The TurtleShell Naming Scheme

Sweeney et al.

2016-02-16

1 The Issue & Background

Today (2016-02-16) I was told by Ginger (Alex Ferguson) at lunch that I needed to completely redo the naming scheme for the codebase and repository. It is currently known as "turtleshell", and has been developed under the naming convention where the prefix "Turtle" indicates the presence of the class in the turtleshell library, to help in differentiating them from WPILib classes, and where the prefix "Turtwig" indicates that the code is intended to be used for our team, and unlikely to be portable to other systems.

2 Technical Reasons

The use of prefixes is well-established best practice. These prefixes allow for code to be easily distinguished as to its origin in purpose, the use of a class with a "Turtle" prefix is implied to not only be a component of the turtleshell library, but also indicates that it is likely an interface rather than an implementation. Similarly, the "Turtwig" prefix identifies code that this is specific to our robot and that is unlikely to be portable without significant modifications, and furthermore is an implementation of the same class with the "Turtle" prefix. Finally, the terms turtle and Turtwig imply the same relation, with Turtwig extending the idea of a turtle and building off of it.

3 Branding & Professionalism

The main allegation against the use of turtle-based naming systems was that it was "unprofessional". This was repeated despite numerous examples from professional software development where similar naming schemes were used. For example, the startup Flycast named two of their webservers "Godzilla" and "Bambi". Fanciful names are the norm in software development, versions of Android are different desserts and the project that won the 2015 Innovation in Control Award, FRC's highest reward for programming, was named the Common Chicken Runtime Environment.

The closest to a valid criticism that has been brought up is a lack of relation to our team.