Actividad 06 - UI: QPlainTextEdit

JAIRO CAIN SANCHEZ ESTRADA// Luis Angel Elisea Graciano

SEMINARIO DE SOLUCION DE PROBLEMAS DE ALGORITMIA

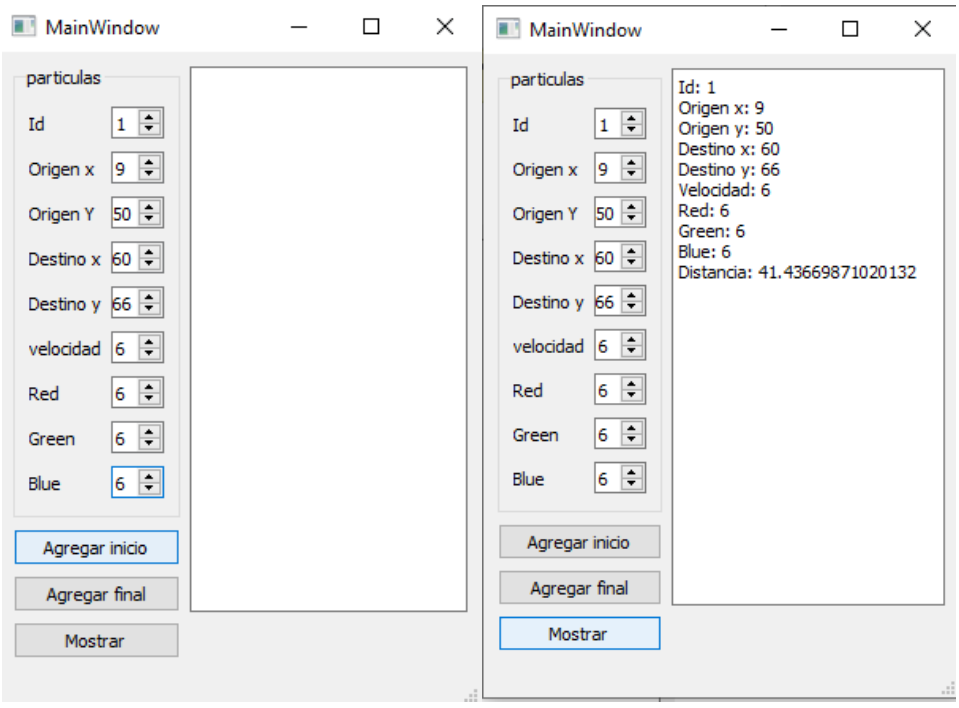
Lineamientos de evaluación

- Se agrego a la interfaz los botones de agregar inicio, agregar final, mostrar y el PlainTextEdit para mostrar la información de las particulas correctamente
- Se muestra la información de las particulas en la interfaz sin problema alguno
- El programa funciona sin problema alguno, al momento de crear los objetos estos funcionan de manera correcta
- Al momento de importar las clases y funciones, estas trabajan sin ningún problema y se realiza las funciones esperadas
- Se muestran en pantalla los elementos de cada partícula sin problemas
- Se calcula la distancia de cada partícula sin problema alguno
- Se almacenan partículas y se insertan en el inicio o final de la lista

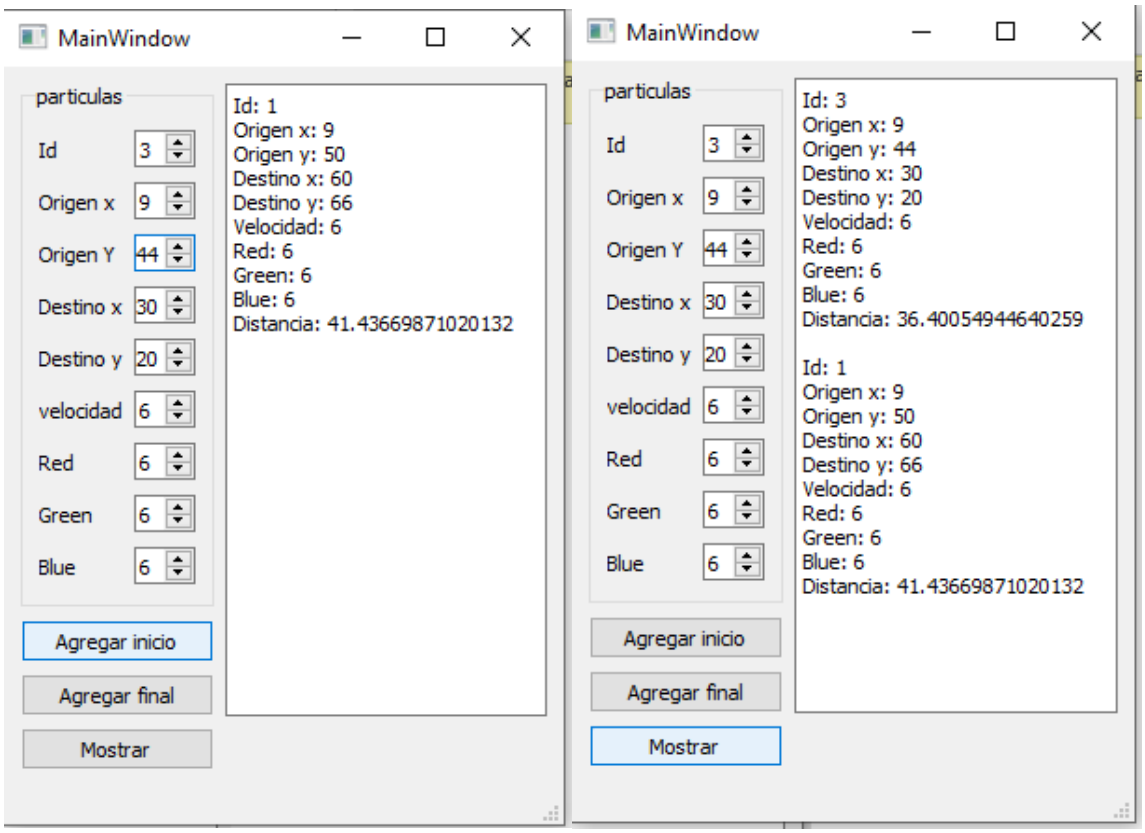
Desarrollo

Implementamos en la interfaz grafica los botones para agregar al inicio, al final y para poder mostrar la información de las particulas y conectamos con nuestras clases desarrolladas en la actividad anterior.

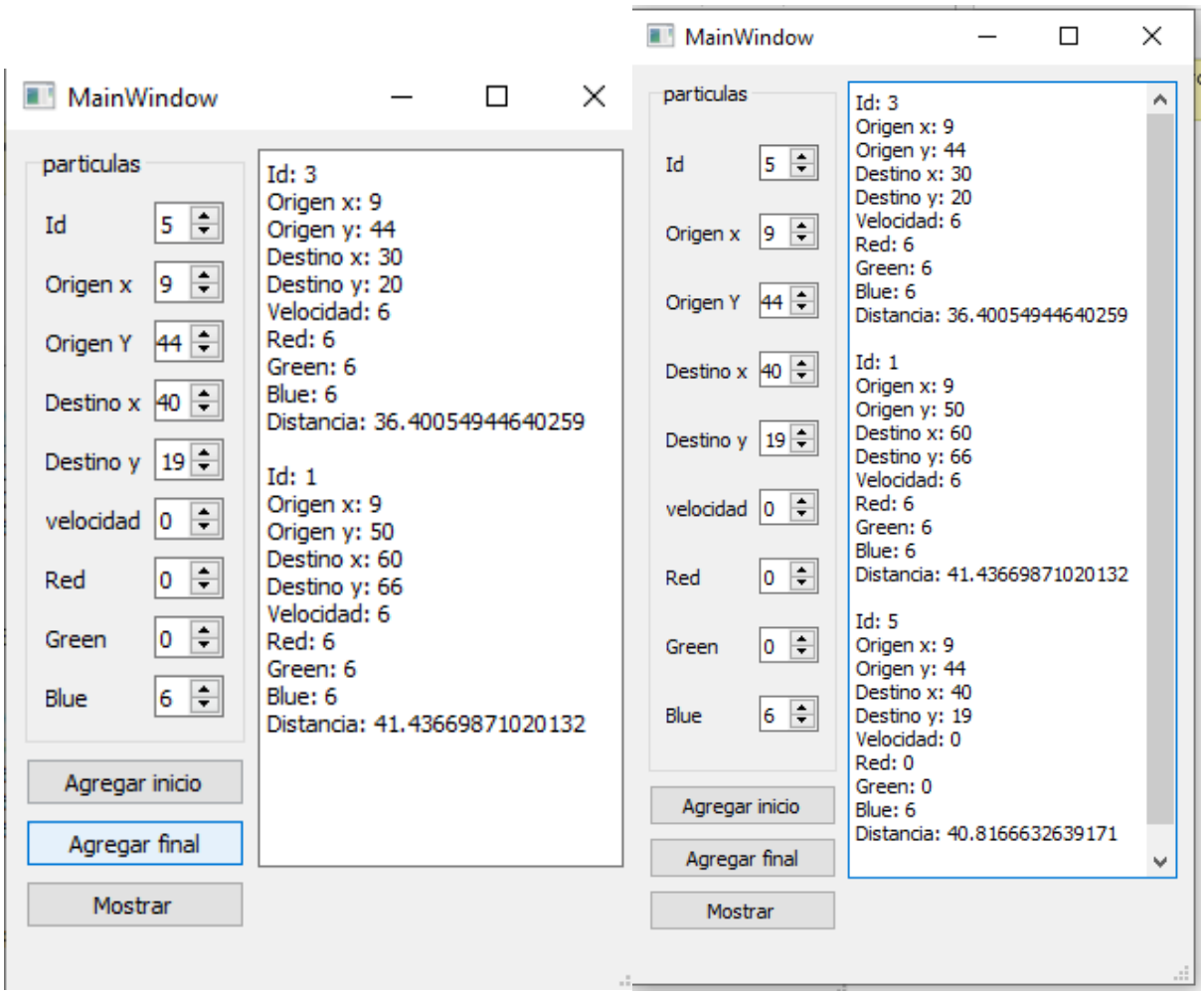
Agregamos una particula al inicio y después la mostramos:



Agregamos otra particula al inicio y después la mostramos:



Agregamos otra particula al final y después la mostramos:



Conclusiones

Esta practica me soluciona una gran duda, la cual era como íbamos a mostrar la información en la interfaz grafica, aparte de que me gusto saber como complementar nuestro código ya echo para que fuera compatible con la interfaz.

Referencias

Boites, M. D. [UC3jaAJ6X3vMZJKpi9KAcY0w]. (2020b, October 14). PySide2 - QPlainTextEdit (Qt for Python)(III). Youtube. https://www.youtube.com/watch?v=5TPKrKlAAU0

Código Particula.py

```
from algoritmos import distancia_euclidiana
class Particula:
    def
__init__(self,id=0,origen_x=0,origen_y=0,destino_x=0,destino_y=0,velocidad=0,red=0,
green=0,blue=0,distancia=0.0):
        self.__id=id
        self.__origen_x=origen_x
        self.__origen_y=origen_y
        self.__destino_x=destino_x
        self.__destino_y=destino_y
        self.__velocidad=velocidad
        self.___red=red
        self.___green=green
        self.__blue=blue
        self.__distancia=distancia_euclidiana(origen_x, destino_x, origen_y,
destino_y)

    def __str__(self):
        return(
            'Id: '+ str(self.__id) + '\n' +
            'Origen x: '+ str(self.__origen_x) + '\n' +
            'Origen y: '+ str(self.__origen_y) + '\n' +
            'Destino x: '+ str(self.__destino_x) + '\n' +
            'Destino y: '+ str(self.__destino_y) + '\n' +
            'Velocidad: '+ str(self.__velocidad) + '\n' +
            'Red: '+ str(self.___red) + '\n' +
            'Green: '+ str(self.___green) + '\n' +
            'Blue: '+ str(self.__blue) + '\n' +
            'Distancia: '+ str(self.__distancia) + '\n'
        )
```

Algoritmos.py

```python
from math import sqrt
def distancia_euclidiana(x_1, y_1, x_2, y_2):
    """ Calcula la distancia euclidiana
    Devuelve el resultado de la fórmula
    También se le conoce a la fórmula como:
    distancia entre dos puntos
    Parámetros:
    x_1 -- origen_x
    y_1 -- origen_y
    x_2 -- destino_x
    y_2 -- destino_y
    """
    return(sqrt(((x_2 - x_1)**2) + ((y_2 - y_1)**2)))
```

particulas.py

```python
from particula import Particula
class Particulas:
    def __init__(self):
        self.__particulas = []

    def agregar_inicio(self,particula:Particula):
        self.__particulas.insert(0,particula)

    def agregar_final(self,particula:Particula):
        self.__particulas.append(particula)

    def mostrar(self):
        for particula in self.__particulas:
            print(particula)

    def __str__(self):
        return "".join(
            str(particula) + '\n' for particula in self.__particulas
        )
```

mainwindow.py

```python
from PySide2.QtWidgets import QMainWindow
from PySide2.QtCore import Slot
from ui_mainwindow import Ui_MainWindow
from particula import Particula
```

```
from particulas import Particulas

#pyside2-uic mainwindow.ui para pasar de .ui a python
class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()

        self.particulas= Particulas()

        self.ui=Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.Agregar_final_pushButton.clicked.connect(self.click_agregar)
        self.ui.Agregar_inicio_pushButton.clicked.connect(self.click_agregar_inicio
)
        self.ui.Mostrar_pushButton.clicked.connect(self.click_mostrar)

    @Slot()
    def click_mostrar(self):
        self.ui.salida.clear()
        self.ui.salida.insertPlainText(str(self.particulas))

    @Slot()
    def click_agregar(self):
        id=self.ui.Id_spinBox.text()
        origen_x=self.ui.Origen_x_spinBox.value()
        origen_y=self.ui.Origen_y_spinBox.value()
        destino_x=self.ui.Destino_x_spinBox.value()
        destino_y=self.ui.Destino_y_spinBox.value()
        velocidad=self.ui.Velocidad_spinBox.value()
        red=self.ui.Red_spinBox.value()
        green=self.ui.Green_spinBox.value()
        blue=self.ui.Blue_spinBox.value()

        Particula1=Particula(id,origen_x,origen_y,destino_x,destino_y,velocidad,red
,green,blue)
        self.particulas.agregar_final(Particula1)

    @Slot()
    def click_agregar_inicio(self):
        id=self.ui.Id_spinBox.text()
        origen_x=self.ui.Origen_x_spinBox.value()
        origen_y=self.ui.Origen_y_spinBox.value()
        destino_x=self.ui.Destino_x_spinBox.value()
        destino_y=self.ui.Destino_y_spinBox.value()
        velocidad=self.ui.Velocidad_spinBox.value()
        red=self.ui.Red_spinBox.value()
```

```
        green=self.ui.Green_spinBox.value()
        blue=self.ui.Blue_spinBox.value()

        Particula1=Particula(id,origen_x,origen_y,destino_x,destino_y,velocidad,red
,green,blue)
        self.particulas.agregar_inicio(Particula1)
```

Main.py

```
from PySide2.QtWidgets import QApplication
from  mainwindow import MainWindow
#pyside2-uic mainwindow.ui para pasar de .ui a python
#pyside2-uic mainwindow.ui >ui_mainwindow.py
import sys
app=QApplication()
window=MainWindow()
window.show()
sys.exit(app.exec_())
```