

Working on the Robot Software

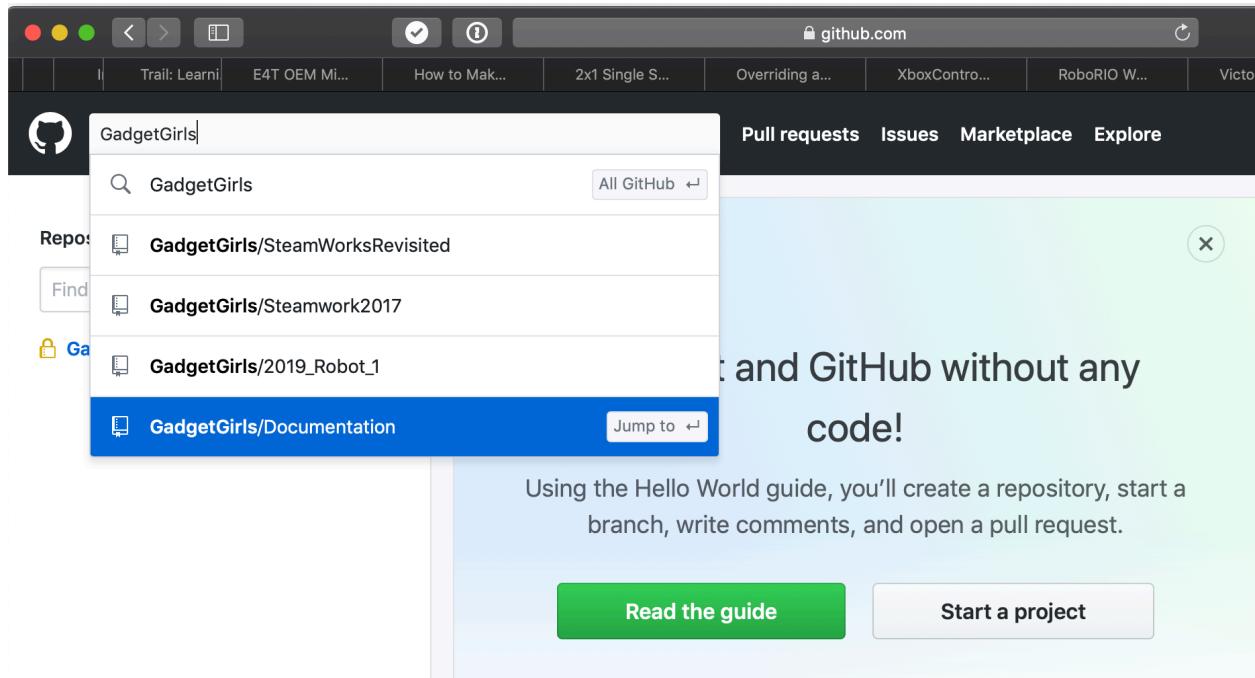
If you're working on the Team laptop, skip ahead to "Working on Software Changes".

Getting Set up with Git and Github

Go to <https://www.github.com/join> and follow the instructions to create a free personal Github account.

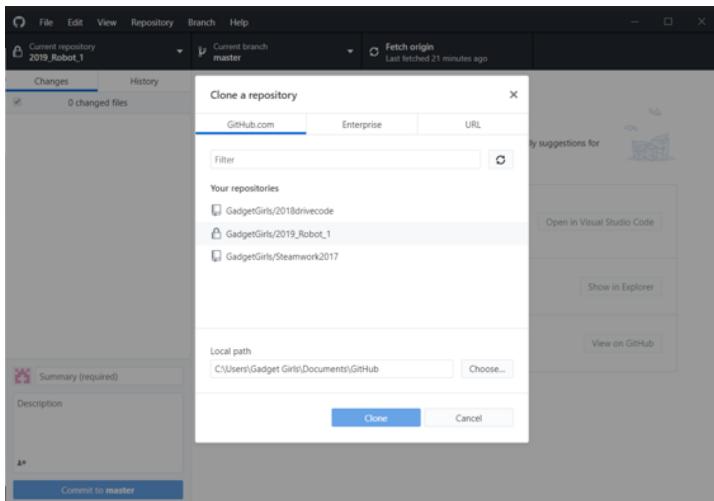
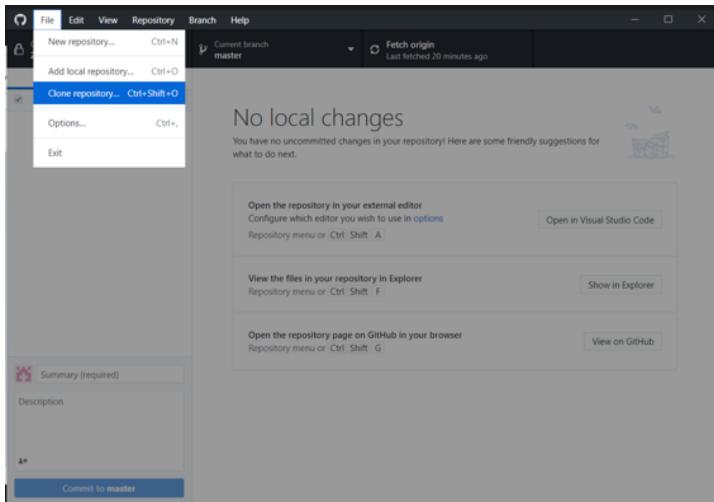
Download Github Desktop from <https://desktop.github.com>, install it, and log in with your github account above.

On the github website, find the GadgetGirls team repositories:



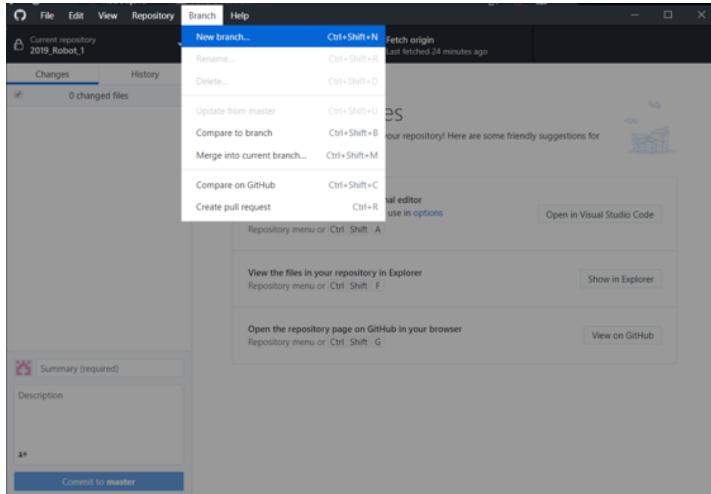
You may need to e-mail one of the mentors with your Github username to add you to the allowed contributors.

First time, only, clone the repository from Github using Github Desktop:

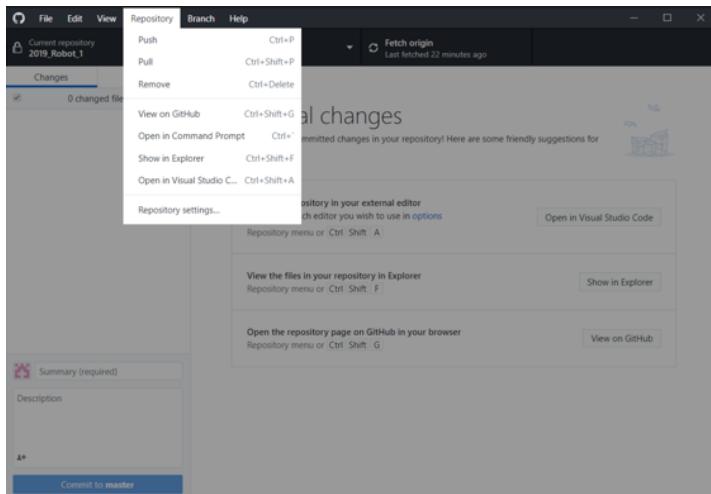


Working on Software Changes

Create a new branch off of 'master' for your work - think about each branch as a mini-project, like "Adding Camera Support". To help other folks, name your branch with your initials, like 'po_cam_support':



Next, open the branch in Visual Studio Code (VSCode):



Make your edits to the code, save the files, and stage the changes:

A screenshot of the Visual Studio Code interface. The title bar says "RobotJava - 2019_Robot_1 - Visual Studio Code". The left sidebar has icons for file operations like Open, Save, Find, and Publish. The main area shows the Java code for Robot.java. In the Source Control pane, under "STAGED CHANGES", there is one entry: "Thinking about teleop steps" with a status of "M" (modified). The code editor shows the following Java code:

```
66 /**
67 * This function is called periodically during autonomous.
68 */
69 @Override
70 public void autonomousPeriodic() {
71     // Drive for 2 seconds
72     if (m_timer.get() < 2.0) {
73         m_robotDrive.arcadeDrive(0.5, 0.0); // drive forwards half speed
74     } else {
75         m_robotDrive.stopMotor(); // stop robot
76     }
77 }
78
79 /**
80 * This function is called once each time the robot enters teleop.
81 */
82 @Override
83 public void teleopInit() {
84     // Thinking about teleop changes
85 }
86
87 /**
88 * This function is called periodically during teleoperated mode.
89 */
90 @Override
91 public void teleopPeriodic() {
92     // Our XboxController sends forward Y as a negative value, so flip
93     // it around.
94     m_robotDrive.arcadeDrive(-m_stick.getY(), m_stick.getX());
95 }
96
97 /**
98 * This function is called periodically during test mode.
99 */
100 @Override
101 public void testPeriodic() {
```

When you're done with a related batch of staged changes, it's time for a 'commit'. In a one-line description, tell what your changes are doing and then hit the check mark to commit:

A screenshot of the Visual Studio Code interface, similar to the previous one but with a commit message. The title bar says "RobotJava - 2019_Robot_1 - Visual Studio Code". The left sidebar has icons for file operations like Open, Save, Find, and Publish. The main area shows the Java code for Robot.java. In the Source Control pane, under "STAGED CHANGES", there is one entry: "Thinking about teleop steps" with a status of "M" (modified). Under "CHANGES", there is one entry: "Thinking about teleop steps" with a status of "C" (committed). The code editor shows the same Java code as the first screenshot.

When it's time to back up your changes, go to the Source Control pane, click '...' and select 'Publish branch':

The screenshot shows the Visual Studio Code interface with the following details:

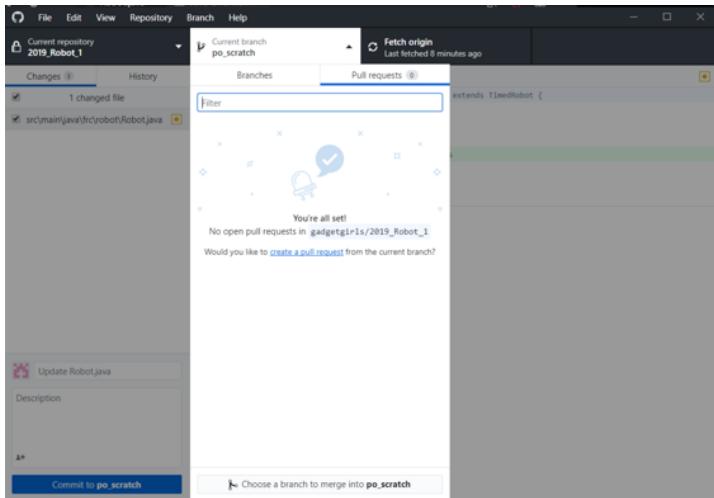
- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Robot.java - 2019_Robot_1 - Visual Studio Code.
- Source Control Sidebar:** Shows "Thinking about teleop steps" and a "STAGED CHANGES" section for Robot.java.
- Context Menu (Push Selected):** The "Push" option is highlighted in blue. Other options include Pull, Pull (Rebase), Pull from..., Push, Push to..., Sync, Publish Branch, Commit All, Commit All (Amend), Commit All (Signed Off), Commit Staged, Commit Staged (Amend), Commit Staged (Signed Off), Undo Last Commit, Discard All Changes, Stage All Changes, Unstage All Changes, Apply Latest Stash, Apply Stash..., Pop Latest Stash, Pop Stash..., Stash, Stash (Include Untracked), and Show Git Output.
- Code Editor:** Displays Java code for Robot.java, specifically the `testPeriodic()` method.
- Bottom Status Bar:** Lines 101-102, Col 37, Spaces: 2, UTF-8, CRLF, Java, WPIlib, 1.

And then 'Push' changes:

The screenshot shows the Visual Studio Code interface with the following details:

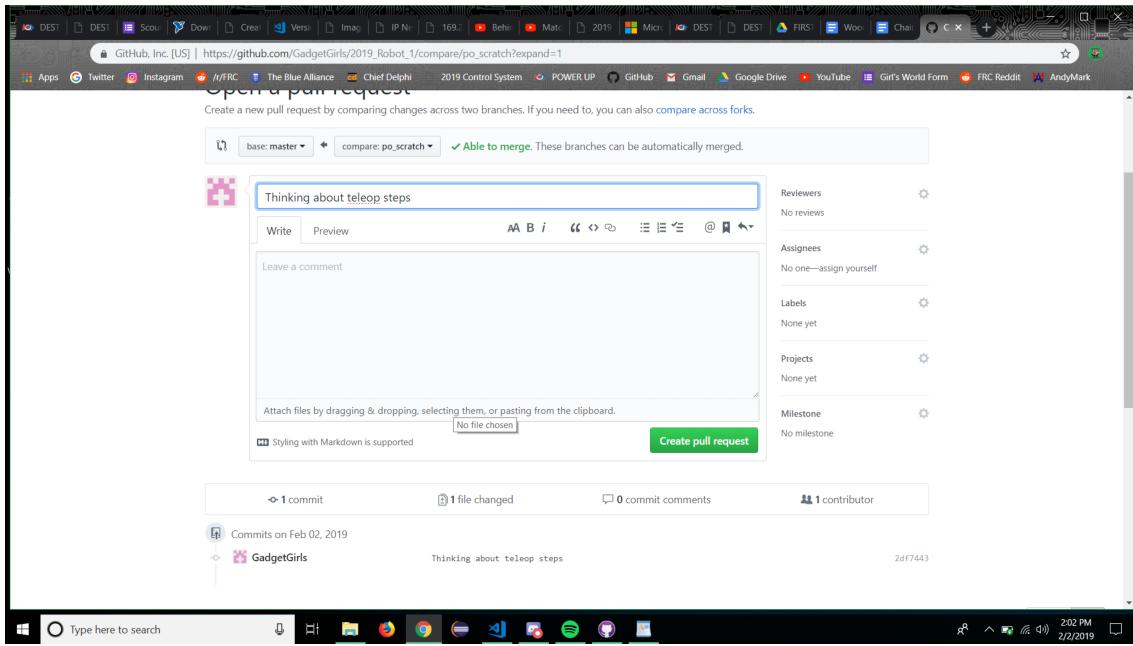
- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Robot.java - 2019_Robot_1 - Visual Studio Code.
- Source Control Sidebar:** Shows "Thinking about teleop steps" and a "STAGED CHANGES" section for Robot.java.
- Context Menu (Push Selected):** The "Push" option is highlighted in blue. Other options include Pull, Pull (Rebase), Pull from..., Push, Push to..., Sync, Publish Branch, Commit All, Commit All (Amend), Commit All (Signed Off), Commit Staged, Commit Staged (Amend), Commit Staged (Signed Off), Undo Last Commit, Discard All Changes, Stage All Changes, Unstage All Changes, Apply Latest Stash, Apply Stash..., Pop Latest Stash, Pop Stash..., Stash, Stash (Include Untracked), and Show Git Output.
- Code Editor:** Displays Java code for Robot.java, specifically the `testPeriodic()` method.
- Bottom Status Bar:** Lines 101-102, Col 37, Spaces: 2, UTF-8, CRLF, Java, WPIlib, 1.

When your mini-project is complete and tested, it's time to merge it into the main code using a Compare & Pull Request in Github Desktop (or the Github website):



The screenshot shows a web browser displaying the GitHub repository page for 'GadgetGirls / 2019_Robot_1'. The repository is private. It shows 1 commit, 2 branches, and 0 releases. The 'po_scratch' branch is highlighted. The commit list includes:

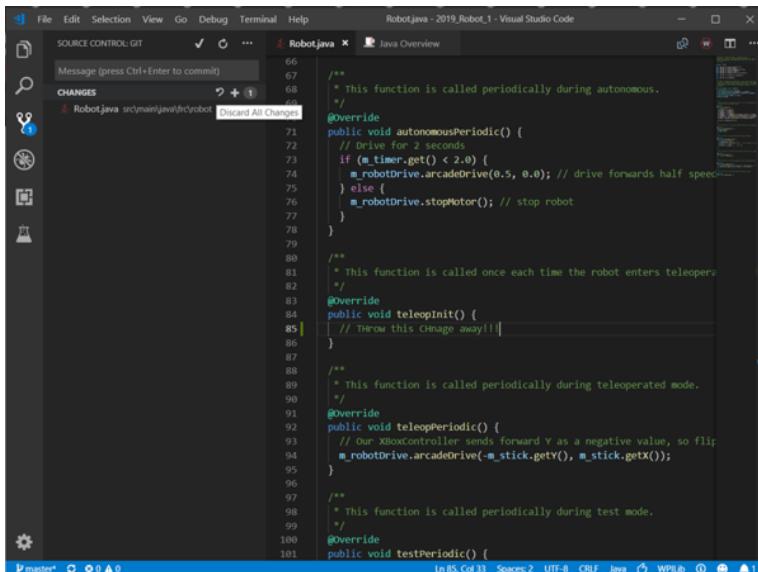
File	Checkin - PO	Last Commit
.vscode	Initial checkin	an hour ago
.wpilib	Initial checkin	an hour ago
gradle/wrapper	Initial checkin	an hour ago
src/main	Initial checkin	an hour ago
.gitignore	Initial checkin	an hour ago
build.gradle	Initial checkin	an hour ago
gradlew	Initial checkin	an hour ago
gradlew.bat	Initial checkin	an hour ago



Now tell one of the code gatekeepers (probably your Programming Mentor) to review and approve the changes! When you're both done, you can Merge the Pull request and delete your branch - it will now be part of the official 'master' branch.

What to do when things go wrong

To discard saved, unstaged changes that haven't been committed yet, just go to the Source Control pane and select the Discard All Changes back arrow:



To unstage a staged change, go to the Source Control pane, STAGED and then the '-' icon:

```
File Edit Selection View Go Debug Terminal Help
SOURCE CONTROL: GIT
Robot.java - 2019.Robot_1 - Visual Studio Code
Message (press Ctrl+Enter to commit)
STAGED CHANGES
RobotJava src/main/java/frcrobot
CHANGES
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
@Override
public void autonomousPeriodic() {
    // Drive for 2 seconds
    if (m_timer.get() < 2.0) {
        m_robotDrive.arcadeDrive(0.5, 0.0); // drive forwards half speed
    } else {
        m_robotDrive.stopMotor(); // stop robot
    }
}

@Override
public void teleopInit() {
    // Throw this Change away!!!
}

@Override
public void teleopPeriodic() {
    // Our XboxController sends forward Y as a negative value, so flip
    m_robotDrive.arcadeDrive(-m_stick.getY(), m_stick.getX());
}

@Override
public void testPeriodic() {
```

To "uncommit" changes before pushing them, go to the Source Control pane, '...', then 'Undo Last Commit':

```
File Edit Selection View Go Debug Terminal Help
SOURCE CONTROL: GIT
Robot.java - 2019.Robot_1 - Visual Studio Code
Message (press Ctrl+Enter to commit)
CHANGES
pull
pull (Rebase)
pull from...
push
push to...
sync
publish branch
Commit All
Commit All (Amend)
Commit All (Signed Off)
Commit Staged
Commit Staged (Amend)
Commit Staged (Signed Off)
Undo Last Commit
Discard All Changes
Stage All Changes
Unstage All Changes
Apply Latest Stash
Apply Stash...
Pop Latest Stash
Pop Stash...
Stash
Stash (Include Untracked)
Show Git Output
@Override
public void autonomousPeriodic() {
    // Drive for 2 seconds
    if (m_timer.get() < 2.0) {
        m_robotDrive.arcadeDrive(0.5, 0.0); // drive forwards half speed
    } else {
        m_robotDrive.stopMotor(); // stop robot
    }
}

@Override
public void teleopInit() {
    // Throw this Change away!!!
}

@Override
public void teleopPeriodic() {
    // Our XboxController sends forward Y as a negative value, so flip
    m_robotDrive.arcadeDrive(-m_stick.getY(), m_stick.getX());
}

@Override
public void testPeriodic() {
```