Roles

The following roles participate in the GODOT community:

Users

Community members who have need for the project. Anyone having joined space-codev.org can have the GitLab Reporter role within the project for which this governance applies. Reporters are considered users of the software and users are the most important members of the community: without them, the project would have no purpose.

Users are encouraged to participate in the life of the project and the community as much as possible. User contributions enable the community to ensure that they are satisfying the needs of those users. Common user activities include (but are not limited to):

- Raising issues and requests for improvements.
- Participating in issues discussions
- Informing developers of project strengths and weaknesses from a new user’s perspective
- Creating and publishing own solution which show others the use of the project
- Providing financial support
- Evangelising about the project

It is hoped that as users become more involved in the project, they may perform more of activities, both in volume and of the types listed above.

Contributing

Any user, having signed the Contributor Agreement through the Contributor Agreement Signature Process, can provide a contribution to the project by cloning the project and then creating a merge request.

The source code of any User's contribution must be positively code-reviewed by at least two contributors. As contributing Users gain experience with the project and the Developers start relying on them more and more, Users may gradually adopt the role of Developer.

Developers

Developers have the GitLab Developer role. Applicable GitLab role permissions are reported at the GitLab Project members permissions page.

Developers are trusted by the maintainers to make the appropriate technical decisions on the code and have significant proven experience on the project. Developers' commit contributions must be positively code-reviewed by at least one other core developer, to ensure peer code review. Developers do not have the authority to make direct changes to protected branches.

Developers must have the Contributor status by signing the Community Contributor License Agreement (CLA).

Common developers’ activities include (but are not limited to):

- Supporting new users and developers
- Reporting Bugs
- Identifying requirements
- Programming
- Writing documentation
- Fixing bugs
- Adding Features

- Other support activities (graphic and web design, project infrastructure, evangelizing project, etc.)

Responsibilities

Developers are responsible for:

- Ensuring a proper peer code review exists and adherence of everyone to the governance and applicable rules.
- The technical evolution of the software (e.g. such as software design and coding aspects).

Membership

Any user can become a Developer. A potential Developer will need to have provided valuable contributions to the project over a period of time, show a willingness and ability to participate in the project as a team player, and show that they have an understanding of the project, its objectives and its strategy.

Maintainers

Maintainers have the GitLab Maintainer role. Applicable GitLab role permissions are reported at the GitLab Project members permissions page.

They are community members who have made several valuable contributions to the project and are now relied upon to both write code directly to the repository and screen the contributions of others. Maintainers are appointed by the Project Leader.

The role of maintainer is not an official one, it is simply a position that influential members of the project will find themselves in as the project lead looks to them for guidance and support. They have no authority over the overall direction of the project. However, they do have the ear of the project lead.

It is a maintainer’s job to ensure that the lead is aware of the community’s needs and collective objectives, and to help develop or elicit appropriate contributions to the project.

Therefore, they are responsible for the general management of open issues:

- Tagging or categorizing issues
- Answer questions to issues
- Prioritizing issues
- Rejecting issues posted by users

Project leader

The Project Leader is self-appointed. However, because the community always has the ability to fork, this person is fully answerable to the community. They are responsible for:

- Establishing the project strategic objectives and communicate them clearly to the community through the communication mechanisms established for this project
- Ensuring the project survives in the long term
- Providing influence on the right people (maintainers) as project expands
- Ensuring that maintainers make the right decisions on behalf of the project
- Understanding the community and strive to satisfy as many conflicting needs as possible