Thank you for considering contributing to Procedural Toolkit! This document outlines the various ways in which you can help.

## How can I contribute?

### Reporting bugs and issues

First, please search the [open issues](https://github.com/Syomus/ProceduralToolkit/issues?q=is%3Aopen)

and [closed issues](https://github.com/Syomus/ProceduralToolkit/issues?q=is%3Aclosed)

to see if your issue hasn't already been reported. If it does exist, add a :thumbsup: to the issue to

indicate this is also an issue for you, and add a comment to the existing issue if there is extra information you can contribute.

If you can't find a matching issue, open a [new issue](https://github.com/Syomus/ProceduralToolkit/issues/new/choose),

choose the right template and provide us with enough information to investigate further.

Alternatively, you can send a message to the [support email](mailto:proceduraltoolkit@syomus.com).

### Suggesting enhancements

If you have an idea on how to improve Procedural Toolkit, don't hesitate to

open a [new issue](https://github.com/Syomus/ProceduralToolkit/issues/new/choose) and describe in details what you have in mind.

### Proofreading

If you find a typo or bad writing, let us know and open a [new issue](https://github.com/Syomus/ProceduralToolkit/issues/new/choose)

or, even better, send a [pull request](https://github.com/Syomus/ProceduralToolkit/pulls).

### Help wanted

If you're looking for something to work on, check out the

[help wanted](https://github.com/Syomus/ProceduralToolkit/issues?q=is%3Aissue+is%3Aopen+label%3A"help+wanted") label.

If you see something you like, please comment on the issue to let the development team know you are interested in the issue.

### Testing

Tests are always appreciated. If you see a feature that doesn't have sufficient test coverage

(especially if it is [Geometry](/Scripts/Geometry)-related), we would be very grateful if you wrote some tests for it!

Tests are written using the [Unity Test Runner](https://docs.unity3d.com/Manual/PlaymodeTestFramework.html).

### Optimization

If you notice some bottleneck or inefficiency, feel free to share it with us by opening a [new issue](https://github.com/Syomus/ProceduralToolkit/issues/new/choose)

or sending a [pull request](https://github.com/Syomus/ProceduralToolkit/pulls).

## Pull requests

Pull requests are welcome. If you are interested in contributing, you will need to clone or download the [development project](https://github.com/Syomus/ProceduralToolkit.UnityProject)

and install the toolkit from your [fork](https://help.github.com/articles/working-with-forks/) into `Unity\_ProceduralToolkit/Packages`.

And that's it! After setting up the development project you can start making changes and committing them to your fork.

\*\*Before you commit:\*\*

\* Save all your changes.

\* Check the console window to make sure that there are no errors or warnings.

\* If you worked on something covered by tests, do not forget to run them too.

For details on creating pull requests from a fork please refer to the official [documentation](https://help.github.com/articles/creating-a-pull-request-from-a-fork/).

By submitting a pull request, you agree that your contribution will be licensed under the [MIT license](https://github.com/Syomus/ProceduralToolkit/blob/master/LICENSE.md) for this project.