

---

---

# Краткая история трехмерной графики

— Михайлов Игорь —

---

---

# Пояснения/Уточнения

1. В качестве платформы рассматриваются в основном персональные компьютеры.
2. Моментом появления технологии считается появление ее аппаратной поддержки на распространённых моделях видеокарт.
3. Примеры графики показаны на примере игр, так как они наиболее полно используют возможности видеокарты.
4. Доклад скорее научно-популярный и в нём могут быть ошибки, не стесняйтесь поправлять.

# Общая хронология

- с 1981 Графические адаптеры
- с 1990 Графические ускорители
- с 2002 Шейдеры
- с 2007 Расчёты общего назначения на видеокартах
- с 2018 Аппаратная поддержка трассировки лучей

# Графические адаптеры

Графический адаптер - устройство, преобразующее графический образ, хранящийся как содержимое памяти компьютера (или самого адаптера), в форму, пригодную для дальнейшего вывода на экран монитора. Первые мониторы, построенные на электронно-лучевых трубках, работали по телевизионному принципу сканирования экрана электронным лучом, и для отображения требовался видеосигнал, генерируемый видеокартой.

# Графические адаптеры

- Monochrome Display Adapter - 80×25 символов
- Color Graphics Adapter
  - 320×200 точек 4 цвета
  - 640×200 точек монохромный
- Enhanced Graphics Adapter - 640×350 16 цветов
- Multicolor Graphics Adapter - 320×200 точек 256 цветов
- Video Graphics Array
  - 640×480×256
  - 1024×768×256
- Super VGA - 65.536 цветов и 16.777.216 цветов

# Примеры рендера без ускорителей



Wolfenstein 3D  
engine



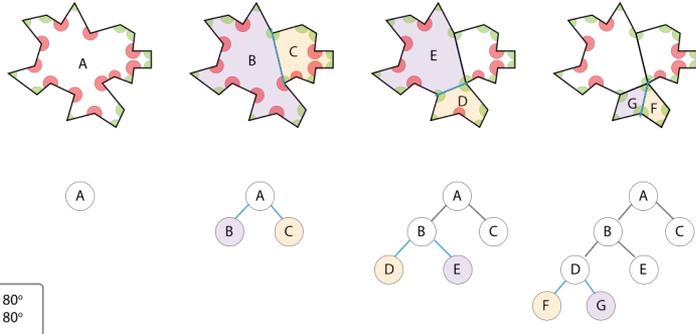
Doom  
id Tech 1

Duke Nukem 3d  
Build Engine



# Технологии

BSP-дерево строится методом рекурсивного разбиения евклидова пространства в выпуклые множества и гиперплоскости.



- Враги и предметы отображались с помощью спрайтов
- Имитация освещения - сменой цвета сектора
- Порталы - позволяющие делать двухэтажные уровни

# Графические ускорители

1984 Professional Graphics Controller

Первый 2d и 3d графический ускоритель для CAD-систем. Имел высокую стоимость ~4000\$

~1992 Matrox Millennium

2d и 3d графический ускоритель для CAD-систем. Поддерживал простую модель затенения без текстур.

# Графические ускорители

## Voodoo Graphics PCI 1996

- процессор кадрового буфера
- процессор наложения текстур
- а также 4 мегабайта (МБ) ОЗУ
- частота 50 МГц
- Требовалась графический адаптер для 2d вывода
- Спецификация Glide (минифицированный аналог OpenGL и Direct3D, с низким уровнем абстракции)

# Графические ускорители

## Voodoo Graphics PCI 1996

- процессор кадрового буфера
- процессор наложения текстур
- 4 мегабайта (МБ) ОЗУ
- частота 50 МГц
- Требовалась графический адаптер для 2d вывода
- Спецификация Glide (минифицированный аналог OpenGL и Direct3D, с низким уровнем абстракции)

# Графические ускорители

RIVA TNT 1998

- mipmaping
- фильтрация текстур
- а также 16 мегабайт (МБ) ОЗУ
- частота 90 МГц
- Требовалась графический адаптер для 2d вывода
- Спецификация Direct3D, частичная OpenGL

# Примеры рендера на первых ускорителях



**Quake II**  
**id Tech 2**



**Half life**  
**GoldSrc**

# Первые ускорители плюсы и минусы

## Плюсы

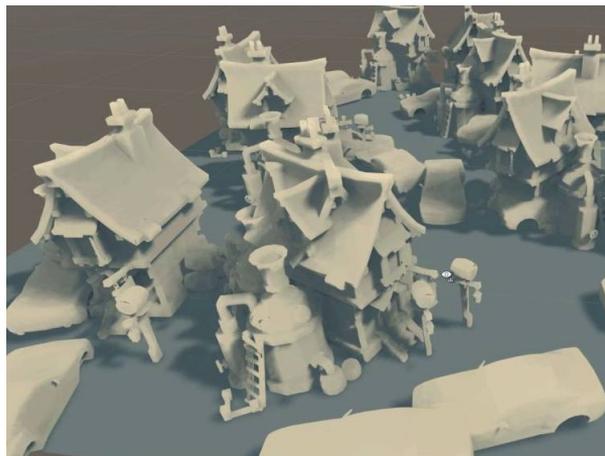
- Высокая производительность
- Разгрузка CPU
- Поддержка простых эффектов из “коробки” (туман, освещение и так далее)

## Минусы

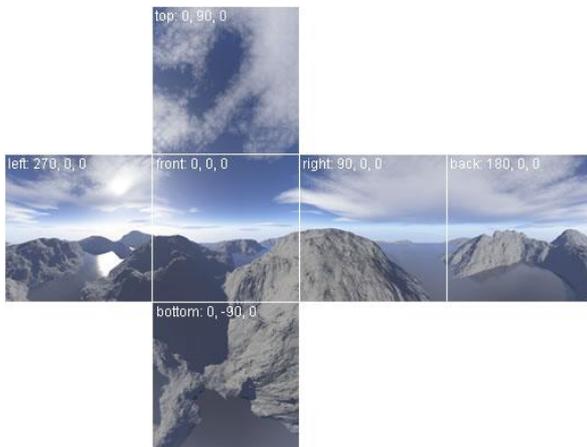
- “Зоопарк” стандартов
- Различные возможности на разных платформах
  - Необходимо проверять возможности видеокарты
  - Специальные возможности
  - Неполное соответствие спецификации
- Если чего то нет, то это нельзя сделать

# Технологии

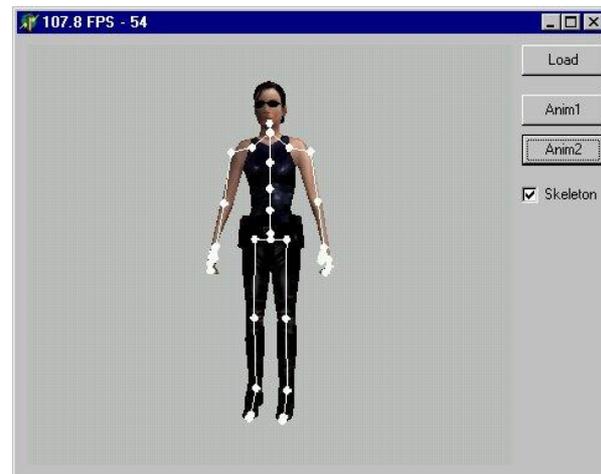
Карты освещения



Sky-box



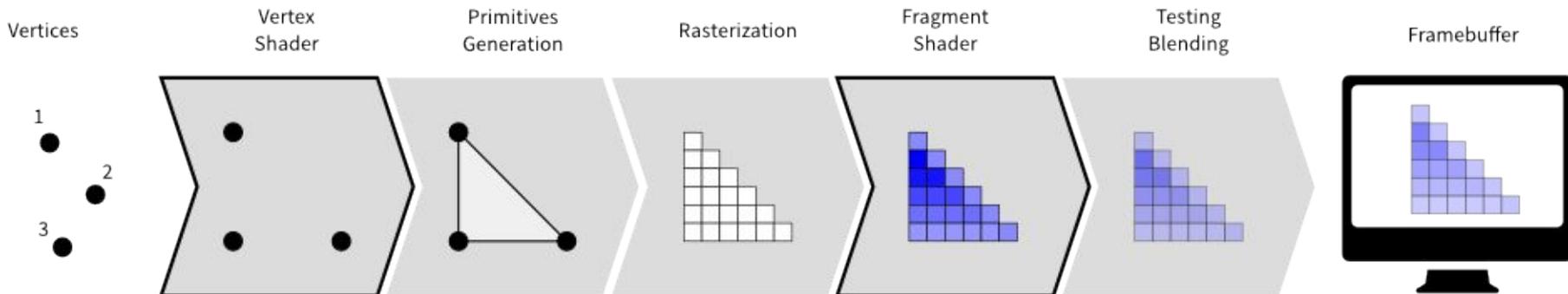
Скелетная анимация



# Шейдеры

Программа выполняемая на видеокарте, существует несколько видов шейдеров.

- Вершинный
- Тесселяции //появился позднее
- Геометрический //появился позднее
- Фрагментный



# Вершинный шейдер

Вершинный шейдер оперирует данными, связанными с вершиной многогранника, например, с координатами вершины в пространстве, с текстурными координатами, с цветом вершины, с вектором касательной, с вектором бинормали, с вектором нормали.

```
#version 330 core
```

```
layout(location = 0) in vec3 vertexPosition_modelspace;
```

```
void main(){
```

```
    gl_Position.xyz = vertexPosition_modelspace;
```

```
    gl_Position.w = 1.0;
```

```
}
```

# Фрагментный шейдер

Пиксельный шейдер работает с фрагментами растрового изображения и с текстурами — обрабатывает данные, связанные с пикселями (например, цвет, глубина, текстурные координаты). Пиксельный шейдер используется на последней стадии графического конвейера для формирования фрагмента изображения.

```
#version 330 core
out vec3 color;
void main(){
    color = vec3(1,0,0);
}
```

# Примеры рендера с использованием шейдеров

**Doom 3**  
**id Tech 4**



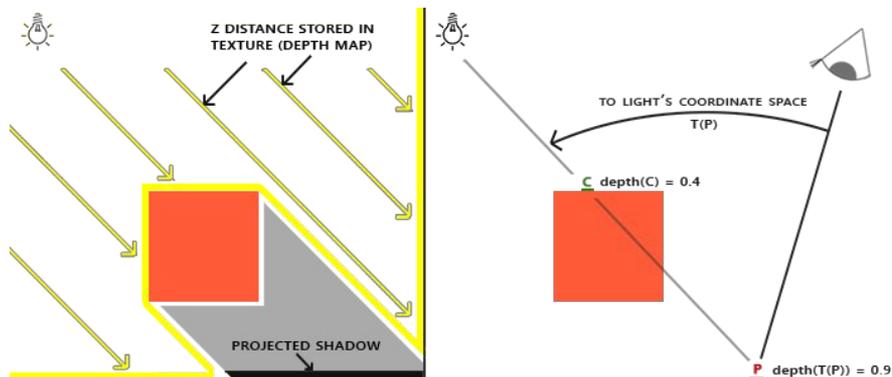
**Witcher 3**  
**REDengine**



**Portal**  
**Source**



# Технологии



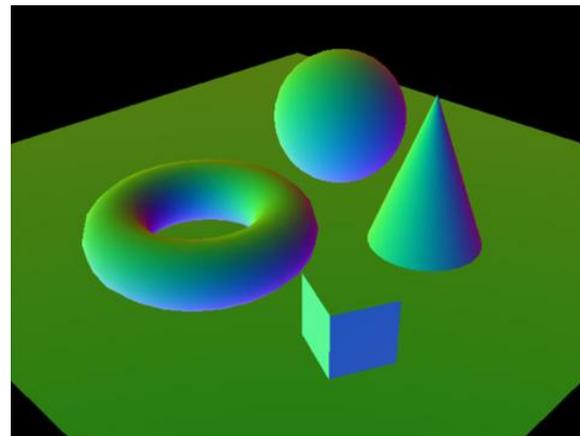
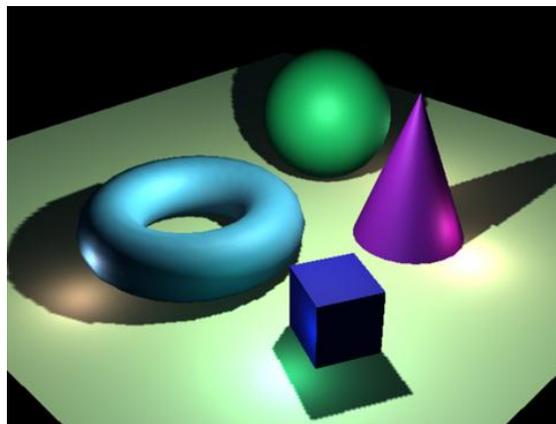
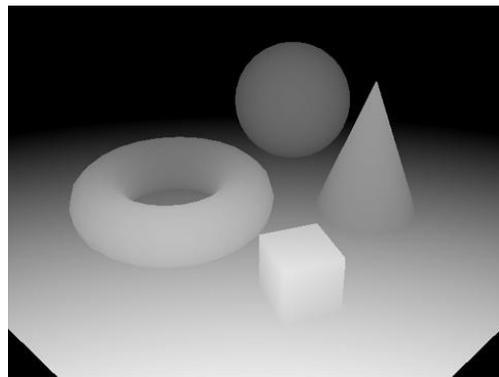
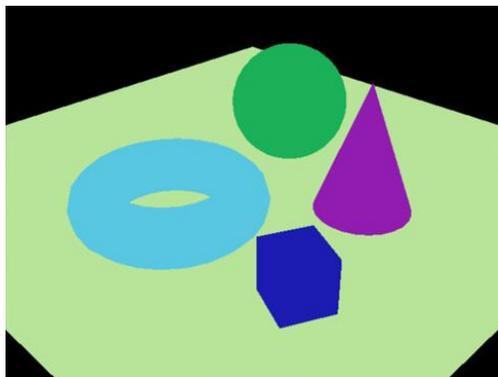
High Dynamic Range

Карты теней



# Технологии

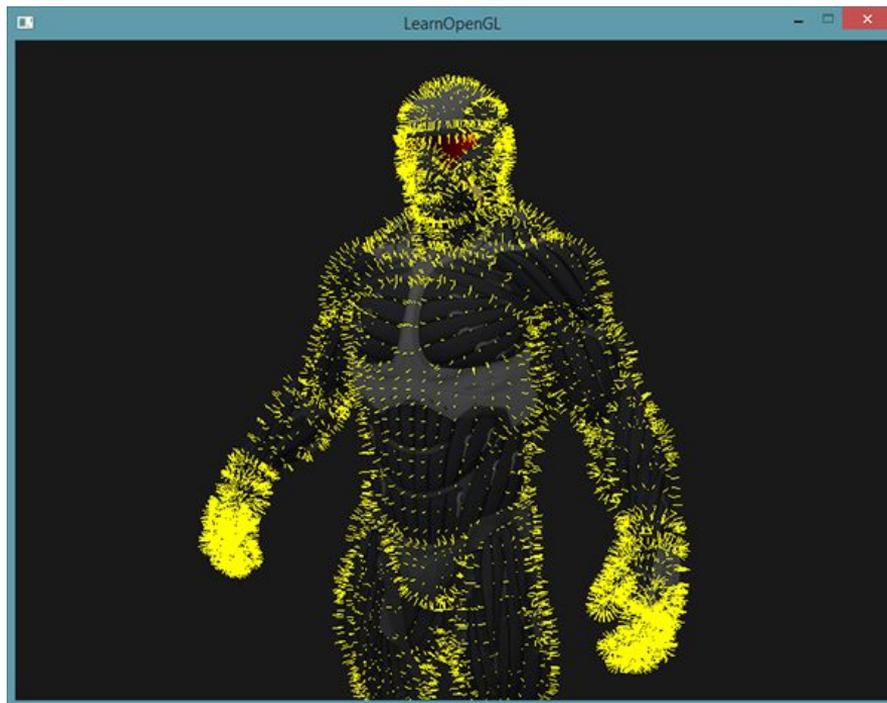
Deferred rendering



# Геометрический шейдер

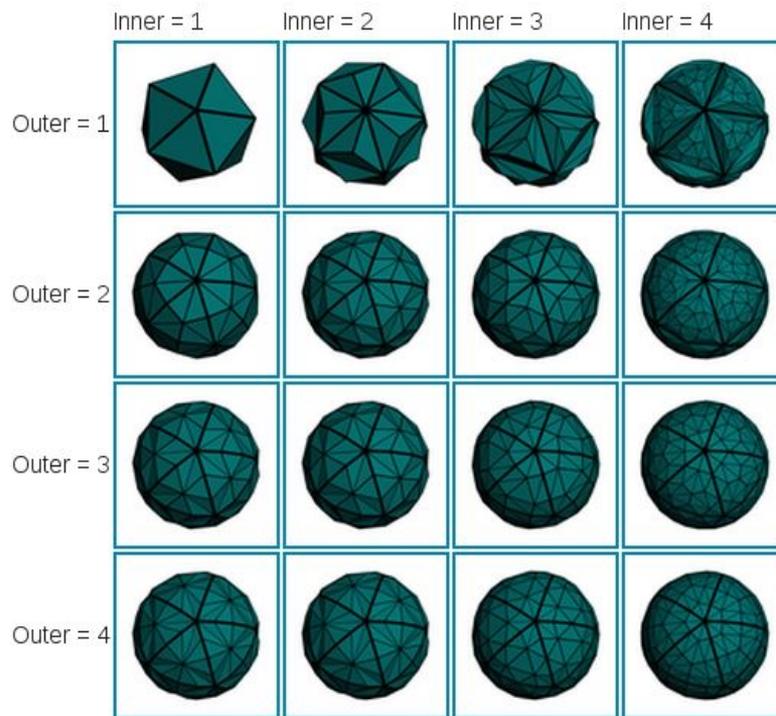
Геометрический шейдер, в отличие от вершинного, способен обработать не только одну вершину, но и целый примитив.

Геометрический шейдер способен генерировать примитивы на лету.



# Тесселяционный шейдер

Тесселяция -  
автоматизированный процесс  
добавления новых выпуклых  
многоугольников в  
полигональную сетку с целью  
повышения детализации сетки.



# Трассировка лучей

Исторический в 3д графике проецирование треугольников на экран использовалось чаще трассировки лучей, из за чего требовались хитрые техники расчета освещения и отражений.

Плюсы от честной трассировки лучей - более естественный рендер сцены, большие возможности визуализации.

