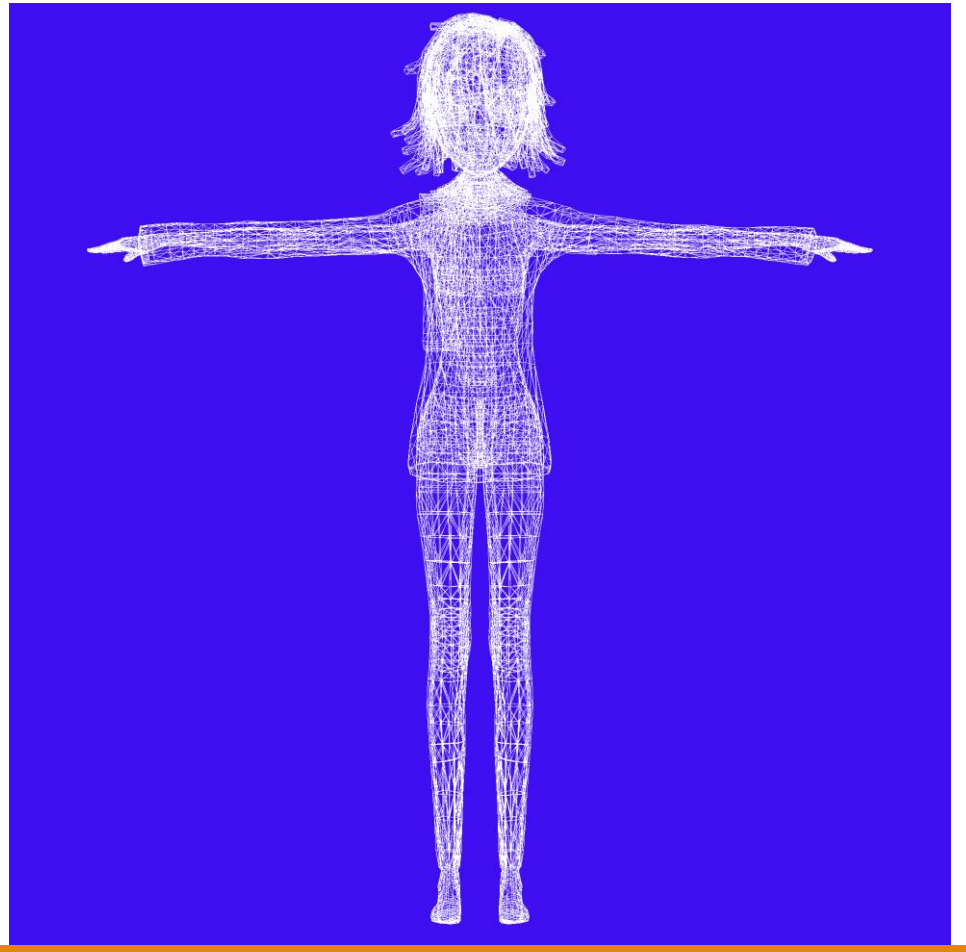
Компьютерная графика

ЛЕКЦИЯ 2.

РАСТЕРИЗАЦИЯ ТРЕУГОЛЬНИКОВ.

Растеризация треугольников

В подавляющем большинстве трёхмерные векторные модели состоят из треугольников, поэтому растеризация треугольников является ключевой составляющей рендеринга.



Что на входе?

Двумерные проекции вершин треугольника на поверхность изображения:

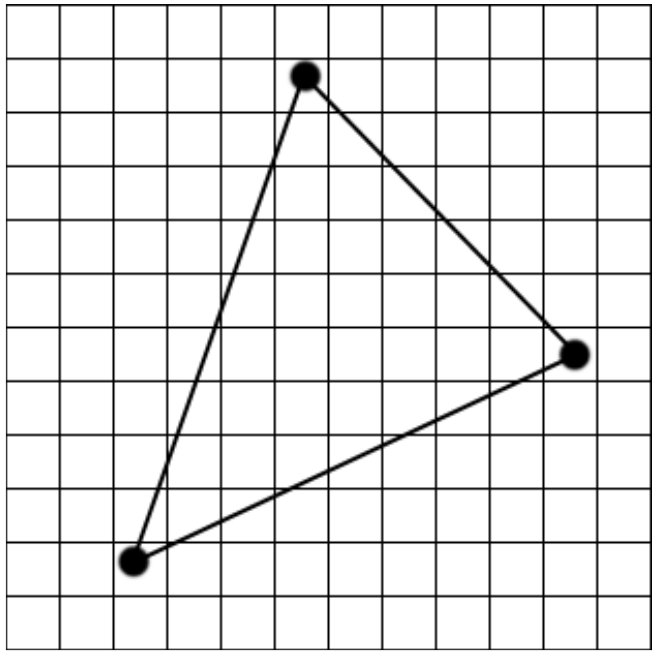
$(X_0, Y_0, Z_0) \rightarrow (x_0, y_0)$

$(X_1, Y_1, Z_1) \rightarrow (x_1, y_1)$

$(X_1, Y_1, Z_1) \rightarrow (x_1, y_1)$

Точки – вещественные!

Приводить к целым пикселям **не нужно**!!!

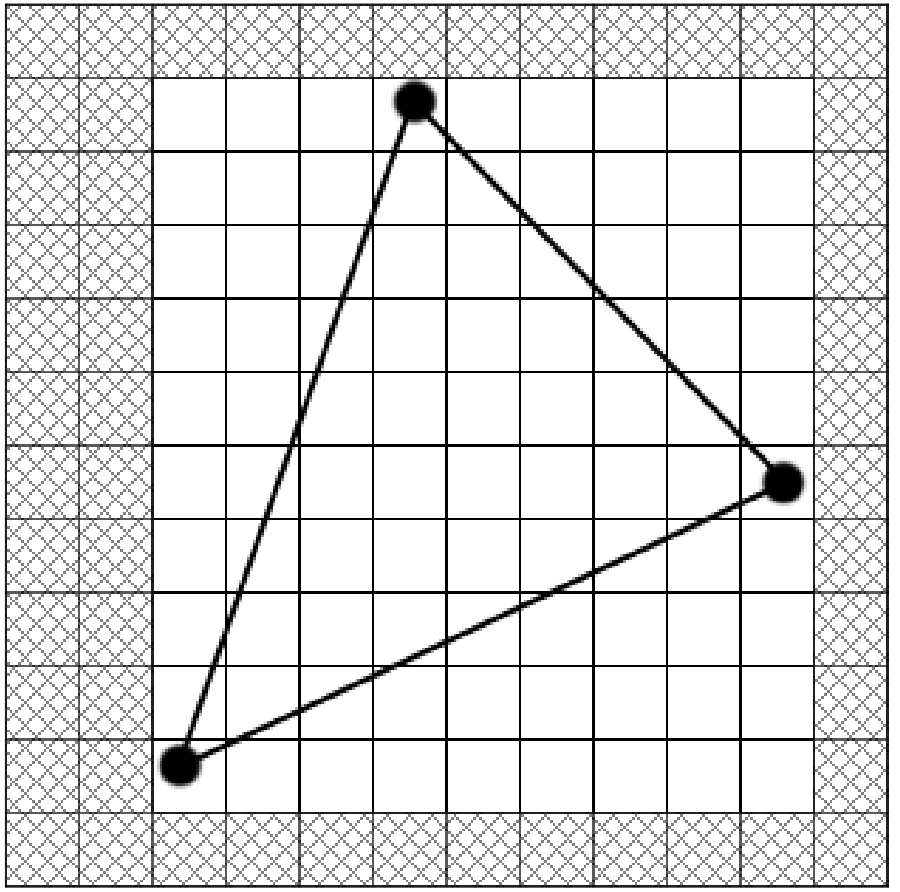


Шаг 1. Определение ограничивающего прямоугольника

Определяем минимальную и максимальную координату треугольника по каждой оси в пикселях.

$(5.5, 1.2), (2.4. 10.3), (10.6, 6.7) \rightarrow$
$r_{min} = 1, r_{max} = 10,\ c_{min} = 2,$
$c_{max} = 10.$

Номера строк и столбцов – **целочисленные**. Это пиксели и они будут использоваться в цикле.



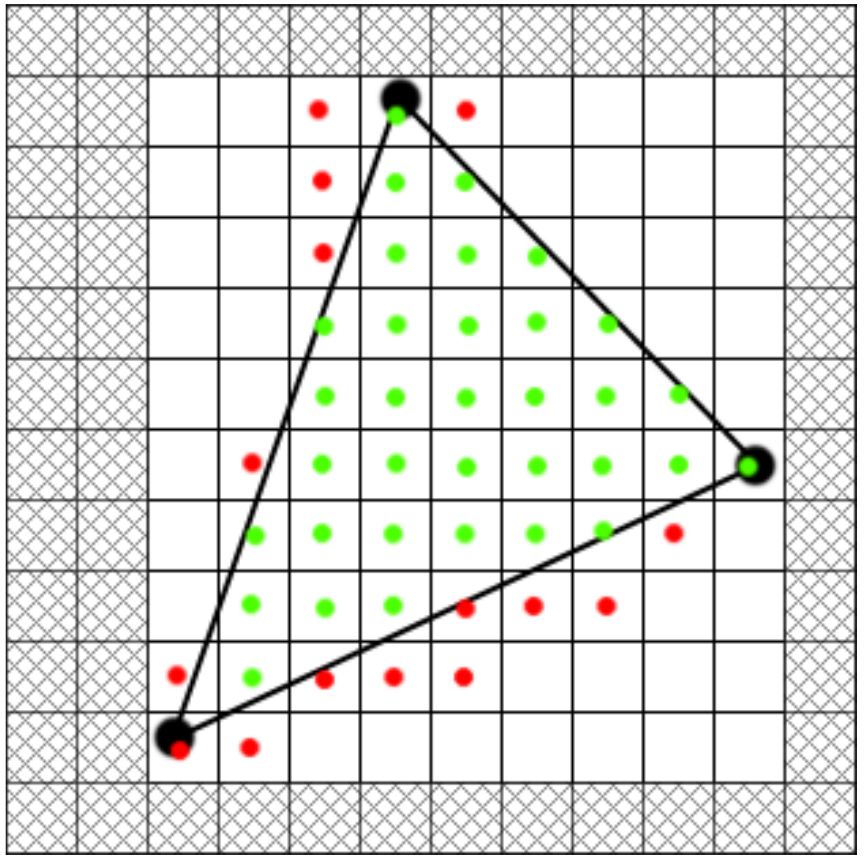
Шаг 2. Определение точек, находящихся внутри треугольника

В цикле по всем точкам внутри ограничивающего прямоугольника проверяем на вхождение точки в треугольник.

Несмотря на то, что цикл проходит по целочисленным значениям пикселей, проверяется точка с **вещественными** координатами, соответствующая центру пикселя относительно **вещественных** координат вершин треугольника.

Пусть $(5.5, 1.2), (2.4. 10.3), (10.6, 6.7)$ – координаты вершин и текущий пиксель – (5,7).

Проверка делается для точки с координатами $(5.0, 7.0)$ относительно треугольника с вершинами $(5.5, 1.2), (2.4. 10.3), (10.6, 6.7)$!



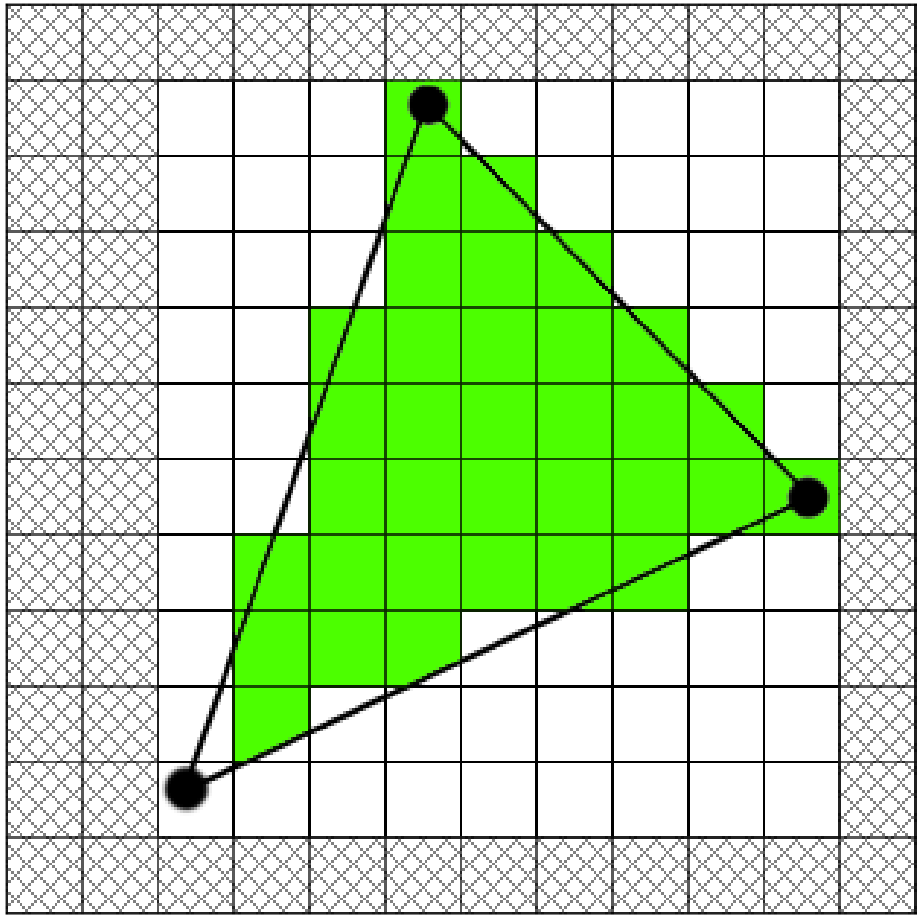
Отрисовка точек

Заполнение соответствующих элементов изображения выбранным цветом.

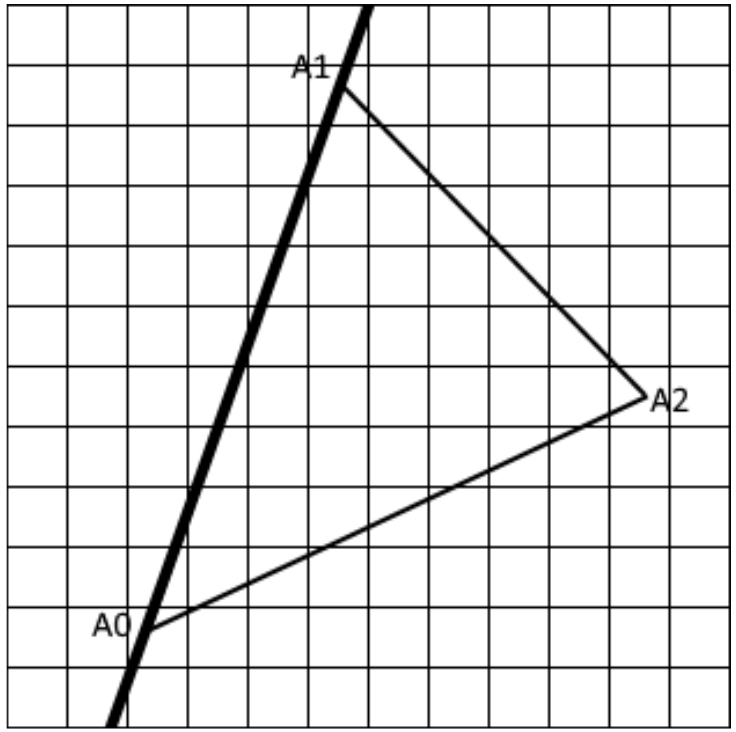
Пусть $(5.5, 1.2), (2.4. 10.3), (10.6, 6.7)$ – координаты вершин и текущий пиксель – (5,7).

Проверка точки с координатами $(5.0, 7.0)$ относительно треугольника с вершинами $(5.5, 1.2), (2.4. 10.3), (10.6, 6.7)$ показала, что эта точка входит в треугольник.

Следовательно, пиксель (5,7) должен быть закрашен.



Как определить внутренние точки?



Даны две точки $A_0(x_0, y_0)$ и $A_1(x_1, y_1)$.

Уравнение прямой, проходящей через эти две точки:

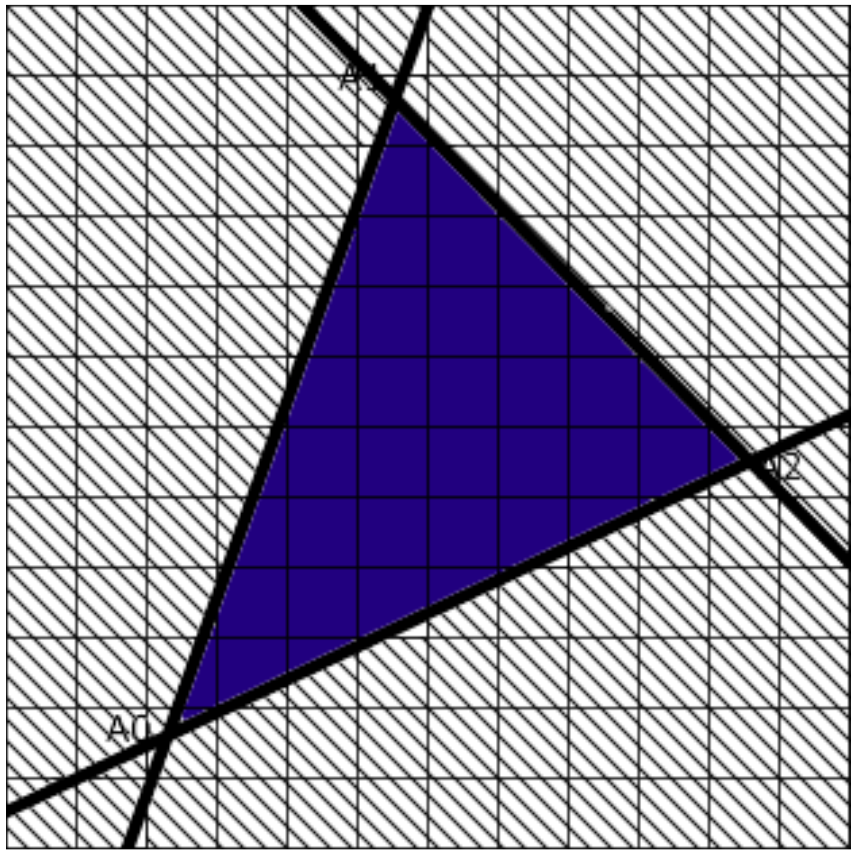
$$\frac{x-x_0}{x_1-x_0} = \frac{y-y_0}{y_1-y_0}$$

или

$$(x - x_0)(y_1 - y_0) - (x_1 - x_0)(y - y_0) = 0.$$

Соответственно, по разные стороны от этой прямой значения этой функции будут отрицательными и положительными.

Определение внутренних точек



Если построить три прямые, то для всех точек внутри треугольника значения всех трёх функций будут больше нуля:

$$f_{01}(x,y) = (x - x_0)(y_1 - y_0) - (x_1 - x_0)(y - y_0) > 0$$

$$f_{12}(x,y) = (x - x_0)(y_1 - y_0) - (x_1 - x_0)(y - y_0) > 0$$

$$f_{20}(x,y) = (x - x_0)(y_1 - y_0) - (x_1 - x_0)(y - y_0) > 0$$

для всех (x, y), лежащих внутри треугольника.

Недостаток предыдущего подхода

Недостатком предыдущего подхода является «бинарность» показателя: либо точка находится внутри треугольника, либо нет. В ряде задач хотелось бы помимо этого определить, где именно относительно вершин находится эта точка.

Пусть даны три вершины треугольника: $A_0(x_0, y_0)$, $A_1(x_1, y_1)$ и $A_2(x_2, y_2)$, не лежащие на одной прямой.

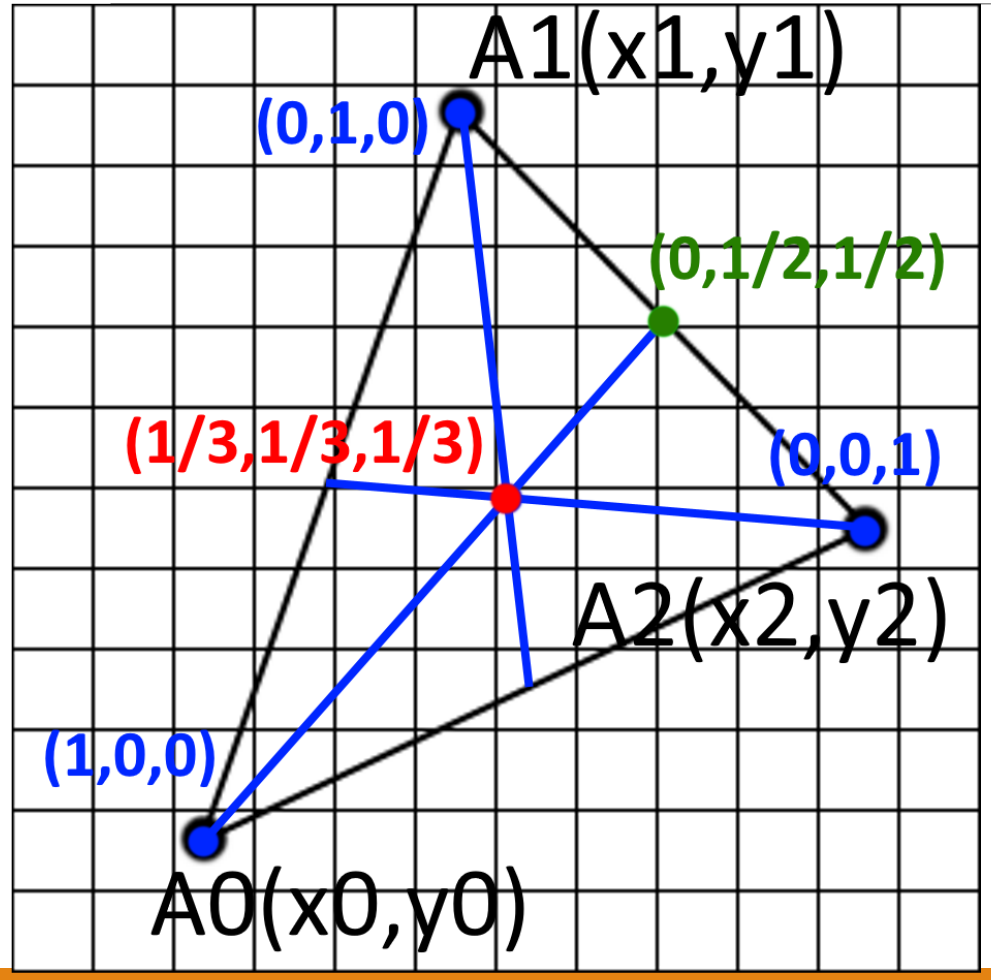
Для произвольной точки $P(x, y)$ нужно найти тройку чисел $(\lambda_0, \lambda_1, \lambda_2)$, которая будет описывать положение этой точки относительно вершин.

Для этого разложим координаты $P(x, y)$ следующим образом:

$$(x, y) = \lambda_0(x_0, y_0) + \lambda_1(x_1, y_1) + \lambda_2(x_2, y_2),$$

наложив ограничение $\lambda_0 + \lambda_1 + \lambda_2 = 1$

Барицентрические координаты



Полученные таким образом координаты $\lambda_0, \lambda_1, \lambda_2$ называются барицентрическими.

Очевидно, что:

координатам $(\lambda_0, \lambda_1, \lambda_2) = (1,0,0)$ соответствует точка A_0,

координатам $(\lambda_0, \lambda_1, \lambda_2) = (0,1,0)$ – точка A_1,

координатам $(\lambda_0, \lambda_1, \lambda_2) = (0,0,1)$ – точка A_2.

А у точек, находящихся внутри треугольника, **все три** барицентрические координаты больше нуля.

Формулы для вычисления барицентрических координат

$$\lambda_0 = \frac{(x-x_2)(y_1-y_2)-(x_1-x_2)(y-y_2)}{(x_0-x_2)(y_1-y_2)-(x_1-x_2)(y_0-y_2)}$$

$$\lambda_1 = \frac{(x_0-x_2)(y-y_2)-(x-x_2)(y_0-y_2)}{(y_0-y_2)(x_1-x_2)-(x_0-x_2)(y_1-y_2)}$$

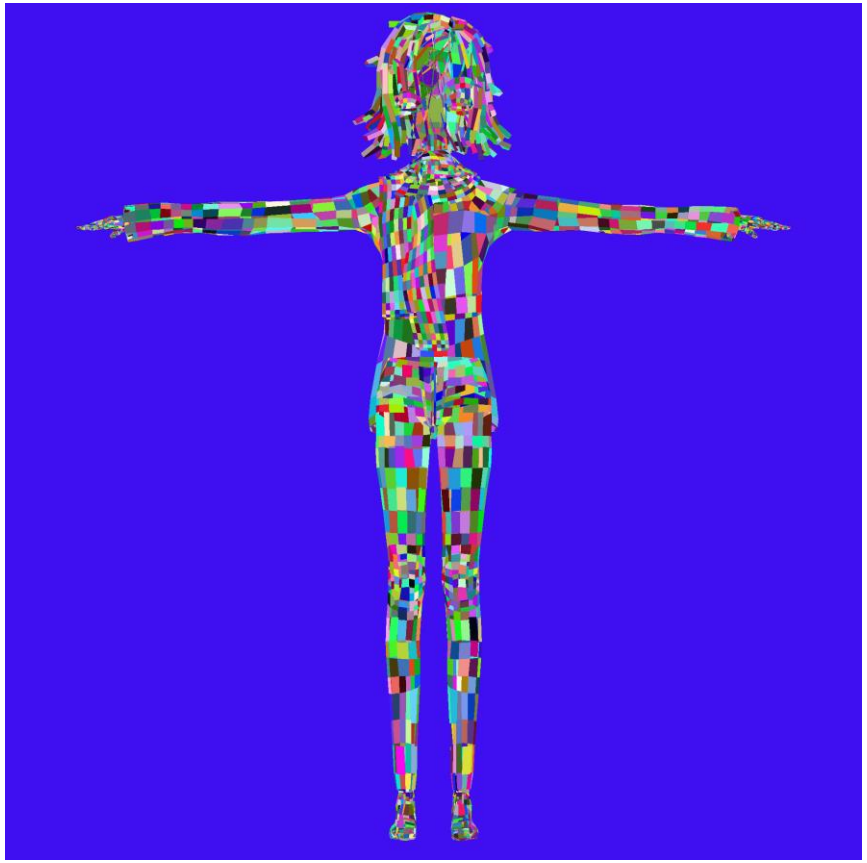
$$\lambda_2 = 1 - \lambda_0 - \lambda_1$$

Упрощённый конвейер рендеринга сцены

Для каждого треугольного полигона сцены:

- Рассчитать область интереса
- Для каждого пиксела внутри области интереса:
  - Рассчитать барицентрические координаты пиксела относительно вершин треугольника.
  - Если все барицентрические координаты больше или равны нулю:
    - Отрисовать пиксель

Результат отрисовки



Отбрасывание нелицевых граней (backface culling)

Простейший способ оценить необходимость отрисовки полигона проверить, направлен ли он к зрителю «лицевой» стороной.

Для того, чтобы определить «лицевую» сторону, можно рассчитать нормаль как векторное произведение двух рёбер полигона.

После этого скалярное произведение вектора нормали на направление камеры даёт косинус угла между ними. Грань видна, если косинус отрицателен.

Расчёт нормали к поверхности полигона

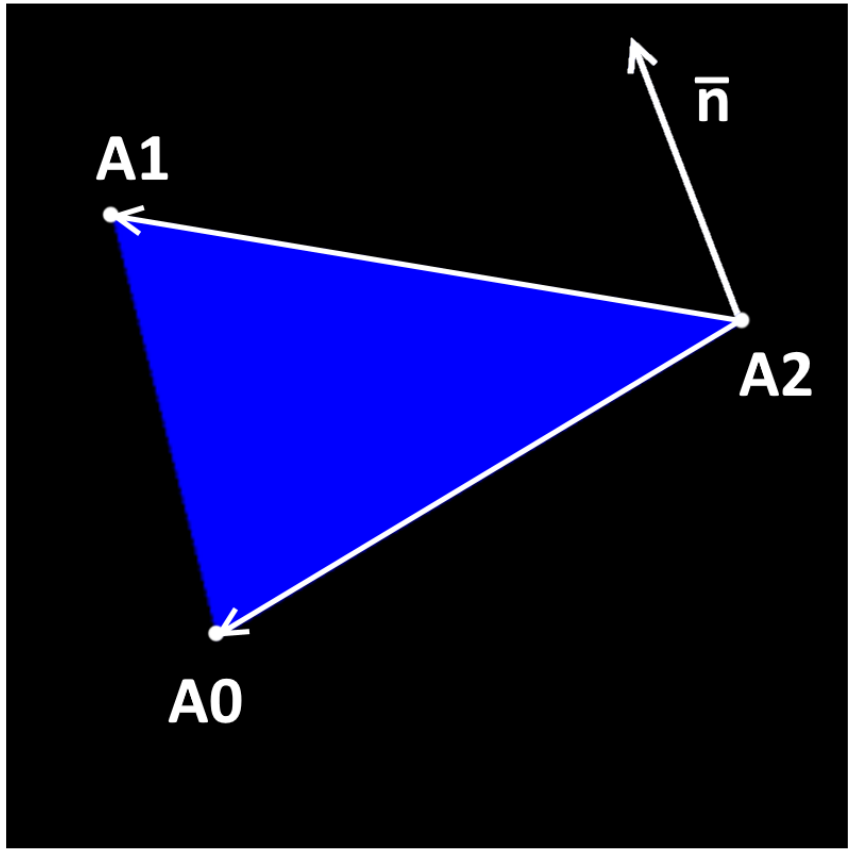
Нормаль:

$$n = (A_2 - A_0) \times (A_1 - A_0)$$

$$= \begin{vmatrix} i & j & k \\ x_2 - x_0 & y_2 - y_0 & z_2 - z_0 \\ x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \end{vmatrix}$$

Косинус угла:

$$\cos \alpha = \frac{\langle n, v \rangle}{\|n\| \|v\|}$$

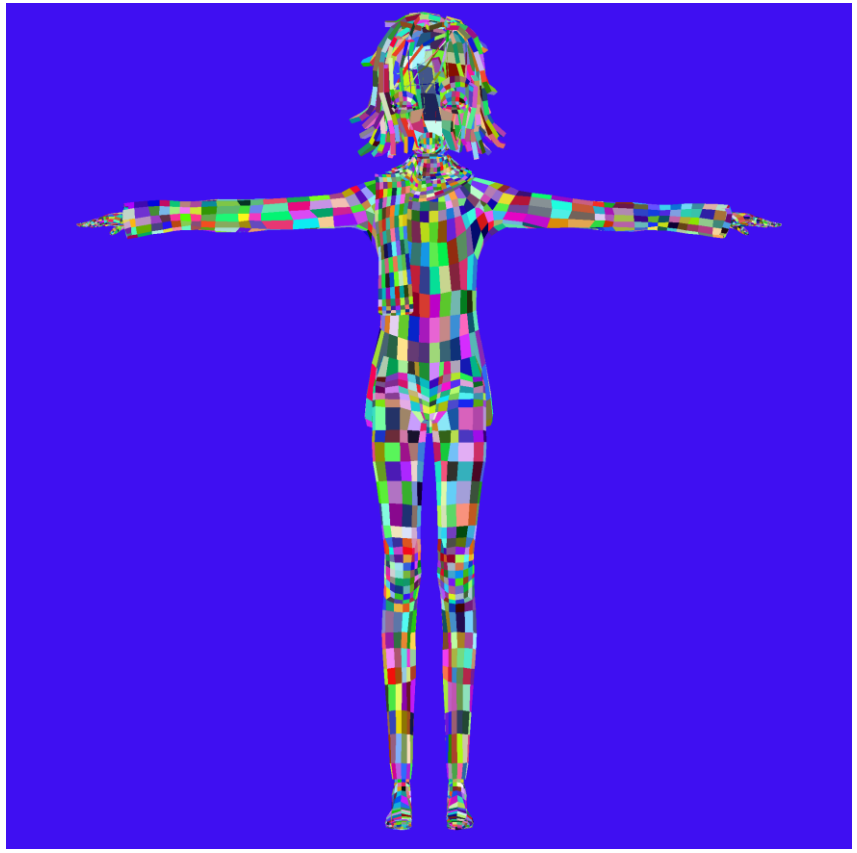


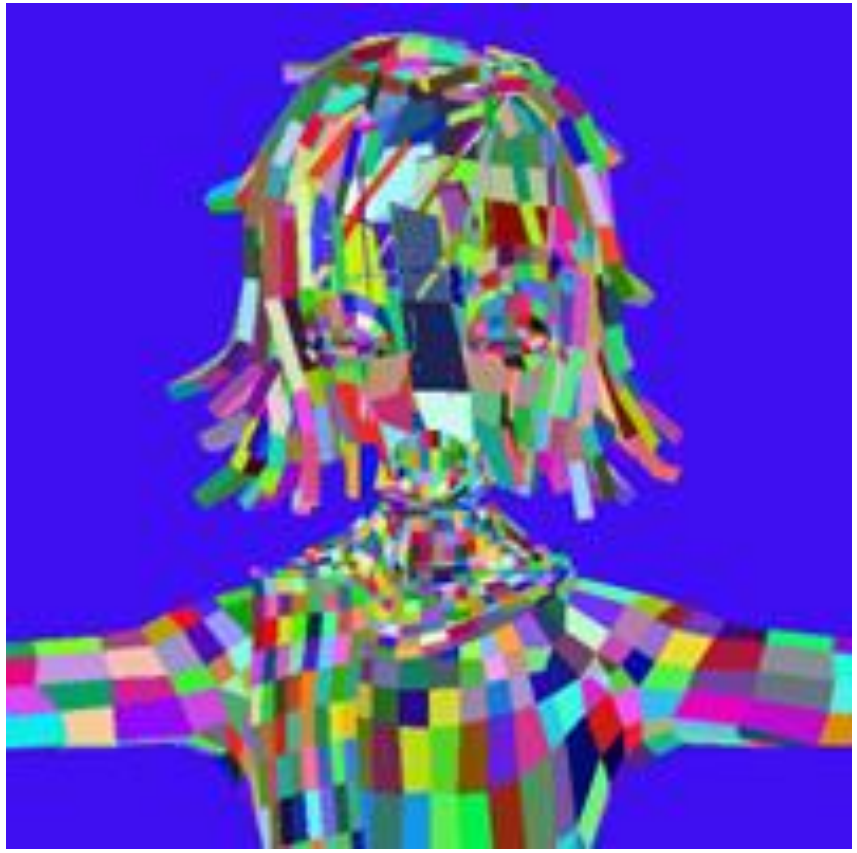
Упрощённый конвейер рендеринга сцены

Для каждого треугольного полигона сцены:

- Рассчитать нормаль к полигону.
- Если нормаль направлена под углом <90 к наблюдателю:
  - Перейти к следующему полигону
- Рассчитать область интереса
- Для каждого пиксела внутри области интереса:
  - Рассчитать барицентрические координаты пиксела относительно вершин треугольника.
  - Если все барицентрические координаты больше или равны нулю:
    - Отрисовать пиксель

Рендеринг случайными цветами



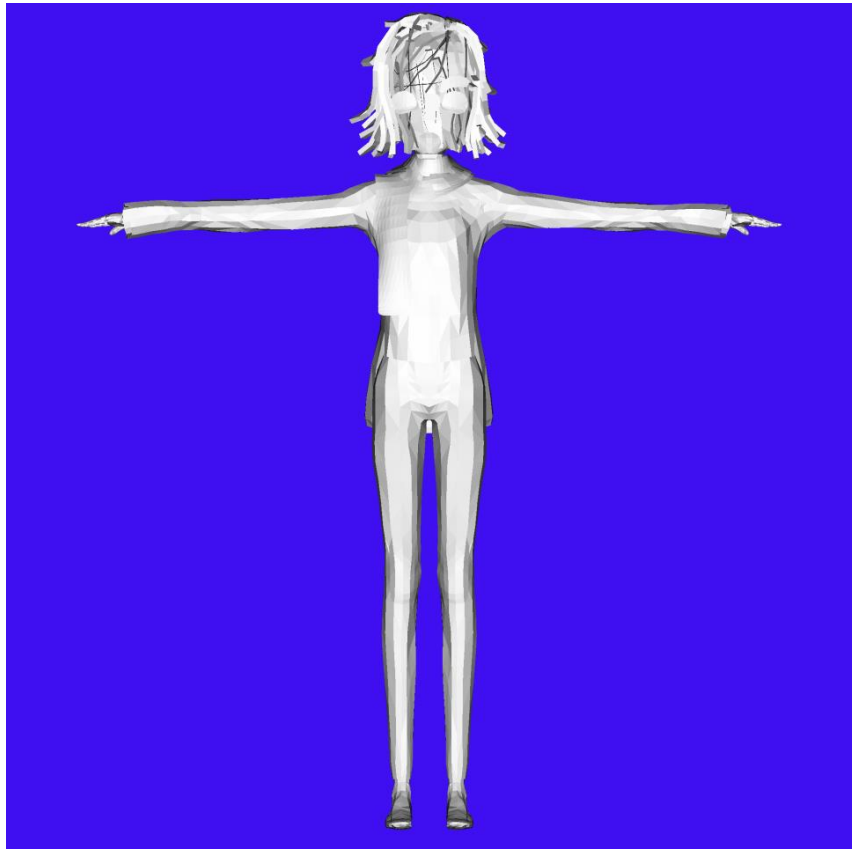


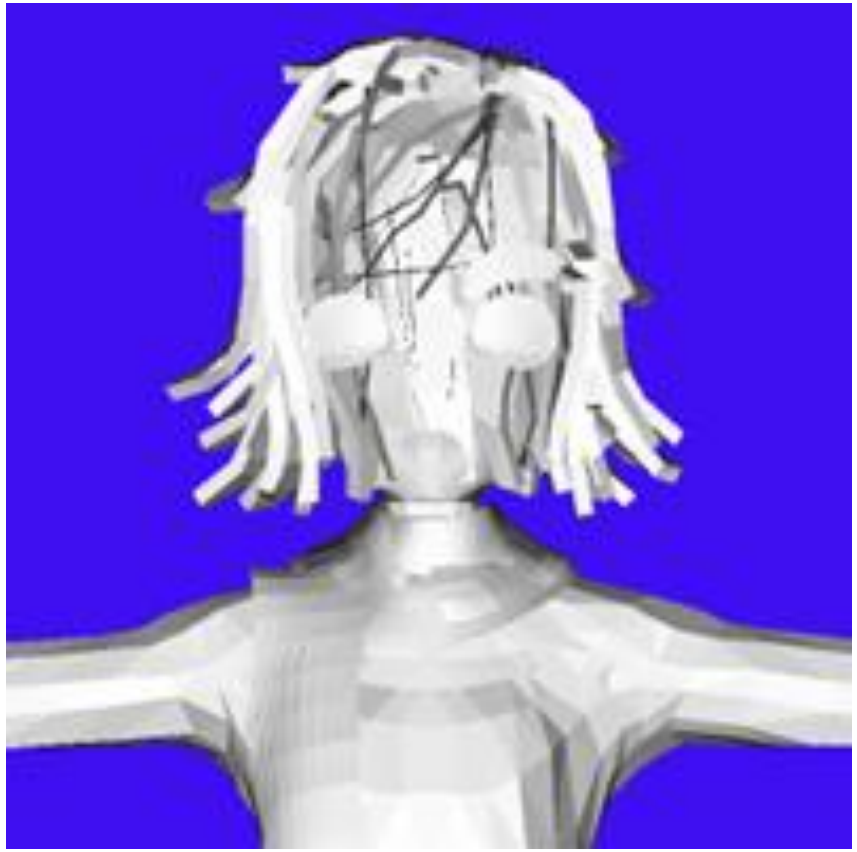
Упрощённый конвейер рендеринга сцены (без глубины и текстур)

Для каждого треугольного полигона сцены:

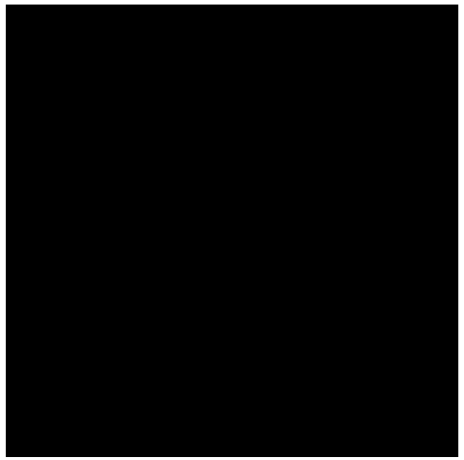
- Рассчитать нормаль к полигону.
- Если нормаль направлена под углом <90 к наблюдателю:
  - Перейти к следующему полигону
- Рассчитать область интереса
- Для каждого пиксела внутри области интереса:
  - Рассчитать барицентрические координаты пиксела относительно вершин треугольника.
  - Если все барицентрические координаты больше или равны нулю:
    - Отрисовать пиксель **значением интенсивности, пропорциональным углу к нормали**

Рендеринг с учётом освещения

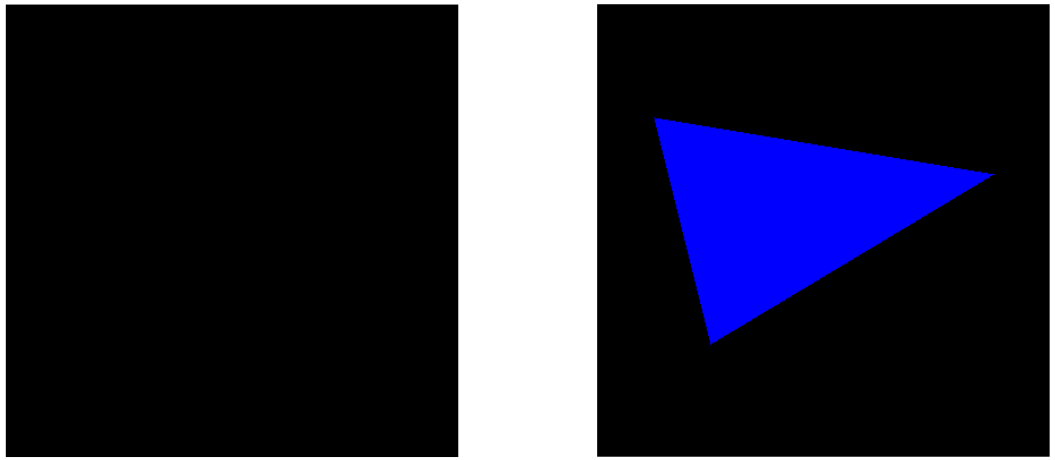




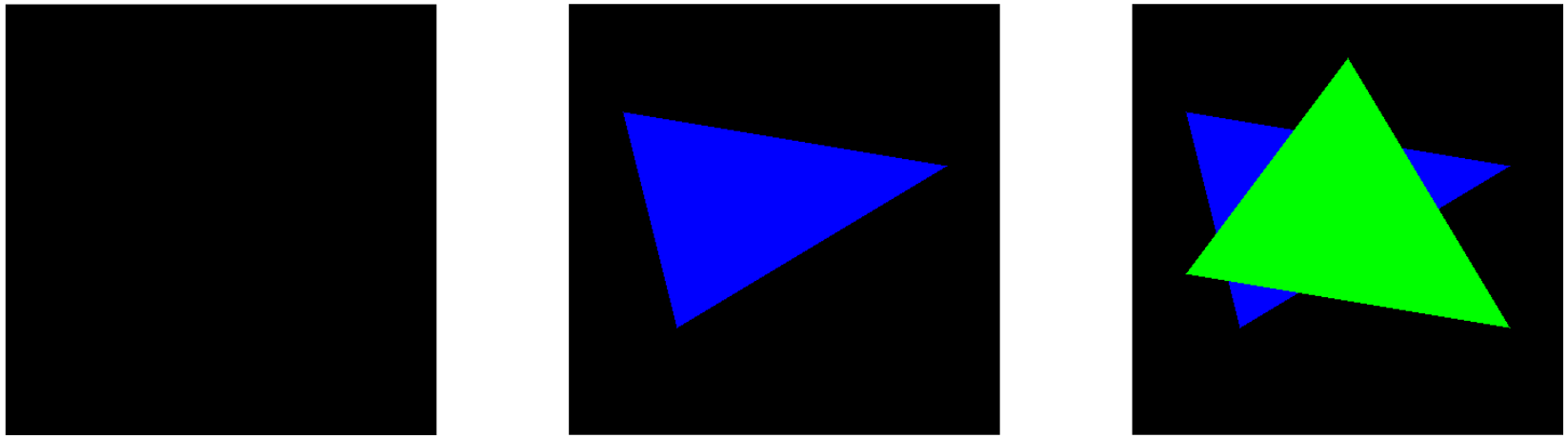
Отрисовка пересекающихся в пространстве треугольников



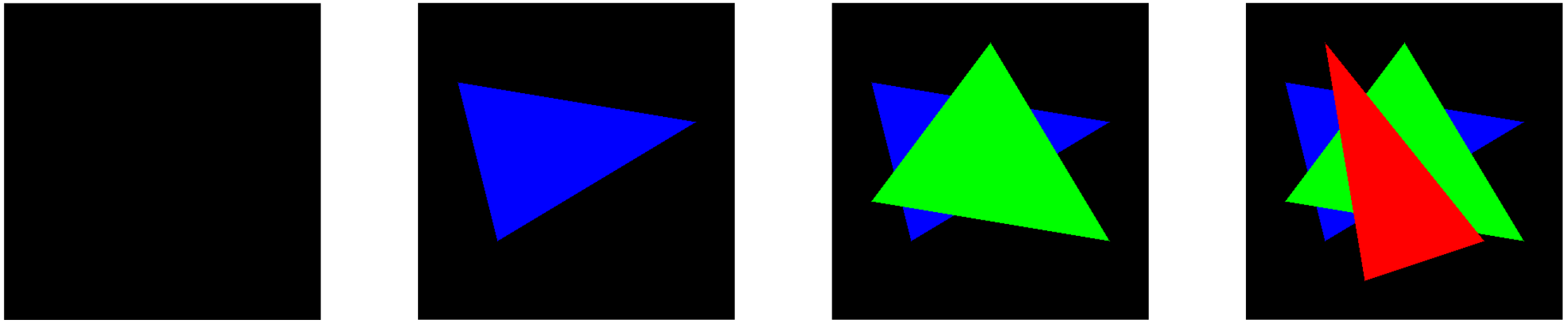
Отрисовка пересекающихся в пространстве треугольников



Отрисовка пересекающихся в пространстве треугольников



Отрисовка пересекающихся в пространстве треугольников



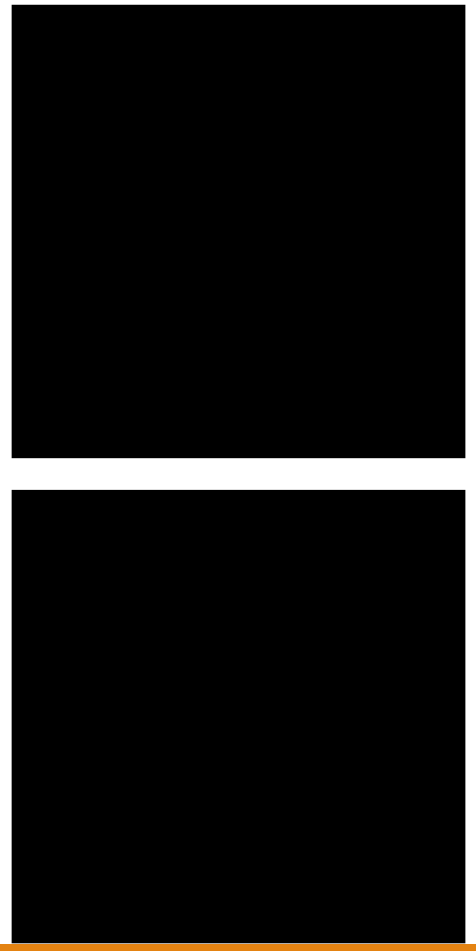
Z-буфер

Идея Z-буфера в том, чтобы хранить расстояния от камеры до точек в специальном массиве, размер которого совпадает с размером изображения.

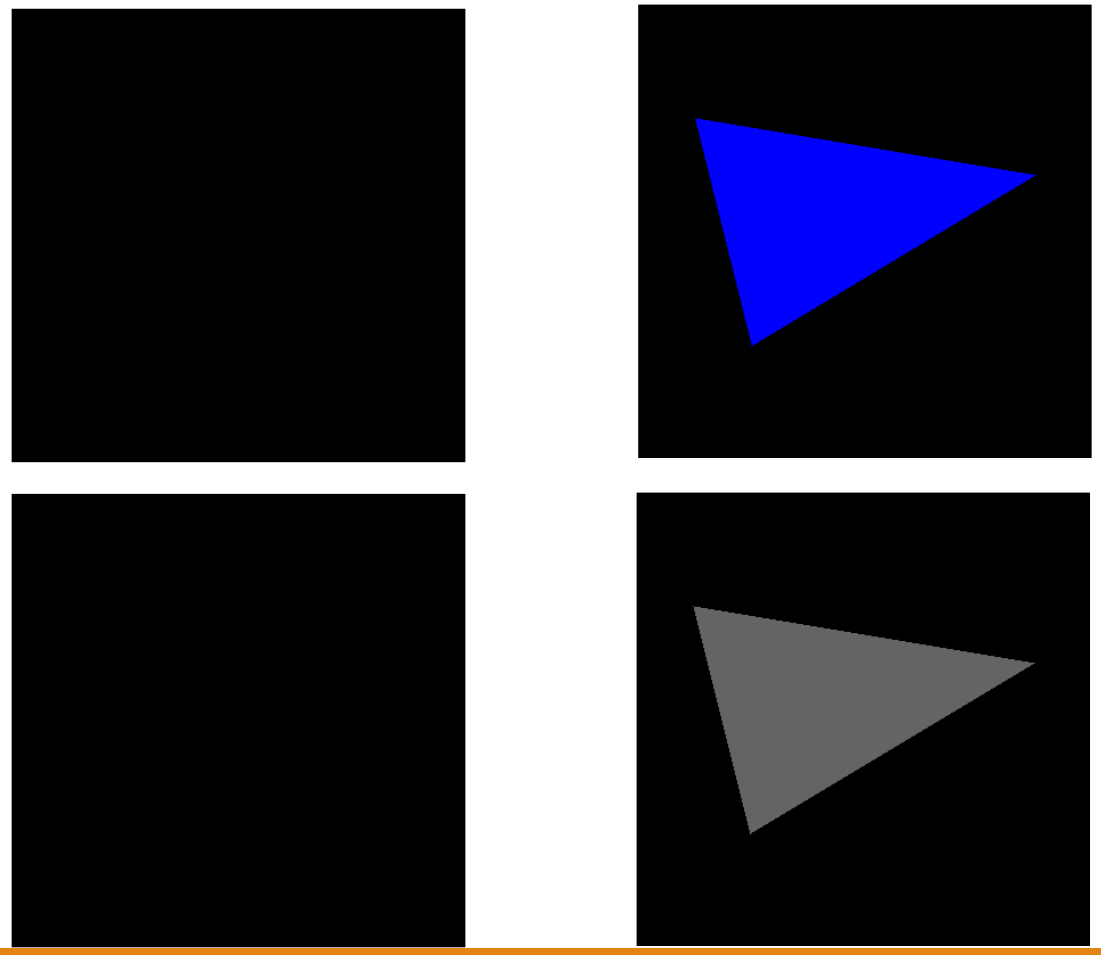
Каждый раз, когда нужно отрисовать точку, мы проверяем, находится ли эта точка ближе или дальше **предыдущей** отрисованной точки в этом **пикселе**. Если дальше, то игнорируем эту точку. Если ближе, то рисуем, а в Z-буфер помещаем новое расстояние.

Для того, чтобы самая первая точка всегда рисовалась в каждом пикселе, изначально инициализируем Z-буфер большими значениями. Например, np.inf в python.

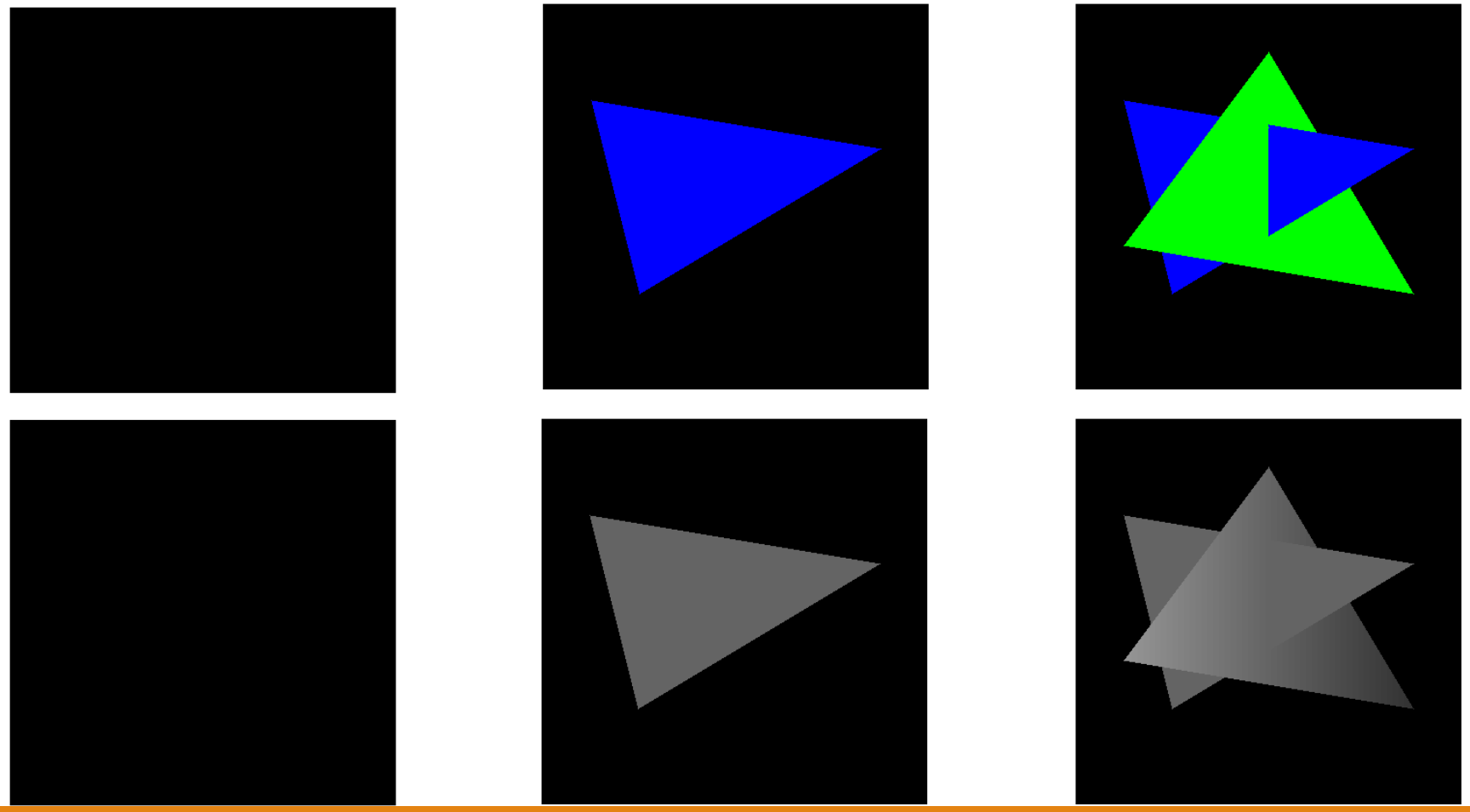
Отрисовка пересекающихся треугольников с z-буфером



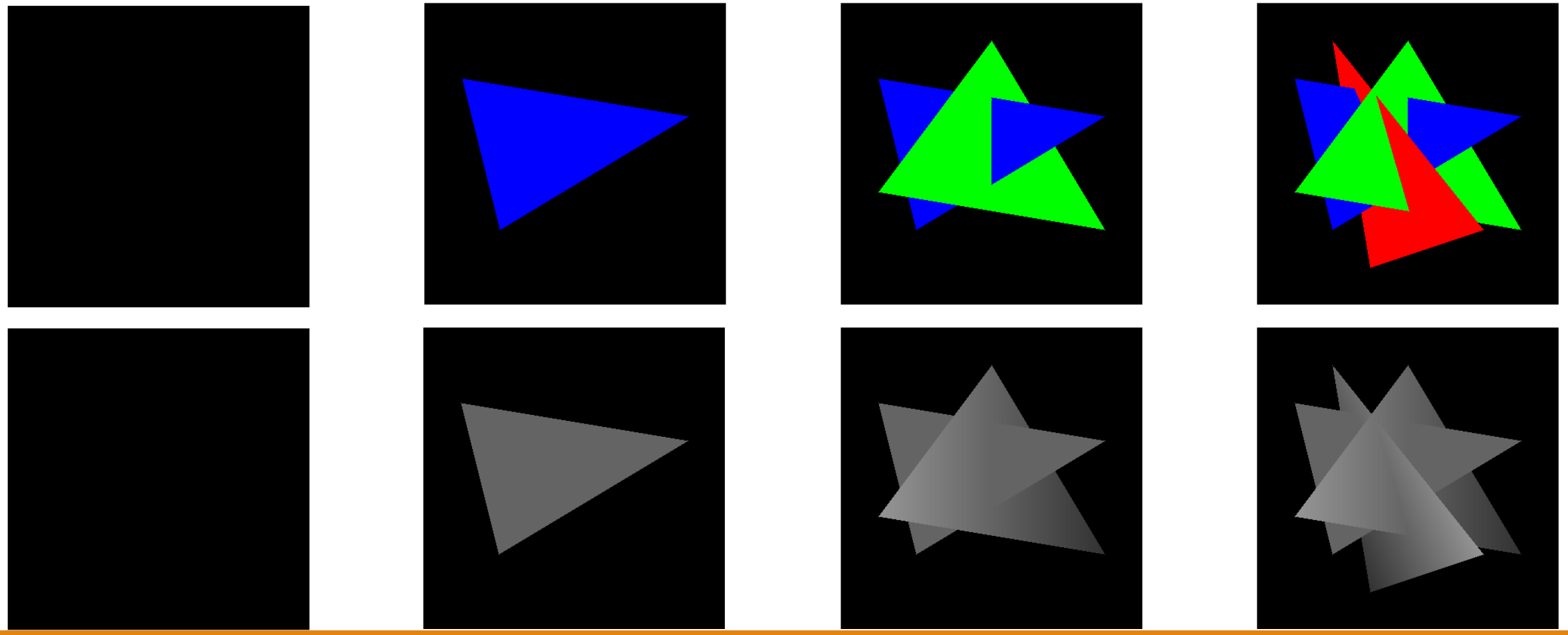
Отрисовка пересекающихся треугольников с z-буфером



Отрисовка пересекающихся треугольников с z-буфером



Отрисовка пересекающихся треугольников с z-буфером



Определение Z-координаты

Барицентрические координаты описывают положение точки относительно её вершин.

Можно вернуться к **исходным** трёхмерным вершинам, и используя барицентрические координаты, получить точку треугольника в трёхмерном пространстве, которая будет соответствовать текущему пикселю.

Координата Z этой точки в этом случае будет вычисляться по формуле:

$$Z = Z_0\lambda_0 + Z_2\lambda_2 + Z_2\lambda_2$$

и покажет удалённость от камеры.

Пример

Пусть даны трёхмерные точки (10.0, 10.0, 100.0), (5.0, 5.0, 200.0), (10.0, 5.0, 300.0).

Им соответствует треугольник на плоскости (10.0, 10.0), (5.0,5.0), (10.0, 5.0).

Мы хотим отрисовать точку, соответствующую пикселю (8, 6).

Его барицентрические координаты: (0.2, 0.4, 0.4).

Можно проверить: 0.2*10.0 + 0.4*5.0 + 0.4*10.0 = 8.0, 0.2*10.0 + 0.4*5.0 + 0.4*5.0 = 6.0.

Координата Z точки в **исходном трёхмерном треугольнике**, соответствующей этому пикселю, будет иметь значение:

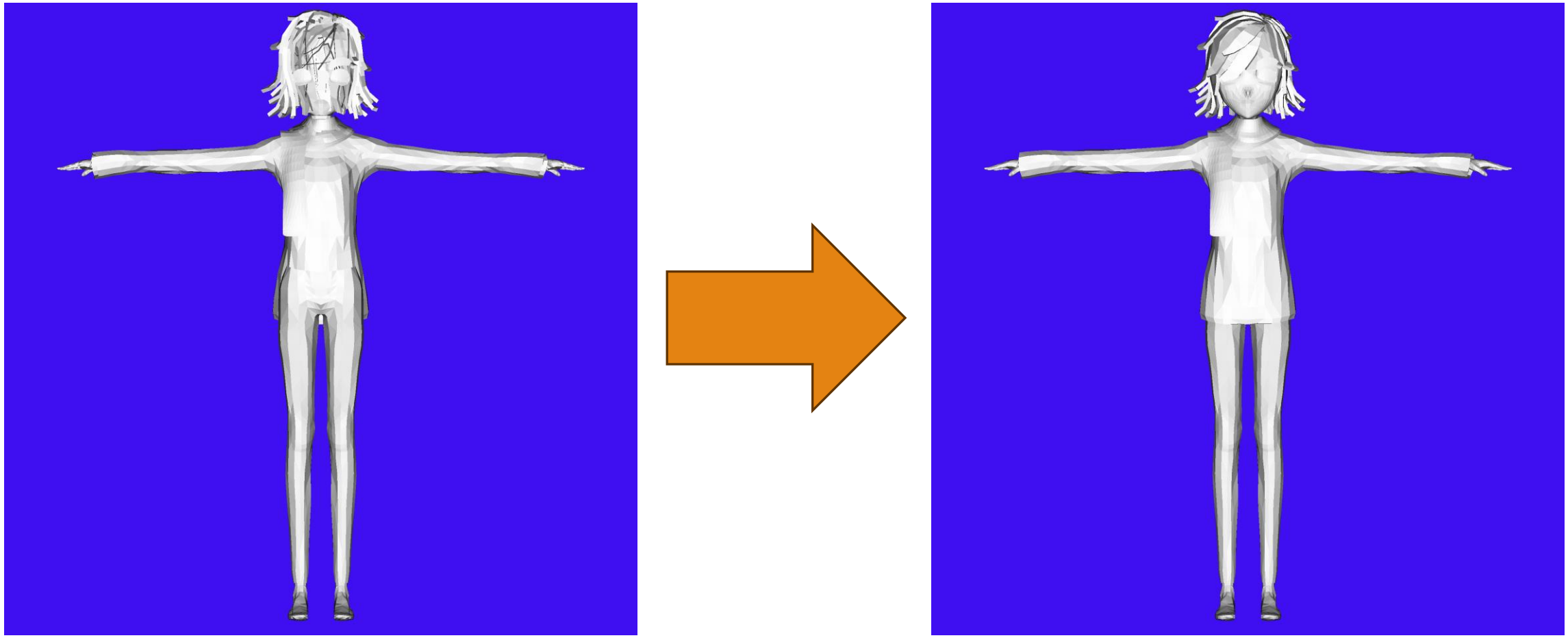
0.2*100.0 + 0.4*200.0 + 0.4*300.0 = 220

Полный конвейер рендеринга сцены

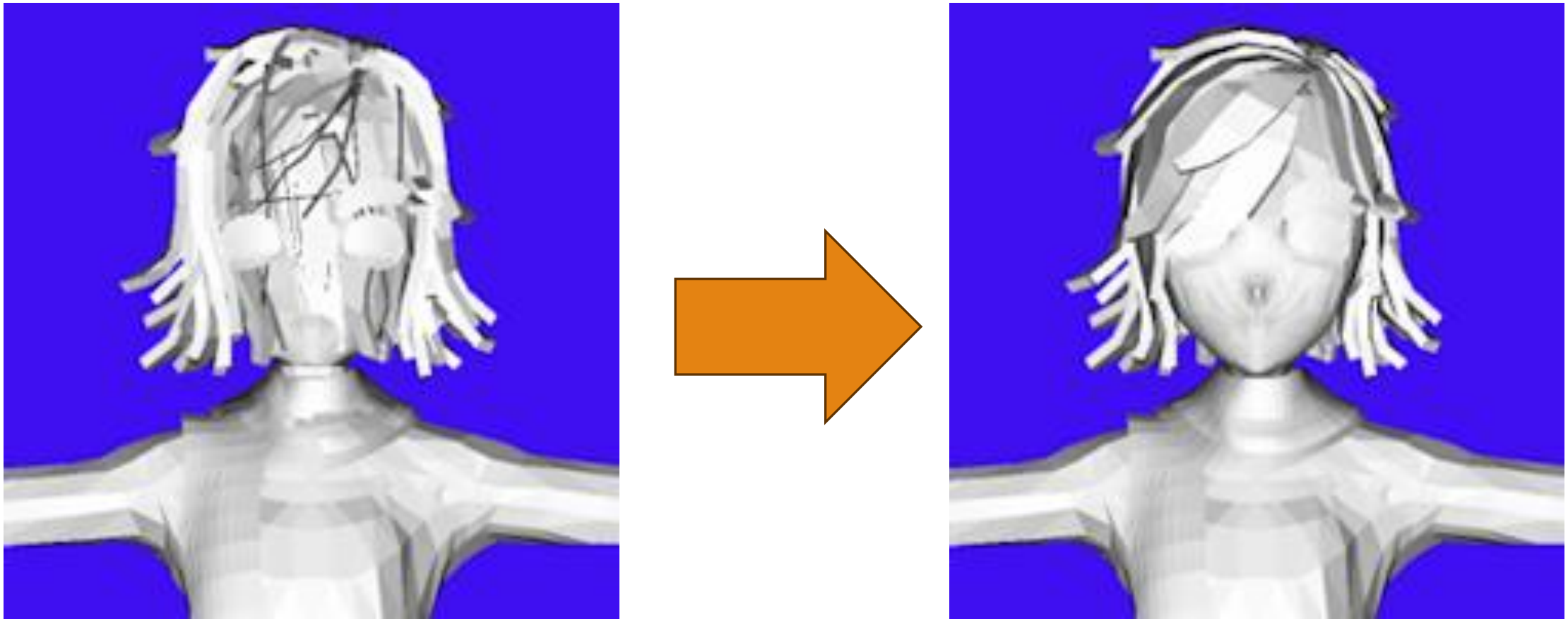
Для каждого треугольного полигона сцены:

- Рассчитать нормаль к полигону.
- Если нормаль направлена под углом <90 к наблюдателю
  - Перейти к следующему полигону.
- Рассчитать область интереса.
  - Для каждого пиксела внутри области интереса:
  - Рассчитать барицентрические координаты пиксела относительно вершин треугольника.
  - Если все барицентрические координаты больше или равны нулю:
    - Если z координата пиксела меньше текущего значения z буфера для этого пиксела:
      - Нарисовать пиксель.
      - Присвоить z буферу значение z координаты пикселя.

Эффект от добавления Z-буфера



Эффект от добавления Z-буфера



Что должно в итоге получиться?