## Contributing to raylib

Hello contributors! Welcome to raylib!

Do you enjoy raylib and want to contribute? Nice! You can help with the following points:

- C programming - Can you write/review/test/improve the code?

- Documentation/Tutorials/Example - Can you write some tutorial/example?

- Porting to other platforms - Can you port and compile raylib on other systems?

- Web Development - Can you help [with the website](https://github.com/raysan5/raylib.com)?

- Testing - Can you find some bugs in raylib?

This document contains a set of guidelines to contribute to the project. These are mostly guidelines, not rules.

Use your best judgement, and feel free to propose changes to this document in a pull request.

### raylib philosophy

- raylib is a tool to enjoy videogames programming, every single function in raylib should be a tutorial on itself.

- raylib is \*\*SIMPLE\*\* and \*\*EASY-TO-USE\*\*, I tried to keep it compact with a small set of functions, if a function is too complex or is not clearly useful, better not including it.

- raylib is open source and free; educators and institutions can use this tool to \*\*TEACH\*\* videogames programming completely for free.

- raylib is collaborative; contribution of tutorials / code examples / bug fixes / code comments are highly appreciated.

- raylib's license (and its external libs respective licenses) allow using it on commercial projects.

### Some interesting reads to start with

- [raylib history](HISTORY.md)

- [raylib architecture](https://github.com/raysan5/raylib/wiki/raylib-architecture)

- [raylib license](LICENSE)

- [raylib roadmap](ROADMAP.md)

[raylib Wiki](https://github.com/raysan5/raylib/wiki) contains some information about the library and is open to anyone for edit.

Feel free to review it if required, just take care not to break something.

### raylib C coding conventions

Despite being written in C, raylib does not follow the standard Hungarian notation for C,

it [follows Pascal-case/camel-case notation](https://github.com/raysan5/raylib/wiki/raylib-coding-conventions),

more common on C# language. All code formatting decisions have been carefully taken

to make it easier for students/users to read, write and understand code.

Source code is extensively commented for that purpose, raylib primary learning method is:

> learn by reading code and examples

For detailed information on building raylib and examples, please see [raylib Wiki](https://github.com/raysan5/raylib/wiki).

### Opening new Issues

To open new issue for raylib (bug, enhancement, discussion...), just try to follow these rules:

- Make sure the issue has not already been reported before by searching on GitHub under Issues.

- If you're unable to find an open issue addressing the problem, open a new one. Be sure to include a

title and clear description, as much relevant information as possible, and a code sample demonstrating the unexpected behavior.

- If applies, attach some screenshot of the issue and a .zip file with the code sample and required resources.

- On issue description, add a brackets tag about the raylib module that relates to this issue.

If don't know which module, just report the issue, I will review it.

- You can check other issues to see how is being done!

### Sending a Pull-Request

- Make sure the PR description clearly describes the problem and solution. Include the relevant issue number if applicable.

- Don't send big pull requests (lots of changelists), they are difficult to review. It's better to send small pull requests, one at a time.

- Verify that changes don't break the build (at least on Windows platform). As many platforms where you can test it, the better, but don't worry

if you cannot test all the platforms.

### Contact information

If you have any doubt, don't hesitate to [contact me](mailto:ray@raylib.com)!.

You can write me a direct mail but you can also contact me on the following networks:

- [raylib reddit](https://www.reddit.com/r/raylib/) - A good place for discussions or to ask for help.

- [raylib Discord](https://discord.gg/VkzNHUE) - A direct communication channel for project discussions.

- [raylib twitter](https://twitter.com/raysan5) - My personal twitter account, I usually post about raylib, you can send me PMs.

- [raylib web](http://www.raylib.com/) - On top-right corner there is a bunch of networks where you can find me.

Thank you very much for your time! :)

----

Here is a list of raylib contributors, these people have invested part of their time

contributing (in some way or another) to make the raylib project better. Huge thanks to all of them!

- [Zopokx](https://github.com/Zopokx) for testing the web.

- [Elendow](http://www.elendow.com) for testing and helping on web development.

- Victor Dual for implementing and testing 3D shapes functions.

- Marc Palau for implementing and testing 3D shapes functions and contribute on camera and gestures modules.

- Kevin Gato for improving texture internal formats support and helping on raygui development.

- Daniel Nicolas for improving texture internal formats support and helping on raygui development.

- Marc Agüera for testing and using raylib on a real product ([Koala Seasons](http://www.koalaseasons.com))

- Daniel Moreno for testing and using raylib on a real product ([Koala Seasons](http://www.koalaseasons.com))

- Daniel Gomez for testing and using raylib on a real product ([Koala Seasons](http://www.koalaseasons.com))

- [Sergio Martinez](https://github.com/anidealgift) for helping on raygui development and tools development (raygui\_styler).

- [Victor Fisac](https://github.com/victorfisac) for developing physics raylib module (physac) and implementing PBR materials and lighting systems... among multiple other improvements and multiple tools and games.

- Albert Martos for helping on raygui and porting examples and game-templates to Android and HTML5.

- Ian Eito for helping on raygui and porting examples and game-templates to Android and HTML5.

- [procedural](https://github.com/procedural) for testing raylib on Linux, correcting some bugs and adding several mouse functions.

- [Chris Hemingway](https://github.com/cHemingway) for improving raylib on OSX build system.

- [Emanuele Petriglia](https://github.com/LelixSuper) for working on multiple GNU/Linux improvements and developing [TicTacToe](https://github.com/LelixSuper/TicTacToe) raylib game.

- [Joshua Reisenauer](https://github.com/kd7tck) for adding audio modules support (XM, MOD) and reviewing audio system.

- [Marcelo Paez](https://github.com/paezao) for helping on OSX High DPI display issue.

- [Ghassan Al-Mashareqa](https://github.com/ghassanpl) for an amazing contribution to raylib Lua module.

- [Teodor Stoenescu](https://github.com/teodor-stoenescu) for improvements on OBJ object loading.

- [RDR8](https://github.com/RDR8) for helping with Linux build improvements.

- [Saggi Mizrahi](https://github.com/ficoos) for multiple fixes on Linux and audio system.

- [Daniel Lemos](https://github.com/xspager) for fixing issues on Linux games building.

- [Joel Davis](https://github.com/joeld42) for adding raycast picking utilities and a [great example](https://github.com/raysan5/raylib/blob/master/examples/models/models\_mesh\_picking.c)

- [Richard Goodwin](https://github.com/AudioMorphology) for adding RPI touchscreen support.

- [Milan Nikolic](https://github.com/gen2brain) for adding Android build support with custom standalone toolchain.

- [Michael Vetter](https://github.com/jubalh) for improvements on build system and his work on Suse Linux package... and multiple fixes!

- [Wilhem Barbier](https://github.com/nounoursheureux) for adding Image generation functions, shaders work and some fixes.

- [Benjamin Summerton](https://github.com/define-private-public) for improving OSX building and his amazing work on CMake build sytem.

- [MartinFX](https://github.com/Martinfx) for adding compilation support for FreeBSD OS and derivatives.

- [Ahmad Fatoum](https://github.com/a3f) for implementing CI support for raylib (Travis and AppVeyor) and great improvements on build system.

- [SamNChiet](https://github.com/SamNChiet) for a greatly improved UWP input implementation.

- [David Reid](https://github.com/mackron) for a complete review of audio module to support his amazing miniaudio library.

- [Kai](https://github.com/questor) for multiple code reviews and improvements.

- [Max Danielsson](https://github.com/autious) for adding support for orthographic 3d camera projection

- [Lumaio](https://github.com/TheLumaio) for his great work on GBuffers and GetCollisionRayModel().

- [Jonas Daeyaert](https://github.com/culacant) for an amazing work on IQM animated models support.

- [Seth Archambault](https://github.com/SethArchambault) for the work on Android Gamepad support (SNES model).

- [D3nX](https://github.com/D3nX) for adding Code::Blocks project template.

- [Jak Barnes](https://github.com/Syphonx) for a great work on `rnet`, new raylib network module

- [Vlad Adrian](https://github.com/Demizdor) for an amazing work on Unicode support, new shapes functions and raygui.

- [Reece Mackie](https://github.com/Rover656) for a great work on improving UWP support

- [flashback-fx](flashback-fx) for improving easings library and example

- [Jorge A. Gomes](https://github.com/overdev) for adding nine-patch textures support and example

- [Berni8k](https://github.com/Berni8k) for improving Raspberry Pi input system, using evdev

- [Wilhem Barbier](https://github.com/wbrbr) for implementing glTF loading support and solving several issues

- [Marco Lizza](https://github.com/MarcoLizza) for improving logging system and multiple issues

- [Anata](https://github.com/anatagawa) for creating amazing examples and contributing with them

- [Narice](https://github.com/narice) made easings.h includable as standalone header

- [Eric J.](https://github.com/ProfJski) for shaders\_eratosthenes example contribution

- [PompPenguin](https://github.com/PompPenguin) for reviewing 3rd person camera

- [Mohamed Shazan](https://github.com/msmshazan) for adding support for ANGLE graphics backend

Please, if I forget someone in this list, excuse me and send me a PR!