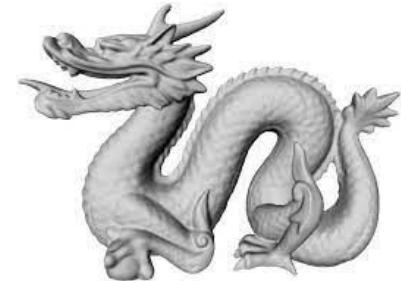
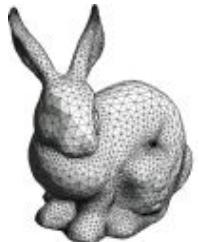


Вся 3D графика

за 25 минут



Почему 3D графика?

- Несколько фриланс проектов
- Развивается вместе с нами
- Красивое

1960

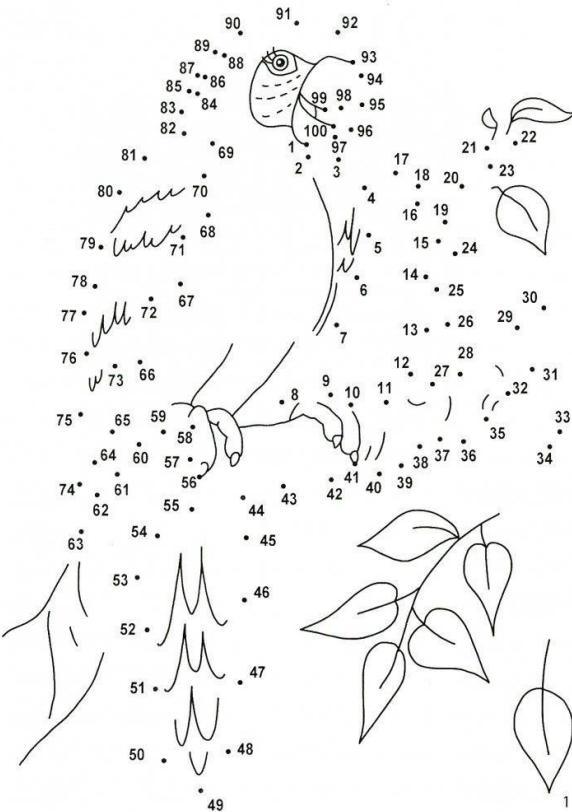


2022

Николай Воронин

Старший разработчик C#

Altenar



Форма

- Рисуем и храним внешнюю оболочку
- Экономим место и время

Для объема

- Используем проекции
- Свет и тень

Превосходство треугольников

- Три точки всегда в одной плоскости
- Всего три вершины. Экономим
- Можно сложить любую фигуру,
лишь бы хватило треугольников →
- Красивое
low-poly направление

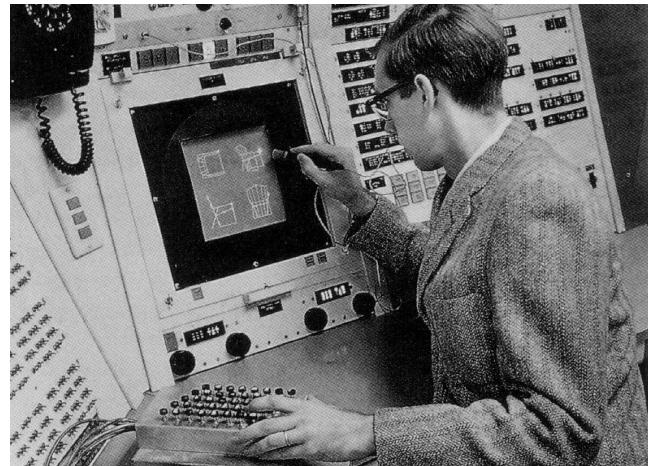


Векторная графика

- Архитектура
- Машиностроение
- Медицина
- Аркадные автоматы

Проблемы

- Мало памяти
- Медленный процессор
- Больше цвета!



Текстура

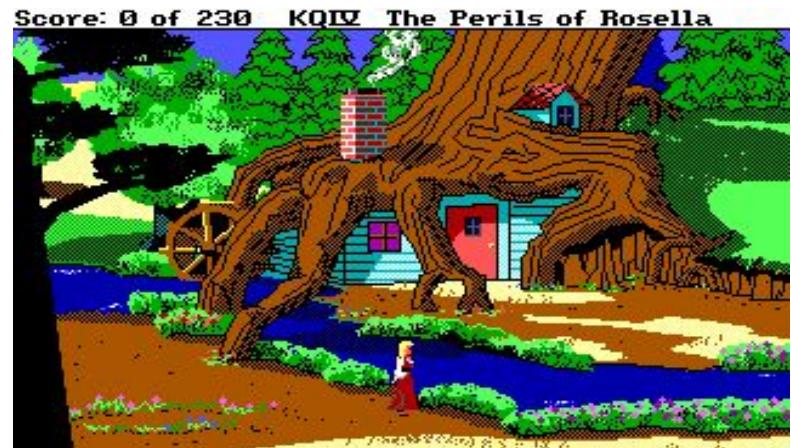
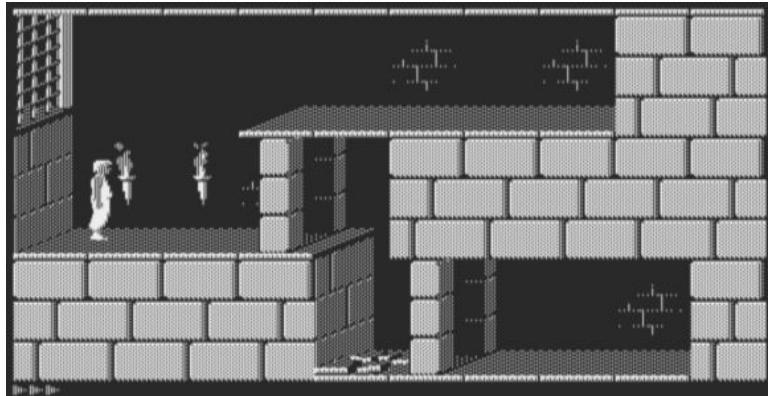
- Заполняем контуры
 - Линии грубые
 - Треугольники скучные
-
- Точки более точные
 - Текстура и фактура



Поль Синьяк. Большой канал в Венеции. 1905 год

Цвет

- 4
- 8
- 16
- 256
- ...
- 16 млн
- 4 млрд
- HDR

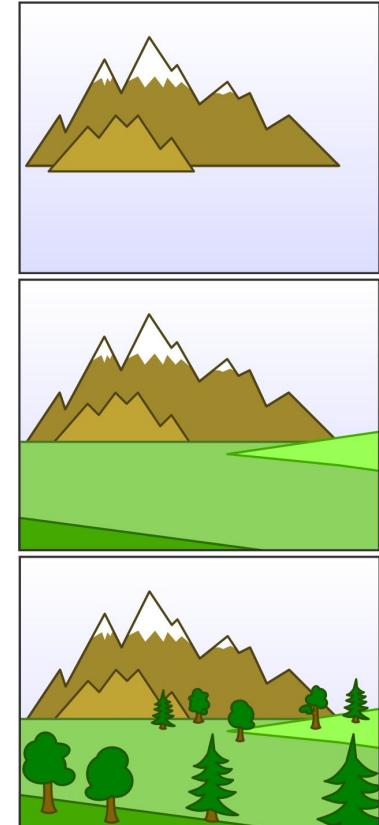


Алгоритм художника

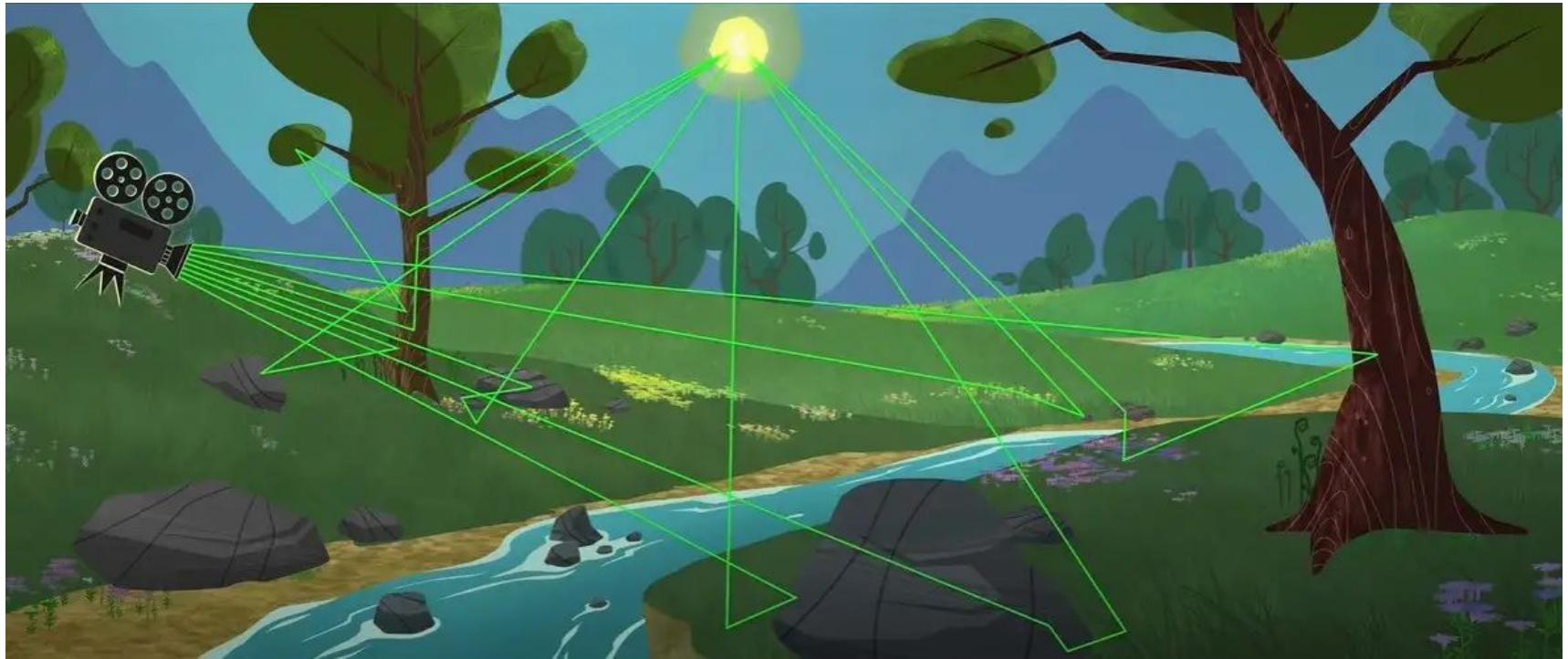
1. Рисуем дальний план
2. Рисуем план поближе
3. ???
4. PROFIT

Проблемы

- Рисует центральный процессор
- Медленное всё, памяти мало
- Рисовать нужное, а ненужное - не рисовать
- В какой цвет красить пиксель?



Свет + камера + треугольники → пиксели

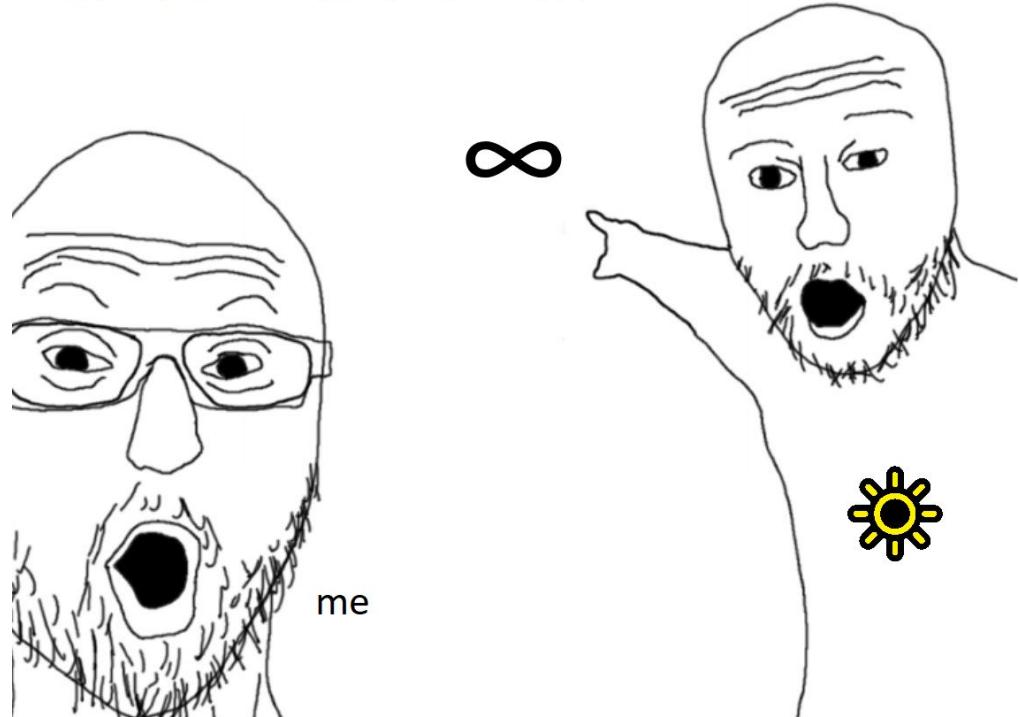


Свет + камера + треугольники → пиксели



$$L(P, \omega_o) = L_e + \int_{\Omega} f(P, \omega_i, \omega_o) L_i(P, \omega_i) (\omega_i \cdot \hat{n}_P)$$

- **Unbiased**
 - долго, дорого, ох красиво!
- **Biased**
 - быстро, играем!
- Иоганн Генрих Ламберт
1760 год



Нужно больше

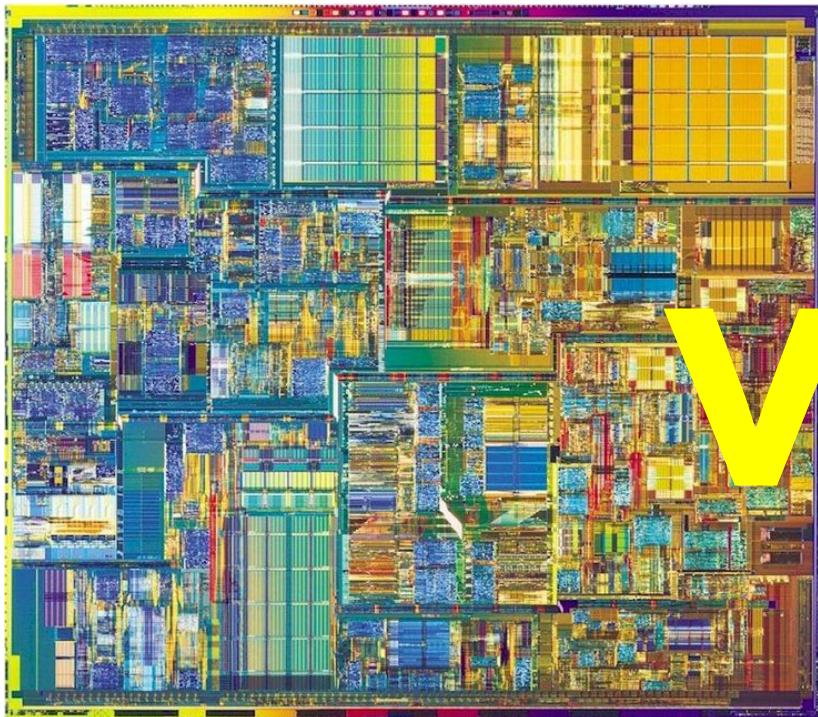
- Грубые модели
- Текстурированные поверхности
- Мало памяти
- Медленный процессор

Решение

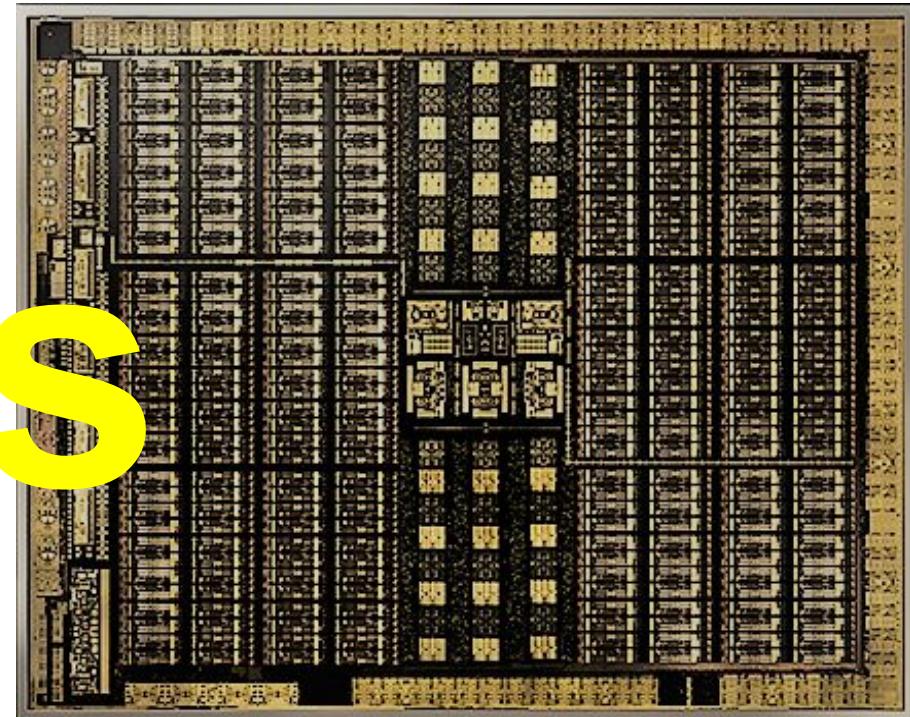
- Покупаем видеокарту



Процессор 1..64

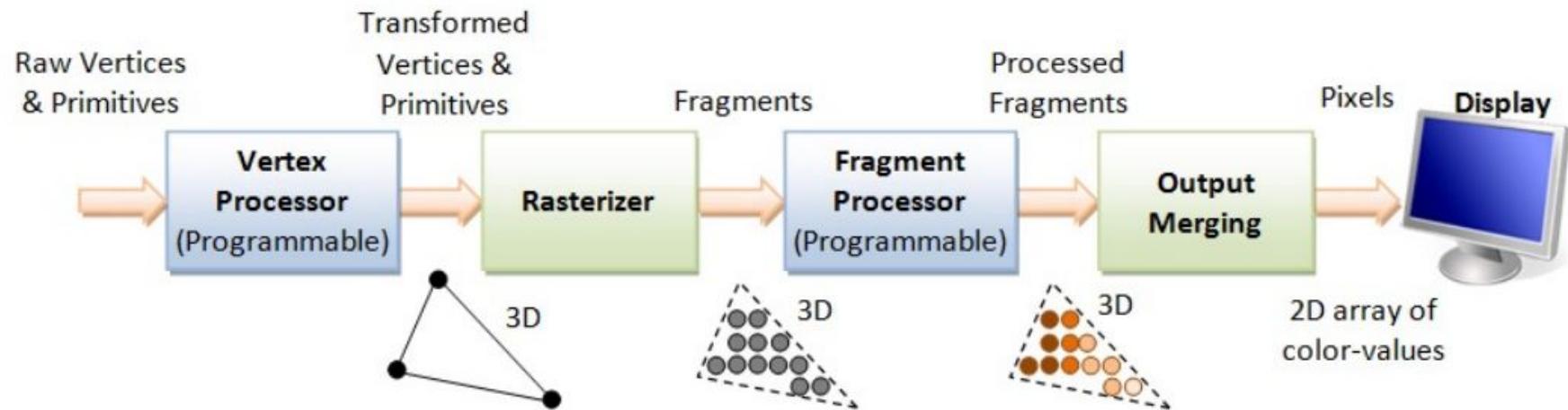


Видеокарта ..20 000



vs

Графический конвейер



Шейдеры

HLSL, GLSL, OpenCL, SPIR-V, ...

C99 на стероидах

- Вершинный или вертексный
Что делать с формой?
- Пиксельный или фрагментный
Каким цветом красить пиксель? →
- А еще: тесселяционный, геометрический, вычислительный, рейтрейсинг, примитиво-



ShaderToy, Inigo Quilez

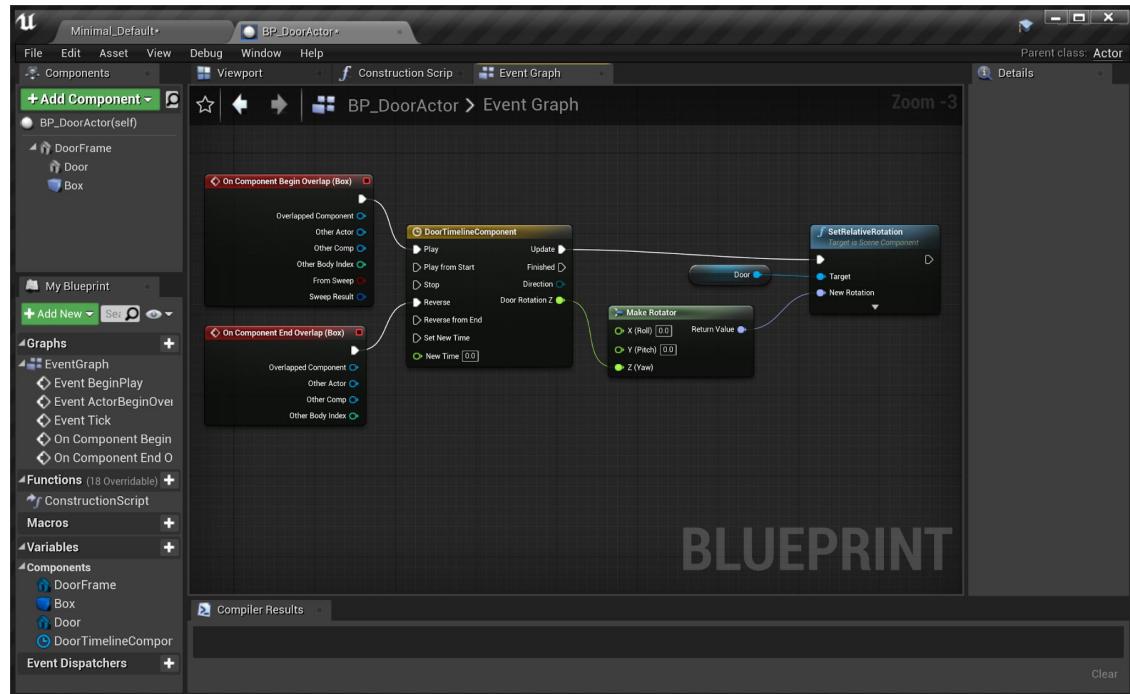
Кронос уже не тот

- OpenGL, Khronos Group 1992 > 30 лет назад
- VRML, ISO 1994
- Glide, 3Dfx 1996
- Direct3D, Microsoft 1996 > 25 лет назад
- OpenGL ES, Khronos 2003
- CUDA, NVIDIA 2007
- OpenCL, Khronos 2009
- OptiX, NVIDIA 2009
- WebGL, Khronos 2011
- Vulkan, AMD + Khronos 2016



IDE + фреймворк

- **Unreal**, Pro / AAA
- **Unity**, Indie / Mobile
- **Godot**, Indie
- Blender
 - Three.js
 - Babylon.js



2D = 3D - 1

• Windows Desktop	DirectX	Я не играю.
• macOS Desktop	Metal	Зачем мне новая видеокарта?
• Linux Desktop	OpenGL, OpenCL	
• Android	OpenGL ES	
• Chrome Browser	Skia	
• Photoshop	OpenGL, D3D, OpenCL, CUDA	Когда не хватает встроенной в процессор.

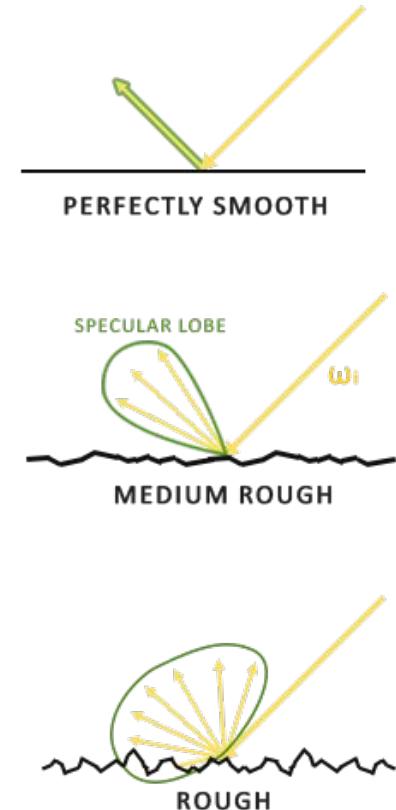
Решения

- **GPU Scheduler**
распределение ресурсов видеокарты

Физически точный рендеринг

PBR, Physically Based Rendering

- Поверхность фактурная. Микрорельеф
 - Фотореалистичные мегатекстуры
 - Фотограмметрия
-
- “Запекаем” свойства материала в текстуры
 - Шейдеры тесселяции поверхности



Стриминг

- Много данных
- Память опять закончилась

Решения

- Все храним на жестком диске
- Загружаем во время работы на лету
- Быстрые SSD диски
- Быстрые каналы памяти
- Прямая связь видеокарты с данными



Marvel's Spider-Man
A Technical Postmortem

Фотореалистичность

Epic Unreal Engine 5
Nanite

Решение



- Детализирует нужное
- а ненужное - упрощает

Unreal 5

Lumen



Matrix Awakens →



Рейтинг возвращается

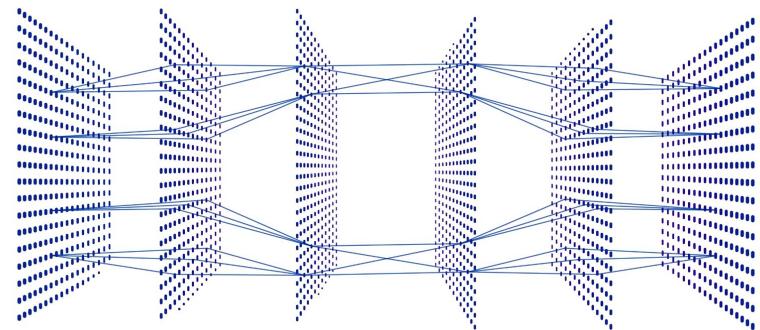
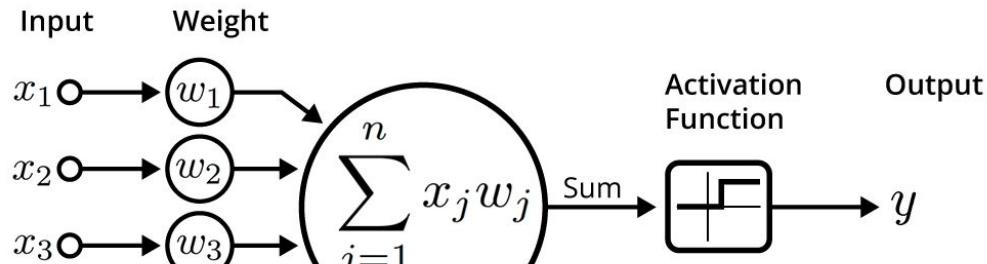
NVIDIA RTX 20..

2018 год



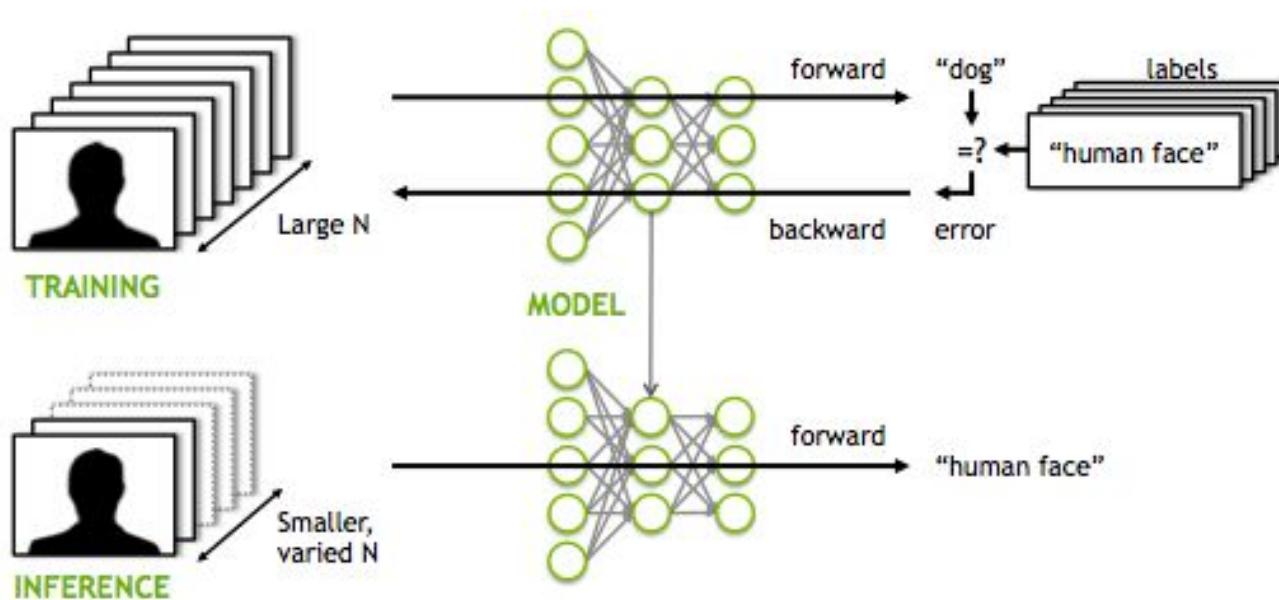
- Улучшена модель освещения

Нейронные сети за 30 секунд



$$\left\{ \begin{array}{l} 2x + 1 = 0, \\ 4x + y - 5 = 0. \end{array} \right.$$

Курсы для нейросетей за 30 секунд



Lensa

256 Nvidia A100 GPU

150 000 GPU-часов

40 000 000 ₽

CUDA

≥ 6 M6 VRAM

Python

70 000 ₽

DLSS, Deep Learning Super Sampling

- Предобучение на образцах изображений
- Считаем мало → рисуем много
- Достраивает целые кадры: 15 fps → 60 fps

Решения

- Копия больше и лучше оригинала
- Удаляет ненужное: фон, шум, целые объекты
- Восстанавливает “слабые” источники: видео, звук, ...

Вся 3D графика за 25 минут

Николай Воронин

GitHub: nikvoronin

Twitter: см. гитхаб

AAA Pro: Unreal 5

Indie/Mobile: Godot, Unity

3D Design: Blender

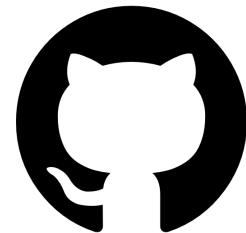
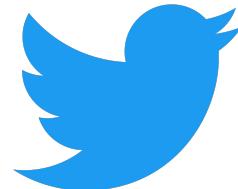
Shaders: ShaderToy

Low-level: Veldrid

2D C#: SkiaSharp, ImageSharp

3D C#: OpenTK

OpenCL C#: Cloo



Не воксели

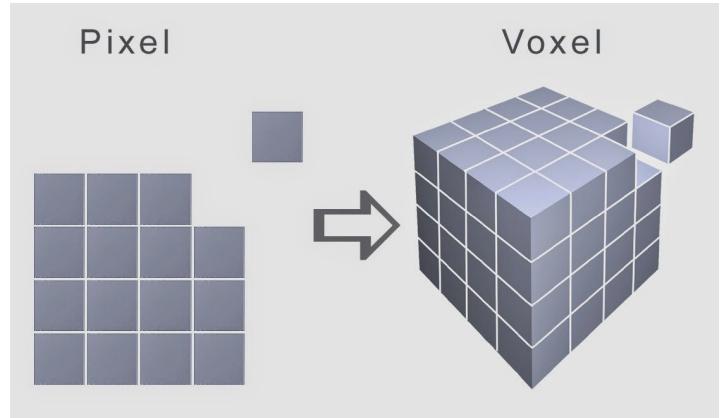
- Только данные
- Стилизация графики



Воксели

- Могут быть любой формы
- Занимают фиксированное место

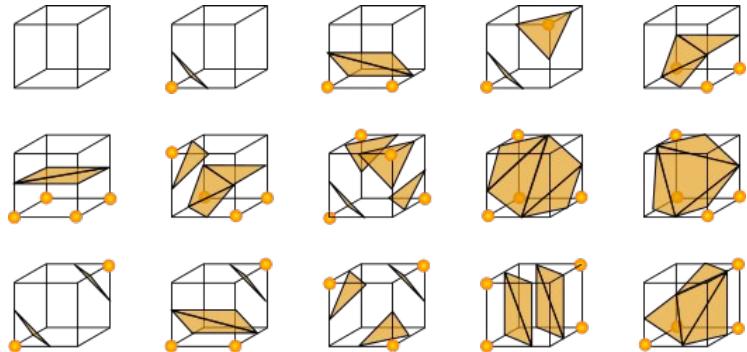
Нет вращения или сдвига



Марширующие кубики

Marching cubes, 1987 г.

Воксели требуют куб памяти



- **Фреймворк:** Voxel Farm →
- **Игры:** Voxelstein 3D, Voxatron
- **Дизайн:** MagicaVoxel



MagicaVoxel

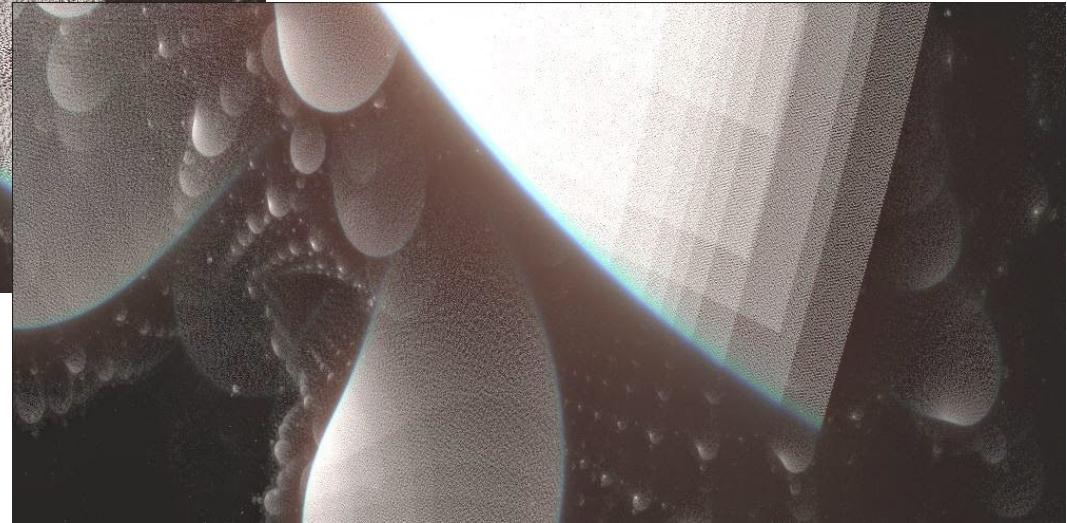
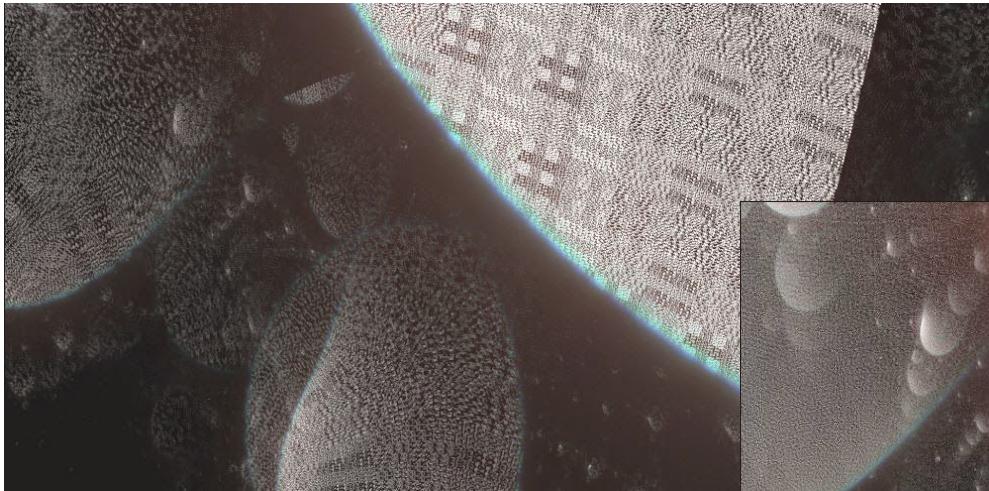
Как пиксель-арт только в 3D

Картинку можно “покрутить”

Свет - “поставить”



/не/ Случайный выбор и фильтрация



SDF, Signed Distance Fields

Объект как мат. функция расстояния до него
или внутри него

- Визуализация математических объектов
Фракталы (Doctor Strange, 2016)
- Дешевые тени
- Масштабирование объектов без потери качества. Шрифты
- Гладкое соединение объектов

