RViz – The tale of a migration to ROS 2.0

Andreas Holzner, andreas.holzner@tngtech.com
ROSCon 2018, Madrid, 29.09.2018
Who was involved?
Goal

- ROS 1 -> ROS 2
- Linux, Windows, Mac C
Where did we start?

- Initial design discussions
- William prepared a minimal skeleton application
- Old code base: 40k loc
- Almost no tests, some manual tests with unknown expectations
Migration steps

- Extend initial skeleton
- Cross platform support
- Migrate each display
- Add tests for migrated/ new code
- Introduce end-to-end tests
ROS 1 vs ROS 2

From the RViz point of view

- No conceptual changes
- Node and tf2 API works more or less the same
- Tf2 in ROS 2 has less features
- Types have slightly different names
End-to-end tests?

RViz in a nutshell

Given a point cloud message
when I activate the point cloud display
then the point cloud is rendered according to the display properties

Conclusion

- 3D rendering must be included
- Assertion with an image
TEST_F(VisualTestFixture, pointcloud2_containing_one_big_point) {
    auto pointcloud2_publisher = 
        std::make_unique<VisualTestPublisher>(
            std::make_shared<nodes::PointCloud2Publisher>(), "pointcloud2_frame");

    setCamPose(Ogre::Vector3(0, 0, 16));
    setCamLookAt(Ogre::Vector3(0, 0, 0));

    auto pointcloud2_display = addDisplay<PointCloud2DisplayPageObject>();
    pointcloud2_display->setTopic("/pointcloud2");
    pointcloud2_display->setStyle("Spheres");
    pointcloud2_display->setSizeMeters(11);
    pointcloud2_display->setColor(0, 255, 0);

    assertMainWindowIdentity();
}
rviz_visual_testing_framework
rviz_visual_testing_framework
Is the migration complete?

- Most features are migrated
- Documentation for plugins and migration

What is missing?

- A few trivial displays (temperature, ...)
- Depth cloud display (requires image_transport)
- Interact tool/ interactive markers
- Stereo (Oculus display)
Thank you!

Contact: andreas.holzner@tngtech.com
Repository: github.com/ros2/rviz