

**Первая лабораторная: задания 1-6.**

**Вторая лабораторная: задания 7-10 + задания 11-14.**

**Третья лабораторная: задание 15 + задание 16.**

**Четвертая лабораторная: задание 17 + задание 18.**

**Пятая лабораторная: обсуждается индивидуально.**

## **1. Работа с изображениями.**

Выбрать язык программирования и библиотеку для записи изображений в файл.

Создать матрицу размера  $H \times W$ , заполнить её элементы нулевыми значениями, сохранить в виде полутонового (одноканального) 8-битового изображения высотой  $H$  и шириной  $W$ , убедиться, что полученное изображение открывается средствами операционной системы и полностью чёрное.

Создать матрицу размера  $H \times W$ , заполнить её элементы значениями, равными 255, сохранить в виде полутонового (одноканального) 8-битового изображения высотой  $H$  и шириной  $W$ , убедиться, что полученное изображение открывается средствами операционной системы и полностью белое.

Создать матрицу размера  $H \times W \times 3$ , заполнить её элементы значениями, равными (255, 0, 0), сохранить в виде цветного (трёхканального) 8-битового изображения высотой  $H$  и шириной  $W$ , убедиться, что полученное изображение открывается средствами операционной системы и полностью красное.

Создать матрицу размера  $H \times W \times 3$ , заполнить её элементы произвольными значениями по выбранной схеме (например, значение элемента равно сумме его координат по модулю 256), сохранить в виде 8-битового изображения высотой  $H$  и шириной  $W$ , убедиться, что полученное изображение открывается средствами операционной системы (в предложенном примере должен получиться плавный градиент от чёрного цвета в верхнем левом углу изображения).

## **2. Отрисовка прямых линий**

Реализовать все описанные в лекциях алгоритмы отрисовки прямых (до алгоритма Брезенхема включительно).

Для каждого алгоритма сохранить в файл изображение размера 200x200 с нарисованной на нём «звездой» (см. лекции).

*Подсказка:*

*начальная координата (100,100)*

*конечная координата  $(100 + 95 \cos(\alpha), 100 + 95 \sin(\alpha))$ ,  $\alpha = \frac{2\pi i}{13}$ ,  $i = 0, 1, \dots, 12$ .*

## **3. Работа с трёхмерной моделью (вершины)**

Считать из приложенного файла obj строки, содержащие информацию о вершинах модели:

v X1 Y1 Z1

v X2 Y2 Z2

<...>

#### 4. Отрисовка вершин трёхмерной модели

Нарисовать вершины модели (игнорируя координату  $Z$ ) на изображении размером (2000, 2000).

Для того, чтобы модель была видна на изображении (и не была слишком большой), поэкспериментируйте с масштабированием и смещением координат точек, например:  $[50 * X + 1000, 50 * Y + 1000]$ .

#### 5. Работа с трёхмерной моделью (полигоны)

Считать из приложенного файла строки, содержащие информацию о полигонах модели.

Сведения о полигонах в файле хранятся в формате:

$f \ v1/vt1/vn1 \ v2/vt2/vn2 \ v3/vt3/vn3$

В рамках лабораторной загрузить в память необходимо только первые значения в каждой тройке – номера вершин, загруженных ранее. Обратите внимание, что вершины нумеруются, начиная с единицы.

*Вообще говоря, есть ещё варианты:*

$f \ v1 \ v2 \ v3$

$f \ v1/vt1 \ v2/vt2 \ v3/vt3$

$f \ v1/vn1 \ v2/vn2 \ v3/vn3$ .

*Если хотите, можете учитывать это*

#### 6. Отрисовка рёбер трёхмерной модели

Отрисовать все рёбра всех полигонов модели с помощью алгоритма

Брезенхема (координаты вершин округляем до ближайшего целого).

## 7. Барицентрические координаты

Написать функцию вычисления барицентрических координат для точки с экранными (целочисленными координатами)  $(x, y)$  относительно вещественных вершин треугольника  $(x_0, y_0)$ ,  $(x_1, y_1)$  и  $(x_2, y_2)$ .

Они вычисляются по формулам:

```
lambda0 = ((x - x2) * (y1 - y2) - (x1 - x2) * (y - y2)) /  
           ((x0 - x2) * (y1 - y2) - (x1 - x2) * (y0 - y2))  
lambda1 = ((x0 - x2) * (y - y2) - (x - x2) * (y0 - y2)) /  
           ((x0 - x2) * (y1 - y2) - (x1 - x2) * (y0 - y2))  
lambda2 = 1.0 - lambda0 - lambda1
```

Обратите внимание, что координаты  $(x, y)$  – экранные, и поэтому целочисленные. В то же время вершины треугольника  $(x_0, y_0)$ ,  $(x_1, y_1)$  и  $(x_2, y_2)$  – вещественные, округлять их перед вычислениями **не надо**.

## 8. Отрисовка треугольников

Написать функцию отрисовки треугольника с вершинами  $(x_0, y_0)$ ,  $(x_1, y_1)$  и  $(x_2, y_2)$ . Для этого выполнить следующие шаги.

1. Определить ограничивающий прямоугольник: минимальные и максимальные возможные значения координат X и Y. Например:

```
xmin = min(x0, x1, x2)
```

2. Для ограничивающего прямоугольника учесть границы изображения, так, например:

```
if (xmin < 0): xmin = 0
```

Разумеется, вы можете сделать 1 и 2 пункты одновременно.

3. Для каждого пикселя внутри ограничивающего прямоугольника вычислить барицентрические координаты относительно вершин треугольника.

Если **все** барицентрические координаты пикселя больше или равны нулю – пиксель рисуется, иначе – переходим к следующему.

Обратите внимание, что отдельно рёбра треугольника рисовать **не надо**.

## 9. Тестирование функции

Протестировать функцию отрисовки треугольника для разных треугольников, в том числе, частично (или полностью) выходящих за пределы изображения.

## 10. Отрисовка полигонов трёхмерной модели

Нарисовать все полигоны модели разными цветами (для одного треугольника – один случайный цвет).

## 11. Вычисление нормали к поверхности треугольника

Для каждого треугольника вычислить нормаль к этому треугольнику по формуле:

$$\bar{n} = [X_1 - X_2 \quad Y_1 - Y_2 \quad Z_1 - Z_2] \times [X_1 - X_0 \quad Y_1 - Y_0 \quad Z_1 - Z_0],$$

где  $(X_0, Y_0, Z_0)$ ,  $(X_1, Y_1, Z_1)$  и  $(X_2, Y_2, Z_2)$  – **исходные** координаты вершин треугольника (до любых преобразований), а  $\times$  – векторное произведение.

Координаты нормали могут быть вычислены через определитель:

$$\bar{n} = \begin{vmatrix} \bar{i} & \bar{j} & \bar{k} \\ X_1 - X_2 & Y_1 - Y_2 & Z_1 - Z_2 \\ X_1 - X_0 & Y_1 - Y_0 & Z_1 - Z_0 \end{vmatrix}.$$

## 12. Отсечение нелицевых граней

Для каждого треугольника определить косинус угла падения направленного света (считать направление света равным  $\bar{l} = [0, 0, 1]$ ) через нормализованное скалярное произведение  $\frac{\langle \bar{n}, \bar{l} \rangle}{\|\bar{n}\| \cdot \|\bar{l}\|}$ .

Изменить цикл отрисовки полигонов таким образом, чтобы были отрисованы только полигоны с  $\frac{\langle \bar{n}, \bar{l} \rangle}{\|\bar{n}\| \cdot \|\bar{l}\|} < 0$ .

## 13. Базовое освещение

Отрисовку полигонов выполнять не случайным цветом, а пропорциональным косинусу угла между  $\bar{n}$  и  $\bar{l}$ , например  $\left(-255 * \frac{\langle \bar{n}, \bar{l} \rangle}{\|\bar{n}\| \cdot \|\bar{l}\|}, 0, 0\right)$ .

*Если вы тестируете на кролике, он должен получиться «стиной» к вам.*

## 14. z-буфер

При отрисовке полигонов проверять перекрытие полигонов с использованием z-буфера. z-буфер – это матрица из вещественных значений по размеру совпадающая с изображением. Все элементы z-буфера изначально инициализируются некоторым достаточно большим значением.

При отрисовке для каждой точки выполняется следующая проверка:

1. Вычисляются барицентрические координаты  $(\lambda_0, \lambda_1, \lambda_2)$ .
2. Если все барицентрические координаты больше нуля, вычисляем z-координату **исходного** полигона через **исходные** z-координаты вершин этого полигона:  
 $\hat{z} = \lambda_0 z_0 + \lambda_1 z_1 + \lambda_2 z_2$ .
3. Если вычисленное значение координаты  $\hat{z}$  больше координаты z-буфера для текущего пикселя, пропускаем точку.
4. Если вычисленное значение координаты  $\hat{z}$  меньше координаты z-буфера для текущего пикселя, рисуем этот пиксель, а соответствующему элементу z-буфера присваиваем значение  $\hat{z}$ .

## 15. Поворот модели

Реализовать поворот и сдвиг модели. Для этого исходные координаты вершин заменить на преобразованные:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix},$$

где

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\alpha, \beta, \gamma$  – углы поворота вокруг осей X, Y и Z, соответственно.

*Комментарий 1. Для начала рекомендуется попробовать реализовать поворот вокруг оси Y. Такой поворот проще всего визуально отследить и проверить.*

*Комментарий 2. Начиная с этого задания, пиксельный сдвиг должен составлять ровно половину высоты и половину ширины изображения. Попробуйте отцентрировать модель за счёт сдвига самой модели, подобрав подходящие  $t_X$  и  $t_Y$ .*

## 16. Проективное преобразование

Заменить преобразование к экранным координатам вида

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 50 * X + 500 \\ 50 * Y + 500 \end{bmatrix}$$

на проективное преобразование:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \propto \begin{bmatrix} a_X & 0 & u_0 \\ 0 & a_Y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

где  $(u, v)$  – экранные координаты,  $a_X, a_Y$  – масштаб,  $(u_0, v_0)$  – центр изображения,  $\propto$  – знак пропорциональности.

*Комментарий 1. Для тех, кому лень разбираться:*

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{a_X X}{Z} + u_0 \\ \frac{a_Y Y}{Z} + v_0 \end{bmatrix}$$

*Комментарий 2. Для того, чтобы проективное преобразование вычислялось правильно, не забывайте отодвигать модель от начала координат (так, чтобы значения Z всех точек стали положительными).*

## 17. Затенение Гуро.

Реализовать сглаженное освещение посредством метода затенения Гуро.

Для этого необходимо будет вычислить нормали ко всем вершинам модели, усреднив нормали ко всем полигонам, соседним с вершиной.

Для каждой вершины вычислить освещение этой вершины  $I_0, I_1, I_2$ .

$$I_0 = \frac{\langle \bar{n}_0, \bar{l} \rangle}{\|\bar{n}_0\| \cdot \|\bar{l}\|}, \quad I_1 = \frac{\langle \bar{n}_1, \bar{l} \rangle}{\|\bar{n}_1\| \cdot \|\bar{l}\|}, \quad I_2 = \frac{\langle \bar{n}_2, \bar{l} \rangle}{\|\bar{n}_2\| \cdot \|\bar{l}\|}$$

где  $\bar{l}$  – направление света, а  $\bar{n}_0, \bar{n}_1, \bar{n}_2$  – нормали соответствующих вершин треугольника

Для каждой точки полигона вычислить освещение через барицентрические координаты относительно вершин этого полигона: Значение яркости (цвета) пикселя равно:

$$I = -225 * (\lambda_0 I_0 + \lambda_1 I_1 + \lambda_2 I_2)$$

*Можно загрузить нормали из модели, но это в качестве бонуса. Основным вариантом будет вычисление вручную.*

## 18. Текстурирование модели.

Реализовать текстурирование модели.

Считать из файла координаты текстур (строки, начинающиеся с vt), считать из файла номера координат текстур. Сведения о номерах текстур в файле хранятся во второй позиции в строках, начинающихся с f:

f v1/vt1/vn1 v2/vt2/vn2 v3/vt3/vn3

Загрузить изображение текстуры в память.

Каждой вершине будет соответствовать относительная координата текстуры  $\begin{bmatrix} u_t \\ v_t \end{bmatrix}$ . Если известны координаты текстур  $\begin{bmatrix} u_t^0 \\ v_t^0 \end{bmatrix}, \begin{bmatrix} u_t^1 \\ v_t^1 \end{bmatrix}, \begin{bmatrix} u_t^2 \\ v_t^2 \end{bmatrix}$  для трёх вершин треугольника, итоговое значение цвета пикселя лежит в изображении текстуры в точке с координатами:

$$\begin{bmatrix} W_T(\lambda_0 u_t^0 + \lambda_1 u_t^1 + \lambda_2 u_t^2) \\ H_T(\lambda_0 v_t^0 + \lambda_1 v_t^1 + \lambda_2 v_t^2) \end{bmatrix},$$

где  $\lambda_0, \lambda_1, \lambda_2$  – барицентрические координаты,  $W_T$  и  $H_T$  – ширина и высота изображения текстуры в пикселях. Не забудьте округлить полученное значение координаты.