Politechnika Warszawska
Wydział Elektryczny
Informatyka Stosowana
19.05.2023

Grafika komputerowa

Projekt "Kamera wirtualna z eliminacją elementów zasłoniętych"

Darya Vasilchyk 317048
Natalia Olszewska 311368

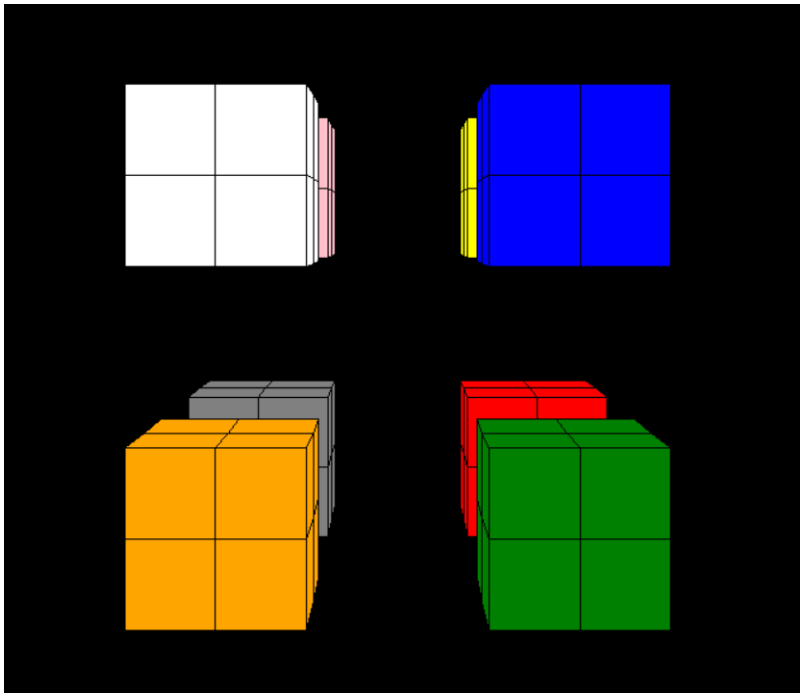
Spis treści

1 Wstęp

Celem projektu "Kamera wirtualna z eliminacją elementów zasłoniętych" jest symulowanie działania kamery wirtualnej w przestrzeni trójwymiarowej. Głównym zadaniem projektu jest stworzenie systemu, który będzie w stanie renderować scenę trójwymiarową z perspektywy kamery i jednocześnie eliminować elementy zasłonięte przez inne obiekty.

W dziedzinie grafiki komputerowej, kamera wirtualna odgrywa kluczową rolę w tworzeniu wizualnych efektów i interakcji w trójwymiarowych środowiskach. Jednak jednym z wyzwań jest odpowiednie renderowanie sceny, tak aby obiekty znajdujące się z przodu były widoczne, a te zasłonięte były pominięte. W tym projekcie zostanie zaimplementowany algorytm malarski (painter's algorithm), który pozwoli na eliminację elementów zasłoniętych i optymalne renderowanie sceny.

W ramach projektu zostanie stworzony kod w języku Python, który będzie symulował działanie kamery wirtualnej dla zdefiniowanej sceny, zawierającej 8 sześcianów. Implementacja będzie obejmować funkcje do przekształceń punktów, eliminacji elementów zasłoniętych oraz renderowania widocznych obiektów. Dodatkowo, projekt uwzględni możliwość obrotu, transformacji i skalowania sześcianów w przestrzeni.



Rysunek 1: Scena z 8 sześcianów

2 Podstawy teoretyczne projektu

Projekt "Kamera wirtualna z eliminacją elementów zasłoniętych" opiera się na kilku podstawowych teoretycznych koncepcjach z dziedziny grafiki komputerowej i przetwarzania obrazu. Poniżej przedstawiam opis tych podstawowych pojęć:

1. Kamera wirtualna: Kamera wirtualna jest abstrakcyjnym modelem, który symuluje działanie prawdziwej kamery w przestrzeni trójwymiarowej. Kamera definiuje perspektywę widoku, kąt widzenia, pozycję oraz orientację w przestrzeni. Dzięki temu możemy renderować sceny trójwymiarowe i tworzyć wizualne efekty.

2. Układ współrzędnych: Układ współrzędnych to system referencyjny, który służy do określania położenia i orientacji obiektów w przestrzeni. Najczęściej stosowanym układem współrzędnych w grafice komputerowej jest układ kartezjański, który składa się z trzech osi: x, y i z. Pozycje obiektów są określane jako kombinacje wartości na tych trzech osiach.

3. Renderowanie: Renderowanie to proces generowania obrazu trójwymiarowego na podstawie sceny, kamery i źródeł światła. W procesie renderowania obliczane są kolory i intensywności światła dla poszczególnych pikseli obrazu. Algorytmy renderowania wykorzystują informacje o geometrii, materiałach, teksturach oraz oświetleniu, aby generować realistyczne obrazy.

4. Rzutowanie: Rzutowanie jest procesem przekształcania obiektów trójwymiarowych na dwuwymiarowy obraz na płaszczyźnie ekranu. Rzutowanie perspektywiczne jest najczęściej stosowanym rodzajem rzutowania w grafice komputerowej, gdzie obiekty są rzutowane na płaszczyznę ekranu z uwzględnieniem perspektywy i oddalenia. Rzutowanie perspektywiczne odwzorowuje efekt głębi i perspektywy w renderowanych obrazach.

5. Przekształcenia przestrzenne: Przekształcenia przestrzenne obejmują operacje takie jak obroty, translacje i skalowanie obiektów w przestrzeni trójwymiarowej. Przekształcenia te pozwalają na manipulację obiektami, zmianę ich pozycji, orientacji i skali. W projekcie wykorzystuje się te przekształcenia do symulacji ruchu kamery oraz manipulacji sześcianami.

6. Obrót: Obrót jest przekształceniem przestrzennym polegającym na zmianie orientacji obiektu wokół określonej osi. Obrót może być wykonywany wokół osi x, y i/lub z. Macierze obracające są używane do wykonywania operacji obrotu na punktach w przestrzeni trójwymiarowej. Macierze te określają kąt i oś obrótu oraz są stosowane do transformacji punktów.

7. Translacja: Translacja jest przekształceniem przestrzennym polegającym na przesunięciu obiektu wzdłuż osi x, y i/lub z. Przesunięcie obiektu o wektor przesunięcia zmienia jego położenie w przestrzeni. Translacja jest

wykorzystywana do manipulacji obiektami w trójwymiarowej scenie, w tym do przemieszczania kamery wirtualnej.

8. Zoom: Zoom, zwany również skalowaniem, jest przekształceniem przestrzennym, które zmienia skalę obiektu. Może to oznaczać zarówno powiększanie, jak i pomniejszanie obiektu wzdłuż osi x, y i/lub z. Skalowanie jest wykorzystywane do zmiany perspektywy widoku oraz do dostosowywania rozmiarów obiektów w scenie.

9. Algorytm malarski (painter's algorithm): Algorytm malarski jest jednym z algorytmów używanych do eliminacji elementów zasłoniętych podczas renderowania sceny trójwymiarowej. Algorytm polega na sortowaniu obiektów na podstawie ich głębokości względem kamery. Następnie obiekty są renderowane w kolejności od najdalszego do najbliższego, a elementy zasłonięte przez inne obiekty są pomijane. Algorytm malarski jest stosowany w przypadkach, gdy nie ma potrzeby uwzględniania przezroczystości i przeplotów między obiektami.

Te podstawowe pojęcia teoretyczne są kluczowe dla projektu "Kamera wirtualna z eliminacją elementów zasłoniętych". Pozwalają na manipulację obiektami w przestrzeni trójwymiarowej, określanie ich położenia, orientacji i skali, a także na renderowanie sceny z uwzględnieniem eliminacji elementów zasłoniętych przy użyciu algorytmu malarskiego.

3 Budowa programu

Projekt składa się z dwóch plików geometry.py i main.py.

W projekcie plik geometry.py zawiera definicje klas Point i Cube, które reprezentują podstawowe elementy geometrii trójwymiarowej, tj. punkty i sześciany. Klasa Point zawiera metody do manipulacji punktami, takie jak transformacja, rzutowanie, obliczanie środka i odległości między punktami. Klasa Cube natomiast zawiera metody inicjalizujące sześcian oraz iterujące po jego punktach.

Plik main.py zawiera definicję klasy Scene, która reprezentuje scenę trójwymiarową. Klasa Scene posiada metody inicjalizujące scenę, wczytujące scenę zdefiniowaną przez użytkownika, przemieszczające kamery, skalujące obiekty, obracające kamery, obliczające odległości od kamery oraz obsługujące zdarzenia klawiatury.

Poprzez podział projektu na te dwa pliki, geometry.py i main.py, możemy logicznie oddzielić operacje związane z geometrią, transformacjami, zarządzaniem sceną i interakcjami użytkownika. Odpowiednie metody i klasy są zdefiniowane w celu umożliwienia manipulacji punktami, sześcianami, a także obsługi kamery wirtualnej i interakcji z użytkownikiem.

4 Funkcjonalność programu

Wszystkie funkcjonalności programu:

1. Przesuwanie kamery wirtualnej: Użytkownik może przesuwać kamerę wirtualną wzdłuż trzech osi (x, y, z) w przestrzeni trójwymiarowej.
2. Obracanie kamery wirtualnej: Użytkownik może obracać kamerę wirtualną wokół trzech osi (x, y, z) w przestrzeni trójwymiarowej.
3. Skalowanie obiektów: Użytkownik może zmieniać skalę obiektów w scenie, powiększając lub pomniejszając je.
4. Eliminacja elementów zasłoniętych: Program wykorzystuje algorytm malarski do eliminacji elementów zasłoniętych w scenie trójwymiarowej, co oznacza, że obiekty zasłonięte przez inne obiekty nie są renderowane.
5. Renderowanie sceny: Program renderuje scenę trójwymiarową na ekranie, wykorzystując przekształcenia geometryczne i perspektywiczne projekcje.
6. Interaktywna eksploracja sceny: Użytkownik może interaktywnie eksplorować scenę trójwymiarową, poruszając się wokół obiektów, obracając kamerę i skalując obiekty.
7. Obsługa zdarzeń: Program obsługuje zdarzenia z klawiatury, które umożliwiają użytkownikowi interakcję z aplikacją, np. przesuwanie kamery, obracanie, skalowanie itp.

Dzięki tym funkcjonalnościom użytkownik będzie mógł eksplorować trójwymiarową scenę, manipulować obiektami, poruszać się w przestrzeni i dostosowywać widok według własnych preferencji. Algorytm malarski pozwala na eliminację elementów zasłoniętych, co przyczynia się do bardziej realistycznego wyświetlania sceny

Sterowanie

Przesuwanie kamery wirtualnej:

- Przemieszczenie kamery w górę (wciśnięcie klawisza "w")
- Przemieszczenie kamery w dół (wciśnięcie klawisza "s")
- Przemieszczenie kamery w lewo (wciśnięcie klawisza "a")
- Przemieszczenie kamery w prawo (wciśnięcie klawisza "d")
- Przemieszczenie kamery do tyłu (wciśnięcie klawisza "q")
- Przemieszczenie kamery do przodu (wciśnięcie klawisza "e")

Obracanie kamery wirtualnej:

- Obrót kamery wzdłuż osi x przeciwnie do ruchu wskazówek zegara (wciśnięcie klawisza "8")

- Obrót kamery wzdłuż osi x zgodnie z ruchem wskazówek zegara (wciśnięcie klawisza "2")
- Obrót kamery wzdłuż osi y zgodnie z ruchem wskazówek zegara (wciśnięcie klawisza "4")
- Obrót kamery wzdłuż osi y przeciwnie do ruchu wskazówek zegara (wciśnięcie klawisza "6")
- Obrót kamery wzdłuż osi z przeciwnie do ruchu wskazówek zegara (wciśnięcie klawisza "7")
- Obrót kamery wzdłuż osi z zgodnie z ruchem wskazówek zegara (wciśnięcie klawisza "9")

Zoom (zmiana skali obiektów):

- Powiększenie obiektów (wciśnięcie klawisza "+")
- Pomniejszenie obiektów (wciśnięcie klawisza "-")

Przy wciśnięciu klawisza "Enter" scena wraca do pozycji początkowej.

Kiedy użytkownik wciśnie odpowiedni klawisz, zostanie wywołana odpowiednia metoda sceny (move, turn, zoom) w zależności od klawisza naciśniętego przez użytkownika. Na przykład, wciśnięcie klawisza "w" spowoduje wywołanie metody move sceny, przekazując jako argumenty "y" (oś) i 1 (wartość przesunięcia w górę).

Dzięki tym funkcjonalnościom użytkownik będzie mógł kontrolować ruch, obrót i skalę kamery oraz interaktywnie eksplorować scenę trójwymiarową.

5 Podsumowanie

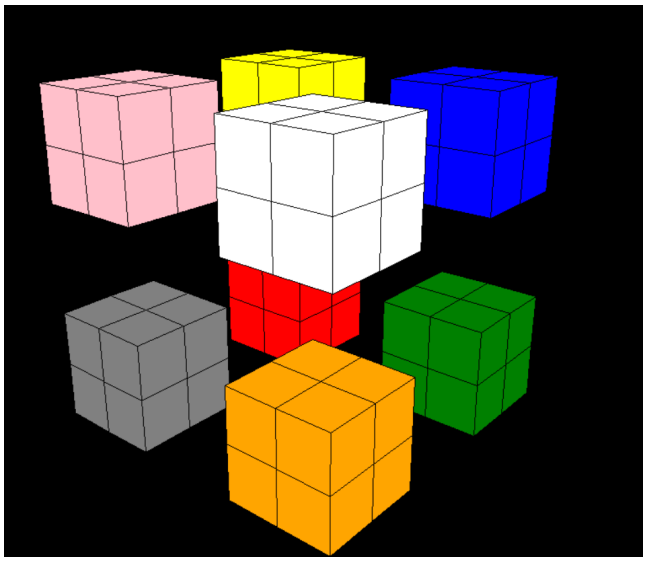
Projekt "Kamera wirtualna z eliminacją elementów zasłoniętych" skoncentrował się na implementacji interaktywnej aplikacji, która umożliwia eksplorację trójwymiarowej sceny przy użyciu kamery wirtualnej. Jednym z kluczowych elementów projektu było zastosowanie algorytmu malarskiego, który umożliwił eliminację elementów zasłoniętych, co przyczyniło się do lepszego renderowania sceny i bardziej realistycznych wizualizacji.

5.1 Screeny z eksploracji

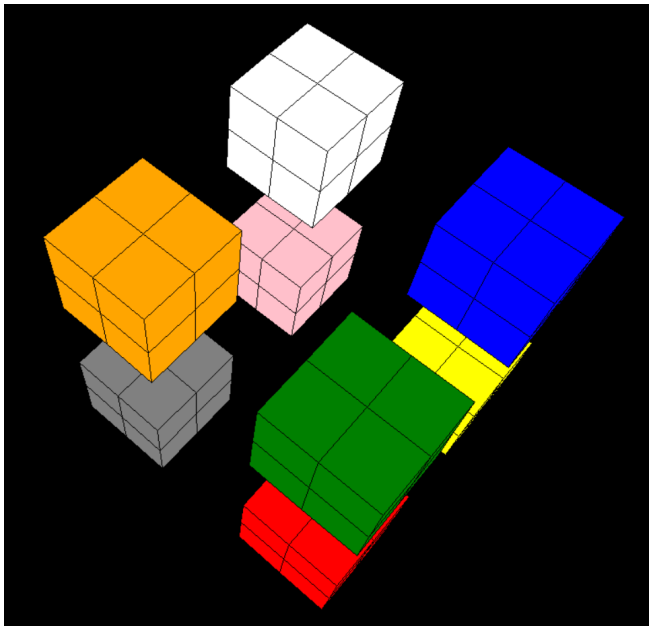
Oto kilka przykładowych zdjęć zrobionych podczas używania projektu:

5.2 Wnioski

Realizacja projektu "Kamera wirtualna z eliminacją elementów zasłoniętych" przyniosła ważne wnioski dotyczące renderowania scen trójwymiarowych.

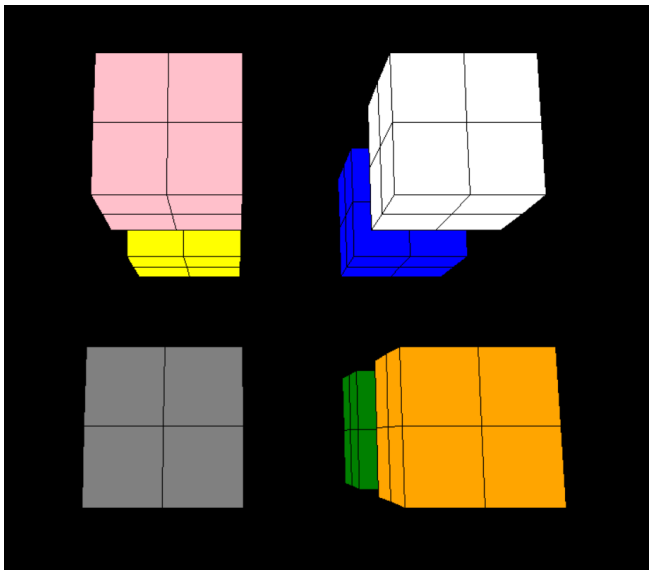


Rysunek 2: Przykład 1

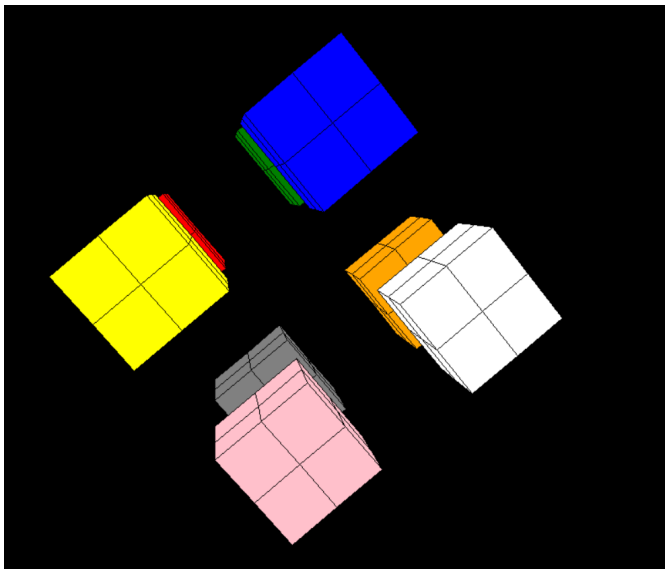


Rysunek 3: Przykład 2

Algorytm malarski okazał się niezwykle przydatny w procesie eliminacji elementów zasłoniętych, co miało kluczowe znaczenie dla uzyskania bardziej reali-



Rysunek 4: Przykład 3



Rysunek 5: Przykład 4

stycznych obrazów. Dzięki temu użytkownik mógł swobodnie eksplorować scenę, unikając nieprawidłowych efektów zasłaniania obiektów.

Eliminacja elementów zasłoniętych jest kluczowym aspektem w grafice trójwymiarowej, ponieważ pozwala na uzyskanie dokładniejszych i bardziej precy-

zyjnych obrazów. Projekt umożliwił praktyczne zastosowanie wiedzy z zakresu przekształceń geometrycznych, perspektywicznego rzutowania oraz manipulacji macierzami transformacji.

Wnioski z tego projektu mogą być wykorzystane jako podstawa do dalszego rozwijania aplikacji grafiki trójwymiarowej oraz implementacji bardziej zaawansowanych technik eliminacji elementów zasłoniętych, takich jak algorytmy z sortowaniem malarskim, bufor Z czy cieniowanie. Rozbudowa programu o takie techniki umożliwiłaby jeszcze bardziej realistyczne renderowanie scen trójwymiarowych.

Projekt "Kamera wirtualna z eliminacją elementów zasłoniętych" dostarczył nie tylko wiedzy i umiejętności w dziedzinie grafiki trójwymiarowej, ale także satysfakcji z tworzenia aplikacji, która potrafi wyświetlić scenę z usunięciem elementów zasłoniętych, co wpływa na lepszą jakość renderowania i bardziej realistyczny efekt wizualny.

Bibliografia

[1] `https://www.gabrielgambetta.com/computer-graphics-from-scratch/`

[2] `http://cgpp.net/`

[3] `https://realpython.com/`

[4] `https://www.pygame.org/docs/`

[5] `https://www.scratchapixel.com/`

[6] `https://realpython.com/tutorials/graphics/`

[7] `https://pl.wikipedia.org/wiki/Algorytm_malarski`