Kevgine (Win32)

This is the Win32 engine created by Kevin, with a strong emphasis on its Win32 backend. There are multiple ways to use this engine.

There is a student-maintained version of this framework that has a switchable backend, meaning you can seamlessly switch between Win32 and SDL2. Using the SDL2 backend, the engine can run natively on Linux (and Windows, of course). You can also always switch it to the WIN32 backend to get perfect compatibility, and test it via WINE.

The project is available in the Solution-Runner Templates directory.

> NOTE: The SDL2 backend is provided as best-effort and will not perfectly match the behavior of the Win32 version. Most parts of the engine are faithfully recreated, including the Win32 quirks that come with it, but it is not perfect. For example, sound is not implemented with the SDL2 backend. Please **make sure to always test a Win32 build with WINE (via Solution-Runner) before hand-in!**

WINE

Using LLVM, you can cross-compile the engine as provided to a Windows .exe file, and run it inside of WINE. **You should always do this at least once before hand-in!** The teachers will grade your code based on how it runs for them on Windows!

There are two ways to run your code on WINE:

Solution-Runner (recommended)

Kevgine is fully supported by the Solution-Runner project. With it, you can compile and run the program via WINE using the Win32 backend.

Manually cross-compile (not recommended)

See this in-depth blog post explaining the process for Kevin's engine: Cross Compiling a Windows Game Engine

Natively

To use the SDL2 backend, simply open the Kevgine template from the Solution-Runner repository in a CMake-enabled IDE.

> NOTE: KDevelop is the recommended IDE for C++ on Linux. It has great CMake support and is tested with these projects. Simply open the folder, and press Execute to run it. The first time, you will have to add a Launch Configuration. Simply select `game` from the `Add` dropdown. (TODO: make this a page)

Please make sure you have CMake, fontconfig, and base development tools installed. On Arch Linux, you can use this command:

```
sudo pacman -S base-devel cmake fontconfig
```

The game should run natively on Linux, and you can use any usual development tools like gdb as usual.

Hello world!

Revision #6
Created 4 February 2024 21:22:25 by Mat
Updated 21 February 2024 15:37:52 by Mat