Application Flow

Below is the current flow of the application in function calls. The order is important for certain calls such as OpenGL functions, any object which relies on a certain system.

The main function is inside entryPoint.h which will create a new Application which is an extern function defined in the sandbox’s Game Application class.

* **entryPoint.h::main**
* **Application Constructor (Engine)**
  + Initialises the System Manager with a value for the maximum number of systems
  + Individual Systems are then added. In the following order:
    - Logger
    - Randomiser
    - TimerSystem
    - WindowAPISystem
      * GLFW is initialised
    - WindowManager
    - EventManager
    - ResourceManager
      * Certain resource configuration values are loaded from the config json file
    - SceneManager
    - FontManager
      * Freetype initialised
      * First & Last Glyph set
    - RenderSystem
      * TextureUnitManager created
  + FPS Timer and TotalTime timers are created
* **GameApplication Constructor (Sandbox)**
  + Window is created
    - OpenGL Context is created (Loads OpenGL Functions)
    - Event callbacks for the window are set
    - Window Icon is set
  + RenderSystem initialised
  + Arial font loaded into Font Manager
  + SceneLoader loads a new scene
    - New scene is created by the Scene Manager
    - Resources needed by the scene are loaded and registered into the Resource Manager (from file)
    - Scene clear colour set (from file)
    - Scene’s layers are loaded into the scene (from file)
    - Scene’s render passes are loaded into the scene (from file)
    - Entities are then loaded into the scene (from file)
      * Entity is added to the scene
      * Entity layer is set
      * Components are attached
* **Application::run (Engine)**
  + Event Manager Update
    - Store the FPS time and reset it
    - Store the Total time of the application
    - Update the event data times
    - Entity preupdate
    - InputPoller checks for all keys currently pressed and calls the appropriate input functions
    - InputPoller checks for all mouse buttons currently pressed and calls the appropriate input functions
    - Active scene is updated
    - Update all registered windows
      * Swap back buffer
      * Poll Events (Window event call backs occur here)
    - Entity postupdate
  + RenderSystem::onRender
    - Entity list is retrieved if the entity list is updated
    - Get all render passes of the scene
    - Go through each render pass and render
  + Set NewEntity’s flag in the active scene is set to false
  + Deregister scheduled scenes
  + Deregister Scheduled Windows
    - Destroys all windows which are set to be destroyed
  + Check the exit conditions of the application
* **GameApplication Destructor (Sandbox)**
  + Currently empty
* **Application Destructor (Engine)**
  + Destroy the system manager
    - Which destroys all systems in reverse order and their resources