

Politechnika Warszawska  
Wydział Elektryczny

SPRAWOZDANIE  
Z PROJEKTU WIRTUALNA KAMERA  
„SPOLENS” - ROZSZERZENIE

*Autor:*  
DANIEL SPORYSZ

17.04.2020

# Spis treści

<b>1</b>	<b>Opis projektu</b>	<b>2</b>
<b>2</b>	<b>Wymagania techniczne</b>	<b>2</b>
<b>3</b>	<b>Funkcjonalność programu</b>	<b>2</b>
3.1	Interfejs graficzny . . . . .	2
3.2	Translacja . . . . .	3
3.3	Rotacja . . . . .	4
3.4	Zoom . . . . .	5
3.5	Płaszczyzna ścinająca (Clipping Plane) . . . . .	6
3.6	Rysowanie płaszczyzn . . . . .	7
3.7	Płaszczyzny i clipping . . . . .	8
<b>4</b>	<b>Struktura pliku konfiguracyjnego</b>	<b>9</b>
4.1	Przykładowy plik konfiguracyjny . . . . .	9
4.2	Linie zaczynające się od P . . . . .	9
4.3	Linie zaczynające się od C . . . . .	9
4.4	Linie zaczynające się od O . . . . .	10
4.5	Wnioski i uwagi . . . . .	11

# 1 Opis projektu

„Spolens” to program do generowania widoku na trójwymiarową przestrzeń, widzianego z wirtualnej kamery, której ruchami i parametrami sterować można za pomocą klawiatury. Konfiguracja położenia i koloru elementów w przestrzeni jest wczytywana z pliku przy starcie programu.

Rozszerzenie wprowadza rysowanie figur z wypełnieniem, co uatrakcyjni widok z wirtualnej kamery, ale wprowadza nowy problem wzajemnego zakrywania się płaszczyzn, co oznacza, że ważny jest dobór kolejności rysowania.

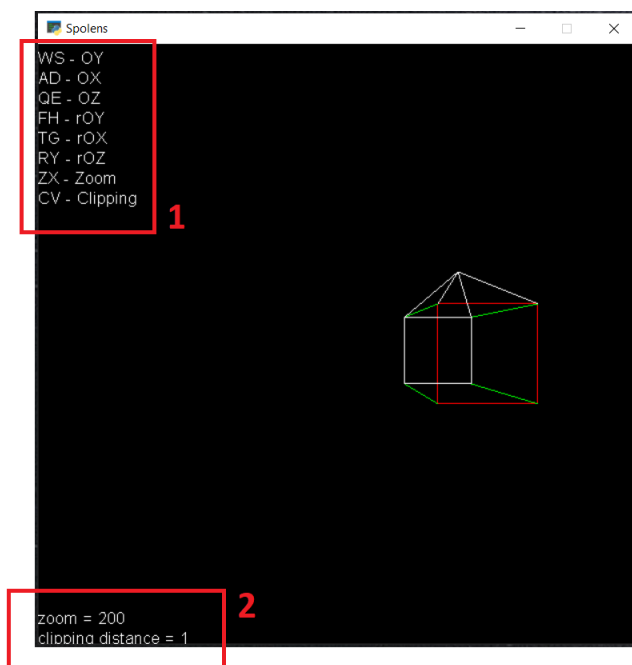
## 2 Wymagania techniczne

Do uruchomienia należy zainstalować Python 3.8, a następnie zainstalować moduł Pyglet do Python.

## 3 Funkcjonalność programu

### 3.1 Interfejs graficzny

Po wywołaniu programu, tworzone jest nowe okno w którym wyświetlany jest widok z wirtualnej kamery, klawisze sterujące oraz wartości parametrów.



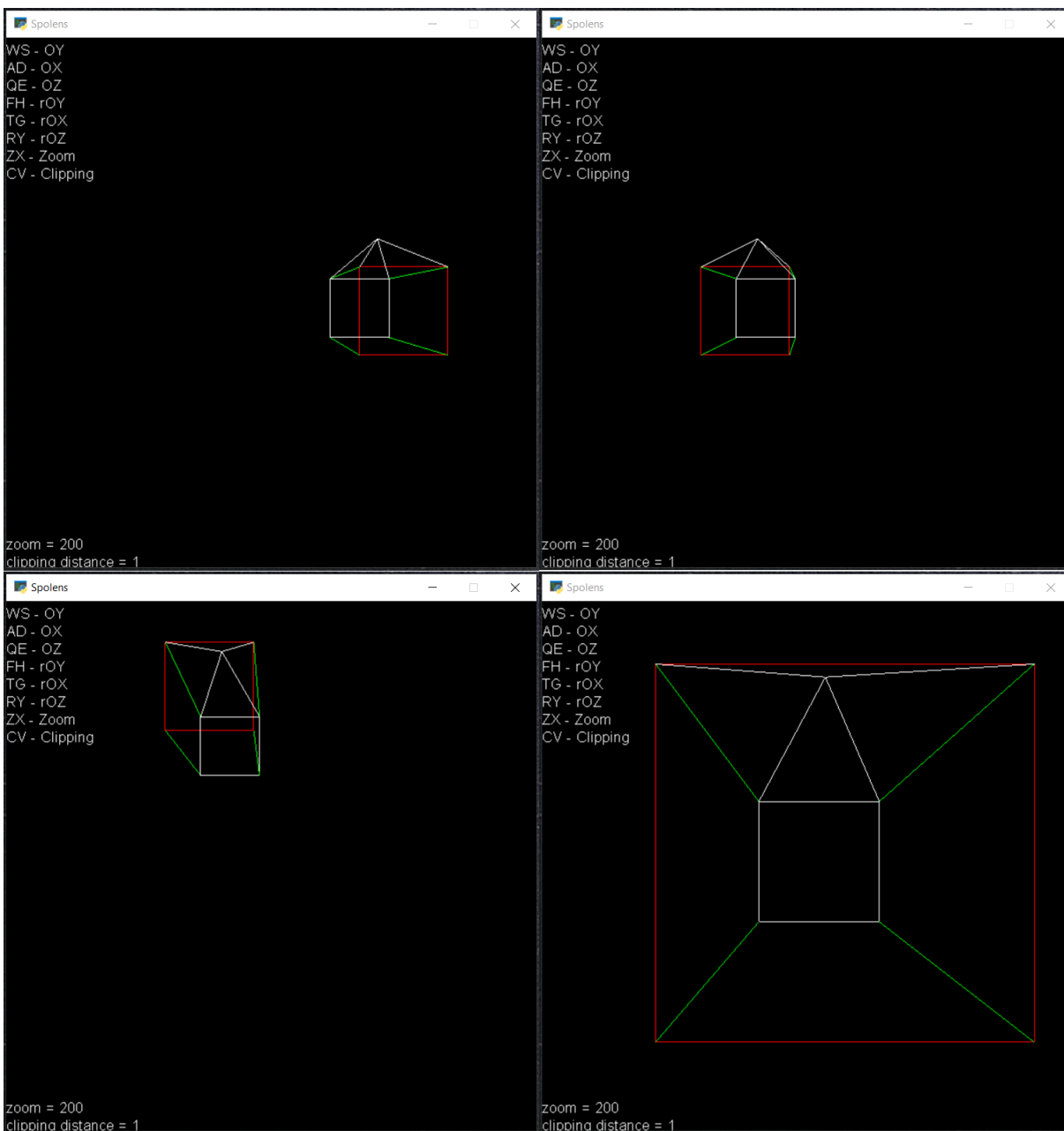
W zaznaczeniu @1 wypisana jest lista klawiszy sterujących wraz z nazwą akcji, która jest z nimi związana.

Zaznaczenie @2 wskazuje na aktualne wartości parametrów.

## 3.2 Translacja

Przesunięciem kontrolujemy za pomocą klawiszy:

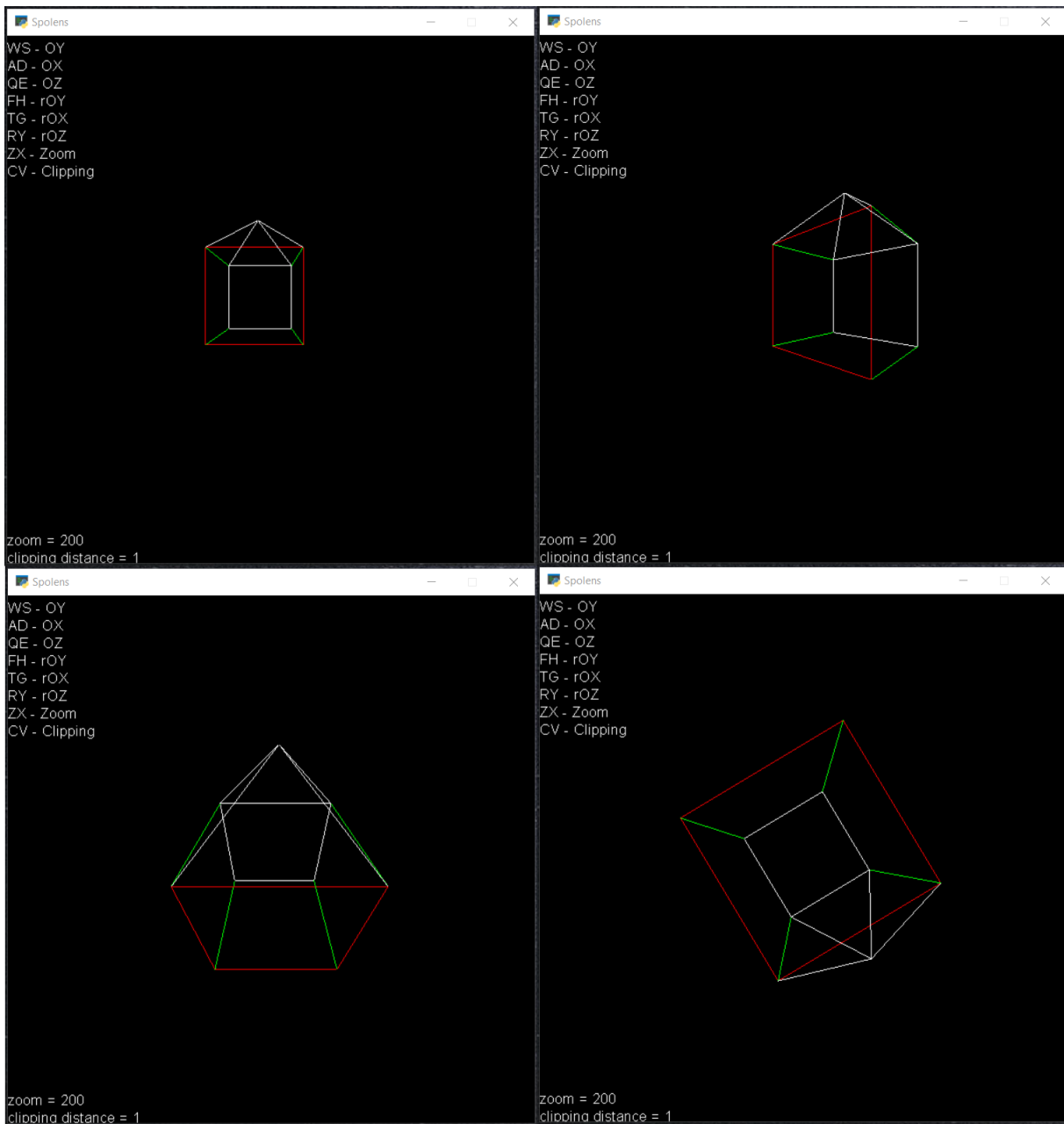
- A/D w osi OY,
- W/S w osi OX,
- Q/E w osi OZ.



### 3.3 Rotacja

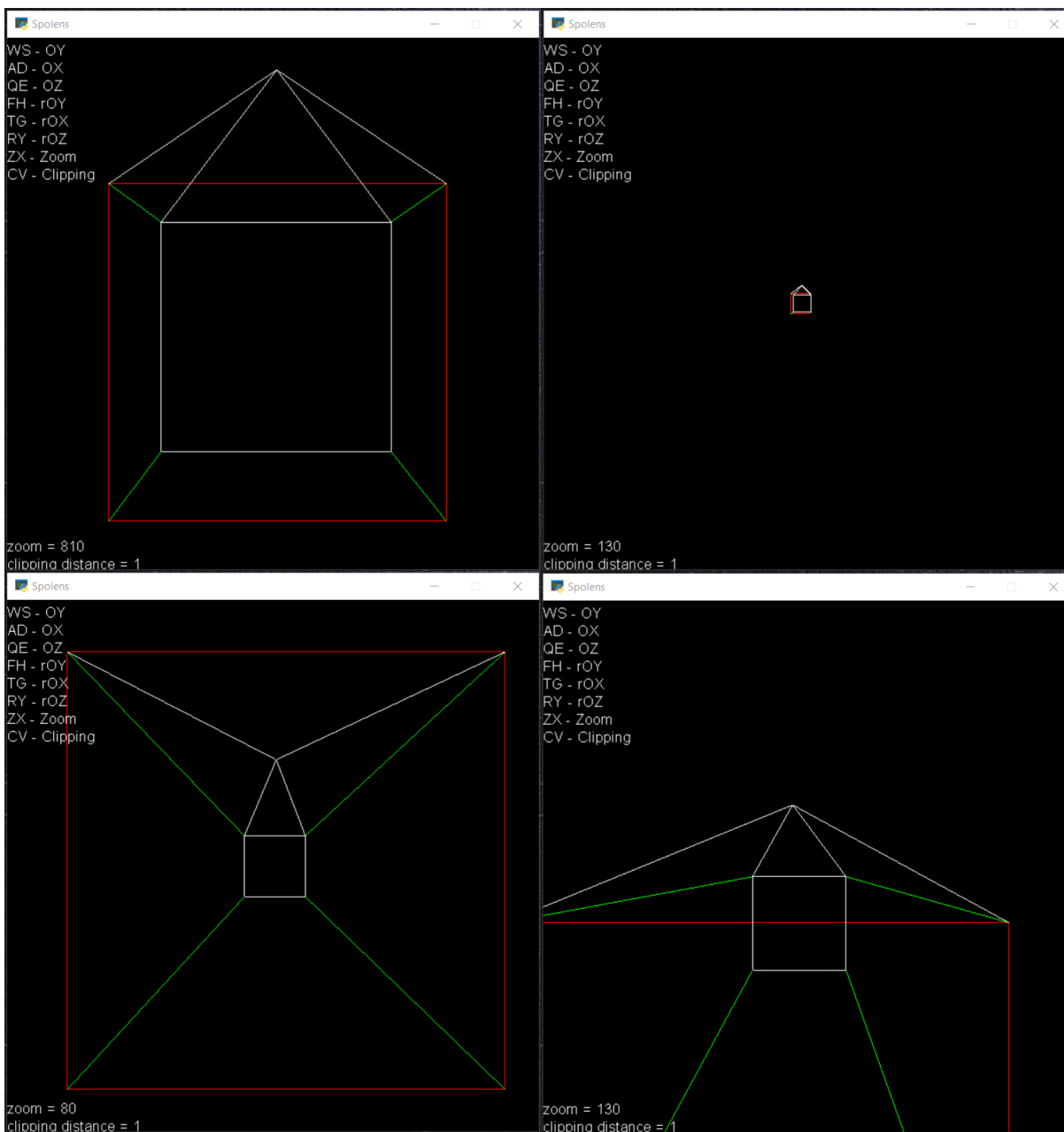
Rotację kontrolujemy za pomocą klawiszy:

- F/H w osi OY,
- T/G w osi OX,
- R/Y w osi OZ.



### 3.4 Zoom

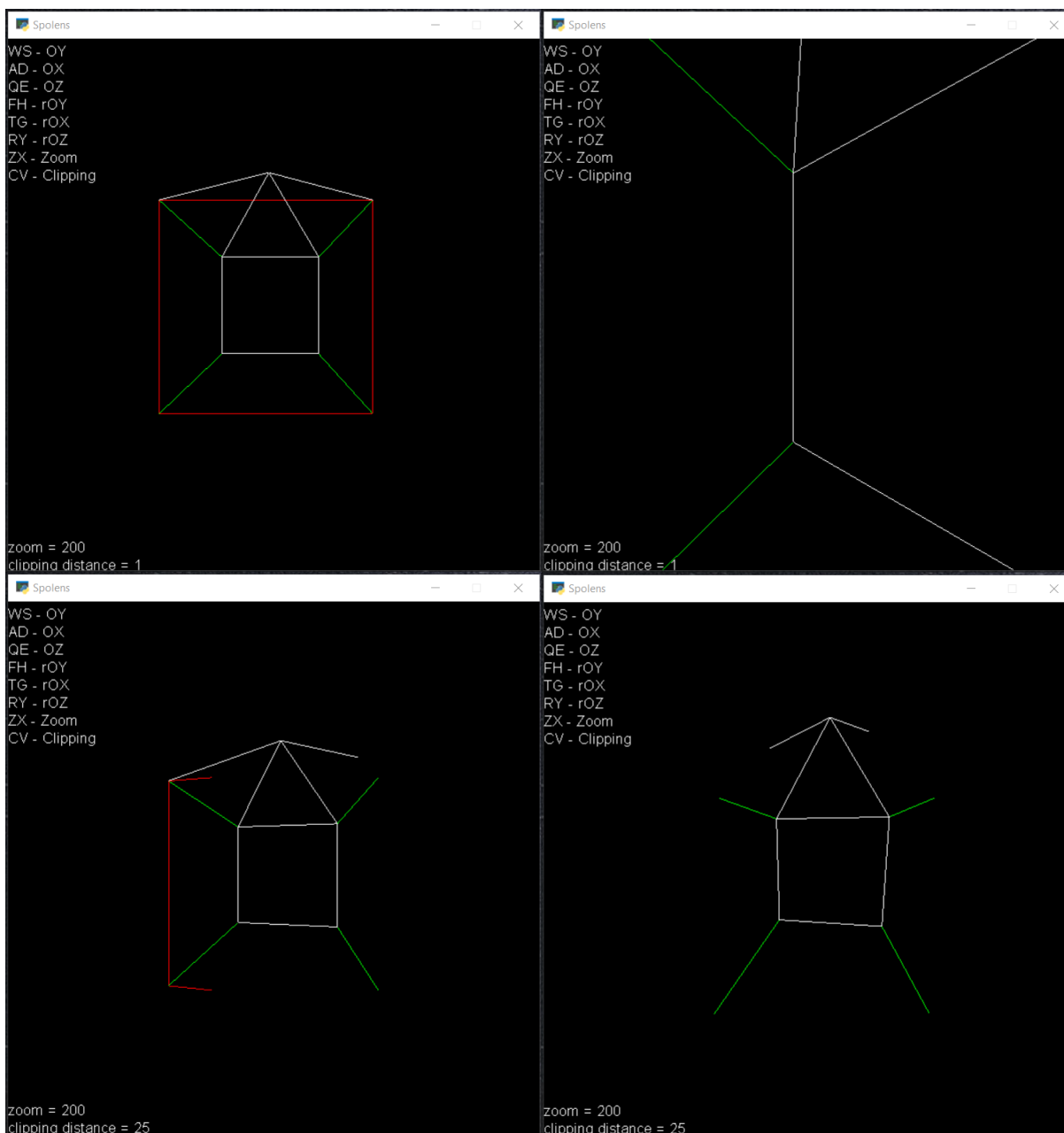
Przybliżeniem kontrolujemy za pomocą klawiszy Z/X. Dodatkowo informacja o wartości tego parametru jest wyświetlana w lewnym dolnym rogu. Jest to operacja różna od przesunięcia w osi OZ. Widać to po rozciągnięciu sześciangu.



### 3.5 Płaszczyzna ścinająca (Clipping Plane)

Linie, które są za płaszczyzną ścinającą (kamerą), są ignorowane przy rysowaniu. Zaś linie które przechodzą przez tę płaszczyznę, są przycinane do punktu przecięcia się płaszczyzny z linią. Zapewnia to poprawne wyświetlanie obrazu w każdej pozycji kamery.

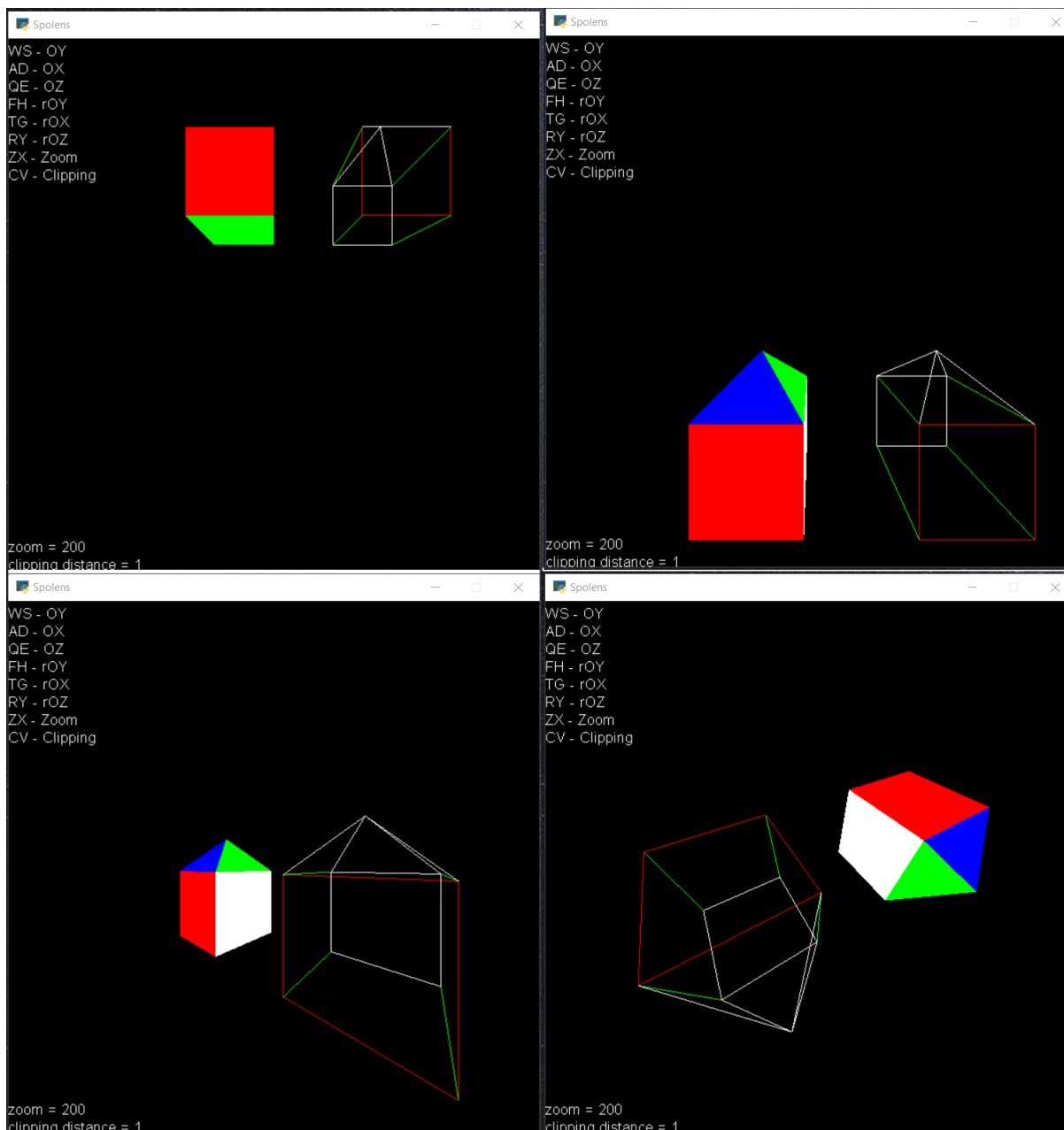
Płaszczyznę ścinającą (clipping plane) sterujemy klawiszami C/V, a wartość parametru jest wyświetlana w lewym dolnym rogu.



### 3.6 Rysowanie płaszczyzn

Rozszerzeniem programu jest rysowanie płaszczyzn. Funkcja ta nie wyklucza pozostałych, takich jak rysowanie linii, translacja, rotacja czy przybliżenie. Program ma elastyczną strukturę kodu, dlatego łatwo się go rozbudowuje bez duplikacji kodu.

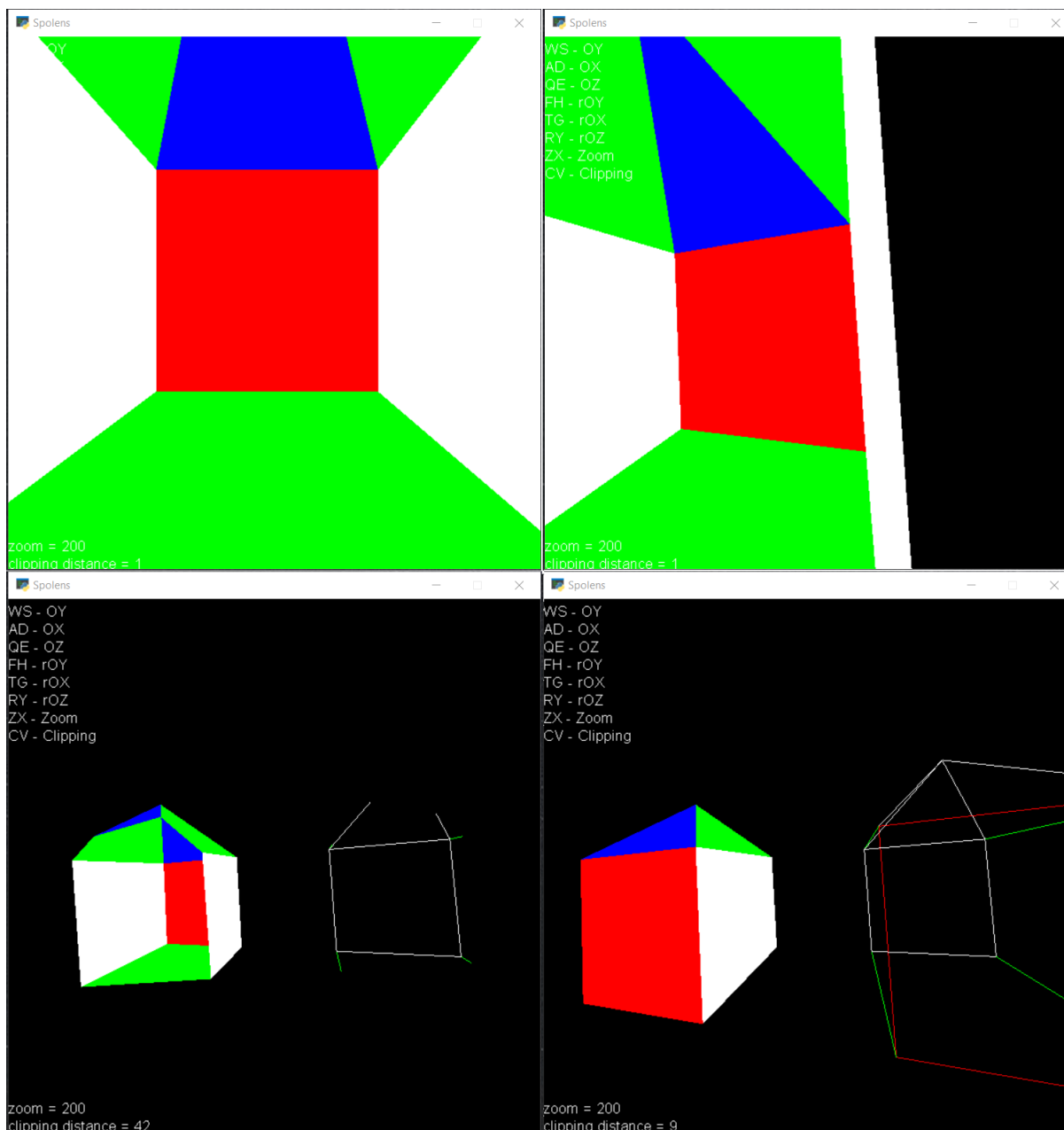
Do rysowania płaszczyzn wykorzystano algorytm malarski. Płaszczyzny są sortowane po skrajnym punkcie (tj. punkcie o największej wartości parametru Z), a jeśli ten warunek nie daje jednoznacznej odpowiedzi, uwzględniana jest odległość środka płaszczyzny do środka ekranu.





### 3.7 Płaszczyzny i clipping

Analogicznie do sytuacji z rysowaniem linii, dla zapewnienia poprawności rysowania płaszczyzn, konieczne jest przycięcie płaszczyzn jeśli przechodzą przez ekran kamery.



## 4 Struktura pliku konfiguracyjnego

Plik w formacie .TXT składa się z linii w których ważny jest pierwszy znak, który określa typ danych jakie zawiera linia.

Tymi znakami są P i C, a linie które zaczynają się z innym znakiem są ignorowane.

### 4.1 Przykładowy plik konfiguracyjny

```
jakis komentarz

P frontDownLeft 30 30 60
P frontDownRight 60 30 60

C frontDownLeft frontDownRight 1 0 0 1

P top 45 75 75

C frontDownRight top 1 1 1 1

O frontDownLeft frontDownRight top 1 1 1 1
```

### 4.2 Linie zaczynające się od P

Linie zaczynające się od P (od Point), definiują punkty i zawierają dane w formacie:

P [nazwa punktu] [x] [y] [z]

Gdzie:

- [nazwa punktu] to unikalny identyfikator punktu, którego należy używać przy definiowaniu połączeń (lini),
- [x] [y] [z] to parametry położenia punktu w przestrzeni trójwymiarowej.

### 4.3 Linie zaczynające się od C

Linie zaczynające się od C (od Connection), definiują połączenia pomiędzy punktami i zawierają dane w formacie:

C [nazwa punktu] [nazwa punktu2] [R] [G] [B] [A]

Gdzie:

- [nazwa punktu] i [nazwa punktu2] to ID punktów między którymi biegnie linia,
- [R] [G] [B] [A] to parametry koloru w formacie RGBA w przedziale wartości od 0 do 1.

## 4.4 Linie zaczynające się od O

Linie zaczynające się od O, definiują płaszczyznę w formacie:

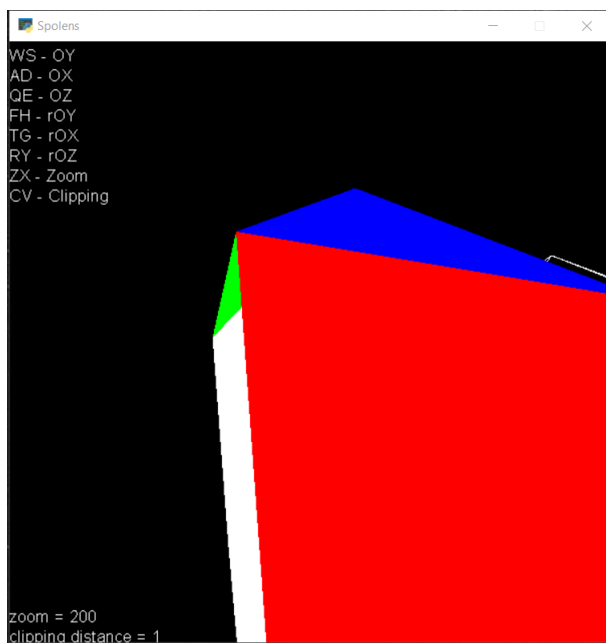
O [nazwy punktów] [R] [G] [B] [A]

Gdzie:

- [nazwy punktów] to ID punktów, które tworzą płaszczyznę, płaszczyzna minimalnie ma 3 punkty, ważne jest, aby podawać punkty w kolejności ich łączenia,
- [R] [G] [B] [A] to parametry koloru w formacie RGBA w przedziale wartości od 0 do 1.

## 4.5 Wnioski i uwagi

Rysowanie płaszczyzn za pomocą algorytmu malarskiego, nie przynosi dobrych efektów dla dowolnych ustawień. Występują błędy takie jak:



Gdzie zielony obszar nie powinien być widoczny.

Podczas pracy przy projekcie, zrozumiałem jak zaawansowane technicznie są silniki graficzne oraz jaki ogrom obliczeń konieczny jest do wygenerowania obrazów, takie jakie mamy w dzisiejszych grach wideo, czyli takie na których są obiekty składające się z tysięcy elementów z uzględnieniem światła, własności materiałów i efektów dodatkowych - a to wszystko generowane w czasie rzeczywistym.