Serenity Project

Day 1:

The following are function for the noGL (that is no graphic library) platform layer.

Use these when you want no dependencies on an graphics library like OpenGL, Direct3D, etc.

void SerenitySetupRenderCallback(RenderCallbackFunction\* renderFunction);

Whenever you try to render a frame (using SerenityRender) it will call the function defined by you and passed to the platform layer using this function call.

Define a function void RenderCallbackFunction(int windowIndex, void\*\* buffer);

The window index (the window the rendering will take place in) and the buffer(the buffer in which to write the pixel values) will be provided as such by the platform layer when it is calling into this function.

int32 SerenityCreateWindow(int32 width, int32 height, const char\* className, const char\* windowName);

Creates a platform window with the given specifications and return the window index that can be used to extract information about the particular window like width, height, etc.

void SerenityDestroyWindow(int32 windowIndex);

Frees the window slot taken by the particular window, frees memory.

boolean SerenityApplicationShouldClose();

Returns true if there are no remaining windows in the application

void SerenityWindowPollEvents();

Updates the message queue, updates information like input, window close, quit, focus, etc.

void SerenityRender();

Goes through all the active (not destroyed) windows and updates the pixels of screen based on the contents of the buffer that were specified by our RenderCallback function.

Use Case:

First call SerenityCreateWindow to create a window.

Then setup a application loop as such

while (!SerenityApplicationShouldClose())

{

}

In the loop we have to Poll Events using SerenityPollEvents() and render on screen using SerenityRender().