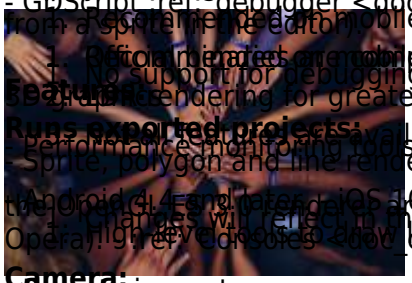
Документация по продукту Lambda-Mu Community Edition Version 3

1. High-end visuals. Recommended on desktop platforms.

Технические характеристики:

- OpenGL ES 2.0 renderer (uses OpenGL 2.1 on desktop platforms).

- GDScript :ref: debugger <doc_debugger_panel>.



1. Efficient rendering for greater compatibility.

Runs exported projects:

- Performance monitoring tools. - Live script reloading. - Live scene editing.

- Android 4.4 and later. - iOS 1.0.0 and later. - HTML5 via WebAssembly (Firefox, Chrome, Edge, Opera). - :ref: Consoles <doc_consoles>.

Camera:

- Animated Sprite as a helper for creating animated sprites. - Parallax layers.

Установка:

- Pseudo-3D support by automatically duplicating a layer several times.

Physically-based rendering:

- 2D lighting with normal maps.

Первые шаги

- 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

Plugins:

- Bitmap fonts can be exported using tools like BMFont.

Инструменты

- Editor plugins can be downloaded from the

Проекты сообщества

- Dithering can be determined on a per-pixel or per-object basis.

Real-time lighting:

- Create your own plugins using GDScript to add new features or speed up your workflow. - Download projects from the asset library in the project manager and import them directly.

Paste:

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

Shadow mapping:

List of features

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

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

- *SpotLight:* Single texture.

Global illumination with indirect lighting:

Features

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

---

1. Lightmaps are baked on the CPU.

Platforms

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

Reflections:

Physics: - Built-in 3D physics engine, using Bullet Physics library. - Rigid body, soft body, vehicle, and ragdoll physics. - Joints (pin, hinge, slider, cone twist, generic 6DOF, and generic 6DOF spring). - Area nodes for gravity/damp overrides. - Continuous collision detection and kinematic motion. - Ray and shape casting. - Support for adding custom forces and custom physics integration. - Physical bones for 3D ragdoll physics. - Soft body simulation. - Cloth simulation. - Support for world boundary shapes, separation rays, and sphere shapes. - Built-in vehicle physics. - Support for 3D joints. - Support for 3D soft bodies. - Support for 3D ragdolls. - Support for 3D cloth. - Support for 3D vehicles. - Support for 3D world boundary shapes, separation rays, and sphere shapes

(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=1605199288

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