



Шейдерный КОКТЕЙЛЬ

Спикер:
Алексей Безгодов

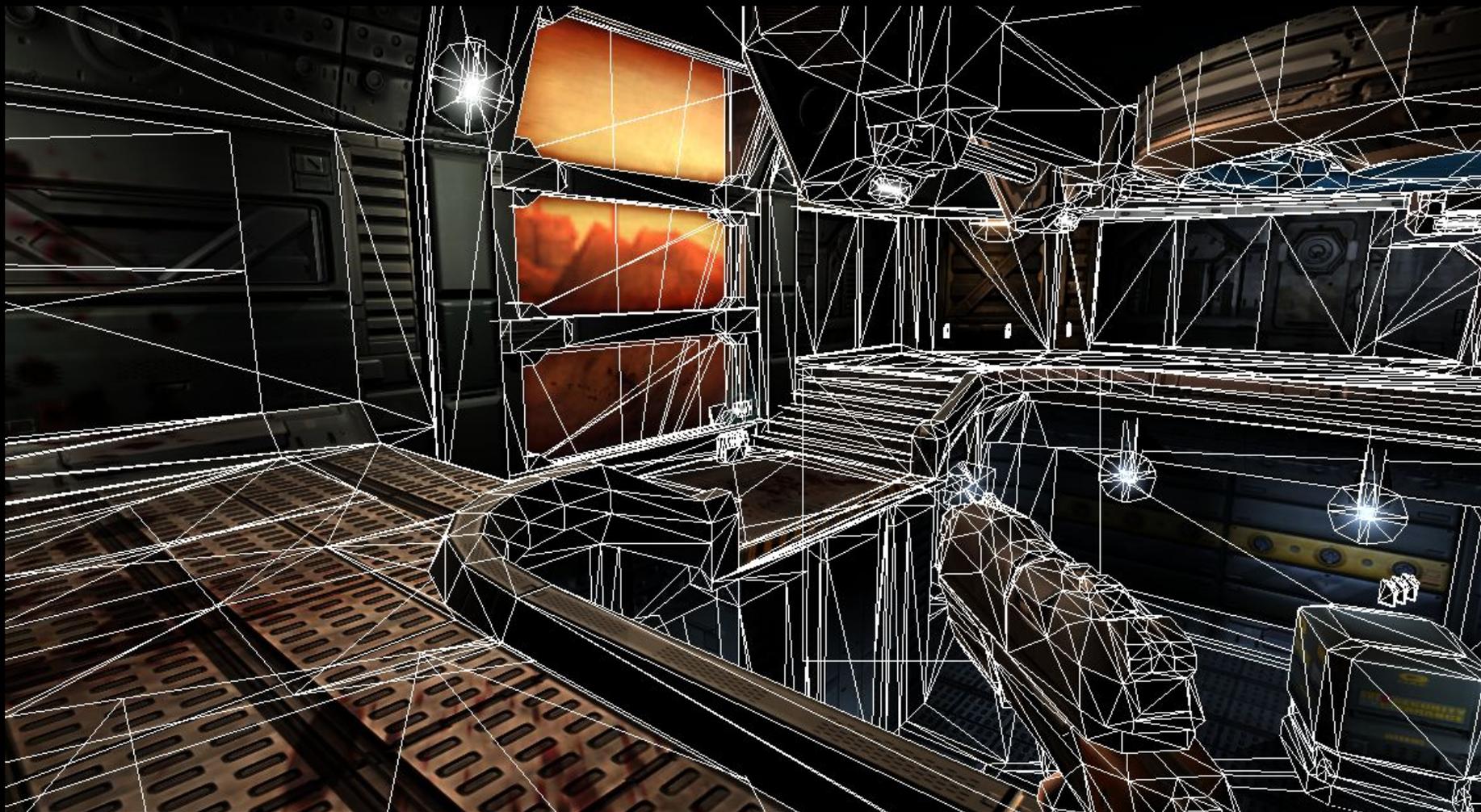
План

- Путь треугольника
- Основы CAPI
- Путь света и тени
- Постобработка
- Вопросы

Drink-Time

- Архитектура
- Приемы оптимизации
- О кафедре
- Лекция №2
- Вопросы online
- Вопросы

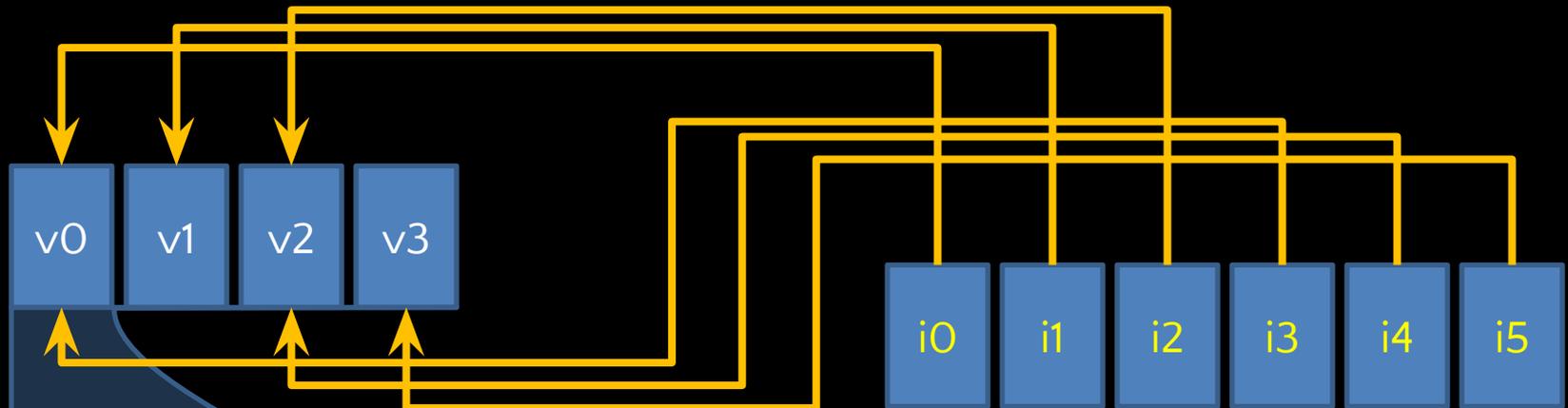
Мир состоит из треугольников



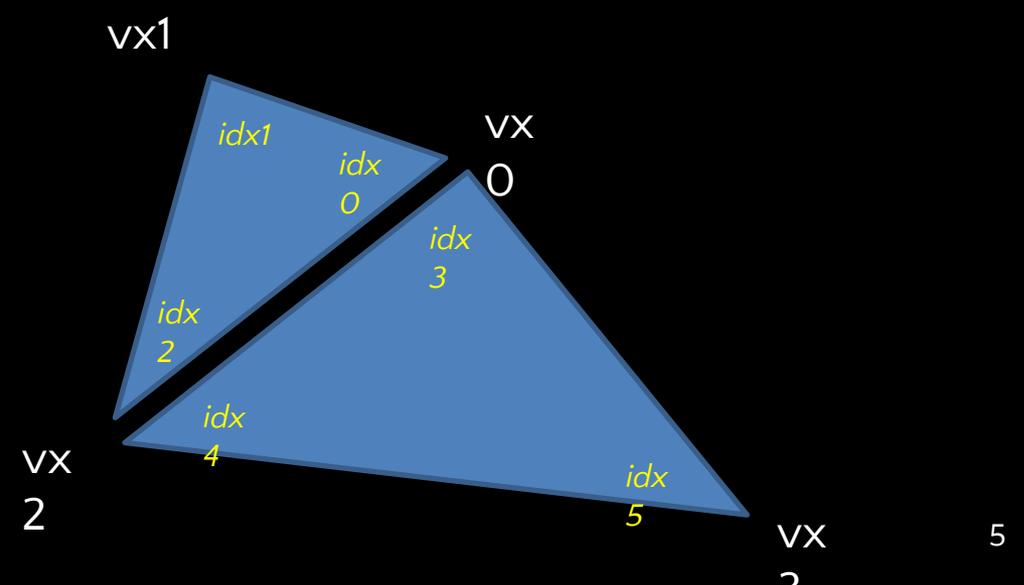
GPU: Объекты и функции

- **Shaders:**
 - VS, HS, DS, GS, PS, CS
- **Resources:**
 - Textures 1D, 2D, 3D, Cube, Array
 - Vertex Buffers
 - Index Buffers
 - Const Buffers
 - Render Targets
 - Read/Write Buffers
- **Render States:**
 - Blending
 - Rasterization
 - Output merging
 - Sampling
- **Queries**
 - Counters, Occlusion, Predicates
- **Компиляция шейдеров**
- **Создание ресурсов**
- **Передача данных**
 - CPU ↔ GPU
 - GPU ↔ GPU
 - Очистка буферов
- **Установка Render States**
- **Установка шейдеров**
- **Привязка ресурсов**
- **Draw Calls**
 - Indexed
 - Instanced
 - Indexed & Instanced
- **Установка Queries**

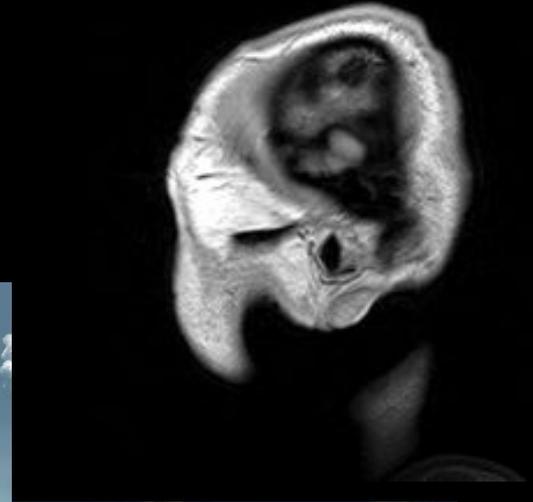
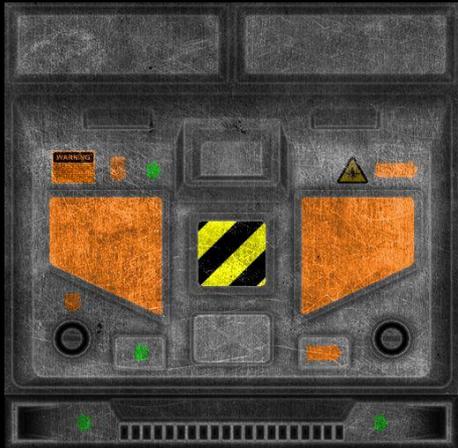
GAPI: Vertex & Index Buffers



Float4	Position
Float3	Normal
Half2	TexCoord
Byte4	Color
...	

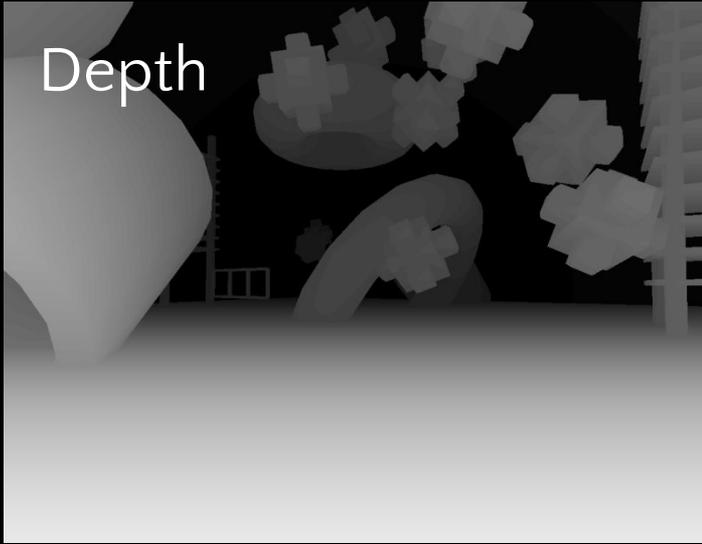


GAPI: Texture 1D, 2D, 3D, Cube



GAPI: Color, Depth, Stencil

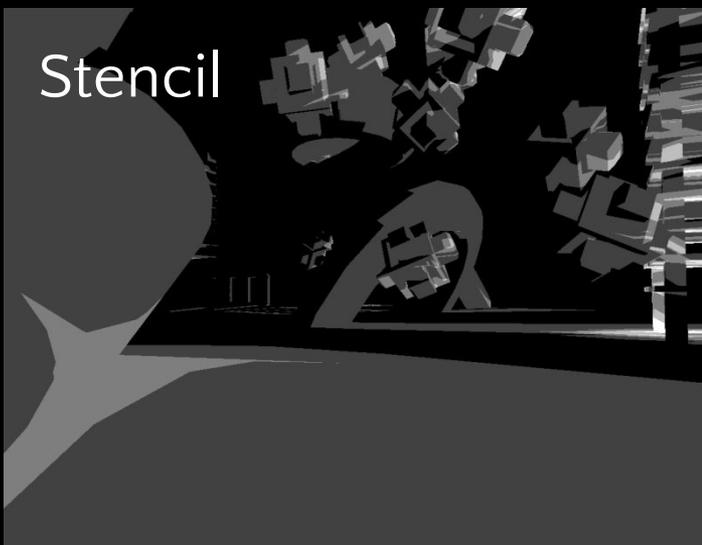
Depth



Color



Stencil



GAPI: Shaders

- **Vertex Shader**
 - преобразование вершин
- **Geometry Shader**
 - порождение примитивов
- **Pixel Shader**
 - расчет цвета фрагмента (пиксела)
- **Compute Shader**
 - общие вычисления
- **Hull Shader**
 - генерация контрольных точек кривых Безье
- **Domain Shader**
 - генерация вершин по результатам тесселяции

- HLSL – Direct3D
- GLSL – OpenGL

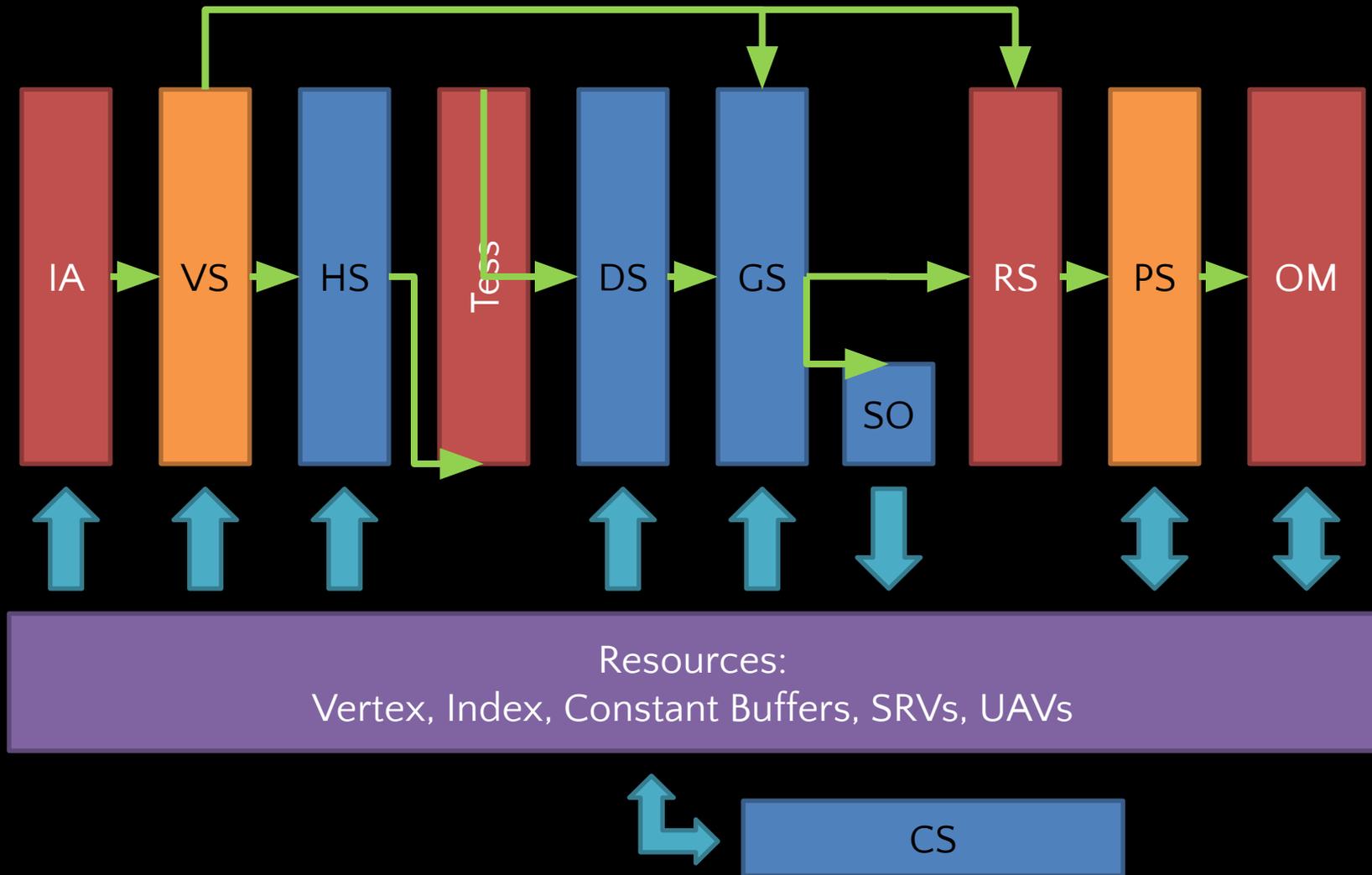
```
float4 PSMain(PS_IN input) : SV_Target
{
    float4 tex = Texture.Sample( Sampler,
                                input.tc );

    float3 bw = dot( tex.rgb,
                    float3(0.3, 0.5, 0.2) );

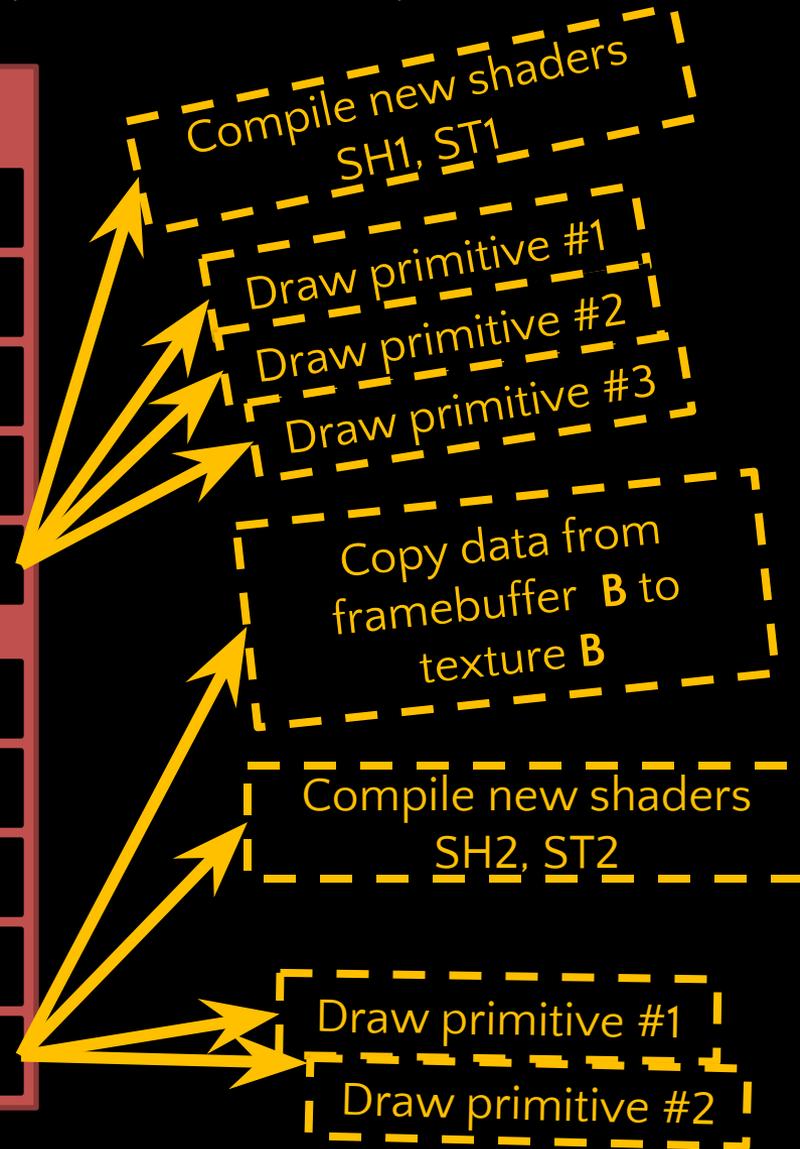
    tex.rgb    = lerp( bw.rgb, tex.rgb,
                     Batch.Saturation );

    return input.col * tex;
}
```

GAPI: Pipeline (Direct3D 11)



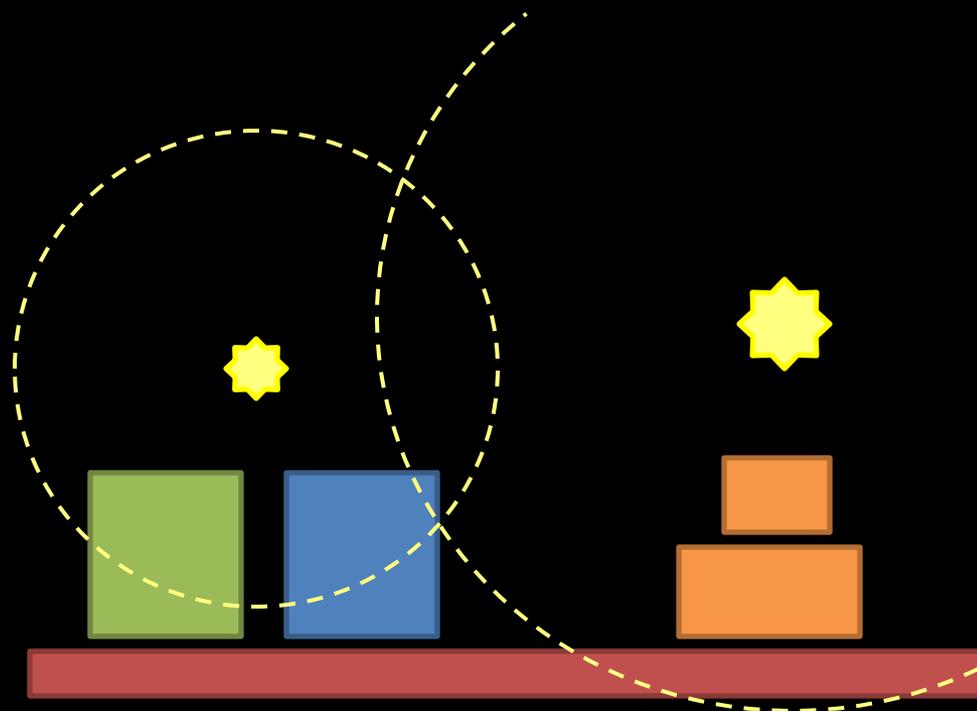
GAPI: Pipeline (~Driver)



Вопросы

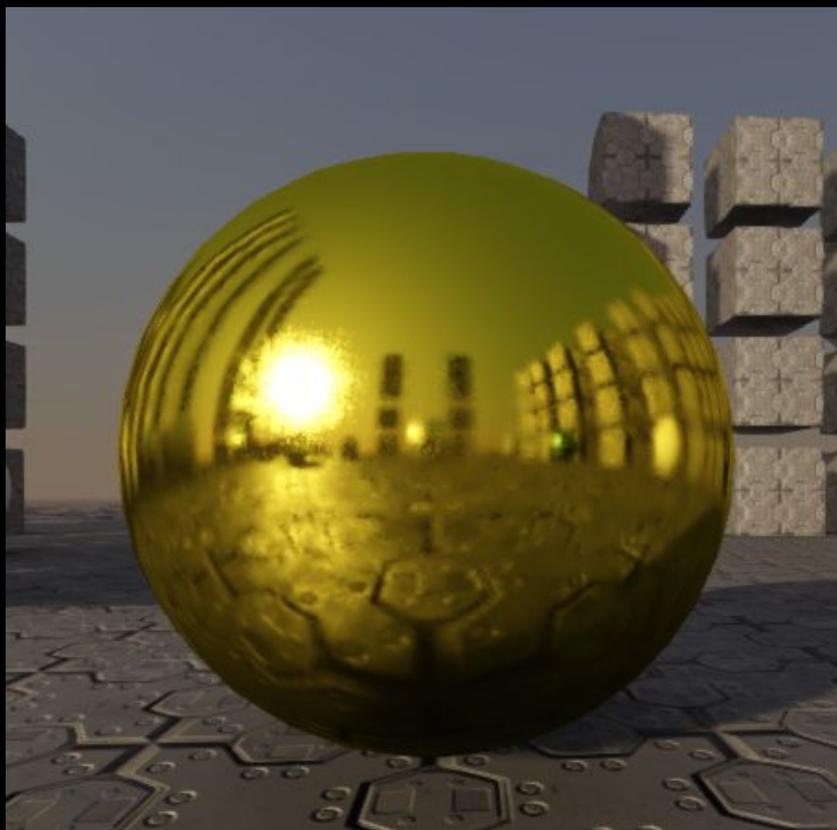
Путь света и тени

- **Рассчитать освещенность**
 - Учесть тени
 - Учесть вторичную освещенность
 - Учесть отражение
- **Вывести на экран**
 - DOF
 - HDR
 - Bloom
 - Постпроцессинг

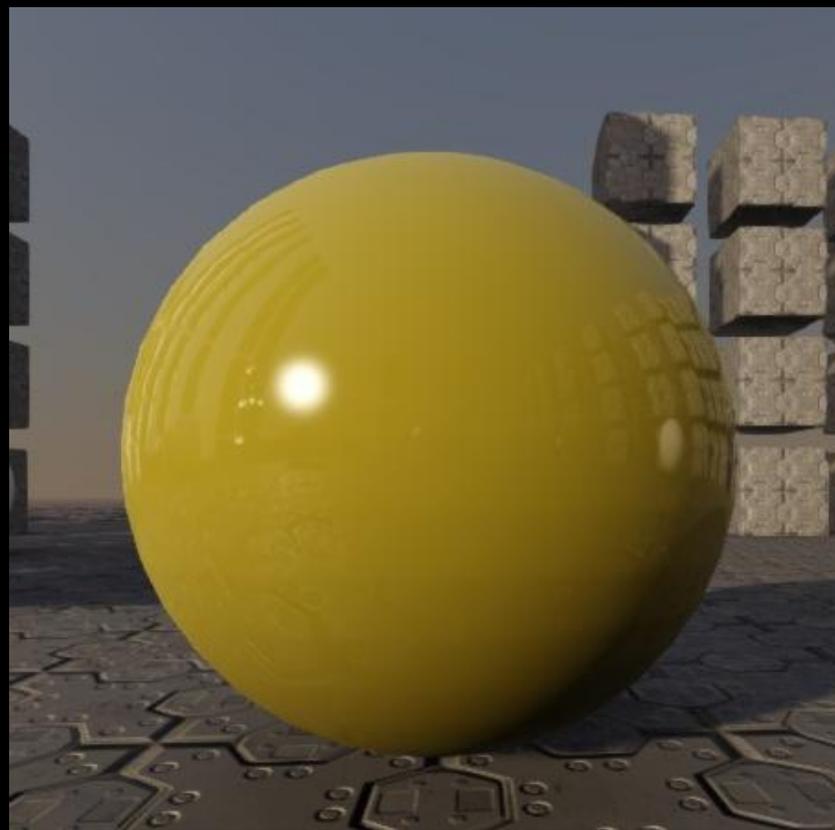


Путь света: свойства материалов

Металл

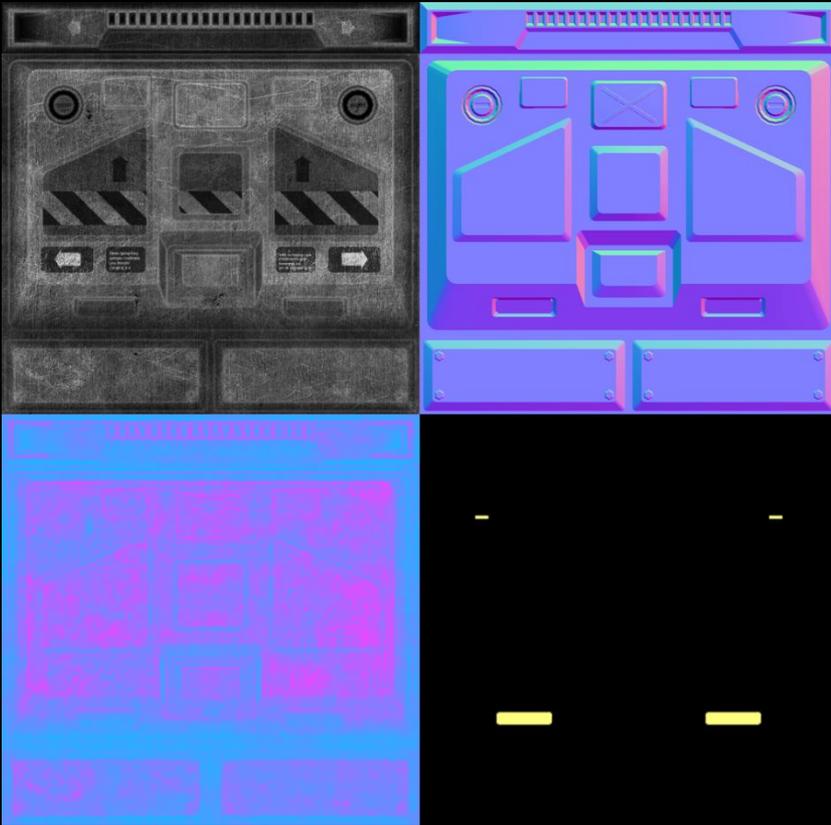


Диэлектрик

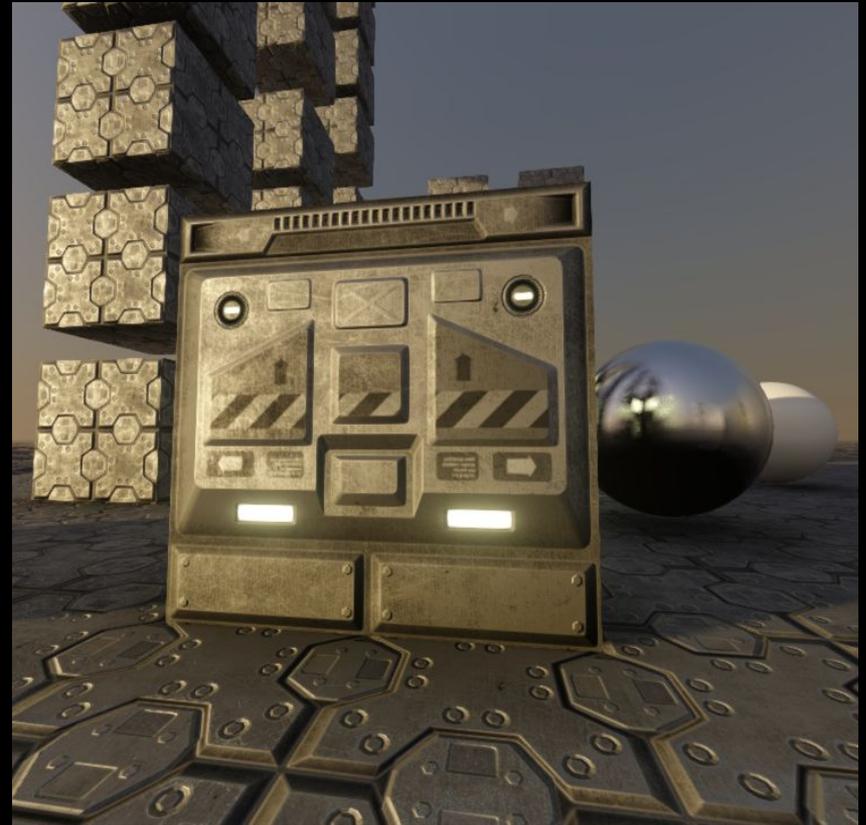


Путь света: свойства материалов

Текстуры



Результат



Путь света и тени

Свет

- Forward
 - Singlepass
 - Multipass

- Deferred
 - Classic
 - Tiled
 - Clustered

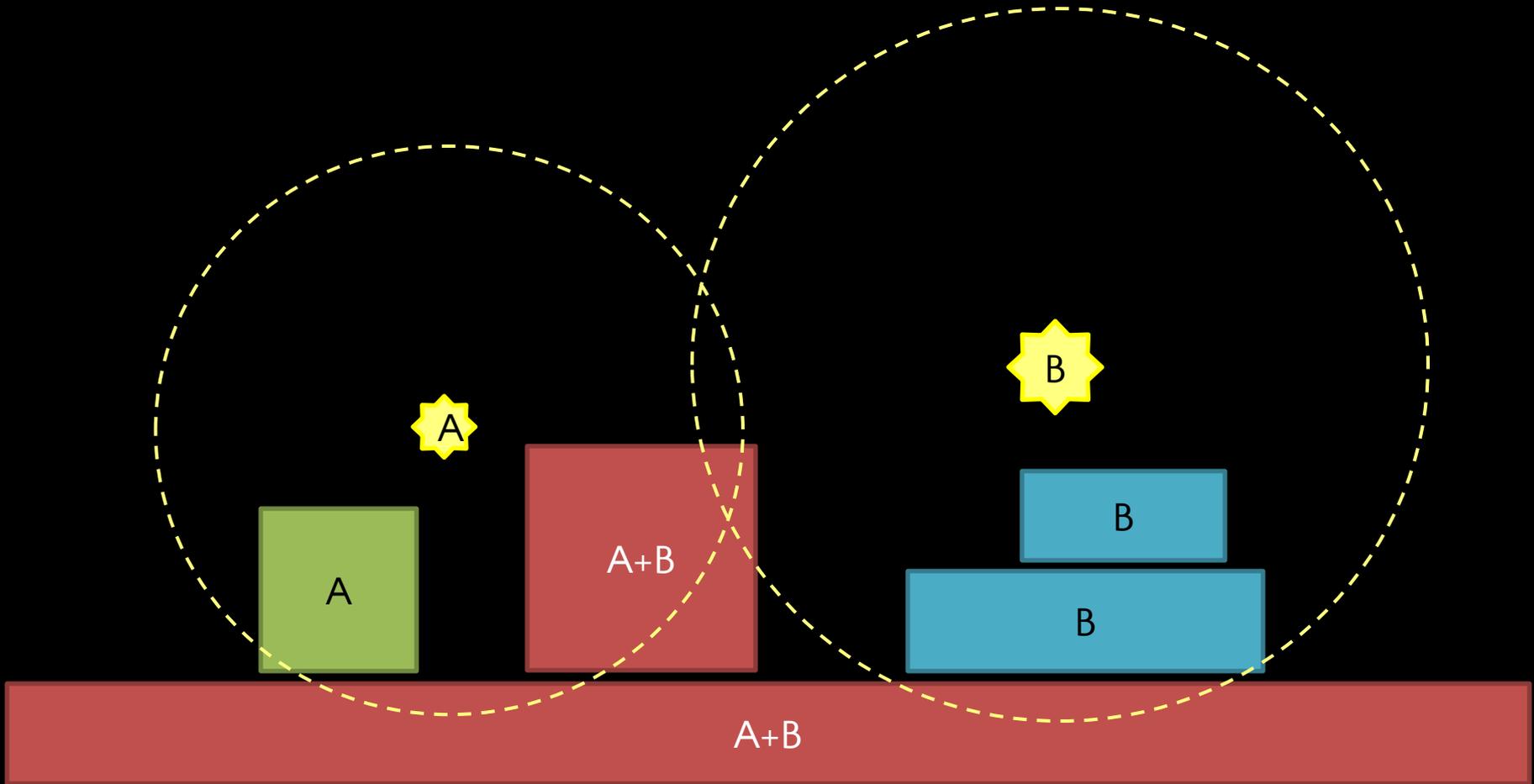
- Forward+
 - Tiled
 - Clustered

Тень

- Light Maps
- Light Grids
- Shadow Volumes

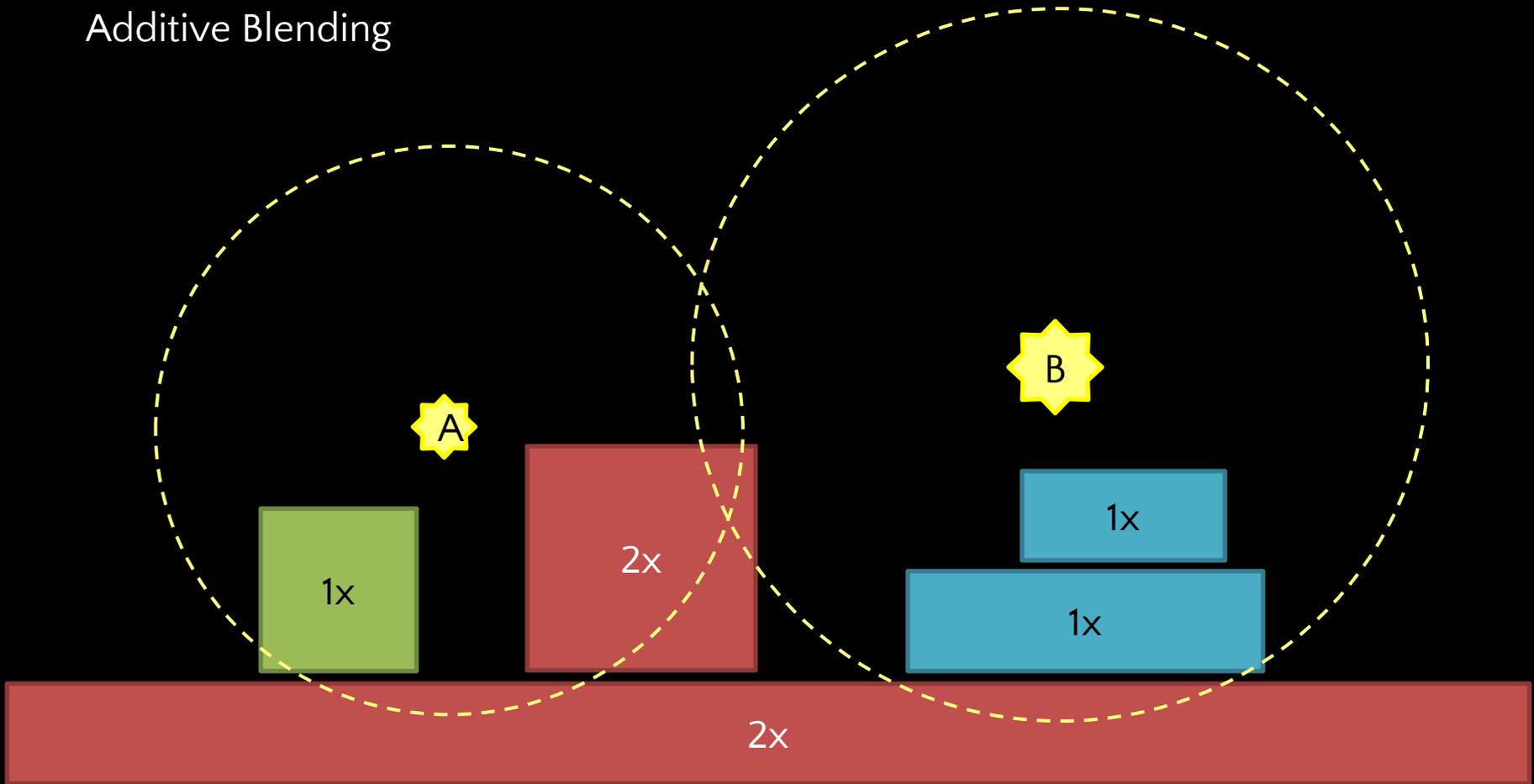
- Shadow Maps
 - Spot
 - Cascade
 - Cube
 - Variable Penumbra

Forward: Singlepass



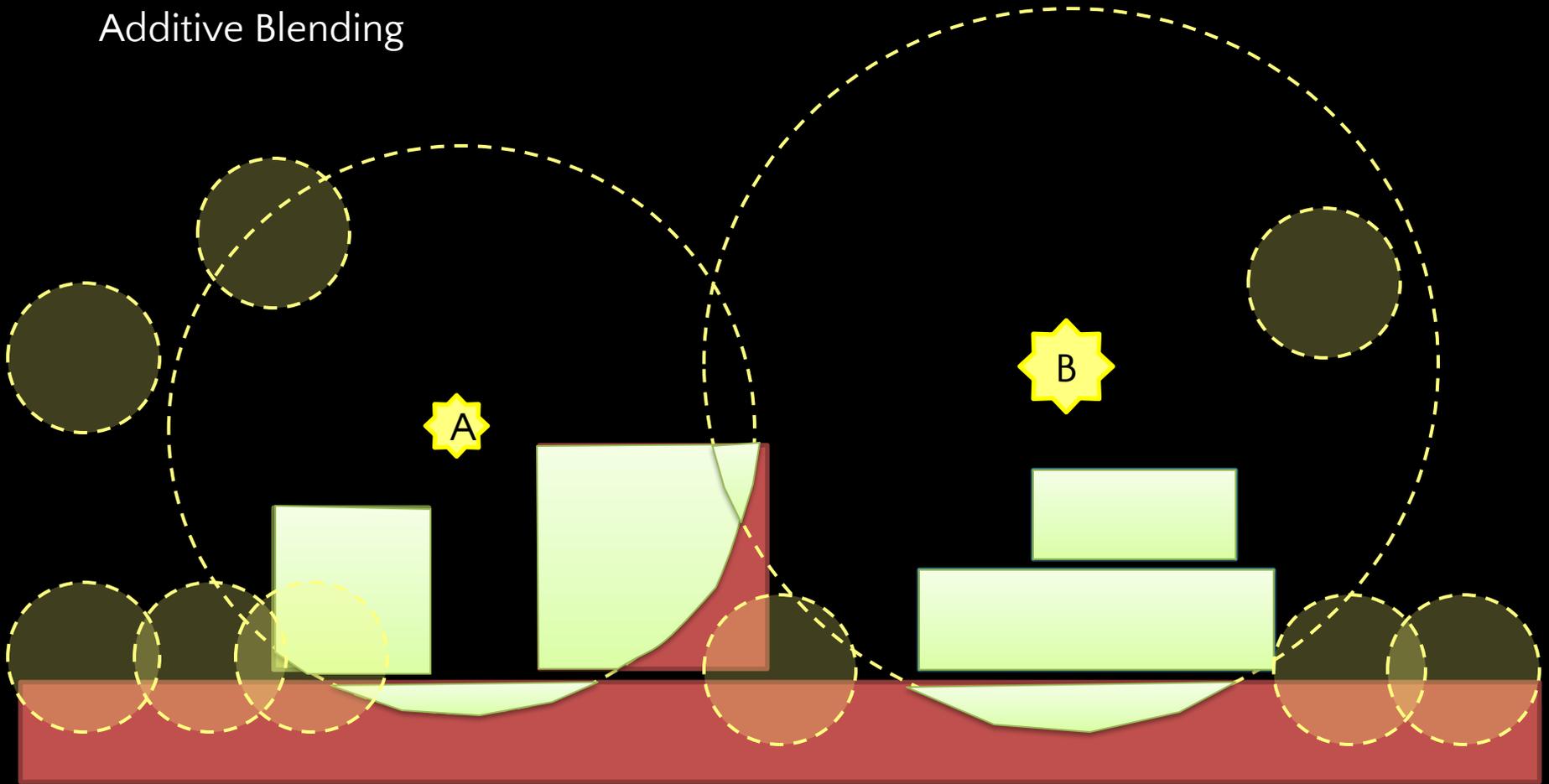
Forward: Multipass

Additive Blending



Deferred: Classic

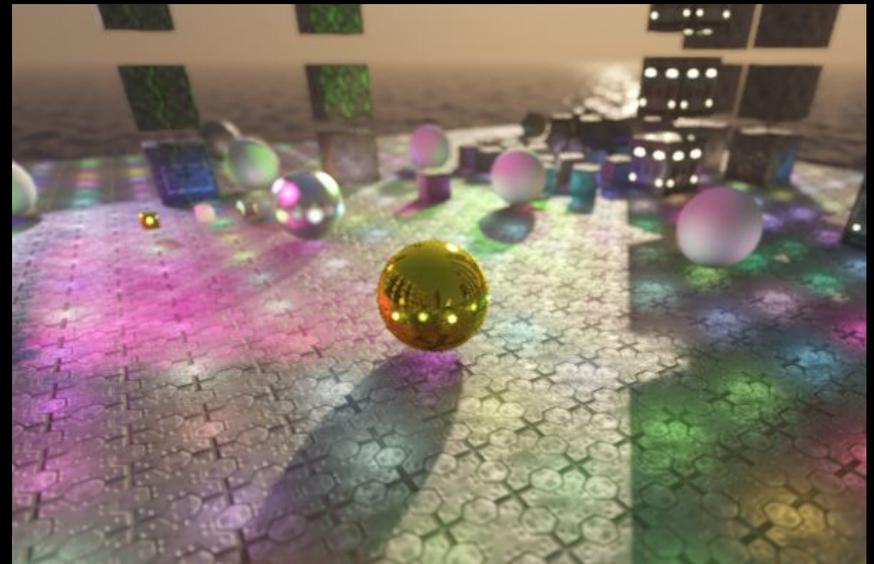
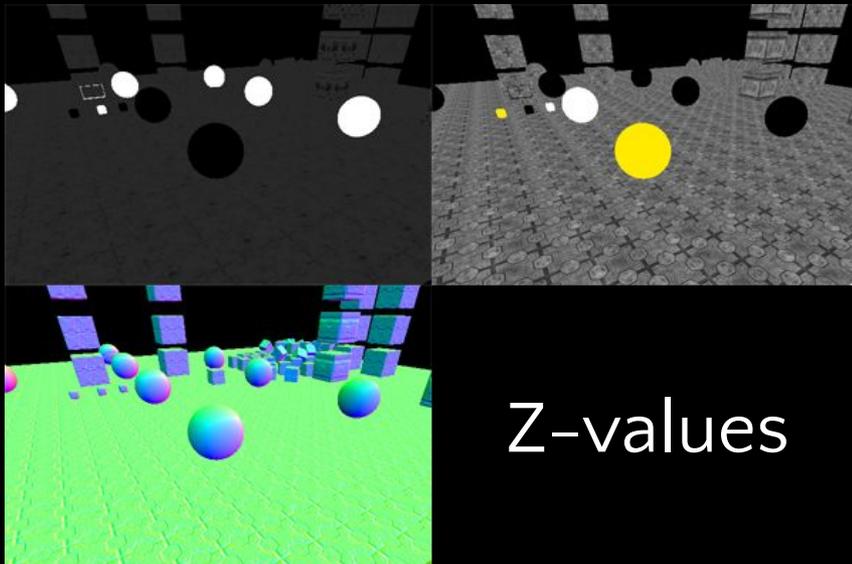
Additive Blending



Deferred Rendering

Геометрический буфер

Результат



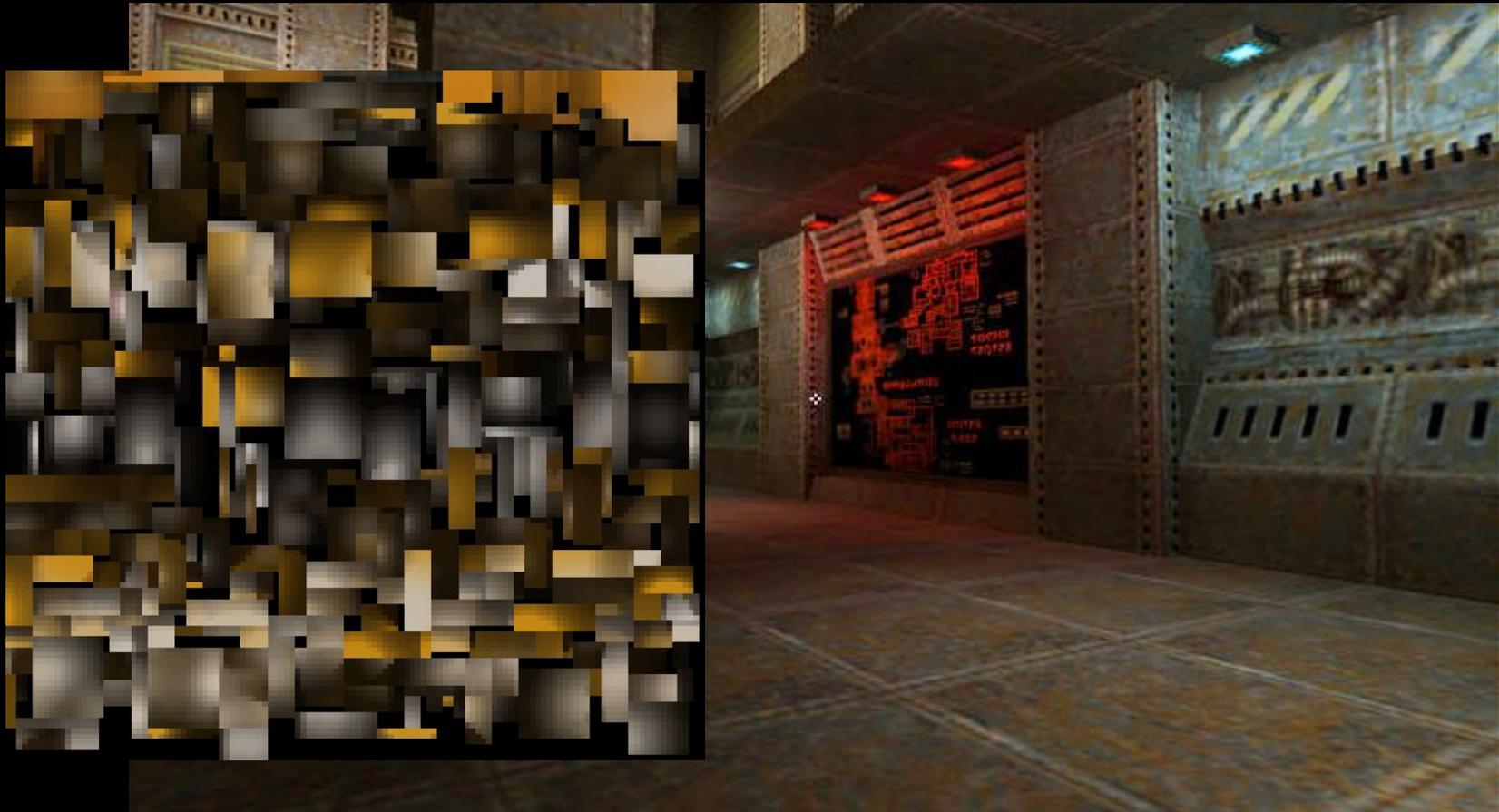
Сравнение

Technique	Num Lights	Bandwidth	Z-pass	Complex Materials Transp.	Complex Lights	Small Lights	Shader Compl.	Shadows
Forward Singlepass	3-5	Very Good	Y/N	Yes	Y/N	No	Yes	Y/N
Forward Multipass	3-5	Bad	Y/N	Yes	Yes	No	No	Yes
Deferred	>1000	Very Bad	No	No	Yes	Yes	No	Yes
Deferred Tiled/Clusterd	>1000	Good	No	No	Y/N	Yes	Y/N	Yes
Forward+ Tiled/Clustered	>1000	Good	Yes	Yes	No	Yes	YES!!!	Y/N

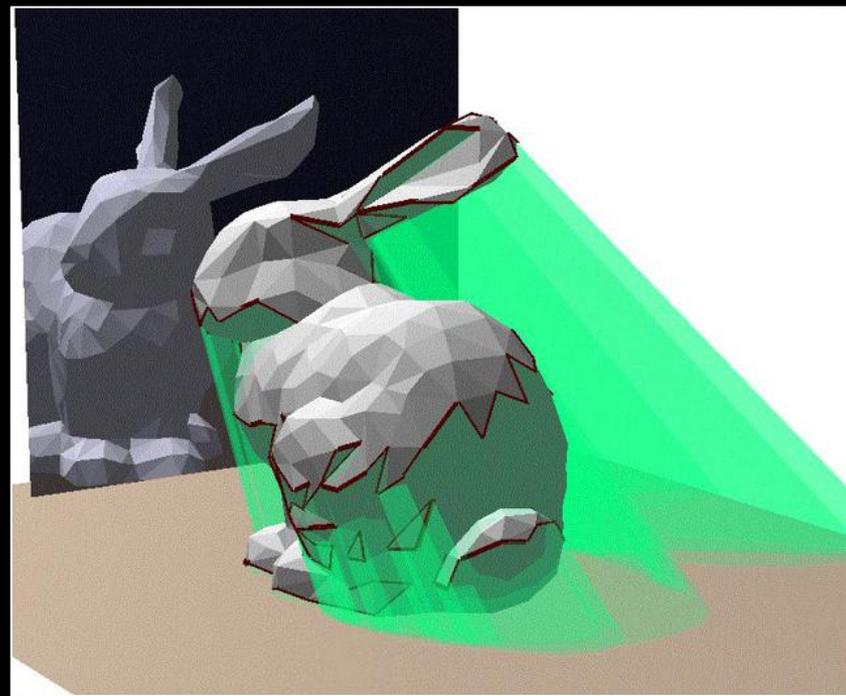
Вопросы

Drink-Time

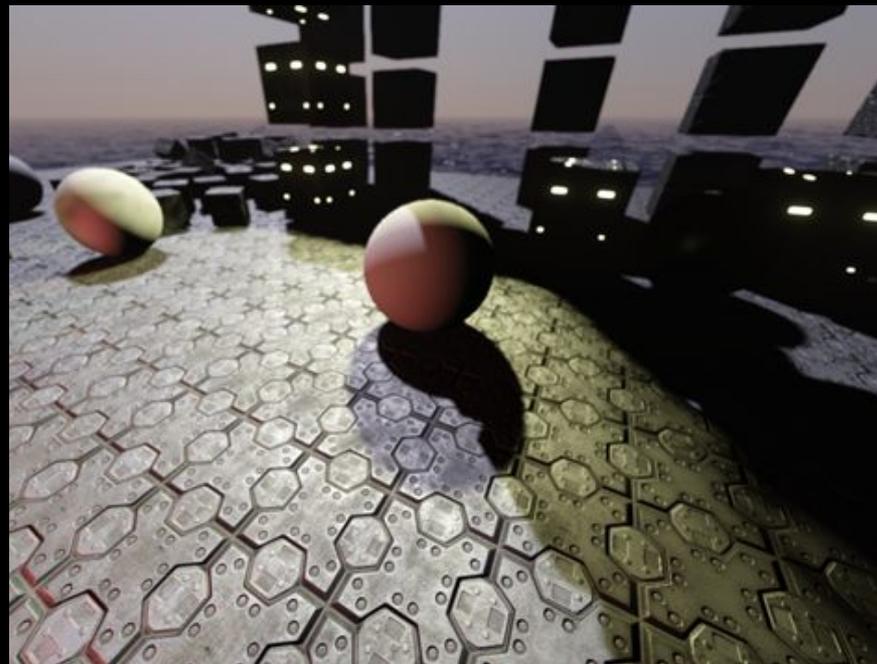
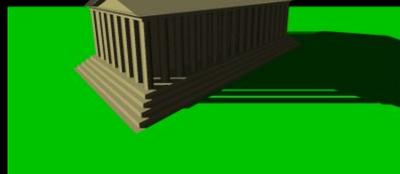
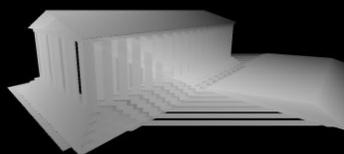
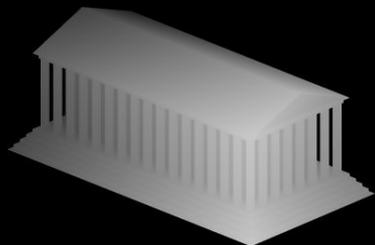
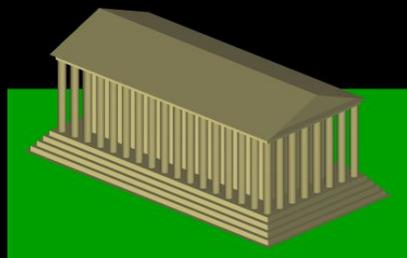
Свет и тени: Light Maps



Тени: Shadow Volumes



Тени: Shadow Maps



Глобальная освещенность

- Мягкие полутени
- Затекание света
- Отражение
- Истинная природа материалов
- Offline:
 - Radiosity
 - Photon Mapping
- Запечь в Light Map
- Запечь в Light Grid
- Запечь в Cube Maps
- SSAO
- Instant Radiosity
- ISM (*Ghost Recon: FS*)
- LPV (*CE 2,3*)
- VCT (*UE 4?*)

Global Illumination: Sky Occlusion

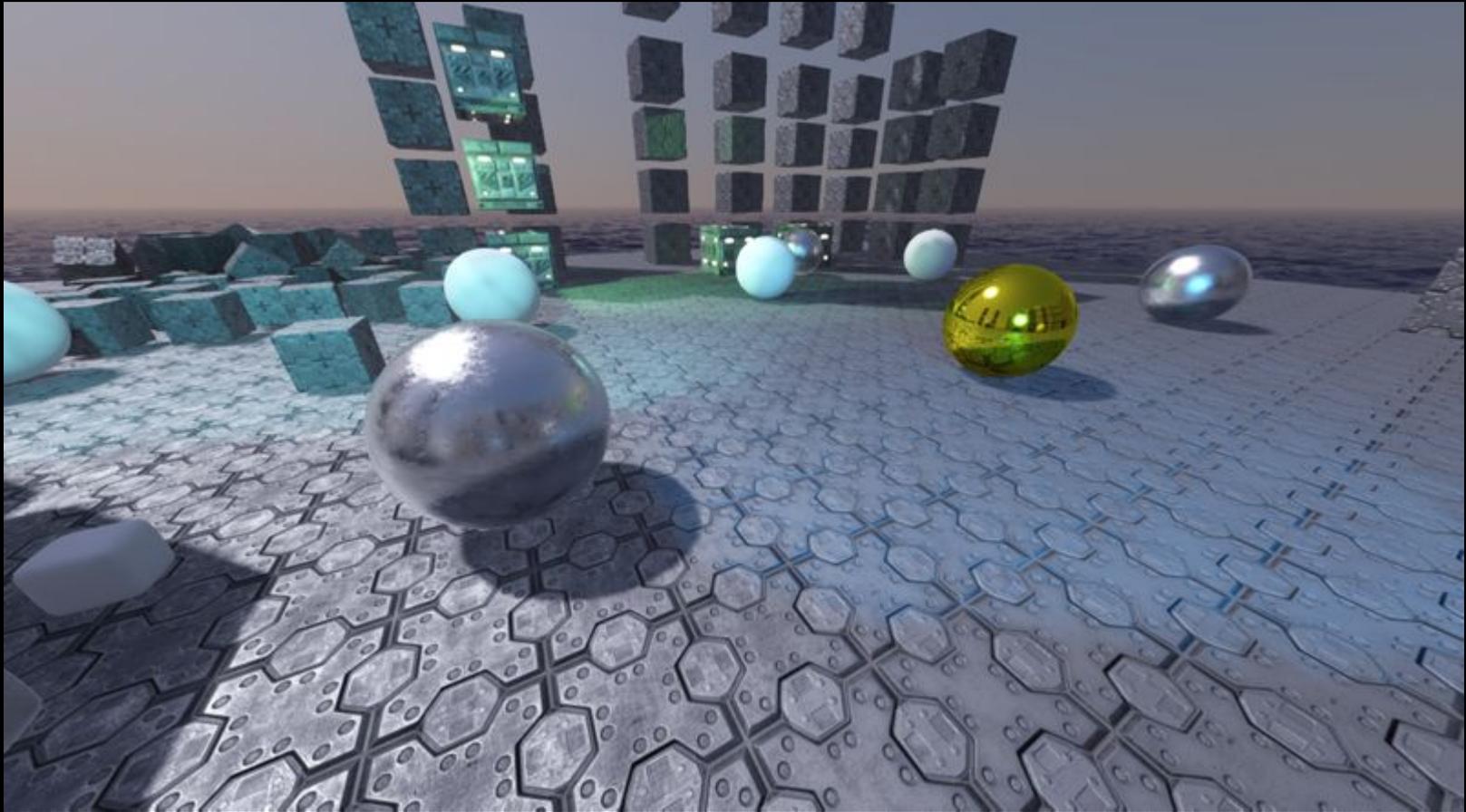


- **Cube Map**
 - 256x256x128
- **Shadow map**
 - 64 выборки
- **Каждый воксель**
 - Кол-во света
 - Усредненное направление

Global Illumination: SSAO



Global Illumination: Cube Maps



Системы частиц

- Генератор частиц (группы):
 - Начальные распределения параметров
- Частица:
 - Время жизни
 - Кинематика:
 - Позиция
 - Скорость
 - Ускорение (лок.+глоб.)
 - Дампинг
 - Цвет, Текстура
 - Анимация:
 - Вращение
 - Цвет
 - Размер
 - Параметры сложного взаимодействия:
 - Упругость
 - Заряд/Масса
- Симулятор частиц:
 - Численный
 - $x = x + v \cdot t; v = v + a \cdot t$
 - Эйлер
 - Рунге-Кутта
 - Аналитический
 - $x = x_0 + v \cdot t + a \cdot t^2 / 2$
- Использование GPU
 - GS Stream Output / CS Append Buffer
 - Моделирование
 - Время жизни
 - Сортировка/Approx. OIT
 - VS / PS
 - Ориентация спрайтов
 - Анимация

Übershaders

```

void ShaderFunc () {
    #ifdef WET
        SetWetSurface ();
    #endif
    #ifdef SKYLIGHT
        ApplySkyLight();
        #ifdef SHADOW
            ApplyShadow();
        #endif
    #endif
    #ifdef LIGHT0
        ApplyLight ( 0 );
    #endif
    #ifdef LIGHT1
        ApplyLight ( 1 );
    #endif
    #ifdef LIGHT2
        ApplyLight ( 2 );
    #endif
    #if XBOX
        // Do XBOX-specific stuff
    #endif
}

// Render-specific
SKYLIGHT = 0x0001
SHADOW = 0x0002
LIGHT0 = 0x0004
LIGHT1 = 0x0008
LIGHT2 = 0x0010

// Game-specific
WET
FREEZED
CLOACKED

// Stage-specific
PIXEL_SHADER
VERTEX_SHADER

// Material-specific
USE_DETAIL_MAP
USE_NORMAL_MAP
USE_PARALLAX
USE_DETAIL_NORMALMAP
USE_WET
USE_TROPIC_SPEC
USE_DETAIL_MAP

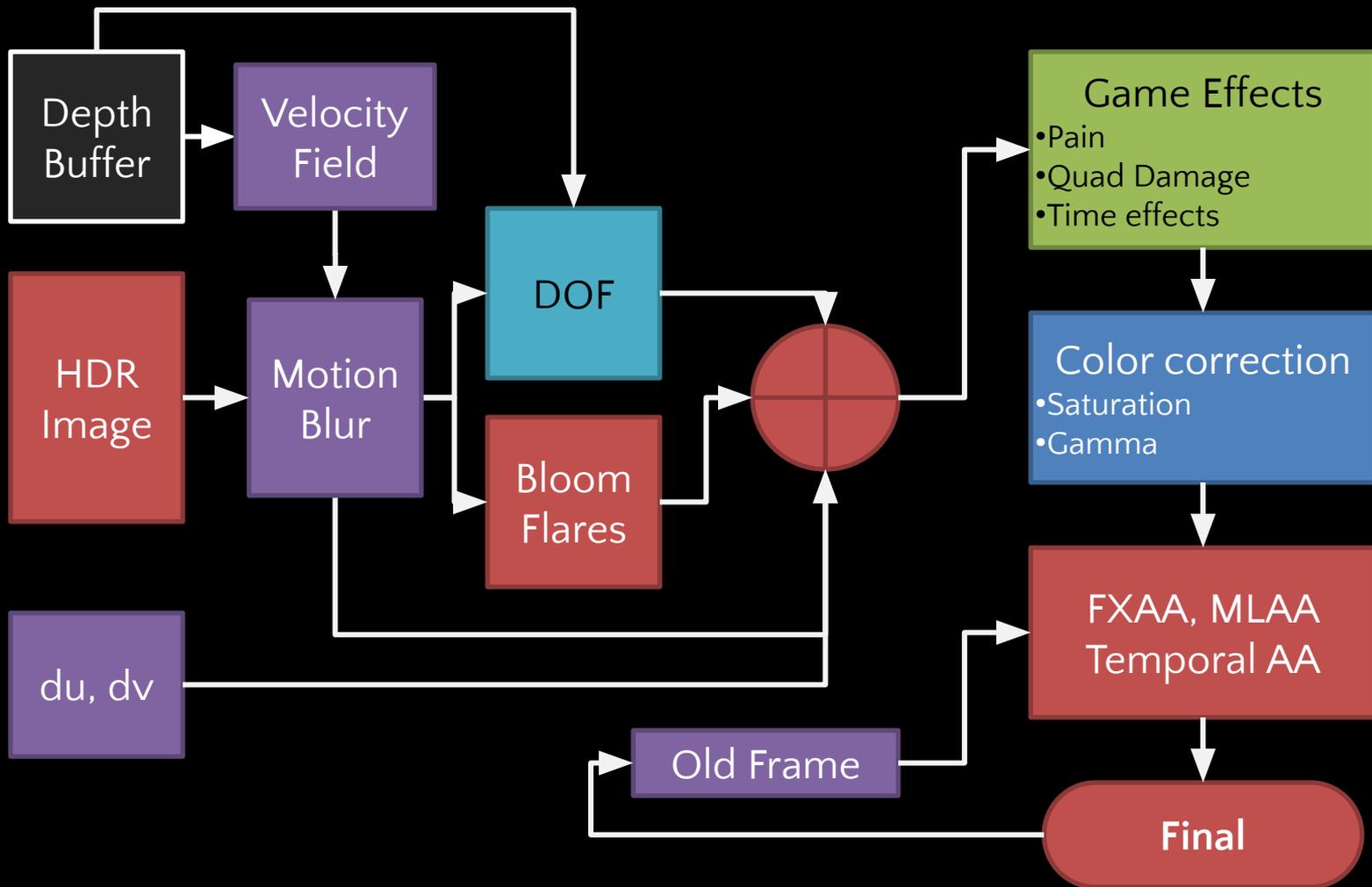
// Platform-specific
PC_OPENGL
PC_DIRECT3D
XBOX
PS3

SetUberShader(
    LIGHT0 | LIGHT1 |
    WET |
    USE_DETAIL_MAP |
    PIXEL_SHADER | PC_DIRECT3D
);

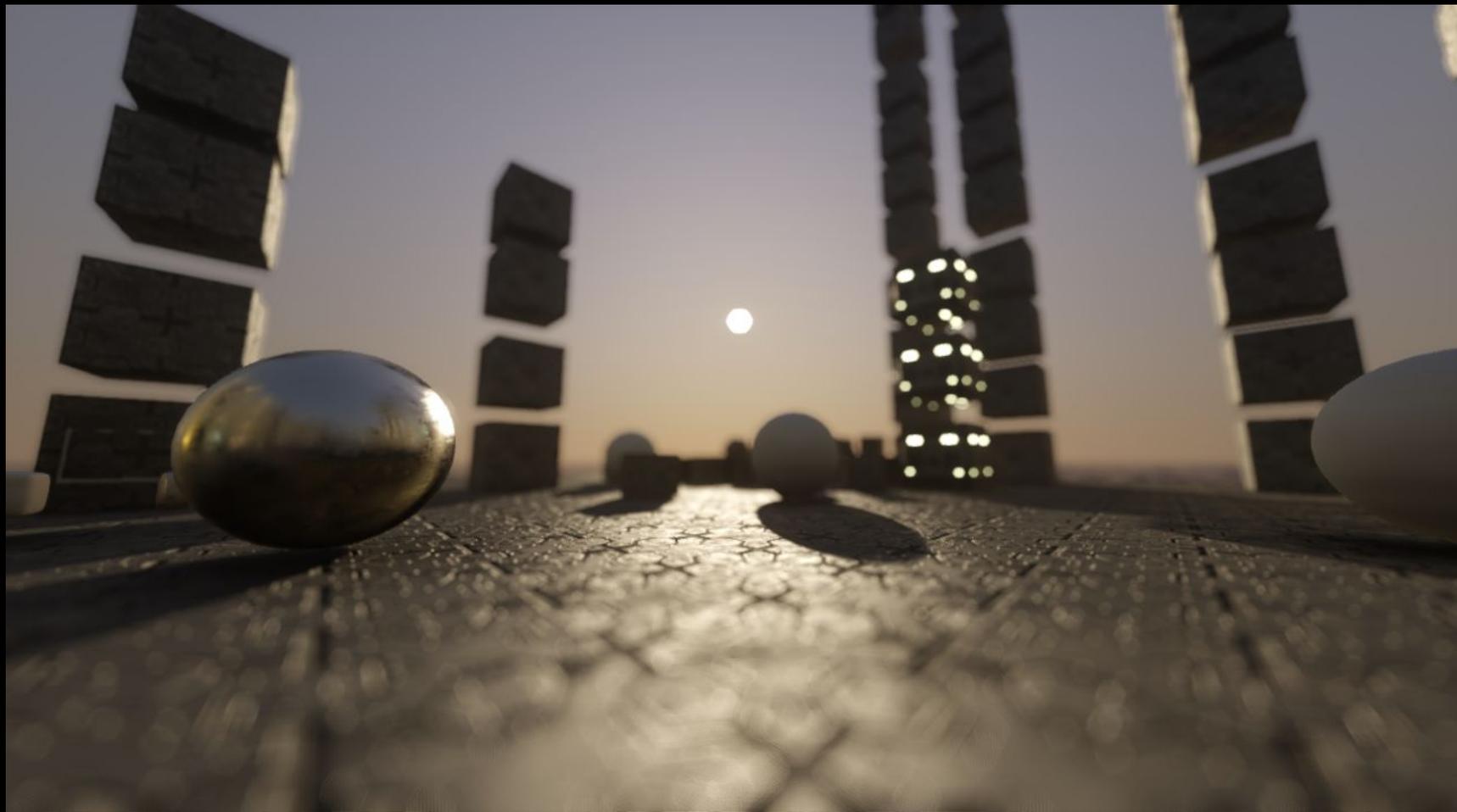
```



Постобработка

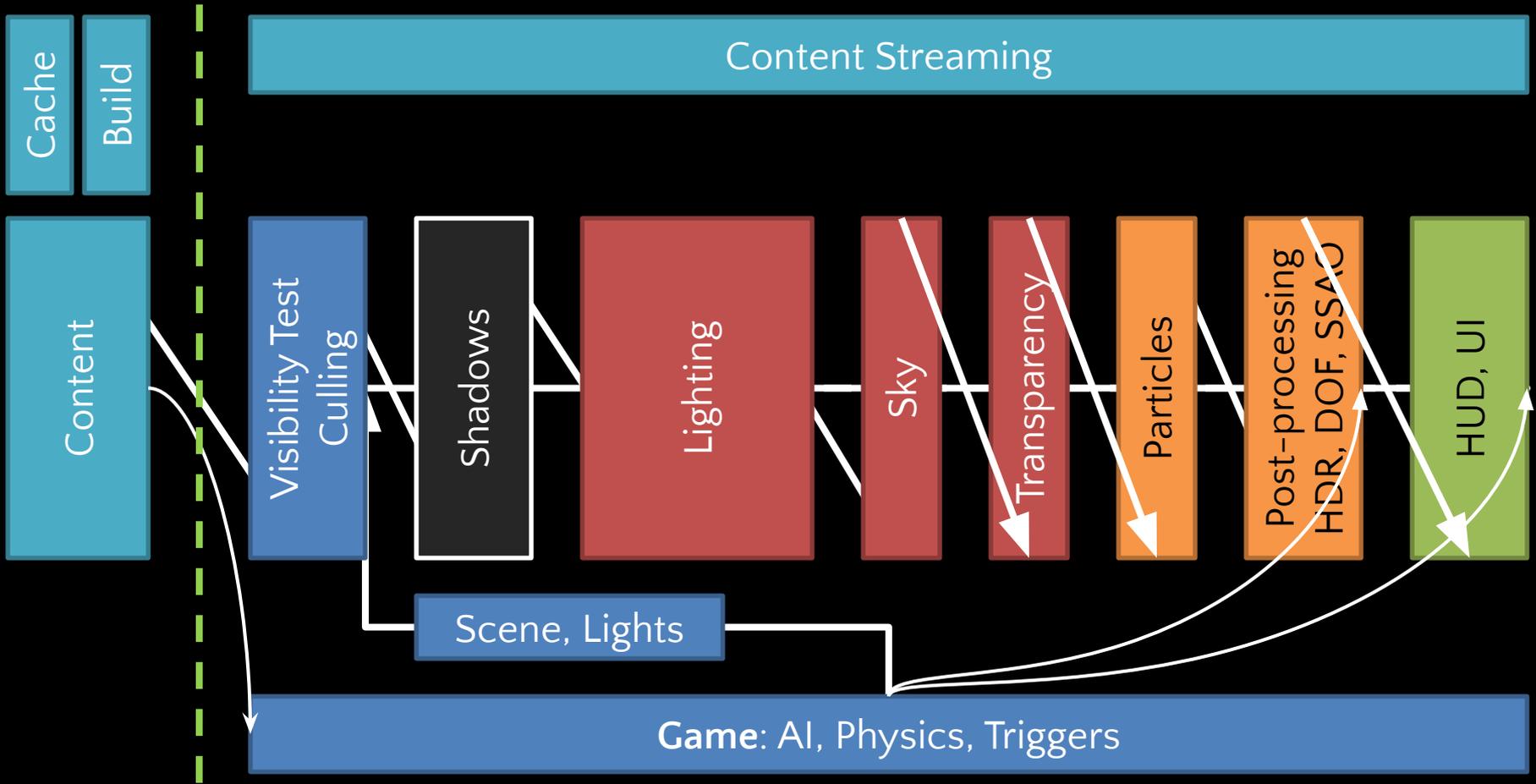


Постобработка



Вопросы

Схема графической системы



Оптимизация: Draw Calls

- Чем меньше Draw Calls – тем лучше
 - 1000-2000 батчей максимум
 - Shadows
 - Z-prepass
- Frustum culling
 - Иерархический
- Occlusion culling
 - Occlusion query/Conditional Rendering
 - Precomputed visibility
- Склеивание мешей
- Instancing

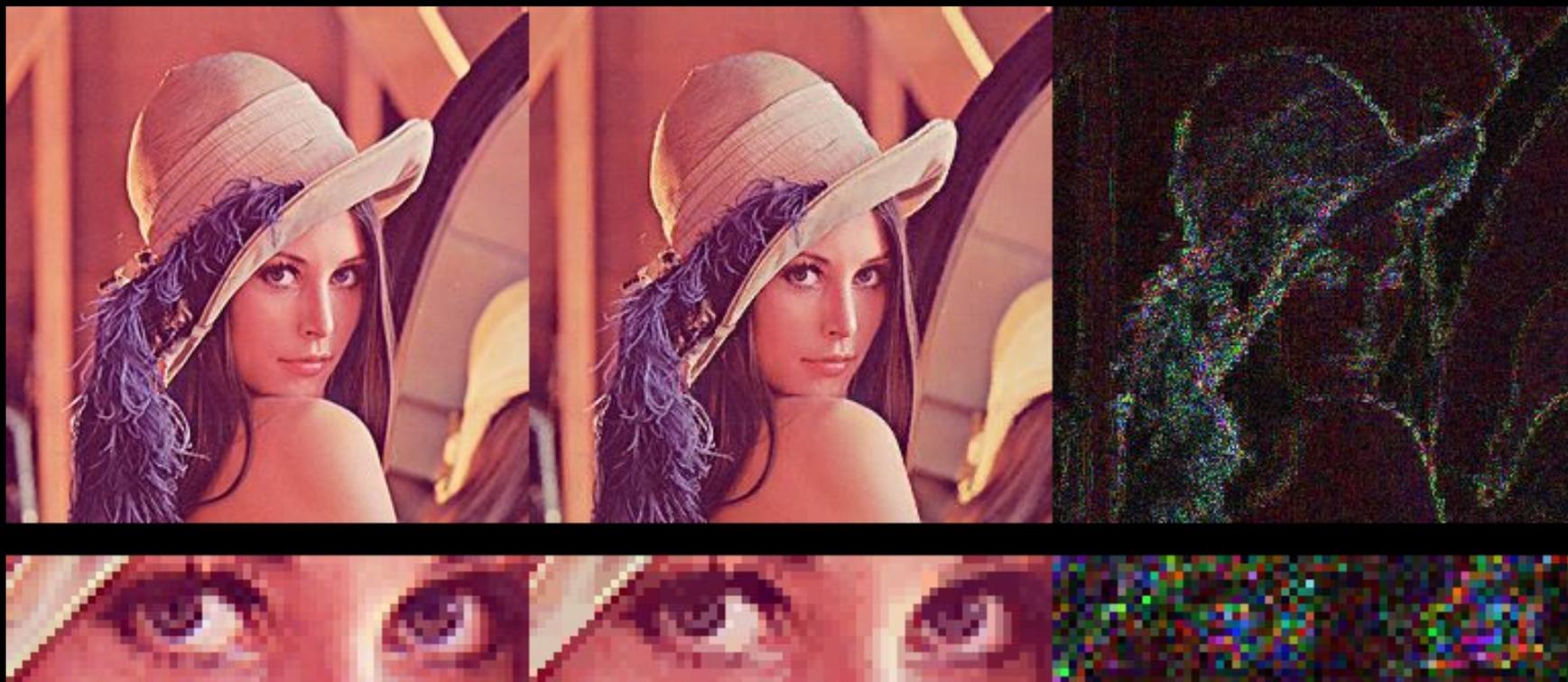
Оптимизация: Pixel shader

- Early Z-cull
 - Рендеринг в порядке возрастания глубины
 - Требуется Z-prepass
- Рендеринг в меньшее разрешение
 - Проблема «краев»
- Reprojection
 - Ghosting
- Tiled techniques

Оптимизация: DXT/BC сжатие

- Уменьшение размеров текстур (в 2 раза)
 - Лучше жать текстуры цвета чем карты нормалей
- BC/DXT-сжатие
 - Экономия памяти
 - Ускорение рендеринга
 - Блоки 4x4:
 - 2 цвета □ 5.6.5
 - 2 бита – параметр интерполяции
- DXT/BC
 - Текстуры поверхностей
 - Фотографии
 - Карты нормалей (нерегулярные)
 - Карты нормалей (регулярные)
 - Диаграммы, схемы
 - Шрифты
- P.S. Нужен контроль!

Оптимизация: DXT/VC сжатие



192 Kb

32 Kb

Diff: x8

Оптимизация: Сжатие вершин

	Атрибут вершины	Наивный подход	Простое сжатие	Сжатие с распаковкой в шейдере
1	Position	float3	float3	half4 + quat sign
2	Normal	float3	half4	byte4 (quat)
3	Tangent	float3	half4 (xyz)	
4	Binormal	float3	half4 (xyz)	
5	Texture Coord	float2	[3].w [4].w	half2
6	Color	float4	byte4	byte4
7	<i>Blend Weights</i>	<i>float4</i>	<i>half4</i>	<i>half4</i>
8	<i>Blend Indices</i>	<i>byte4</i>	<i>byte4</i>	<i>byte4</i>
	Skinned	92 bytes	52 bytes	32 bytes
	Rigid	72 bytes	40 bytes	20 bytes

Вопросы?

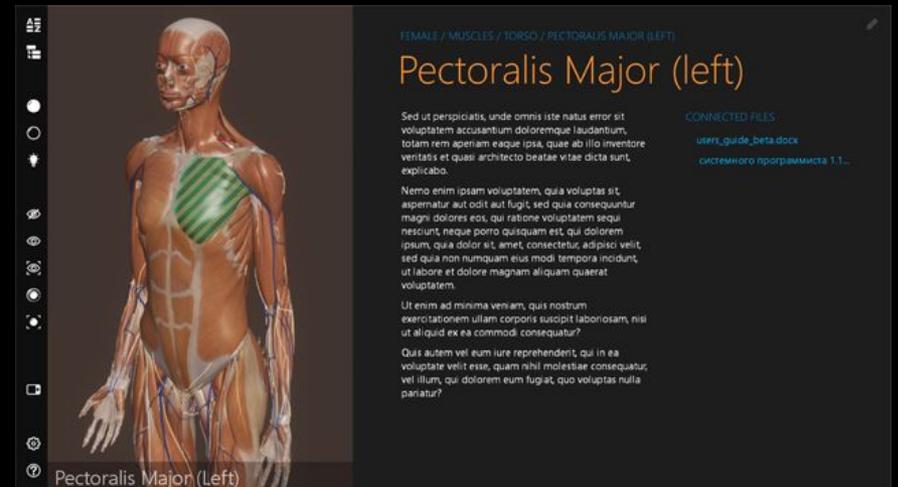
Вопросы online

- Почему в OpenGL нет моделей освещения?
- Связана ли дисперсия с пинк фloydом?) [Y/N]
- Можно ли варить современный графический движок в кладовке или лоджии? Легально ли это?
- Чем новые API (Mantle, DX12, Apple Metal и тд) отличаются от старых?
- Используются ли модели глобального освещения в играх на данный момент?
- Насколько можно быстро сварить простой движок для 3D визуализации дилетанту, который в последний раз графику программировал для режима 320x200x256 в ассемблере через сегмент A000 (это было в "лохматом" 1996 году)))))), а в этом году баловался Unity и SandBox?
- Как расположить span по центру div (и вертикально и горизонтально) не используя абсолютного позиционирования?
- Можно ли научиться варить движок, если в программировании почти ноль (0.1)?

Вопросы online

- Насколько целесообразно создавать собственный движок с нуля? И с чем может быть связана такая потребность?
- Почему в современных графических движках так усердно стараются наращивать количество полигонов в сцене, при этом отводя на задний план качество текстур, освещение и тени? Ведь из-за этого во многих современных играх наблюдается некая "мультишность" изображения.
- Что вы думаете о перспективах алгоритмов unbiased rendering в играх?
- Что там насчет массовой применимости на сегодня и ближайшее будущее у техник, альтернативных растеризации треугольничков? Воксели, рейтрейс – вот это вот все.
- Скорость распространения света в первой среде 225 000 км/с, а во второй – 200 000 км/с. Луч света падает на поверхность раздела этих сред под углом 30° и переходит во вторую среду. Чему равен угол преломления луча?
- Насколько сильно различаются API графики для PC и консолей? Различаются ли принципиально методики разработки и общий цикл?
- Что вы думаете о будущем использования мегатекстур (как в игре Rage)?
- Чем отличается тесселяция от триангуляции?

Проекты НИИ НКТ НИУ ИТМО



Кафедра

- **День открытых дверей:**
 - 8 июля в 18:00
 - Биржевая линия, д. 4
- **Двойной диплом**
 - Университет
г. Амстердам,
Нидерланды



Программы и преимущества

- Суперкомпьютерные технологии в междисциплинарных исследованиях
- Суперкомпьютерные технологии в исследовании процессов большого города
- Экстренные вычисления и обработка сверхбольших объемов данных
- Возможность совмещать учебу и работу
 - 25 т.р. за 1/2
- Возможность обучения за рубежом
- Автоматическое поступление в аспирантуру