**1.0 User Guide**

The purpose of the gameofcatz menu module is to load input file and let users to find, add, remove or update their nodes and edges and moves through a world made of those nodes and edges. The menu contains 10 options for the user to select such as to allow the user to load the data that they want as well as to display and save the data in to file. The simulation mode has not been implemented so will be omitted from this description.

**2.0 Descriptions Of Class**

Classes:

* DSABinarySearchTree
  + Created to store data and for searching for specific data according to the menu option selected by the user.
* DSALinkedList
  + Created to store data and for searching for specific data according to the menu option selected by the user.
* DSAListNode
  + Created for class DSALinkedList
* readGraph
  + Created to handle any fileIO tasks
* DSAStack
  + Contains several methods that should to assist in depth-first search
* DSAQueue
  + Contains several methods that should to assist in breadth-first search
* DSAGraph
  + Creates graphs
* DSAGraphVertex
  + Contains several methods that should to assist in DSAGraph
* DSAGraphEdge
  + Contains several methods that should to assist in DSAGraph
* gameofcatz.java
  + Contains the main method which contains the menu page to run the entire program.

**3.0 Justification of Decision**

I chose the ADT graph because it is required by the assignment to use it to implement the options. I stored data into graphs after reading the input file as opposed to using a linked list to store the data as it is direct and will be stored in a vertex or edge list within the graph class. I chose the .txt format is because it is the same format as the input file and is easier to see the output.

For the gameofcatz menu, I chose to use a switch case because it is much easier and less tedious to implement compared to using if and else if. This is because the if and else if would require me to make comparison of the user input to the options. Also in the menu, there can be seen 2 sc.nextLine() for everytime the program prompts the user to enter a string input. This is because the next line will be ignored and it will not take in any user input and produces an error or do nothing.

**4.0 Showcase**

**4.1 Introduction**

After compiling all the necessary files and run the program, a menu will be shown o the terminal screen prompting an input from the user. After selecting an input it will enter into one of the menu options and perform its task. Some options might require another user input as it has multiple task it can perform. After going through the menu first 9 menu option, the user can exit the menu after they are done.

**4.2 Compilation**

**Step 1: Compile all necessary files as shown below:**

user@ubuntu:~$javac readGraph.java

user@ubuntu:~$javac DSAGraphVertex.java

user@ubuntu:~$javac DSAGraphEdge.java

user@ubuntu:~$javac DSAGraph.java

user@ubuntu:~$javac DSAStack.java

user@ubuntu:~$javac DSAQueue.java

user@ubuntu:~$javac DSABinarySearchTree.java

user@ubuntu:~$javac DSALinkedList.java

user@ubuntu:~$javac DSAListNode.java

user@ubuntu:~$javac gameofcatz.java

Note: if you do not want to type all this, you can do

this: user@ubuntu:~$javac \*.java

**Step 2: After compiling all necessary files, compile the main program**

user@ubuntu:~$ javac gameofcatz.java

**Step 3:After compiling all the required files, run the program by entering the following terminal command:**

user@ubuntu:~$java gameofcatz

**\*A message on how to use the program will be displayed on the terminal screen.**

**4.1.2 Modes**

**Step 1: Read the program message shown below and select which files to load**

Usage: java gameofcatz x

where

x is one of:

-i -> Interacting testing environment

-s -> simulation mode (usage: gameofcatz -s infile savefile)

**Step 2: To enter report mode, type in the following command**

user@ubuntu:~$java gameofcatz -s

**\*Simulation mode should display all of the options in the menu one by one instead of the user going through the interactive mode. It will need the input file for data and will save the progress into a new text file. However, for this program, the simlation mode is not implemented.**

**Step 3: Enter the following terminal command with reference to the output message in Step 1**

user@ubuntu:~$java gameofcatz -i

**4.1.3 Menu**

**Select option 1 in the menu option as shown below**

Please select one of the following options:

1) Load input file

2) Node Operations

3) Edge Operations

4) Parameter Tweaks

5) Display Graph

6) Display World

7) Generate Routes

8) Display Routes

9) Save Network

10) EXIT

Answer:

**\*A table of content will be displayed. The menu will loop back to the menu page.**

**OPTION 1**

**Select option 1 in the menu shown in the previous step**

Please enter the file name:

**\*A message will be displayed (shown above) on the terminal prompting the user to enter their desired input file and then load the file.**

**Step 3:** Enter the file name (e.g. gameofcatz.txt)

**\* It will read the file and loop back to the main menu.**

**OPTION 2**

S**elect one of the options given by the sub menu of option 2**

Please select one of the following options:\n");

1) Find a node

2) Add a node

3) remove a node

4) update a node's value

5) EXIT

Answer:

**\*It will perform tasks depending on what the user selected. After the user is done they can select option 5 and exit to the main menu .**

**OPTION 3**

Please select one of the following options:\n");

1) Find an edge

2) Add an edge

3) remove an edge

4) update an edge

5) EXIT

Answer:

**\*It will perform tasks depending on what the user selected. After the user is done they can select option 5 and exit to the main menu .**

**OPTION 7**

**Will prompt user to select 1 or 2 to perform Depth First Search and Breadth First Search respectivelyas shown below:**

Please select 1 or 2:

**\*It will perform tasks depending on what the user selected. After the user is done they brought back to the main menu .**

**OPTION 8**

**Will diplay the graph on the terminal screen to the user as shown below:**

Please select one of the following options:

1) Load input file

2) Node Operations

3) Edge Operations

4) Parameter Tweaks

5) Display Graph

6) Display World

7) Generate Routes

8) Display Routes

9) Save Network

10) EXIT

Answer: 8

Displaying route:

A

B

E

F

H

I

J

**OPTION 9**

**Will save the graph into a .txt file.**

**\*Not properly implemented so it will not show understandable output.**

**UNIMPLEMENTED FUNCTIONS**

Options 4 - 6 and simulation mode have not been implemented in the menu option of the program.

**5.0 Future Work**: Missing items or suggested enhancements

Missing items would include menu options 2-8 that were not implemented .

**6.0 Traceability Matrix Map**

This has been included in the assignment submission in blackboard.

**7.0 Class Diagram**

The UML class diagram has been included in the assignment submission in blackboard and the classes have test harnesses but are not included in the UML diagram as it was mentioned in Piazza that it was not needed.

**8.0 Conclusion & Future Work**

The assignment required an extension of my graph code in order to go through a world. However, I am unable to demonstrate the use of my graph because I am unable to store the data to my graph. I have dealt with files that only gave the data without comments. So, I find it a little difficult with trying to tokenize the information that I want.

My prediction for the performance of my code will be quite underwhelming. The readGraph.java should read into the input file and stores them into a graph and writes them to a new file. It stores data from the files as String into an array then it will be stored into a vertex and edge. I used arrays in the fileIO then store the data to the graph. As I am having problems with getting the wanted data from the unnecessary comments, it will only read and store each line to an array.

My implementation of the menu is simple and can be improved in many areas.

**9.0 References** Chicago referencing style

Maxville, Valerie. 2021. “UML Diadram.” COMP 1002\_5008. Piazza. <https://piazza.com/class/krd3qbda5sy510?cid=317>

# Heemayl. 2015. “What Does ['~$' mean? [closed]](https://askubuntu.com/questions/607347/what-does-mean) ” Ask UBUNTU, April 9th, 2015 . <https://askubuntu.com/questions/607347/what-does-mean>

Curtin University of Technology. 2019. COMP1002 DATA STRUCTURES AND ALGORITHMS. "LECTURE 4:Linked Lists and Iterators"

Curtin University of Technology. 2019. COMP1002 DATA STRUCTURES AND ALGORITHMS. "LECTURE 5:Trees"

Curtin University of Technology. 2019. COMP1002 DATA STRUCTURES AND ALGORITHMS. "LECTURE 6:Graphs"