**ARTIFICIAL INTELLIGENCE(INT 404)**

**SIMPLE NAVIGATOR FOR LPU**



Student Name: Joseph Acquah

Student Name: Munnaluri Venkata Kutumba Hanumath Vishal

Student Name: K. Pavan Kalyan

Student name: Swagat Shandilya

Student name;

Roll Number: 21

Roll Number: 22

Roll Number: 23

Roll Number: 24

Section: K18XC

Date of Submission: 10/04/2020

Student ID: 11816135

Student ID: 11804617

Student ID: 11809995

Student ID: 11813767

Faculty Member: Dr. Sukhvir Kaur

**TABLE OF CONTENTS**

1. **ABSTRACT**
2. **INTRODUCTION**
3. **OBJECTIVES**
4. **METHODOLOGY**
5. **RESULTS**
6. **CONCLUSION**
7. **FUTURE SCOPE**

**ABSTRACT**

**Objective:**

* **Keep the travel costs as low as possible**
* **Each edge or path between locations are assigned a given rate or value. One of the main objective is to make sure with each path taken its value is small.**
* **Keep the total distance travelled as low as possible**
* **Focus on Optimization**

**Methods:**

* **Breadth First Search**
* **Traveling Salesman**
* **Depth First Search**

**INTRODUCTION**

**Definition**

Given a set of locations, and known distances between each pair of locations, the simple navigator which is going to be used to choose a route exactly once, minimizes the total distance travelled to get to a specific location.

The simple navigator is an application of the travelling salesman problem as it deals with logistics. The travelling Salesman has a task of visiting N number of cities. He will start from a home location and would want to visit each city just once and return back to the original location from where he starts. Travelling salesman route will be planned in such a way that in a given N number of cities, what is the minimum round trip route that visits each city once and returns to the starting place.

It was used to find good routes or schedules for trucks, order-pickers in a warehouse, service engineer, aircraft and tourists. There are however some less obvious applications like scheduling jobs on machines, controlling satellites, computing DNA sequences, designing telecommunications networks, x-ray crystallography and clustering data arrays.

**OBJECTIVES**

* Most people will have had the experience of giving or receiving directions for navigating through an unfamiliar geographic environment. For example, visiting LPU for the first time, an individual might ask a passer-by for directions to a location or block.
* Deliveries on the go is also one of the many areas that this program can aid since it is helpful in knowing large locations easily.
* Mathematically, automotive navigation is based on the shortest path problem, within graph theory, which examines how to identify the path that best meets some criteria (shortest, cheapest, fastest, etc.) between two points in a large network. Simple navigation systems are a crucial for the development of wireless self-driving cars.

**METHODOLOGY**

The simple navigator incorporates methods such as breadth-first-search algorithm and depth-first-search algorithm.

**BREADTH FIRST SEARCH ALGORITHM**

It is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root, and explores all of the neighbor nodes (locations) at the present depth prior to moving on to the nodes at the next depth level. It uses the opposite strategy as depth-first search. In breadth first search it is assumed that all vertices are reachable from the starting vertex or starting location.

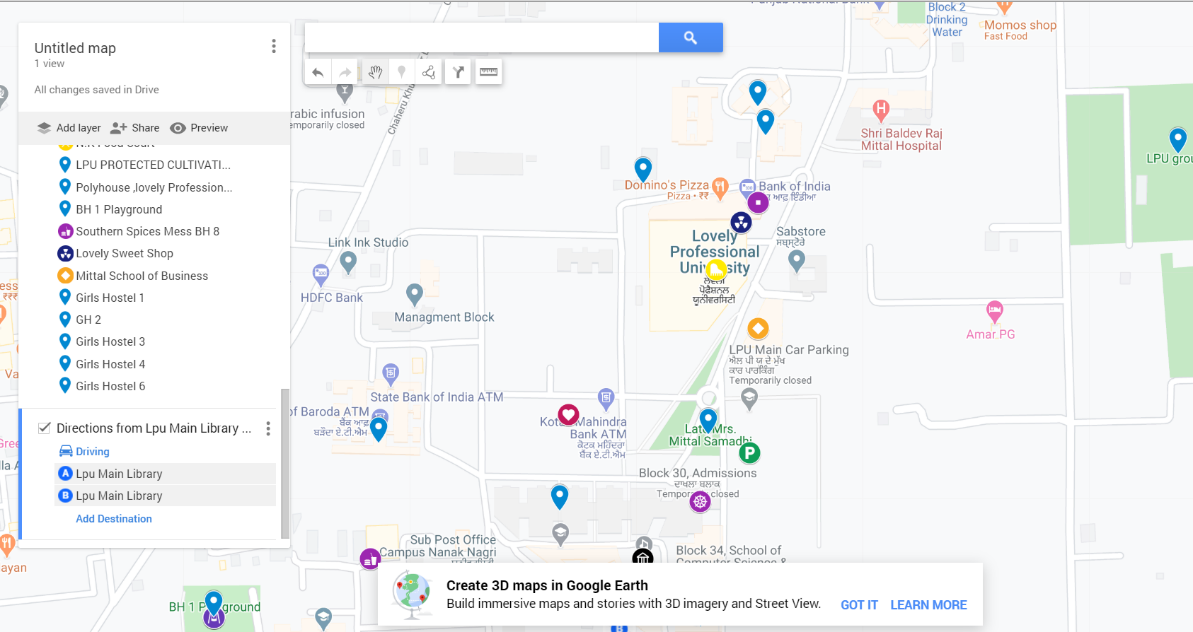
It also uses depth first search, similar to breadth first search.

**DEPTH FIRST SEARCH ALGORITHM**

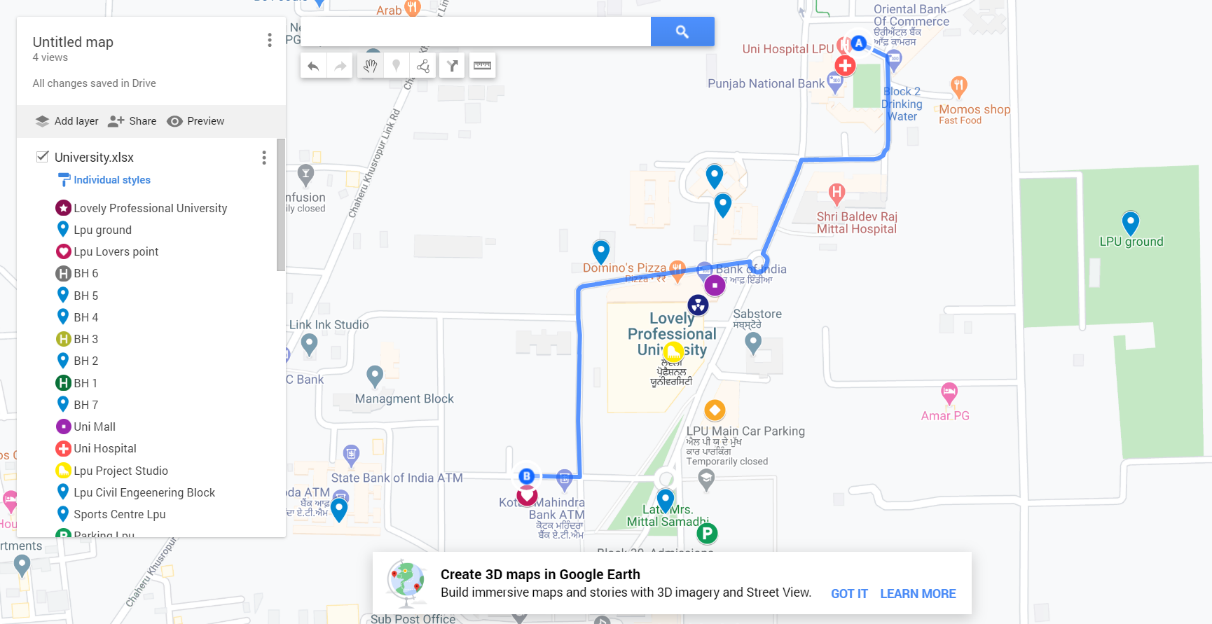
Depth first search is also an algorithm for traversing or searching tree or graph data structures except that the algorithm starts at the root node and explores as far as possible along each branch before backtracking. The time and space analysis of DFS differs according to its application area just as it is applied in the simple navigator for LPU. In this case it is performed to a limited depth.

**RESULTS**

**POINTS OF LOCATION**



**PATH GENERATED**



**CONCLUSIONS**

Numerous cognitive studies have indicated that the form and complexity of route navigation may be as important to human navigators as the overall length of route. Almost without exception, automated navigation systems rely on computing the solution to the shortest path problem, and not the problem of finding the “simplest” path.

This report addresses the issue of finding the “simplest” paths through a network, in terms of their ease of description. We propose a simplest path program that is computationally as efficient and provides an optimized solution.

**FUTURE SCOPE**

**Deliverables**

* Individuals that are new to the campus would be able to remember the paths and know where each block.

**Constraints**

* This program is limited to the locations based in LPU and cannot be used for any other location.
* There may be a need for optimized internet access to tap into the full resources of the program.

**Assumptions**

* The program is assumed to be used by an individual by foot or in a car.
* There are no objects or blockades in the path to be used by the individual.

**CONTRIBUTIONS**

Munnaluri Venkata Kutumba Hanumath Vishal

* Information gatherer and basic implementation of the simple navigator

Joseph Acquah

* Elaborating the project

Swagat Shandilya

* Web research

Pavan Kalyan

* Design and analysis of project report