Rasterizer – Technical documentation

# Coding conventions

* lowerCamelCase (CamelCase but with a lower first letter)
* Hungarian notation :
  + Function/Method parameter : p\_[variableName];
  + Class (Not structs) attributes : m\_[variableName];
* Macro naming : MACRO\_NAME
* Class constants : \_\_CONSTANT\_NAME
* If a method is empty, put {} in the .cpp (EX : Class::~Class() {})

# Version 1 running process:

* Create SDL window (1024 x 768)
* Create an instance of the Rasterizer class (Called RI)
* RI must be composed of a Scene attribute
* Scene must have a MeshManager (MM) to avoid duplicating meshes
* Every created meshes must get added to MM and will be referenced by an ID (std::vector<Mesh\*> elementID or std::map<std::string, Mesh\*>)
* RI create a 1024 x 768 texture
* RI must loop as follows:
  + Evaluate every entity of the scene and render them to the texture
  + Render the texture to the SDL Window
* Application should close on Echap or “X” window icon

# Architecture

Every single class of the project is a child of the Object class. This class got a static counter to how many objects are instantiated or get instantiated since the program is running.