Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Московский политехнический университет»

**факультет информационных технологий**

**Кафедра СМАРТ-технологий**

Дисциплина: Технологии программирования инженерных систем

Отчёт по лабораторной работе №2

«Освоение базовых инструментов обработки и хранения и визуализации массивов данных на языке программирования»

Вариант: 4

Работу выполнил\_и

студент\_ка 3 курса

очного отделения

<ФИО>

Научный руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<ФИО>

**Цель работы**

Разработать базовые алгоритмы обмена данными с устройствами хранения, статистической обработки данных и визуализации результатов анализа.

**Задачи**

* Подготовить описание проектов задач для обработки данных;
* Подготовить модуль загрузки, сохранения и отображения массивов данных, хранимых в текстовом представлении (формат CSV);
* Подготовить модуль визуализации исходных (сырых) данных с использованием 3D-представления данных.
* Разработать модуль генерации модельных данных в соответствии с заданными условиями

**Индивидуальное задание**

L02\_04: использовать в качестве пространственной функции параметрическое

представление функции.

При необходимости выполнить масштабирование в интервал [ -1 ÷ 1] по

каждому измерению.

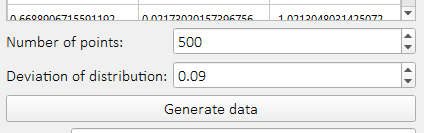
x = 0.7 \* cos (6 \* pi \* (i / N))

y = 0.5 \* sin (4 \* pi \* (i / N))

z = -1 + 2 \* i / N (0 <= i <= N)

**Ход работы**

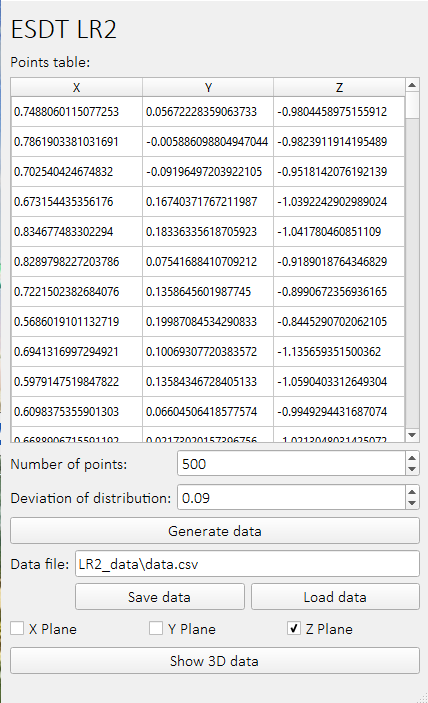
Для генерации данных на форме имеется возможность выбрать количество точек и среднеквадратическое отклонение координат точки от вычисленного положения.



Значения генерируются по заданной формуле, в данном случае, генерация координат выглядит следующим образом:

**for** i **in** range(points\_num):  
 self.points.append([0.7 \* math.cos(6 \* math.pi \* (i / points\_num)) +  
 random.normalvariate(0, deviation\_val),  
  
 0.5 \* math.sin(4 \* math.pi \* (i / points\_num)) +  
 random.normalvariate(0, deviation\_val),  
  
 -1 + 2 \* i / points\_num + random.normalvariate(0, deviation\_val)])

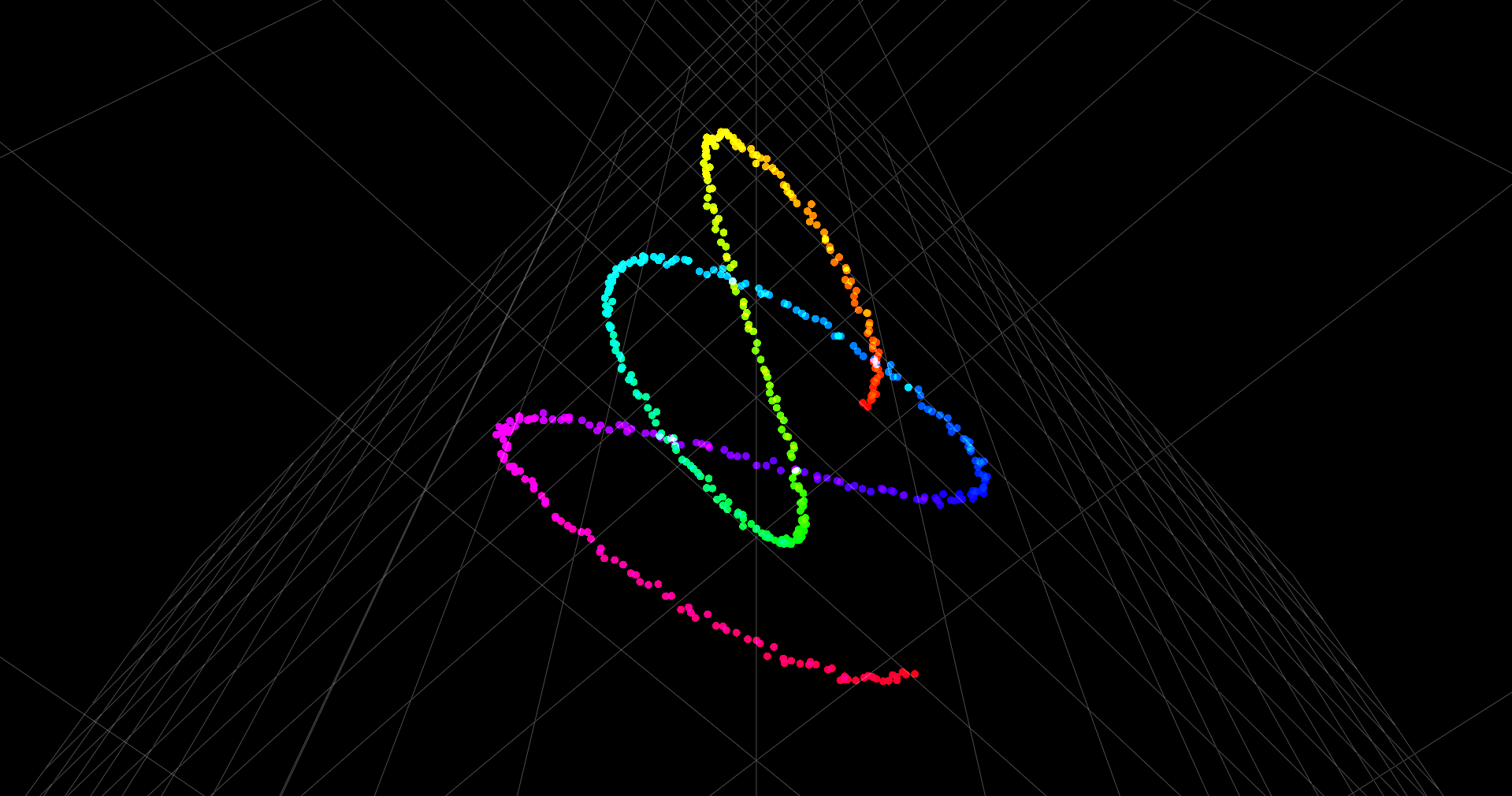
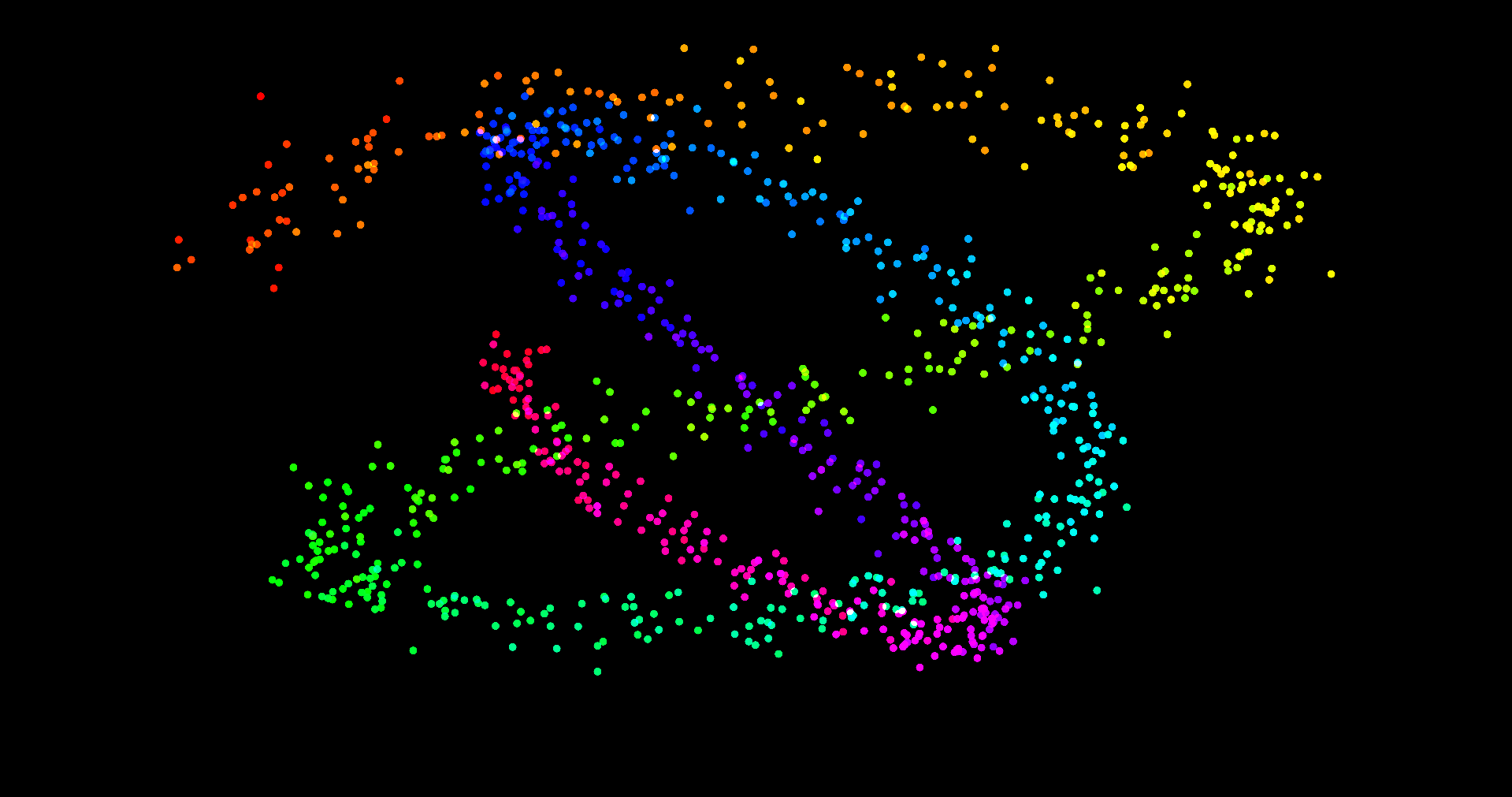
Сгенерированные данные хранятся в массиве. После генерации или загрузки из CSV-файла, координаты отображаются в таблице

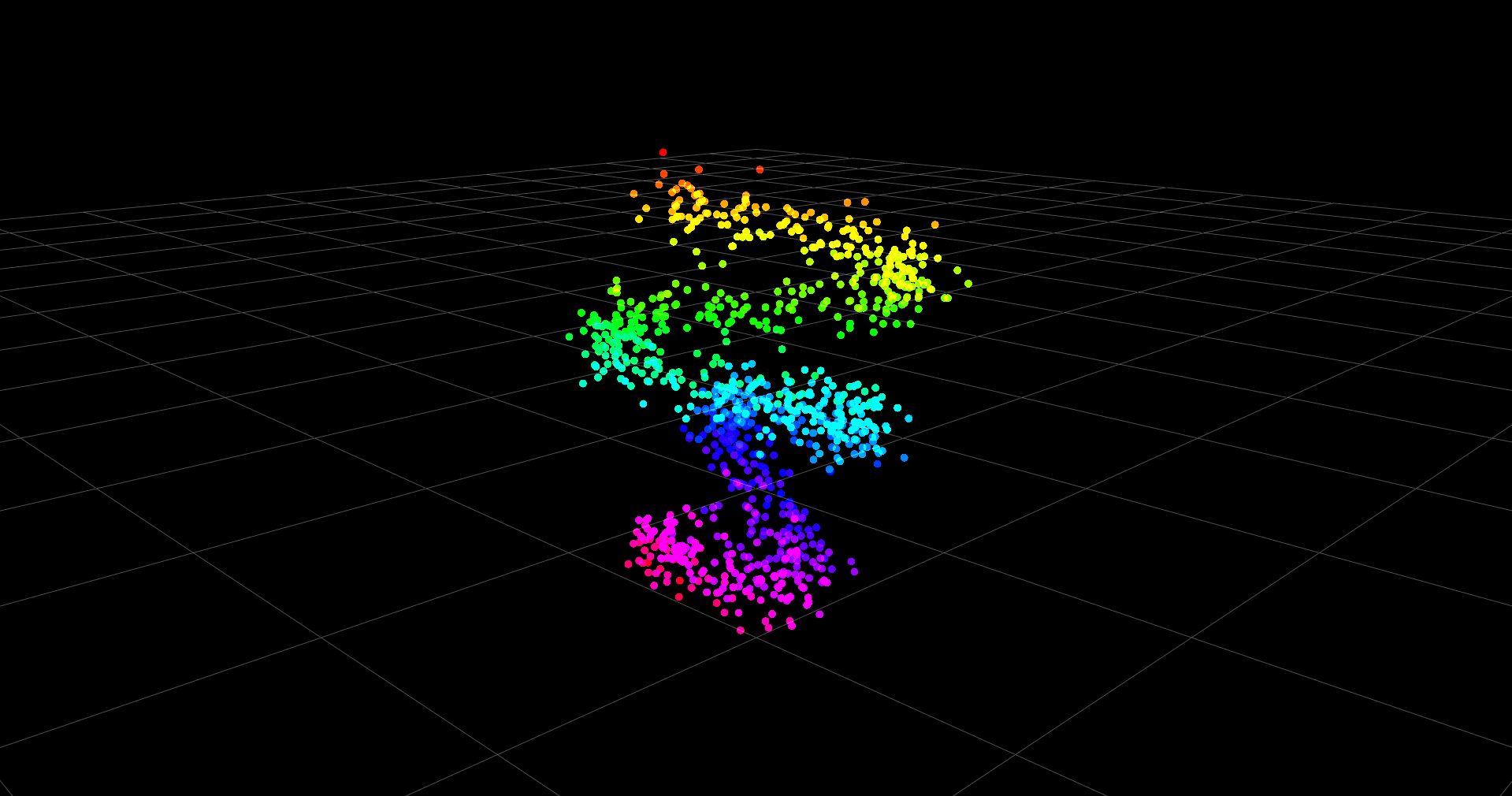


Также, сгенерированные данные можно сохранить в CSV-файл. В качестве разделителя используется символ пробела, как описано в задании.

С помощью библиотеки pyqtgraph.opengl имеется возможность отрисовки точек, а также координатных плоскостей (их можно включить на форме)

Результаты отрисовки выглядят следующим образом:





**Вывод**

В ходе данной работы было создано приложение на языке Python с использованием библиотек PyQTgraph, OpenGL и PyQT5, реализующее возможность генерации точек по заданному алгоритму, сохранение и загрузка из файла, а также отображение в 3D - пространстве.

**Исходный код (основный скрипт)**

**import** math  
**import** os  
**import** sys  
**import** random  
**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**import** pyqtgraph.opengl **as** gl  
**from** PyQt5 **import** uic  
**from** PyQt5.QtWidgets **import** QApplication, QMainWindow, QTableWidgetItem  
**from** PyQt5 **import** QtWidgets  
  
  
**class** Window(QMainWindow):  
 **def** \_\_init\_\_(self):  
 super(Window, self).\_\_init\_\_()  
 self.gui = uic.loadUi(**'LR2.ui'**)  
 self.gui.show()  
  
 self.gui.btn\_generate\_data.clicked.connect(self.generate\_data)  
 self.gui.btn\_save\_data.clicked.connect(self.save\_data)  
 self.gui.btn\_load\_data.clicked.connect(self.load\_data)  
 self.gui.btn\_show\_3d.clicked.connect(self.show\_3d)  
  
 self.gui.points\_table.setColumnCount(3)  
 self.gui.points\_table.verticalHeader().setVisible(**False**)  
 self.gui.points\_table.setEditTriggers(QtWidgets.QAbstractItemView.NoEditTriggers)  
 self.gui.points\_table.setHorizontalHeaderItem(0, QtWidgets.QTableWidgetItem(**'X'**))  
 self.gui.points\_table.setHorizontalHeaderItem(1, QtWidgets.QTableWidgetItem(**'Y'**))  
 self.gui.points\_table.setHorizontalHeaderItem(2, QtWidgets.QTableWidgetItem(**'Z'**))  
 header = self.gui.points\_table.horizontalHeader()  
 header.setSectionResizeMode(0, QtWidgets.QHeaderView.Stretch)  
 header.setSectionResizeMode(1, QtWidgets.QHeaderView.Stretch)  
 header.setSectionResizeMode(2, QtWidgets.QHeaderView.Stretch)  
  
 self.points = **None  
  
 def** generate\_data(self):  
 print(**'Generating data...'**)  
 points\_num = self.gui.points\_num.value()  
 deviation\_val = self.gui.deviation\_val.value()  
 self.points = []  
 **for** i **in** range(points\_num):  
 self.points.append([0.7 \* math.cos(6 \* math.pi \* (i / points\_num)) +  
 random.normalvariate(0, deviation\_val),  
  
 0.5 \* math.sin(4 \* math.pi \* (i / points\_num)) +  
 random.normalvariate(0, deviation\_val),  
  
 -1 + 2 \* i / points\_num + random.normalvariate(0, deviation\_val)])  
 self.points = np.array(self.points)  
  
 self.show\_on\_table()  
 self.gui.btn\_save\_data.setEnabled(**True**)  
 print(**'Data generated.'**)  
  
 **def** save\_data(self):  
 print(**'Saving data...'**)  
 np.savetxt(self.gui.data\_file.text(), self.points, delimiter=**' '**)  
 print(**'File'**, self.gui.data\_file.text(), **'saved.'**)  
  
 **def** load\_data(self):  
 **if** os.path.exists(self.gui.data\_file.text()):  
 print(**'Loading data...'**)  
 self.points = np.loadtxt(self.gui.data\_file.text(), delimiter=**' '**)  
 self.show\_on\_table()  
 print(**'File'**, self.gui.data\_file.text(), **'loaded.'**)  
 **else**:  
 print(**'File'**, self.gui.data\_file.text(), **'doesn\'t exist!'**)  
  
 **def** show\_on\_table(self):  
 self.gui.points\_table.setRowCount(0)  
 **for** point **in** self.points:  
 row\_position = self.gui.points\_table.rowCount()  
 self.gui.points\_table.insertRow(row\_position)  
 self.gui.points\_table.setItem(row\_position, 0, QTableWidgetItem(str(point[0])))  
 self.gui.points\_table.setItem(row\_position, 1, QTableWidgetItem(str(point[1])))  
 self.gui.points\_table.setItem(row\_position, 2, QTableWidgetItem(str(point[2])))  
 self.gui.btn\_save\_data.setEnabled(**True**)  
  
 **def** show\_3d(self):  
 **if** self.points **is not None and** len(self.points) > 0:  
 view = gl.GLViewWidget()  
 view.show()  
  
 *# Walls* gx = gl.GLGridItem()  
 gy = gl.GLGridItem()  
 gz = gl.GLGridItem()  
 gx.rotate(90, 0, 1, 0)  
 gy.rotate(90, 1, 0, 0)  
 gx.translate(0, 0, 0)  
 gy.translate(0, 0, 0)  
 gz.translate(0, 0, 0)  
 **if** self.gui.x\_plane.isChecked():  
 view.addItem(gx) *# X Plane* **if** self.gui.y\_plane.isChecked():  
 view.addItem(gy) *# Y Plane* **if** self.gui.z\_plane.isChecked():  
 view.addItem(gz) *# Z Plane* view.addItem(gl.GLLinePlotItem())  
  
 *# Color rainbow* z = np.array(np.array([item[2] **for** item **in** self.points]))  
 cmap = plt.get\_cmap(**'jet'**)  
 min\_z = np.min(z)  
 max\_z = np.max(z)  
 rgba\_img = cmap((z - min\_z) / (max\_z - min\_z))  
  
 *# Points cloud* surface = gl.GLScatterPlotItem(pos=self.points, color=rgba\_img)  
 view.addItem(surface)  
 **else**:  
 print(**'No points! Please generate or load.'**)  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 app = QApplication(sys.argv)  
 app.setStyle(**'fusion'**)  
 win = Window()  
 sys.exit(app.exec\_())

**Исходный код (LR2.ui)**

*<?***xml version="1.0" encoding="UTF-8"***?>*<**ui version="4.0"**>  
 <**class**>MainWindow</**class**>  
 <**widget class="QMainWindow" name="MainWindow"**>  
 <**property name="geometry"**>  
 <**rect**>  
 <**x**>0</**x**>  
 <**y**>0</**y**>  
 <**width**>428</**width**>  
 <**height**>704</**height**>  
 </**rect**>  
 </**property**>  
 <**property name="font"**>  
 <**font**>  
 <**family**>Calibri Light</**family**>  
 <**pointsize**>12</**pointsize**>  
 </**font**>  
 </**property**>  
 <**property name="windowTitle"**>  
 <**string**>ESDT LR2</**string**>  
 </**property**>  
 <**widget class="QWidget" name="centralwidget"**>  
 <**layout class="QVBoxLayout" name="verticalLayout\_3"**>  
 <**item**>  
 <**widget class="QLabel" name="label"**>  
 <**property name="font"**>  
 <**font**>  
 <**pointsize**>22</**pointsize**>  
 </**font**>  
 </**property**>  
 <**property name="text"**>  
 <**string**>ESDT LR2</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 <**item**>  
 <**widget class="QLabel" name="label\_4"**>  
 <**property name="text"**>  
 <**string**>Points table:</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 <**item**>  
 <**widget class="QTableWidget" name="points\_table"**>  
 <**property name="sortingEnabled"**>  
 <**bool**>true</**bool**>  
 </**property**>  
 <**attribute name="horizontalHeaderShowSortIndicator" stdset="0"**>  
 <**bool**>true</**bool**>  
 </**attribute**>  
 <**attribute name="verticalHeaderShowSortIndicator" stdset="0"**>  
 <**bool**>false</**bool**>  
 </**attribute**>  
 </**widget**>  
 </**item**>  
 <**item**>  
 <**layout class="QVBoxLayout" name="verticalLayout\_2"**>  
 <**item**>  
 <**layout class="QFormLayout" name="formLayout"**>  
 <**item row="0" column="0"**>  
 <**widget class="QLabel" name="label\_2"**>  
 <**property name="text"**>  
 <**string**>Number of points:</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 <**item row="0" column="1"**>  
 <**widget class="QSpinBox" name="points\_num"**>  
 <**property name="minimum"**>  
 <**number**>1</**number**>  
 </**property**>  
 <**property name="maximum"**>  
 <**number**>99999</**number**>  
 </**property**>  
 <**property name="value"**>  
 <**number**>1000</**number**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 <**item row="1" column="0"**>  
 <**widget class="QLabel" name="label\_3"**>  
 <**property name="text"**>  
 <**string**>Deviation of distribution:</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 <**item row="1" column="1"**>  
 <**widget class="QDoubleSpinBox" name="deviation\_val"**>  
 <**property name="value"**>  
 <**double**>0.050000000000000</**double**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 </**layout**>  
 </**item**>  
 <**item**>  
 <**widget class="QPushButton" name="btn\_generate\_data"**>  
 <**property name="text"**>  
 <**string**>Generate data</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 </**layout**>  
 </**item**>  
 <**item**>  
 <**layout class="QGridLayout" name="gridLayout\_4"**>  
 <**item row="1" column="1"**>  
 <**widget class="QLineEdit" name="data\_file"**>  
 <**property name="text"**>  
 <**string**>LR2\_data\data.csv</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 <**item row="1" column="0"**>  
 <**widget class="QLabel" name="label\_19"**>  
 <**property name="text"**>  
 <**string**>Data file:</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 <**item row="2" column="1"**>  
 <**layout class="QGridLayout" name="gridLayout\_5"**>  
 <**item row="0" column="2"**>  
 <**widget class="QPushButton" name="btn\_load\_data"**>  
 <**property name="text"**>  
 <**string**>Load data</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 <**item row="0" column="1"**>  
 <**widget class="QPushButton" name="btn\_save\_data"**>  
 <**property name="enabled"**>  
 <**bool**>false</**bool**>  
 </**property**>  
 <**property name="text"**>  
 <**string**>Save data</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 </**layout**>  
 </**item**>  
 </**layout**>  
 </**item**>  
 <**item**>  
 <**layout class="QVBoxLayout" name="verticalLayout"**>  
 <**item**>  
 <**layout class="QHBoxLayout" name="horizontalLayout"**>  
 <**item**>  
 <**widget class="QCheckBox" name="x\_plane"**>  
 <**property name="text"**>  
 <**string**>X Plane</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 <**item**>  
 <**widget class="QCheckBox" name="y\_plane"**>  
 <**property name="text"**>  
 <**string**>Y Plane</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 <**item**>  
 <**widget class="QCheckBox" name="z\_plane"**>  
 <**property name="text"**>  
 <**string**>Z Plane</**string**>  
 </**property**>  
 <**property name="checked"**>  
 <**bool**>true</**bool**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 </**layout**>  
 </**item**>  
 <**item**>  
 <**widget class="QPushButton" name="btn\_show\_3d"**>  
 <**property name="text"**>  
 <**string**>Show 3D data</**string**>  
 </**property**>  
 </**widget**>  
 </**item**>  
 </**layout**>  
 </**item**>  
 </**layout**>  
 </**widget**>  
 <**widget class="QMenuBar" name="menubar"**>  
 <**property name="geometry"**>  
 <**rect**>  
 <**x**>0</**x**>  
 <**y**>0</**y**>  
 <**width**>428</**width**>  
 <**height**>21</**height**>  
 </**rect**>  
 </**property**>  
 </**widget**>  
 <**widget class="QStatusBar" name="statusbar"**/>  
 </**widget**>  
 <**resources**/>  
 <**connections**/>  
</**ui**>