Carnegie Mellon University

16-682

MRSD Project II

Task 03 Progress Review 9
Team C - COBORG

Jonathan Lord-Fonda
Teammates: Husam Wadi, Feng Xiang, Yuqing Qin, Gerry D’ascoli

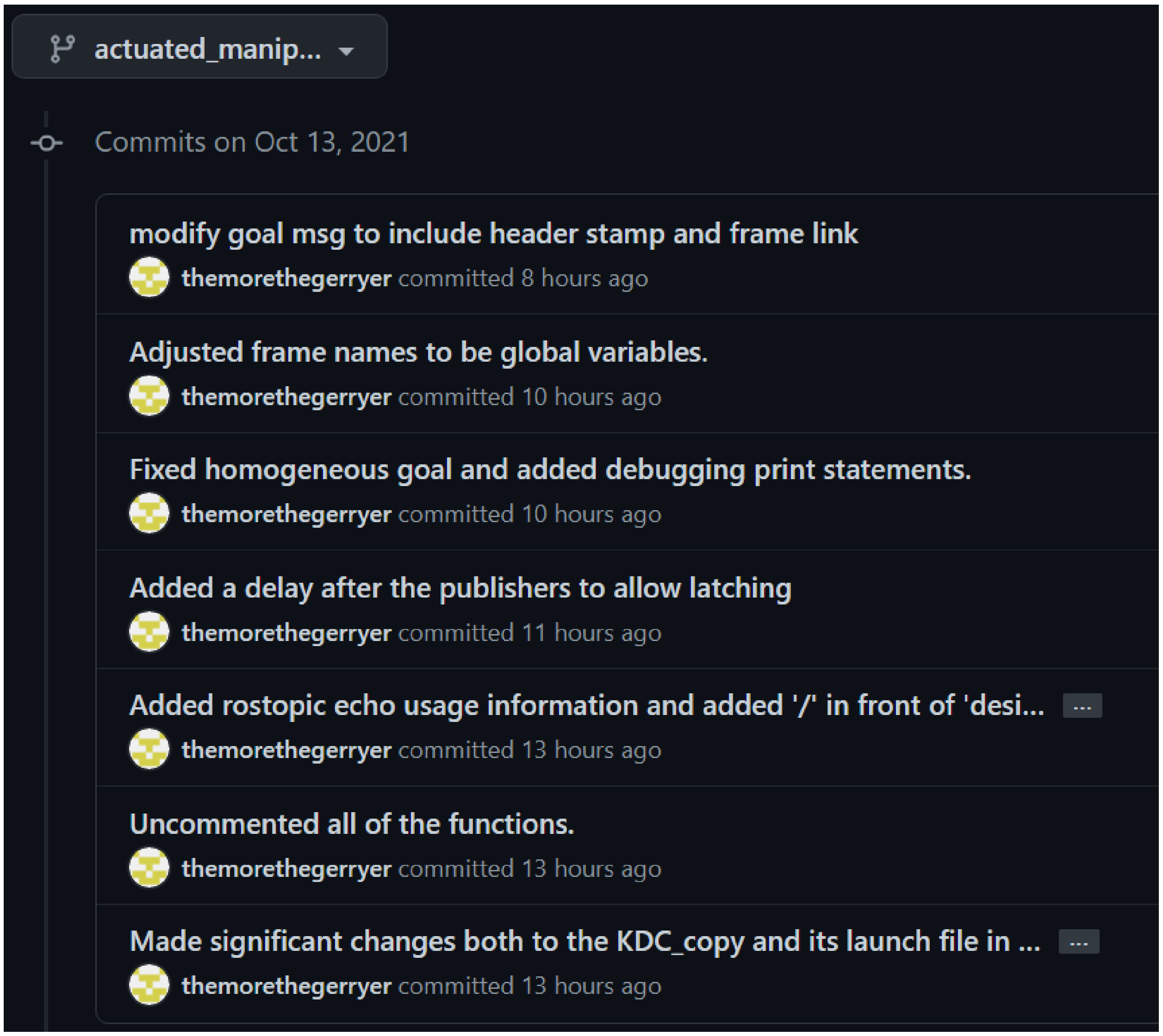
October 14, 2021

Table of Contents

1. Individual Progress

My primary tasks for this progress review were centered on debugging the smart manipulation software node. In order to achieve this I needed to start by setting up my own branch for the task on Github, internalizing the Github commands that I've been putting off learning, and creating my workflow. I developed my workflow from Oliver Steele's blog post about Git (seen here: https://blog.osteele.com/2008/05/my-git-workflow/). Simply, I use "git add ." followed by "git commit -m "Your message here." and "git push" whenever I make a small change that compiles and runs as expected. Creating a granular version history allows me to revert changes when I do something incorrectly and accurately describe the adjustments made in a short comment. It also helps me focus on completing individual aspects of my code instead of getting side-tracked onto different tasks before finishing the one I had previously started. Figure 1 below shows a snippet of the commits that I made yesterday on the day of the progress review.

Figure 1 - Git Rhythm



actuated_manip...

Commits on Oct 13, 2021

modify goal msg to include header stamp and frame link
themorethegerryer committed 8 hours ago

Adjusted frame names to be global variables.
themorethegerryer committed 10 hours ago

Fixed homogeneous goal and added debugging print statements.
themorethegerryer committed 10 hours ago

Added a delay after the publishers to allow latching
themorethegerryer committed 11 hours ago

Added rostopic echo usage information and added '/' in front of 'desi... ...
themorethegerryer committed 13 hours ago

Uncommented all of the functions.
themorethegerryer committed 13 hours ago

Made significant changes both to the KDC_copy and its launch file in ... ...
themorethegerryer committed 13 hours ago

This figure shows a sample of commits I made on the same day as the progress review. The comments clearly express what was changed so that it is unnecessary to scroll through, analyzing the code, when trying to look for a specific spot where things went awry. All of the commits are listed under Gerry D'ascoli's username because we tend to work on a shared pc and it isn't worth logging users in and out of the terminal whenever you want to make commits at the computer. Our project is small enough that only one or two people tend to work on a given branch.

Creating my Github workflow allowed me to work efficiently and safely without interfering with my teammates, which enabled me to progress on debugging the smart manipulation node. I wrote most of the code for this last semester, but quit working on it after it successfully compiled, followed by immediately erroring out. This meant that a variety of the most difficult bugs were left in the program and saw me adjusting such things as adding options to the launch file, writing down usage instructions as I figured out how to boot up the system as a whole, and fixing the problems left over from last semester. One of the most frustrating problems I ran into was my C++ program trying to create a ROS node before ROS was initialized in the program. In order to resolve this issue I commented out every callback and global variable and created a local copy of the global variables in the main loop, after ROS initialization. Then I compiled and ran the program while, line by line, uncommenting the global variables. When I found the variables responsible for the error I changed them to pointers and created references in the main body of the code after initialization. Many errors like this cropped up, particularly ROS-related issues that aren't captured on program compilation. Eventually the entire program ran smoothly and I began running it independently while using "rostopic pub" to simulate the other nodes that would be interacting with it. This process has taught me how important it is to unit test your code as you go and to build it in such a way that it is modular so that individual parts can be fully debugged without requiring the entire node to be written and working.

Beyond these primary tasks I discussed project aspects with the other team members, coordinating my activities with them and utilizing their expertise to help push me through some difficult debugging challenges in minimal time.

2. Challenges

Debugging is difficult and dry. It takes a significant amount of work and can induce procrastination by being a large unknown. Debugging can seem like an insurmountable task because it is difficult to know beforehand how long it will take and how many bugs there are, thus disincentivizing work on the issue(s). I believe that the solution to this moving forward is to spend extra time working out the code architecture ahead of time and break it up into more manageable chunks that can be individually tested and debugged.

3. Teamwork

Jason's work during this previous cycle was primarily working with Gerry on implementing and testing both the resolved rate code and goal stabilization capabilities. Additionally he worked with Gerry on obstacle avoidance and calibrated the robot model. Beyond these activities he worked with Yuqing to upgrade the vision system, measuring and implementing the new positions of the Realsense cameras.

Gerry’s work during this previous cycle included finalizing the goal stabilization code using resolved rate alongside Jason. Additionally, he tested the object detection and avoidance system and tuned parameters for its performance.

Yuqing’s work during this previous cycle was primarily executing the vision system upgrade. Along with this she created a simple demo featuring two different YOLO networks, with one executing on each camera in the vision system. Additionally she worked with Jason to measure out the new positions and orientations.

Husam’s work during this previous cycle primarily included physical upgrades to the Coborg’s components. Husam cut and assembled the carbon fiber version of the arm and worked on creating a URDF model in Solidworks. Additionally he followed up on items our team has been waiting for to complete the hardware upgrades and began considering parallel objectives for working on the dynamixel setup.

4. Plans

Before the next progress review I will finish debugging the smart manipulation node and integrate it with the other nodes and smart manipulation features we are including in the final build. Beyond this I will be revalidating aspects of the system and tuning the smart manipulation so that it functions as expected.