Project Structure

NOTE: Here, we don’t describe OpenDRIVE spec

For it, please refer to xsds and <https://releases.asam.net/OpenDRIVE/1.6.0/ASAM_OpenDRIVE_BS_V1-6-0.html>

# 1. Sub modules

NOTE: First, please focus on folders rather than individual files.

And then files inside folders if needed

And then classes/functions

**WrappedTypes, Common, Math, GeoCoordinate, OSM, OpenDRIVE, OCS**

WrappedTypes

We just wrapped Qt types. If we use another sdk/framework/library in future, we will change this part

And we will make minor changes for other parts.

This is the most basic sub module

Common

Some convenient classes/functions for common usage

Depends on WrappedTypes

Math

Point/Rectangle/Matrix …

Depends on Common

GeoCoordinate

For geographic coordinate systems

This wrapped PROJ4 library

Depends on Math

OSM

For OSM data

Depends on GeoCoordinate

OpenDRIVE

For OpenDRIVE

Depends on GeoCoordinate

OCS

GPS + OSM => OpenDRIVE

Depends on OSM and OpenDRIVE

# 2. OSM

OSM/Main

Just get OSM data using overpass API and analyze

Includes OSM Node

OSM/Way

OSM Way object and some related parts

OSM/PriorKnowledgeProvider

For OSM way, Fill missing info using prior knowledge

# 3. OpenDRIVE

This structure is fully equal to OpenDRIVE spec.

Has no special parts

Just structure definition and I/O

# 4. OCS

This is most/very important sub module

We will handle GPS file and get OSM data

Also will convert it to OpenDRIVE data

And will manage xodr file database (including memory-caching)

## 4.1. Database

Manage xodr I/O between memory and local storage

Has caching capability

## 4.2. Main

### 4.2.1. FileReader

Reads csv file

### 4.2.2. EgoTrajectoryCreator

Creates trajectory using KalmanFilter

#### 4.2.2.1. KalmanFilter

### 4.2.3. TrajectorySpliter

Split trajectory by grids

### 4.2.4. ExternalSourceProvider

Gets OSM data using OSM sub-module(overpass API)

### 4.2.5. OpenDRIVEBuilder::buildRoads

#### 4.2.5.1 OSMToOpenDRIVE::**buildRoads**

Convert GPS+OSM => OpenDRIVE content(header, roads (attrs, lanes, geometry))

Here, we don’t handle objects & signals & junctions & roadLinks

### 4.2.6. RoadFusor

Fuse old and new OpenDRIVE content

Focus on roads (attrs, lanes, geometry)

### 4.2.7. OpenDRIVEBuilder::updateOpenDRIVE

Convert OSM nodes => OpenDRIVE objects & signals

NOTE: Objects & signals belong to OpenDRIVE road

### 4.2.8. OpenDRIVEBuilder::updateJunctionLink

NOTE: Junction & RoadLink don’t belong to OpenDRIVE road

These belong to OpenDRIVE content

Once OpenDRIVE content or road is changed, then it’s and neighbour’s junctions & roadLinks must be updated

- Remove old junctions & roadLinks

- Fully reconstruct junction & roadLinks from 9 contents (itself, and 8 neighbours)

### 4.2.9. OpenDRIVEBuilder::updateNeighbour

Now it just calls updateJunctionLink

### 4.2.10. DatabaseConnector

This acts with DatabaseCacheManager

We keep OpenDRIVE content cache in memory

If need detailed algorithms for specific parts, will make it later

Almost workflows were explained previously