**IGAD Template documentation**

**Introduction**

The IGAD template has been developed for first-year students of the IGAD program of the NHTV University of Applied Sciences, and provides a basic framework for low-level game graphics programming. Its main function is to open a window and provide access to a linear 32-bit ARGB frame buffer for plotting. It provides classes for pixel storage (Surface and Surface8), basic sprites (Sprite), and a basic game loop (Game::Init, Game::Tick, Game::Shutdown).

The template is distributed as a small set of source files. Low-level window handling is handled by template.h/template.cpp; these files are not supposed to be modified. Surface classes are implemented in surface.h/surface.cpp; these files can be changed at will.

Starting a new experimentation project is generally a matter of extracting the template to a fresh folder, and modifying game.h / game.cpp.

**Game class**

The Game class contains a dummy implementation of the game loop. Initialization code can be added to Game::Init. The template code will then call Game::Tick once per frame, with the ‘dt’ parameter containing the duration of the previous frame. Upon exit (default: ‘ESC’) the Game::Shutdown method is called.

Additional methods, data and classes can obviously be added to the Game class or the project.

**Surface class**

The main class provided by the template is the surface class. A surface contains a 1D array of pixels, and methods to operate on these pixels, such as line drawing, basic text rendering and copying.

The template provides one surface to the Game object, ‘screen’, which represents the pixels in the window.

Surfaces can be constructed easily from bitmap files.

**External libraries**

The template uses GLM for vector and matrix math. FreeImage is used for image loading. SDL2 is used for low-level window handling.

**Advanced**

When starting the template, two windows are opened: the graphical window, and a console window. The console window can be used to print to using printf. This is useful for debugging. To disable the console window, remove the redirection line in the main function in template.cpp.

The size of the window can be set in game.h (default: 1280 by 800). Fullscreen output can be obtained by adding SDL\_WINDOW\_FULLSCREEN to the SDL\_CreateWindow command in template.cpp. Be careful with debugging; a full-screen window will not handle breakpoints very well.

Template.h contains a number of convenience functions, macros, typedefs and structs; PI is defined, an accurate timer is provided, and functions for aligned memory allocation are available. Feel free to add your own material here for future projects.

A clean.bat is included to remove all temporary files in a single (double)click. Please use this before handing in anything!

**Problems compiling?**

Under VS2015, you may get missing include files. If this happens: right click the solution, and retarget the app. If this still fails, download and install the Windows 10 SDK and retry the retarget.

Happy coding,

Jacco Bikker.