Uder Taxi Application

v3.0

Designed by AlphaGun

Chaoran Nong
Yijian Wang
Yanjie Xu
Yifan Xu
Xiaoyan Yang
Yan Zhang

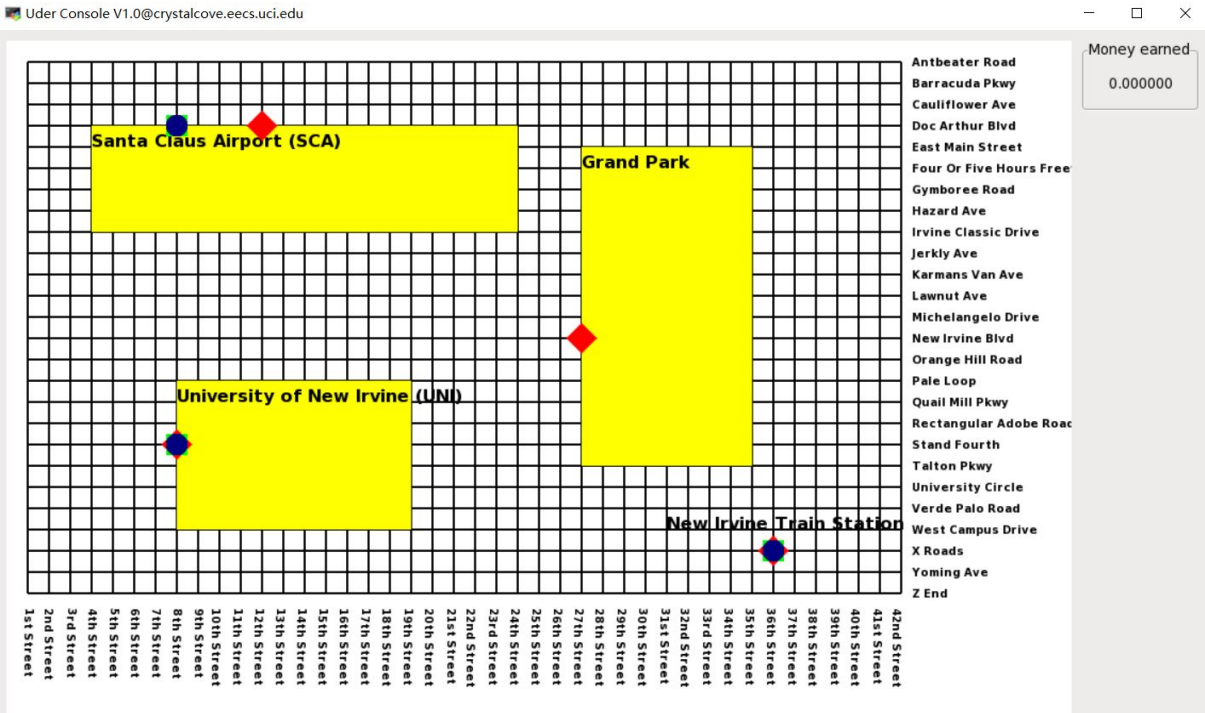
Glossary (of Taxi Cab Management)

Car: a kind of taxi that can take at most 4 people.
Van: a kind of vehicle that can take at most 6 people.
Server: a management software for route management and fee calculation.
Client: a customer application which can help users set pick up order

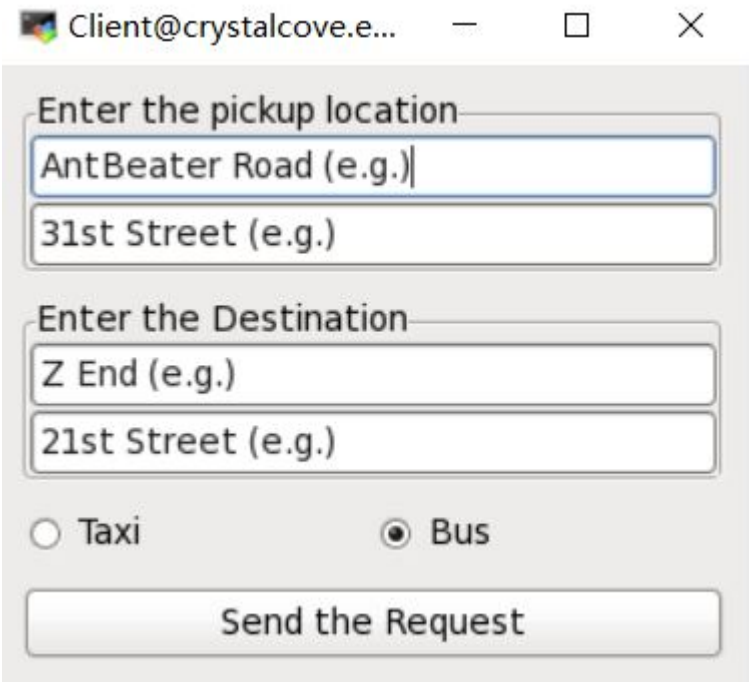
1 Taxi Cab Management

1.1 Usage scenario

Server:



Client:



1.2 Goals

Our program is an universal Taxi-Van schedule system. It aims to build a bridge between passengers and drivers by planning routes Intelligently to save both time and costs for both sides. It can function at any cities or towns no matter how big they are. All what a operator needs to do is create a city map that compatible with our API.

1.3 Features

- Auto Routes Planning
- Auto Order Dispatch
- Auto Price Calculation
- Taxi/Van Choice
- Real-time Operation Display
- Carpool/First Class Choice
- Background Operations Management

2 Installation

2.1 System requirements

Operating System:

Linux OS or Unix

Processer:

Intel Pentium (@ 1.5 GHz)

AMD ATHLON (@ 1.6GHz)
Or later versions

Graphic Requirements:

Nvidia GT 9800
AMD Radeon HD 2000
Intel HD graphics 500
Or later products

Disk Space:

At least 100M HDD space

RAM:

At least 10M

2.2 Setup and configuration

Type: “tar -zxvf uder.tar.gz”
Enter to confirm installation

2.3 Installing

1. Decompression package
2. Type **make**
3. Type **make runs** to start server
4. Type **make runc** to start client

3 Program Functions and Features

3.1 Detailed description of client and server communication

The client and server will communicate through Socket, which is a point-to-point communication solution for client-server model. The client will initiate the communication and send requests while the server will wait until services client requests and then sends corresponding responses. All the communication will base on Transmission Control Protocol over Internet Protocol, which is also known as TCP/IP. The Host is crystalcove.eecs.uci.edu and the Host IP address is 128.200.85.14. The client and server will use port 2017. If that port is occupied, port 2018, 2019, 2020, etc will be the used instead as backup.

3.2 Detailed description of routing algorithm, scheduling algorithm

Routing Algorithm:

1. If a client request: use a search algorithm to find all the cars near by, find the shortest one and make the car move to the destination.

2. If no request: do one search and move the taxies which near to the parking lot and make them enter the parking lot, and all the other cars randomly moving around.

User input: pick up location and drop-off location
Program output: arrival time and waiting time and fees.

Scheduling Algorithm:
Use the system time function to record each request including the future request and immediate request and then dispatch taxi according to the request list.
If a user make an appointment of the pick up order, the system will reserve the request in a list and use the routing algorithm to calculate the route nearby taxi and find the right taxi that can arrive on time.
There will be a time range for waiting time to make sure the company and the taxi system works more efficiently and make more profit.
range: (max waiting time, 0).

User input: pickup location, pick up time, destination address
Program output: request confirmation, pick up time, arrival time and fees.

4. Appendix

4.1 Copyright

First Printing, 2016