

Traffic Lights MarkerArrays/Perception Lights ON/OFF #3153

KhalilSelyan started this conversation in **Design**



KhalilSelyan on Dec 20, 2022

Collaborator

edited ▼

Traffic Lights

I'm currently working on a threejs 3D viz for autoware, while working on visualizing traffic lights. I realized that there was a bit of problem connecting traffic light spheres to the messages incoming from the perception stack to light them on.

The way the traffic lights are being published in this case, is such that each sphere (red,amber,green) is separate from eachother as such there is no direct way to figure out which 3 or 2 sphere belong with eachother in a cluster of an actual traffic light. You would have ids : 8828, 8879, 8832 etc... the only thing that builds a connection to them is their position and the 3d text object that labels which id this traffic light should have.



Green Light Set to ON for traffic light with ID 384

```
ros2 topic echo /perception/traffic_light_recognition/traffic_light_recognition/header:
  stamp:
    sec: 1671531101
```

Category



Design

Labels

component:percept...

2 participants



```
      nanosec: 737712828
      frame_id: ''
signals:
- map_primitive_id: 384
  lights:
  - color: 3
    shape: 5
    status: 14
    confidence: 1.0
```

Searching for id 384 on the vector map markers array

```
ros2 topic echo /map/vector_map_marker
```



```
- header:
  stamp:
    sec: 0
    nanosec: 0
  frame_id: map
ns: traffic_light_id
id: 834
type: 9
action: 0
pose:
  position:
    x: 3754.2395501458377
    y: 73742.62399991415
    z: 22.621
  orientation:
    x: 0.0
    y: 0.0
    z: 0.0
    w: 1.0
scale:
  x: 0.0
  y: 0.0
  z: 1.0
color:
  r: 0.5
  g: 0.5
  b: 0.5
  a: 0.800000011920929
lifetime:
  sec: 0
  nanosec: 0
frame_locked: true
points: []
colors: []
texture_resource: ''
texture:
  header:
    stamp:
      sec: 0
      nanosec: 0
    frame_id: ''
  format: ''
  data: []
uv_coordinates: []
text: '384'
mesh_resource: ''
```

```
mesh_file:
  filename: ''
  data: []
mesh_use_embedded_materials: false
```

The ID of 384 is linked to the 3d text object of said traffic light, but not the actual spheres themselves.

Example of what one of the traffic light sphere markers ID is like : 8828

```
- header:
  stamp:
    sec: 0
    nanosec: 0
  frame_id: map
ns: traffic_light
id: 8828
type: 2
action: 0
pose:
  position:
    x: 3702.145300047472
    y: 73710.75489951344
    z: 24.91
  orientation:
    x: 0.0
    y: 0.0
    z: 0.0
    w: 1.0
scale:
  x: 0.30000001192092896
  y: 0.30000001192092896
  z: 0.30000001192092896
color:
  r: 0.30000001192092896
  g: 0.0
  b: 0.0
  a: 0.30000001192092896
lifetime:
  sec: 0
  nanosec: 0
frame_locked: true
points: []
colors: []
texture_resource: ''
texture:
  header:
    stamp:
      sec: 0
      nanosec: 0
    frame_id: ''
  format: ''
  data: []
uv_coordinates: []
text: '384'
mesh_resource: ''
mesh_file:
  filename: ''
```



```
data: []
mesh_use_embedded_materials: false
```

Issues

When receiving a message from the perception stack that one of the lights is on, there isn't a way to call the sphere marker to change it's opacity as the message's linked id is to the 3d text label and not the actual sphere.

Proposed Solutions

1. The spheres' opacity gets automatically updated, and the `/map/vector_map_marker` topic republishes on change. (but this might have a performance issue as everything in the map would technically rerender)
2. Separate the traffic lights from the rest of the `/map/vector_map_marker` markerArrays as such if they get automatically updated only the traffic light get rerendered instead of all 3d environment.
3. Perception stack message publishers, publish sphere IDs instead of label IDs in the following shape:

- Original:

```
header: {
  stamp: {
    sec: number,
    nanosec: number,
  },
  frame_id: string
},
signals: {
  map_primitive_id: number,
  lights: {
    color: number,
    shape: number,
    status: number,
    confidence: number
  },
}
```



- Updated:

```
header: {
  stamp: {
    sec: number,
    nanosec: number,
  },
  frame_id: string
},
signals: {
  map_primitive_sphere_id: number,
  lights: {
    shape: number,
```



```
status: number,  
confidence: number  
},  
}
```

By changing the ID, now the message would specify which light specifically is being turned on. Hence, removing the color attribute since the sphere id now targets a specific color.

↑ 1

1 comment · 2 replies

Oldest

Newest

Top



isamu-takagi on Dec 20, 2022

Maintainer

edited ▼

I think proposed solution 1 or 2 is better because perception message should not be affected by visualization. ~~In addition, it is difficult to implement proposed solution 3. The map_primitive_sphere_id such as 8828, 8879, 8832 is the visualization marker IDs, not the lanelet map IDs. So only lanelet2_map_visualization_node know about this correspondence.~~

In the current implementation, the visualization marker ID and lanelet ID are the same.

https://github.com/autowarefoundation/autoware_common/blob/main/tmp/lanelet2_extension/lib/visualization.cpp#L115

↑ 2

2 replies



isamu-takagi on Dec 20, 2022

Maintainer

And I am currently designing a traffic signal interface including V2I. I will make a new discussion soon. Please participate if you like.



isamu-takagi on Dec 22, 2022

Maintainer

edited ▼

[autowarefoundation/autoware.universe#2567](https://github.com/autowarefoundation/autoware.universe#2567)

I created an issue about traffic signal interface. For visualization, I think it can be achieved by combining traffic signal status, nearest regulatory element ID, and correspondence between regulatory element ID and light bulbs ID from lanelet map.