8/25/2023

LƯU QUỐC BẢO

22125008 – 22CTT1

DATA VISUALIZATION

CS163 – Data Structures - Lab work Project Report

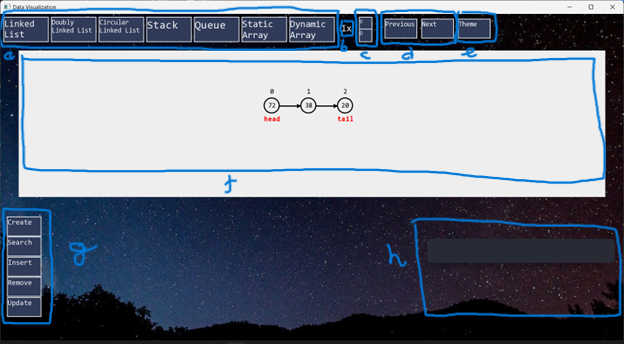
1. **Features**

Details on the features of functional used in this project can be found in “Features.pdf” in the same folder as this report.

1. **Program structure**

* There are some main struct in this program:
  + Game: This structure will be the main control part of the program. The operations that receive input from the user, set frame and render are done here.
  + World: The main thing in this struct is graph-type buttons and a liveData pointer. This pointer will indicate what type of graph is currently being used. And the graph buttons will let the user want to change to another graph.
  + Menu: Main menu is constructed here.
  + Datatypes: This is the base class for all types of graphs. This class contains a graph object for showing graph, a highlight object for showing code and highlight, and some function buttons. Some specific buttons will be adjusted in subclass.
  + AVL, GraphMain, HashTable, Trie, TwoThree, MaxHeap, MinHeap: All of these are classes for all types of graph.
    - These classes will be inherited from Datatypes class and will adjust the maingraph, list of functionButton, BaseButton, inputBox from base class.
    - In checkPress or draw function, they will call this function in base class (in base class these functions declared as virtual).
  + Graph, Array: This class contains all parameters of a graph, including nodes and arrows. All transformations of the graph will be constructed here. That is make new node, remove node, moving graph. It contains an object list of Node.
  + Node, TwoThreeNode: This class is for node. It contains text objects at 5 directions around the node and in the node, the shape of the node, and two objects of arrow for next and previous, and the pointer to the next and previous nodes if there are. All transparent of node will be constructed here. Such as changing node size based on time, changing node color based on time and moving arrow based on the angel and position.
  + DynArrow: This class is for arrow and just contains some basic functions such as moving arrow, changing size of arrow, rotating arrow. The arrow is constructed from a rectangle shape and a triangle shape.
  + Highlight: This class is for showing code and highlight. The code is showed in form of image. This class contains a texture object and sprite object to get image from computer. And also contains a rectangle shape without fill color. It is used for highlighting each line of code.
* The program will start at DataVisualiztion.cpp, the main function is stored there. The smart pointer of Game object will be initiated here. Then the run function of Game will be called.
* Then in run function, there are handleEvent function. It will take user’s input and solve it. Below that is render function, it will draw and render the program to the screen.

1. **Tutorial**
2. General



1. Switch the data button. There are all data types in this app provided. Press on it will change to those types.
2. Speed display. It shows us what speed of the graph now.
3. Change speed button. The top one for increasing speed and the down is for decreasing speed.
4. Change steps button. You can use this button to run step-by-step any function or to go back to any step you want. The graph and the code will show at the step you want. If pressed but no change, it means you have moved to the last step or the first step.
5. Change theme button. When pressed, the app will change theme from night to light or vice versa. However, the graph and graph box will not change due to give the unified look to the graph representation.
6. Graph box. All the parameters and changing of graph are displayed in this rectangle box.
7. Functional button. This is all the function used for the type of data being displayed. Some functions when you press will show more minimal function, some function you press will show the box for input or will immediately execute the corresponding function. Some examples of these functions.

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. The highlight corresponding source codes or pseudocode of the function just executed. This box will show the code, and a small rectangle to highlight which line are executed.
2. Get data from file.

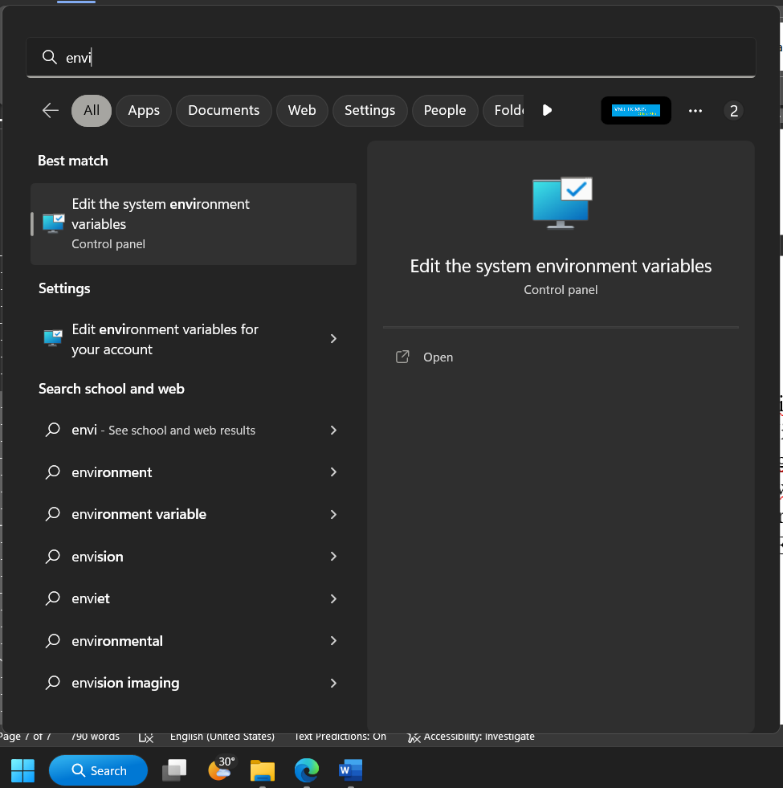
This app provides the user with the ability to input data from file in “Create” function in all data types. In bin folder, there is a file name “customInput.txt”, the data is only got from file with exact that name and correct format. Users can only input a list value of data. They don’t need to input the number of values and always have “;” after each value, including the last value.

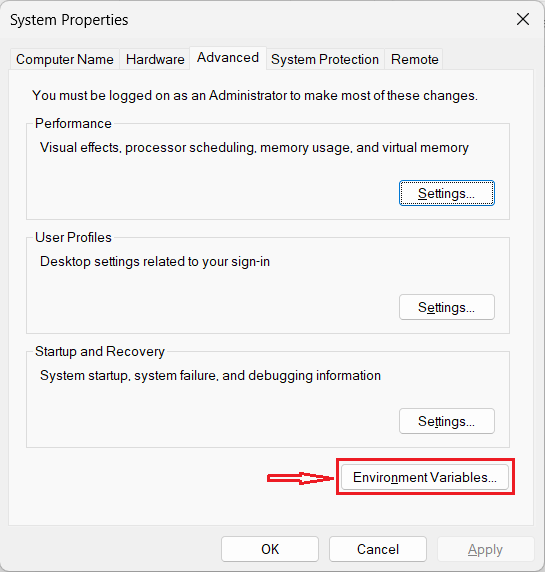
1. Input data from keyboard.

* Users can click on the text to move the cursor to this position of text.
* This box only takes numbers. The value in box is automatically random when then function button contain it pressed. The blank box means that the value in it is 0.
* Some functions which contain input boxes have “Go” button. When “Go” is pressed, the corresponding function gets data from input box and executes. If nothing happened, it is possible that the input data exceeds the limit.
* To the left of each input box there is a short text indicating what the type of value of the box is. Some examples of it:
* “n =” or “c =”: this input box stores the value of size of graph. The maximum value for graph size in this program is 15.
* “i =”: this input box stores the value of index of graph. The maximum value for this box is the graph size being displayed.
* “v =”: this input box stores the value of node. The maximum value for this box is 99.

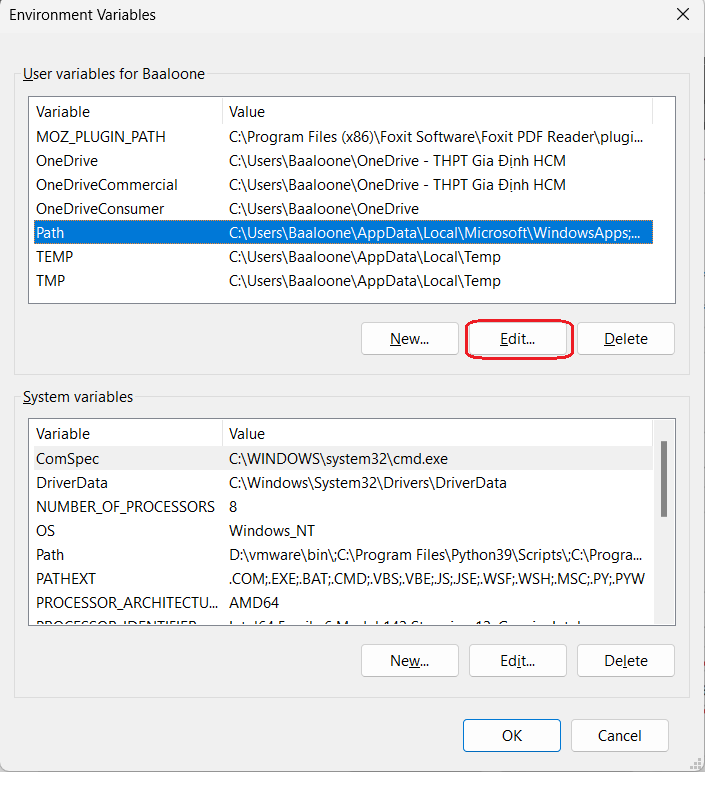
1. **How to compile**

* Get the folder of program from git link and download and extract this folder to setup: https://drive.google.com/file/d/1BR3XW9-rNcZWWzK8cvmCibozhu1-gIAP/view?usp=sharing
* In this folder will have 2 achieved files are “SFML.zip” and “mingw32.zip”. Extract the file “mingw32.zip” and put it to any folder you want, in this tutorial we assume it is in disk C:\. This is important, the “SFML.zip” must be extracted and put in same folder with “Data Visualization” – app folder.
* Follow these steps below to setup Environment:
  + Find “Edit the system environment variables”





* + Choose “Path” line below and click “Edit…”



* + Choose “New”, paste the address of bin folder of mingw, default is “C:\mingw64\bin” and click “Ok”

Graphical user interface, application

Description automatically generated

* + Run file “setup.bat” in “Data Visualization” folder. Then it will run a little bit and “Data Visualization.exe” appeared in this “bin” folder. Click and play.

1. **Github:**

* Link of Git: <https://github.com/luuquocbao56132/Data-Visualization.git>
* Number of commits: 45. Some typical commits:
  + Setting arrow of 1 node can follow the next node – aead0d0.
  + Change size node animation – 902ce00.
  + Complete input box from user – 86ba0ac.
  + Input file – be6eda5.
  + Add change speed graph – 3a05e7b.
  + Animation for create graph and node – 960a18e.
  + Step-to-step for graph – 585ea5b.
  + Convert from using std::vector to LinkedList – c6ce80c.
  + Change Theme color – 949e246.

1. **Video demo: https://youtu.be/0-WmCYVtyX0**