**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по практической работе**

**по дисциплине «Компьютерная графика»**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 5371 |  | Локкина О.С. |
|  |  | Бергер Э.Э. |
|  |  | Уруков С.Д. |
| Преподаватель |  | Матвеева И.В. |

Цель работы:

Научиться строить изометрическую и диметрическую проекции с помощью матриц для проволочного тела, ребра которого параллельны осям координат.

Задание:

Нарисовать заданную фигуру (ступеньки) в трехмерном пространстве. Получить диметрическую и изометрическую проекцию фигуры по желанию пользователя.

Изначальные точки координат «ступеньки» (x, y, z).

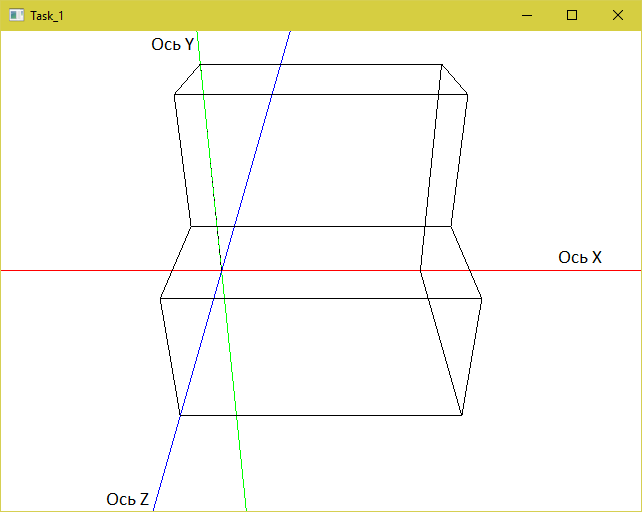
verticesOriginal = ((2, 0, 0), (2, 0, 2), (0, 0, 2), (0, 0, 0), (2, 1, 2), (0, 1, 2), (2, 1, 1), (0, 1, 1), (2, 2, 1), (0, 2, 1), (0, 2, 0), (2, 2, 0))

Ход работы:

Использовалась библиотека OpenGL с языком программирования Python 3.6.x. В трехмерном пространстве по вершинам были соединены вышеуказанные точки. Для наглядности были добавлены следующие особенности:

* Возможность передвигать точку наблюдения в пространстве по координате X при помощи клавиш “W” и “S”;
* Отрисованы оси координат (X – красная, Y – зеленая, Z – синяя)

Полученный результат имеет вид:



При построении диметрической и изометрической проекций использовались матрица с вычисленными коэффициентами, которые были даны на лекции.

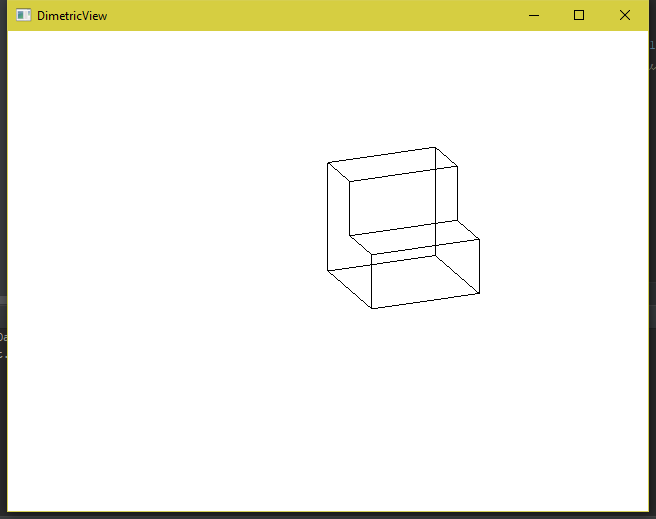
Диметрическая матрица

|  |  |  |
| --- | --- | --- |
| 0.92582 | 0.13363 | -0.353553 |
| 0 | 0.935414 | 0.353553 |
| 0.377964 | -0.327329 | 0.866025 |

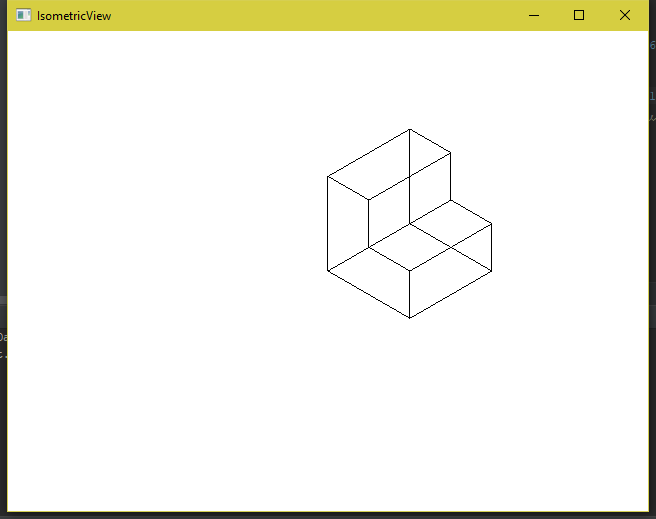
Изометрическая матрица

|  |  |  |
| --- | --- | --- |
| 0.707107 | 0.408248 | -0.577353 |
| 0 | 0.816497 | 0.577345 |
| 0.707107 | -0.408248 | 0.577353 |

При нажатии на клавишу «2» открывается новое окно, в котором получается диметрическая проекция фигуры.



При нажатии на клавишу «3» открывается новое окно, в котором получается изометрическая проекция фигуры.



**Выводы**

В ходе работы были изучены построения проекций матрицами трехмерной фигуры. Были освоены навыки работы с PyOpenGL.

**Код программы**

from OpenGL.GL import \*

from OpenGL.GLU import \*

from OpenGL.GLUT import \*

#from math import cos, sin, pi

window = 0

verticesOriginal = (

(2, 0, 0), (2, 0, 2), (0, 0, 2), (0, 0, 0), (2, 1, 2), (0, 1, 2), (2, 1, 1), (0, 1, 1), (2, 2, 1), (0, 2, 1),

(0, 2, 0), (2, 2, 0))

eye = (2,3,5)

edges = (

(0, 1), (0, 3), (0, 11), (1, 2), (1, 4), (2, 5), (2, 3), (4, 5), (4, 6), (5, 7), (6, 8), (7, 9), (6, 7), (8, 9),

(8, 11), (9, 10), (10, 3), (10, 11))

def matrixMultiplication(matrixOne, matrixTwo):

"""

Эта функция используется для умножения двух матриц

:param matrixOne:

:param matrixTwo:

:return: результат умножения матриц, вложенный массив

"""

result = []

for i in range(12):

result.append([0] \* 3)

for j in range(3):

for k in range(3):

result[i][j] += matrixOne[i][k] \* matrixTwo[k][j]

return result

def dimetryMatrix():

"""

диметрическая матрица

:rtype:

"""

return [[0.92582, 0.133631, -0.353553], [0, 0.935414, 0.353553], [0.377964, -0.327329, 0.866025]]

def isometryMatrix():

"""

Возвращает изометрическую матрицу

:rtype:

"""

return [[0.707107, 0.408248, -0.577353], [0, 0.816497, 0.577345], [0.707107, -0.408248, 0.577353]]

#return [[0.707107, 0.408248, 0], [0, 0.816497, 0], [0.707107, -0.408248, 0]]

def keyPressed(bkey, x, y):

"""

Обработчик событий клавиш в программе

:rtype: object

"""

global eye

# Convert bytes object to string

try:

key = bkey.decode("utf-8")

# Allow to quit by pressing 'Esc' or 'q'

if key == chr(27):

sys.exit()

if key == chr(51):

initIsometricView()

if key == chr(50):

initDimetricView()

if key == 'q':

print("Bye!")

sys.exit()

if key == 'w':

eye = (eye[0] + 1, eye[1], eye[2])

if key == 's':

eye = (eye[0] - 1, eye[1], eye[2])

except:

pass

def initFigure3D(vertices):

"""

Отрисовывает линии ступенек

:rtype: object

"""

glBegin(GL\_LINES)

glColor3f(0, 0, 0)

for edge in edges:

for vertex in edge:

glVertex3fv(vertices[vertex])

glEnd()

def initDimetry():

"""

Отрисовывает линии диметрической проекции

"""

vertices = matrixMultiplication(verticesOriginal, dimetryMatrix())

for element in vertices:

element.pop()

glBegin(GL\_LINES)

glColor3f(0, 0, 0)

for edge in edges:

for vertex in edge:

glVertex2dv(vertices[vertex])

glEnd()

def initIsometry():

"""

Отрисовывает линии изометрической проекции

"""

vertices = matrixMultiplication(verticesOriginal,isometryMatrix())

for element in vertices:

element.pop()

glBegin(GL\_LINES)

glColor3f(0, 0, 0)

for edge in edges:

for vertex in edge:

glVertex2dv(vertices[vertex])

glEnd()

def initDimetricView():

"""

Инициализирует вид. Запускает окно и обработчики событий.

"""

global window

glutInit(sys.argv)

glutInitDisplayMode(GLUT\_RGBA | GLUT\_DOUBLE | GLUT\_DEPTH)

glutInitWindowSize(640, 480)

glutInitWindowPosition(200, 200)

window = glutCreateWindow(b"DimetricView")

glutDisplayFunc(drawDimetricScene)

glutIdleFunc(drawDimetricScene)

glutKeyboardFunc(keyPressed)

initGL(640, 480)

glutMainLoop()

def drawIsometricScene():

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)

glLoadIdentity()

gluLookAt(0, 0, 10, 0, 0, 0, 0, 1, 0)

# glTranslatef(0.0, 0.0, -6.0)

initIsometry()

glutSwapBuffers()

def drawDimetricScene():

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)

glLoadIdentity()

gluLookAt(0, 0, 10, 0, 0, 0, 0, 1, 0)

# glTranslatef(0.0, 0.0, -6.0)

initDimetry()

glutSwapBuffers()

def initGL(Width, Height):

glClearColor(1.0, 1.0, 1.0, 1.0)

glClearDepth(1.0)

glDepthFunc(GL\_LESS)

glEnable(GL\_DEPTH\_TEST)

glShadeModel(GL\_SMOOTH)

glMatrixMode(GL\_PROJECTION)

glLoadIdentity()

gluPerspective(45.0, float(Width) / float(Height), 0.1, 100.0)

glMatrixMode(GL\_MODELVIEW)

def init\_axes():

"""

Отрисовывает оси координат

"""

glBegin(GL\_LINES)

glColor3f(1, 0, 0)

glVertex3f(-100, 0, 0)

glVertex3f(100, 0, 0)

glColor3f(0, 1, 0)

glVertex3f(0, -100, 0)

glVertex3f(0, 100, 0)

glColor3f(0, 0, 1)

glVertex3f(0, 0, -100)

glVertex3f(0, 0, 100)

glEnd()

def draw3DScene():

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)

glLoadIdentity()

gluLookAt(eye[0], eye[1], eye[2], 0, 0, 0, 0, 1, 0)

# glTranslatef(0.0, 0.0, -6.0)

init\_axes()

initFigure3D(verticesOriginal)

glutSwapBuffers()

def initMainWindow():

global window

glutInit(sys.argv)

glutInitDisplayMode(GLUT\_RGBA | GLUT\_DOUBLE | GLUT\_DEPTH)

glutInitWindowSize(640, 480)

glutInitWindowPosition(200, 200)

window = glutCreateWindow(b"Task\_1")

glutDisplayFunc(draw3DScene)

glutIdleFunc(draw3DScene)

glutKeyboardFunc(keyPressed)

initGL(640, 480)

glutMainLoop()

def initIsometricView():

global window

glutInit(sys.argv)

glutInitDisplayMode(GLUT\_RGBA | GLUT\_DOUBLE | GLUT\_DEPTH)

glutInitWindowSize(640, 480)

glutInitWindowPosition(200, 200)

window = glutCreateWindow(b"IsometricView")

glutDisplayFunc(drawIsometricScene)

glutIdleFunc(drawIsometricScene)

glutKeyboardFunc(keyPressed)

initGL(640, 480)

glutMainLoop()

def main():

initMainWindow()

if \_\_name\_\_ == "\_\_main\_\_":

print("Hit ESC key to quit.\n" +

"Press W or S to rotate figure\n" +

"Hit 2 to get dimetric view and 3 to get isometric one")

main()