```
'''
Created on Jul 16, 2011

@author: neugebauer
'''
from PyQt4 import QtGui, QtCore
from kmcHamiltonViewerUI import Ui_KMCHamiltonViewer_GUIClass as Dlg
#from databaseUtilities import *
from KMCHamilton import atomKMC, KMC_control
import StructureInspectorGUI
import utilities as u
import threading,time

class Timer(threading.Thread):
    def __init__(self, interval, routine):
        threading.Thread.__init__(self)
        self.interval = interval
        self.routine = routine

    def run(self):
        time.sleep(self.interval)
        self.routine()


class MainWindow(QtGui.QWidget, Dlg):
    def __init__(self, kmcHamilton, parent = None):
        print "parent", parent
        QtGui.QWidget.__init__(self,parent)
        self.setupUi(self)
        self.hamilton = kmcHamilton

        self.labelNumAtoms.setText("#Atoms: " + str(len(self.hamilton.structure)))
        if self.hamilton.hasProjectID():
            self.struct0 = self.hamilton.getStructure(1)
            self.timeSteps = len(self.hamilton.timeList)
            self.labelProjectName.setText("Project: " + self.hamilton.getProjectName())
            self.labelProjectID.setText("ID: " + str(self.hamilton.getProjectID()))
            temperature = self.hamilton.getItem("temperature")
            self.labelTemperature.setText("T=" + str(temperature))
            self.labelIterations.setText("#Steps: " + str(self.timeSteps))
            self.horizontalSliderStep.setTickInterval(self.timeSteps)

        self.select = "Atomic Structure"
        self.comboBoxPlotType.insertItem(0, "Atomic Structure")
        self.comboBoxPlotType.insertItem(1, "Cluster analysis")
#        self.comboBoxPlotType.insertItem(2, "Time vs iterations")
# TODO: evolution of average cluster size

        self.index = 0
        self.sleep_time = 0.3
        self.step_distance = 1

# connections
        self.connect(self.pushButtonClose, QtCore.SIGNAL("clicked(bool)"),
                     self.exitAction)
        self.connect(self.pushButtonEdit, QtCore.SIGNAL("clicked(bool)"),
                    self.editHam)
        self.connect(self.pushButtonInitialStructure, QtCore.SIGNAL("clicked(bool)"),
                     self.showInitialStructure)
        self.connect(self.pushButtonFinalStructure, QtCore.SIGNAL("clicked(bool)"),
                     self.showFinalStructure)
        self.connect(self.pushButtonStartAnimation, QtCore.SIGNAL("clicked(bool)"),
                     self.startAnimation)
        self.connect(self.horizontalSliderStep, QtCore.SIGNAL("valueChanged(int)"),
                     self.sliderChanged)
        self.connect(self.pushButtonViewLog, QtCore.SIGNAL("clicked(bool)"),
                     self.viewLogfile)
        self.connect(self.comboBoxPlotType, QtCore.SIGNAL("currentIndexChanged(int)"),
                     self.showPlotType)

    def runAnalysis(self, id = 0):
#        self.hamilton.plot2d(iStep = id, pltExt = self.mplwidget)
        if self.select == "Cluster analysis":
            self.hamilton.plotHistogram(iStep = id, pltExt = self.mplwidget)
        elif self.select ==  "Atomic Structure":
            self.hamilton.plot2d(iStep = id, pltExt = self.mplwidget)

    def exitAction(self):
        self.enabled = False
        self.close()

    def editHam(self):
        print "to be implemented"

    def showInitialStructure(self):
        self.dialog = StructureInspectorGUI.MainWindow(self.struct0)
        self.dialog.show()

    def showFinalStructure(self):
        self.dialog = StructureInspectorGUI.MainWindow(self.hamilton.structure)
        self.dialog.show()

    def startAnimation(self, status):
        print "status: ", status
        if status:
            self.pushButtonStartAnimation.setText("Stop Animation")
            self.enabled = True
            timer = Timer(self.sleep_time, self.animate)
            timer.start()
        else:
            self.pushButtonStartAnimation.setText("Start Animation")
            self.enabled = False

    def animate(self):
        self.index = (self.index + self.step_distance) % self.timeSteps
        print "index", self.index
        self.runAnalysis(id = self.index)
        self.labelStep.setText("step: " + str(self.index))
        self.horizontalSliderStep.setSliderPosition(self.index)
        if self.enabled:
            timer = Timer(self.sleep_time, self.animate)
            timer.start()

    def sliderChanged(self, ind):
        self.runAnalysis(id = ind)
        self.labelStep.setText("step: " + str(ind))

    def viewLogfile(self):
        print "implement"

    def showPlotType(self, selIndex):
        self.select = self.comboBoxPlotType.itemData(selIndex,2).toString()
        print "selected: ", self.select
        self.runAnalysis(id = self.index)


#        if selected == "Cluster analysis":
#            self.hamilton.plotHistogram(iStep = self.index, pltExt = self.mplwidget)
#        elif selected ==  "Atomic Structure":
#            self.hamilton.plot2D(iStep = self.index, pltExt = self.mplwidget)

#from hamilton import AtomHamilton
if __name__ == '__main__':
#    import pickle

    dictProject = {"Name": "KMC56", "ID": 2}
    fileName = u.DumpDir("KMC56")
    kmc_control = u.load(fileName)

    kmc = atomKMC(dictProject = dictProject)
    kmc.showGUI()
```