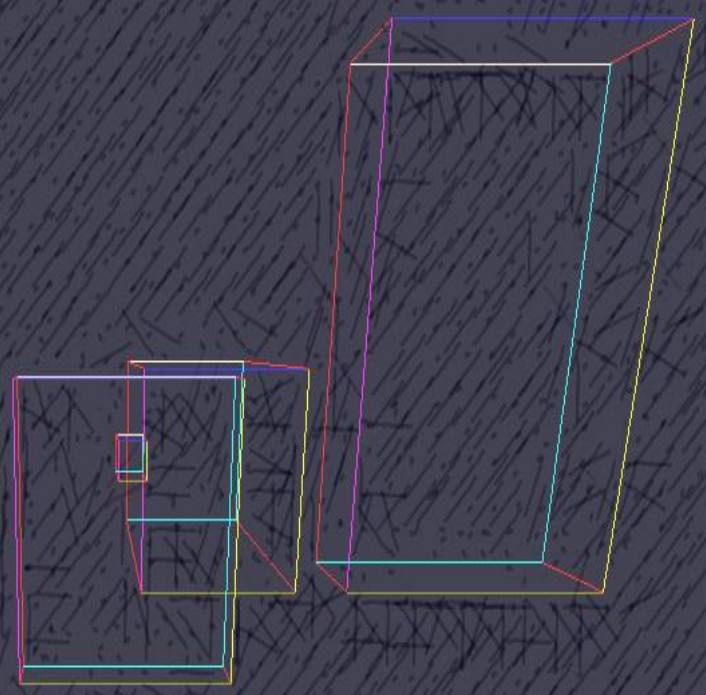
Projekt 1
KAMERA



17 KWIETNIA 2020

Jan Łukomski

Przedmiot: Grafika Komputerowa

Cel programu

Program naśladuje poruszanie się kamery. Umożliwia wyświetlanie brył na płaszczyźnie pod odpowidnimi kątami.

Sposób realizacji

Ogólny pomysł

Wyświetlenie bryły w odpowidnim miejscu w odpowiedni sposób sprowadziłem do problemu odwzorowania punktu w ogólnym układzie współrzędnym na układzie kamery. Każda bryła potrafi zwrócić listę swoich krawędzi, gdzie krawędź to para punktów (3d). Dla każdej krawędzi program znajduje miejsce obydwu punktów w układzie kamery, następnie rysuje linię z jednego punktu do drugiego.

Stosuję lewoskrętny układ współrzędnych

Sprowadzenie punktu do układu kamery

Kamera ma swój punkt położenia (Position), ma swój naturalnego położenia, w które jest skierowana, zanim zacznie być obracana. Naturalne położenie ustaliłem dla wygody na wektor [0,0,1].

Dodatkowo przechoduje wektor kątów (StandardAttitude) o które jest obrócona od pierwotnego położenia. Na starcie jest on wyzerowany. To ten wektor będę modyfikował w celu obsługi obrotów kamery.

Sprowadzeni punktu do układu kamery polega na przesunięciu środka płaszczyzny do początku układu współrzędnych oraz wykonanie obrotów przeciwnych do tych zapisanych w wektorze StandardAttitude.

Rzutowanie punktu na płaszczyznę

Kamera ma ustaloną odległość od ekranu (którą mogę modyfikwać dając efekt zoom). Płaszczyzna zawsze jest prostopadła do wektora nachylenia kamery.

Dzięki umieszczeniu płaszczyzny po po przekształceniach na osi OX, wyznaczenie rzutu punktu na płaszczynę sprowadza się do wyznaczenia wektora CA pomiedzy kamerą oraz punktem po drugiej stronie płaszczyzny, oraz z proporcji przeskalowania tego wektora.

Uznaję za niewidoczny wszystkie punkty znajdujące się przed płaszczyzną. Jeżeli co najmniej jeden punkt z krędzi znajduje się przed płaszczyzną, wtedy nie wyświetlam tej płaszczyzny.

Wyświetlanie płaszczyzny na ekranie

Dzięki ułożeniu płaszczyzny na osi OX, w celu wyświetlania płaszczyzny na ekranie, wyświetlam współrzędne X oraz Y, ponieważ Z jest wyzerowana.

Obsługa programu

Konfiguracja brył

Pozycja oraz wymiary brył wczytywane są z pliku blocs.txt znajdującego się w katalogu razem z innnymi plikami źródłowymi.

Każda linia reprezentuje jeden blok. Blok jest reprezentowany przez 6 liczb całkowitych oddzielonych spacjami oznaczających odpowiednio pozycję bloku (X, Y, Z) oraz wymiary bryły (X, Y, Z).

Sterowanie kamerą

W	***Obrót góra***
S	***Obrót dół***
A	***Obrót lewo***
D	***Obrót prawo***
E	***Obrót wokół osi Z zgodnie z ruchem zegara***
Q	***Obrót wokół osi Z przeciwnie z ruchem zegara***
I	***Przesunięcie do przodu***
K	***Przesunięcie do tyłu***
J	***Przesunięcie w lewo***
L	***Przesunięcie w prawo***
O	***Przesunięcie do góry***
U	***Przesunięcie w dół***
P	***Powiększenie***
;	***Pomniejszenie***

Środowisko

Napisałem program w C# korzystając z biblioteki graficznej OpenTK. Wszystkie kształty rysowałem poleceniem rysującym linię od punktu do punktu na ekranie. Program dedykowany jest dla systemu Windows 10.

Kod programu

Program znajduje się w publicznym repozytorium na githubie: https://github.com/lukomski/GKCamera

Podsumowanie

Na każdym prawie etapie konieczne było stosowanie przekształceń wokół osi. W tym celu oczywiście stosowałem macierze przekształceń. Do funkcji dokonujących

przekształeceń pisałem testy jednostkowe, ponieważ poziom abastrakcji był na tyle zawiły, że bez podziału na drobniejsze problemu i osobne ich przetestowanie, ugrzązłbym w drobnych błędach.

Kilka zrzutów z ekranu