



Tortilla Engine quick manual

Video games engine built in C++ with SDL and OpenGL.

Tortilla Engine folders hierarchy

The engine's folder has three folders inside: code, libraries and projects.

“Code” has the headers and sources subfolders with the code of all the engine aspects.

“Libraries” has the dependencies of the engine.

“Projects” has the projects in different IDEs.

How to compile Tortilla Engine

The engine project is a Visual Studio 2017 project that can be open with Microsoft Visual Studio 2017 or greater without conflicts. Once the project is downloaded in your computer, the project solution is in the “vs-2017” subfolder of “projects” folder.

The compilation of this project generates the .dll of the engine.

Tortilla Engine dependencies

The engine uses SDL2 ([link](#)), OpenGL Toolkit ([link](#)) and SDL-Mixer ([link](#)). All the dependencies are added into the folder “libraries”.

How to add code files to Tortilla Engine

In the engine project you can add your files in the code folder with new behaviours, modify the existing...

Tortilla Engine Systems

Tortilla Engine is based on entities (TEntity), components (TComponent) and tasks (TTask). The congregation of related components and tasks that manage them is what we call a system.

Tortilla Engine has the following systems by purpose:

- Audio System: with the audio (TAudio) and sound effect (TSound) components.
- Collision System: with the collider (TCollider), sphere collider (TSphereCollider) and collision task (TCollisionsTask).
- Input System: with the keyboard support (TKeyboardInput), the management of the input events (TInputTask) and the input messages mapping (TInputMapperTask).
- Message System: with the sender (TDispatcher), the message (TMessage) and the listeners (TObserver).
- Render System: with the camera (TCameraComponent), light (TLightComponent), 3D model/mesh (TRenderComponent) and the render task (TRenderTask).
- Script System: with the scripts (TScriptComponent), a special script to update components (TUpdateComponent) and the task to execute the scripts (TScriptTask).
- Transform System: with the component that stores and apply transformations to the entities (TTransformComponent).

All the details of class dependencies, communication between classes and class members can be found in the documentation folder.