Geometric Tools Engine Version 1.0
Installation Manual and Release Notes

Document Version 1.0
August 11, 2014

Contents

1 Introduction

You are about to install Geometric Tools Engine 1.0 (GTE1). Version 1.0 is source code that is the companion to the book GPGPU Programming for Games and Science. The source code was developed on Microsoft Windows 8.1 using Microsoft Visual Studio 2013 and Direct3D 11.1. We are now using C++ 11 constructs, so it is necessary to use version 2013 of the compiler. Linux-capable and Macintosh-capable distributions are under development, mainly requiring an OpenGL 4.x-based graphics engine. The mathematics algebra code (vectors, matrices, quaternions, and so on) have been written to compile using older versions of the compiler, so if you have a non-graphics application, you can at least use that code. Other mathematics code will probably compile also with older versions, but we have not yet run a test to determine all those files. We have also not run a test to determine which files compile using Linux or Macintosh OS X.

You should visit our website regularly for updates, bug fixes, known problems, new features, and other materials. The update history page always has a date for the last modification, so you should be able to track what you have or have not downloaded. The source files themselves are labeled with a version number of the form *1.minor.revision*, where *minor* is the minor version in which the file shipped and where *revision* is the number of times the file has been revised. The source files in the initial distribution of Geometric Tools Engine 1.0 are labeled with 1.0.0.

2 License

Geometric Tools Engine 1.0 uses the Boost License. The license in its entirety is listed next.

3 Copying the Distribution to Your Machine

You may unzip the distribution to a folder of your choice. The top-level folder of the distribution is `GeometricTools` and the subfolder for the distribution is named `GTEngine`. The projects all use paths relative to `GTEngine` and they do not rely on the top-level directory being located at the root of a hard drive.

Some of the folder hierarchy is shown next. The `Include` and `Source` folders contain all the code for the engine. We chose to omit the post-build copy of header and inline files to an `SDK` folder (as done in Wild Magic). If you have projects outside the `GTEngine` hierarchy, you can add a search path in a Microsoft Visual Studio project for the header and inline files using `$(GTE_PATH)/Include`. We use Microsoft Visual Studio's reference system in the project settings for the applications to find the libraries to link to.

```
GeometricTools
  GTEngine                      // Root folder for Geometric Tools Engine 1.0, set GTE_PATH to here.
    Include                     // Location for *.h, *.inl, *.hlsl, and *.hlsli files.
    Source                      // Location for *.cpp files.
    Samples                     // Sample applications, many discussed in the GPGPU book.
      #Data                     // A small number of data files for the samples.
      Basics                    // Basic tutorials for several HLSL concepts.
      Geometrics                // Samples for computational geometry.
      Graphics                  // Samples for graphics and video streams (parallel copy).
      Imagics                   // Samples for 2D and 3D image processing.
      Numerics                  // Samples for numerical methods.
      Physics                   // Samples for 2D and 3D physics.
    Tools                       // Several convenient tools.
      BitmapFontCreator         // Generate .h/.cpp file to represent a graphics font.
      GenerateApproximations    // Used to generate the minimax approximations for common functions.
      GenerateOpenGLWrapper     // Source-code generator for gl* wrappers drive by glcorearb.h.
      GenerateProject           // Generate a skeleton (vcxproj, sln, h, cpp) for GTE1 applications.
    Executable                  // Output folders for generated executables.
      v120
        Win32
          Debug
          Release
        x64
          Debug
          Release

    // These folders and files are separate because you need to download AMD's SDK
    // for GPU performance measurements.
    #Samples
      Basics
        PerformanceAMD          // Sample to illustrate how to use AMD's SDK.
    #Tools
        GPUPerfAPI-2.11.739.0   // Download AMD's SDK to here, some of our source is also here.
```

The `Samples` subfolders are many. Listing them here would make the displayed hierarchy difficult to read.

4 Environment Variables

Data files are found using an environment variable that you must create; its name must be `GTE_PATH` and its value must be the path to where you installed `GTEngine`. For example, if your user account is in the default location and you installed the `GeometricTools` folder to your user account folder, you will create an environment variable

```
GTE_PATH = C:\Users\YourAccount\GeometricTools\GTEngine
```

To create an environment variable, you need to launch the System window that is accessible through the Control Panel. The simplest way to do this is via the Windows shortcut: `Windows Key` + `Pause/Break`.

Figure 1 shows the top-half of the window.

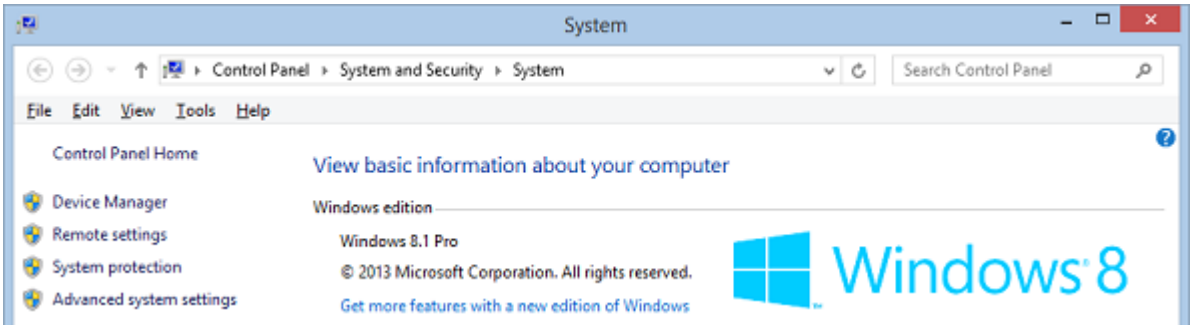

Figure 1. The System dialog window.

Select `Advanced system settings` to launch the dialog shown in Figure 2(a).

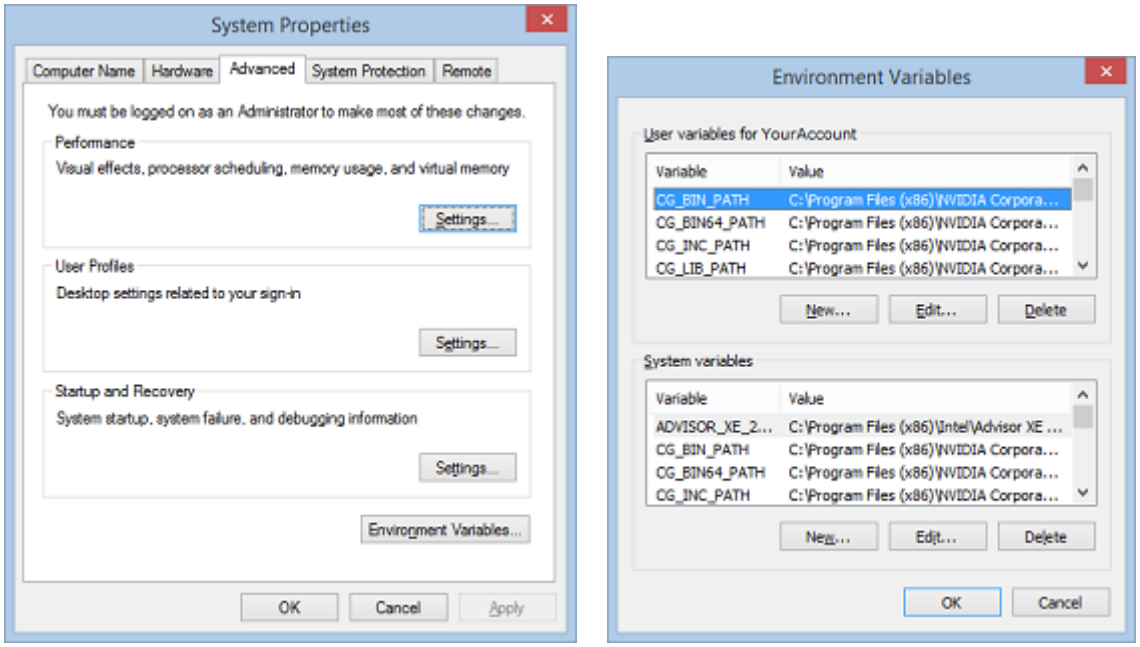

Figure 2. (a) The Advanced System Settings dialog window. (b) The Environment Variables dialog window.

Select the `Enviornment Variables ...` button to launch the dialog shown in Figure 2(b). Under `User variables for YourAccount`, select the `New ...` button to launch the dialog shown in Figure 3.

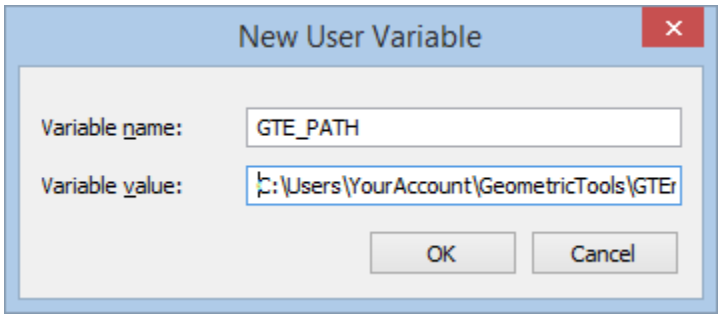

Figure 3. The New User Variable dialog window.

Enter the name and path as shown in the figure. Naturally, you will use the path to your installation. The dialog cannot be resize, so the environment variable value is not fully displayed. Press the `OK` button.

5 Compiling the Source Code

The engine solution is `GeometricTools/GTEngine/GTEngine.sln`. Each sample application or tool has its own solution with all dependencies listed, so it is possible to open a sample application and compile and run it without explicitly building the engine solution first. The folder `GTEngine` contains the solution `GTBuildAll.sln` if you want to build the engine, samples, and tools rather than building the projects separately.

Currently, the engine solution generates static libraries. We have the hooks in place for dynamic libraries but have not yet created the build configurations. We will provide the dynamic library configurations eventually.

6 The Next Version of the Geometric Tools Engine

The following items are planned for the next version of the engine.

- We are working on an OpenGL 4.x-based engine so that the engine may be used with Linux or Macintosh OS X.
- Dynamic build configurations will be available.
- We are still writing unit tests for some of the core code and end-to-end tests for more complicated algorithms that were ported from Wild Magic 5. Although we could have shipped GTE1 without the tested files, we chose to do so anyway. If you run into problems with any of our code, please be patient and simply file a bug report. We will prioritize our testing so that bug reports get attention as soon as possible.
- Support 256-bit registers when AVX SIMD is available.
- Port most of the Wild Magic 5 sample applications.
- Add custom visualizers for the mathematics code.