CMPE 565 AUTONOMOUS ROBOTS
Assignment 2: Sensor Visualization and Obstacle Avoidance

Deadline: 22.02.2018

For this assignment, you are expected to get data from sensors and process it. If you successfully completed the assignment 1, you have a fully working VREP and ROS.

Do not forget to update your repositories from assignment repositories.

```
# use the following command if you changed something in your repository and
   did not commit (info about stashing: https://git-scm.com/book/no-nb/v1/Git-
   Tools-Stashing)
git stash
# ros-ws repository
git pull http://robot.cmpe.boun.edu.tr/gitlab/cmpe-565-2018/ros-ws.git
# if you did 'git stash' as above, then run the following
git stash apply
# simulation repository
git stash
git pull http://robot.cmpe.boun.edu.tr/gitlab/cmpe-565-2018/simulation.git
# if you did git stash run the following
git stash apply
```

After update, you should have one new ros package "obstacle_avoidance" and two new simulation scenes "assignment2_robotino_kinect" and "assignment2_robotino_laser".

1 Sensor Visualization

- You will create a ros package as described here by following the tutorial `http://wiki.ros.org/ROS/Tutorials/CreatingPackage`.
- Your package will listen the "sensor_msgs/PointCloud2" data and print to the console the "width" and "height" field of the message. **HINT**: The name of the topic for the point cloud data is "/vrep/kinectPoints" and you can use this tutorial `http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29`
- You will open the scene assignment2_robotino_kinect from simulation/scenes folder.
- You will run teleoperation node with the following code (like in assignment 1):
  roslaunch vrep_apps base.launch

- You will run rviz and add kinect point cloud from the menu by chosing add by topic and PointCloud2.
- In a new console you also run your own package as ”rosrun package_name node_name”
- You will move the robot with keyboard via vrep_teleop node and record your whole screen and upload to youtube (Make sure that your video includes rviz screen with point cloud data and also console output produced with your package)
- You will add your youtube video link to your wiki page and also explain your package and the experience.

2 Obstacle Avoidance

- You will implement a robot controller for obstacle avoidance.
- The expected behavior of the robot is to move forward if there is no obstacle otherwise avoid the obstacle.
- You will modify the ”obstacle_avoidance” package and write a publisher and a subscriber
- You will implement your controller in the laser callback method called ”laserCallback”
- You will open the scene assignment2_robotino_laser from simulation/scenes folder.
- You will run obstacle_avoidance (rosrun obstacle_avoidance obstacle_avoidance) node.
- You will record a video showing behavior of robot.
- You will add your video link to your wiki page and write a short report explaining your approach and experience.

3 Bonus

- You will change obstacle_avoidance node to listen point cloud data (specifically sensor_msgs/PointCloud2) from kinect sensor.
- You will write a code to extract plane and obstacle from point cloud data. The basic approach might to calculate average height of taken point cloud data.
- Based on the obstacle detection, you can use the same controller which you developed for laser sensor.