

Министерство образования Республики Беларусь
Учреждение образования “Белорусский государственный университет
информатики и радиоэлектроники”

Факультет информационных технологий и управления
Кафедра интеллектуальных информационных технологий
Дисциплина: Графический интерфейс интеллектуальных систем

Отчёт к лабораторной работе №4

Выполнил:
Группа:
Проверила:

Гафаров М.С.
221702
Жмырко А.В.

Минск 2024

Лабораторная работа №4

Геометрические преобразования

Цель: изучить основные типы преобразований двумерных и трёхмерных геометрических фигур в пространстве.

Задание: разработать графическую программу, выполняющую следующие геометрические преобразования над трёхмерным объектом: перемещение, поворот, скалирование(масштабирование), отображение, перспектива(перспективная проекция). В программе должно быть предусмотрено считывание координат 3D объекта из текстового файла, обработка клавиатуры и выполнение геометрических преобразований в зависимости от нажатых клавиш. Все преобразования следует производить с использованием матричного аппарата и представления координат в однородных координатах.

Теоретические сведения:

Геометрическое преобразование – это взаимно однозначное отображение прямой, плоскости или пространства на себя.

Наиболее общей разновидностью преобразований являются аффинные преобразования.

Аффинные преобразования – это точечные взаимно однозначные отображения плоскости (пространства) на себя, при которых прямые переходят в прямые (сохраняется параллельность линий).

Основные свойства Аффинных преобразований: сохраняют прямые и параллельные линии, эллипсы и кривые Безье, а также могут быть определены тремя вершинами неколлинеарных векторов p, q, r в исходной системе координат и тремя вершинами неколлинеарных векторов p', q', r' в результирующей системе координат.

Над трёхмерным объектом определены следующие Аффинные преобразования: сдвиг (перемещение), поворот, скалирование (масштабирование), отображение, проекция(параллельная и центральная (перспективная)).

Программная реализация:

Построение трёхмерной фигуры по её матричному представлению:

```
def draw_figure_by_matrix(self, matrix): 4 usages
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    x = [row[0] for row in matrix]
    y = [row[1] for row in matrix]
    z = [row[2] for row in matrix]
    ax.scatter(x, y, z, c='r', marker='o')
    ax.plot(x + [x[0]], y + [y[0]], z + [z[0]], c='b')
    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('Z')
    plt.show()
```

Операция сдвига (перемещения):

```
def moving(self, new_coords: list, figure_matrix: list): 3 usages
    avgx, avgy, avgz = 0, 0, 0
    for i in range(len(figure_matrix)):
        avgx += figure_matrix[i][0]
        avgy += figure_matrix[i][1]
        avgz += figure_matrix[i][2]
    avgx, avgy, avgz = avgx / len(figure_matrix), avgy / len(figure_matrix), avgz / len(figure_matrix)
    center_coords = [avgx, avgy, avgz]
    Dx, Dy, Dz = (int(new_coords[0]) - center_coords[0], int(new_coords[1]) - center_coords[1],
                  int(new_coords[2]) - center_coords[2])
    T = [[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [Dx, Dy, Dz, 1]]
    M = matrix_multiplication(figure_matrix, T)
    return M
```

Операция поворота:

```
def turn(self, angle: str, figure_matrix: list, axis: str): 3 usages
    rad = math.radians(int(angle))
    sinQ = math.sin(rad)
    cosQ = math.cos(rad)
    M = []
    if axis == 'x' or axis == 'X':
        T = [[1, 0, 0, 0], [0, cosQ, sinQ, 0], [0, -sinQ, cosQ, 0], [0, 0, 0, 1]]
        M = matrix_multiplication(figure_matrix, T)
    elif axis == 'y' or axis == 'Y':
        T = [[cosQ, 0, -sinQ, 0], [0, 1, 0, 0], [sinQ, 0, cosQ, 0], [0, 0, 0, 1]]
        M = matrix_multiplication(figure_matrix, T)
    elif axis == 'z' or axis == 'Z':
        T = [[cosQ, sinQ, 0, 0], [-sinQ, cosQ, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
        M = matrix_multiplication(figure_matrix, T)
    return M
```

Операция масштабирования (скалирования):

```
def scaling(self, coeffs: list, figure_matrix: list): 3 usages
    Sx, Sy, Sz = float(coeffs[0]), float(coeffs[1]), float(coeffs[2])
    S = [[Sx, 0, 0, 0], [0, Sy, 0, 0], [0, 0, Sz, 0], [0, 0, 0, 1]]
    M = matrix_multiplication(figure_matrix, S)
    return M
```

Операция отображения:

```
def display(self, figure_matrix: list, axis: str): 5 usages
    M = []
    if axis == 'y0z':
        T = [[-1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
        M = matrix_multiplication(figure_matrix, T)
    elif axis == 'x0z':
        T = [[1, 0, 0, 0], [0, -1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
        M = matrix_multiplication(figure_matrix, T)
    elif axis == 'x0y':
        T = [[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, -1, 0], [0, 0, 0, 1]]
        M = matrix_multiplication(figure_matrix, T)
    return M
```

Операция проекции (центральной/перспективной):

```
def perspective_projection(self, figure_matrix: list, axis__: str, project_distance: str):
    d = int(project_distance)
    M = []
    if axis == 'x' or axis == 'X':
        T = [[1, 0, 0, 1 / d], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
        M = matrix_multiplication(figure_matrix, T)
    elif axis == 'y' or axis == 'Y':
        T = [[1, 0, 0, 0], [0, 1, 0, 1 / d], [0, 0, 1, 0], [0, 0, 0, 1]]
        M = matrix_multiplication(figure_matrix, T)
    elif axis == 'z' or axis == 'Z':
        T = [[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 1 / d], [0, 0, 0, 1]]
        M = matrix_multiplication(figure_matrix, T)
    for i in range(len(M)):
        last_element = M[i][-1]
        if last_element > 1:
            M[i] = [x / last_element for x in M[i]]
    return M
```

Вывод к лабораторной работе: в результате выполнения лабораторной работы были изучены алгоритмы основных аффинных преобразований над трёхмерным объектом: перемещение, поворот, масштабирование, отображение, центральная проекция. Изученные алгоритмы были реализованы на практике с использованием матричного аппарата и представления координат в однородных координатах.

