Документация по продукту Lambda-Mu Community Edition Version 3

Туториалы

- Мастер-экземляр платформы ("балансировщик" нагрузки)
- Мультиязычность

Технические характеристики



Установка

Первые шаги

How-To

Иснтрументы

Проекты сообщества

Разное

.. _doc_list_of_features:

List of features

This page aims to list all features currently supported by Godot.

.. note::

```
  This page lists features supported by the current stable version of
  Godot (3.2). `More features
<https://docs.godotengine.org/en/latest/about/list_of_features.html>`__
  are available in the latest development version (4.0).
```

Features

Physics and Visual Studio Code or Vim.

Platforms:

- OpenGL ES 2.0 renderer (uses OpenGL 2.1 on desktop platforms).
- OpenGL ES 2.0 renderer (uses OpenGL 2.1 on desktop platforms).
- Recommended on mobile and Web platforms.
- Recommended on compiled Web platforms.

Features:

- No support for debugging in threads yet.
- Not all features are available. Features available only when using the OpenGL ES 3.0 renderer are marked *GLES3* below.
- Changes will reflect in the editor and will be kept after closing the running project.

Camera:

- Remote inspector.
- Pseudo-3D support by automatically duplicating a layer several times.
- Changes won't reflect in the editor and won't be kept after closing the running project.

Physically-based rendering:

- Live camera replication.
- Follows the Disney PBR model. - Uses a roughness-metallic workflow with support for ORM textures. - Normal mapping. - *GLES3:* Parallax/relief mapping with automatic level of detail based on distance. - *GLES3:* Sub-surface scattering and transmittance. - *GLES3:* Proximity fade (soft particles). - Distance fade which can use alpha blending or dithering to avoid going through the transparent pipeline.
- Font rendering using bitmaps (BitmapFont) or rasterization using FreeType (DynamicFont).

Plugins:

- Editor plugins can be downloaded from the :ref:`asset_library <doc_what_is_assetlib>` to extend editor functionality.
- Create your own plugins using GDScript to add new features or speed up your workflow. - Download GPU-based particles with support for custom particle shaders. - CPU-based particles.

Real-time lighting:

- Directional lights (sun/moon). Up to 4 per scene. - Omnidirectional lights. - Spot lights with adjustable cone angle and attenuation.

Shadow mapping:

- *DirectionalLight:* Orthogonal (fastest), PSSM 2-split and 4-split.

```
Supports blending between splits.
```

- *OmniLight:* Dual paraboloid (fast) or cubemap (slower but more accurate).

```
Supports colored projector textures in the form of panoramas.
```

- *SpotLight:* Single texture.

Global illumination with indirect lighting:

- Baked lightmaps (fast, but can't be updated at run-time).

1. Lightmaps are baked on the CPU.

- *GLES3:* GI probes (slower, semi-real-time). Supports reflections.

Reflections:

- *GLES3:* Voxel-based reflections (when using GI probes). - Fast baked reflections or slow real-time reflections using ReflectionProbe.

```
Parallax correction can optionally be enabled.
```

- *GLES3:* Screen-space reflections. - Reflection techniques can be mixed together for greater accuracy.

Sky:

- Panorama sky (using an HDRI). - Procedural sky.

Fog:

- Depth fog with an adjustable attenuation curve. - Height fog (floor or ceiling) with adjustable attenuation. - Support for automatic depth fog color depending on the camera direction

```
(to match the sun color).
```

- Optional transmittance to make lights more visible in the fog.

Particles:

- *GLES3:* GPU-based particles with support for custom particle shaders. - CPU-based particles.

Post-processing:

- Tonemapping (Linear, Reinhard, Filmic, ACES). - *GLES3:* Automatic exposure adjustments based on viewport brightness. - *GLES3:* Near and far depth of field. - *GLES3:* Screen-space ambient occlusion. - Glow/bloom with optional bicubic upscaling and several blend modes available:

```
Screen, Soft Light, Add, Replace.
```

- Color correction using an one-dimensional ramp. - Brightness, contrast and saturation adjustments.

Texture filtering:

- Nearest, bilinear, trilinear or anisotropic filtering.

Texture compression:

- *GLES3:* BPTC for high-quality compression (not supported on macOS). - *GLES3:* ETC2 (not supported on macOS). - ETC1 (recommended when using the GLES2 renderer). - *GLES3:* S3TC (not supported on mobile/Web platforms).

Anti-aliasing:

- Multi-sample antialiasing (MSAA).

Most of these effects can be adjusted for better performance or to further improve quality. This can be helpful when using Godot for offline rendering.

Collision detection:

APIs used:

GDNative (C, C++, Rust, D,):

(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 BÃ©zier 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 developement 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://wiki.lambda-mu.com/ - **Lambda-Mu Wiki**

Permanent link:
https://wiki.lambda-mu.com/lm3/ce/start?rev=1606472834

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