# Setup

1. Download and install [Visual Studio Community 2017](https://www.visualstudio.com/downloads/). During install, make sure ".NET desktop development" is checked.

2. Download and install [Git](https://git-scm.com/download). During install, change "Use Git from Git Bash only" to one of the other two options, depending on your preference.

3. Clone the repository by opening command prompt/powershell in the folder of your choice and typing `git clone https://github.com/HearthSim/Hearthstone-Deck-Tracker`.

4. Download the recommended latest [nuget executable](https://dist.nuget.org/index.html) and copy it into your PATH, or where you cloned the git repository (where bootstrap.ps1 is).

5. run bootstrap.ps1, preferably from a powershell window to see any errors.

6. Open the solution file with VS17 and do a build to make sure everything is working properly.

# Creating Issues

- Have a look at the [FAQ](https://github.com/HearthSim/Hearthstone-Deck-Tracker/wiki/FAQ).

- Check if a similar issue already exists (use the search function).

- In case of a bug/crash/problem: Add as much detail as possible, including crash reports/logs ([see here](https://github.com/HearthSim/Hearthstone-Deck-Tracker/wiki/Reporting-Issues)).

# Contributing

## Contributor License Agreement

HearthSim requires a signed CLA from any contributor before his/her PR can be merged.

### What's a CLA?

> A CLA is a legal document in which you state you are entitled to contribute

the code/documentation/translation to the project you’re contributing to and are

willing to have it used in distributions and derivative works. This means that

should there be any kind of legal issue in the future as to the origins and

ownership of any particular piece of code, then that project has the necessary

forms on file from the contributor(s) saying they were permitted to make this

contribution.

> The CLA also ensures that once you have provided a contribution, you cannot try

to withdraw permission for its use at a later date. People and companies can

therefore use that software, confident that they will not be asked to stop using

pieces of the code at a later date.

- [Tony Guntharp (CLAHub)](http://fusion94.org/2013-01-16-clahub-clas-done-right/)

### Why?

HearthSim uses a Contributor License Agreement to ensure we can legally use,

redistribute and relicense external contributions to our various projects.

### How do I sign it?

Contact any admin on our [Developer Discord](https://discord.gg/hearthsim-devs) or send us an email at contact@hearthsim.net. Please include your Github username.

## Project approval

For trivial changes or fixes (i.e. a couple lines of code) on new or existing issues and projects feel free to just open a PR.

We would like to avoid wasting your time, so for non-trivial changes or fixes please do one of the following before starting to work on it:

- For work on an existing issue please comment on that issue and wait for approval.

- If no issue exists, please open a new issue with the proposal and wait for approval.

- Propose the project in #hdt on the [Developer Discord](https://discord.gg/hearthsim-devs) and wait for approval.

## Coding style

1. Always use tabs.

2. Use LF line endings.

3. Always place braces on new lines.

4. Use [C# 6](https://github.com/dotnet/roslyn/wiki/New-Language-Features-in-C%23-6)/[C# 7](https://blogs.msdn.microsoft.com/dotnet/2016/08/24/whats-new-in-csharp-7-0/) syntax whenever possible.

5. Follow the [standard MS C# naming conventions](https://msdn.microsoft.com/en-us/library/ms229002(v=vs.110).aspx)

([short version](http://programmers.stackexchange.com/a/224910)).

Also see: [How to name things in programming](http://www.slideshare.net/pirhilton/how-to-name-things-the-hardest-problem-in-programming)

6. Know when to make exceptions.

## Commits and Pull Requests

Keep the commit log as healthy as the code. It is one of the first places new contributors will look at the project.

1. No more than one change per commit. There should be no changes in a commit which are unrelated to its message.

2. Follow [these conventions](http://chris.beams.io/posts/git-commit/) when writing the commit message.

3. Keep the diffs as clean as possible. Avoid unnecessary line changes.

When filing a Pull Request, make sure it is rebased on top of most recent master.

If you need to modify it or amend it in some way, you should always appropriately

[fixup](https://help.github.com/articles/about-git-rebase/) the issues in git and force-push your changes to your fork.

Also see: [Github Help: Using Pull Requests](https://help.github.com/articles/using-pull-requests/)

## Translations

Want to help translate the app? Take a look at the [HDT-Localization repository](https://github.com/HearthSim/HDT-Localization).

## Need help?

You can always ask for help on [Gitter](https://gitter.im/HearthSim/Hearthstone-Deck-Tracker), #hdt in the [Developer Discord](https://discord.gg/hearthsim-devs), or the HearthSim IRC channel, `#Hearthsim` on [Freenode](https://freenode.net/).