CS 378 - Autonomous Vehicles in Traffic I

Week 14a - Assignment 7 & Final Research Project
Reading Responses

- I was very impressed by some of the questions asked in the reading response and in class for last week's reading
  - Keep up the good work!

- For someone interested in research, learning how to read these papers is a great skill to acquire.
Final Presentation and Report

• I wish to divide the 7 project presentations among the 2 days
• 1 person has volunteered to present on Monday
• The presentation will be evaluated on:
  ○ Can you explain the problem?
  ○ What is the motivation for the problem?
  ○ What solution are you trying to apply to the problem? Why?
• Also, if you have any preliminary results, then definitely try and present them
• Each presentation will be for 15 minutes
• Please let me know by the end of today if you are willing to present on Monday -- otherwise I will have to randomly decide the order
A* Assignment

- The assignment is due next Friday (5/4) -- last class day.
- You'll have to work on implementing pseudo-code, using a library inside a different package and STL data structures.
- Let us take a look in detail.
A* Assignment

- [http://www.openstreetmap.org](http://www.openstreetmap.org)
- Can export data in a number of formats (We use xml)
- We have some c++ code using [TinyXml](http://www.openstreetmap.org) to parse this graph structure.
- You have to implement the A* pseudo-code available on the wiki page [here](http://www.openstreetmap.org).
A* Assignment - Graph

```cpp
class Graph {
public:
    std::vector<Segment> segments_;
    std::vector<Vertex> vertices_;

    void load(std::string fileName);
};
```
A* Assignment - Segment

class Vertex;

class Segment {
    public:
    std::string name_;  
    RoadType type_;    
    double laneWidth_;  
    SegmentId segId_;      
    VertexId terminalVertex1_; 
    VertexId terminalVertex2_; 

};
A* Assignment - Vertex

```cpp
class Vertex {
    public:
        LatLong ll_; from OSM
        MapXY xy_; UTM
        bool onRoad_; 
        VertexId vertexId_; 

        std::vector<SegmentId> segmentList_; 
};
```
A* - Main STL datatypes


Sequences
- vector - ordered list
- set - unordered list
- priority_queue - automatically updating ordered list

Associative containers
- map - key-value mapping
- hashmap - faster insertions/deletions than map
- set - it is actually a associative container where the key is the value itself

A lot more, but these give you some basic idea.
A* - Main STL datatypes

- When working with STL containers, I typically *always* pull up the reference.
- [www.cplusplus.com](http://www.cplusplus.com) has excellent documentation
Final Presentation and Report

• I wish to divide the 7 project presentations among the 2 days
• 1 person has volunteered to present on Monday
• The presentation will be evaluated on:
  ○ Can you explain the problem?
  ○ What is the motivation for the problem?
  ○ What solution are you trying to apply to the problem? Why?
• Also, if you have any preliminary results, then definitely try and present them
• Each presentation will be for 15 minutes
• Please let me know by the end of today if you are willing to present on Monday -- otherwise I will have to randomly decide the order