

Lambda-Mu Wiki - <https://www.wiki.lambda-mu.com/>

(horizontal, vertical, grid, center, margin, draggable splitter, ...). - Controls can be rotated and scaled. **Sizing:** - Anchors to keep GUI elements in a specific corner, edge or centered. - Containers to place GUI elements automatically following certain rules. - :ref:`Stack <class\_BoxContainer>` layouts. - :ref:`Grid <class\_GridContainer>` layouts. - :ref:`Margin <class\_MarginContainer>` and :ref:`centered <class\_CenterContainer>` layouts. - :ref:`Draggable splitter <class\_SplitContainer>` layouts. - Scale to multiple resolutions using the ``2d`` or ``viewport`` stretch modes. - Support any aspect ratio using anchors and the ``expand`` stretch aspect. **Theming:** - Built-in theme editor. - Generate a theme based on the current editor theme settings. - Procedural vector-based theming using :ref:`class\_StyleBoxFlat`. - Supports rounded/beveled corners, drop shadows and per-border widths. - Texture-based theming using :ref:`class\_StyleBoxTexture`. Godot's small distribution size can make it a suitable alternative to frameworks like Electron or Qt. Animation ^^^^^^^ - Direct kinematics and inverse kinematics. - Support for animating any property with customizable interpolation. - Support for calling methods in animation tracks. - Support for playing sounds in animation tracks. - Support for Bézier curves in animation. Formats ^^^^^^^ - Scenes and resources can be saved in :ref:`text-based <doc\_tscn\_file\_format>` or binary formats. - Text-based formats are human-readable and more friendly to version control. - Binary formats are faster to save/load for large scenes/resources. - Read and write text or binary files using :ref:`class\_File`. - Can optionally be compressed or encrypted. - Read and write :ref:`class\_JSON` files. - Read and write INI-style configuration files using :ref:`class\_ConfigFile`. - Can (de)serialize any Godot datatype, including Vector, Color, ... - Read XML files using :ref:`class\_XMLParser`. - Pack game data into a PCK file (custom format optimized for fast seeking), into a ZIP archive, or directly into the executable for single-file distribution. - :ref:`Export additional PCK files<doc\_exporting\_pcks>` that can be read by the engine to support mods and DLCs. Miscellaneous ^^^^^^^^^^^^^^^ - :ref:`Low-level access to servers <doc\_using\_servers>` which allows bypassing the scene tree's overhead when needed. - Command line interface for automation. - Export and deploy projects using continuous integration platforms. - `Completion scripts <<https://github.com/godotengine/godot/tree/master/misc/dist/shell>>`

are available for Bash, zsh and fish.

- Support for :ref:`C++ modules <doc\_custom\_modules\_in\_c++>` statically linked

into the engine binary.

- Engine and editor written in C++03.

1. Can be :ref:`compiled <doc\_introduction\_to\_the\_buildsystem>` using GCC,

Clang and MSVC. MinGW is also supported.

1. Friendly towards packagers. In most cases, system libraries can be used

instead of the ones provided by Godot. The build system doesn't download anything.

Builds can be fully reproducible.

- Godot 4.0 will be written in C++17.

- Licensed under the permissive MIT license.

1. Open development process with :ref:`contributions welcome <doc\_ways\_to\_contribute>`.

.. seealso::

The ``roadmap`` <<https://github.com/godotengine/godot-roadmap>>`__ repository documents features that have been agreed upon and may be implemented in future Godot releases.

From:

<https://www.wiki.lambda-mu.com/> - **Lambda-Mu Wiki**

Permanent link:

<https://www.wiki.lambda-mu.com/lm3/ce/start?rev=1605199288>

Last update: **2020/11/30 14:12**

