In this project you'll revisit the lake race track from the Behavioral Cloning Project. This time, however, you'll implement a PID controller in C++ to maneuver the vehicle around the track!

The simulator will provide you the cross track error (CTE) and the velocity (mph) in order to compute the appropriate steering angle.

One more thing. The speed limit has been increased from 30 mph to 100 mph. Get ready to channel your inner Vin Diesel and try to drive **SAFELY** as fast as possible! **NOTE: you don't have to meet a minimum speed to pass.**

**What You'll Receive**

1. A [GitHub repo with starter code](https://github.com/udacity/CarND-Controls-PID) that you can fork.
2. A simulator, download from the [releases](https://github.com/udacity/self-driving-car-sim/releases) page of the project repo. You should download the most recent version.

(**Note:** If you choose to utilize the Workspace for this project, both the simulator and related project repository are included therein, so you won't need to download/clone them locally except as desired).

**Using GitHub and Creating Effective READMEs**

If you are unfamiliar with GitHub , Udacity has a brief [GitHub tutorial](http://blog.udacity.com/2015/06/a-beginners-git-github-tutorial.html) to get you started. Udacity also provides a more detailed free [course on git and GitHub](https://www.udacity.com/course/how-to-use-git-and-github--ud775).

To learn about REAMDE files and Markdown, Udacity provides a free [course on READMEs](https://www.udacity.com/course/writing-readmes--ud777), as well.

GitHub also provides a [tutorial](https://guides.github.com/features/mastering-markdown/) about creating Markdown files.

# Project Instructions

1. You can solve this project using the **workspaces** we have prepared for you [here](https://classroom.udacity.com/nanodegrees/nd013/parts/30260907-68c1-4f24-b793-89c0c2a0ad32/modules/3e2b027e-1822-4638-a58a-def1e64258a8/lessons/b6e7a810-49ec-48eb-8ac6-bab376cb94fd/concepts/6d1cbc73-9e87-4f33-adb2-0a12950a0221?contentVersion=2.0.0&contentLocale=en-us) or you can clone/fork the project's template files from the [project repository](https://github.com/udacity/CarND-Controls-PID) if you want to install all on your local machine. Have a look at the [rubric](https://review.udacity.com/#!/rubrics/1972/view) as well!
2. Build a PID controller and tune the PID hyperparameters by applying the general processing flow as described in the previous lessons.
3. Test your solution on the simulator!
4. When the vehicle is able to drive successfully around the track, submit! Remember to include a file addressing the reflection portion of the project in .txt, .md, .html, or .pdf format.
5. Try to see how fast you get the vehicle to **SAFELY** go!

## Accessing and using the workspace:

* Navigate to the workspace node. If you need to test your solution, enable the GPU - otherwise, you won't be able to run the simulator. Remember to turn it off when you code your solution so that you don't run out of GPU hours.
* Navigate to the repository CarND-PID-Control-Project using menu on the left.
* Complete the TODO in main.cpp, PID.cpp and add any helper files you need using the text editor in the workspace.
* Navigate to the project repository in the terminal.

**Note**: You can add any necessary helper files within the src (or other) folder by clicking on the + in the top left, selecting Create New File, and then naming your file with the relevant extension (such as .h, .hpp, .cpp, etc).

In order to build and compile the code, from within the main repository directory, you can:

* mkdir build && cd build to create and enter the build directory
* cmake .. && make to compile your project
* ./pid to run your code

Click on the "Simulator" button in the bottom of the Udacity workspace, which will open a new virtual desktop. You should see a "Simulator" icon on the virtual desktop. Double-click the "Simulator" icon in that desktop to start the simulator.

**Important:** You need to open a terminal before attempting to run the simulator.

All Project instructions can be found in the [README.md](https://github.com/udacity/CarND-Path-Planning-Project/blob/master/README.md) (you can view the instruction in an easy-to-read format by visiting the previous link.

## Commit to GitHub

Students are highly encouraged to commit their project to a GitHub repo. To do this, you must change the upstream of the current repository and add your credentials. We have supplied a bash script to help you do this. Please open up a terminal, navigate to the project repository, and enter: ./set\_git.sh, then follow the prompts. This will set the upstream remote to your own repository and add your email and username to the git configuration. At this time we are not configuring passwords, so you will need to enter your username and password for each push. Since credentials are not persistent, it will be necessary to run this script each time you open, refresh, or reset the workspace.

### Ready to Submit?

When you are ready to submit, just make sure you have in your files:

1. Write up
2. A complete solution, with all the files needed to run your project.

And double-check your work against the [rubric](https://review.udacity.com/#!/rubrics/1972/view).

If you use **workspaces** click on the button on the right bottom and follow the instruction. Alternatively, you may submit a link to your GitHub repo for the project or a zip file with your files in the last item in this lesson.

### Using GitHub and Creating Effective READMEs

If you are unfamiliar with GitHub , Udacity has a brief [GitHub tutorial](http://blog.udacity.com/2015/06/a-beginners-git-github-tutorial.html) to get you started. Udacity also provides a more detailed free [course on git and GitHub](https://www.udacity.com/course/how-to-use-git-and-github--ud775).

To learn about README files and Markdown, Udacity provides a free [course on READMEs](https://www.udacity.com/course/writing-readmes--ud777), as well.

GitHub also provides a [tutorial](https://guides.github.com/features/mastering-markdown/) about creating Markdown files.

## Project Support

If you are stuck or having difficulties with the project, don't lose hope! Remember to talk to your mentors and fellow students in your Study Group, as well as ask (and answer!) questions on [Knowledge](https://knowledge.udacity.com/) tagged with the project name. We also have a previously recorded project Q&A that you can watch [here](https://youtu.be/YamBuzDjrs8)!