# Contributing to cocos2d-x

## For general questions

You can ask general questions by using:

- Forum (preferred way): http://discuss.cocos2d-x.org/

- IRC: https://webchat.freenode.net/ (Use the \*cocos2d\* or \*cocos2d-x\* channels)

- Weibo: http://t.sina.com.cn/cocos2dx

- Twitter: http://www.twitter.com/cocos2dx

## Reporting bugs

To report bugs, please use the [Issue Tracker](https://github.com/cocos2d/cocos2d-x/issues)

Steps to report a bug:

\* Open the [url](https://github.com/cocos2d/cocos2d-x/issues/new)

\* Add all the needed information to reproduce the bug, the information include

\* engine version

\* steps to reproduce the bug

\* some pseudocode

\* resources link if needed

## Submitting patches

If you want to contribute code, please follow these steps:

(If you are new to git and/or GitHub, you should read [Pro Git](http://progit.org/book/) , especially the section on [Contributing to a project:Small/Large Public Project](http://progit.org/book/ch5-2.html#public\_small\_project) )

- Download the latest cocos2d-x develop branch from github:

```

$ git clone git://github.com/cocos2d/cocos2d-x.git

$ cd cocos2d-x

$ git checkout v3

$ ./download-deps.py

$ git submodule update --init

```

- Apply your changes in the recently downloaded repository

- Commit your changes in your own repository

- Create a new branch with your patch: `$ git checkout -b my\_fix\_branch`

- Push your new branch to your public repository

- Send a “pull request” to user “cocos2d”

- It must be \_complete\_. See the definition below

- It must follow the \_Releases\_ rules. See the definition below

## Only \_complete\_ patches will be merged

The patch must be \_complete\_. And by that, we mean:

- For C++ code follow the [Cocos2d C++ Coding Style][1]

- For Python code follow the [PEP8 guidelines][3]

- Describe what the patch does

- Include test cases if applicable

- Include unit tests if applicable

- Must be tested in all supported platforms [\*]

- Must NOT degrade the performance

- Must NOT break existing tests cases

- Must NOT break the Continuous Integration build

- Must NOT break backward compatibility

- Must compile WITHOUT warnings

- New APIs MUST be \*\*easy to use\*\*, \*\*familiar\*\* to cocos2d-x users

- Code MUST be \*\*easy to extend\*\* and \*\*maintain\*\*

- Must have documentation: C++ APIs must use Doxygen strings, tools must have a README.md file that describe how to use the tool

- Must be efficient (fast / low memory needs)

- It must not duplicate existing code, unless the new code deprecates the old one

- Patches that refactor key components will only be merged in the next major versions.

[\*]: If you don't have access to test your code in all the supported platforms, let us know.

\_\_TBD\_\_: Is this applicable for big features ? What is the best way to merge big features ?

# Promoting cocos2d

Help us promote cocos2d-x by using the cocos2d logo in your game, or by mentioning cocos2d in the credits.

[Logo Resources of Cocos2d-x][2]

[1]: https://github.com/cocos2d/cocos2d-x/blob/v3/docs/CODING\_STYLE.md

[2]: http://www.cocos2d-x.org/wiki/Logo\_Resources\_of\_Cocos2d-x

[3]: https://www.python.org/dev/peps/pep-0008