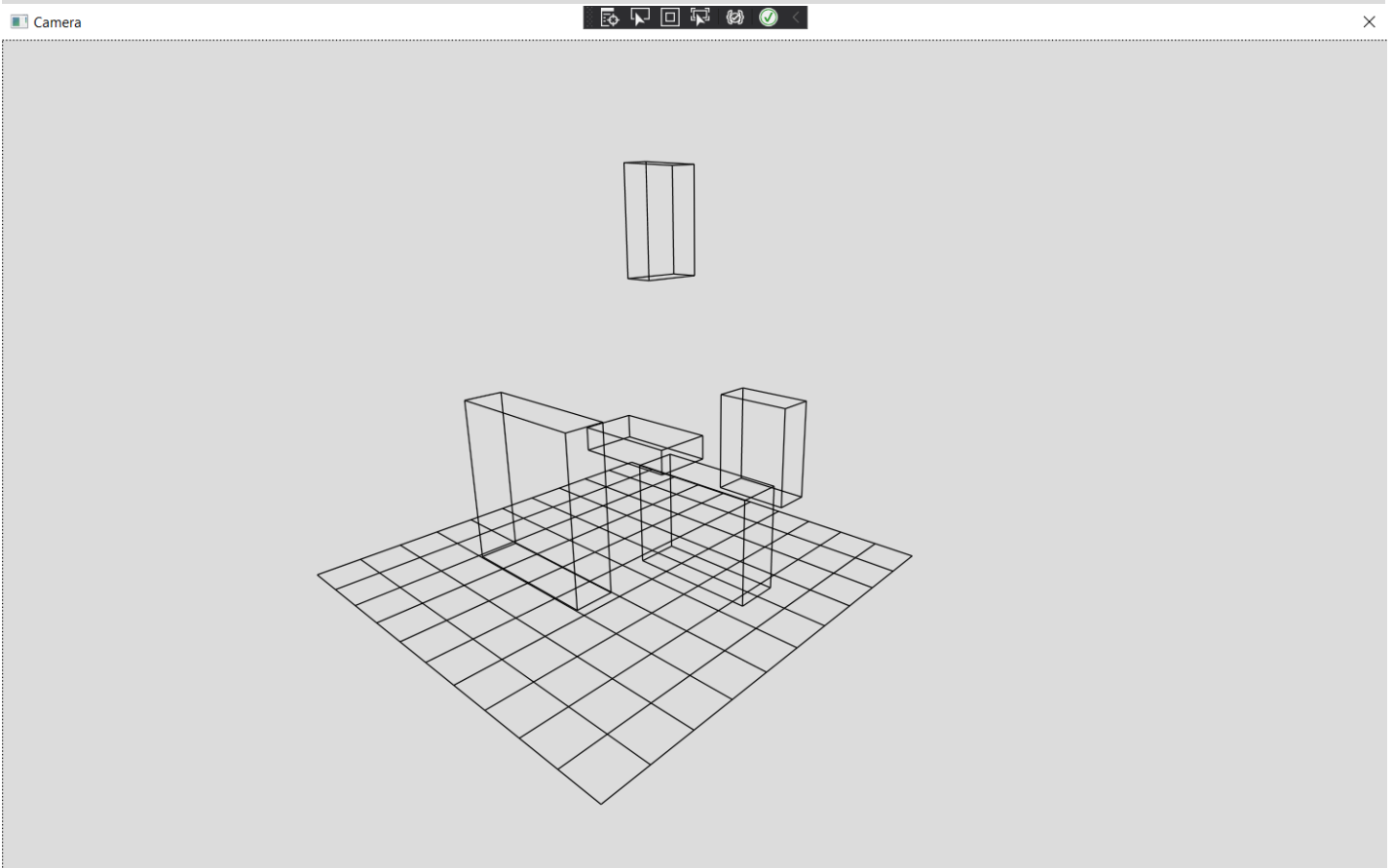
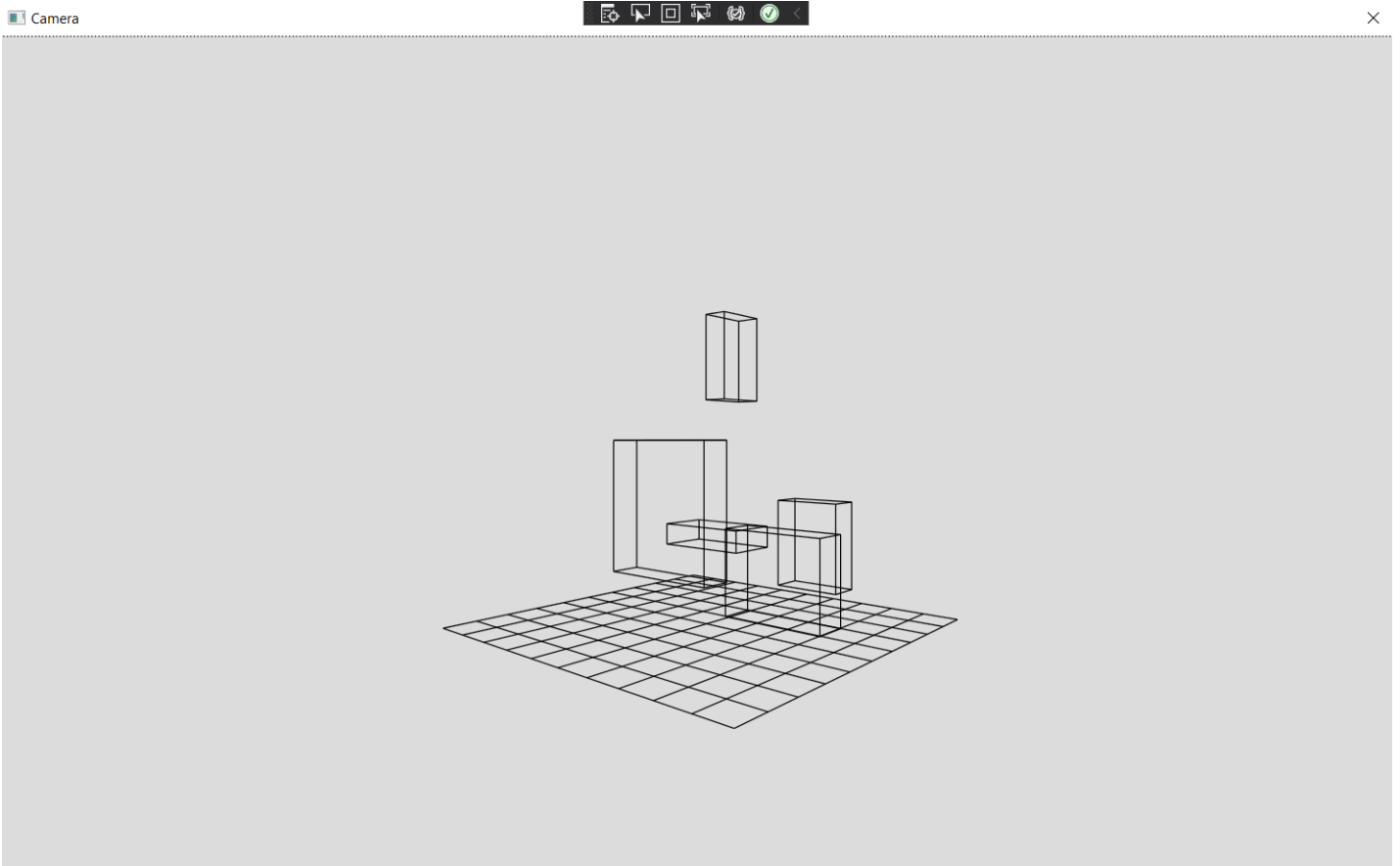
- W/S – ruch w osi X (przód/tył);
- Q/E – ruch w osi Y (góra/dół);
- A/D – ruch w osi Z (lewo/prawo);
- I/K – obrót wokół osi X (śruba prawo/lewo skretna);
- J/L – obrót wokół osi Y (śruba prawo/lewo skretna);
- U/O – obrót wokół osi Z (śruba prawo/lewo skretna);
- Z/X – zoom (zbliżenie/oddalenie).

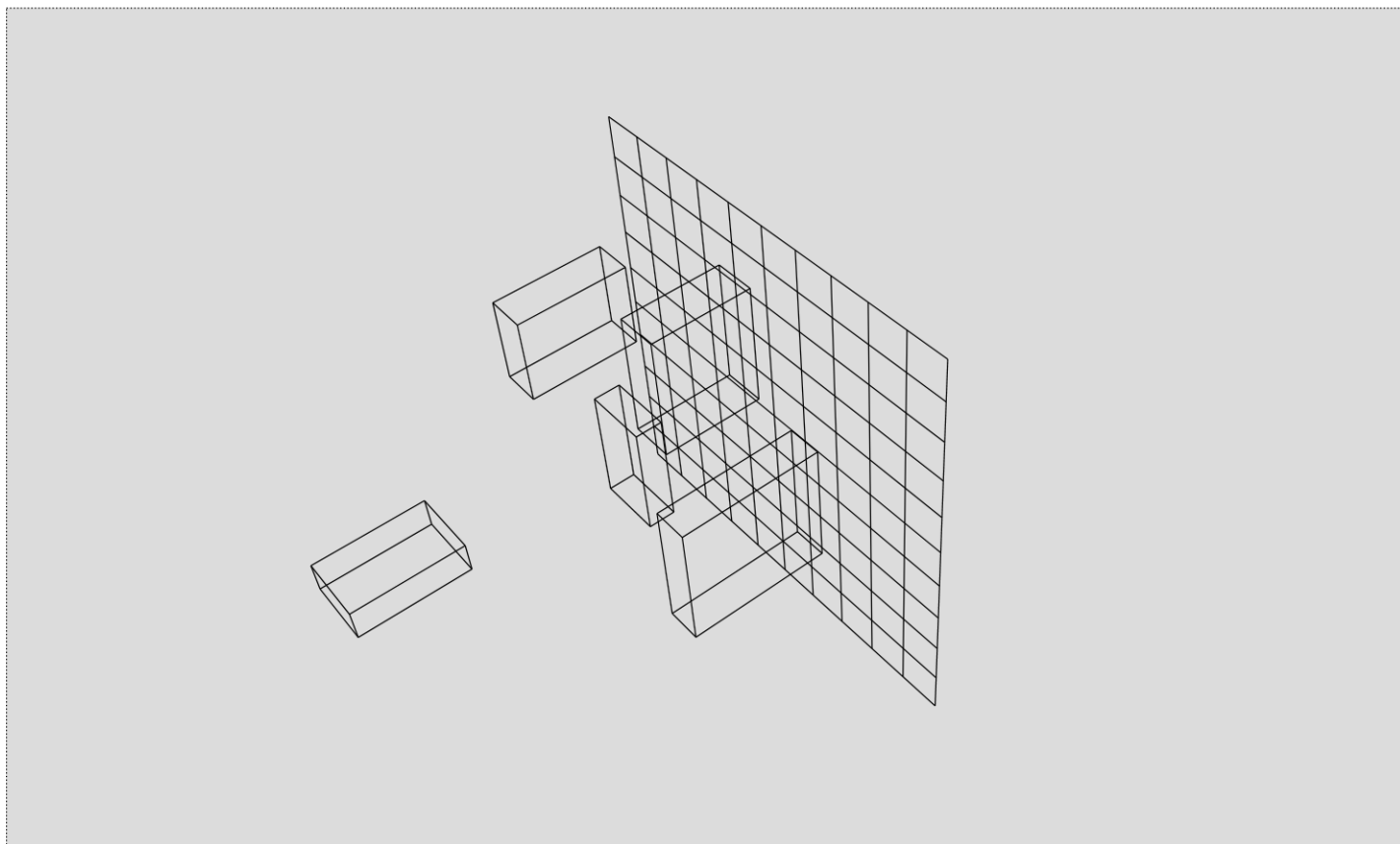
7 Przykład działania programu

Poniżej zamieszone zostały zrzuty ekranu różnych ujęć z dwóch scen.

7.1 Scena pokazowa 1.

7.2 Scena 1.





7.3 Scena 2.

