

СОФИЙСКИ УНИВЕРСИТЕТ  
„СВ. КЛИМЕНТ ОХРИДСКИ“



# Изрязване

ТЕМА №17

# Съдържание

## Тема 17: Изрязване

- Полупространства
- Задни повърхнини
- Скрити линии
- Избор на дати за изпит

# Полупространства

# Изрязване

## Изрязване

- Процес на премахване на примитиви или части от примитиви от видимата част на сцената
- Различно от отсичането, което е премахване на нещата извън видимата част от сцената

# Методи за методи

- Изрязване чрез полупространства с равнини  
(на англ. *clipping planes*)
- Изрязване на стени  
(на англ. *face culling*)
- Изрязване на скрити линии  
(на англ. *hidden line removing*)

# Полупространства

## Изрязване с полупространства

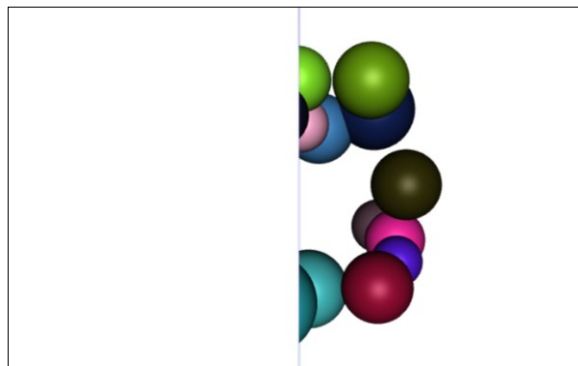
- Равнина с уравнение  $ax + by + cz + d = 0$
- Разделя пространството на две полупространства
- Рисува се само тази част от обекта, която се намира в положителното полупространство

## Кое е то?

- За точка  $(P_x, P_y, P_z)$  имаме  $aP_x + bP_y + cP_z + d > 0$

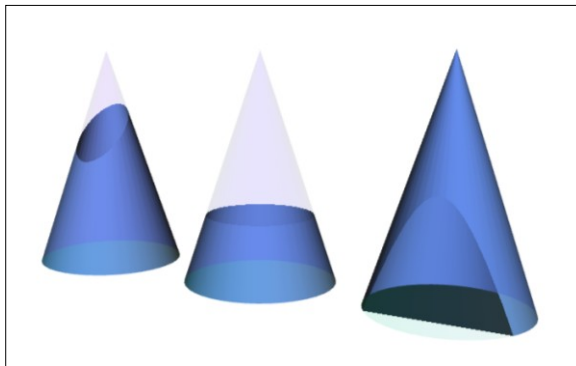
## Пример с изрязване

- Изрязване по равнината  $YZ$ , т.е.  $x = 0$
- Едната половина  $+1x + 0y + 0z + 0$
- Другата половина  $-1x + 0y + 0z + 0$



## Отместена равнина т.е. $x = d$

- Опитайте се да разберете кои от тези четири варианта са:  $\pm 1x + 0y + 0z \pm d$





# Произволна равнина

## Равнина може да се дефинира

- По три неколинеарни точки
- По точка и нормален вектор

## Дефиниране по три точки

- Има три точки  $P_i(x_i, y_i, z_i)$  за  $i = 1, 2, 3$
- С надежда, че те не са колинеарни  
(иначе няма да има еднозначно дефинирана равнина)

– Уравнение на равнината

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = 0$$

– След малко доработка се получава

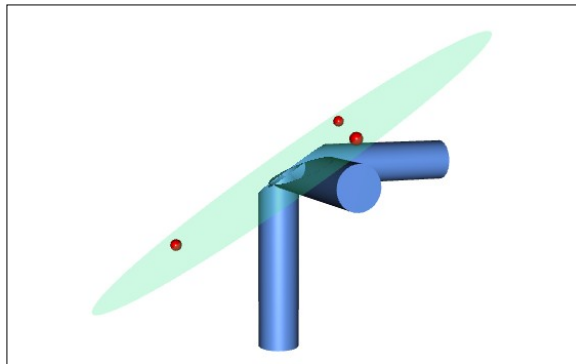
$$a = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)$$

$$b = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2)$$

$$c = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)$$

$$d = -[x_1(y_2z_3 - y_3z_2) + x_2(y_3z_1 - y_1z_3) + x_3(y_1z_2 - y_2z_1)]$$

- Методът е удобен, ако се знаят трите точки
- Не е интуитивен, ако не се знаят



# Нормален вектор

## Равнина може да се дефинира

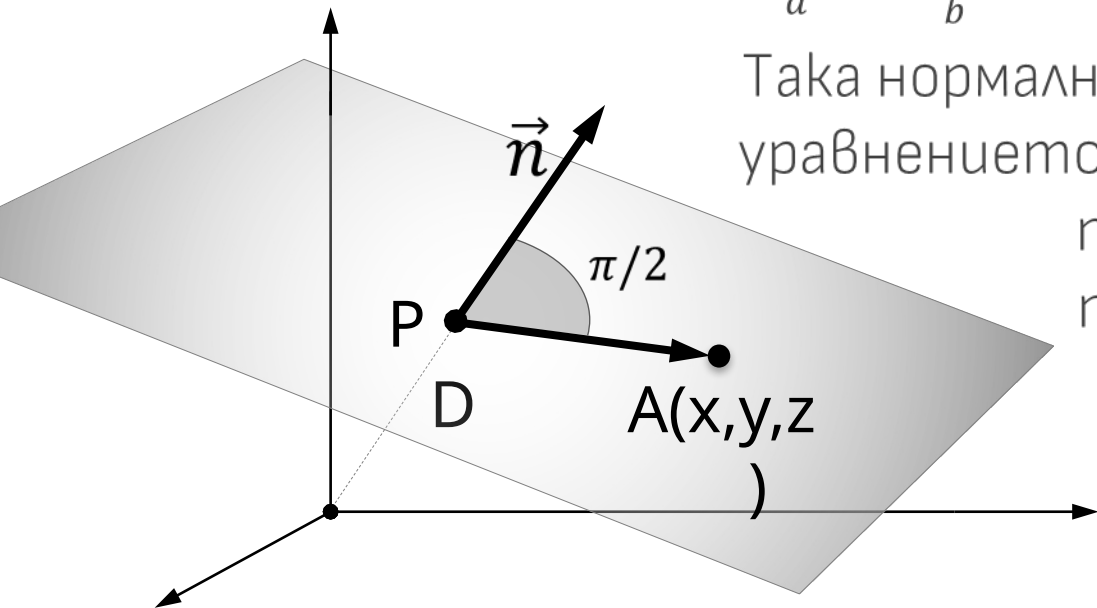
- Точка + нормален вектор
- Силно интуитивна представа: ако се хване векторът като ръчка, може да се върти равнината
- Ако се мести точката – мести се и равнината
- Дали е по-лесен и математически?

– Ако  $\vec{n}$  е нормалният вектор, а  $(x, y, z)$  е точка от равнината, то:  $\vec{n} \cdot (A - P) = 0$

$$\Rightarrow n_x(x - p_x) + n_y(y - p_y) + n_z(z - p_z) = 0$$

$$\Rightarrow \underbrace{n_x x}_a + \underbrace{n_y y}_b + \underbrace{n_z z}_c - \underbrace{n_x p_x + n_y p_y + n_z p_z}_d = 0$$

Така нормалният вектор определя 3/4 от уравнението на изрязващата равнина, последната ѝ четвъртина се получава от разстоянието D



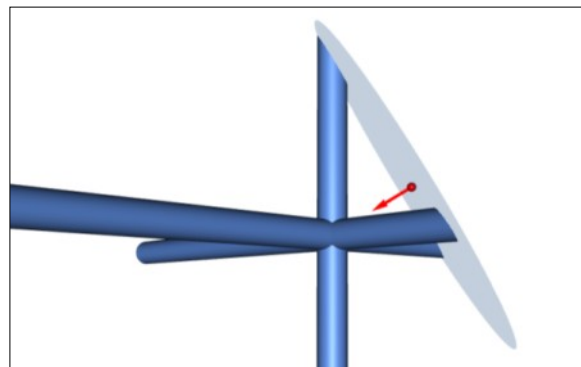
$$D = \frac{d}{\sqrt{a^2 + b^2 + c^2}}$$

## Интуитивността за равнината идва от

- Решава се колко е отдалечена от  $(0,0,0)$
- Решава се накъде е обърната
- Това еднозначно я определя

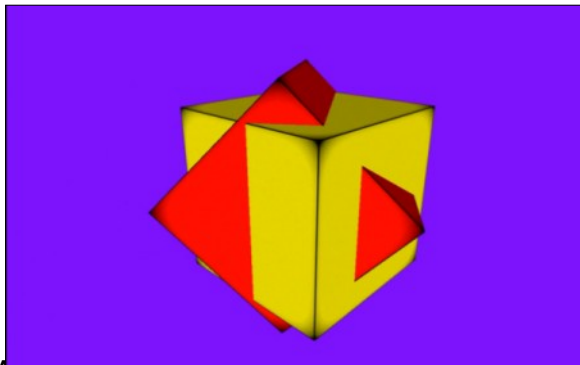
## А полупространството?

- Нормалният вектор  $\vec{n}$  буквално показва кое полупространство да остане



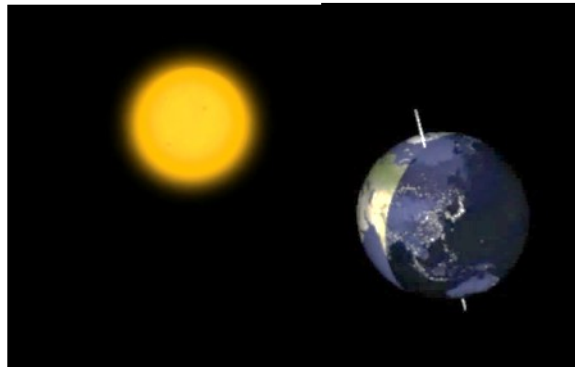
# Примери с изрязващи равнини

- Куб през куб
- Ден и нощ



"A cube through a cube of the same size"

<http://youtu.be/D-W2QMSXSG4>



"Day and Night"

<http://youtu.be/LCLw1s5oD8w>

Изрязване на стени



# Изрязване на стени

## Характеристики

- Изрязване на цели стени от графичен обект
- Изрязване на стени от мрежата му

## Две доста различни цели

- Премахване на задни стени
- Рисуване с прозрачности

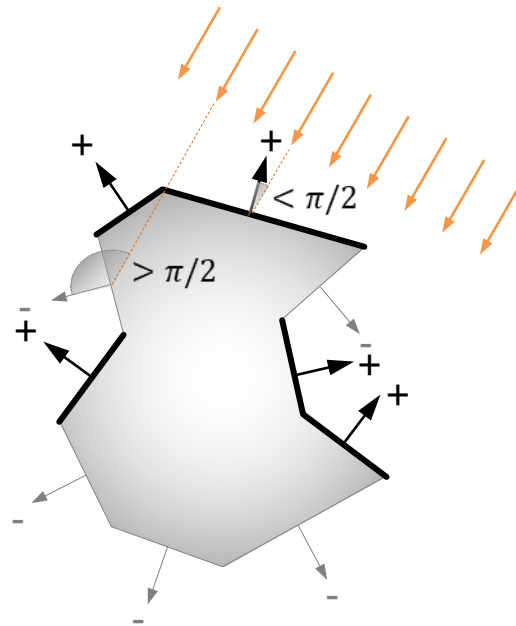
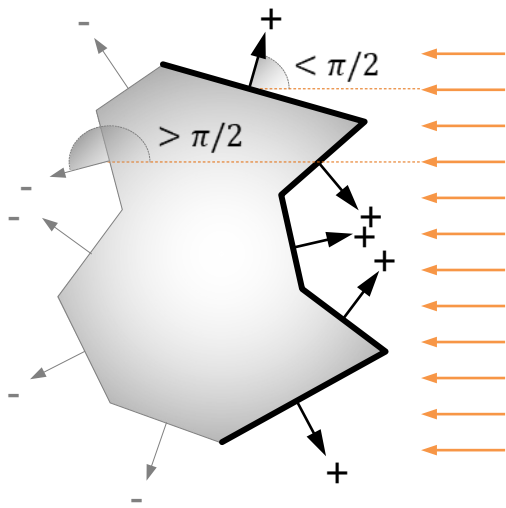
# Цел 1

## Премахване на задни стени

- Повечето обекти са плътни и непрозрачни
- Няма смисъл да се отделят ресурси за растеризиране на задните стени – те не се виждат
- Приложимо при липса на отражения, сенки и т.н.

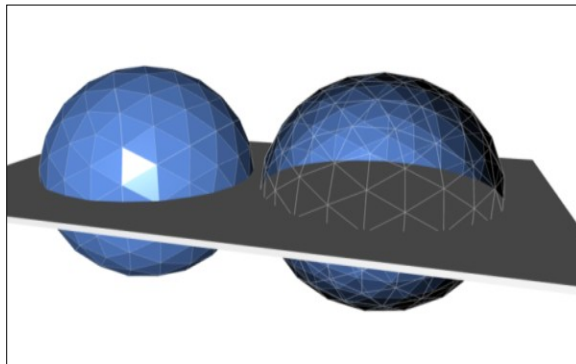
# Намиране на задни стени

- Знак на скалярно произведение
- От нормалния им вектор
- И гледната точка



# Бонус – няма принципна разлика

- Изрязване на задни стени
- Изрязване на предни стени



# Цел 2

## Рисуване с прозрачности

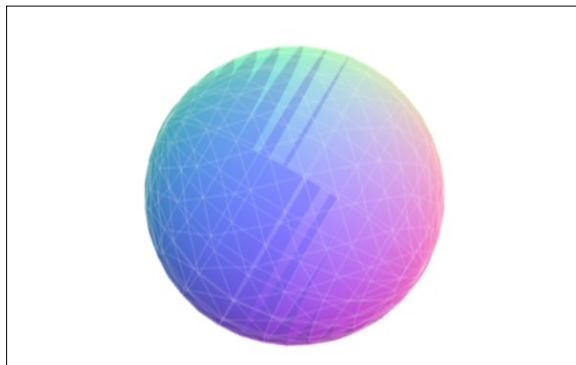
- Прозрачни стени – лесен алгоритъм за рисуване, ако те са подредени отзад напред

## Проблем

- При въртене на обекти
- При промяна на гледната точка

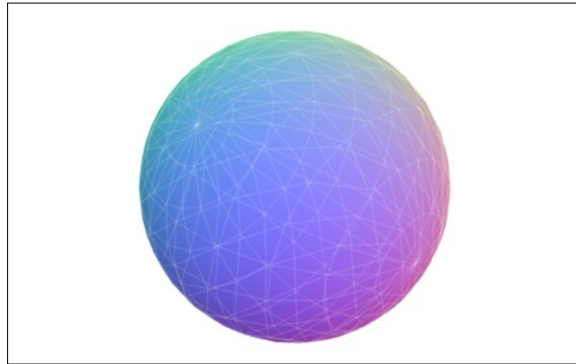
# Дефектна прозрачност

- Тежи да се сортират стените отзад-напред (особено, ако се прави на всеки кадър)
- Ако не се прави, става това:



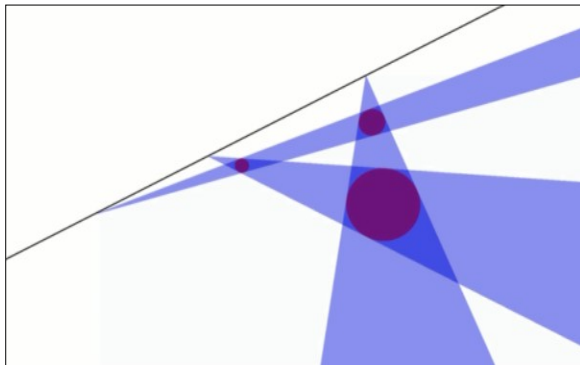
# Решение с рисуване на две фази

- Фаза 1: рисуват се всички задни стени
- Фаза 2: рисуват се всички предни стени



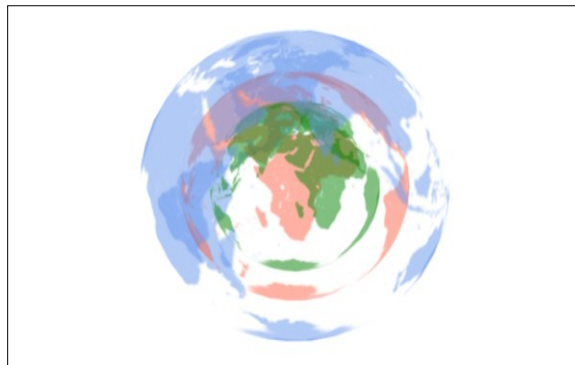
# Примери с изрязване на стени

- Теорема на Монж
- Вложени земни кълба



"Monge's Circle Theorem"

<http://youtu.be/LE3gQKeIyLM>





Изрязване на  
скрити линии

# Изрязване на линии

## Използване

- Премахване на линии, които са скрити
- Симулиране на непрозрачни стени

## Някои алгоритми

- Алгоритъм на художника
- Алгоритъм на Нюел-Нюел-Санча  
(Newell-Newell-Sancha)
- Използване на  $Z$ -буфер

# Алгоритъм на художника

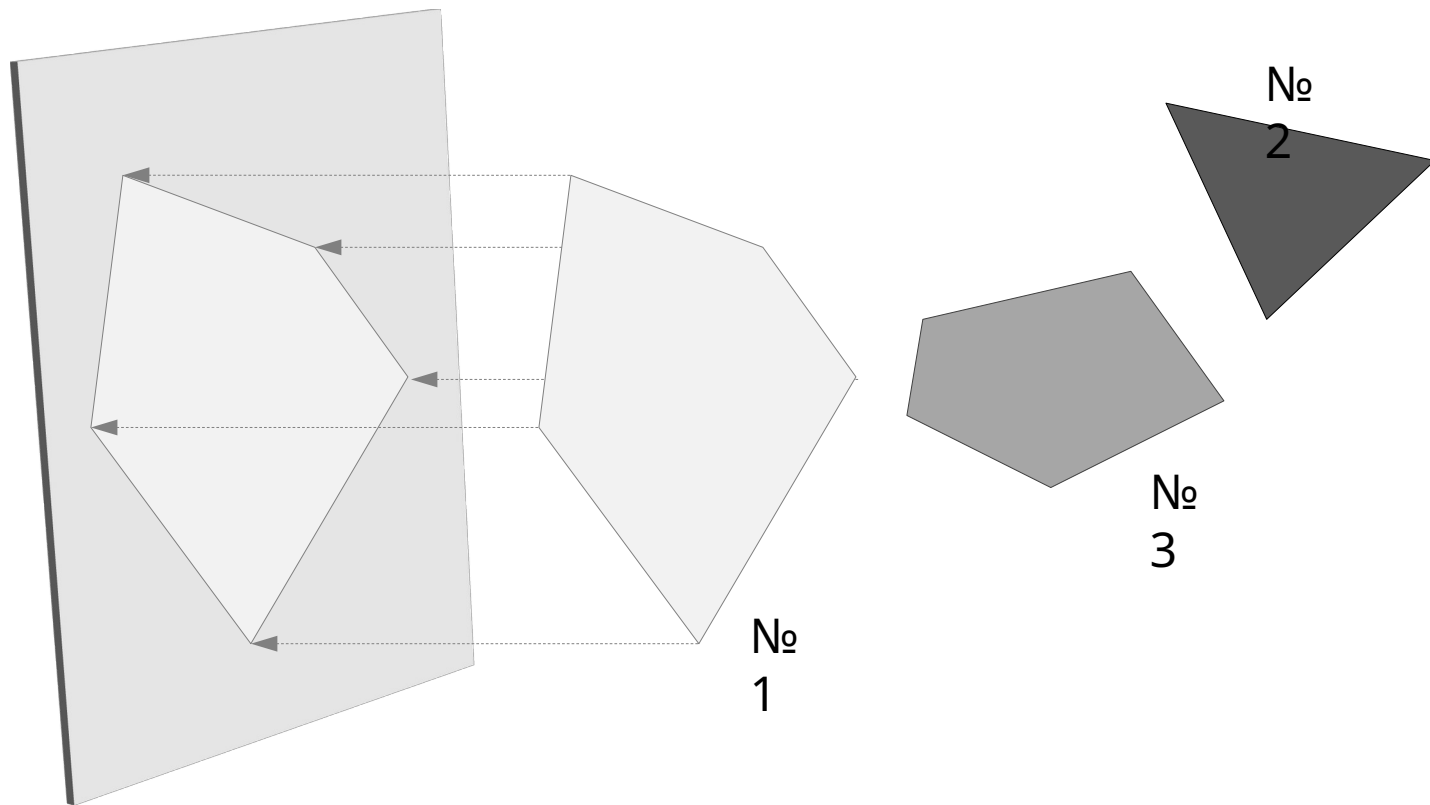
## Основна идея

- Стените имат фиксиран приоритетен номер
- Рисуват се според този номер
- Новите стени се рисуват върху старите, заличавайки скритите елементи

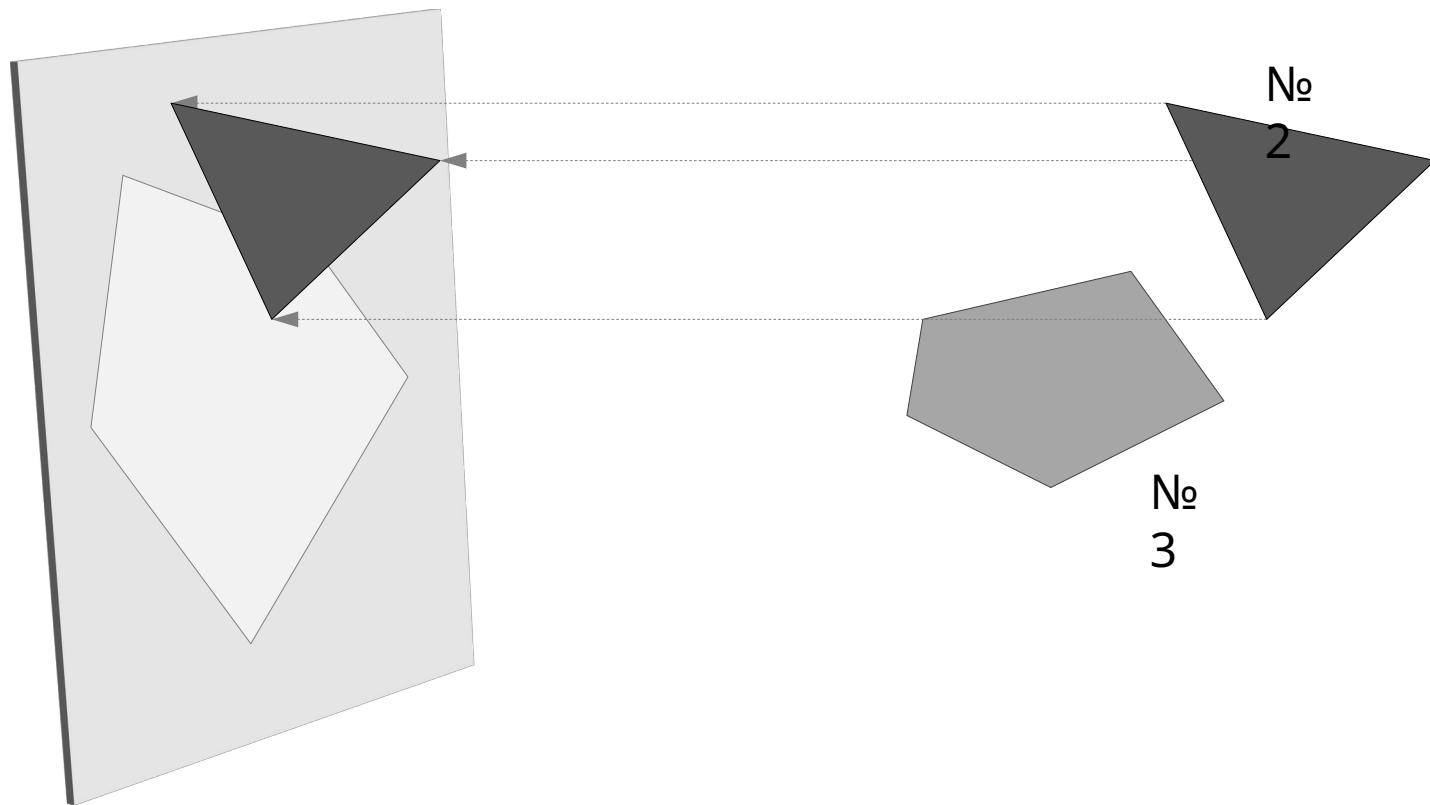
## Основна трудност

- Установяване на номерацията

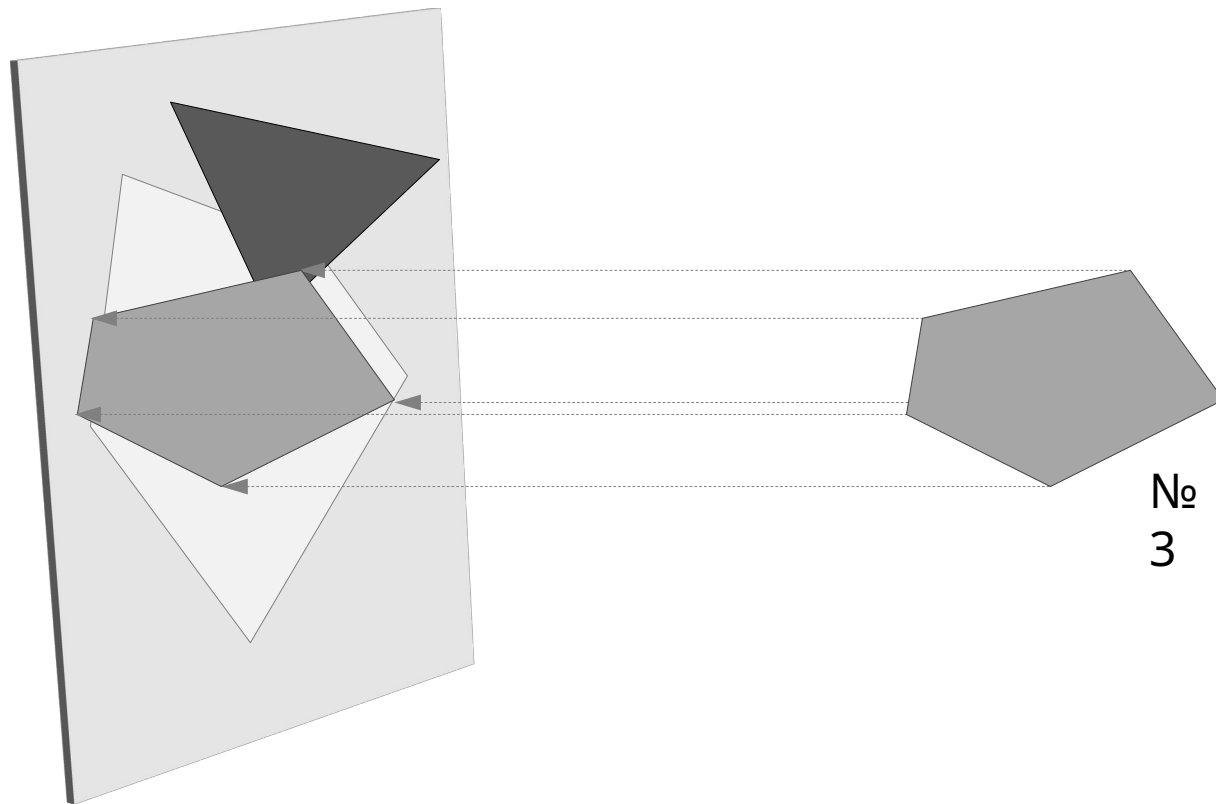
– Рисува се най-задния обект



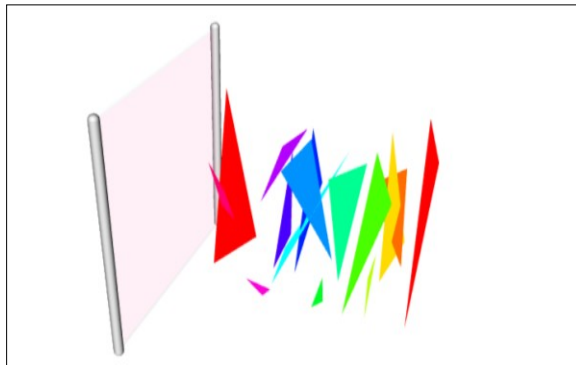
– Рисува се обектът над него



– Рисува се следващият обект



– Изглежда така



# Алгоритъм на Нюел-Нюел-Санча

## Основна идея

- Стените се рисуват последователно
- Всяка стена се рисува върху нарисуваното преди нея
- Ако две стени са пресичат, едната от тях се разделя

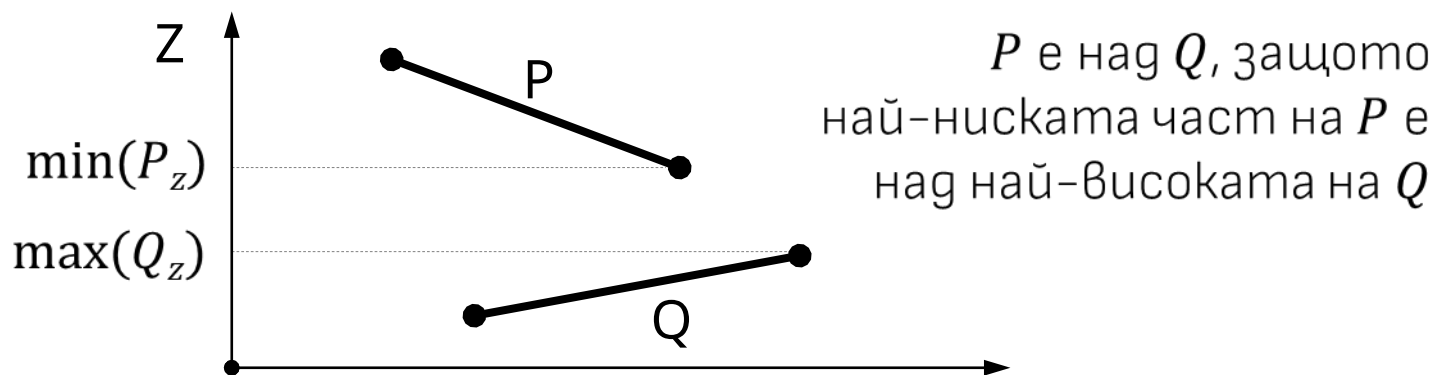


# Особености

- Подредбата се прави в реално време
- Приложим само за равнинни стени
- Работи се на стъпки, започва се от най-леката
- Ако някоя стъпка може да определи ред на рисуване, стъпките след нея не се прилагат

## Стъпка №1 – пресичане по $Z$

- Проверява се дали две стени имат пресичане по  $Z$
- Ако нямат, по-далечната може да се нарисува първа
- Понякога се прави предварително сортиране по  $Z$



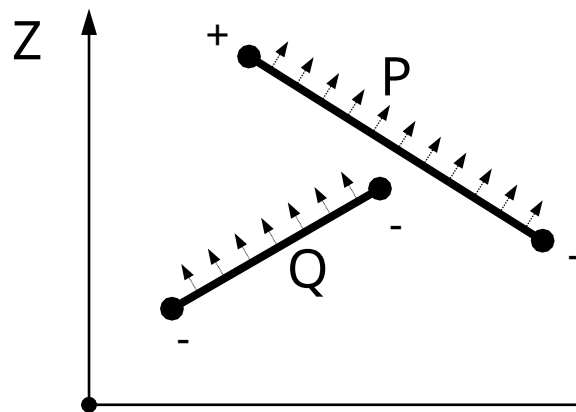
## Стъпки №2 и №3 – пресичане по $X$ и $Y$

- Дали две стени имат пресичане по  $X$  и после по  $Y$
- Ако нямат, значи двете могат да се нарисуват независимо една от друга
- Картинката е аналогична

## Стъпка №4 – част от полупространство

- Всяка стена разделя пространството
- Ако обект е изцяло в едното полупространство, значи ясно е над или под
- Двупосочна проверка  $P \leftrightarrow Q$  и  $Q \leftrightarrow P$

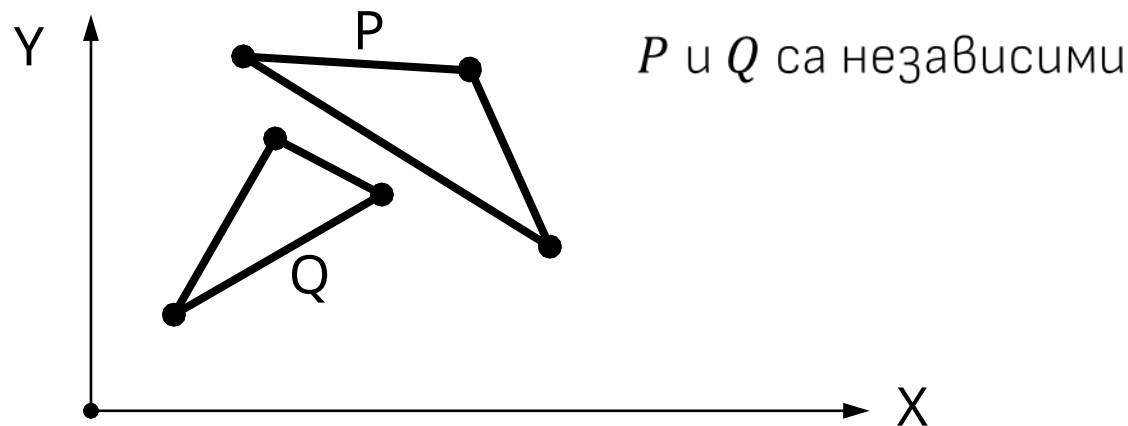
Проверката на  $P$  спрямо  $Q$  не дава никакъв резултат!



$Q$  е под  $P$ , защото всички точки на  $Q$  са под равнината на  $P$

## Стъпка №5 – сечение

- Ако растеризираните образи на  $P$  и  $Q$  не се пресичат, значи не си пречат
- Реализира се чрез сечение на многоъгълници



## Ако все още не може да се определи

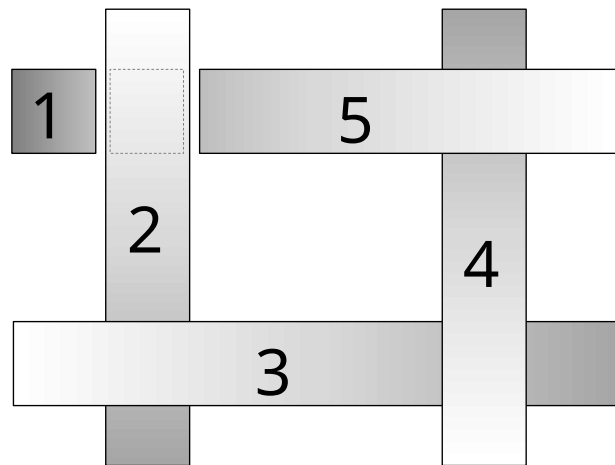
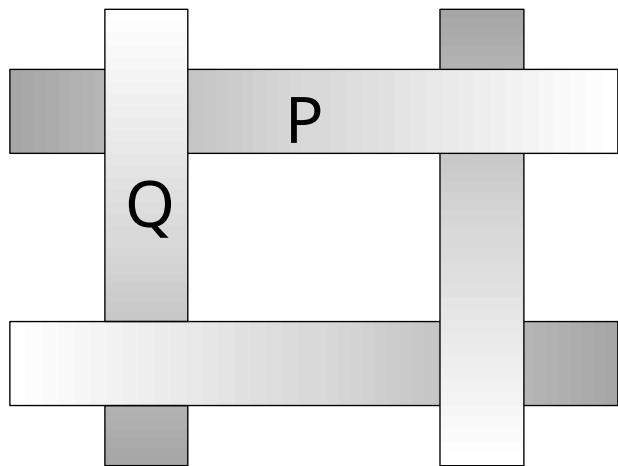
- Че  $P$  е преди  $Q$
- Или че  $Q$  е преди  $P$
- Или че  $P$  и  $Q$  са независими

## Едната стена се разделя

- Според сечението с другата стена
- Използва се резултатът от стъпка №5

# Проблемна конфигурация

— Разбиране на  $P$  от  $Q$



# **Z-буфер**

## **Какво е Z-буфер**

- Допълнителен растерен слой
- Всеки пиксел съдържа дълбочина

## **Характеристики**

- Скоростта не зависи от броя стени
- Съществена е дълбочината на буфера



## Процедура на $Z$ -буфер

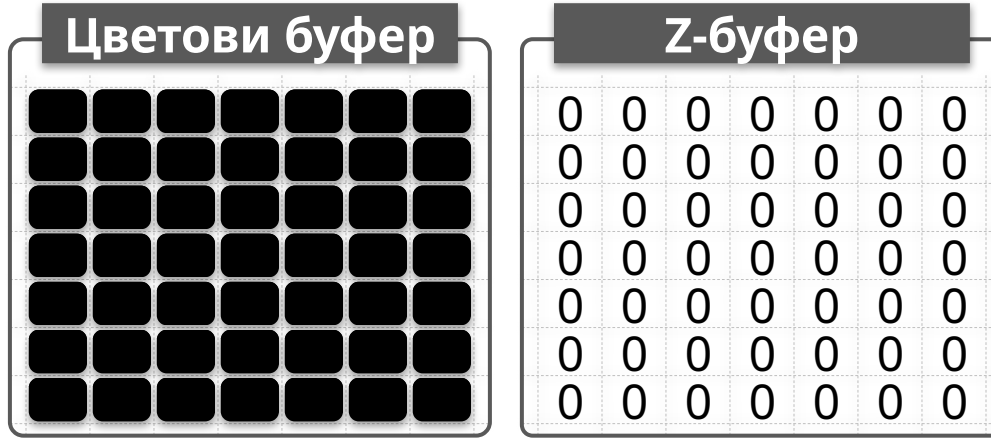
- Работи се пиксел по пиксел
- Пиксел с цвят  $C_{xy}$  и дълбочина  $Z_{xy}$
- От друг примитив се получава за същия пиксел нов цвят  $C$  и нова дълбочина  $Z$

### Ако...

- $Z$  е по-близка от  $Z_{xy}$  то  $C_{xy} \leftarrow C$  и  $Z_{xy} \leftarrow Z$
- $Z$  е по-далечна, то се игнорира  $C$  и  $Z$

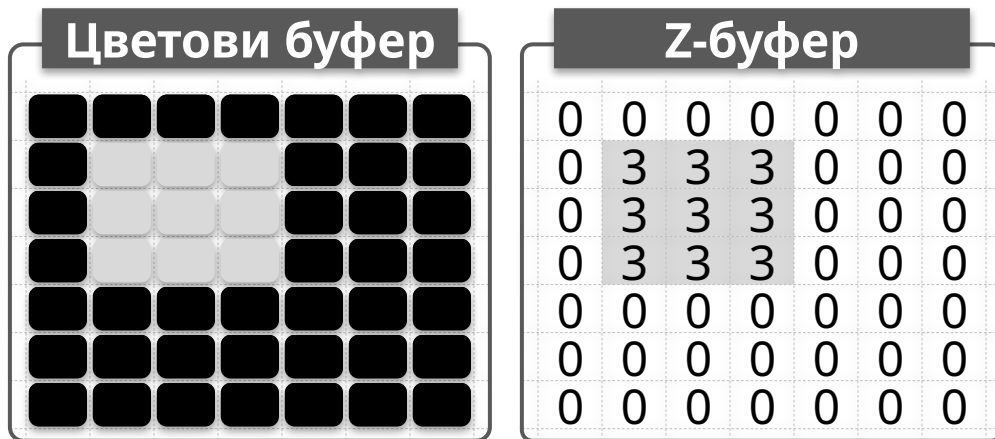
## Примерна начална конфигурация

- Двата буфера са празни (занулени)



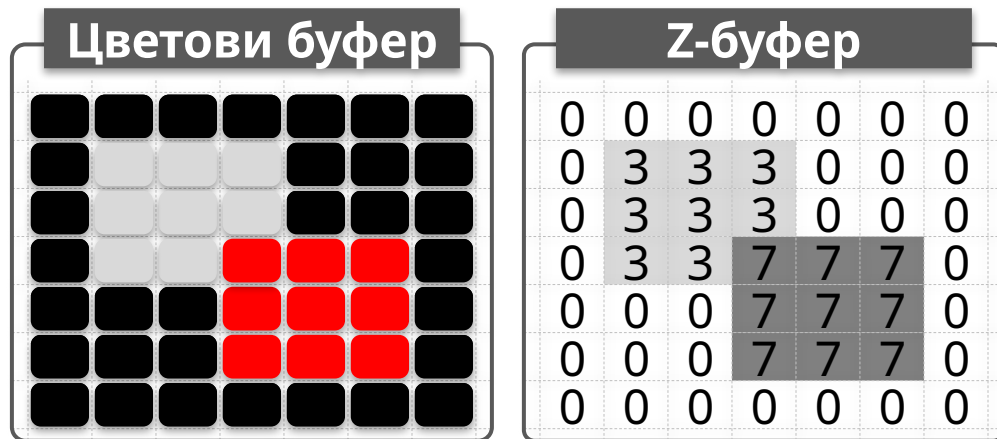
# Бял квадрат на дълбочина 3

- Само за примера се приема за **Z**-буфера:  
Най-далечният пиксел е 0, а най-близкият е 9  
Примитивите са хоризонтални квадрати



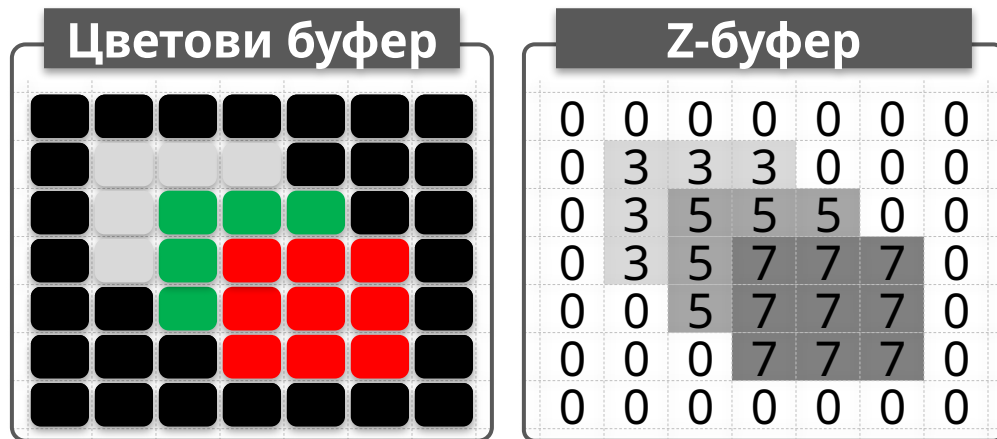
# Червен квадрат на дълбочина 7

- Червените пиксели са над белите и черните
- Дълбочина 7 е по-плитка от 0 и от 3



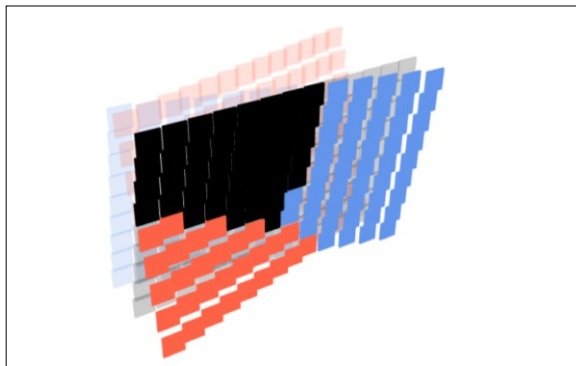
# Зелен квадрат на дълбочина 5

- Зелените пиксели се разполагат над белите
- В същото време те не променят червените



# Илюстрация

- С наклонени равнини
- Пресичащи се



Въпроси?

# Повече информация

<b>AGO2</b>	стр. 266-284
<b>ALZH</b>	гл. 6 и 7
<b>KLAW</b>	стр. 161-176
<b>LASZ</b>	стр. 145-154, 248-253
<b>MORT</b>	стр. 308-312
<b>SEAK</b>	стр. 39-41, 73, 161, 169

## А също и:

- Wolfram Mathworld: Plane  
<http://mathworld.wolfram.com/Plane.html>



Край