AuE 893: Sp’21: Autonomy Science and Systems

Department of Automotive Engineering, Clemson University

Venkat Krovi, Utkarsha Chaudhari and Huzefa Kagalwala

Assignment 2

Learning outcomes:

1. Working with Git.
2. Creating a ROS workspace
3. Writing a python script for operating the turtle in turtlesim

TASK 1

Instructions:

First setup
If you’re using git on your PC for the first time, certain configurations steps need to be followed.
You need to associate code uploads to the git server with the username and email used to log
onto
the git server. This is how a remote server like GitHub is able to associate your commit/change
with your account, so setup git on your PC the following way:
Read more here .
You will need to generate an SSH Key and add it to your GitHub account. An SSH key is an
alternate
way to identify yourself that doesn't require you to enter you username and password every time
you
want to use Git.
Type `ssh-keygen -t rsa` in a terminal window and follow the instructions. Open the generated `.ssh/id_rsa.pub` files and copy the content. Login to your GitHub account and navigate to Settings>SSH Keys>Add SSH Key to paste the content.

Follow this link for better instructions.

First Repository

Go to GitHub. Create a new repository and call it *AuE893Spring21_YourName*
Click on the repository you created and look for the URL on the webpage. Copy the URL.

Now go to the git workspace in your desktop, and clone this newly created repository here. Cloning creates a local copy of this repository.

```
cd git_ws
git clone <Insert URL>
```

Enter that repo and initiate git.

```
cd <your repo>
git init
```

Create a file here called README.md

```
gedit README.md
```

Write a formal introduction about yourself and a description of what this repository will contain. Save and close.

Add and commit changes
The `commit` command lets Git take note of what files you're planning to upload and any messages you as a developer was to attach to the files you are committing. The add command is used to collect the files to be committed.

```
git add README.md
git commit -m "write a message about your files here"
```

The command `$ git add .` adds all changes in the repository in one shot. Use `$ git status` to check the adds and commits on your repo.
Now git needs to recognize your repository in your online server. This step is redundant right now because when you cloned the repository the URL of the repo was automatically attributed to the remote. You can see this when you type `git remote -v` . However, this may not always be the case. If nothing shows up, or the wrong URL shows up, type

```
git remote add origin <your repo's URL>
```

Now push the commits onto your online repository.

```
git push -u origin master
```

You should see the README file reflect in your repository. (Click on Browse)
You are almost done with the basics.

Workspace management
This class will be divided into groups of 5. You will be expected to submit the code for your mini projects and finals on GitHub on your personal repos. When you collaborate with your team members you can pull their code from their repositories and you may push your code to theirs.

All your assignment code submission is expected to be pushed to your GitHub repository. You may maintain a single catkin_ws folder and organize the code within according to the file submissions. Each folder within the /src folder corresponds to a ROS package. For example:

```
AuE893Spring19_YourName
  README.md
  /catkin_ws
    /build
    /devel
    /src
      /Assignment2_TurtleSim
          /launch
          /scripts
          /videos
          README.md
          CMakeLists.txt
          package.xml
      /Assignment3
          /launch
          /scripts
          /videos
          README.md
```

```
        CMakeLists.txt
        package.xml
    /Final_Project
        ...
```

The README files should contain instructions on how to run your code.

TASK 2

Create a new catkin workspace called "assignment2_ws".
In this assignment you will write two python codes in a "src" folder:

1. circle.py: Make the Turtle move in a circle with constant twist velocity
2. square_openloop.py: Make the Turtle move in a square of 2x2 units with 0.2 linear velocity and 0.2 rad/s angular velocity.
   Use this Move in a Straight Line tutorial as a reference.
3. square_closedloop.py: Make the Turtle move in a square of 3x3 with velocity control. Predefine
   the coordinate points of the square in the program. Make the turtle move from (5,5) ->(8,5) -> (8,8) -> (5,8) -> (5,5).

Use the above code and Go to Goal tutorial as a reference.

Create a README.md with a brief explanation of what each code is and include a screenshot (or GIF) of the trajectory in that file. Record your screen as you run each program and include the videos in your workspace folder under a folder titled 'videos'.
Push this to your GitHub repo.

III. Submission
You will submit the link to your created repository on Canvas.

Further reading: Version control using Git