Justine - a rapid prototype for development of GNU Robocar City Emulator and Robocar World Championship

0.0.12

Generated by Doxygen 1.8.6

Tue Mar 10 2015 16:54:19

Contents

Chapter 1

Justine - this is a rapid prototype for development of Robocar City Emulator

Authors

Norbert Bátfai nbatfai@gmail.com

1.1 Introduction

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

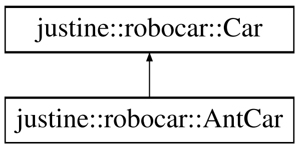
Here is a list of all documented files with brief descriptions:

Chapter 5

Class Documentation

5.1 justine::robocar::AntCar Class Reference

Inheritance diagram for justine::robocar::AntCar:



Public Member Functions

- **AntCar** (Traffic &traffic)
- virtual void **nextSmarterEdge** (void)
- virtual void **print** (std::ostream &os) const
- osmium::unsigned_object_id_type **ant** (void)
- osmium::unsigned_object_id_type **ant_rnd** (void)
- osmium::unsigned_object_id_type **ant_rernd** (void)
- osmium::unsigned_object_id_type **ant_mrernd** (void)

Static Public Attributes

- static AdjacencyList **alist**
- static AdjacencyList **alist_evaporate**

Additional Inherited Members

5.1.1 Detailed Description

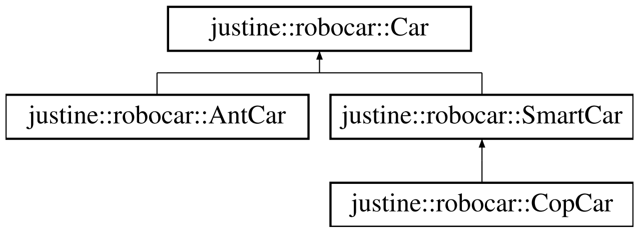
Definition at line 126 of file car.hpp.

The documentation for this class was generated from the following files:

- src/car.hpp
- src/car.cpp

5.2 justine::robocar::Car Class Reference

Inheritance diagram for justine::robocar::Car:



Public Member Functions

- **Car** (Traffic &traffic, CarType type=CarType::NORMAL)
- virtual void **init** ()
- virtual void **step** ()
- osmium::unsigned_object_id_type **from** () const
- osmium::unsigned_object_id_type **to** () const
- osmium::unsigned_object_id_type **get_step** () const
- CarType **get_type** () const
- void **set_type** (CarType type)
- osmium::unsigned_object_id_type **to_node** () const
- osmium::unsigned_object_id_type **get_max_steps** () const
- virtual void **nextEdge** (void)
- virtual void **nextSmarterEdge** (void)
- virtual void **print** (std::ostream &os) const

Protected Attributes

- Traffic & **traffic**
- CarType **m_type** {CarType::NORMAL}
- osmium::unsigned_object_id_type **m_from** {3130863972}
- osmium::unsigned_object_id_type **m_to** {0}
- osmium::unsigned_object_id_type **m_step** {0}

Friends

- std::ostream & **operator<<** (std::ostream &os, Car &c)

5.2.1 Detailed Description

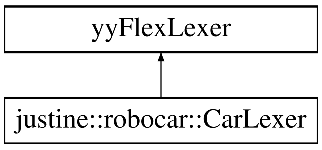
Definition at line 55 of file car.hpp.

The documentation for this class was generated from the following files:

- src/car.hpp
- src/car.cpp

5.3 justine::robocar::CarLexer Class Reference

Inheritance diagram for justine::robocar::CarLexer:



Public Member Functions

- virtual int **yylex** ()
- char ∗ **get_name** ()
- char **get_role** () const
- int **get_num** () const
- int **get_errnumber** () const
- bool **get_guided** () const
- int **get_cmd** () const
- int **get_id** () const
- std::vector< unsigned int > & **get_route** (void)
- unsigned int **get_from** () const
- unsigned int **get_to** () const

Friends

- std::ostream & **operator**<< (std::ostream &os, CarLexer &cl)

5.3.1 Detailed Description

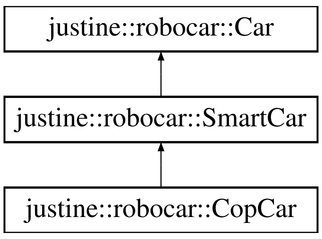
Definition at line 50 of file carlexer.hpp.

The documentation for this class was generated from the following file:

- src/carlexer.hpp

5.4 justine::robocar::CopCar Class Reference

Inheritance diagram for justine::robocar::CopCar:



Public Member Functions

- **CopCar** (Traffic &traffic, bool guided, const char ∗name)
- virtual void **print** (std::ostream &os) const
- std::string **get_name** () const
- int **get_num_captured_gangsters** () const
- void **captured_gangster** (void)

Protected Attributes

- int **m_num_captured_gangsters** {0}
- std::string **m_name**

5.4.1 Detailed Description

Definition at line 203 of file car.hpp.

The documentation for this class was generated from the following files:

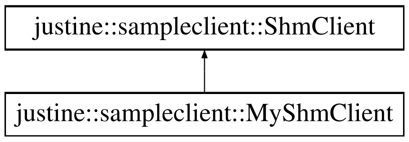
- src/car.hpp
- src/car.cpp

5.5 justine::sampleclient::MyShmClient Class Reference

A sample class used for testing the routing algorithms.

```
#include <myshmclient.hpp>
```

Inheritance diagram for justine::sampleclient::MyShmClient:


justine::sampleclient::ShmClient
justine::sampleclient::MyShmClient

Public Member Functions

- MyShmClient (const char ∗shm_segment, std::string teamname)

  This constructor creates the BGL graph from the map graph.
- ∼MyShmClient ()

  Dtor.
- void start (boost::asio::io_service &io_service, const char ∗port)

  This function starts the client.
- void **start10** (boost::asio::io_service &io_service, const char ∗port)
- int num_vertices (int &sum_edges)

  This function counts the number of vertices and number of edges in the map graph.
- void print_edges (unsigned more)

  This function prints the edges of the map graph.
- void print_vertices (unsigned more)

  This function prints the vertices of the map graph.
- NodeRefGraph ∗ bgl_graph (void)

This function create the BGL graph.

- std::vector
  $<$ osmium::unsigned_object_id_type $>$ hasDijkstraPath (osmium::unsigned_object_id_type from, osmium-::unsigned_object_id_type to)

  This function solves the shortest path problem using Dijkstra algorithm.
- std::vector
  $<$ osmium::unsigned_object_id_type $>$ hasBellmanFordPath (osmium::unsigned_object_id_type from, osmium::unsigned_object_id_type to)

  This function solves the shortest path problem using Bellman-Ford algorithm.

Protected Attributes

- NodeRefGraph $*$ **nr_graph**
- std::string **m_teamname**

5.5.1 Detailed Description

A sample class used for testing the routing algorithms.

This sample class shows how client agents can create BGL graph from data can be found in the shared memory.

Author

Norbert Bátfai

Date

Dec. 7, 2014

Definition at line 105 of file myshmclient.hpp.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 justine::sampleclient::MyShmClient::MyShmClient (const char $*$ *shm_segment,* std::string *teamname*) [inline]

This constructor creates the BGL graph from the map graph.

Parameters

shm_segment	the shared memory object name

This constructor creates the BGL graph from the map graph that is placed in the shared memory segment.

Definition at line 116 of file myshmclient.hpp.

5.5.3 Member Function Documentation

5.5.3.1 NodeRefGraph$*$ justine::sampleclient::MyShmClient::bgl_graph (void) [inline]

This function create the BGL graph.

Returns

he pointer of the created BGL graph.

Definition at line 249 of file myshmclient.hpp.

5.5.3.2 std::vector<osmium::unsigned_object_id_type> justine::sampleclient::MyShmClient::hasBellmanFordPath (osmium::unsigned_object_id_type *from*, osmium::unsigned_object_id_type *to*) `[inline]`

This function solves the shortest path problem using Bellman-Ford algorithm.

Parameters

source	the source node
target	the target node

Returns

the shortest path between nodes source and target

This function determines the shortest path from the source node to the target node.

Definition at line 400 of file myshmclient.hpp.

5.5.3.3 std::vector<osmium::unsigned_object_id_type> justine::sampleclient::MyShmClient::hasDijkstraPath (osmium::unsigned_object_id_type *from*, osmium::unsigned_object_id_type *to*) `[inline]`

This function solves the shortest path problem using Dijkstra algorithm.

Parameters

source	the source node
target	the target node

Returns

the shortest path between nodes source and target

This function determines the shortest path from the source node to the target node.

Definition at line 329 of file myshmclient.hpp.

5.5.3.4 int justine::sampleclient::MyShmClient::num_vertices (int & *sum_edges*) `[inline]`

This function counts the number of vertices and number of edges in the map graph.

Parameters

`out`	*sum_edges*	the number of edges

Returns

the number of vertices

This function counts the number of vertices and number of edges in the map graph that is placed in the shared memory segment.

Definition at line 162 of file myshmclient.hpp.

5.5.3.5 void justine::sampleclient::MyShmClient::print_edges (unsigned *more*) `[inline]`

This function prints the edges of the map graph.

Parameters

more	the maximum number of printed items

Definition at line 189 of file myshmclient.hpp.

5.5.3.6 void justine::sampleclient::MyShmClient::print_vertices (unsigned *more*) `[inline]`

This function prints the vertices of the map graph.

Parameters

more	the maximum number of printed items

Definition at line 214 of file myshmclient.hpp.

5.5.3.7 void justine::sampleclient::MyShmClient::start (boost::asio::io_service & *io_service*, const char ∗ *port*)

This function starts the client.

Parameters

io_service	
port	the TCP port of the traffic server

This method does the following: retrieves a value from shared memory, then establishes a connection with the traffic server, finally sends some client commands.

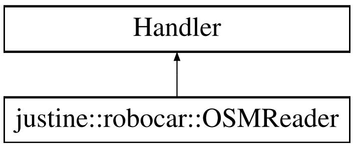
Definition at line 265 of file myshmclient.cpp.

The documentation for this class was generated from the following files:

- src/myshmclient.hpp
- src/myshmclient.cpp

5.6 justine::robocar::OSMReader Class Reference

Inheritance diagram for justine::robocar::OSMReader:



Public Member Functions

- **OSMReader** (const char ∗osm_file, AdjacencyList &alist, AdjacencyList &palist, WaynodeLocations &waynode_locations, WayNodesMap &busWayNodesMap, Way2Nodes &way2nodes)
- std::size_t **get_estimated_memory** () const
- bool **edge** (osmium::unsigned_object_id_type v1, osmium::unsigned_object_id_type v2)
- void **node** (osmium::Node &node)
- void **way** (osmium::Way &way)
- void **relation** (osmium::Relation &rel)

Public Attributes

- int **onewayc** {0}
- int **onewayf** {false}

Protected Attributes

- Vertices **vert**
- int **nOSM_nodes** {0}
- int **nOSM_ways** {0}

- int **nOSM_relations** {0}
- int **sum_unique_highhway_nodes** {0}
- int **sum_highhway_nodes** {0}
- int **sum_highhway_length** {0}
- int **edge_multiplicity** = 0
- int **nbuses** {0}
- double **max_edge_length** {0.0}
- double **mean_edge_length** {0.0}
- int **cedges** {0}
- OSMLocations **locations**

5.6.1 Detailed Description

Definition at line 72 of file osmreader.hpp.

The documentation for this class was generated from the following file:

- src/osmreader.hpp

5.7 justine::robocar::SharedData Class Reference

Public Member Functions

- **SharedData** (const void_allocator &void_alloc)

Public Attributes

- uint_vector **m_alist**
- uint_vector **m_salist**
- uint_vector **m_palist**
- int **lon**
- int **lat**

5.7.1 Detailed Description

Definition at line 62 of file smartcity.hpp.

The documentation for this class was generated from the following file:

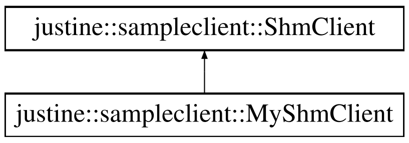
- src/smartcity.hpp

5.8 justine::sampleclient::ShmClient Class Reference

A sample class used for testing IPC mechanisms (SHM and sockets) which are used by the city emulator.

`#include <shmclient.hpp>`

Inheritance diagram for justine::sampleclient::ShmClient:


justine::sampleclient::ShmClient
justine::sampleclient::MyShmClient

Public Member Functions

- ShmClient (const char ∗shm_segment)

  This constructor initializes the shared memory segment.
- void start (boost::asio::io_service &io_service, const char ∗port)

  This function starts the client.
- virtual
  osmium::unsigned_object_id_type get_random_node (void)

  This function returns a randomly chosen node from the map.
- size_t num_edges (osmium::unsigned_object_id_type from) const

  Returns the number of out edges of a given vertex.
- osmium::unsigned_object_id_type alist (osmium::unsigned_object_id_type from, int to) const

  Returns the i-th neighbor of the actual vertex.
- int **alist_inv** (osmium::unsigned_object_id_type from, osmium::unsigned_object_id_type to) const
- osmium::unsigned_object_id_type **salist** (osmium::unsigned_object_id_type from, int to) const
- void **set_salist** (osmium::unsigned_object_id_type from, int to, osmium::unsigned_object_id_type value)
- osmium::unsigned_object_id_type **palist** (osmium::unsigned_object_id_type from, int to) const
- bool **hasNode** (osmium::unsigned_object_id_type node)
- double **dst** (osmium::unsigned_object_id_type n1, osmium::unsigned_object_id_type n2) const
- double **dst** (double lon1, double lat1, double lon2, double lat2) const
- void **toGPS** (osmium::unsigned_object_id_type from, double ∗lo, double ∗la) const
- void **toGPS** (osmium::unsigned_object_id_type from, osmium::unsigned_object_id_type to, osmium::unsigned_object_id_type step, double ∗lo, double ∗la) const

Protected Attributes

- boost::interprocess::offset_ptr
  < justine::robocar::shm_map_Type > shm_map

  The OSM map data stored in a shared memory segment.

5.8.1 Detailed Description

A sample class used for testing IPC mechanisms (SHM and sockets) which are used by the city emulator.

This sample class shows how client agents can connect and communicate with traffic emulator using shared memory.

Author

Norbert Bátfai

Date

Dec. 7, 2014

Definition at line 66 of file shmclient.hpp.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 justine::sampleclient::ShmClient::ShmClient (const char ∗ *shm_segment*) `[inline]`

This constructor initializes the shared memory segment.

Parameters

shm_segment	the shared memory object name

This constructor attaches the shared memory segment identified by the param shm_segment.

Definition at line 76 of file shmclient.hpp.

5.8.3 Member Function Documentation

5.8.3.1 osmium::unsigned_object_id_type justine::sampleclient::ShmClient::alist (osmium::unsigned_object_id_type *from*, int *to*) const `[inline]`

Returns the i-th neighbor of the actual vertex.

Parameters

to	the index i

Returns

the (osmium) reference number of the i-th neighbor

This method returns the i-th neighbor of the actual vertex in the shared adjacency list.

Definition at line 141 of file shmclient.hpp.

5.8.3.2 virtual osmium::unsigned_object_id_type justine::sampleclient::ShmClient::get_random_node (void) `[inline]`, `[virtual]`

This function returns a randomly chosen node from the map.

Returns

the randomly chosen node

This method may be useful if you want to add a new car to the map.

Definition at line 110 of file shmclient.hpp.

5.8.3.3 size_t justine::sampleclient::ShmClient::num_edges (osmium::unsigned_object_id_type *from*) const `[inline]`

Returns the number of out edges of a given vertex.

Parameters

from	a given vertex

Returns

the number of edges

This method returns the size of the vector of neighboring vertices in the shared adjacency list.

Definition at line 126 of file shmclient.hpp.

5.8.3.4 void justine::sampleclient::ShmClient::start (boost::asio::io_service & *io_service*, const char $*$ *port*)

This function starts the client.

Parameters

io_service	
port	the TCP port of the traffic server

This method does the following: retrieves a value from shared memory, then establishes a connection with the traffic server, finally sends some client commands.

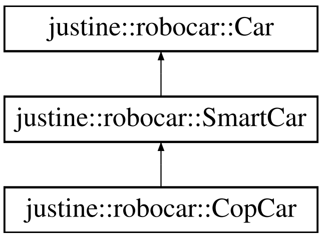
Definition at line 233 of file shmclient.cpp.

The documentation for this class was generated from the following files:

- src/shmclient.hpp
- src/shmclient.cpp

5.9 justine::robocar::SmartCar Class Reference

Inheritance diagram for justine::robocar::SmartCar:



Public Member Functions

- **SmartCar** (Traffic &traffic, CarType type, bool guided)
- virtual void **step** ()
- virtual void **init** ()
- virtual void **print** (std::ostream &os) const
- bool **get_guided** () const
- bool **set_route** (std::vector< unsigned int > &route)
- virtual void **nextEdge** (void)
- virtual void **nextGuidedEdge** (void)
- bool **set_fromto** (unsigned int from, unsigned int to)

Additional Inherited Members

5.9.1 Detailed Description

Definition at line 163 of file car.hpp.

The documentation for this class was generated from the following files:

- src/car.hpp
- src/car.cpp

5.10 justine::robocar::SmartCity Class Reference

Public Member Functions

- **SmartCity** (const char ∗osm_file, const char ∗shm_segment, const char ∗map_file)

- **SmartCity** (const char ∗osm_file, const char ∗shm_segment)
- void **processes** ()
- virtual void **city_run** ()
- double busWayLength (bool verbose)

  This function gives a list of all the bus services operating in a given city.

Protected Attributes

- boost::interprocess::managed_shared_memory ∗ **segment**
- boost::interprocess::offset_ptr
  < shm_map_Type > **shm_map**
- int **m_delay** {5000}
- bool **m_run** {true}

Friends

- std::ostream & **operator**<< (std::ostream &os, SmartCity &t)

5.10.1 Detailed Description

Definition at line 83 of file smartcity.hpp.

The documentation for this class was generated from the following files:

- src/smartcity.hpp
- src/smartcity.cpp

5.11 justine::robocar::Traffic Class Reference

Public Member Functions

- **Traffic** (int size, const char ∗shm_segment, double catchdist, TrafficType type=TrafficType::NORMAL, int minutes=10)
- void **processes** ()
- std::string **get_title** (std::string name)
- virtual
  osmium::unsigned_object_id_type **node** ()
- virtual void **traffic_run** ()
- void **steps** ()
- void **pursuit** (void)
- size_t **nedges** (osmium::unsigned_object_id_type from) const
- osmium::unsigned_object_id_type **alist** (osmium::unsigned_object_id_type from, int to) const
- int **alist_inv** (osmium::unsigned_object_id_type from, osmium::unsigned_object_id_type to) const
- osmium::unsigned_object_id_type **salist** (osmium::unsigned_object_id_type from, int to) const
- void **set_salist** (osmium::unsigned_object_id_type from, int to, osmium::unsigned_object_id_type value)
- osmium::unsigned_object_id_type **palist** (osmium::unsigned_object_id_type from, int to) const
- bool **hasNode** (osmium::unsigned_object_id_type node)
- void **start_server** (boost::asio::io_service &io_service, unsigned short port)
- void **cmd_session** (boost::asio::ip::tcp::socket sock)
- osmium::unsigned_object_id_type **naive_node_for_nearest_gangster** (osmium::unsigned_object_id_type from, osmium::unsigned_object_id_type to, osmium::unsigned_object_id_type step)
- double **dst** (osmium::unsigned_object_id_type n1, osmium::unsigned_object_id_type n2) const

- double **dst** (double lon1, double lat1, double lon2, double lat2) const
- void **toGPS** (osmium::unsigned_object_id_type from, osmium::unsigned_object_id_type to, osmium::unsigned_object_id_type step, double ∗lo, double ∗la) const
- osmium::unsigned_object_id_type **naive_nearest_gangster** (osmium::unsigned_object_id_type from, osmium::unsigned_object_id_type to, osmium::unsigned_object_id_type step)
- TrafficType **get_type** () const
- int **get_time** () const

Protected Attributes

- boost::interprocess::managed_shared_memory ∗ **segment**
- boost::interprocess::offset_ptr
< shm_map_Type > **shm_map**
- int **m_delay** {200}
- bool **m_run** {true}
- double **m_catchdist** {15.5}

Friends

- std::ostream & **operator**<< (std::ostream &os, Traffic &t)

5.11.1 Detailed Description

Definition at line 81 of file traffic.hpp.

The documentation for this class was generated from the following files:

- src/traffic.hpp
- src/traffic.cpp

5.12 yy_buffer_state Struct Reference

Public Attributes

- std::istream ∗ **yy_input_file**
- char ∗ **yy_ch_buf**
- char ∗ **yy_buf_pos**
- yy_size_t **yy_buf_size**
- int **yy_n_chars**
- int **yy_is_our_buffer**
- int **yy_is_interactive**
- int **yy_at_bol**
- int yy_bs_lineno
- int yy_bs_column
- int **yy_fill_buffer**
- int **yy_buffer_status**

5.12.1 Detailed Description

Definition at line 208 of file carlexer.cc.

5.12.2 Member Data Documentation

5.12.2.1 int yy_buffer_state::yy_bs_column

The column count.

Definition at line 246 of file carlexer.cc.

5.12.2.2 int yy_buffer_state::yy_bs_lineno

The line count.

Definition at line 245 of file carlexer.cc.

The documentation for this struct was generated from the following file:

- src/carlexer.cc

5.13 yy_trans_info Struct Reference

Public Attributes

- flex_int32_t **yy_verify**
- flex_int32_t **yy_nxt**

5.13.1 Detailed Description

Definition at line 341 of file carlexer.cc.

The documentation for this struct was generated from the following file:

- src/carlexer.cc

Chapter 6

File Documentation

6.1 src/car.cpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <car.hpp>
#include <traffic.hpp>
#include <boost/iterator/iterator_concepts.hpp>
```

6.1.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai `nbatfai@gmail.com`

Version

0.0.10

6.1.2 LICENSE

6.1.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file car.cpp.

6.2 src/car.hpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <osmium/osm/types.hpp>
#include <iostream>
#include <vector>
#include <osmreader.hpp>
#include <algorithm>
```

Classes

- class justine::robocar::Car
- class justine::robocar::AntCar
- class justine::robocar::SmartCar
- class justine::robocar::CopCar

Enumerations

- enum **CarType** : unsigned int { **NORMAL** =0, **POLICE**, **GANGSTER**, **CAUGHT** }

6.2.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai `nbatfai@gmail.com`

Version

0.0.10

6.2.2 LICENSE

6.2.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file car.hpp.

6.3 src/carlexer.hpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <FlexLexer.h>
#include <iostream>
#include <sstream>
#include <cstring>
#include <cstdio>
#include <vector>
```

Classes

- class justine::robocar::CarLexer

6.3.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai `nbatfai@gmail.com`

Version

0.0.10

6.3.2 LICENSE

6.3.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file carlexer.hpp.

6.4 src/mainpage.h File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

6.4.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai nbatfai@gmail.com

Definition in file mainpage.h.

6.5 src/myshmclient-main.cpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <myshmclient.hpp>
#include <boost/program_options.hpp>
```

Functions

- int **main** (int argc, char ∗argv[ ])

6.5.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai nbatfai@gmail.com

Version

0.0.10

6.5.2 LICENSE

6.5.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file myshmclient-main.cpp.

6.6 src/myshmclient.cpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <myshmclient.hpp>
```

Variables

- char **data** [524288]

6.6.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai `nbatfai@gmail.com`

Version

0.0.10

6.6.2 LICENSE

6.6.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file myshmclient.cpp.

6.7 src/myshmclient.hpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <boost/interprocess/managed_shared_memory.hpp>
#include <boost/interprocess/allocators/allocator.hpp>
#include <boost/interprocess/containers/map.hpp>
#include <boost/interprocess/containers/vector.hpp>
#include <boost/interprocess/containers/string.hpp>
#include <smartcity.hpp>
#include <car.hpp>
#include <cstdlib>
#include <iterator>
#include <boost/asio.hpp>
#include <limits>
#include <memory>
#include <boost/graph/adjacency_list.hpp>
#include <boost/graph/graph_traits.hpp>
#include <boost/graph/dijkstra_shortest_paths.hpp>
#include <boost/graph/properties.hpp>
#include <boost/property_map/property_map.hpp>
#include <shmclient.hpp>
#include <algorithm>
#include <boost/graph/bellman_ford_shortest_paths.hpp>
#include <boost/graph/graphviz.hpp>
#include <fstream>
```

Classes

- class justine::sampleclient::MyShmClient

  A sample class used for testing the routing algorithms.

Typedefs

- typedef boost::adjacency_list
  < boost::listS, boost::vecS,
  boost::directedS,
  boost::property
  < boost::vertex_name_t,
  osmium::unsigned_object_id_type >
  , boost::property
  < boost::edge_weight_t, int > > **justine::sampleclient::NodeRefGraph**
- typedef boost::graph_traits
  < NodeRefGraph >
  ::vertex_descriptor **justine::sampleclient::NRGVertex**
- typedef boost::graph_traits
  < NodeRefGraph >
  ::vertex_iterator **justine::sampleclient::NRGVertexIter**
- typedef boost::graph_traits
  < NodeRefGraph >
  ::edge_descriptor **justine::sampleclient::NRGEdge**
- typedef boost::graph_traits
  < NodeRefGraph >
  ::edge_iterator **justine::sampleclient::NRGEdgeIter**
- typedef boost::property_map
  < NodeRefGraph,
  boost::vertex_name_t >::type **justine::sampleclient::VertexNameMap**
- typedef boost::property_map
  < NodeRefGraph,
  boost::vertex_index_t >::type **justine::sampleclient::VertexIndexMap**

- typedef
  boost::iterator_property_map
  < NRGVertex ∗, VertexIndexMap,
  NRGVertex, NRGVertex & > **justine::sampleclient::PredecessorMap**
- typedef
  boost::iterator_property_map
  < int ∗, VertexIndexMap, int,
  int & > **justine::sampleclient::DistanceMap**
- typedef boost::property_map
  < NodeRefGraph,
  boost::edge_weight_t >::type **justine::sampleclient::EdgeWeightMap**

6.7.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai nbatfai@gmail.com

Version

0.0.10

6.7.2 LICENSE

Copyright (C) 2014 Norbert Bátfai, batfai.norbert@inf.unideb.hu

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.

6.7.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file myshmclient.hpp.

6.8 src/osmreader.hpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <osmium/io/any_input.hpp>
#include <osmium/handler.hpp>
#include <osmium/visitor.hpp>
#include <osmium/osm/node.hpp>
#include <osmium/osm/way.hpp>
#include <osmium/osm/relation.hpp>
#include <osmium/index/map/sparse_mem_table.hpp>
#include <osmium/index/map/sparse_mem_map.hpp>
#include <osmium/handler/node_locations_for_ways.hpp>
#include <osmium/geom/haversine.hpp>
#include <osmium/geom/coordinates.hpp>
#include <iostream>
#include <map>
#include <set>
#include <vector>
#include <string>
#include <algorithm>
#include <fstream>
#include <exception>
#include <stdexcept>
```

Classes

- class justine::robocar::OSMReader

Typedefs

- typedef
  osmium::index::map::SparseMemMap
  < osmium::unsigned_object_id_type,
  osmium::Location > **justine::robocar::OSMLocations**
- typedef std::vector
  < osmium::unsigned_object_id_type > **justine::robocar::WayNodesVect**
- typedef std::map< std::string,
  WayNodesVect > **justine::robocar::WayNodesMap**
- typedef std::map
  < osmium::unsigned_object_id_type,
  osmium::Location > **justine::robocar::WaynodeLocations**
- typedef std::map
  < osmium::unsigned_object_id_type,
  WayNodesVect > **justine::robocar::Way2Nodes**
- typedef std::map
  < osmium::unsigned_object_id_type,
  WayNodesVect > **justine::robocar::AdjacencyList**
- typedef
  osmium::index::map::SparseMemMap
  < osmium::unsigned_object_id_type,
  int > **justine::robocar::Vertices**

6.8.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai nbatfai@gmail.com

Version

0.0.10

6.8.2 LICENSE

6.8.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file osmreader.hpp.

6.9 src/shmclient-main.cpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <shmclient.hpp>
#include <boost/program_options.hpp>
```

Functions

- int **main** (int argc, char ∗argv[])

6.9.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai nbatfai@gmail.com

Version

0.0.10

6.9.2 LICENSE

6.9.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file shmclient-main.cpp.

6.10 src/shmclient.cpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <shmclient.hpp>
```

Variables

- char **data** [524288]

6.10.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai nbatfai@gmail.com

Version

0.0.10

6.10.2 LICENSE

6.10.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file shmclient.cpp.

6.11 src/shmclient.hpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <boost/interprocess/managed_shared_memory.hpp>
#include <boost/interprocess/allocators/allocator.hpp>
#include <boost/interprocess/containers/map.hpp>
#include <boost/interprocess/containers/vector.hpp>
#include <boost/interprocess/containers/string.hpp>
#include <smartcity.hpp>
#include <car.hpp>
#include <cstdlib>
#include <iterator>
#include <boost/asio.hpp>
#include <limits>
#include <memory>
```

Classes

- class justine::sampleclient::ShmClient

  A sample class used for testing IPC mechanisms (SHM and sockets) which are used by the city emulator.

6.11.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai nbatfai@gmail.com

Version

0.0.10

6.11.2 LICENSE

6.11.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file shmclient.hpp.

6.12 src/smartcity.hpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <osmreader.hpp>
#include <thread>
#include <chrono>
#include <condition_variable>
#include <mutex>
#include <boost/interprocess/managed_shared_memory.hpp>
#include <boost/interprocess/allocators/allocator.hpp>
#include <boost/interprocess/containers/map.hpp>
#include <boost/interprocess/containers/vector.hpp>
#include <boost/interprocess/containers/string.hpp>
#include <exception>
#include <stdexcept>
#include <iomanip>
```

Classes

- class justine::robocar::SharedData
- class justine::robocar::SmartCity

Typedefs

- typedef
  boost::interprocess::managed_shared_memory::segment_manager **justine::robocar::segment_manager-_Type**
- typedef
  boost::interprocess::allocator
  < void, segment_manager_Type > **justine::robocar::void_allocator**
- typedef
  boost::interprocess::allocator
  < unsigned int,
  segment_manager_Type > **justine::robocar::uint_allocator**
- typedef
  boost::interprocess::vector
  < unsigned int, uint_allocator > **justine::robocar::uint_vector**

- typedef
  boost::interprocess::allocator
  < uint_vector,
  segment_manager_Type > **justine::robocar::uint_vector_allocator**
- typedef std::pair< const
  unsigned int, SharedData > **justine::robocar::map_pair_Type**
- typedef
  boost::interprocess::allocator
  < map_pair_Type,
  segment_manager_Type > **justine::robocar::map_pair_Type_allocator**
- typedef
  boost::interprocess::map
  < unsigned int, SharedData,
  std::less< unsigned int >
  , map_pair_Type_allocator > **justine::robocar::shm_map_Type**

6.12.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai `nbatfai@gmail.com`

Version

0.0.10

6.12.2 LICENSE

6.12.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file smartcity.hpp.

6.13 src/traffic-main.cpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <traffic.hpp>
#include <boost/program_options.hpp>
```

Functions

- int **main** (int argc, char ∗argv[])

6.13.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai nbatfai@gmail.com

Version

0.0.10

6.13.2 LICENSE

Copyright (C) 2014 Norbert Bátfai, batfai.norbert@inf.unideb.hu

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http-://www.gnu.org/licenses/.

6.13.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file traffic-main.cpp.

6.14 src/traffic.cpp File Reference

Justine - this is a rapid prototype for development of Robocar City Emulator.

```
#include <traffic.hpp>
```

6.14.1 Detailed Description

Justine - this is a rapid prototype for development of Robocar City Emulator.

Author

Norbert Bátfai nbatfai@gmail.com

Version

0.0.10

6.14.2 LICENSE

6.14.3 DESCRIPTION

Robocar City Emulator and Robocar World Championship

desc

Definition in file traffic.cpp.

Index