```
#maya/Qt_Dev/matrix_sliders_ui.py

#save this file as "matrix_sliders_ui.py" before using it

"""
import sys
sys.path.append("$USER/Documents/maya/Qt_Dev")

import matrix_sliders_ui
reload(matrix_sliders_ui)

mayaWin = matrix_sliders_ui.getMayaMainWindow()
dialog = matrix_sliders_ui.MatrixDialog(mayaWin)
"""
# Run this script in Maya Script Editor.
#Aternatively, run this script directly from Cutter.

isMaya = False
try:
    from PyQt4.QtCore import *
    from PyQt4.QtGui import *
    from PyQt4 import uic
except ImportError:
    from PySide2.QtGui import *
    from PySide2.QtWidgets import *
    from PySide2.QtCore import *
    from PySide2.QtUiTools import *
    isMaya = True
    import maya.OpenMayaUI as omui
    import shiboken2
    from windmill import makeWindmill
    from handle import makeCenter
    from bubbles import makeBubbles

import sys
import os
import math
from connection_utils import *
import maya.cmds as cmds
from random import uniform

def getMayaMainWindow():
    winPtr = omui.MQtUtil.mainWindow()
```

```
    return shiboken2.wrapInstance(long(winPtr), QWidget)

#_____________________________________________________
#                  Widget Names
#_____________________________________________________
#    bladeNum_SpinBox       bladeNum_Slider
#    ringNum_SpinBox        ringNum_Slider
#    ringRad_DSpinBox       ringRad_Slider
#    ringHeight_DSpinBox    ringHeight_Slider
#    ringThickness_DSpinBox  ringThickness_Slider
#    ringSubdiv_DSpinBox    ringsubdiv_Slider
#    bubbleNum_SpinBox      bubbleNum_Slider
#    bubbleRad_DSpinBox        bubbleRad_Slider
#    bubbleMaxDist_DSpinBox  bubbleMaxDist_Slider
#    BubbleRadrange_lineEdit
#
#                  doit_Button
#_____________________________________________________
class MatrixDialog(QDialog):
    def __init__(self, parent=None):
        super(MatrixDialog, self).__init__(parent)
        pathToUi = os.path.join(os.path.dirname(__file__), 'ui', 'matrix_sliders.ui')
        self.iconPath = os.path.join(os.path.dirname(__file__), 'icons',
'melMatrix.jpg')
        if isMaya == False:
            self.ui = uic.loadUi(pathToUi,self)
            self.setWindowFlags(Qt.WindowStaysOnTopHint)
        else:
            loader = QUiLoader()
            self.ui = loader.load(pathToUi, self)
            self.setWindowFlags(self.windowFlags() | Qt.WindowStaysOnTopHint)
        self.makeConnections()
        self.ui.show()
    #_____________________________________________
    def makeConnections(self):
        pixmap = QPixmap(self.iconPath)
        self.ui.graphic_QLabel.setPixmap(pixmap)
        self.ui.graphic_QLabel.setMask(pixmap.mask())

        self.bladeNum = IntSlider(self.ui.bladeNum_SpinBox,
self.ui.bladeNum_Slider)
        self.ringNum = IntSlider(self.ui.ringNum_SpinBox, self.ui.ringNum_Slider)
        self.ringRad = FloatSlider(self.ui.ringRad_DSpinBox, self.ui.ringRad_Slider)
```

```
        self.ringThickness = FloatSlider(self.ui.ringThickness_DSpinBox,
self.ui.ringThickness_Slider)
        self.ringSubdiv = IntSlider(self.ui.ringSubdiv_SpinBox,
self.ui.ringSubdiv_Slider)
        self.bubbleNum = IntSlider(self.ui.bubbleNum_SpinBox,
self.ui.bubbleNum_Slider)
        self.bubbleMaxDist = FloatSlider(self.ui.bubbleMaxDist_DSpinBox,
self.ui.bubbleMaxDist_Slider)

        self.ui.setFixedSize(500, 615)
        self.ui.doit_Button.clicked.connect(self.doitAction)
#_____________________________________________
def doitAction(self):
        print('Making the Matrix')
        makeWindmill(self.bladeNum.getValue())
        cmds.xform( ws=True, piv=(0, 0, 0) )
        cmds.move(-1.6, 1.43, 0)
        cmds.rotate(0, 0, '-90deg')

        makeCenter(self.ringNum.getValue(), self.ringRad.getValue()/100,
self.ringHeight.getValue()/100, self.ringThickness.getValue()/100,
self.ringSubdiv.getValue())
        cmds.rotate( 0, 0, '90deg', 'center1' )
        cmds.move(-3.1, 1.5, 0, 'center1')
        cmds.duplicate( 'center1', n='center2' )
        cmds.move(1.4, 1.5, 0)
        cmds.polyCylinder(n='center_pole', r=1)
        cmds.setAttr('center_pole.scaleY', 2.7)
        cmds.rotate( 0, 0, '90deg', 'center_pole' )
        cmds.move(-0.8, 1.5, 0, 'center_pole')
        cmds.polyCylinder(n='pole', r=0.4, h=20)
        cmds.move(-2.5, -8, 0, 'pole')
        cmds.group('center1', 'center2', 'center_pole', 'pole', n='handle')

        makeBubbles(self.bubbleNum.getValue(), self.bubbleMaxDist.getValue()/50,
[0.5, 2])

#=======================================================
if __name__ == '__main__':
    app = QApplication(sys.argv)
    dialog = MatrixDialog()
    dialog.show()
    sys.exit(app.exec_())
```