

СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“



Графични примитиви

ТЕМА №9

Съдържание

Тема 9: Графични примитиви

- Моделиране
- Графични примитиви
- Свойства
- Примери

Моделиране

Графичните обекти

Основна дейност в КГ

- Създаване и визуализиране на графични сцени

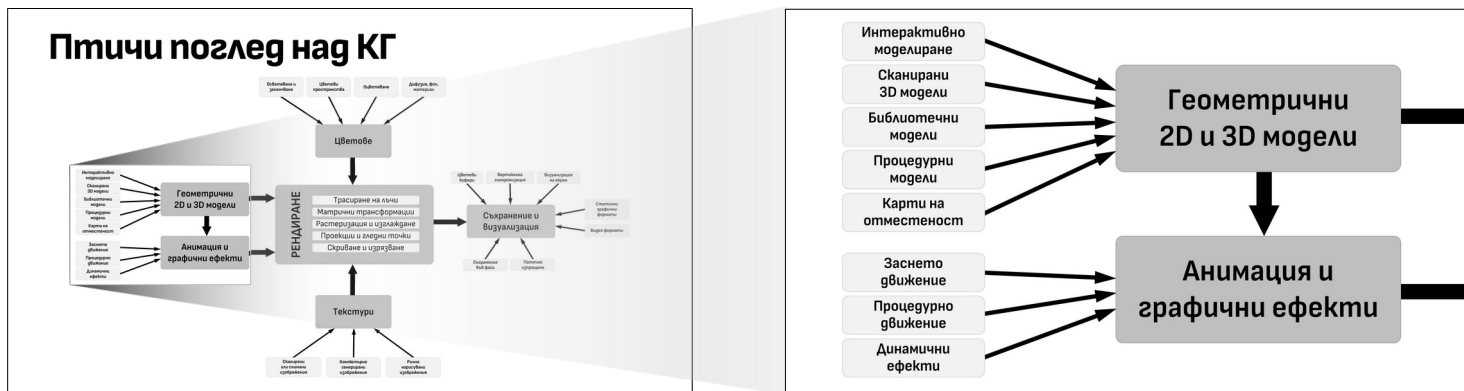
За целта

- Използват се графични обекти
- Те са безкрайно, но изброимо много и различни
- Някоя графична система не ги поддържа всичките
(това всъщност е добра новина)

Моделиране в КГ

Основна дейност в КГ

- Моделиране и визуализиране на изображения и сцени



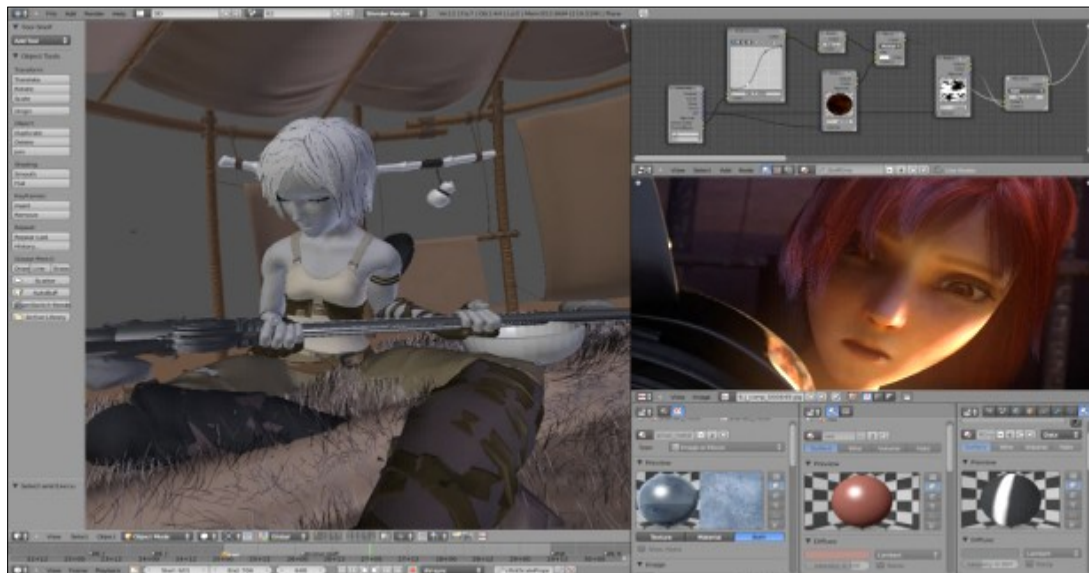
Интерактивно моделиране

- CAD/CAM системи 3D редактори
CAD = computer-aided design
CAM = computer-aided manufacturing
- Използване за архитектурен и инженерен дизайн,
виртуални 3D герои за филми и игри



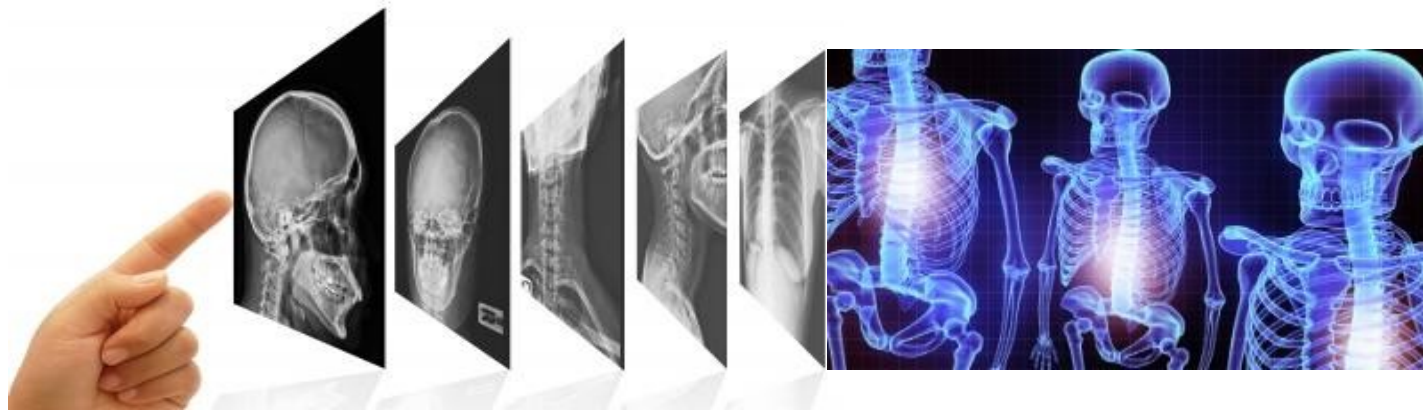
Графични примитиви

- Моделите се правят от широк набор графични примитиви, които допълнително се модифицират
- Някои среди имат и скриптов езици



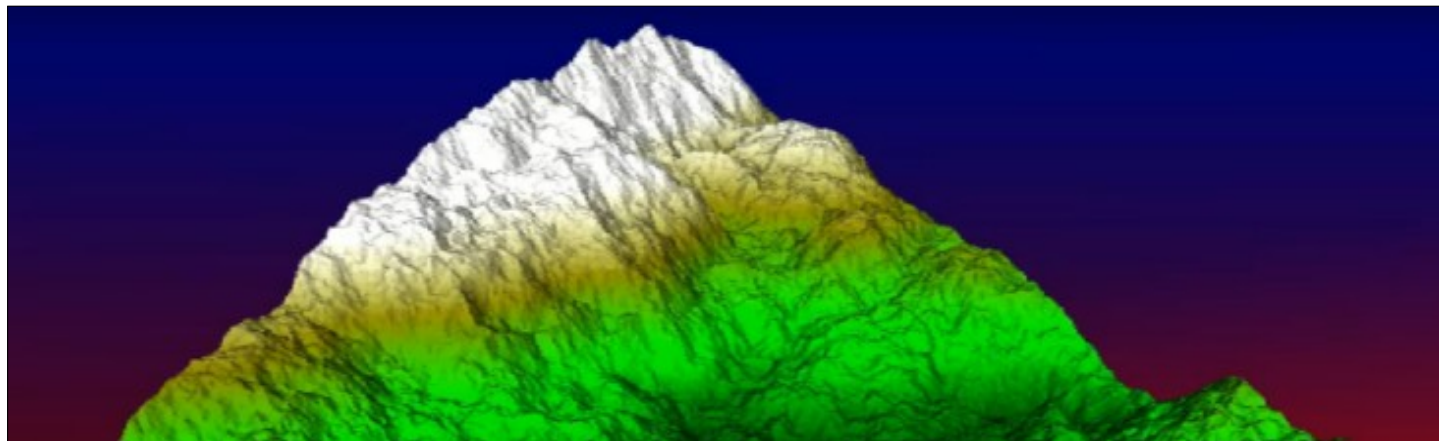
Сканирани 2D/3D модели

- Сканиране за медицински цели
- Сканиране за виртуални 3D модели
- Създаване на първични модели, които се дообработват ръчно



Процедурно моделиране

- Създаване чрез алгоритъм
- Традиционни геометрични обекти – сфера, куб, ...
- Сложни, математически дефинирани форми



Други подходи

Карти на отместеност

- Кодиране на обемност чрез цвят в изображения
(аналогично на релефна карта)

Библиотечни модели

- Библиотеки, които предоставят „вградени“ модели
(често тези модели са генерирани по някой от другите начини)

Графічні примітиви

Графични примитиви

Базисни графични обекти

- Наричани още графични примитиви
- С тях се създават останалите графични обекти

Сложност

- Повечето примитиви не са примитивни
- Вътрешна йерархия на примитивите

Размерност на примитивите

- Три различни размерности
- Обектите притежават и трите

Обектна размерност

- Геометрична размерност на самия обект
- Точката е 0D, отсечката е 1D, квадратът е 2D, кубът е 3D, а тесерактът е 4D

Пространствена размерност

- Размерност на пространството, където е обектът
- В планиметрия е 2D пространство
- В стереометрията е 3D пространство

Визуална размерност

- Обектна (геометрична) размерност на елементите съставлящи образа на обект
- Рисуване с отсечки е едномерно представяне

Пример

Квадрат висящ „във въздуха“ има

- Едномерна визуална размерност
- Двумерна обектна размерност
- Тримерна пространствена размерност

Коя размерност се има предвид?

- Зависи от контекста на споменаване
(Например за 3D точка най-често се очаква пространствената размерност)

Графични примитиви

OpenGL

- Стандарт за графичен програмен интерфейс и в същото време е библиотека
- Прародител на WebGL

Three.js

- Графична библиотека от високо ниво
- Използваме я на упражнения
- Вътрешно използва WebGL и WebGPU

Примитиви в OpenGL

Примитиви

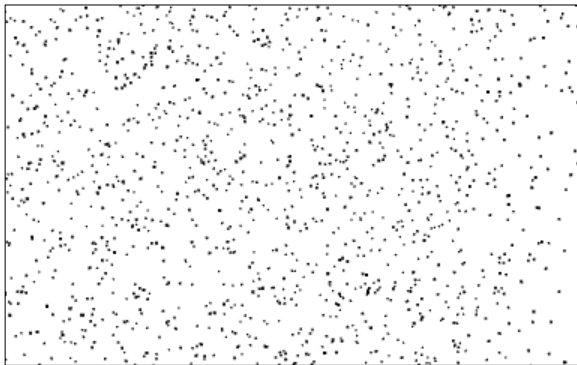
- Създават се с поредица от координати на върхове

Използване на върховете

- По различен начин от различните примитиви
При някои те са части от обектите (напр. отсечка)
При други те характеризират обекти, но не са част от тях (напр. сплайн)

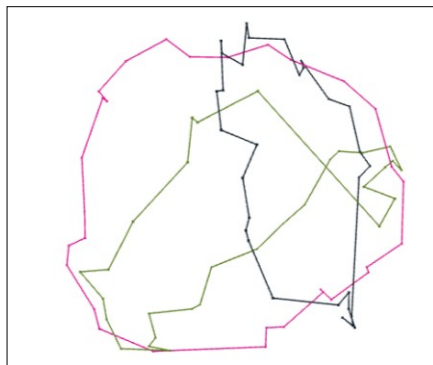
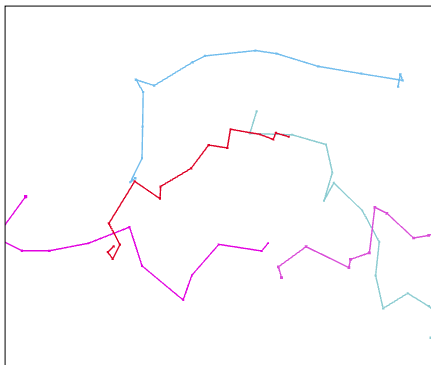
Точки

- Поревица от 3D координати
- Определят независими точки в пространството



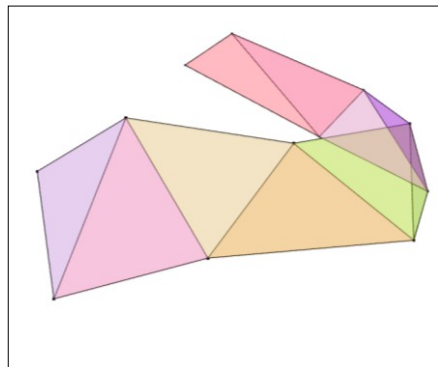
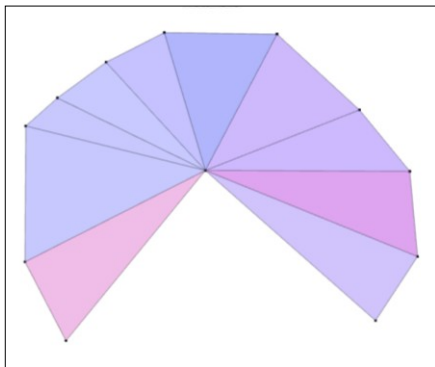
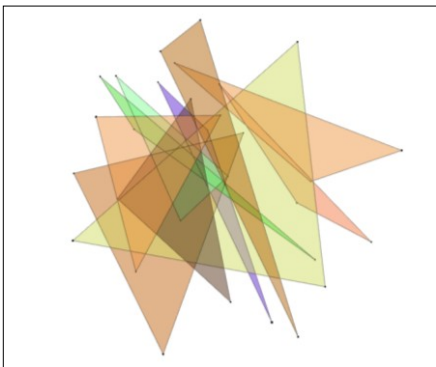
Линии

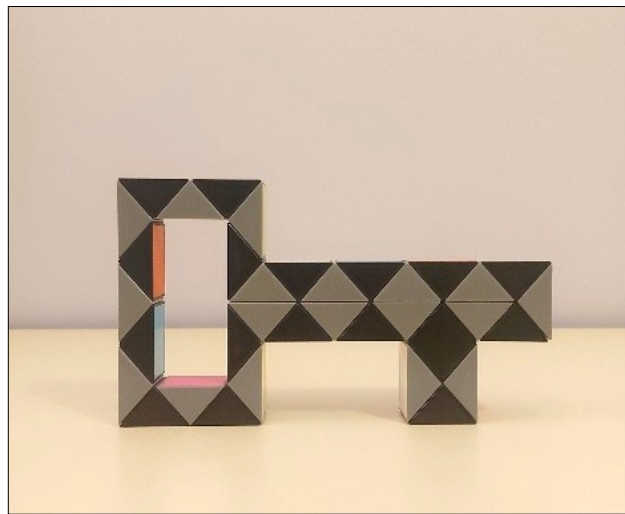
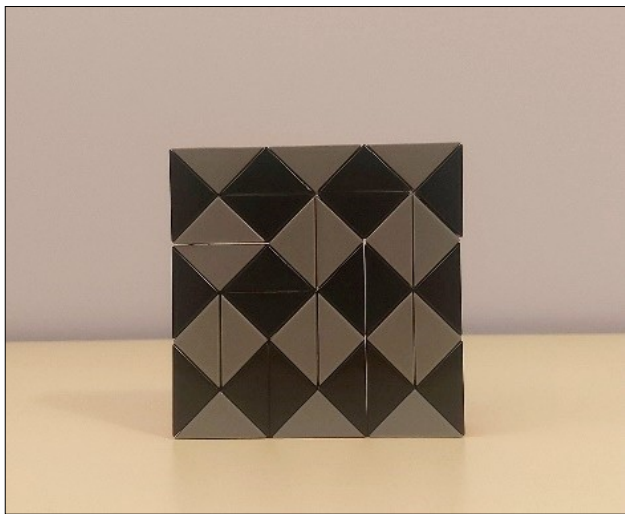
- Независими линии (отсечки)
- Начупени линии
- Затворени линии (примки)



Триъгълници

- Независими триъгълници
- Ветрила от триъгълници
- Ивици от триъгълници





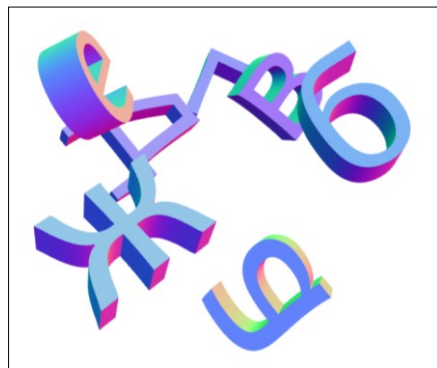
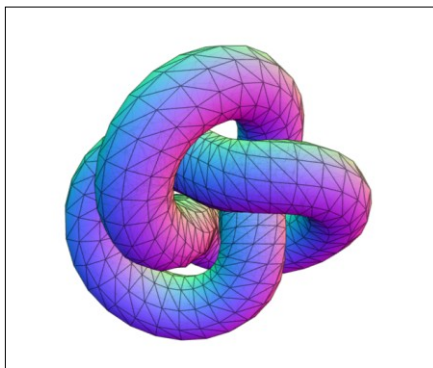
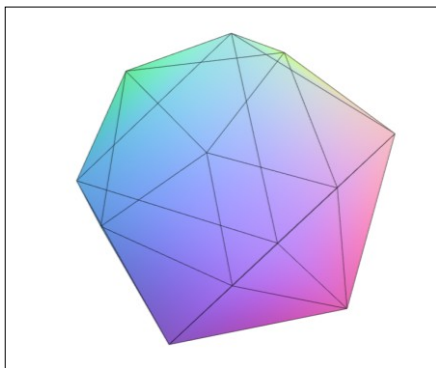
Примитиви в ThreeJS

Примитивни примитиви

- 0D – точки
- 1D – линии
- 2D – равнини, окръжности, пръстени
- 3D – кубове, сфери, цилиндри, конуси

Някои други примитиви

- Икосаедър
- Торичен възел
- Тримерен текст



Свойства

Свойства на обектите

Роля на свойствата

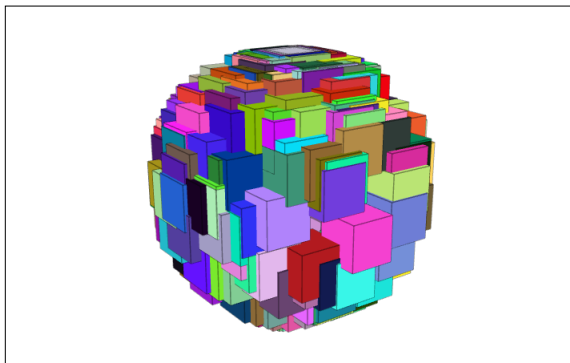
- Примитивите могат да се деформират
- Получават се производни обекти

Общи свойства

- Позиция, размер и ориентация
- Поделементи

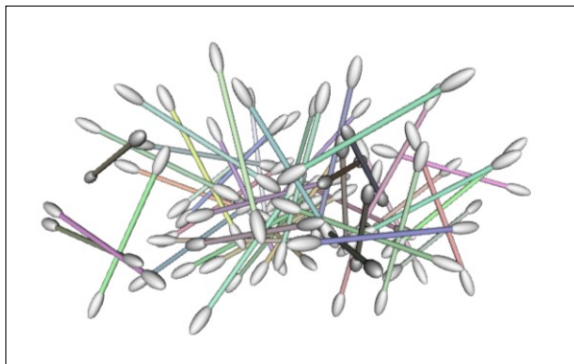
Позиция

- Невидима 3D точка за положение на обект в сцена
- Обектът е „закачен“ за нея, тя е „началото“ му
- С тази точка се променя положението на обект



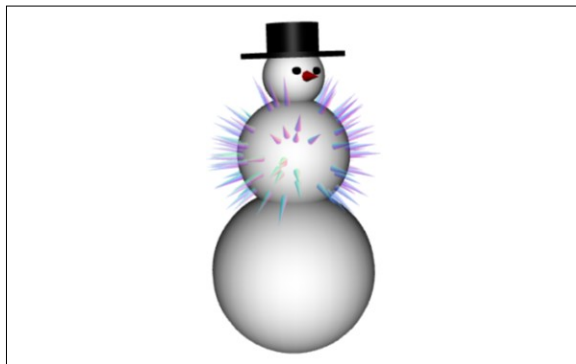
Размер

- Три размера – мащаби по осите
- Смачкване или разпъване на обекта
- От куб се получава паралелепипед



Ориентация

- Три ъгъла около невидими оси
(Защо точно три ще се разбере в тема №14)
- Обект е завъртян около осите



Поделементи

- „Изключване“ на част от обект
- Окрупняване на по-едри фрагменти

По този начин

- От конус се получава пресечен конус
- От сфера – полусфера
- От окръжност – многоъгълник
- От цилиндър – призма

Пет плюс пет пéти

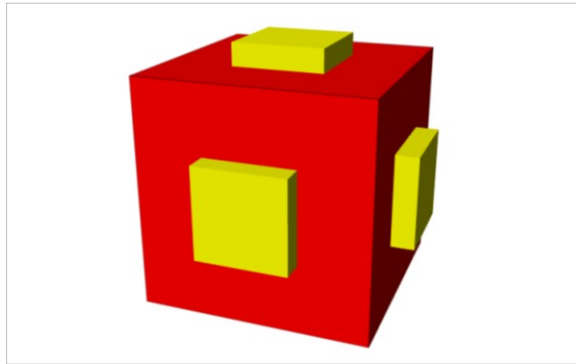
примитивни примера

(предимно примери, показани посредством примитиви)

Куб с издутини

Да се нарисува куб с 6 издутини

- С минимален брой примитиви
- Очевидно, „минимален“ <7

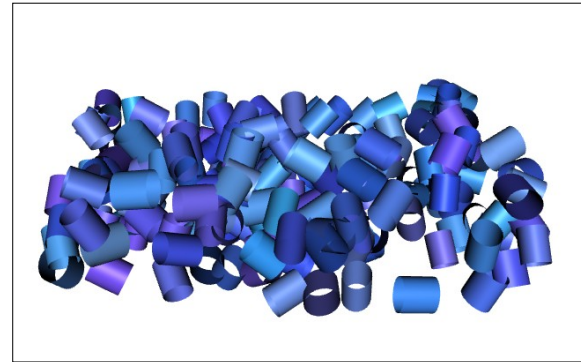
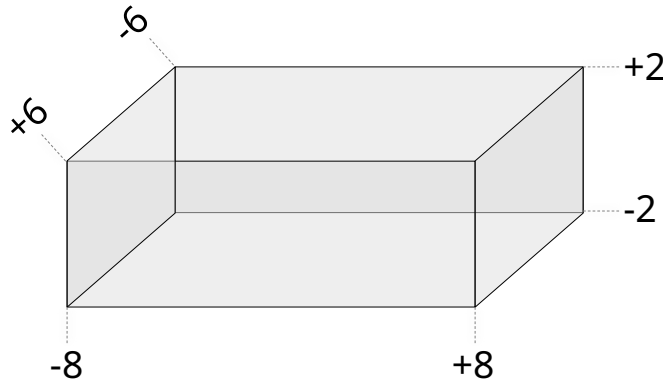


Японска възглавница



Реализация

- Пси – $\Psi(x, y)$ е случайно число в $[x, y)$
- Координати $\vec{p}[\Psi(-4,4), \Psi(-1,1), \Psi(-3,3)]$
- Ориентация $\vec{r}[\Psi(0,2\pi), \Psi(0,2\pi), \Psi(0,2\pi)]$
- Цвят $\vec{c}[\Psi(0.2,0.5), \Psi(0.3,0.7), 1]$



Форма за кекс

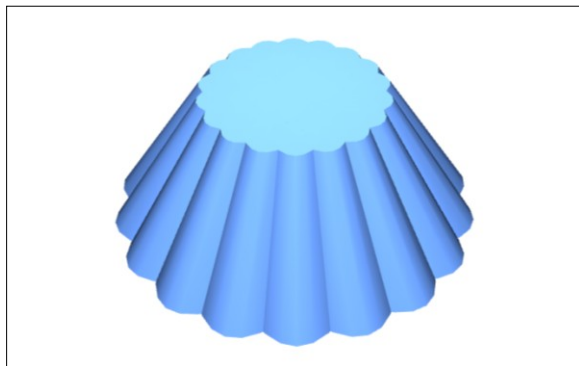
Реализация с n пресечени конуса

– Сплескани конуси с $r_x = \frac{2}{5}r_z$

– Правим копия чрез завъртане $a = \frac{\pi}{n}$

Бонус 1т.

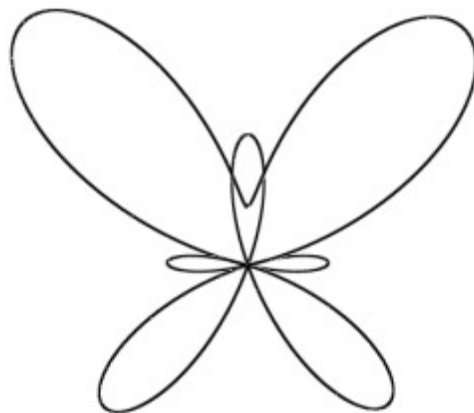
Защо π , а не 2π ?



Пеперуда

Да се нарисува пеперуда

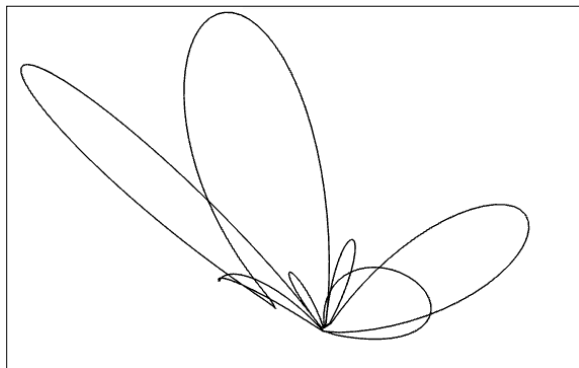
- Параметрично! С полярни координати!
- Но и да се параметризира точноста



Реализация

- Нужна ни е $R(\alpha) = e^{\cos \alpha} - 2 \cos 4\alpha + \left(\sin \frac{\alpha}{12}\right)^5$
- Разбиваме на n точки $P_1 \dots P_n$, свързани с отсечки
- При $\alpha_k = \frac{2\pi k}{n}$, P_k е с полярни координати $(R(\alpha_k), \alpha_k)$ и със сферични $(R(\alpha_k) \sin \alpha_k, 0, R(\alpha_k) \cos \alpha_k)$

Бонус от 2т
да иска
някой?



Тор

Да се нарисува тор

- Математически, а не селскостопански
Синоними: кравай, геврек, поничка...



"Mathematics ... is infinitely sweet"

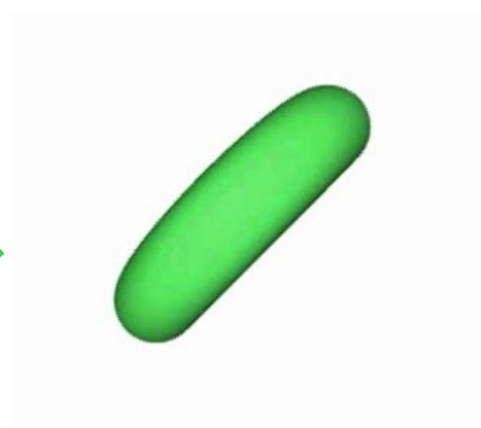
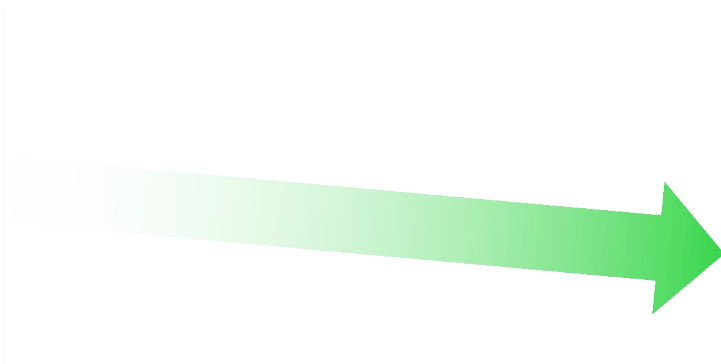
<http://youtu.be/eADcA3iouCk>

Много идеи

- Идея №1: готов примитив в ThreeJS
Не, целта е да се създаде „ръчно“
- Идея №2: параметрично уравнение
Не, целта е да се сглоби от примитиви
- Идея №3: завъртане на изправен кръг
Не, целта е повърхност, не само арматура
- Идея №4: завъртане на къс цилиндър
Не, целта е идеално снаждане

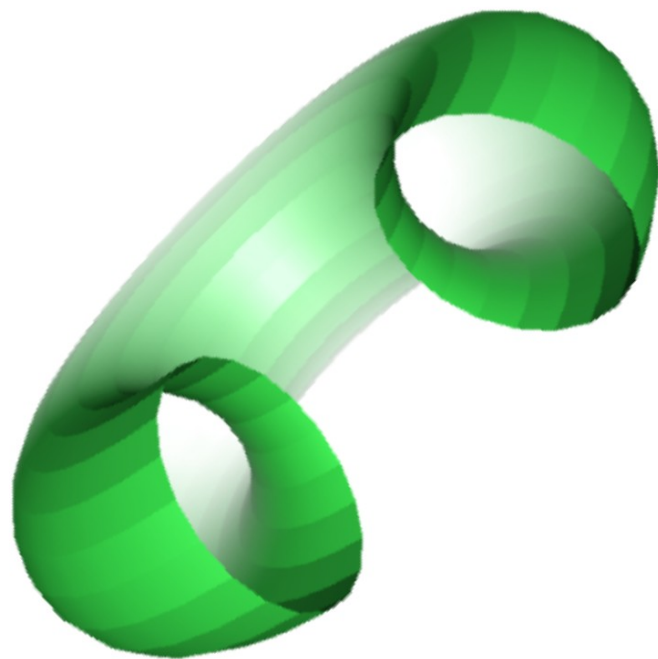
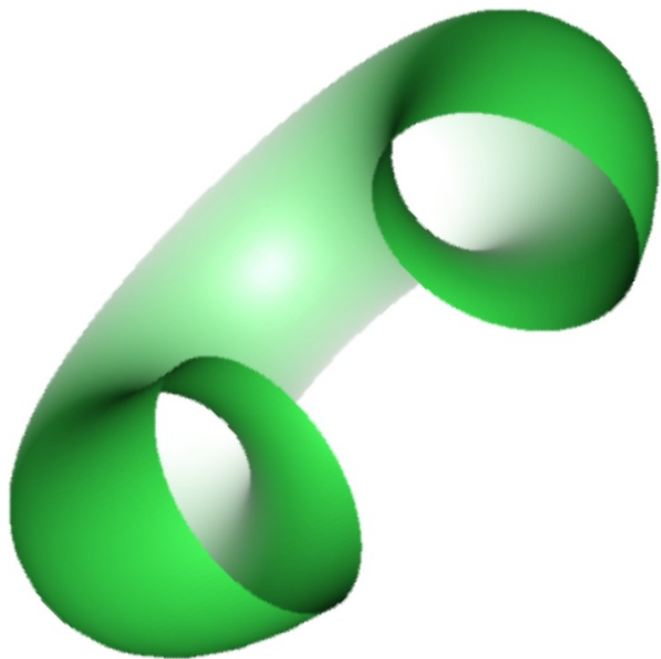
Идея №5: с пресечени конуси

- Да, пресеченият конус не е част от тор (ама никак – това е нечовешки абсурдна идея)



Торът се нагруббява (т.е. прави се по-груб)

- Така се разпознават отделните примитиви, ако изобщо има такива



Още малко по-груб

- Отлепват се отделните „пръстени“
- Те са стени на пресечени конуси



Алгоритъм

- Представя се напречното сечение на тор — окръжност — като n -ъгълник
- Всяка негова страна, завъртяна около вертикална ос, образува околната стена на пресечен конус
- При достатъчно голямо n торът ще изглежда гладък

А сега математическото представяне

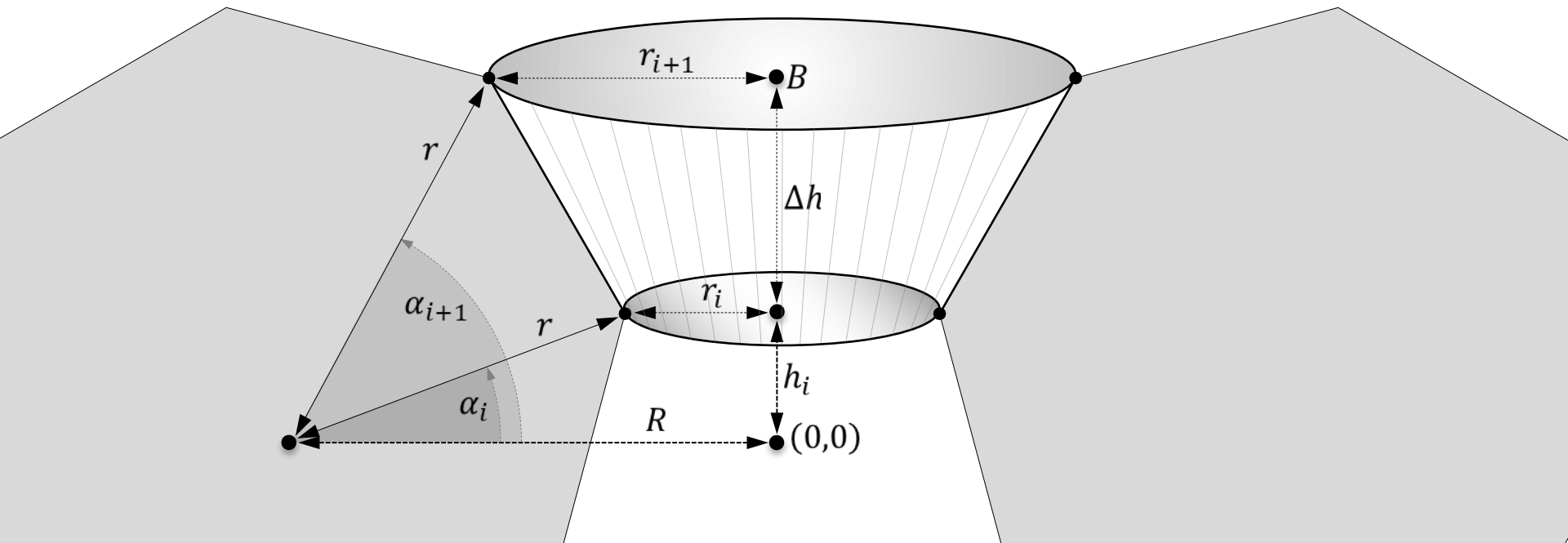
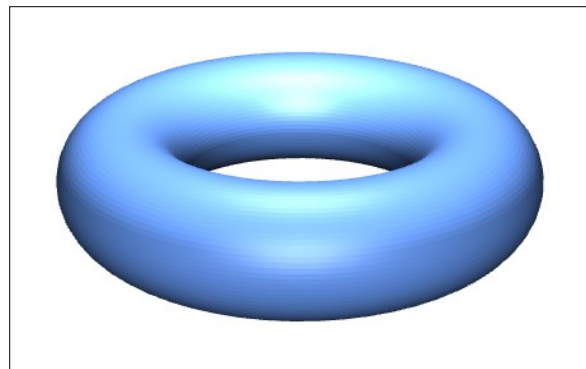
- Дадено е n (брой стени), R (радиус на тора) и r (радиус на сечението)

$$\alpha_i = \frac{2\pi}{n} i$$

$$r_i = R - r \cos \alpha_i$$

$$h_i = r \sin \alpha_i$$

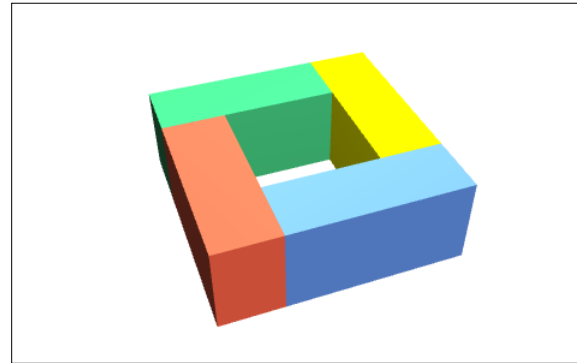
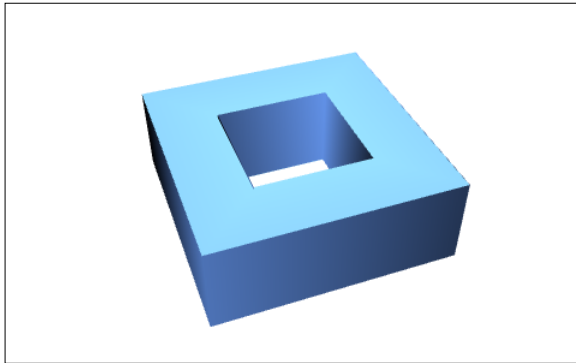
$$\Delta h = h_{i+1} - h_i$$



А по-ръбест?

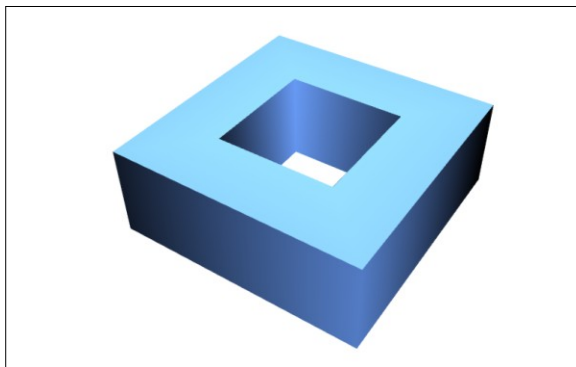
А може ли да е толкова по-ръбест

- Че да изглежда като левия пример и да не е направен като десния
- Да се получава пак от пресечени конуси



Разбира се. Ето как:

- Сечението е толкова грубо, че е четириъгълник
- Начален ъгъл $\alpha_0 = \frac{\pi}{4}$ изправя ромбовете в квадрати
- Пресечените конуси са груби като четириъгълници



Въпроси?

Повече информация

SEAK стр. 12-18 **KLAW** стр. 129-139
AGO2 стр. 164-166 **MORT** стр. 205-214
ALZH гл. 5.4, E.2, E.3
BAGL стр. 13-19

А също и:

- OpenGL - The Industry Standard for High Performance Graphics

<http://www.opengl.org>

Край