Sprawozdanie – System Wirtualnej Kamery

Adrian Rybaczuk 318483

7 kwietnia 2025

Spis treści

1 Wstęp

Projekt polega na implementacji systemu wirtualnej kamery. Celem jest przedstawienie sceny przy pomocy wirtualnej kamery. Użyłem do tego podejścia z wykorzystaniem 2 układów, układu kamery oraz układu sceny.

2 Transformacja obiektu 3D

W tej sekcji przedstawiamy szczegółowy opis procesu przekształcania obiektu 3D z jego pierwotnego umiejscowienia w scenie do ostatecznego obrazu na ekranie. Wszystkie operacje wykonujemy w przestrzeni jednorodnej (homogenicznej), co umożliwia łączenie transformacji translacji, rotacji oraz skalowania w jedną macierz.

2.1 Krok 1: Umiejscowienie Obiektu na Scenie (Modeling)

Obiekt w scenie jest pierwotnie opisany w swoim lokalnym układzie współrzędnych. Aby umiejscowić go w przestrzeni świata, stosujemy modelową macierz transformacji M,

$$M = T \cdot R \cdot S,$$

gdzie:

- T – macierz translacji,
- R – macierz rotacji,
- S – macierz skalowania.

Wszystkie współrzędne obiektu zapisujemy jako wektory jednorodne:

$$P_{model} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

W wyniku przekształcenia mamy:

$$P_{world} = M \cdot P_{model}.$$

2.2 Krok 2: Przelozenie Obiektu z Widoku Sceny na Widok Kamery (View Transformation)

Następnie, aby przejść z układu świata (sceny) do układu kamery, stosujemy macierz widoku V. Jest ona konstruowana na bazie pozycji kamery E (ang. eye), punktu, na który kamera patrzy, oraz wektorów definiujących orientację kamery. Metoda `lookAt` definiuje trzy ortonormalne wektory:

- **u** – wektor określający prawą stronę kamery,
- **v** – wektor skierowany do góry,

- $\mathbf{n}$ – wektor przeciwległy do kierunku patrzenia.

Macierz widoku ma postać:

$$V = \begin{bmatrix} u_x & u_y & u_z & -\mathbf{u} \cdot E \\ v_x & v_y & v_z & -\mathbf{v} \cdot E \\ n_x & n_y & n_z & -\mathbf{n} \cdot E \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Przekształcenie do układu kamery wykonujemy jako:

$$P_{camera} = V \cdot P_{world}.$$

2.3 Krok 3: Przelozenie Obiektu z Widoku Kamery na Widok 2D (Rzutowanie Perspektywiczne)

Aby odwzorować trójwymiarowy widok kamery na dwuwymiarowy obraz, stosujemy macierz rzutowania P korzystającą z zasad perspektywicznego rzutowania. Nowe współrzędne (x', y', z', w') są obliczane jako:

$$P' = P \cdot P_{camera}.$$

Następnie wykonujemy perspektywiczne dzielenie przez w':

$$(x_{ndc}, y_{ndc}, z_{ndc}) = \left(\frac{x'}{w'}, \frac{y'}{w'}, \frac{z'}{w'} \right),$$

gdzie $(x_{ndc}, y_{ndc}, z_{ndc})$ to współrzędne w przestrzeni znormalizowanej (ND). Macierz perspektywiczna P może mieć postać:

$$P = \begin{bmatrix} \frac{f}{a} & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & \frac{N+F}{N-F} & \frac{2NF}{N-F} \\ 0 & 0 & -1 & 0 \end{bmatrix},$$

gdzie:

- $f = \frac{1}{\tan(\theta/2)}$ – zależny od kąta widzenia θ,
- a – współczynnik proporcji ekranu,
- N i F – odległości do płaszczyzny bliskiej (near) i dalszej (far).

2.4 Krok 4: Przelozenie Obiektu na Viewport (Viewport Transformation)

Ostatecznie, współrzędne z przestrzeni znormalizowanej, gdzie $x_{ndc}, y_{ndc} \in [-1, 1]$, są mapowane do rzeczywistych współrzędnych ekranu. Dla ekranu o szerokości W i wysokości H stosujemy następujące przekształcenie:

$$x_{screen} = \frac{W}{2}(x_{ndc} + 1), \quad y_{screen} = \frac{H}{2}(1 - y_{ndc}).$$

Zapewnia to, że punkt $(-1, -1)$ trafia do lewego dolnego rogu ekranu, a $(1, 1)$ do prawego górnego.

Każdy z tych kroków wykorzystuje macierze 4x4 oraz operacje w przestrzeni jednorodnej, co pozwala na efektywne łączenie transformacji oraz zapewnia spójność obliczeń podczas renderowania scen 3D. Metody te stanowią fundament nowoczesnych algorytmów renderowania w grafice komputerowej.

3 Podsumowanie Matematyczne

System wirtualnej kamery opiera się na fundamentalnych zasadach algebry liniowej:

- **Homogeniczne współrzędne:** Pozwalają na łączenie różnych transformacji (translacji, rotacji, skalowania) w jedną operację.
- **Macierze transformacji:** Umożliwiają precyzyjne operacje na punktach poprzez mnożenie macierzy.
- **Twierdzenie Eulera:** Każdy obrót w przestrzeni można sprowadzić do pojedynczego obrotu wokół ustalonej osi, co upraszcza modelowanie ruchu kamery.
- **Rzutowanie perspektywiczne:** Dzielenie przez w oraz przekształcenie do układu widoku umożliwiają realistyczną projekcję 3D na 2D.

4 Sterowanie Kamerą

System wirtualnej kamery oferuje intuicyjne sterowanie poprzez kombinację klawiatury i myszy, umożliwiając pełną kontrolę nad sześcioma stopniami swobody kamery:

4.1 Translacja (Przesunięcie)

Translacja kamery w przestrzeni 3D jest realizowana za pomocą klawiszy:

- **W/S** - przesunięcie w przód/tył (oś Z)
- **A/D** - przesunięcie w lewo/prawo (oś X)
- **Shift/Ctrl** - przesunięcie w górę/dół (oś Y)

4.2 Rotacja (Obrót)

Rotacja kamery jest kontrolowana za pomocą myszy:

- **Lewy przycisk myszy + ruch** - obrót wokół osi X (pitch) i Y (yaw)
- **Środkowy przycisk myszy + ruch** - obrót wokół osi Z (roll)

4.3 Dodatkowe Kontrolki

- **Kółko myszy** - zoom in/out (zmiana pola widzenia)
- **R** - reset kamery do pozycji początkowej
- **E** - stabilizacja kamery (wyprostowanie)
- **ESC** - wyjście z aplikacji

5 Literatura i Źródła

1. Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1990). *Computer Graphics: Principles and Practice* (2nd ed.). Addison-Wesley.
2. Takahashi, S. (2006). *The Manga Guide to Linear Algebra*. No Starch Press.
3. Lengyel, E. (2004). *Mathematics for Game Programming and Computer Graphics*. CRC Press.
4. Shirley, P., Marschner, S. (2013). *Fundamentals of Computer Graphics* (4th ed.). A K Peters/CRC Press.
5. Pygame Documentation.
6. NumPy Documentation.