

# Konspekt – Grafika Komputerowa

## Ćwiczenie 1 – wirtualna kamera

Adrian Nowosielski

### 1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z podstawowymi translacjami macierzowymi oraz metodami rzutowania na przykładzie tworzenia programu wirtualnej kamery. Wirtualna kamera ma umożliwiać użytkownikowi poruszanie się wzdłuż osi  $x$ ,  $y$ ,  $z$ , umożliwiać wykonanie przybliżenia (zmiany ogniskowej) oraz podstawowe obroty.

### 2. Opis realizacji ćwiczenia

Kolejnym punktem konspektu jest opis podejścia do rozwiązania problemu wirtualnej kamery:

#### 2.1 - Rysowanie sześciątów

Do prezentacji ćwiczenia mam zamiar wykorzystać 4 sześciiany, między którymi będę się poruszać wirtualną kamerą. Na początku zacznę od zdefiniowania odpowiednich wierzchołków sześcianu, potem krawędzi, które występują w bryle. Następnie do narysowania sześcianu wystarczy tylko połączyć krawędzi. Nie będzie to prawdopodobnie wymagać wczytywania położenia odpowiednich wierzchołków z pliku – po prostu zdefiniuję sobie jeden przykładowy, ustalę długość krawędzi – reszta sześciątów będzie wyglądać tak samo, tylko będzie oddalona o krawędź w wybranej osi.

#### 2.2: Ruchy wirtualnej kamery

Następnym problemem do rozwiązania w ćwiczeniu jest implementacja poruszania się wirtualnej kamery. Do rozwiązania problemu mam zamiar wykorzystać dwie macierze:

- **widoku** – odpowiada za przekształcenie sceny w układzie współrzędnych kamery – dzięki tej macierzy wprowadzamy efekt ruchu kamery. Zasadniczo do stworzenia tej macierzy skorzystam z wektorów wzdłuż osi  $Z$  (kierunek patrzenia kamery), wektora  $up$  (wzdłuż osi  $Y$ ) oraz punktu  $position$  (określa położenie kamery w przestrzeni trójwymiarowej – punkt, z którego obserwator patrzy na scenę). Następnie do konstrukcji macierzy wystarczy obliczyć położenie wektora wzdłuż osi  $X$  (mnożenie  $forward$  i  $up$ ) i konstrukcja macierzy widoku.

- **projekcji** – umożliwi przekształcenie punktów w przestrzeni trójwymiarowej do przestrzeni znormalizowanych urządzeń, co umożliwi renderowanie ich na płaskim ekranie. Dzięki tej macierzy poprzez zmienienie parametru ( $field\ of\ view$ ) będziemy w stanie wykonać powiększenie, zmniejszenie.

Następnie w momencie, gdy użytkownik naciśnie przycisk zmieniamy odpowiednio położenie kamery oraz dla każdego wierzchołka sześcianu wykonujemy transformację na podstawie macierzy projekcji oraz widoku.

