

Actividad 06- QPlainTextEdit

Ramirez Orduna Geovanni

Lineamiento de evaluación

- [] El reporte está en formato Google Docs o PDF.
- [] El reporte sigue las pautas del Formato de Actividades .
- [] El reporte tiene desarrollada todas las pautas del Formato de Actividades.
- [] Se muestra la captura de pantalla de los datos antes de usar el botón para agregar_inicio() y la captura de pantalla de mostrar partículas en el QPlainTextEdit después de haber agregado la Partícula.
- [] Se muestra la captura de pantalla de los datos antes de usar el botón para agregar_final() y la captura de pantalla de mostrar partículas en el QPlainTextEdit después de haber agregado la Partícula.

DESARROLLO

tenemos el main principal

```
main.py > ...
1  from PySide2.QtWidgets import QApplication
2  from mainWindow import MainWIndow
3  import sys
4
5  app=QApplication()
6
7  window=MainWIndow()
8  window.show()
9  sys.exit(app.exec_())
10
11
12
```

tenemos admin que importa la clase algoritmos para calcular la distancia

```
admin.py > Admin > __str__
1  from particula import Particula
2  from algoritmos import distacia_euclidiana
3  class Admin:
4      def __init__(self):
5          self.__almacen=[]
6
7
8      def agregar_final(self, particula:Particula):
9          self.__almacen.append(particula)
10
11      def agregar_inicio(self, particula:Particula):
12          self.__almacen.insert(0, particula)
13
14      def mostrar(self):
15          for particula in self.__almacen:
16              print(particula)
17
18      def __str__(self):
19          return "".join(
20              str(admin)+ "\n" for admin in self.__almacen )
21
22
23
24
25
```

Tenemos la clase partícula

```
particula.py > Particula > _str_
1 from algoritmos import distancia_euclidiana
2
3
4 class Particula:
5     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):
6         self.__id=id
7         self.__origen_x=origen_x
8         self.__origen_y=origen_y
9         self.__destino_x= destino_x
10        self.__destino_y=destino_y
11        self.__velocidad=velocidad
12        self.__red=red
13        self.__green=green
14        self.__blue=blue
15        self.__distancia=distancia_euclidiana(origen_x,origen_y,destino_x,destino_y)
16
17        #convertir todos los atributos a un string para poderlos imprimir
18    def __str__(self):
19        return(
20            'id: ' + str(self.__id) + '\n'+
21            'origen x: ' + str(self.__origen_x) + '\n'+
22            'origen y: ' + str(self.__origen_y) + '\n'+
23            'destino x: ' + str(self.__destino_x) + '\n'+
24            'destino y: ' + str(self.__destino_y) + '\n'+
25            'velocidad: ' + str(self.__velocidad) + '\n'+
26            'red: ' + str(self.__red) + '\n'+
27            'green: ' + str(self.__green) + '\n'+
28            'blue: ' + str(self.__blue) + '\n'+
29            'distancia: ' + str(self.__distancia) + '\n'
30        )
31
```

tenemos el mainWindow donde se desarrollo la conexión con la interfaz grafica

```
mainWindow.py > ...
1 import PySide2
2 from PySide2.QtWidgets import QMainWindow
3 from PySide2.QtCore import Slot
4 from ui_mainwindow import Ui_MainWindow
5 from admin import Admin
6 from particula import Particula
7 from algoritmos import distancia_euclidiana
8
9 class MainWindow(QMainWindow):
10     def __init__(self):
11         super(MainWindow, self).__init__()
12
13         self.admin=Admin()
14
15         self.ui=Ui_MainWindow()
16         self.ui.setupUi(self) ##mete la intrerfaz
17         self.ui.Agre_Final_button.clicked.connect(self.click_agregar) ##conectar clase a boton
18         self.ui.Agre_Inicio_Button.clicked.connect(self.click_agregar_inicio)
19         self.ui.Mostrar_button.clicked.connect(self.click_mostrar)
20
21     @Slot()
22     def click_agregar(self):
23         id=self.ui.Id_spinBox.value()
24         OrigenX=self.ui.OrigenX_spinBox.value()
25         OrigenY=self.ui.OrigenY_spinBox.value()
26         DestinoX=self.ui.DestinoX_spinBox.value()
27         DestinoY=self.ui.DestinoY_spinBox.value()
28         Velocidad=self.ui.Velocidad_spinBox.value()
29         Red=self.ui.Red_spinBox.value()
30         Green=self.ui.Green_spinBox.value()
31         Blue=self.ui.Blue_spinBox.value()
32         distancia=distancia_euclidiana
33
34         particula=Particula(id,OrigenX,OrigenY,DestinoX,DestinoY,Velocidad, Red,Green,Blue,distancia)
35         self.admin.agregar_final(particula)
36
37         #print(OrigenX,OrigenY, DestinoX, DestinoY, Velocidad, Red, Green, Blue)
38         #self.ui.salida.insertPlainText(str(OrigenX) + str(OrigenY) + str(DestinoX) + str(DestinoY) + str(Velocidad) + str(Red) + str(Green) + str(Blue))
39
40     @Slot()
41     def click_agregar_inicio(self):
42         id=self.ui.Id_spinBox.value()
43         OrigenX=self.ui.OrigenX_spinBox.value()
44         OrigenY=self.ui.OrigenY_spinBox.value()
45         DestinoX=self.ui.DestinoX_spinBox.value()
46         DestinoY=self.ui.DestinoY_spinBox.value()
47         Velocidad=self.ui.Velocidad_spinBox.value()
48         Red=self.ui.Red_spinBox.value()
49         Green=self.ui.Green_spinBox.value()
```

tenemos la interfaz corriendo con una captura de datos

MainWindow

Particulas

ID

1

Origen X

3

Origen Y

23

Destino x:

23

Destino y:

3

Velocidad:

4

Rojo:

1

Verde:

1

Azul:

1

Agregar Final

Agregar inicio

Mostrar

tenemos la impresión de los datos

The screenshot shows a Qt application window titled "MainWindow". The window contains a form for entering particle data and a list of calculated particles.

Form Fields:

- ID: 3
- Origen X: 56
- Origen Y: 10
- Destino x: 9
- Destino y: 3
- Velocidad: 89
- Rojo: 1
- Verde: 1
- Azul: 1

Buttons:

- Agregar Final
- Agregar inicio
- Mostrar

Particle Data List:

```
id: 2
origen x: 3
origen y: 6
destino x: 9
destino y: 3
velocidad: 4
red: 1
green: 1
blue: 1
distancia: 6.708203932499369

id: 1
origen x: 3
origen y: 23
destino x: 23
destino y: 3
velocidad: 4
red: 1
green: 1
blue: 1
distancia: 28.284271247461902

id: 1
origen x: 3
origen y: 23
destino x: 23
destino y: 3
velocidad: 4
red: 1
green: 1
blue: 1
distancia: 28.284271247461902
```

CONCLUSIONES:

Tenemos la conexión de la interfaz gráfica para poder agregar datos de la clase particular así como calcular la distancia.

Referencias:

https://www.youtube.com/watch?v=5TPKrKIAAU0&feature=emb_title

Código

```
import PySide2
from PySide2.QtWidgets import QMainWindow
from PySide2.QtCore import Slot
from ui_mainwindow import Ui_MainWindow
from admin import Admin
from particula import Particula
from algoritmos import distancia_euclidiana

class MainWIndow(QMainWindow):
    def __init__(self):
        super(MainWIndow, self).__init__()

        self.admin=Admin()

        self.ui=Ui_MainWindow()
        self.ui.setupUi(self) ##mete la intrerfaz

self.ui.Agre_Final_button.clicked.connect(self.click_agregar)
##conectar clase a boton

self.ui.Agre_Inicio_Button.clicked.connect(self.click_agregar_inicio)

        self.ui.Mostrar_button.clicked.connect(self.click_mostrar)

    @Slot()
    def click_agregar(self):
        id=self.ui.Id_spinBox.value()
        OrigenX=self.ui.OrigenX_spinBox.value()
        OrigenY=self.ui.OrigenY_spinBox.value()
        DestinoX=self.ui.DestinoX_spinBox.value()
        DestinoY=self.ui.DestinoY_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()
        distancia=distancia_euclidiana

particula=Particula(id,OrigenX,OrigenY, DestinoX, DestinoY, Velocidad
, Red, Green, Blue, distancia)

        self.admin.agregar_final(particula)
```

```

        #print(OrigenX,OrigenY,
DestinoX,DestinoY,Velocidad,Red,Green,Blue)

        #self.ui.salida.insertPlainText(str(OrigenX) + str(
OrigenY) + str(DestinoX) + str (DestinoY) + str (Velocidad) +str(
Red)+ str(Green) + str(Blue))

    @Slot()
    def click_agregar_inicio(self):
        id=self.ui.Id_spinBox.value()
        OrigenX=self.ui.OrigenX_spinBox.value()
        OrigenY=self.ui.OrigenY_spinBox.value()
        DestinoX=self.ui.DestinoX_spinBox.value()
        DestinoY=self.ui.DestinoY_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()
        distancia=distacia_euclidiana

particula=Particula(id,OrigenX,OrigenY, DestinoX, DestinoY, Velocidad
, Red,Green,Blue,distancia)

        self.admin.agregar_inicio(particula)

    @Slot()
    def click_mostrar(self):
        # self.admin.mostrar()
        self.ui.salida.clear()
        self.ui.salida.insertPlainText(str(self.admin))

```

