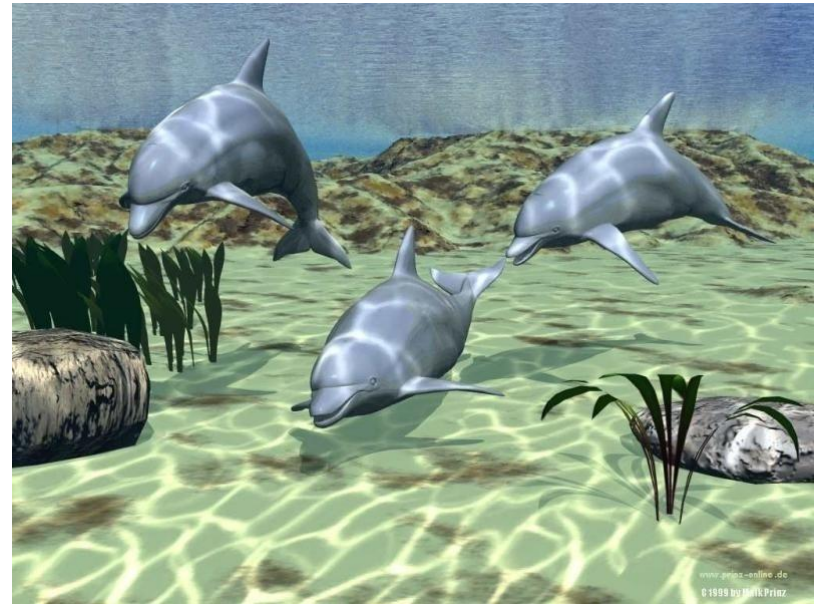


# Трёхмерная графика, 3D графика

**Трёхмерная графика** - компьютерная графика для отображения изображений, имеющих длину, ширину и глубину.

Трёхмерная графика в отличие от двухмерной дает более реалистичное представление образов

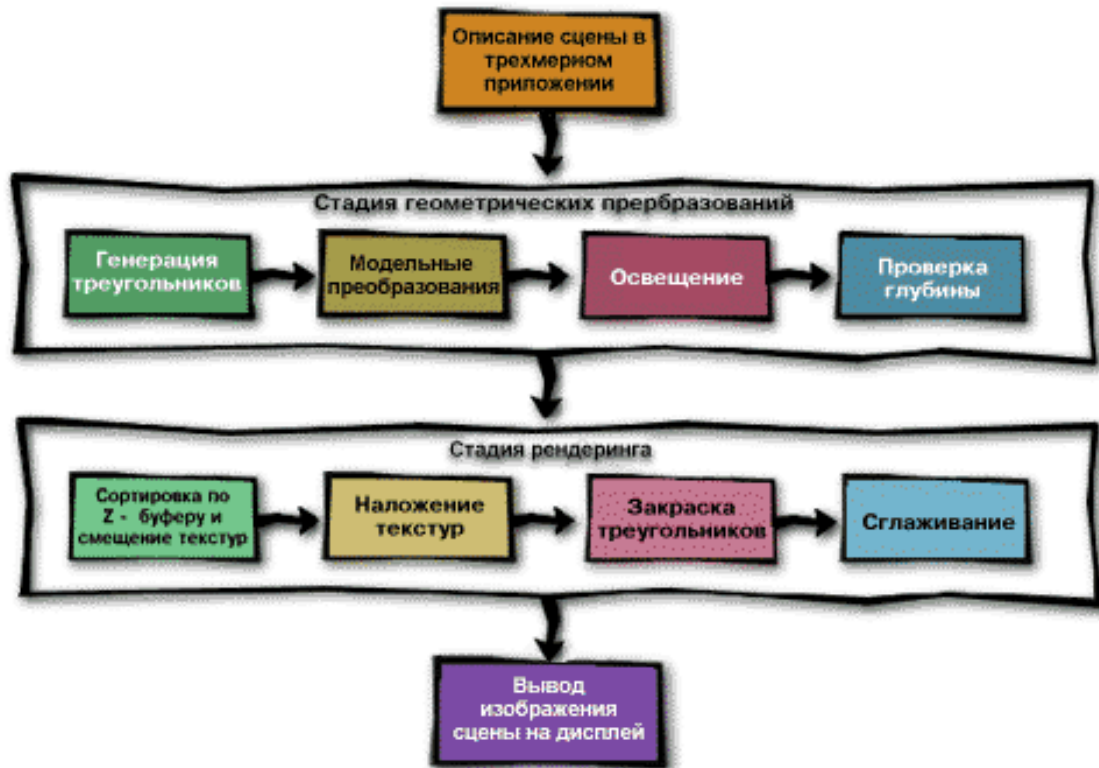
Главной целью системы трёхмерного графического синтеза является создание высококачественных фотореалистичных изображений со скоростью видео.



# Графический конвейер

В большинстве подсистем трехмерной графики применяется графический конвейер.

**Конвейер** - это логическая группа вычислений, выполняемых последовательно, которые дают на выходе синтезируемую сцену



---

Конвейер разделен на множество этапов, на каждом из которых *аппаратно* или *программно* выполняется некоторая функция. Наличием переходов между этапами конвейера обеспечивается возможность выбора между программной и аппаратной реализацией очередного этапа.

Такой подход к настройке конвейера позволяет приложениям трехмерной графики получать преимущества аппаратной реализации

Реализация конвейера может быть чисто *программной*, полностью *аппаратной* или *смешанной* (программно-аппаратной)

---



## *Описание сцены*

До начала работы геометрических преобразований необходимо описать трехмерную сцену, изображение которой необходимо синтезировать.

Трехмерное приложение оперирует объектами, описанными в некоторой глобальной системе координат.

Чаще всего здесь используется ортогональная (декартова) система координат, в которой положение каждой точки задается ее расстоянием от начала координат по трем взаимно перпендикулярным осям  $X$ ,  $Y$  и  $Z$ .

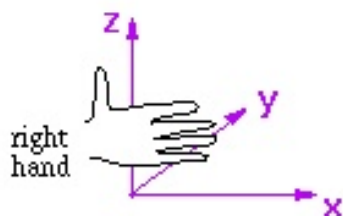
В некоторых случаях используется и сферическая система координат, в которой положение точки задается удалением от центра и двумя углами направления.

В глобальных координатах приложение создает объекты. В этом же пространстве располагаются источники освещения, а также определяется точка зрения и направление взгляда наблюдателя.

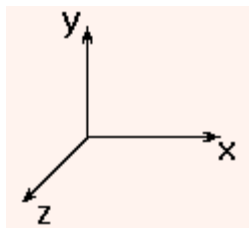
### *Правосторонняя*

### *Левосторонняя*

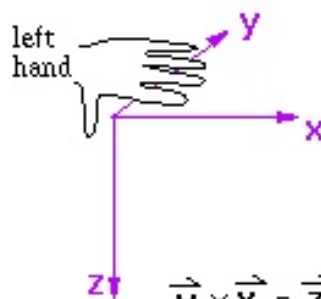
RHCS



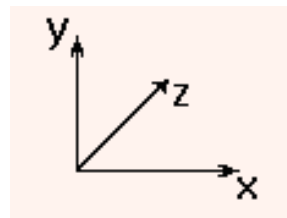
$$\vec{x} \times \vec{y} = \vec{z}$$



LHCS



$$\vec{y} \times \vec{x} = \vec{z}$$



## *Описание объекта*

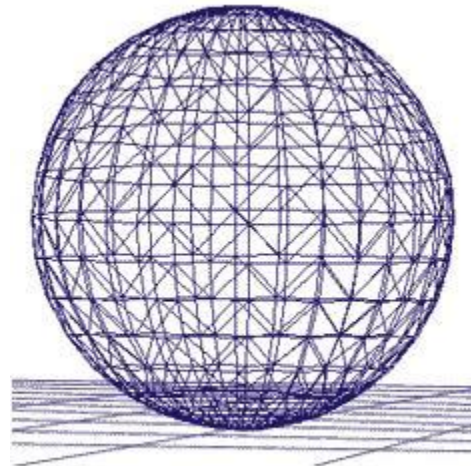
Объекты могут иметь разнообразную форму, описанную каким-либо математическим способом. Проще всего иметь дело с многогранниками, у которых каждая грань представляет собой часть плоскости, ограниченной полигоном. Описание такого тела относительно несложно - оно состоит из упорядоченного списка вершин.

Современные методы компьютерной графики основаны на представлении объекта в виде набора плоских многоугольников (Практически всегда эти многоугольники разбиваются на простейшие **треугольники**. На то есть много причин — удобство работы, ограниченные возможности оборудования, но главная — большинству алгоритмов закрашки изображения нужно, чтобы полигоны были плоскими, то есть чтобы все их вершины лежали в одной плоскости. А для треугольников это требование выполняется автоматически).



Объект задается вершинами, определяющими ключевые точки, и полигонами, которые образованы линиями, соединяющими вершины. Такая модель объекта называется ***проволочной***.

```
*MESH_VERTEX 577 21.3728 46.4102 -4.6324
*MESH_VERTEX 578 20.5746 46.4102 -7.0891
*MESH_VERTEX 579 19.4747 46.4102 -9.4265
*MESH_VERTEX 580 18.0906 46.4102 -11.6075
*MESH_VERTEX 581 16.4440 46.4102 -13.5979
*MESH_VERTEX 582 14.5609 46.4102 -15.3662
*MESH_VERTEX 583 12.4711 46.4102 -16.8845
*MESH_VERTEX 584 10.2075 46.4102 -18.1290
*MESH_VERTEX 585 7.8057 46.4102 -19.0799
*MESH_VERTEX 586 5.3037 46.4102 -19.7223
*MESH_VERTEX 587 2.7409 46.4102 -20.0461
*MESH_VERTEX 588 0.1577 46.4102 -20.0461
*MESH_VERTEX 589 -2.4051 46.4102 -19.7223
```

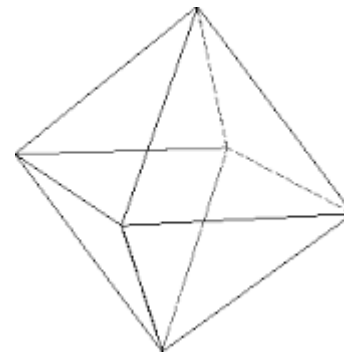


Для написания программы необходимо хранить информации об объектах в следующем виде:

- Координаты вершин хранятся в одномерных массивах X[], Y[], Z[].
- Информация о плоскостях - массив, в котором хранятся номера образующих его вершин.
- Конечный элемент - со значением не попадающим в область допустимых значений

Например в VRML октаэдр можно задать как совокупность точек:

```
Coordinate3 {  
  point [  
    0      0      1.41421,  
    1.41421 0      0,  
    0      1.41421 0,  
    -1.41421 0      0,  
    0      -1.41421 0,  
    0      0      -1.41421,  
  ]  
}
```





Затем надо указать какие точки  
составляют плоскость:

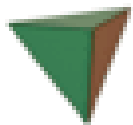
```
IndexedFaceSet {  
  coordIndex [  
    0,1,2,-1,  
    0,2,3,-1,  
    0,3,4,-1,  
    0,4,1,-1,  
    1,4,5,-1,  
    1,5,2,-1,  
    2,5,3,-1,  
    3,5,4,-1,  
  ]  
}
```

И если надо какие точки составляют ребра:

```
IndexedLineSet {  
  coordIndex [  
    0,1,-1,  
    0,2,-1,  
    0,3,-1,  
    0,4,-1,  
    1,2,-1,  
    1,4,-1,  
    1,5,-1,  
    2,3,-1,  
    2,5,-1,  
    3,4,-1,  
    3,5,-1,  
    4,5,-1,]  
}
```

В трёхмерном евклидовом пространстве существует всего пять правильных многогранников. Многогранник называется правильным, если:

- он выпуклый;
- все его грани являются равными правильными многоугольниками;
- в каждой его вершине сходится одинаковое число рёбер.



### **Тетраэдр**

Число вершин – 4

Число рёбер – 6

Число граней 4



### **Гексаэдр или куб**

Число вершин – 8

Число рёбер – 12

Число граней 6



### **Октаэдр**

Число вершин – 6

Число рёбер – 12

Число граней 8



### **Додекаэдр**

Число вершин – 20

Число рёбер – 30

Число граней 12



### **Икосаэдр**

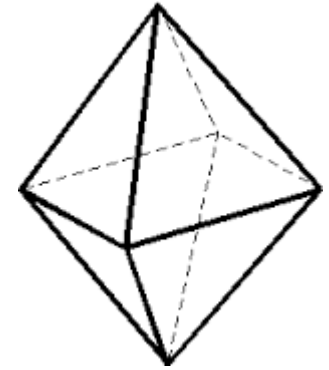
Число вершин – 12

Число рёбер – 30

Число граней 20

## Октаэдр

Число вершин – 6 Число ребер – 12 Число граней 8



```
Coordinate3 {  
  point [  
    0          0          1.41421,  
    1.41421    0          0,  
    0          1.41421    0,  
    -1.41421   0          0,  
    0          -1.41421   0,  
    0          0          -1.41421,  
  ]  
}
```

```
IndexedFaceSet {  
  coordIndex [  
    0,1,2,-1,  
    0,2,3,-1,  
    0,3,4,-1,  
    0,4,1,-1,  
    1,4,5,-1,  
    1,5,2,-1,  
    2,5,3,-1,  
    3,5,4,-1,  
  ]  
}
```

```
IndexedLineSet {  
  coordIndex [  
    0,1,-1,  
    0,2,-1,  
    0,3,-1,  
    0,4,-1,  
    1,2,-1,  
    1,4,-1,  
    1,5,-1,  
    2,3,-1,  
    2,5,-1,  
    3,4,-1,  
    3,5,-1,  
    4,5,-1,  
  ]  
}
```

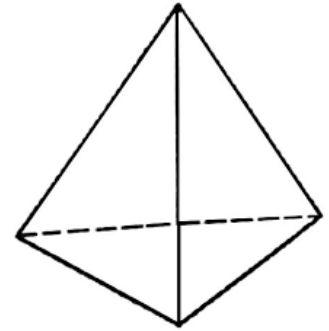
## *Тетраэдр*

Число вершин – 4    Число ребер – 6    Число граней 4

```
Coordinate3 {  
  point [  
    1 1 1,  
    1 -1 -1,  
    -1 1 -1,  
    -1 -1 1,  
  ]  
}
```

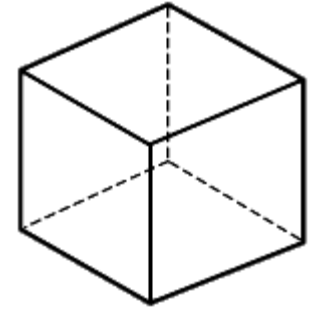
```
IndexedFaceSet {  
  coordIndex [  
    0,1,2,-1,  
    0,2,3,-1,  
    0,3,1,-1,  
    1,3,2,-1,  
  ]  
}
```

```
IndexedLineSet {  
  coordIndex [  
    0,1,-1,  
    0,2,-1,  
    0,3,-1,  
    1,2,-1,  
    1,3,-1,  
    2,3,-1,  
  ]  
}
```



## *Гексаэдр или куб*

Число вершин – 8 Число ребер – 12 Число граней 6



```
Coordinate3 {  
  point [  
    0 0 1.22474,  
    1.1547 0 0.408248,  
    -0.57735 1. 0.408248,  
    -0.57735 -1. 0.408248,  
    0.57735 1. -0.408248,  
    0.57735 -1. -0.408248,  
    -1.1547 0 -0.408248,  
    0 0 -1.22474,  
  ]  
}
```

```
IndexedFaceSet {  
  coordIndex [  
    0,1,4,2,-1,  
    0,2,6,3,-1,  
    0,3,5,1,-1,  
    1,5,7,4,-1,  
    2,4,7,6,-1,  
    3,6,7,5,-1,  
  ]  
}
```

```
IndexedLineSet {  
  coordIndex [  
    0,1,-1,  
    0,2,-1,  
    0,3,-1,  
    1,4,-1,  
    1,5,-1,  
    2,4,-1,  
    2,6,-1,  
    3,5,-1,  
    3,6,-1,  
    4,7,-1,  
    5,7,-1,  
    6,7,-1,  
  ]  
}
```

# Додекаэдр

Число вершин – 20 Число ребер – 30 Число граней 12

Coordinate3 {

point [

0 0 1.07047,

0.713644 0 0.797878,

-0.356822 0.618034 0.797878,

-0.356822 -0.618034 0.797878,

0.797878 0.618034 0.356822,

0.797878 -0.618034 0.356822,

-0.934172 0.381966 0.356822,

0.136294 1. 0.356822,

0.136294 -1. 0.356822,

-0.934172 -0.381966 0.356822,

0.934172 0.381966 -0.356822,

0.934172 -0.381966 -0.356822,

-0.797878 0.618034 -0.356822,

-0.136294 1. -0.356822,

-0.136294 -1. -0.356822,

-0.797878 -0.618034 -0.356822,

0.356822 0.618034 -0.797878,

0.356822 -0.618034 -0.797878,

-0.713644 0 -0.797878,

0 0 -1.07047,

]

IndexedFaceSet {

coordIndex [

0,1,4,7,2,-1,

0,2,6,9,3,-1,

0,3,8,5,1,-1,

1,5,11,10,4,-1,

2,7,13,12,6,-1,

3,9,15,14,8,-1,

4,10,16,13,7,-1,

5,8,14,17,11,-1,

6,12,18,15,9,-1,

10,11,17,19,16,-1,

12,13,16,19,18,-1,

14,15,18,19,17,-1,

]

}

IndexedLineSet {

coordIndex [

0,1,-1,

0,2,-1,

0,3,-1,

1,4,-1,

1,5,-1,

2,6,-1,

2,7,-1,

3,8,-1,

3,9,-1,

4,7,-1,

4,10,-1,

5,8,-1,

5,11,-1,

6,9,-1,

6,12,-1,

7,13,-1,

8,14,-1,

9,15,-1,

10,11,-1,

10,16,-1,

11,17,-1,

12,13,-1,

12,18,-1,

13,16,-1,

14,15,-1,

14,17,-1,

15,18,-1,

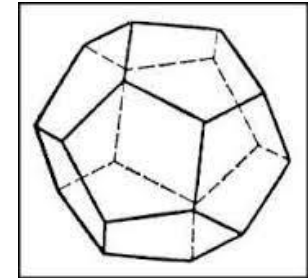
16,19,-1,

17,19,-1,

18,19,-1,

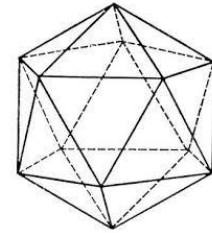
]

}



# Икосаэдр

Число вершин – 12 Число ребер – 30 Число граней 20



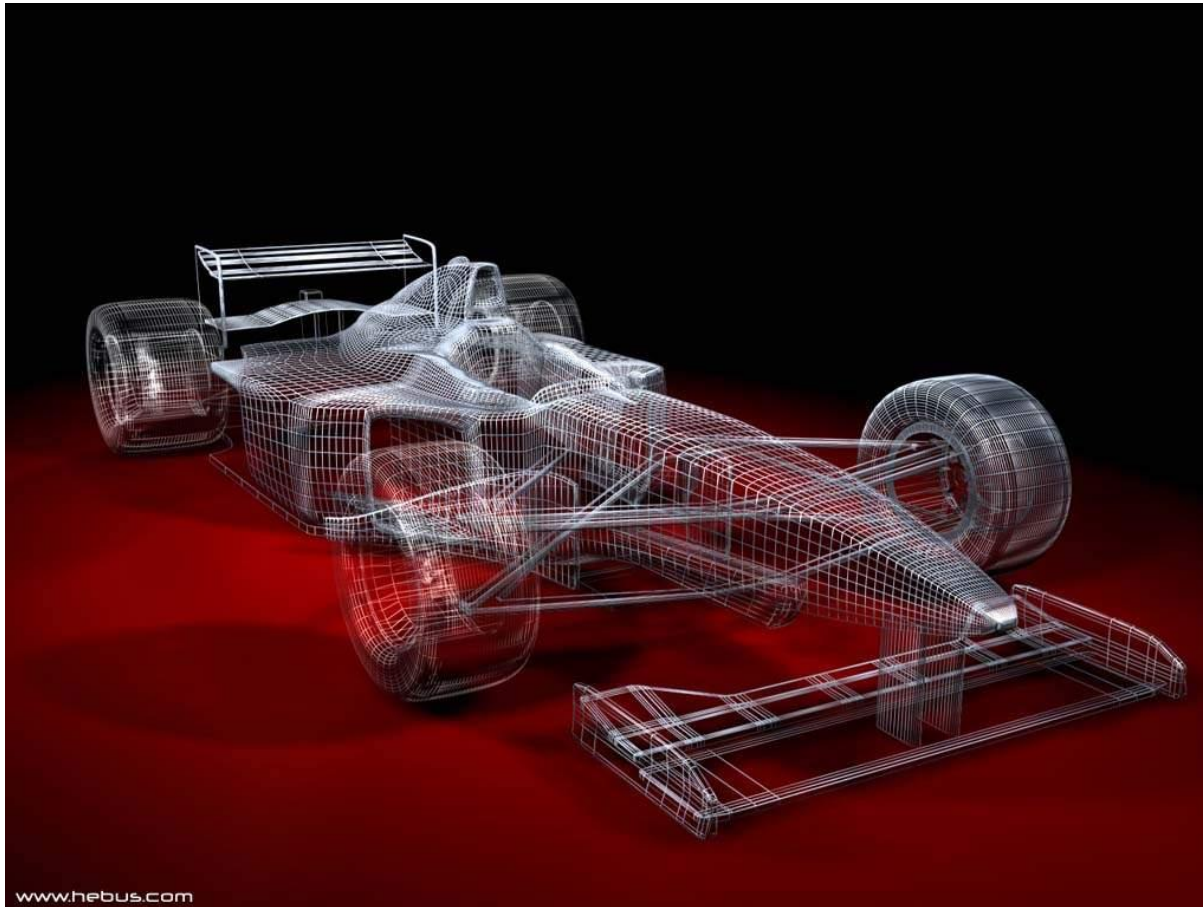
```
Coordinate3 {
  point [
    0 0 1.17557,
    1.05146 0 0.525731,
    0.32492 1. 0.525731,
    -0.850651 0.618034 0.525731,
    -0.850651 -0.618034 0.525731,
    0.32492 -1. 0.525731,
    0.850651 0.618034 -0.525731,
    0.850651 -0.618034 -0.525731,
    -0.32492 1. -0.525731,
    -1.05146 0 -0.525731,
    -0.32492 -1. -0.525731,
    0 0 -1.17557,
  ]
}

IndexedFaceSet {
  coordIndex [
    0,1,2,-1,
    0,2,3,-1,
    0,3,4,-1,
    0,4,5,-1,
    0,5,1,-1,
    1,5,7,-1,
    1,7,6,-1,
    1,6,2,-1,
    2,6,8,-1,
    2,8,3,-1,
    3,8,9,-1,
    3,9,4,-1,
    4,9,10,-1,
    4,10,5,-1,
    5,10,7,-1,
    6,7,11,-1,
    6,11,8,-1,
    7,10,11,-1,
    8,11,9,-1,
    9,11,10,-1,
  ]
}

IndexedLineSet {
  coordIndex [
    0,1,-1,
    0,2,-1,
    0,3,-1,
    0,4,-1,
    0,5,-1,
    1,2,-1,
    1,5,-1,
    1,6,-1,
    1,7,-1,
    2,3,-1,
    2,6,-1,
    2,8,-1,
    3,4,-1,
    3,8,-1,
    3,9,-1,
    4,5,-1,
    4,9,-1,
    4,10,-1,
    5,7,-1,
    5,10,-1,
    6,7,-1,
    6,8,-1,
    6,11,-1,
    7,10,-1,
    7,11,-1,
    8,9,-1,
    8,11,-1,
    9,10,-1,
    9,11,-1,
    10,11,-1,
  ]
}
```

Таким же образом можно задавать и более сложные объекты.

[ris//vessel.txt](#)





Правильные многогранники характерны для философии Платона, в честь которого и получили название «*ПЛАТОНОВЫ ТЕЛА*»

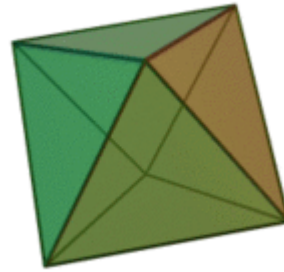
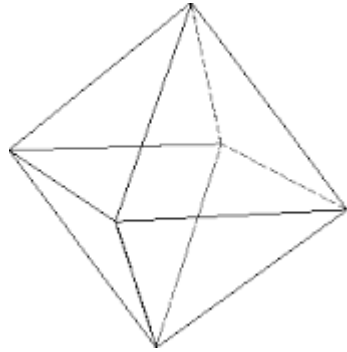
Евклид дал полное математическое описание правильных многогранников в последней, XIII книге Начал.

В размерности  $n = 4$

Существует 6 правильных четырёхмерных многогранников:

- 4-мерный симплекс (грань — тетраэдр, символ Шлефли  $\{3, 3, 3\}$ ).
- Тессеракт или 4-мерный куб (грань — куб,  $\{4, 3, 3\}$ ).
- 16-гранник (грань — тетраэдр,  $\{3, 3, 4\}$ ).
- 24-гранник (грань — октаэдр,  $\{3, 4, 3\}$ ).
- 120-гранник (грань — додекаэдр,  $\{5, 3, 3\}$ ).
- 600-гранник (грань — тетраэдр,  $\{3, 3, 5\}$ ).





К сожалению, картинка из одних лишь вершин (и даже вершин, соединенных ребрами) удовлетворит далеко не всех пользователей. Поэтому на стадии рендеринга производится удаление невидимых линий.