1 Problem Overview

- Sort and order packages onto truck
- Determine packages that can be delivered on time.
    - Driver has 8 hours to return to warehouse.
- Determine and deliver driving directions from currently location to driver for next address.

2 Classes

2.1 Package

- UUID
  - Each package exists as **one package object** in the system
- Package pointers to two client objects
  - Sender Client
  - Receiver Client
- Structure to hold packages currently
  - In the warehouse
  - On the truck
- Packages have a priority
  - Overnight
  - Two day
  - Regular

2.2 Clients

- Separate client information from packages
- Client object should represent one physical client.
- Each client (sender/receiver) points to one or multiple packages.
  - Separate sent package pointers
  - Separate received package pointers

3 Trucks

- Trucks have a weight limit.
- Trucks, and drivers, have 8 hours to return to the warehouse after start.

4 City

- City is divided into quadrants.
  - Streets North of center (Main Street) are numbered sequentially (1st Street North, 2nd Street North, etc.)
  - Streets South of center (Main Street) are numbered as (1st Street South, 2nd Street South, etc.)
  - Avenues West of center (Central Avenue) are numbered as (1st Avenue West, etc)
  - Similar for East of Central Avenue.
- Assume truck can drive one block per minute.
  - Stops take 5 minutes.

5 Program Overview

- Make file

5.1 Structures

5.1.1 Map

- Collection of nodes with edge weights as grid driving distance (truck and only go north/south or east/west.)

5.1.2 Packages

- Age variable for two-day or regular packages.
  - ◇ Birthdate or mail dates.
  - ◇ Prevents late packages.

5.2 Style

- Classes (method) and object names are initial capitalized (MyClass)
- Class member variables are initial lowercase (myVar)
- All variables, functions, methods, etc are camelCase.
- Following Qt Doxygen style. Doxygen documentation`http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html`

6 Classes

- Client class
- Package class
- Truck class
  - Weight initialize
  - Trucks are contained in a vector
    - Multiple trucks are passed to other class objects
- Map class
  - Creates adjacency list of packages.
- Routing class (algorithm)
  - Truck vector is passed to the algorithm
  - Map graph adjacency list is passed to the algorithm
  - Routing algorithm loads truck based on “something”

7 Input Files

Comma separated list of packages to deliver.

- Each line includes the sender and receiver.