

# Введение в Шейдеры

Компьютерная графика

# Шейдер. Перевод с английского

сущ    оттенок м, тень ж, тон м, нюанс м, полумрак м  
different shades – разные оттенки  
cool shade – прохладная тень  
red shade – красный тон  
delicate shade – тонкий нюанс  
pleasant shade – приятный полумрак

гл      затенять

# Шейдер. Что это?

shader — схема затемнения, программа построения теней

# Определение из Wiki

Программа для одной из ступеней графического конвейера, используемая в трёхмерной графике для определения окончательных параметров объекта или изображения

# Впервые — RenderMan

Впервые программируемые шейдеры были представлены в RenderMan компании Pixar

RenderMan создан компанией Pixar и используется с 1986 года

Существуют коннекторы к большинству программ 3d-моделирования: Autodesk Maya, Cinema 4D.

Начиная с 23 марта 2015 года студия Pixar начала бесплатное распространение полнофункциональной версии RenderMan для некоммерческого использования

# RenderMan

Программный продукт, пакет программ, промышленный стандарт рендеринга для 3D-анимации.

Тип	система рендеринга
Разработчик	Pixar
Операционная система	Microsoft Windows, Mac OS X, Linux
Последняя версия	23.4 / июль, 2020
Лицензия	Проприетарное программное обеспечение
Сайт	<a href="http://renderman.pixar.com">renderman.pixar.com</a>

# RenderMan использован в фильмах

В поисках Дори (2016)  
Аватар (2009)  
2012 (2009)  
ВАЛЛ-И (2008)  
Гарри Поттер и Орден Феникса (2007)  
Пираты Карибского моря: На краю Света (2007)  
Рататуй (2007)  
Лови волну! (2007)  
Трансформеры (2007)  
Король лев (1994)  
Маска (1994)  
Парк юрского периода (1993)  
Аладдин (1992)  
Терминатор 2: Судный день (1991)  
Университет монстров (2013)

# Только 2022

Chip 'N' Dale: Rescue Rangers

Doctor Strange in the Multiverse of Madness

Elvis

Fantastic Beasts: The Secrets of Dumbledore

Jurassic World: Dominion

Lightyear

Nope

Pinocchio

Prey

Prey

Sonic the Hedgehog 2

Texas Chainsaw Masacre

The Batman

The Bubble

Thor: Love and Thunder

Top Gun: Maverick

Turning Red

Uncharted

<https://renderman.pixar.com/movies>



# Появление видеоакселераторов

Ограничение тем набором эффектов, который заложен в аппаратное обеспечение

Вода в Quake 2 на программном рендеринге — эффект плеска воды

Вода на OpenGL-рендеринге — просто синий светофильтр

В Counter-Strike 1.6 эффект ослепления от светового гранаты на программном рендеринге — белая вспышка и пикселизированный экран

В Counter-Strike 1.6 эффект ослепления от светового гранаты на аппаратном рендеринге — белая вспышка

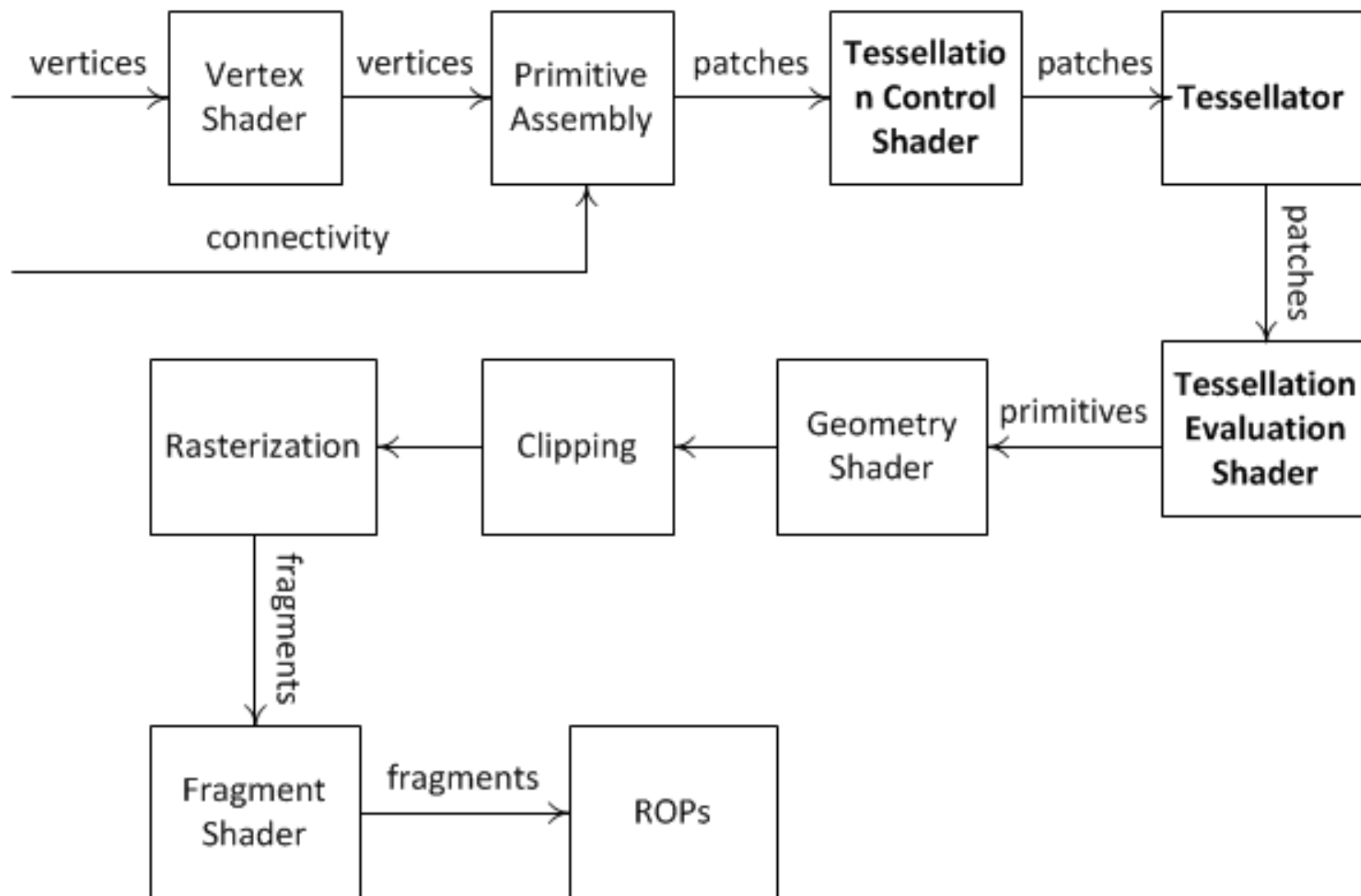
# Пути преодоления ограничения

- Расширять набор стандартных эффектов
- Реализовывать сложные эффекты из простых операций на уровне видеокарты

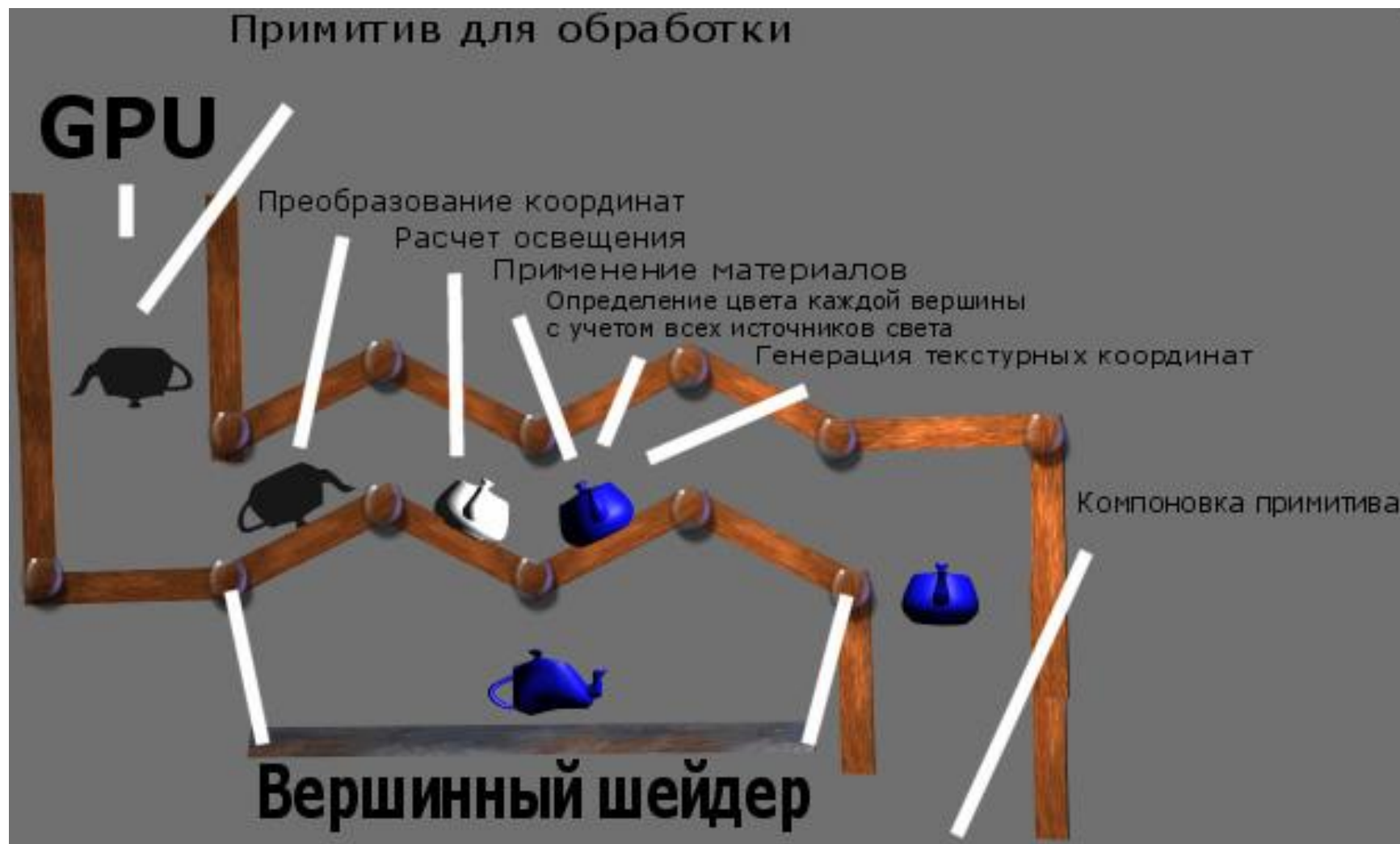
# Типы шейдеров

- Вершинные шейдеры (Vertex Shader)
- Тесселяционные шейдеры (Tessellate Shader)
- Геометрические шейдеры (Geometry Shader)
- Пиксельные шейдеры (Pixel Shader)

# Схема шейдеров

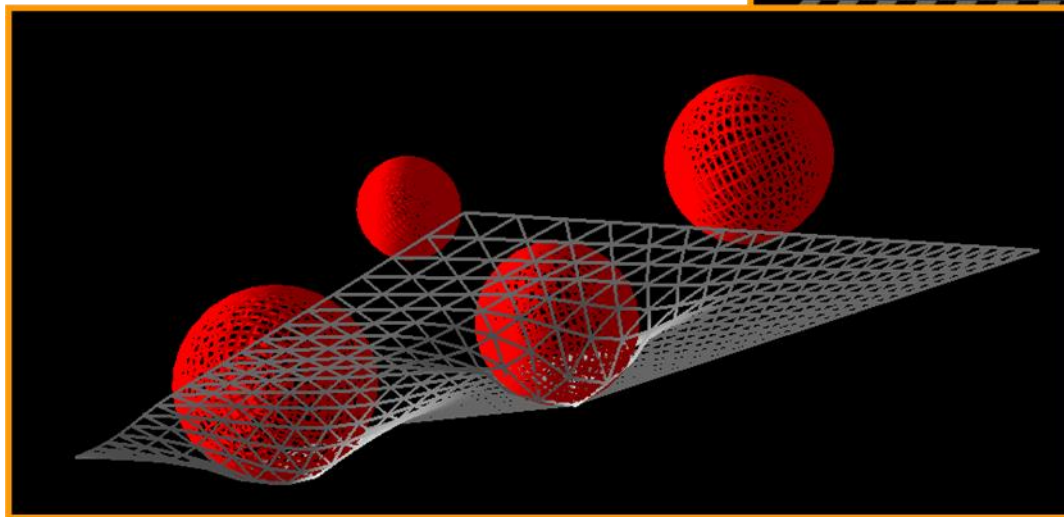
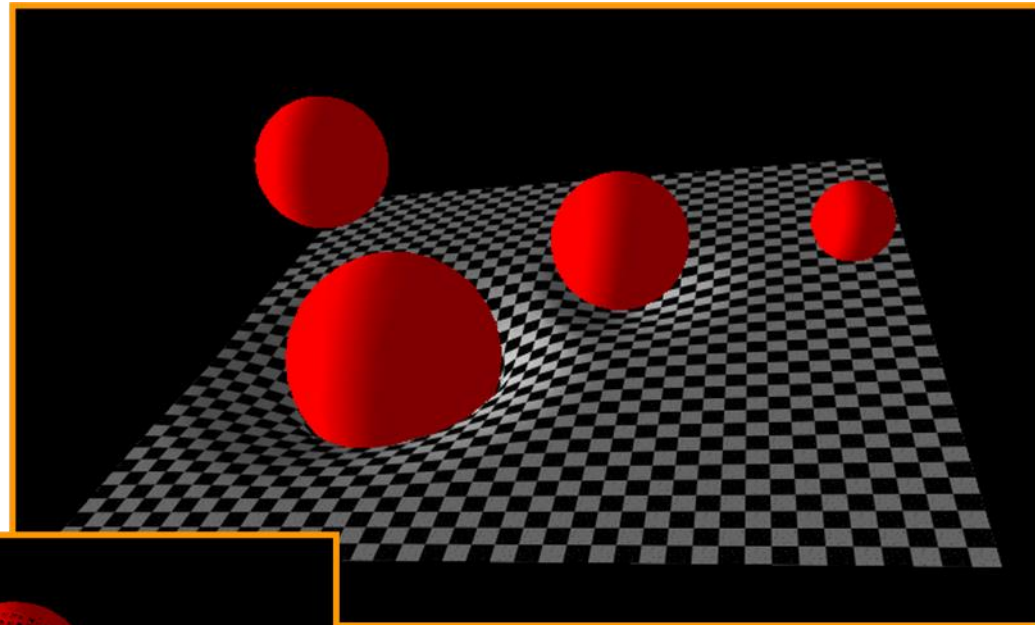


# Функции вершинного шейдера

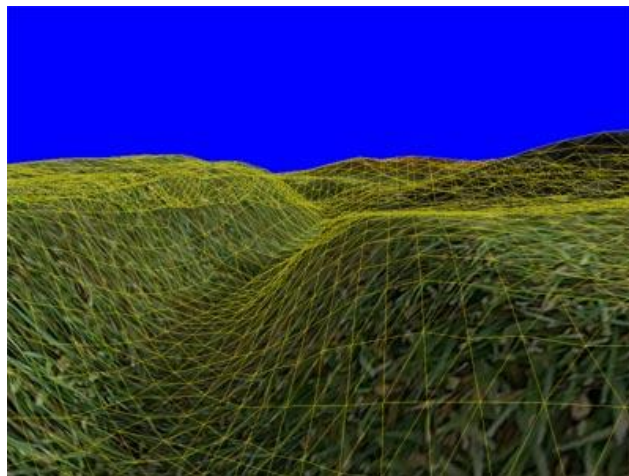


# Примеры использования вершинных шейдеров

Морфинг



# Вершинные шейдеры. Основное



Координаты вершины в пространстве, текстурные координаты, вектор касательной, вектор бинормали, вектор нормали

Для видового и перспективного преобразования вершин, генерации текстурных координат, расчета освещения и т. д.

Каждая вершина обрабатывается, не имея никакой информации об остальных вершинах

Легко распараллеливать обработку вершин

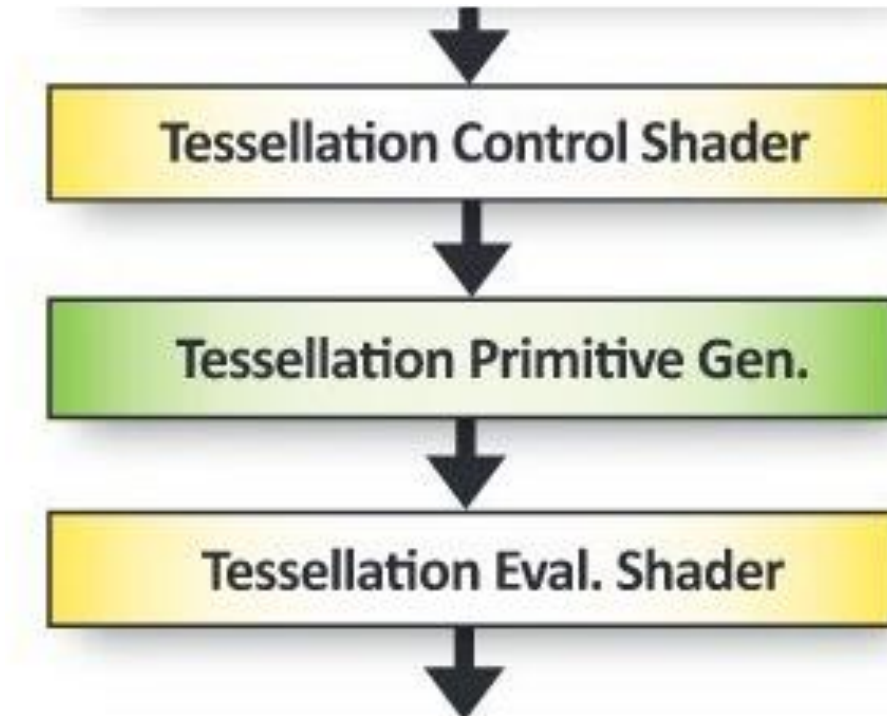
# Ограничения вершинных шейдеров

Не может создавать новые вершины и удалять существующие

Не обладает топологической информацией (границы и ребра, соседние вершины)



# Тесселяция примитивов



Тесселяционные шейдеры — это новейшие 3D-шейдеры; они воздействуют на партии вершин одновременно, чтобы добавить детали, например, разбить модель на более мелкие группы треугольников или других примитивов во время выполнения, улучшить такие вещи, как кривые и выпуклости, или изменить другие атрибуты.

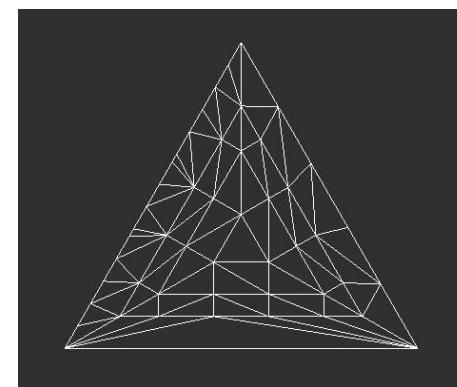
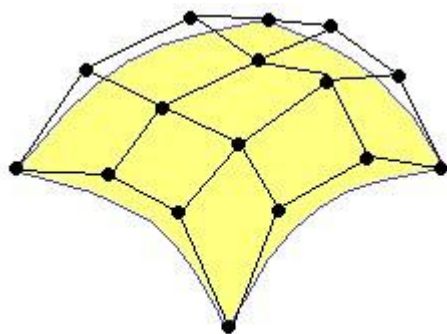
# Tessellation Control Shader

Задача данного шейдера — задать параметры для тесселятора и tessellation evaluation shader'a

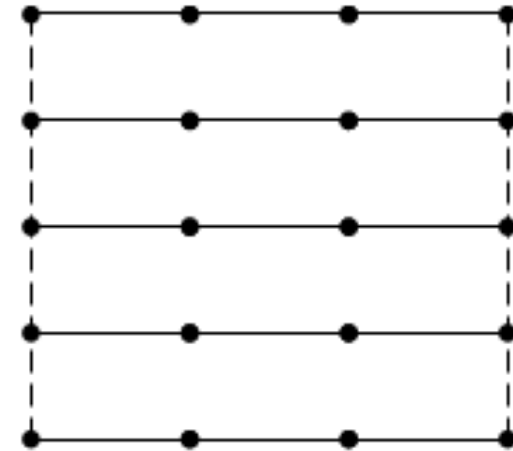
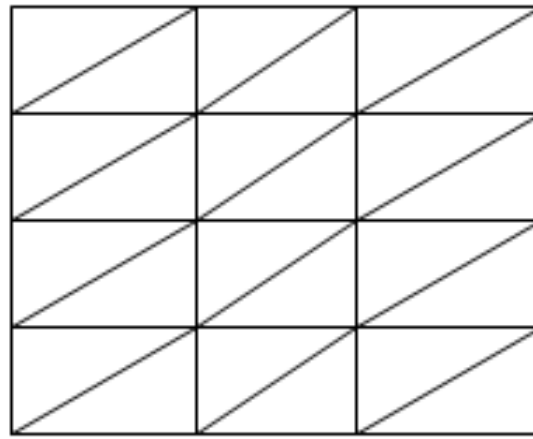
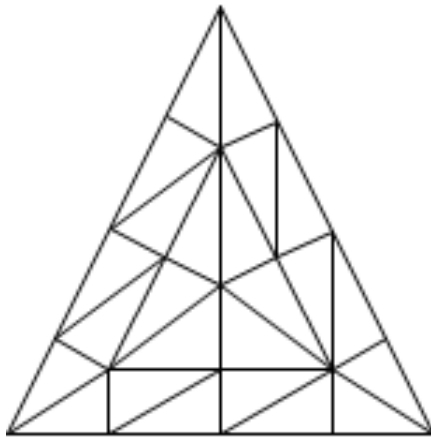
Позволяет задавать параметры для тесселятора

Вызывается для каждой вершины патча

Имеет доступ ко всем его вершинам



# Тесселяция примитивов

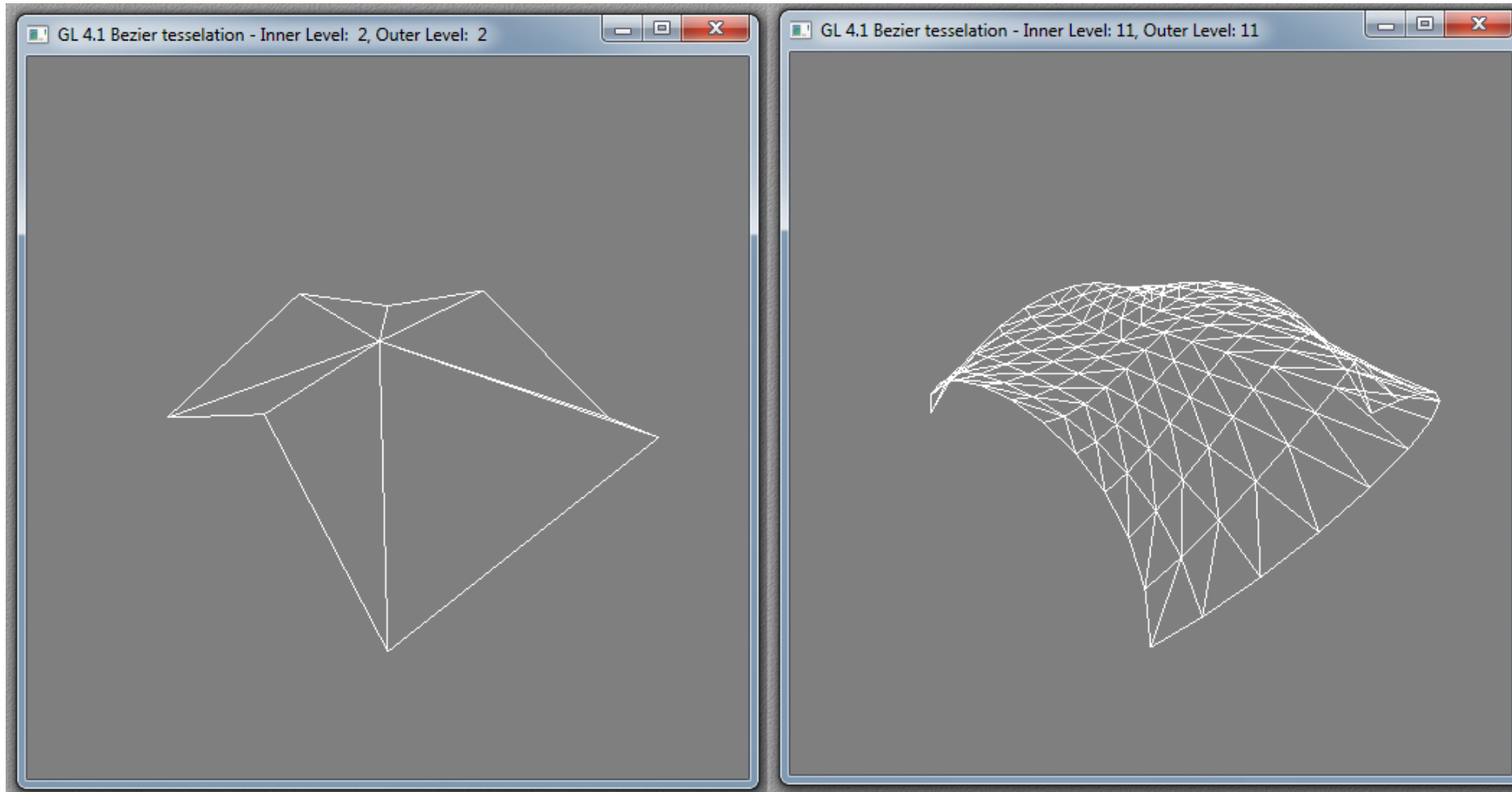


Три типа тесселяции примитивов - triangles, quads и isolines

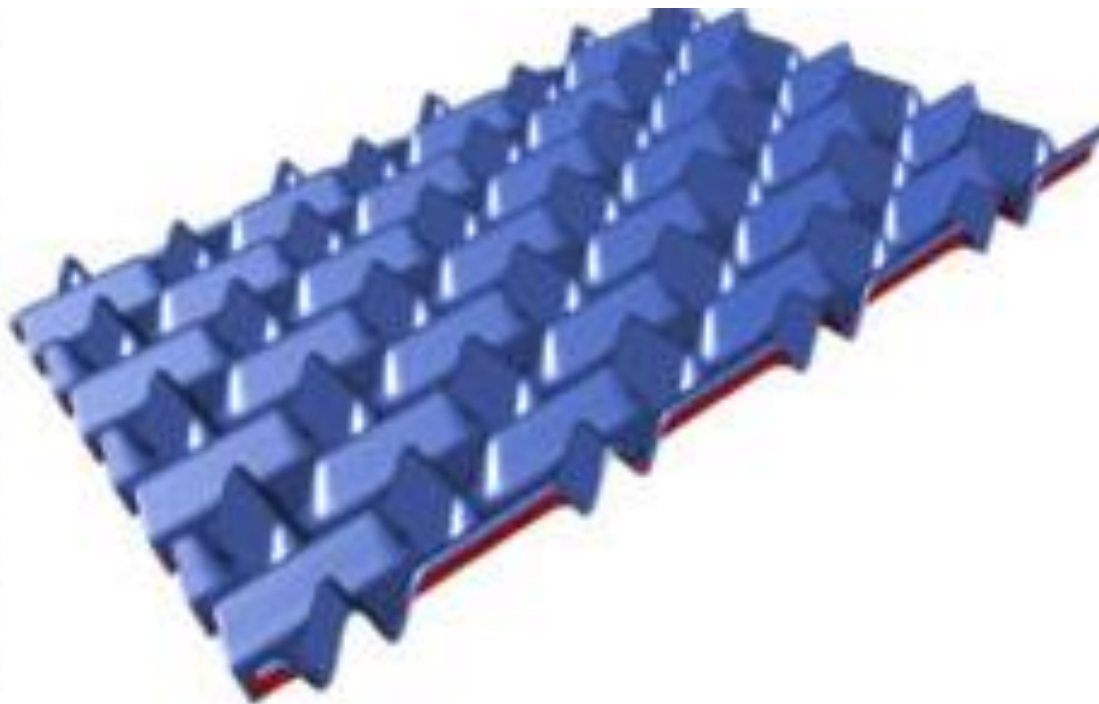
# Tessellation Evaluation Shader — Согласование уровней тесселяции для набора патчей

4	4	4	4	5	5	6
4	4	4	4	5	5	6
5	4	5	5	5	6	6
5	5	5	5	6	6	7
5	5	5	5	6	6	6

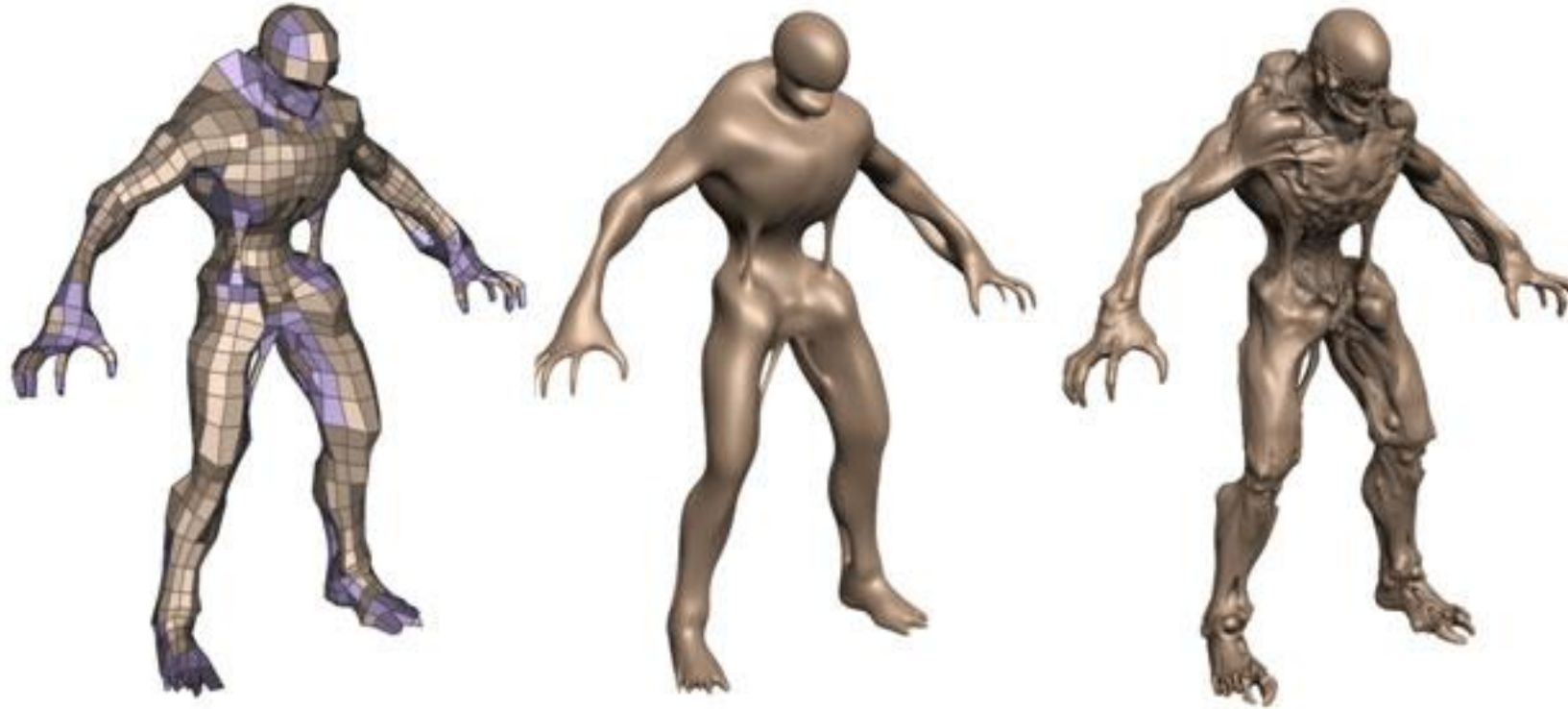
# Поверхности Безье с разными уровнями тесселяции



Применение карты смещения (слева) к плоской поверхности



# Применение тесселяции



Применение тесселяции к грубой модели (слева) позволяет создавать более гладкую модель (посередине). Использование карт смещения (справа) обеспечивает персонажам реалистичность кинематографического уровня.

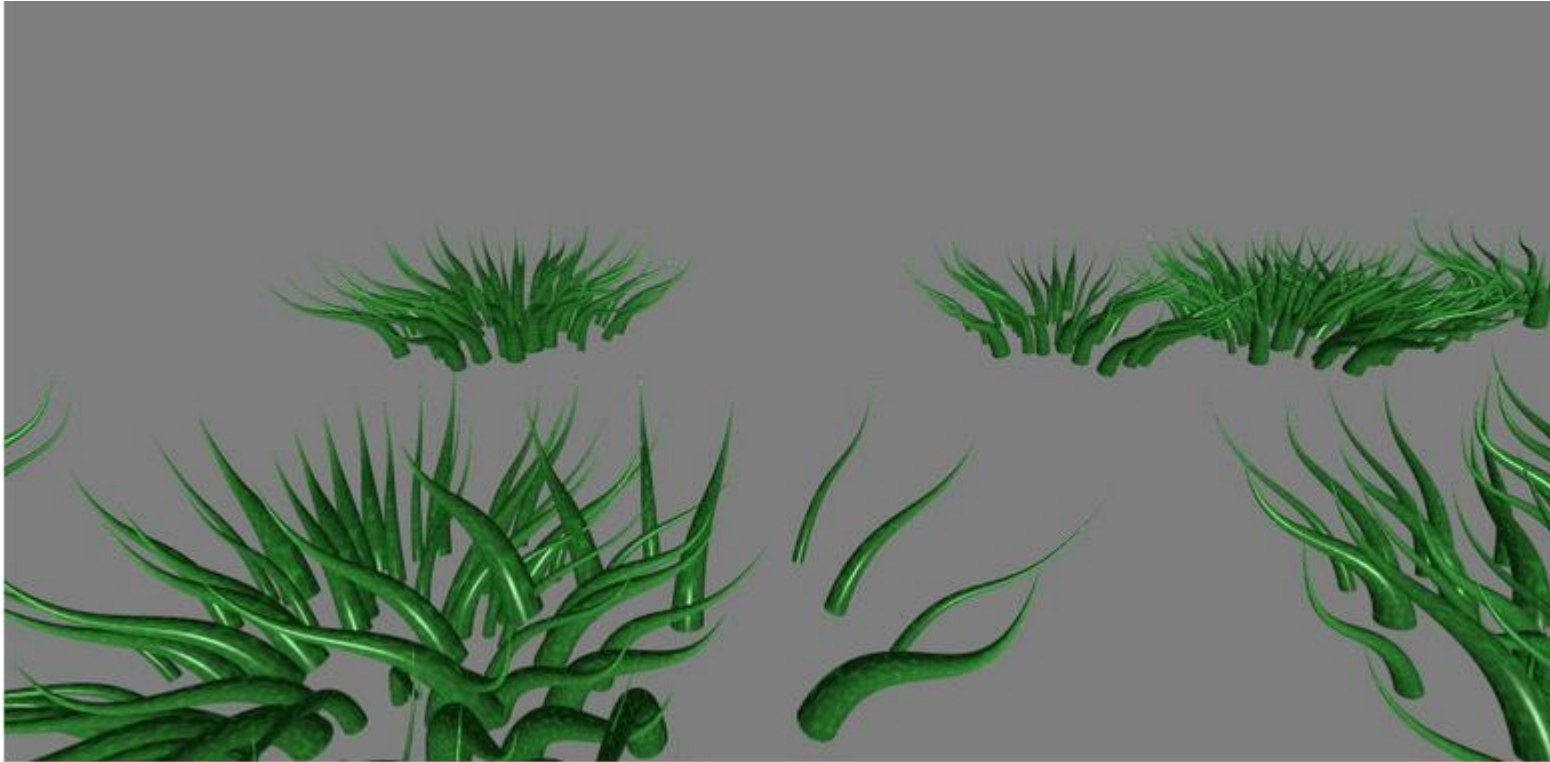
© Kenneth Scott, id Software 2008

# Геометрические шейдеры

- На вход целый примитив. Это может быть отрезок (две вершины) и треугольник (три вершины), а при наличии информации о смежных вершинах (adjacency) может быть обработано до шести вершин для треугольного примитива.
- Кроме того, геометрический шейдер способен генерировать примитивы «на лету», не задействуя при этом центральный процессор
- Впервые начал использоваться на видеокартах Nvidia серии 8

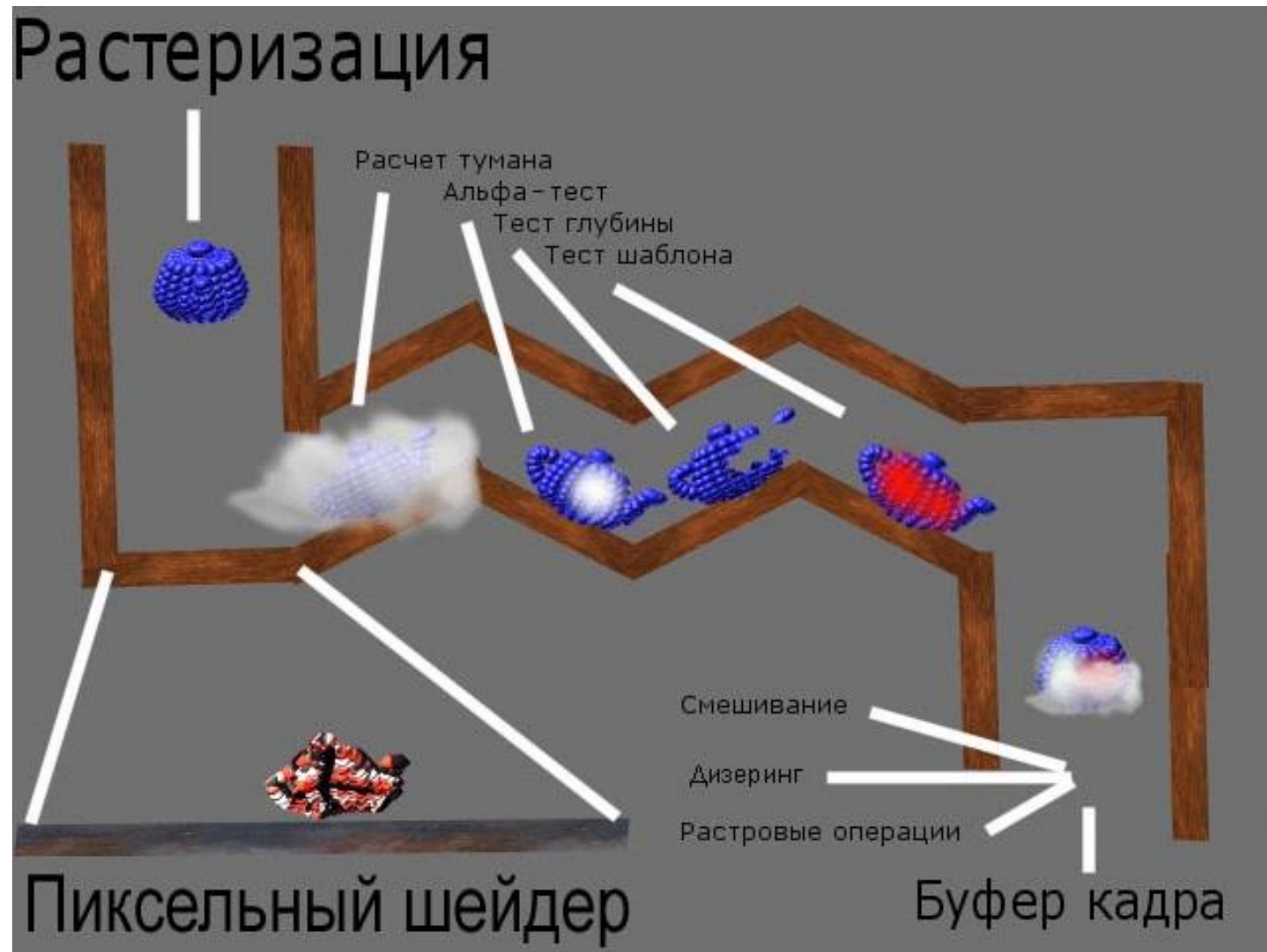


# Шейдеры тесселяции и геометрический

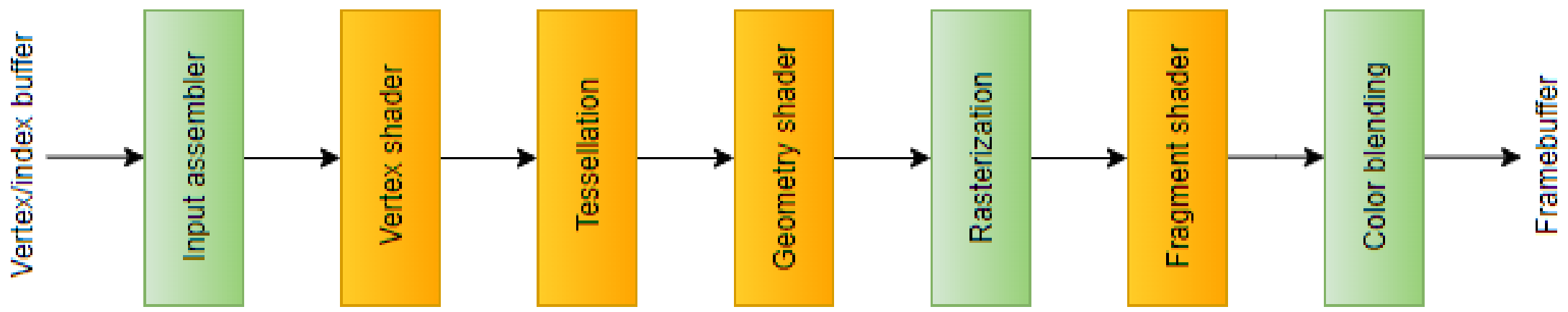
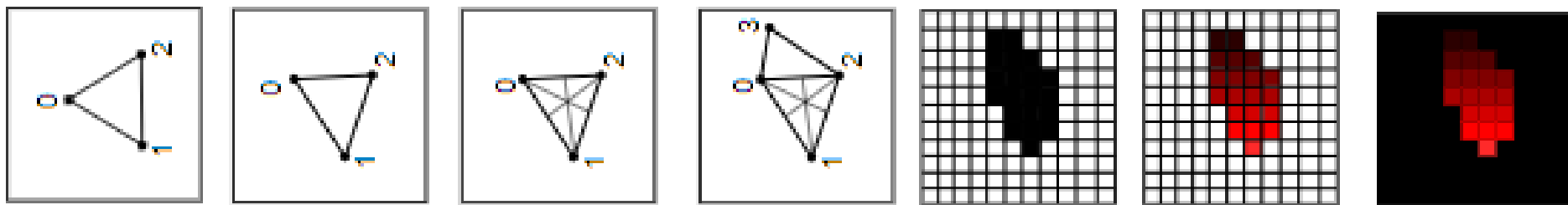


- В шейдере тесселяции генерируется скелет куста со всеми деформациями, вносимыми шевелением и ветром.
- В геометрическом шейдере на этот скелет натягивается полигональная «плоть», толщина которой зависит от высоты кости на скелете.

# Функции фрагментного шейдера



# Конвейер шейдеров



# Применение вершинных шейдеров

# Скининг (skinning) — скриншот из Call of Duty 2

Matrix pallette skinning для скелетной анимации персонажей с большим количеством «костей».

Над вершинами каждого из персонажей поработал алгоритм скининга.

Причём, с шейдерами версии 3.0 сделать скининг стало заметно проще, для шейдеров версии 1.1 нужно было писать несколько шейдеров для каждого вида скининга (с определенным количеством «костей»).



# Деформация объектов + Displacement Mapping



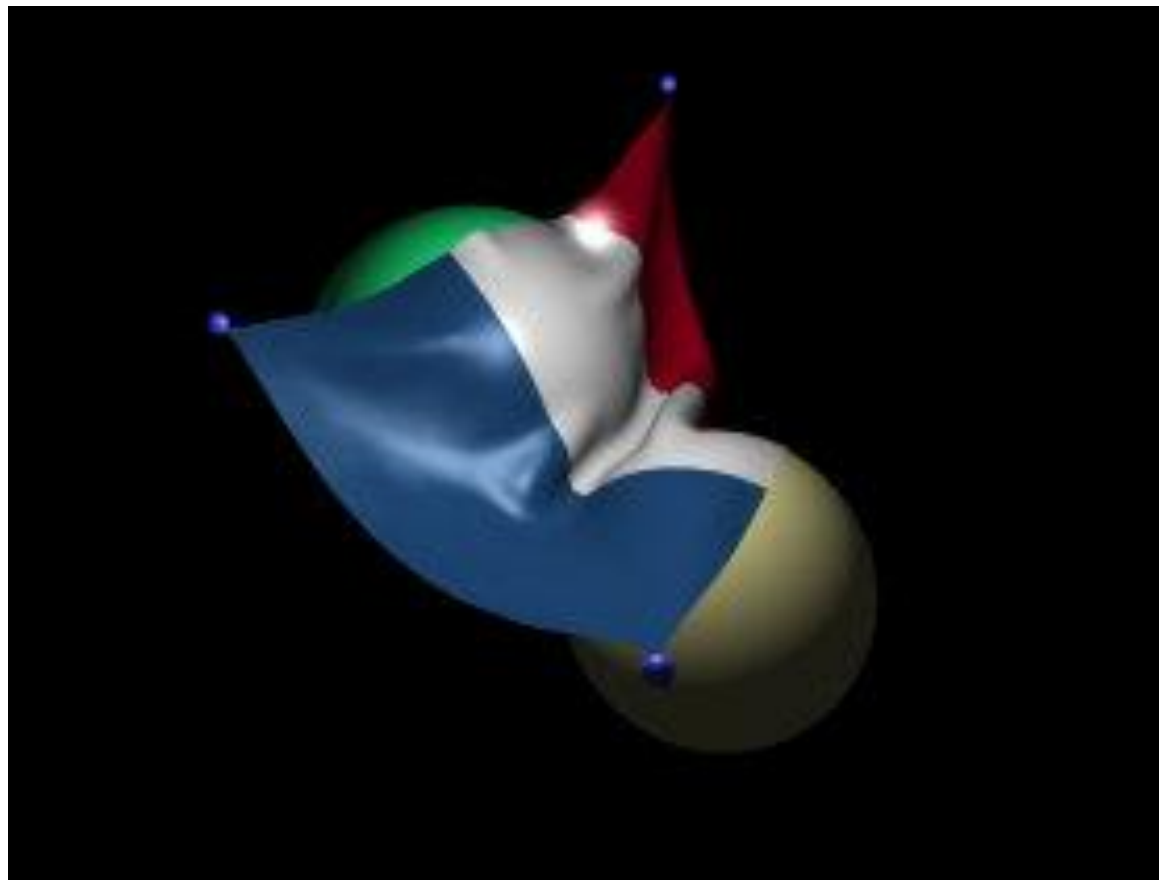
Как самый явный и эффектный пример — создание реалистичных волн в динамике. Примеры подобных решений наблюдаются в играх F.E.A.R. и Pacific Fighters. Применяются вершинные шейдеры 3.0 и доступ к текстурам из них, настоящий Displacement Mapping в дополнение к Bump Mapping. Конечно, похожий эффект волн в динамике, как в F.E.A.R., может быть запрограммирован и на пиксельном уровне (Morrowind), но в данном случае речь об изменении реальной геометрии, что всегда реалистичнее выглядит.

# Анимация объектов



Анимация травы и деревьев в одном из первых применений - 3DMark 2001 SE, алгоритм анимации был значительно улучшен в следующем 3DMark 03

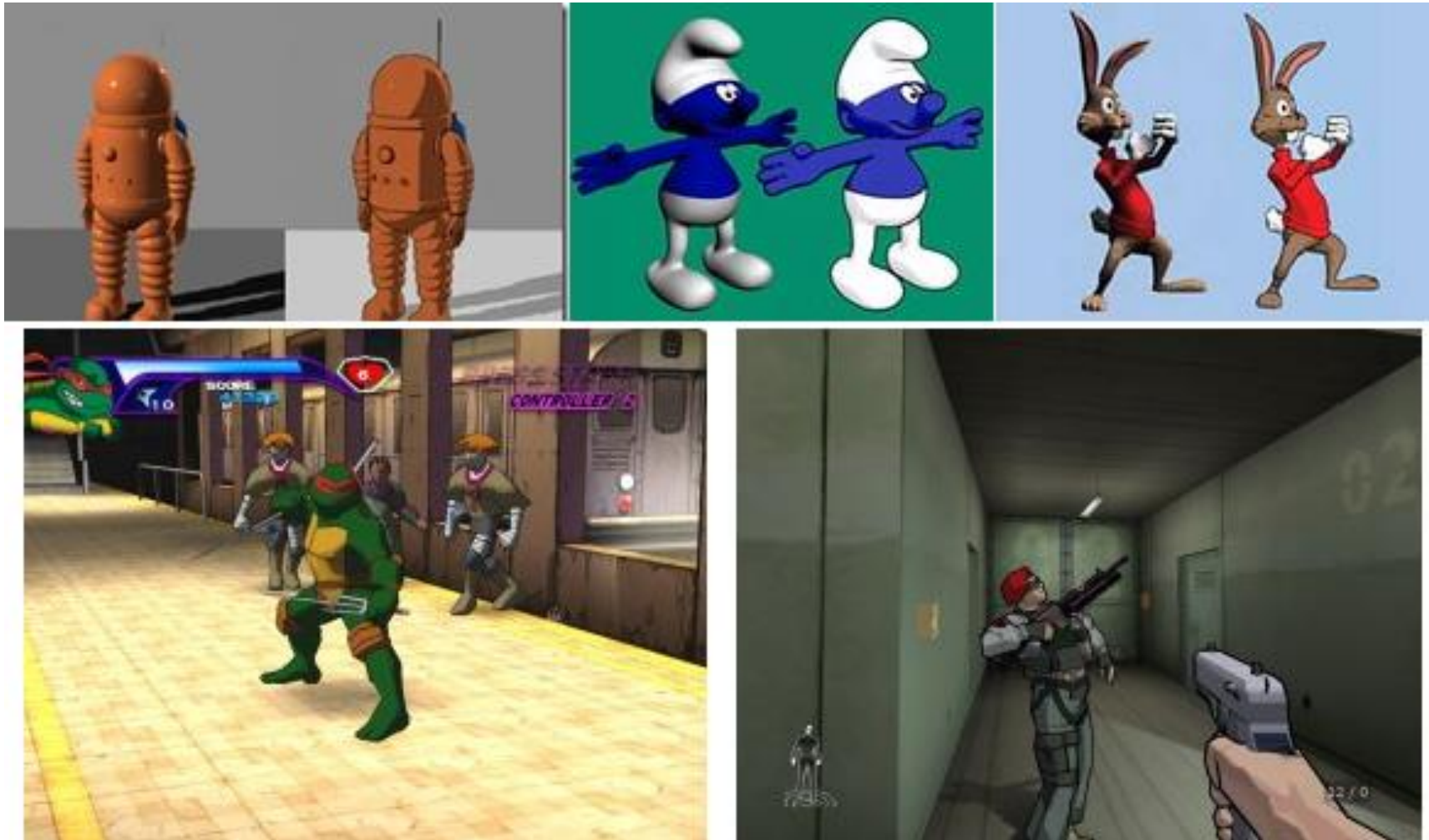
# Имитация ткани (Cloth Simulation)



Для имитации поведения материалов, подобных ткани



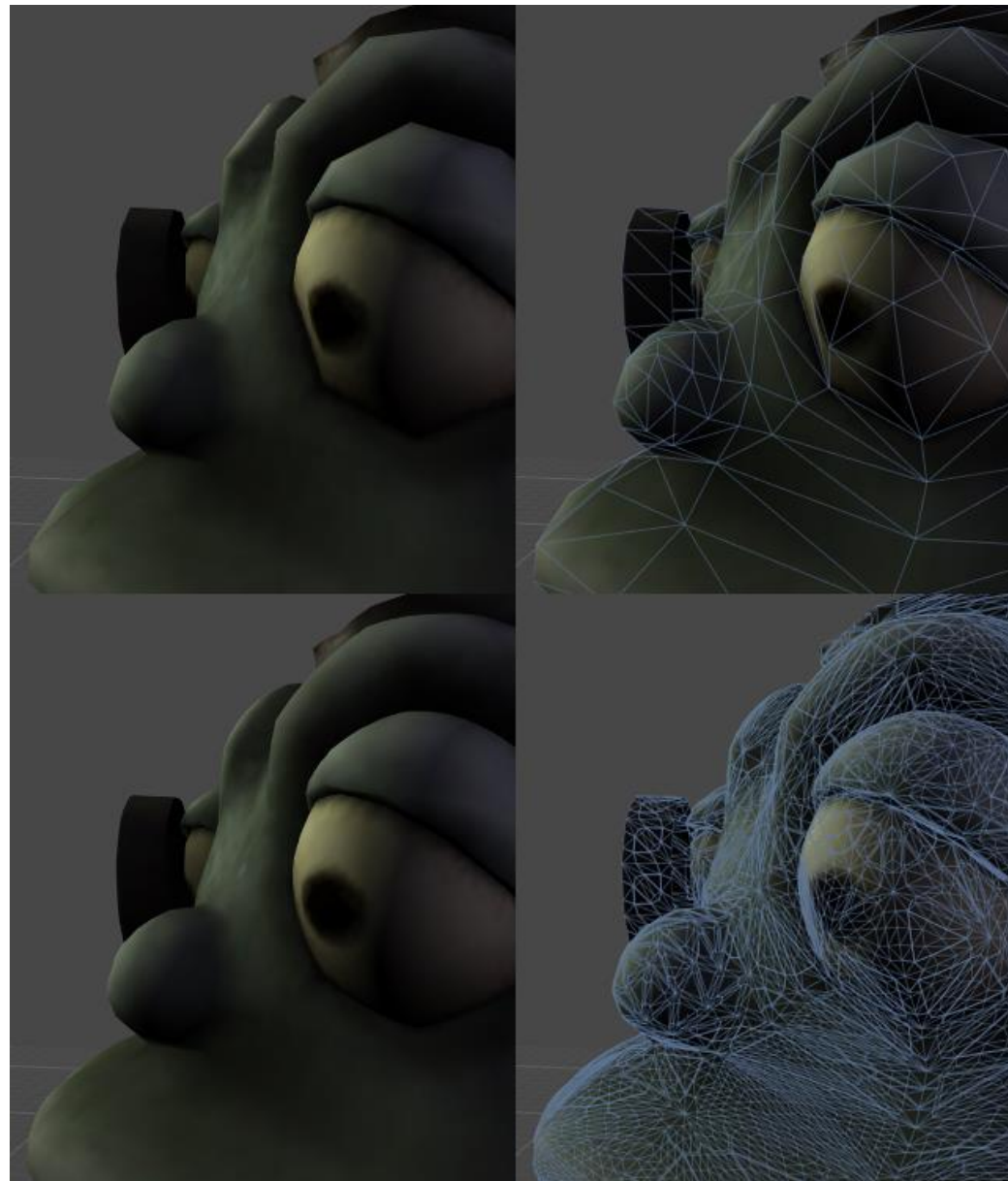
# Toon shading/Cel shading



# Применение Toon shading — Мультфильмы и мультсериалы

- Человек-паук — новые серии
- Футурама — на сел-шейдерах сделан корабль, а также трёхмерные сцены с подвижной камерой
- АКВ0048 — сел-шейдинг используется в танцевальных сценах
- Appleseed — манга, аниме
- Freedom — аниме
- Симпсоны — 19 сезон (не все серии), 20 и 21 (все серии)

# Применение тесселяционных шейдеров



# Применение фрагментных шейдеров

# Мультитекстурирование



Несколько слоев текстур (colormap, detailmap, lightmap и т.д.).  
Используется вообще во всех играх.

# Попиксельное освещение. Bump mapping. Normal mapping.



С некоторых пор применяется практически везде

# Постобработка кадра: Bloom

Эффект используется разработчиками игр для создания эффекта размытости света на ярких гранях сцены, передержки камерой при съёмке, а также для добавления большей кинематографичности изображению.



# Постобработка кадра: Depth of Field





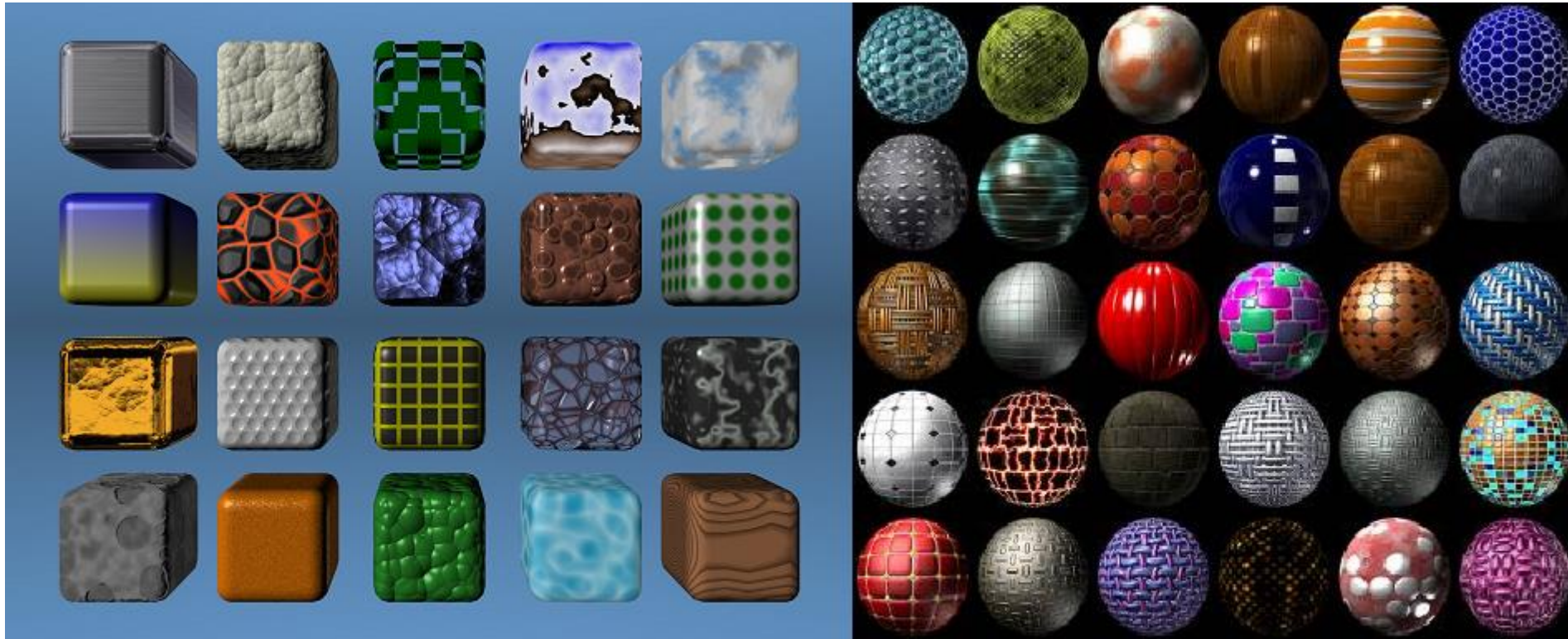
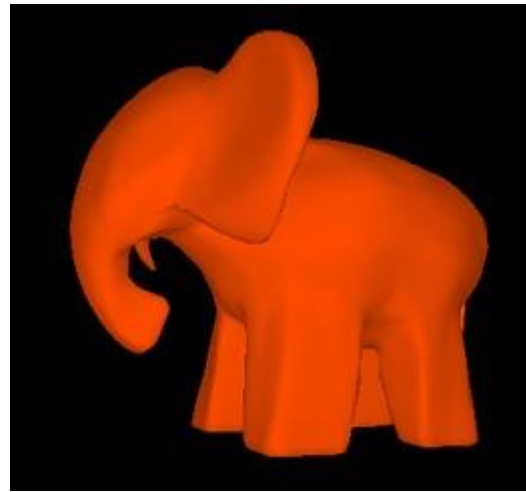
# Постобработка кадра: Motion Blur



# Процедурные текстуры



# Процедурные текстуры



# Классы шейдерных языков

- Для профессионального рендеринга
- Для рендеринга в реальном времени

# Профессиональный рендеринг

Максимальное качество визуализации

Используется для создания фотореалистичных изображений

Особые навыки программирования не требуются

Обработка таких шейдеров — ресурсоёмкая задача

Большие компьютерные кластеры или блэйд-системы

# Языки для профессионального рендеринга

- Шейдерный язык RenderMan
- Open Shading Language

# Шейдерный язык RenderMan

- первый из реализованных шейдерных языков
- является фактическим стандартом для профессионального рендеринга
- API RenderMan разработан Робом Куком
- используется во всех работах студии Pixar

# Open Shading Language (OSL)

- небольшой, но богатый язык для программирования шейдеров в развитых рендерах и других программных средствах
- разработан Sony Pictures Imageworks для использования в своем внутреннем рендере
- также используется в пакете для создания трёхмерной компьютерной графики Blender



# Языки для рендеринга в реальном времени

- GLSL — шейдерный язык OpenGL
- Шейдерный язык Cg (C for Graphics)
- HLSL(High Level Shader Language) — высокоуровневый шейдерный язык DirectX
- Низкоуровневый шейдерный язык DirectX (DirectX ASM)

# GLSL

- GLSL — шейдерный язык OpenGL
- Основное преимущество — переносимость кода между платформами и ОС

# Шейдерный язык Cg

Разработан nVidia совместно с Microsoft

Без проблем работает и с видеокартами ATI

# HLSL

Был создан Microsoft и включён в пакет DirectX 9.0

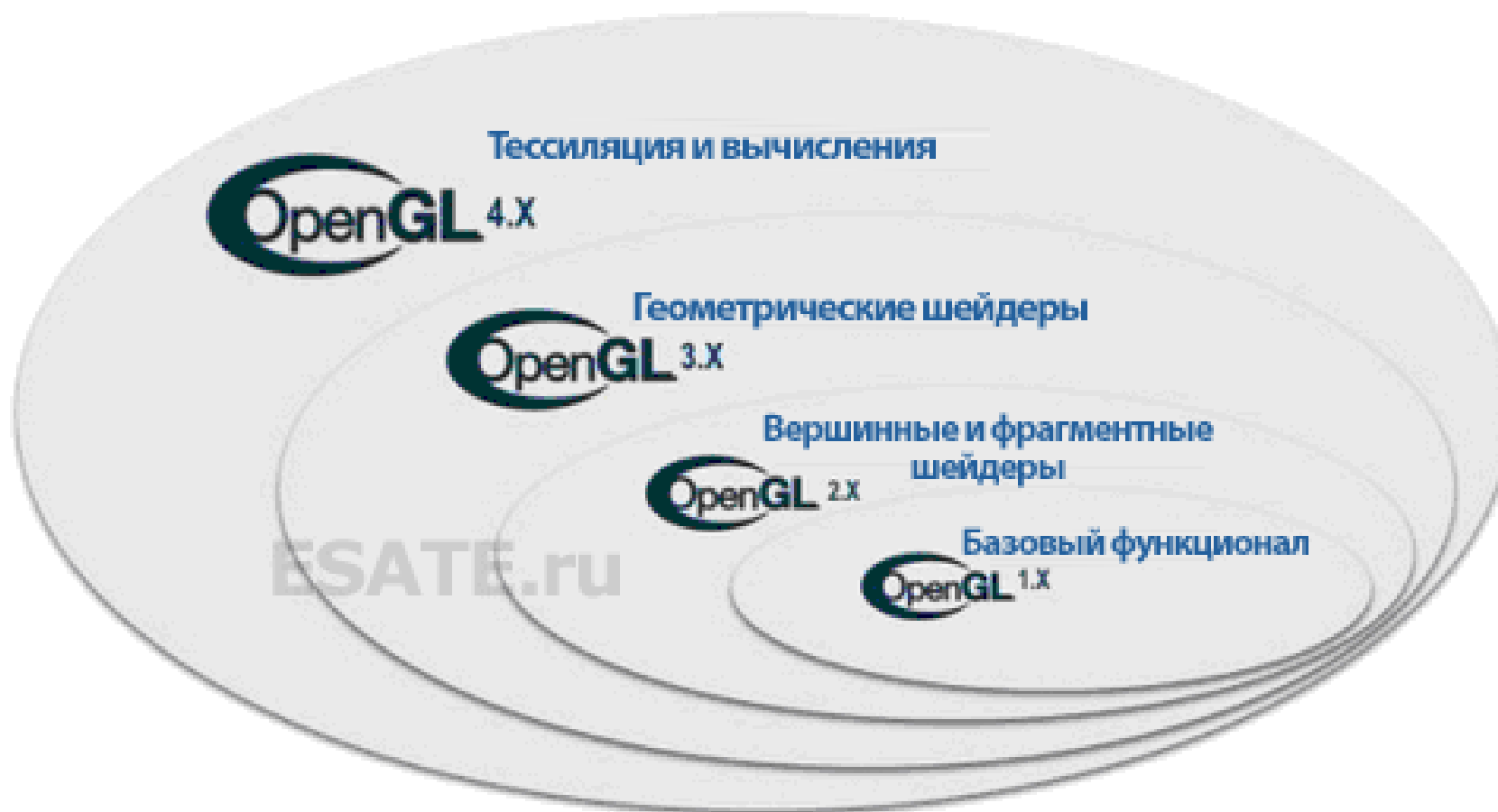
Является надстройкой над DirectX ASM

По синтаксису сходен с C

# DirectX ASM

По синтаксису схож с языком ассемблера для x86

# Эволюция OpenGL 1 - 4



# Основные вехи

GLSL 1.10 — доступен в виде набора расширений:

GL\_ARB\_shading\_language\_100,

GL\_ARB\_shader\_objects,

GL\_ARB\_vertex\_shader ,

GL\_ARB\_fragment\_shader

GLSL 2.10 — Начиная с OpenGL 2.0, GLSL включен в ядро

GLSL 3.3 — Начиная с релиза OpenGL 3.3, номер версии GLSL соответствует версии OpenGL

GLSL 4.0 — Финальный вариант спецификации OpenGL 4.0 и языка шейдеров GLSL 4.0

Две новые ступени обработки шейдеров, что позволяет перенести обработку тесселяции с центрального процессора на GPU