CIS 22C Team Project Report #3

Team # 2 Team Leader: Bruce Decker

Team Members: Bruce Decker, Bao Chau, So Choi, Vignesh Senthin

- Show what each screen will look like (main menu, sub-menues)
- Describe or show what the format of any output will be (for example, list of city names indicate visitation order, list of states with its color, list of network connections and their weights, etc.)
- Indicate what your DATA type will be and representing what (be specific)

For the person(s) solving "the graph problem"

- show the UML class diagram of the subclass of Graph. See the Team Project Code Files folder for the Graph and my example Kruskal.java file.
- give the name of the algorithm you will use to solve your graph problem (or if you don't have a "name" for it, please describe it or show me the algorithm here).

Please list the tasks (with updated ones you may have changed from Report #2), along with your progress (what each member has done so far)

Task	Team Member	Progress
Team project report #3	Bao	Due on Monday
Write pseudocode & for algorithm.	Bruce, Vignesh	Done
Write Euler Circuit algorithm.	Bruce, Vignesh	In progress
Define models for graph, edge, vertex	Bao	Done
Test cases & input data	So Choi, Bao	In progress
File input/output	So Choi, Bao	In progress
Write code to add/remove edges.	Bruce, Vignesh	In progress
Code to display data	Bruce, So Choi	In progress
Driver code	Bao	In progress
PowerPoint slides, Prooread/review report	So Choi	In progress

EULER CIRCUIT

Problem to Solve:

Every of us probably had seen a mailman walking through the neighborhood, dropping off mails and letters from one mailbox to another. Most, if not all, of the times, the mailman drives to the neighborhood in a vehicle, parks on a street and walks through the neighborhood to deliver the mails. After that, he must return to his vehicle, drives to the next neighborhood and repeat the process. As a mailman, he will need to walk through all streets and corners of the neighborhood. Evidently, the mailman would love to use the path that allow him to visit all streets and corners of the neighborhood, and come back to his vehicle without repeating any paths or streets in between. This is where Euler circuit can help.

In graph theory, Euler circuit is a path that starts at one vertex, visits every edge exactly once, and ends at the starting vertex. In the mailman's situation, the location of an intersection and the end of a court (or dead-end street) in the neighborhood are vertices. The streets connect 2 vertices are the edges.

Sample output format for Euler Circuit:

```
Euler Circuit for "Some-Name" neighborhood:
        Starting at Aborn-Brigadoon Intersection
                Head toward:  Aborn-DanderHall Intersection
                Head toward:  Aborn-Nieman Intersection
                Head toward:  End of Annerly Court
                Head toward:  …………………………………….
                Head toward:  …………………………………….
                Head toward:  …………………………………….
                Head toward:  …………………………………….
                Head toward:  End of Boswall Court
                Head toward:  TurnHouse-Laddie Intersection
                Head toward:  Laddie-Nieman Intersection
        Go back to Aborn-Brigadoon Intersection
```

Data Type:

LocationPoint – An object represents an intersection or the end of a dead-end street.

Main menu & Sub-menues

Main menu:

```
MAIN MENU
    1. Select a Neighborhood
    2. Quit

Please enter your selection:
```

Submenu 1:

```
SELECTING A NEIGHBORHOOD
    1. Name of Neighborhood 1 (Filename)
    2. Name of Neighborhood 2 (Filename)
    3. Name of Neighborhood 3 (Filename)
    4. Return to previous Menu

Please enter your selection:
```

Submenu 2:

```
NEIGHBORHOOD MENU
    1. Update Neighborhood Map
    2. Show Euler Circuit
    3. Show Breadth First Traversal
    4. Show Depth First Traversal
    5. Show Adjacent List Table
    6. Return to Previous Menu
    7. Main Menu

Please enter your selection:
```

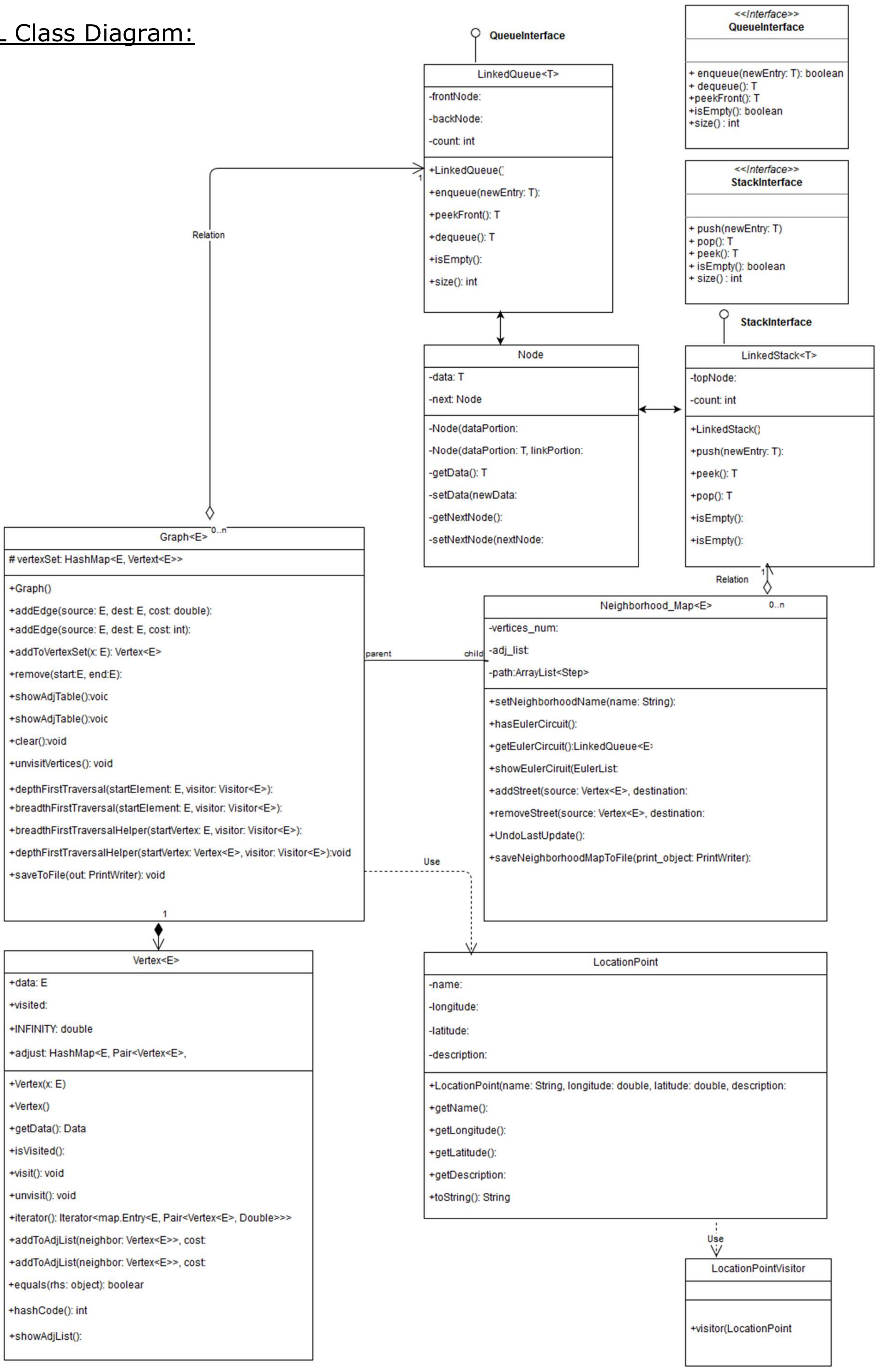
Submenu 3:

UPDATE NEIGHHOOD MAP

1. Change Neighborhood Name
2. Report New Street
3. Report Closed Street
4. Undo Last Update
5. Save Neighborhood Map
6. Return to previous Menu
7. Main Menu

Please enter your selection:

UML Class Diagram:



Fleury’s Algorithm (Euler Algorithm):

Let G be a connected graph. If G is Eulerian then Fleury’s algorithm will produce an Eulerian trail in G. In a connected graph G, a bridge is an edge which, if removed, produces a disconnected graph.

Fleury’s Algorithm

Let G be an Eulerian graph.

Step 1: Choose any vertex v of G (Graph) and set current vertex equal to v and current trail equal to the empty sequence of edges.

Step 2: Select any edge e incident with the current vertex but choosing a bridge only if there is no alternative.

Step 3: Add e to the current trail and set the current vertex equal to the vertex at the ‘other end’ of e. (If e is a loop, the current vertex will no move.)

Step 4: Delete e from the graph. Delete any isolated vertices.

Repeat steps 2-4 until all edges have been deleted from G. The final current trail is an Eulerian trail in G.