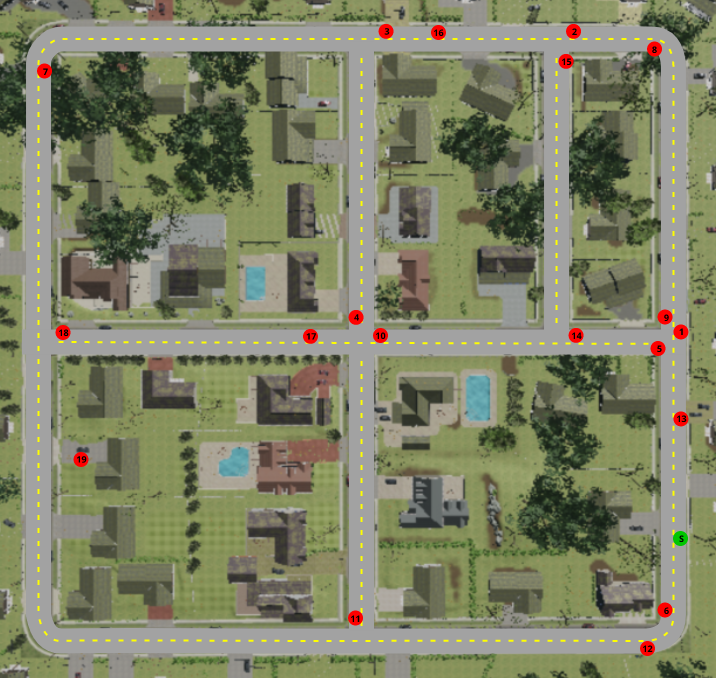
1/7/2021

* These example code came from <https://ecomarathon.swri.org/code>
* Instructions from the website:
* Important Note
* As per the rules, please note that some details, including the maximum simulation time, coordinates of the starting point and total number and coordinates of the available goal points will be announced on this website at the start of the competition, on January 11th. For practice purposes, below is an example of what that could look like.

### Requirements

* The software requires [Ubuntu Linux 18.04](https://ubuntu.com/download/desktop) and [ROS Melodic](http://wiki.ros.org/melodic) to build and run.

### Usage

* Important considerations:
* Project submissions are compiled in a fresh Ubuntu Linux 18.04 / ROS Melodic environment.
* Any software dependencies must be defined properly in the [catkin package manifest](http://wiki.ros.org/catkin/package.xml) and will be installed at build time by [rosdep](http://wiki.ros.org/rosdep).
* All ROS packages must [install](http://wiki.ros.org/catkin/CMakeLists.txt#Optional_Step:_Specifying_Installable_Targets) themselves when built; the source code will not be present in the simulation environment, only installed targets.
* All project submissions **must** have a package named shell\_simulation with a launch file named shell\_simulation.launch that requires no parameters or arguments in order to run; this is used as the entry point for launching the project.
* Uploaded projects **must** be named project.zip and contain only the source code of ROS packages.
* The simulation will automatically end after either all goals have been reached or after it has been running for 10 minutes.
* There are currently nineteen goal points defined with the following X/Y/Z coordinates, based on the /PhysXCar/odom\_local\_ned frame:
* [ [-87.0, 1.5, 0.0], [-213.0, -47.0, 0.0], [-213.0, -120.0, 0.0], [-94.0, -130.0, 0.0], [-81.7, -8.3, 0.0], [34.5, -2.6, 0.0], [-199.7, -254.0, 0.0], [-209.3, -8.1, 0.0], [-93.5, -2.0, 0.0], [-86.0, -119.4, 0.0], [33.7, -129.4, 0.0], [45.9, -9.8, 0.0], [-64.5, 1.8, 0.0], [-86.3, -37.3, 0.0], [-204.2, -45.8, 0.0], [-213.0, -60.0, 0.0], [-86.3, -139.2, 0.0], [-86.4, -248.2, 0.0], [-33.2, -240.4, 0.0] ]
* The first goal point is directly in front of the vehicle, so simply increasing the throttle is enough to reach it.
* 
* The following ROS topics are available within the simulation:
* Published topics:▪ /airsim\_node/origin\_geo\_point [airsim\_car\_ros\_pkgs/GPSYaw]▪ /airsim\_node/PhysXCar/collision\_count [std\_msgs/Int32]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/DepthPlanner/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/DepthPlanner [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/Scene/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/Scene [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/front\_middle\_bumblebee/DepthPlanner/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_middle\_bumblebee/DepthPlanner [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/front\_right\_bumblebee/Scene/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_right\_bumblebee/Scene [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/imu/Imu [sensor\_msgs/Imu]▪ /airsim\_node/PhysXCar/lidar/VLP16\_1 [sensor\_msgs/PointCloud2]▪ /airsim\_node/PhysXCar/odom\_local\_ned [nav\_msgs/Odometry]▪ /rosout\_agg [rosgraph\_msgs/Log]▪ /rosout [rosgraph\_msgs/Log]s▪ /tf\_static [tf2\_msgs/TFMessage]▪ /tf [tf2\_msgs/TFMessage]Subscribed topics:▪ /airsim\_node/PhysXCar/collision\_count [std\_msgs/Int32]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/DepthPlanner/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/DepthPlanner [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/Scene/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/Scene [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/front\_middle\_bumblebee/DepthPlanner/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_middle\_bumblebee/DepthPlanner [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/front\_right\_bumblebee/Scene/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_right\_bumblebee/Scene [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/imu/Imu [sensor\_msgs/Imu]▪ /airsim\_node/PhysXCar/lidar/VLP16\_1 [sensor\_msgs/PointCloud2]▪ /airsim\_node/PhysXCar/odom\_local\_ned [nav\_msgs/Odometry]▪ /brake\_command [std\_msgs/Float64]▪ /gear\_command [std\_msgs/String]▪ /handbrake\_command [std\_msgs/Bool]▪ /rosout\_agg [rosgraph\_msgs/Log]▪ /rosout [rosgraph\_msgs/Log]s▪ /steering\_command [std\_msgs/Float64]▪ /tf\_static [tf2\_msgs/TFMessage]▪ /tf [tf2\_msgs/TFMessage]▪ /throttle\_command [std\_msgs/Float64]These topics have data from sensors that can be used to observe the environment:▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/DepthPlanner/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/DepthPlanner [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/Scene/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_left\_bumblebee/Scene [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/front\_middle\_bumblebee/DepthPlanner/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_middle\_bumblebee/DepthPlanner [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/front\_right\_bumblebee/Scene/camera\_info [sensor\_msgs/CameraInfo]▪ /airsim\_node/PhysXCar/front\_right\_bumblebee/Scene [sensor\_msgs/Image]▪ /airsim\_node/PhysXCar/imu/Imu [sensor\_msgs/Imu]▪ /airsim\_node/PhysXCar/lidar/VLP16\_1 [sensor\_msgs/PointCloud2]And messages can be published to this topic to control the vehicle:▪ /brake\_command [std\_msgs/Float64] Valid values range from 0.0 (no brake) to 1.0 (full brake)▪ /gear\_command [std\_msgs/String] Valid values are "forward" or "reverse"▪ /handbrake\_command [std\_msgs/Bool] If set to "true", throttle will be ignored▪ /steering\_command [std\_msgs/Float64] Valid values range from -1.0 (full right) to 1.0 (full left)▪ /throttle\_command [std\_msgs/Float64] Valid values range from 0.0 (no throttle) to 1.0 (full throttle)