

HW 6 GUI-pyqt5

Chen Ren/1665951

Peter Mackenzie-Helnwein

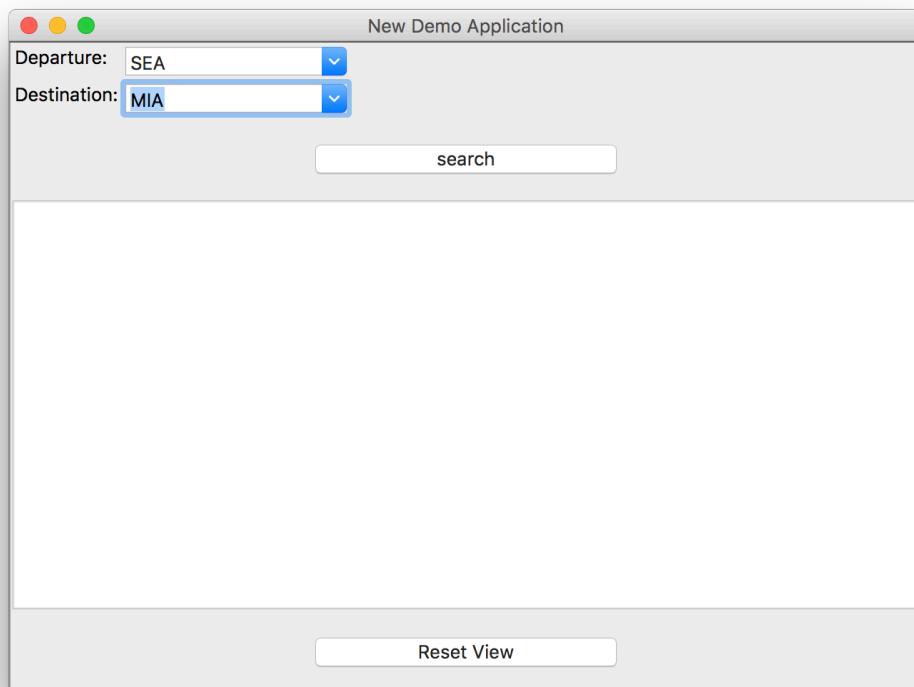
1. Problem Statement:

Given:

- Your flight database from your midterm solution
- Your Graph class, including a `findShortestPath(self)` method

ToDo:

- create a GUI that creates a Graph containing your flight database as part of the constructor
- implement a method `Graph.getAirportCodes(self)` to `Graph` that returns a sorted list of available airports (by three-character code)
- create a GUI similar to the following image:



•

- use the list of airport codes provided by your new Graph.getAirportCodes() to populate both QComboBoxes
- implement a callback function to find the shortest flight between the selected airports. Display an error message in the QTextEdit if departure airport matches the destination.
- show the result of your search in the shown QTextEdit -- only if no input error occurred.
- make sure that a new selection of either airport erases the text field.
- make the "reset view" -button move the app to the center of the screen and resize to half screen width and half screen height.
- allow selection and copy from the text field. Prevent the user from changing that text other than through a search.
- add a menu that presents a simple About dialog and a Quit feature.

Deliverables:

- Full documentation including the problem statement, an explanation of your algorithms (not the Graph, just the GUI), screenshots of various states of the program (with explanation!).
- All code and the database file needed to execute your code.

2. Algorithm

Firstly, we build a Class named Airport which has the object QWidget. Connect the Graph.db we created in Midterm. Create the layout and add two lables(Departure, Destination), two QComboBox, a QTextEdit and two QPushButton(Search, Reset View) to the interface. We set appropriate size of different components. Define the editText as setReadOnly(1) to prevent the user from changing that text other than through a search, but users could select and copy from the text field.

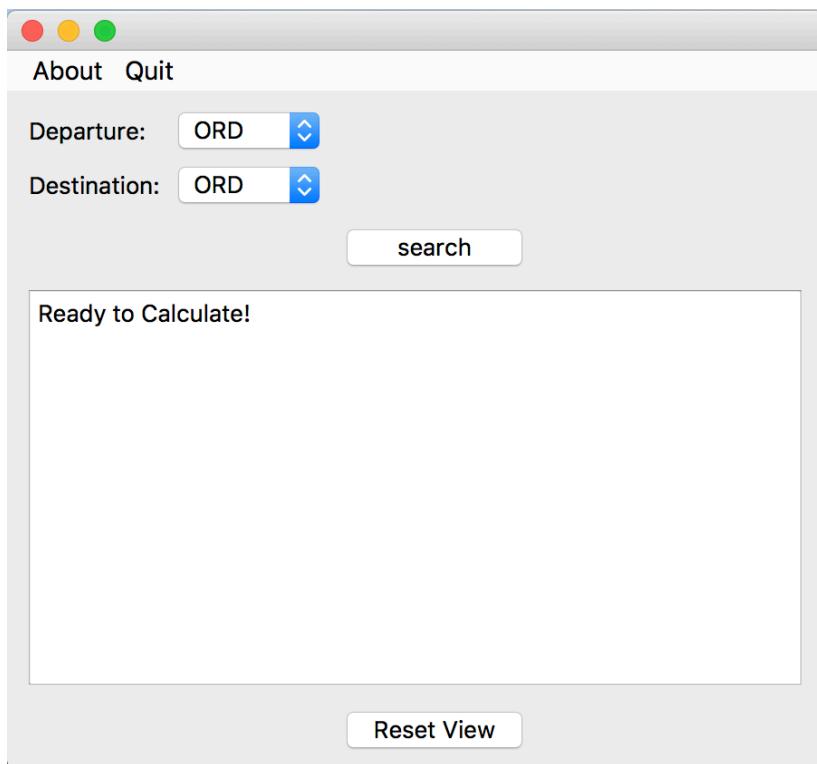
Then, we define several functions such as on_departureChanged, on_destinationChanged, on_Search_clicked, on_reset_clicked to change the status of search and result.

Finally, we set a menu in a new Class App with the object QMainWindow and set the Class Airport to be the centralwidget. Create the menubar with option “About”

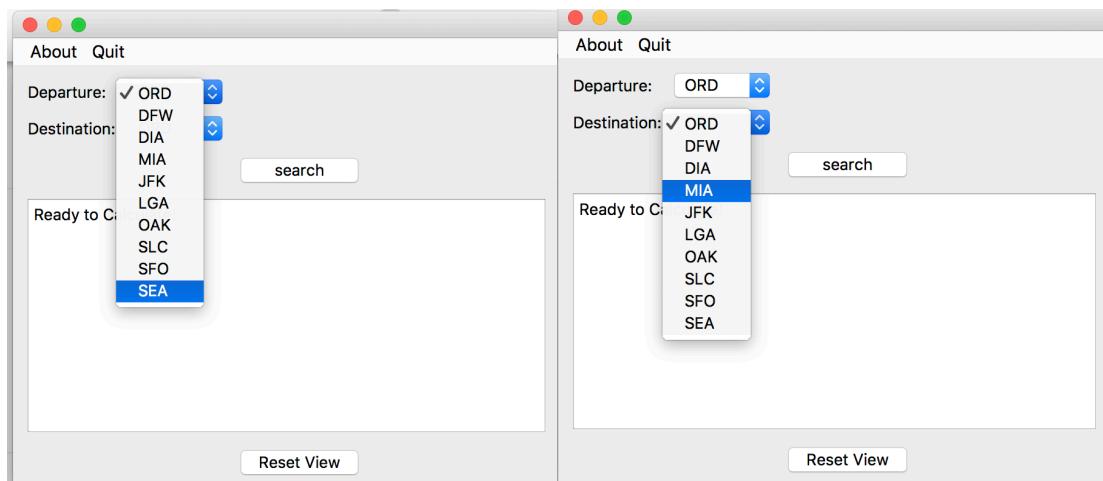
and “Quit” in this class.

3. ScreenShots

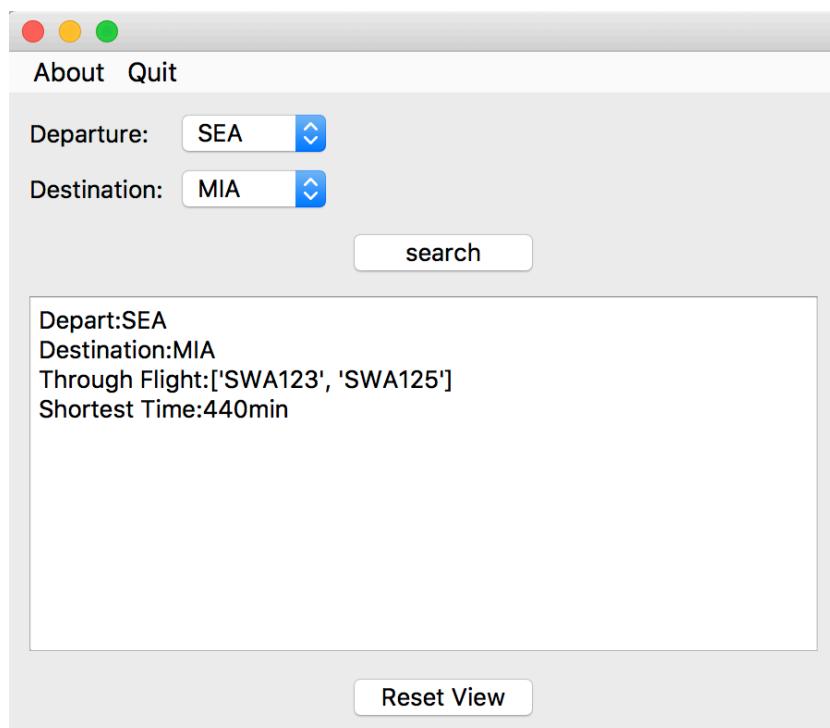
3.1 The Starting view of the program



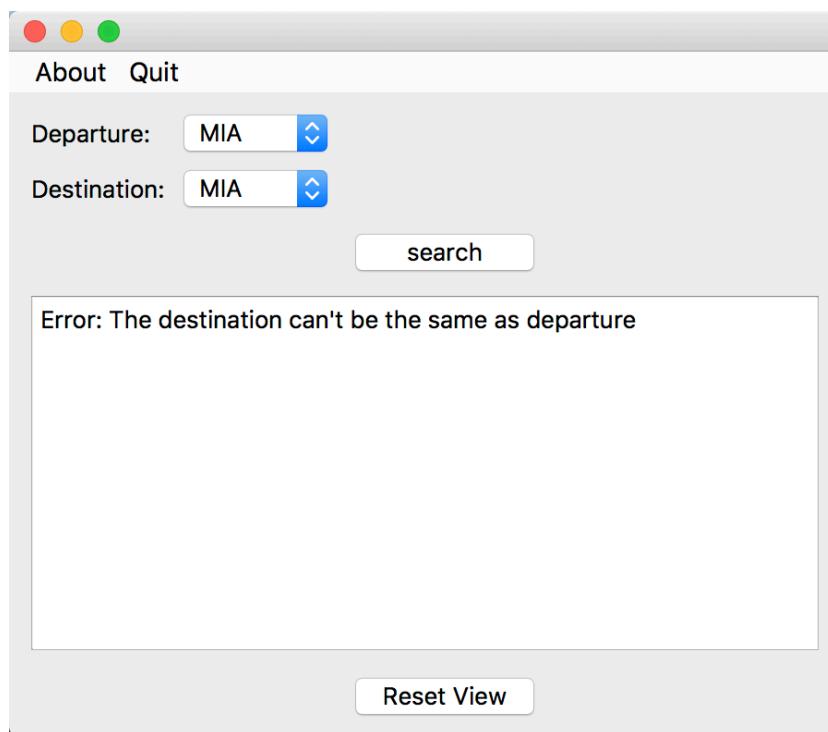
3.2 Select the departure and destination airport.



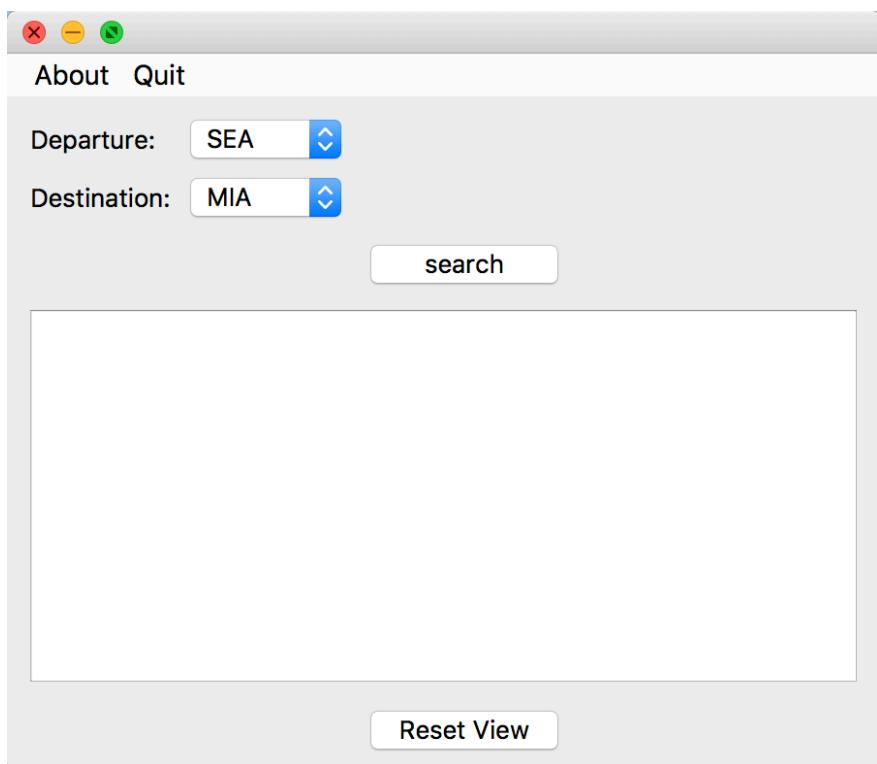
3.3 Search the shortest route and show the result in the following text field.



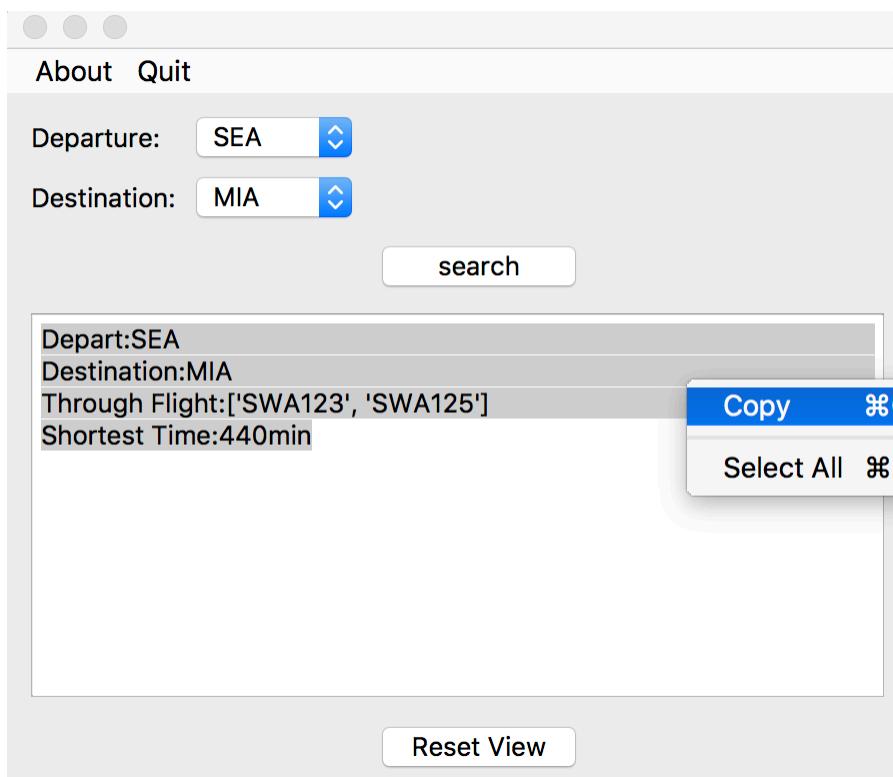
3.4 If the departure airport is the same as destination airport, display an error message in the text field.



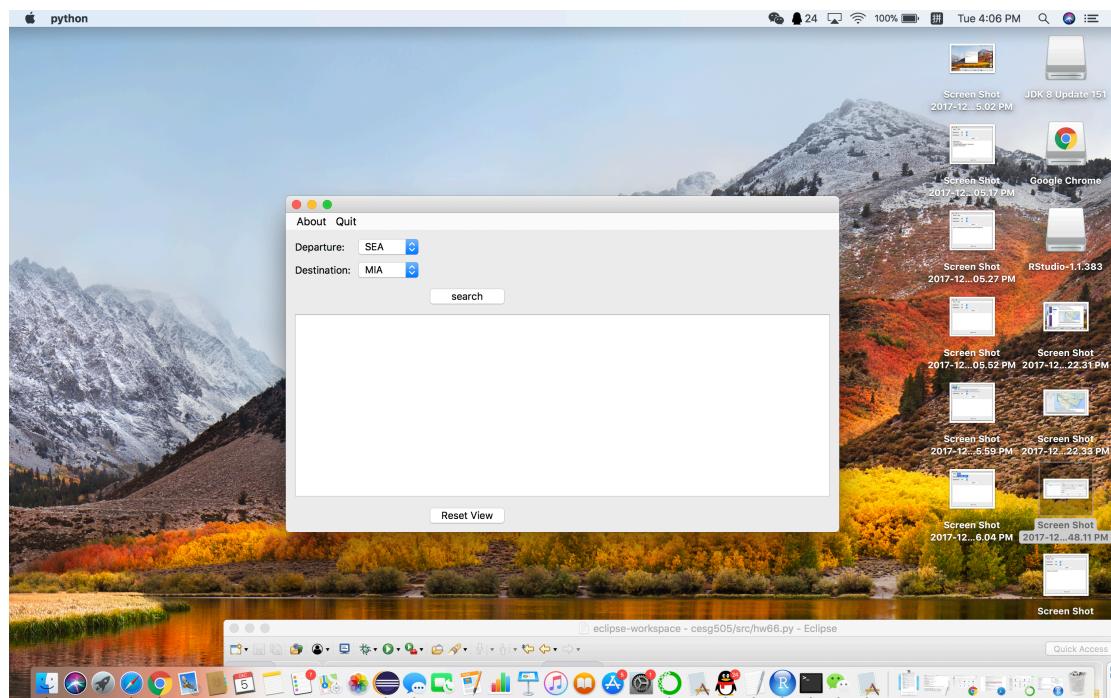
3.5 A new selection of either airport erases the text field.



3.6 Allow selection and copy from the text field.



3.7 Make the "reset view"-button move the app to the center of the screen and resize to half screen width and half screen height.



3.8 There is a menu that presents a simple About dialog and a Quit feature.

