



Przedmiot Grafika komputerowa - Projekt 2
Eliminacja elementów zasłoniętych

Wykonanie:

1. Ivan Prapakets

Data: 21.04.2020-29.04.2020

Warszawa

Spis treści

1	Cel projektu	2
2	Realizacja zadania	2
2.1	Szczegóły	2
3	Rysowanie prostopadłościanów	3
3.1	Budowa	3
3.2	Wierzchołki	3
3.3	Renderowanie 3d do 2d	3
3.4	Krawędzie	4
3.5	Podstawa	4
4	Sortowanie ścian	4
5	Pierwotny stan programu	5
6	Prezentacja działania	5
7	Wnioski	8

1 Cel projektu

Celem projektu jest rozszerzyć swoje umiejętności oraz dowiedzieć się więcej o grafice komputerowej i eliminacji elementów zasłoniętych. Jednym słowem poczuć ją z punktu własnego doświadczenia.

2 Realizacja zadania

Zadanie projektu jest napisać program, który będzie umożliwiał zobaczenia eliminacji elementów zasłoniętych. Dla realizacji tego zadania użyłem już zrobioną Wirtualną kamerę. Przerobiłem projekt poprzedni (czyli Wirtualna kamera) na projekt 2 (eliminacja elementów zasłoniętych). Projekt został zrealizowany za pomocą Python3.

2.1 Szczegóły

1. Poruszania po całej przestrzeni i zobaczenie elementów zasłoniętych.
2. Jako obraz sceny, który będzie prezentował możliwości będą cztery prostopadłościany
3. Program ma dwa tryby:
 - z eliminacją elementów zasłoniętych
 - bez eliminacji elementów zasłoniętych

Do realizacji oraz pokazania eliminacji elementów zasłoniętych użyłem algorytm malarski innym słowem **algorytm sortowania ścian**. Posortowanie elementarnych fragmentów prostopadłościanu zależnie od odległości od obserwatora: od najdalszego do najbliższego. Zastosowałem porównanie położenia środków ciężkości wielokątów – bliżej obserwatora znajduje się ten, którego środek ciężkości znajduje się bliżej. Dokładny opis poniżej.

3 Rysowanie prostopadłościanów

Dla poprawnego działania i pokazania eliminacji elementów zasłoniętych, na początku trzeba zastanowić się co będziemy zasłaniać. Najfajniejszym przykładem dla zilustrowania działania eliminacji elementów zasłoniętych jest zwykłe prostopadłościany, bo widać wszystkie ściany. Dla jeszcze lepszego ilustrowania i pokazania poprawnego działania są 4 prostopadłościany.

3.1 Budowa

Budowa prostopadłościana:

- wierzchołki
- ściana
- krawędzi
- podstawa

3.2 Wierzchołki

Zacząłem od rysowania wierzchołków:

$(-1, -1, -1), (1, -1, -1), (1, 1, -1), (-1, 1, -1), (-1, -1, 1), (1, -1, 1), (1, 1, 1), (-1, 1, 1)$

Robimy to wokół nas, więc przesuniemy do przodu.

3.3 Renderowanie 3d do 2d

Można zauważyć, że na ekranie nie ma osi **Z**, więc użyjemy tylko **X** i oś **Y**, ale w zależności od osi **Z** punktu (czyli głębokości), będziemy kierować w kierunku odległości środka.

3.4 Krawędzie

Następnie jest rysowanie krawędzi:

$(0,1), (0,3), (0,4)$

$(2,1), (2,3), (2,6)$

$(5,1), (5,4), (5,6)$

$(7,3), (7,4), (7,6)$

Kiedy mamy krawędzi oraz wierzchołki musimy uzyskać lokalizację 2D z nich obu umieścimy je na liście.

3.5 Podstawa

Rysowanie podstaw:

$(0,1,2,3), (4,5,6,7)$

$(0,1,5,4), (2,3,7,6)$

$(1,5,6,2), (0,3,7,4)$

Dla poprawnego działania algorytmu musiałem podzielić podstawy na mniejsze części w pętli for, żeby znaleźć centrum:

$(0,1), (1,2), (2,3), (3,0)$

4 Sortowanie ścian

Skoro już mamy punkty środkowe prostopadłościanów. Możemy posortować w następujący sposób:

- Sortowanie odległości od osłony widoku (czyli środkowych punktów)

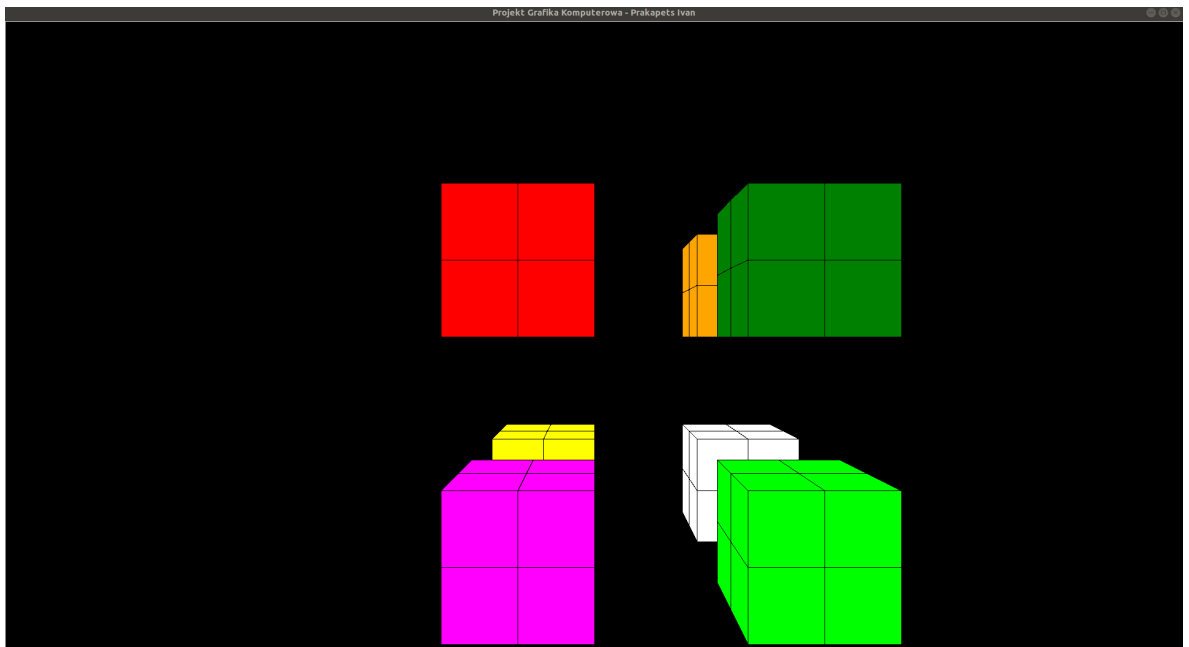
Po sortowaniu są następujący przykładowy wynik scena:

$[(642.0, 362.0), (517.0, 362.0), (517.0, 237.0), (642.0, 237.0)]$

Z powodu dużych podziałów oraz wyników dojść trudno do pokazania punktów środkowych w sprawozdaniu, więc można to zobaczyć na działaniu(screenach).

5 Pierwotny stan programu

Poniżej jest pierwotny stan programu(po uruchomieniu).

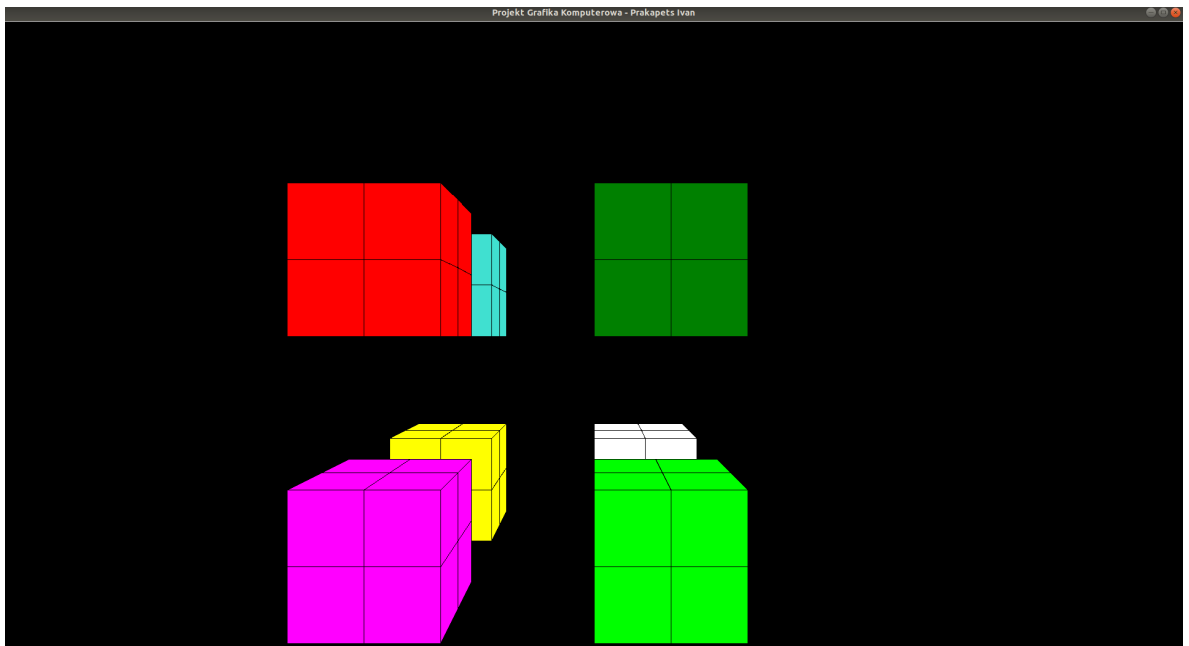


Rysunek 1: pierwotny stan

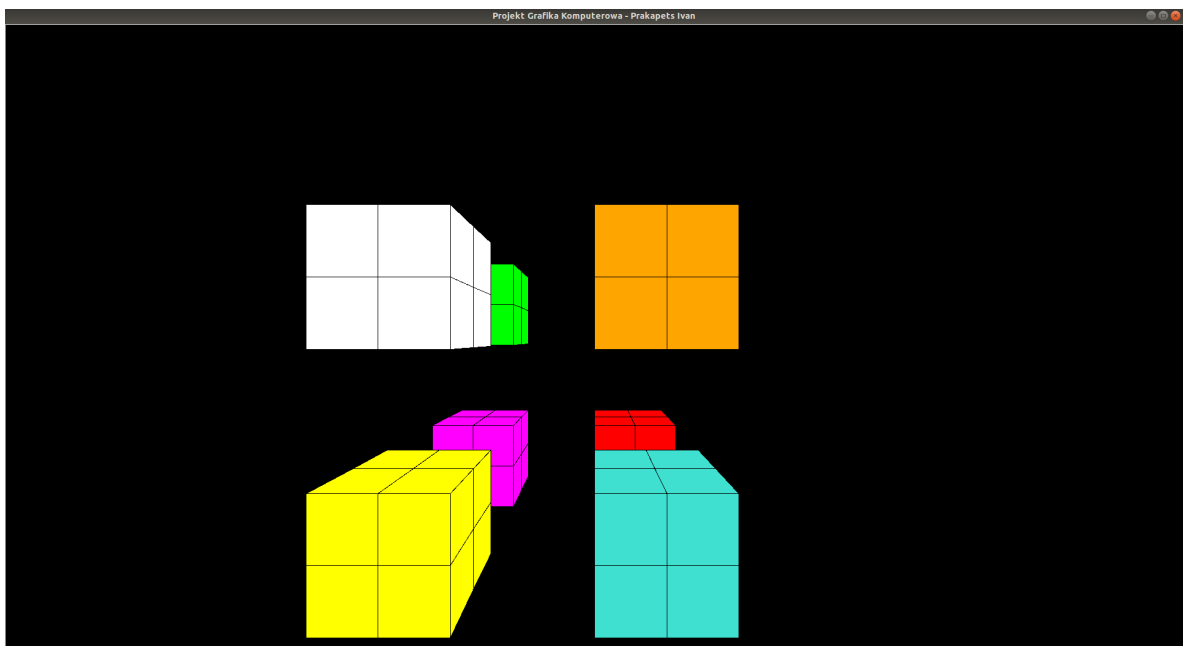
6 Prezentacja działania

Poniżej jest widoczna eliminacja elementów zasłoniętych:

Na rysunku 2 można zobaczyć jak zielony w całości zasłania pomarańczowy prostopodłościan. Pozostałe tylko części, jak w rzeczywistości.

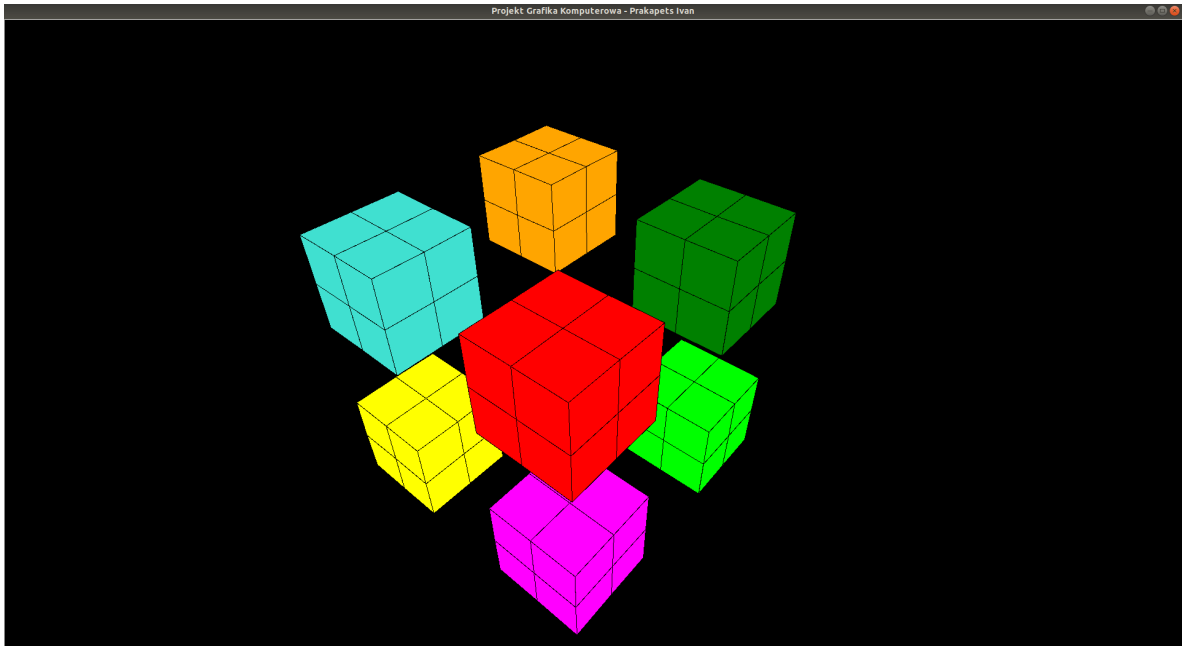


Rysunek 2: eliminacja elementów zasłoniętych

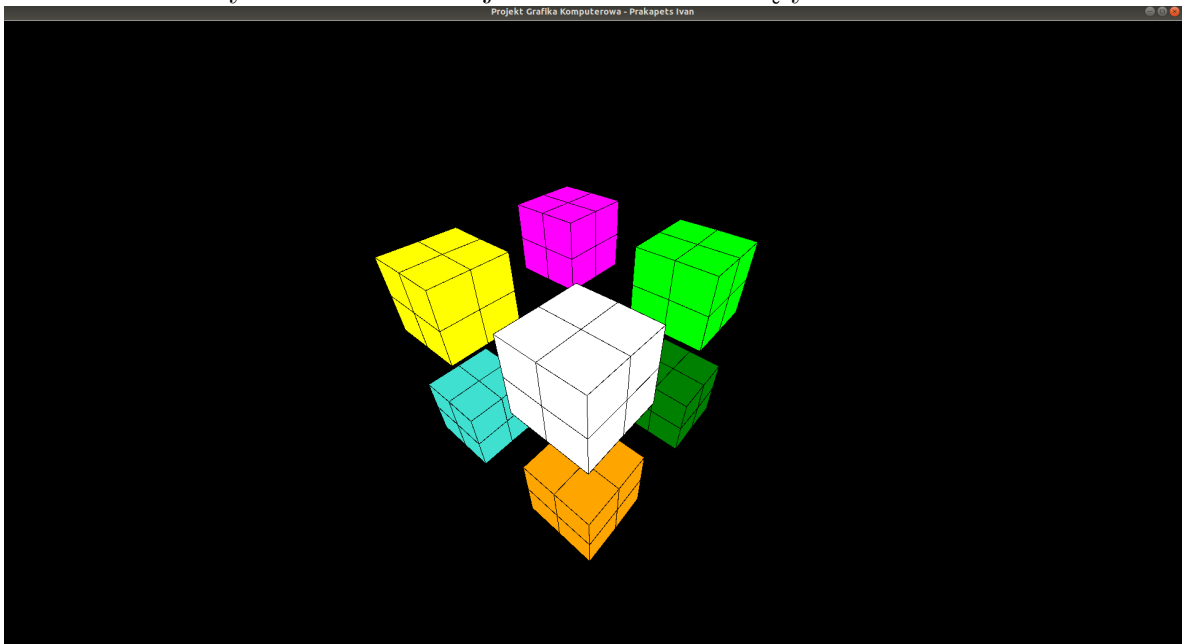


Rysunek 3: eliminacja elementów zasłoniętych

Na rysunku 3 można zaobserwować efekt odwrotny.



Rysunek 4: eliminacja elementów zasłoniętych



Rysunek 5: eliminacja elementów zasłoniętych

Powyżej zobaczmy jak czerwony zasłania biały oraz na odwrót.

7 Wnioski

Wszystkie wymagania zostały zrealizowane, program działa. Zostały pokazane elementy zasłonięte jak w rzeczywistości. Widać wszystkie pozycje prostopadłościanów. Jestem zadowolony swojej pracą i mam nadzieję, że wszystkie wymagania zostały zrealizowane poprawnie i tak jak powinni być.