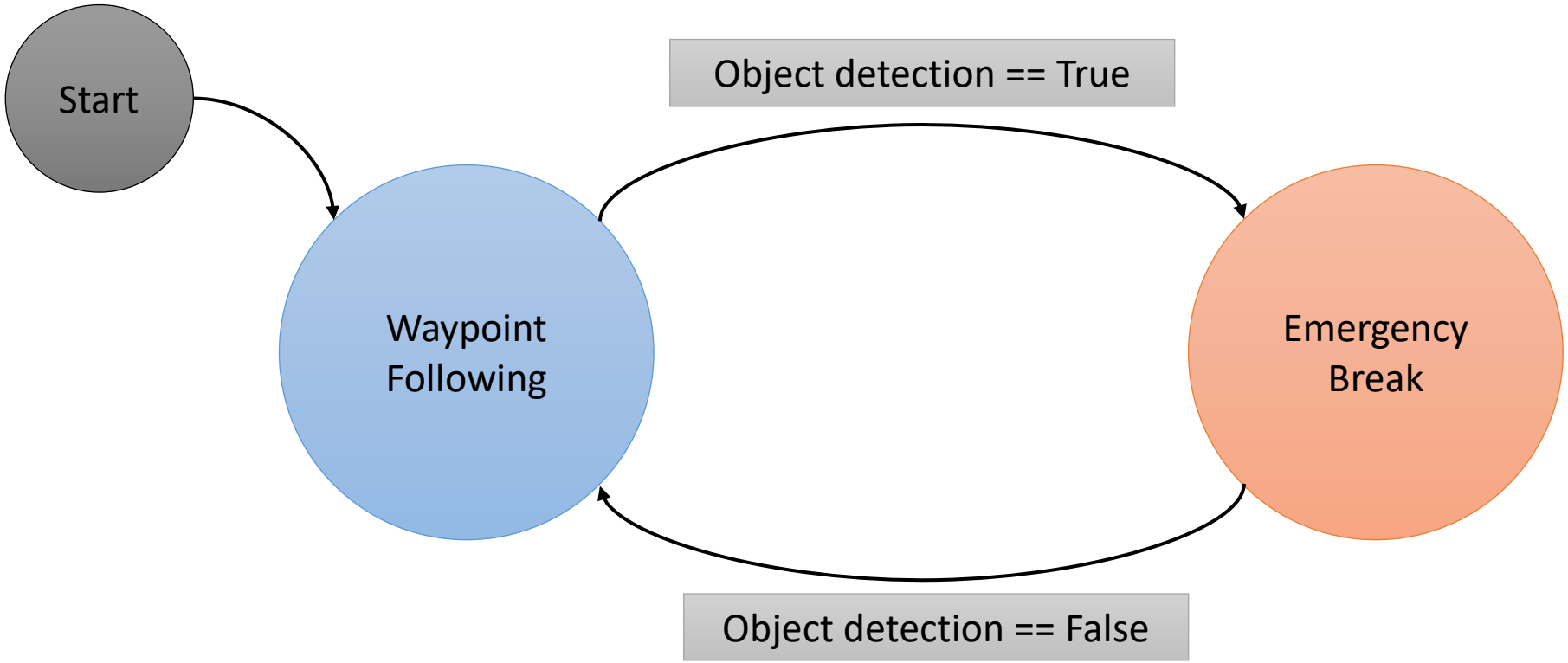
NETH CARLA Challenge demo

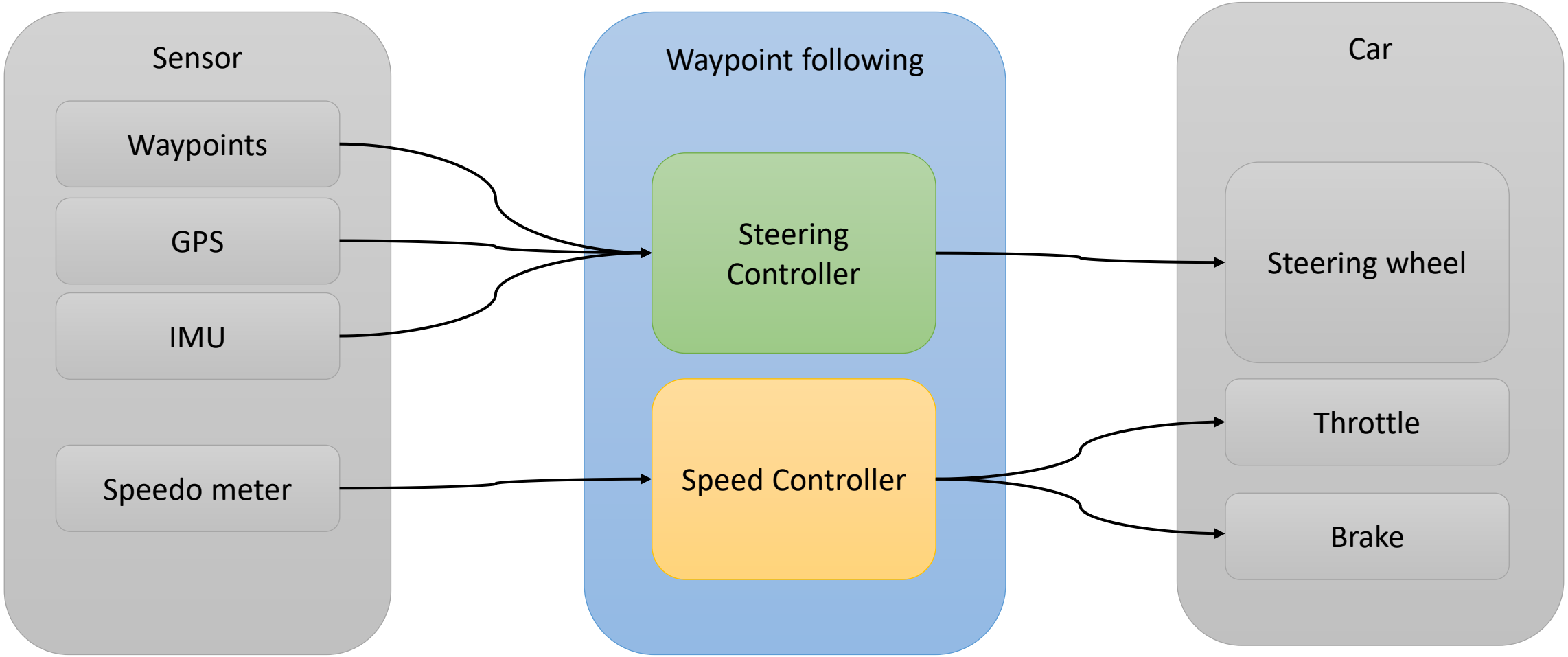
Outline

- Demo Agent Overview
  - Waypoint following
  - Emergency break using LiDAR
- Cybertruck dimension
- Sensor data (TBD.)

Demo Agent Overview

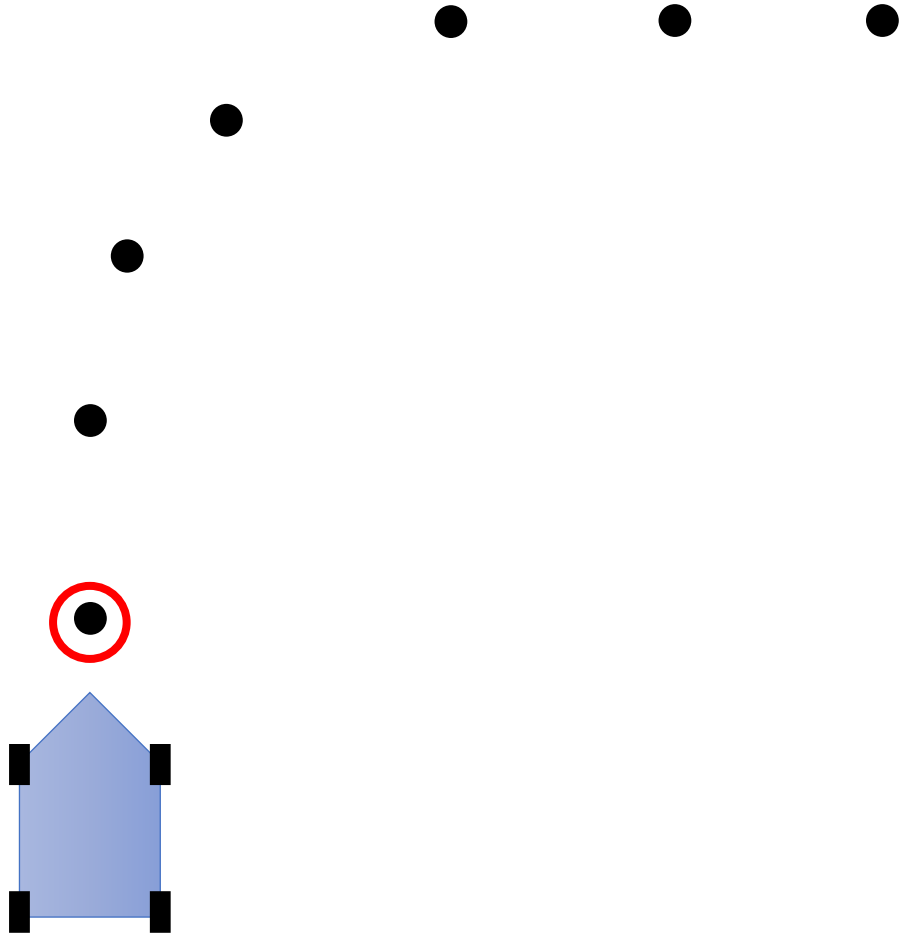


Waypoint following Overview



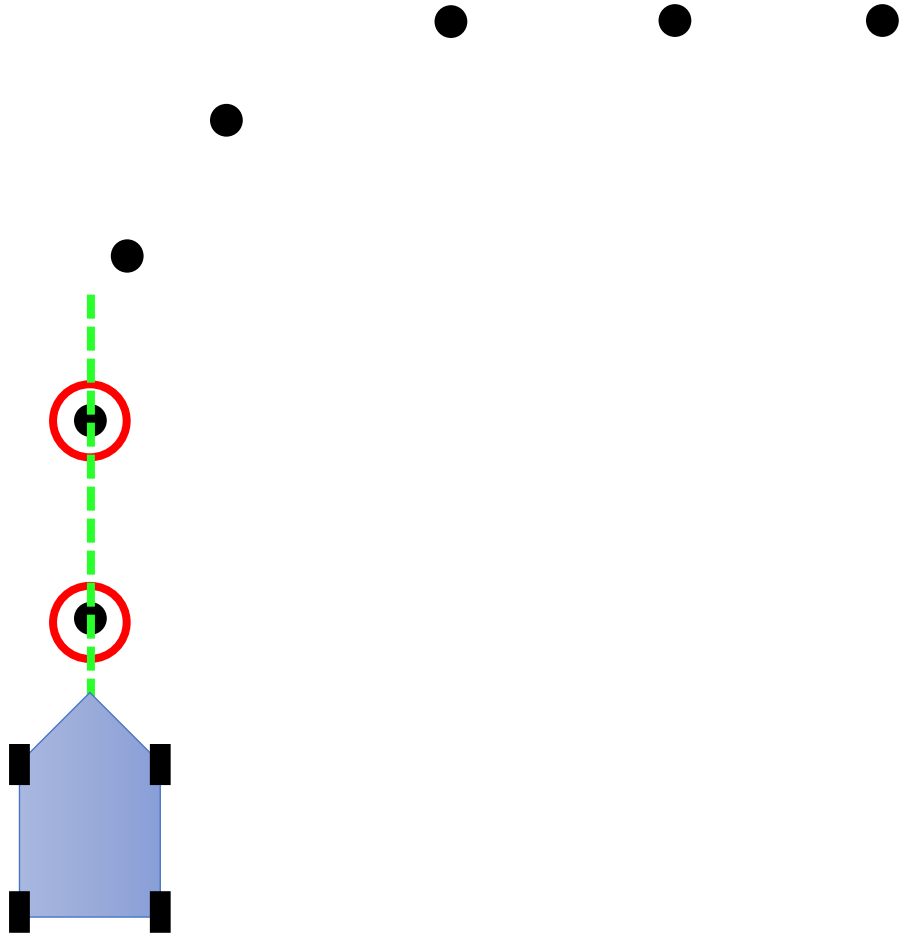
Waypoint following (Steering)

1. Find relate waypoint base on our GPS location.



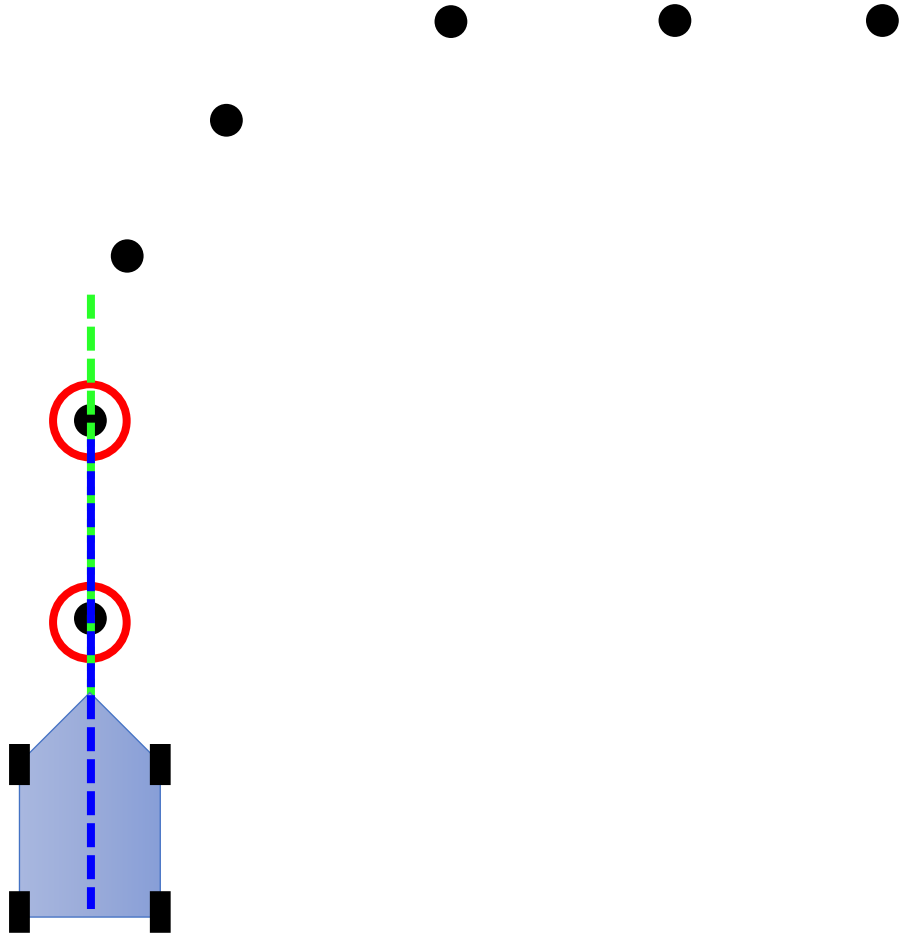
Waypoint following (Steering)

2. Find next waypoint and conected with first with line.



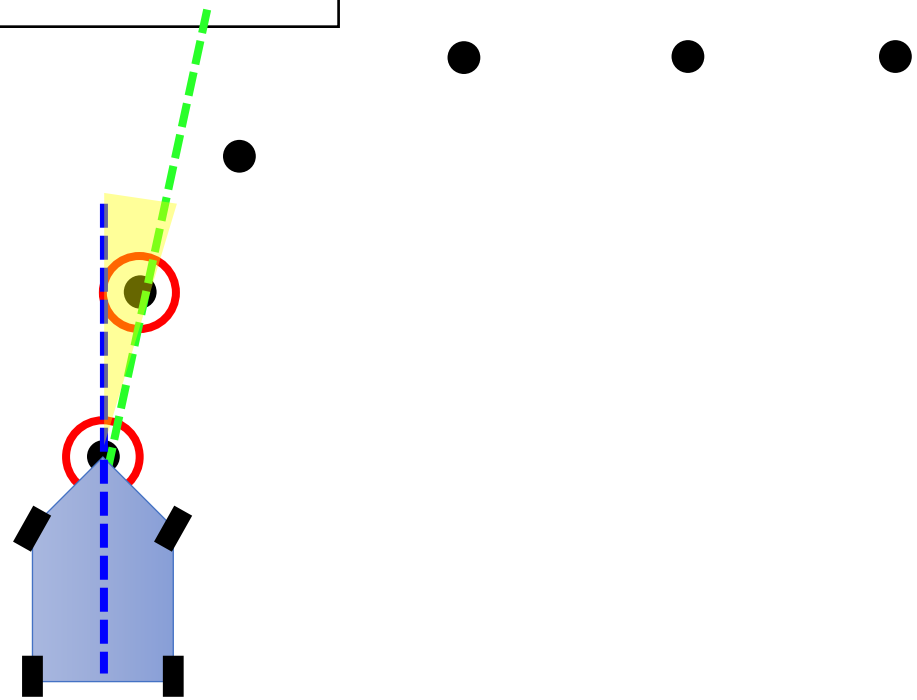
Waypoint following (Steering)

3. Find the difference angle between waypoint heading and car heading (From IMU[compass]). Then adjudst steering angle.



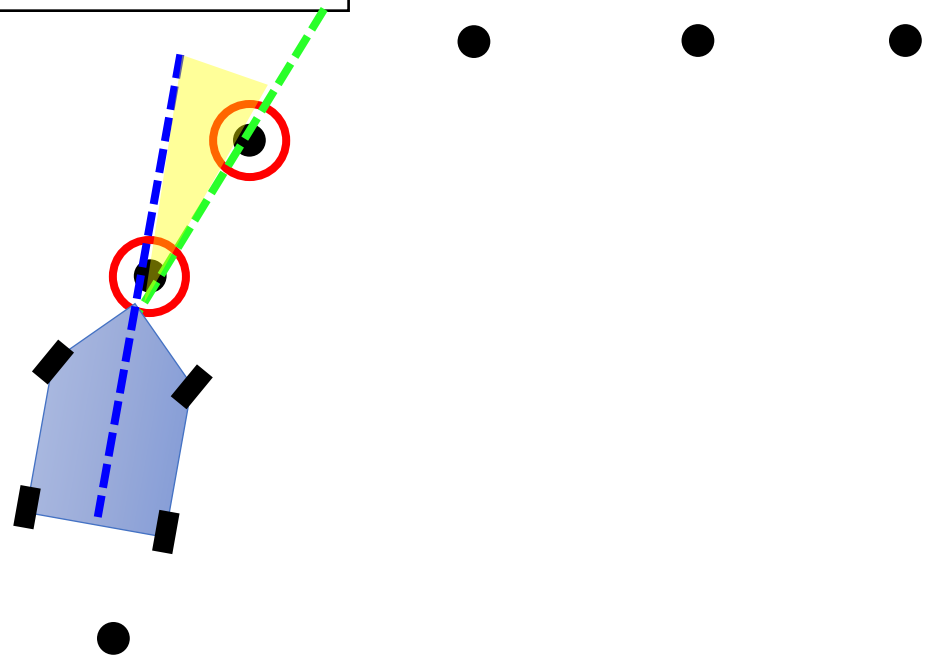
Waypoint following (Steering)

4. Step through next waypoint. And do the same as previous step.

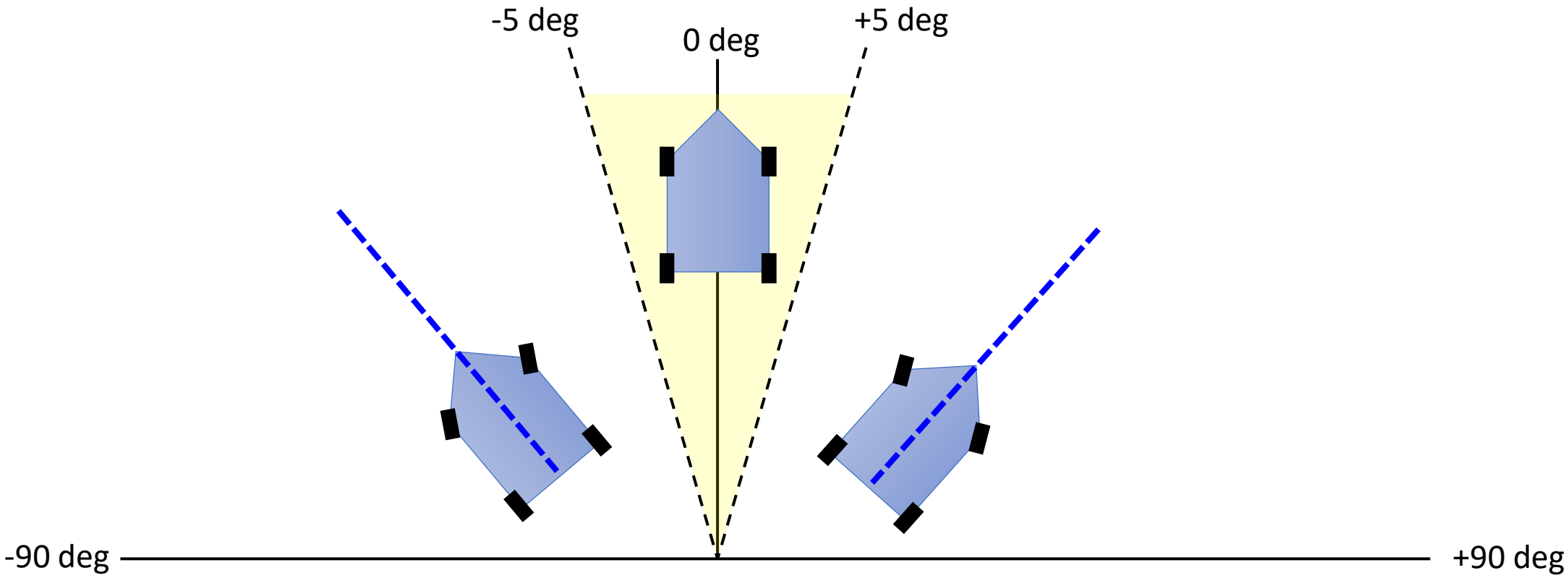


Waypoint following (Steering)

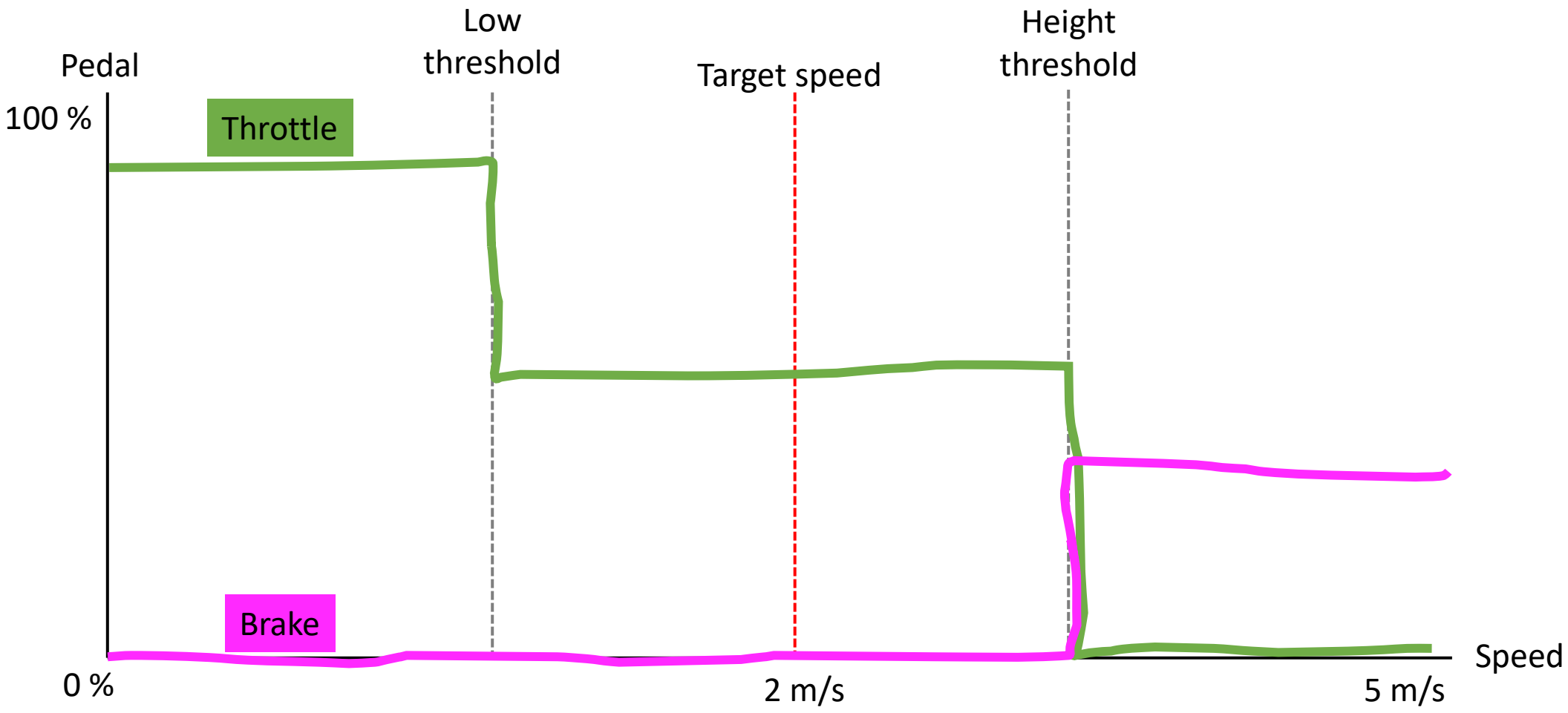
5. Step through next waypoint. And do the same as previous step. Until end of the waypoints.



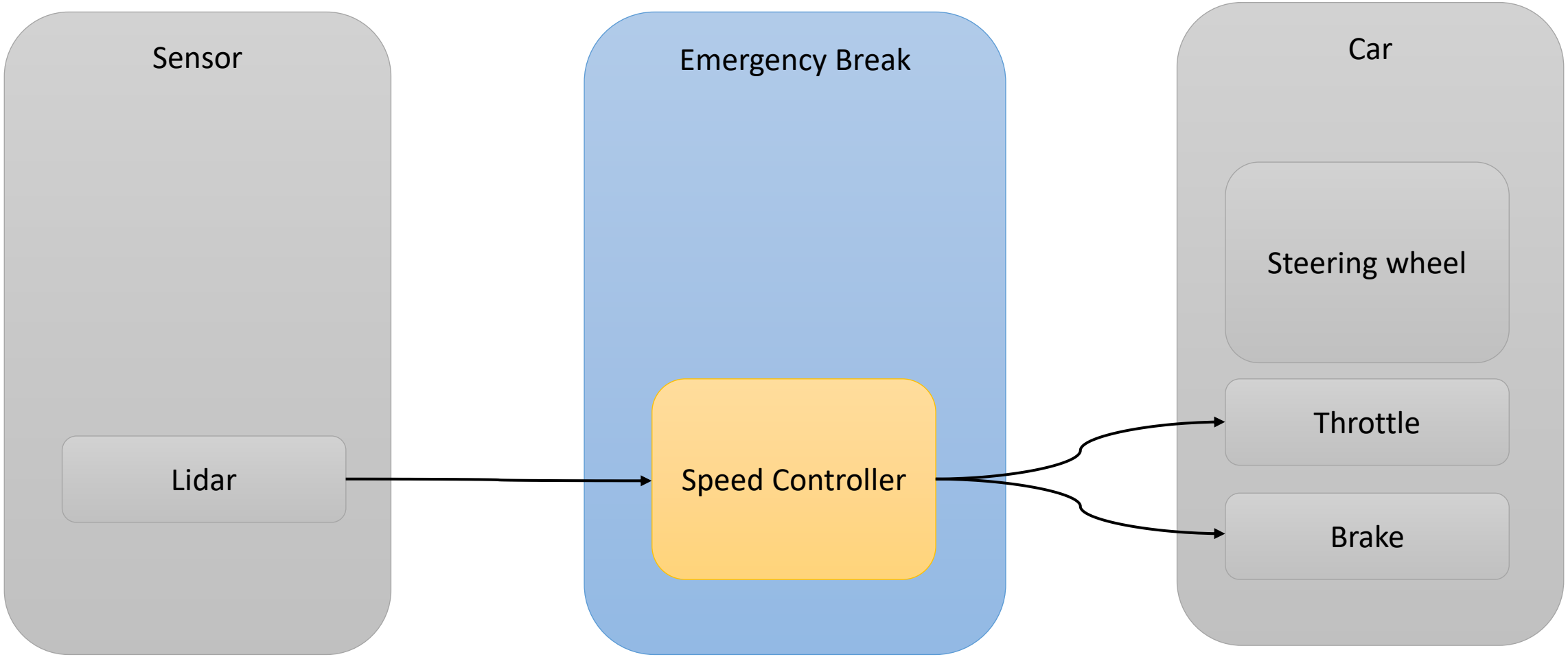
Waypoint following (Steering controller)



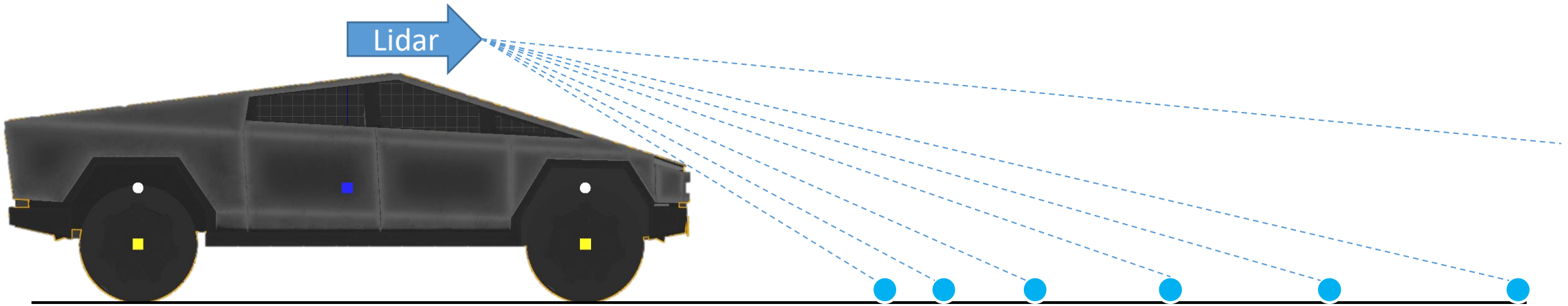
Waypoint following (Speed Controller)



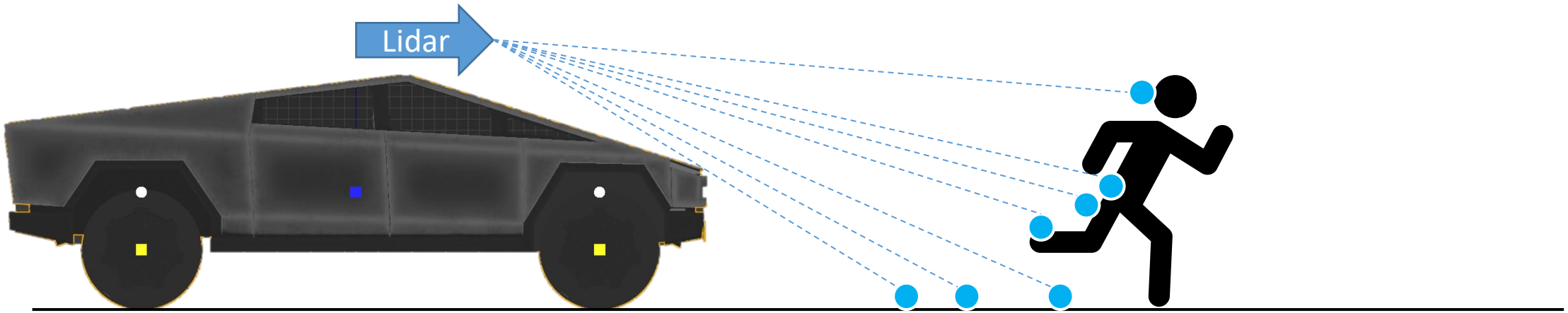
Emergency Break Overview



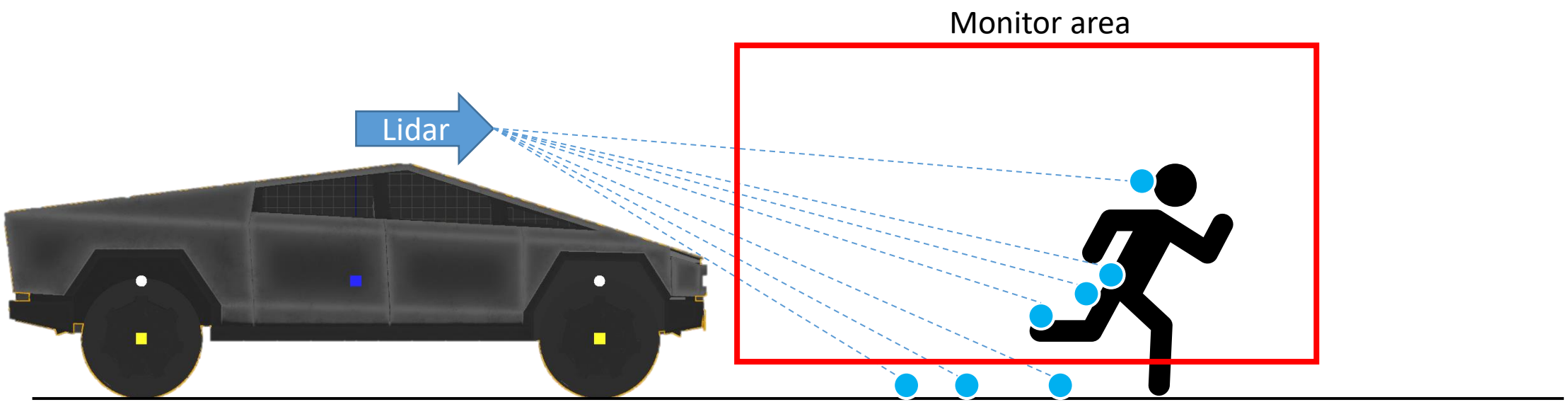
Emergency Break (Lidar Normal)



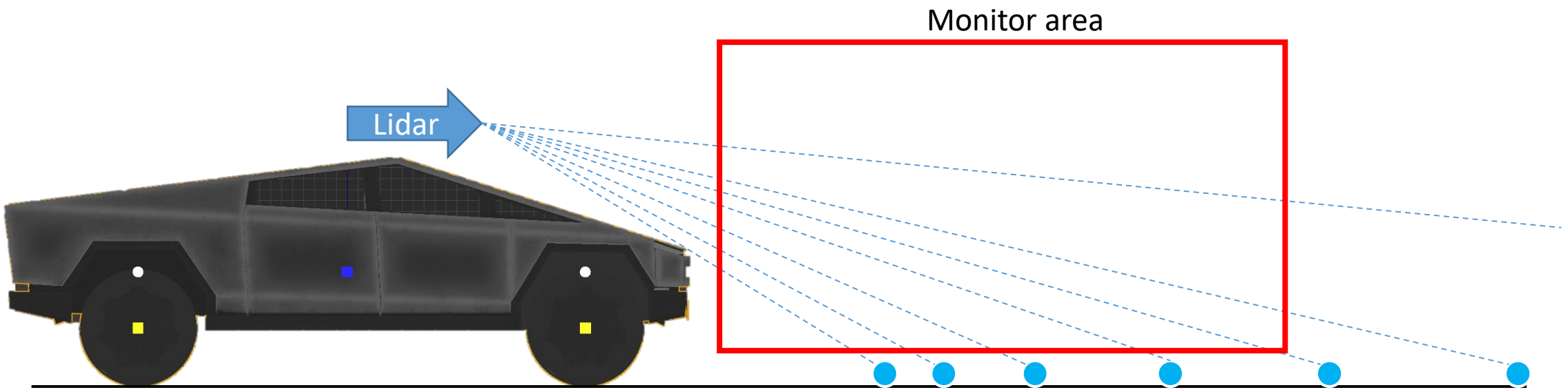
Emergency Break (Lidar Obstacle detected)



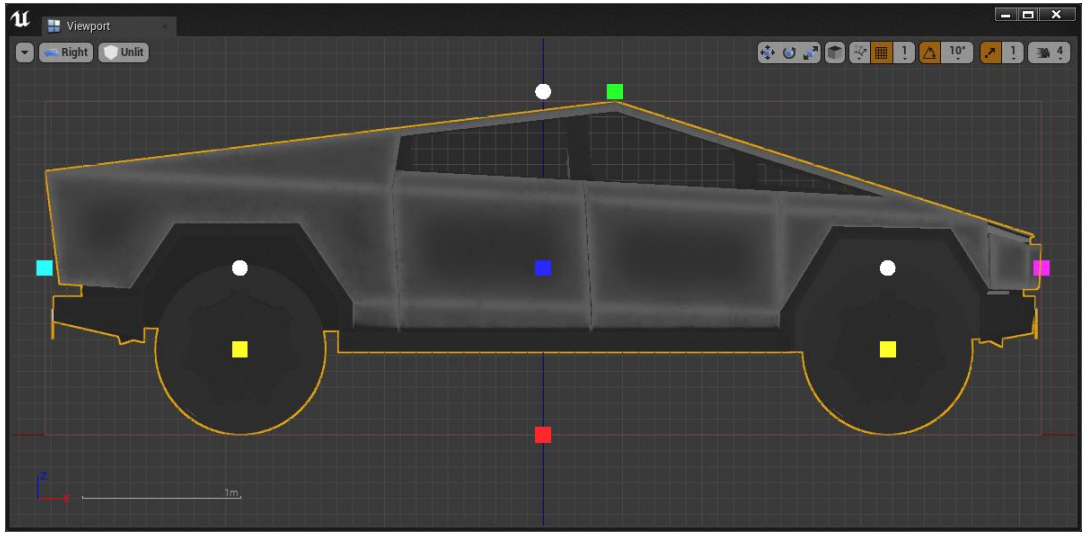
Emergency Break (Lidar Obstacle detected)



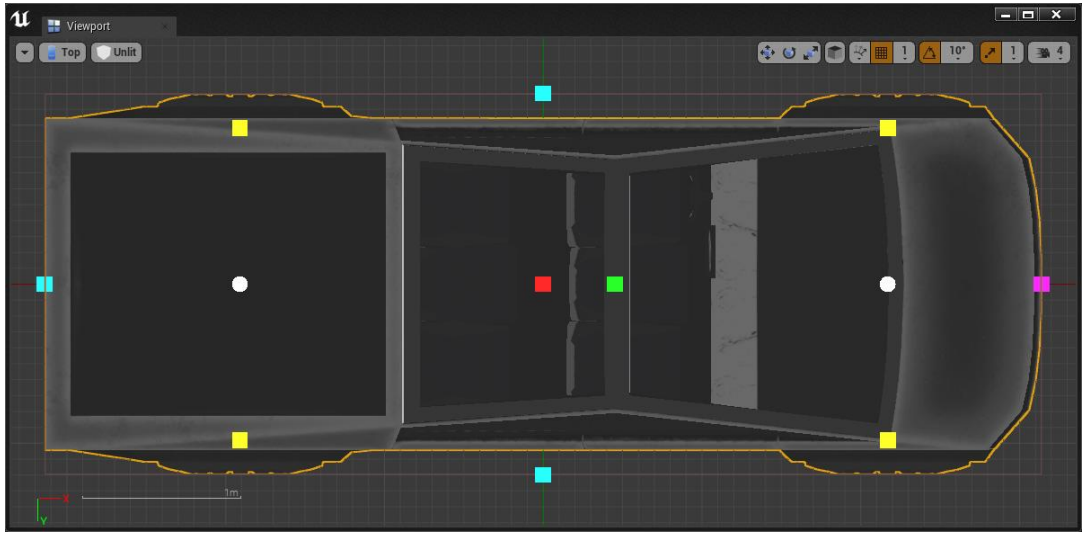
Emergency Break (Lidar Normal)

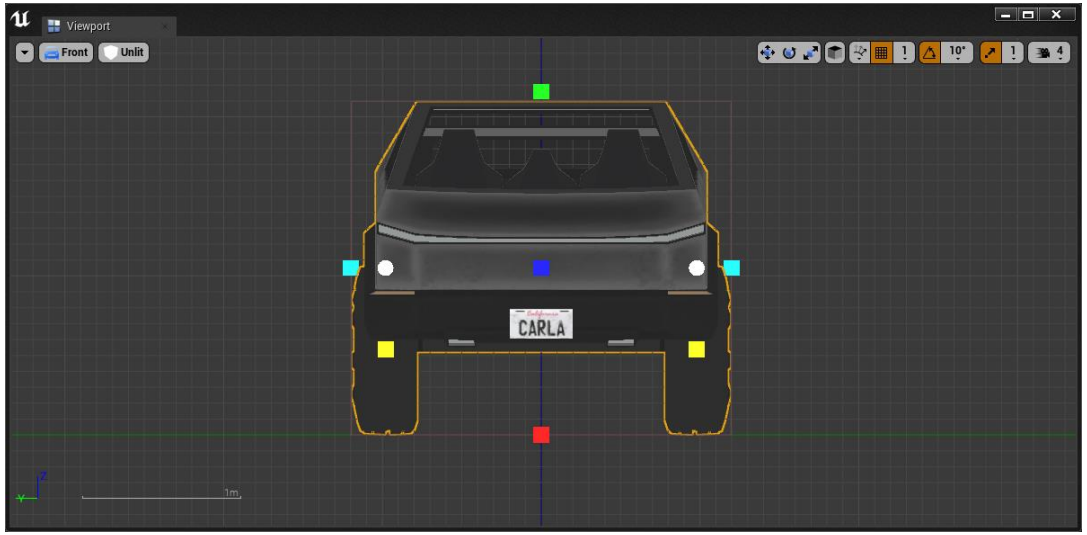


Cybertruck dimension

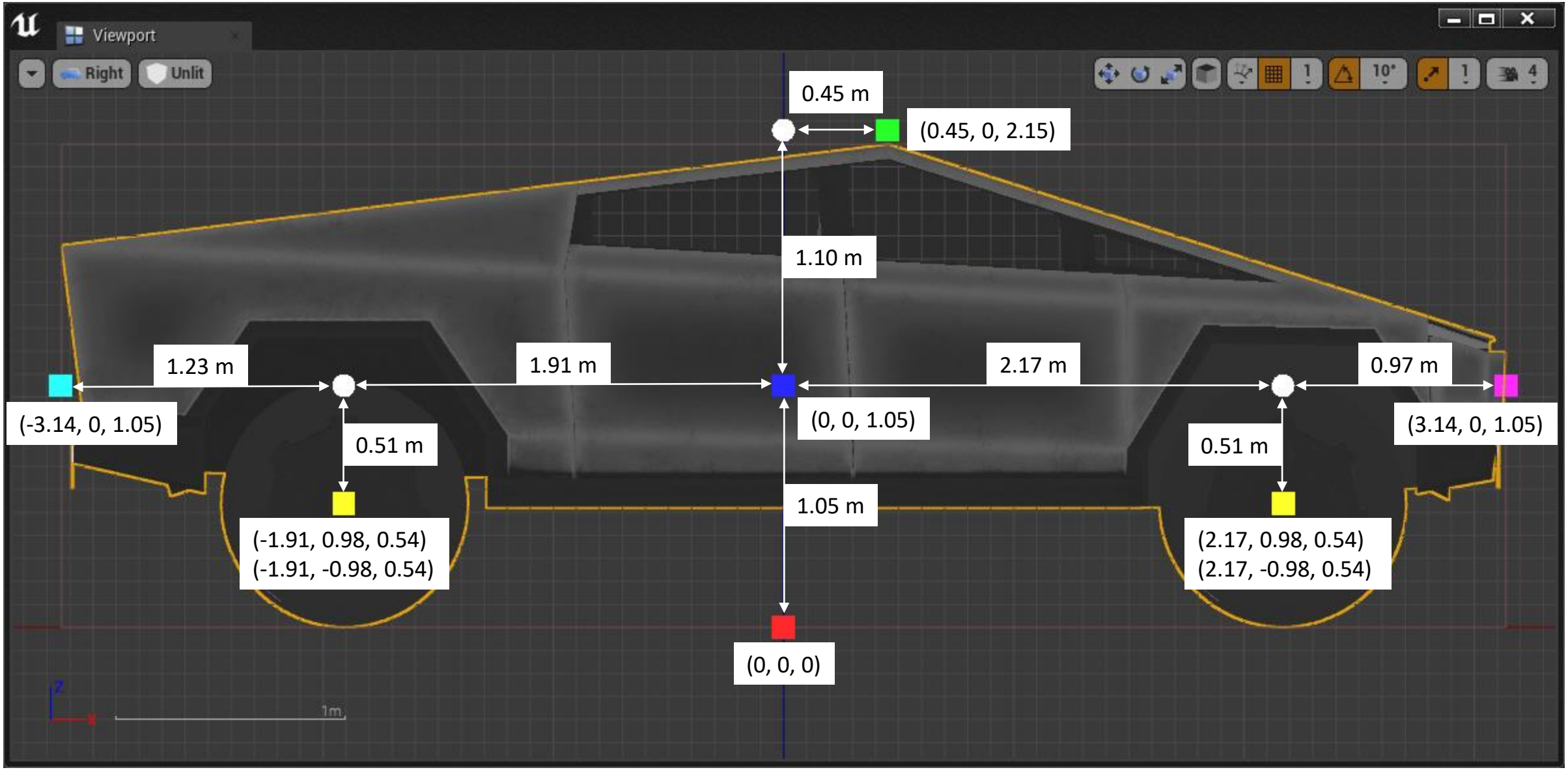


- Origin
- Car center
- Sensors
- Wheels
- Car front
- Car back and side
- Reference point

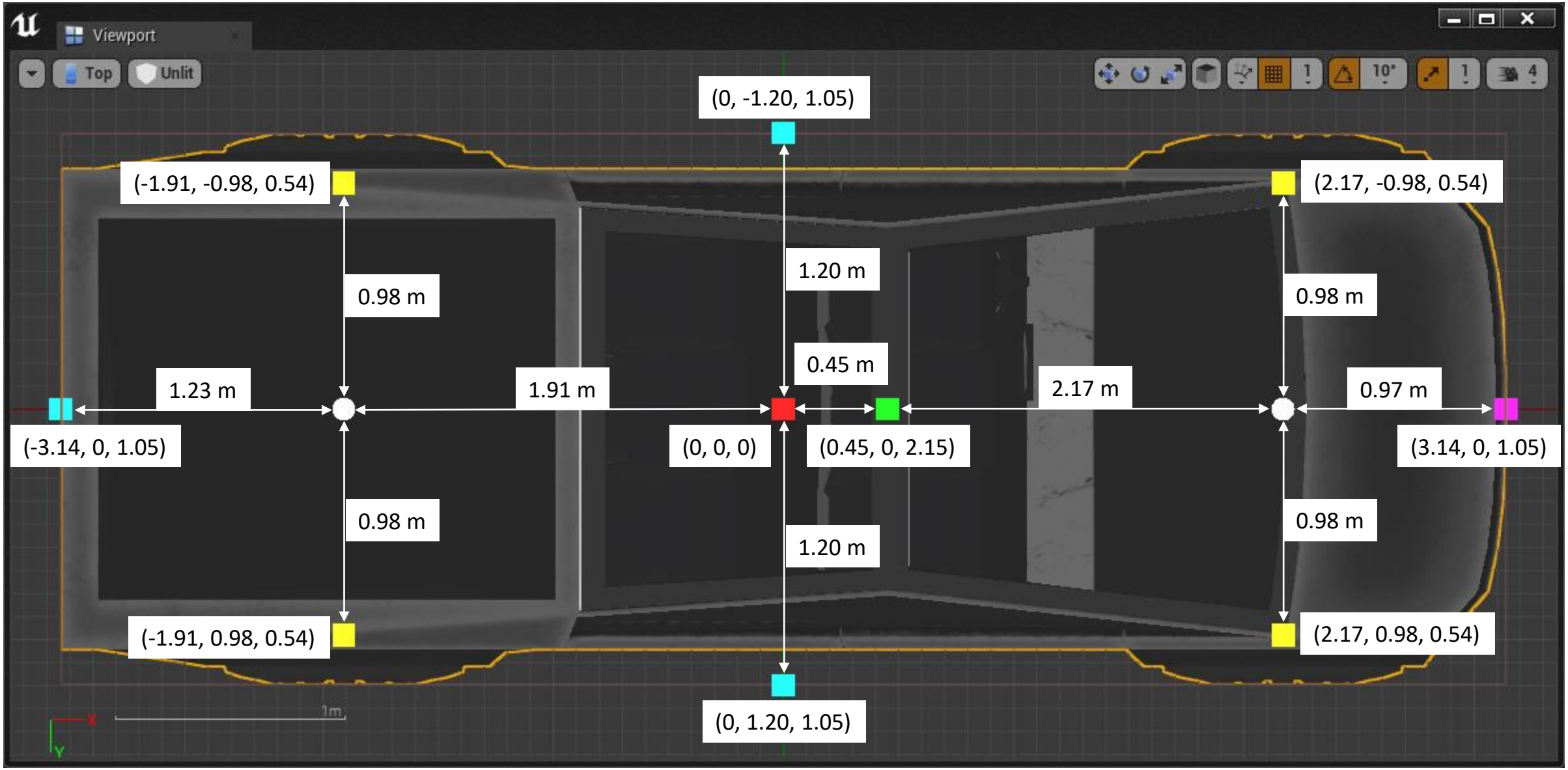




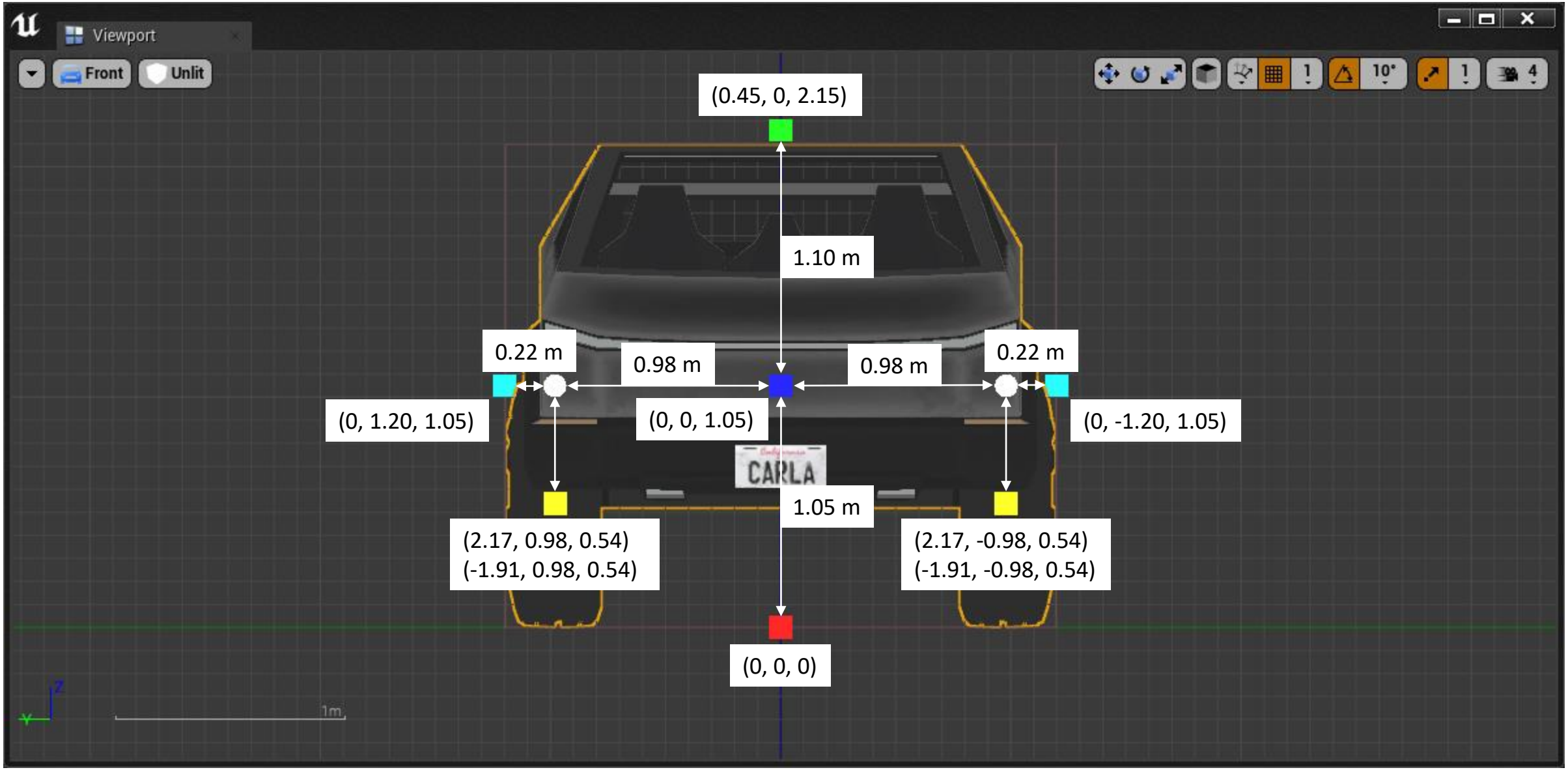
Cybertruck dimension (side view)



Cybertruck dimension (top view)



Cybertruck dimension (front view)



Sensor data

- TBD.

Sensor data (Waypoint)

Sensor Data

({'lat': 6.06789539148167e-05, 'lon': 0.00103811046000933 5, 'z': 0.1097366809844970 7}, <RoadOption.LANEFOLLOW: 4>)

How to get value

- lat = _global_plan[i][0]['lat']
- lon = _global_plan[i][0]['lon']
- z = _global_plan[i][0]['z']
- RoadOption = _global_plan[i][1]

i is index of waypoint

- [lat, lon, z] = _global_plan[i][0].values()
- RoadOption = _global_plan[i][1]

Sensor data (GPS)

```
Sensor Data
(204, array([4.92380653e-05, 1.03663280e-03, 2.14793539e+00]))
```

How to get value

- `[frame, [lat, lon, z]] = input_data['GPS’]`
- `frame = input_data['GPS'][0]`
- `[lat, lon, z] = input_data['GPS'][1]`
- `lat = input_data['GPS'][1][0]`
- `lon = input_data['GPS'][1][1]`
- `z = input_data['GPS'][1][2]`

Sensor data (IMU)

Sensor Data

(386, array([-5.84830472e-04, 1.24953993e-04, 9.81373119e+00,
-8.90446594e-04, -4.19984921e-04, 8.40062334e-04, 4.72732639e+00]))

How to get value

- [frame, [ax, ay, az, gx, gy, gz, compass]] = *input_data*['IMU']
  Accelerometer, Gyroscope, Compass
- frame = *input_data*['IMU'][0]
- [ax, ay, az, gx, gy, gz, compass] = *input_data*['IMU'][1]
- ax = *input_data*['IMU'][1][0]
- ay = *input_data*['IMU'][1][1]
- az = *input_data*['IMU'][1][2]
- gx = *input_data*['IMU'][1][3]
- gy = *input_data*['IMU'][1][4]
- gz = *input_data*['IMU'][1][5]
- compass = *input_data*['IMU'][1][6]