Prototyping Assignment

By Ricardo Alvarez and Oscar Galindo

For this assignment we decided to implement features of our final project DSP system. We implemented three distinct controls, which are described below.

First a small look of our system, our implementation includes the options next to Apple’s logo which include “File”,” View”,” Run”, and “Help”.

A screenshot of a computer

Description automatically generated with medium confidence

1. Control and handling of text area.

For this element we added the functionality that in our main window every time the text in the text area changes then we check to see if there is a function that was declared. The way we do this is by running a regular expression on the text area content and we identify the name of functions. If the function is already declared we added to the list of tracked function on the sidebar’s combo box, else we do not add the function to the combo box’s options.

The tracked regular expression is ***r"def function \w+\(\)\:"***

For example,

Before input:

A picture containing text, screenshot, monitor

Description automatically generated

\*Notice that Function 1 and Function 2 are placeholders

Input written, function not completedA picture containing text, monitor, screenshot

Description automatically generated

Input completed; combo box populated

Graphical user interface

Description automatically generated

Code

Dependency initialization:

Text

Description automatically generated

\*Notice extra dependencies are included as we were testing ideas

Listener construction:



Linking listener to tracked object:

In main:



\*Notice we pass a function from an object that will listen when a change happens

In the visualization/main\_window object:



Handling of input

Population of the listener dependencies:

Text

Description automatically generated

\*Notice combo box is stored as a dependency that can be referenced within the object of the listener

When text has changed:

Text

Description automatically generated

1. File Explorer functionality when option “Open” is triggered.

For this implementation we added the functionality of being able to open or selecting a folder from any directory in the machine. Once the directory is selected, we are able to populate the tree widget or the navigation pane with the data found in that directory. For example,

System initiated; no directory selected

A picture containing text, monitor, screenshot

Description automatically generated

Open option triggered

Graphical user interface, text, application, chat or text message

Description automatically generated

Dialog appears and folder is selected

Graphical user interface, text, application, email

Description automatically generated

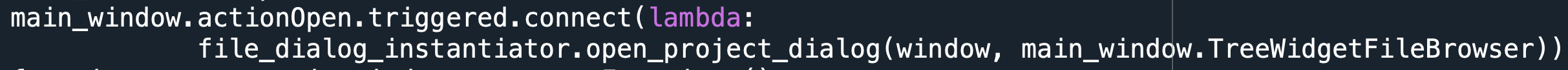
Once the option open is clicked the selected folder’s contents are shown in the file explorer

Graphical user interface

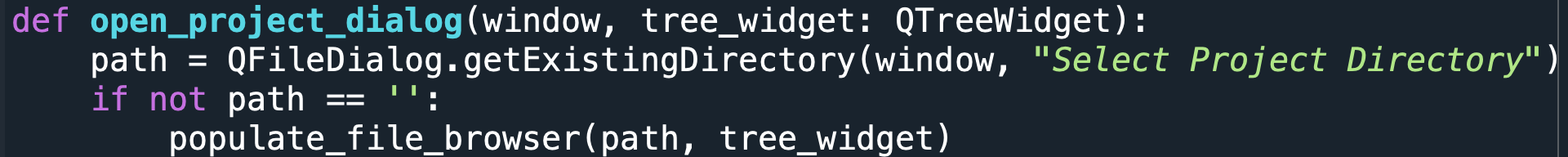
Description automatically generated with low confidence

Code

Connecting the listener to the action of triggering the “Open” option.



Handling when a trigger occurs:



Populate with contents:

Text

Description automatically generated

\*Notice there is a helper function “add\_children\_to\_folder” that aids on the discovery of new files.

\*Notice this function populates the top tree, further items are populated by the helper function.

Helper function population at lower items of the tree

Text

Description automatically generated

\*Notice this function populates by discovering from the context of an item and adds children as needed and discovered.