Setting Up Microsoft Visual Studio to Work with Allegro

For you to have the best learning experience and the best learning outcome, it is very important that you choose the best combination of an IDE and a version of Allegro game library.

Chapter 2 in the e-textbook provides useful instructions about getting started with the Allegro game library.

This document gives you more specific guidance on choosing and installing an IDE and a version of Allegro game library. (We will try our best to keep this document updated, but if the instructions here do not work for you, the course forum, { allegro.cc }, and Allegro: A game programming library can be good sources of help from your peers and others within the global Allegro community. If you still need help, please don't hesitate to contact your tutor.)

The combination of Allegro 4.4.2, available on the files page of {allegro.cc}, and the updated version of Dev-C++, available at sourceforge Dev-C++ , has for years been the most favoured combination for students completing the learning tasks in COMP 369, including all the assignment projects.

Being a multi-platform game development library, Allegro 4.4.2 should also work with Visual Studio 8, 9, 10, 2012, and 2013 versions on Windows XP, Windows 7, and Windows 8.

To set up an Allegro 4.4.2 library to work with Microsoft Visual Studio:

1) Download and install your preferred version of Microsoft Visual Studio from the versions mentioned above. You can download a copy at https://triton2.athabascau.ca/dreamspark_scis/, through the Microsoft DreamSpark program. It is free for students and faculty members of eligible institutions. If you are unable to access the URL, you may not yet have an account; in that case, please contact our IT support by sending email to scistech@athabascau.ca.

   From time to time, Microsoft may remove older versions of Visual Studio from the site. If that is the case, please try to get a newer one that is still available, or email your tutor or course professor for help.

2) Then, you need to get the right Windows binaries of Allegro 4.4.2 from https://www.allegro.cc/files/?v=4.4 for the Visual Studio you have installed on your computer. For example, for MSVC 10 and higher, you would download http://cdn.allegro.cc/file/library/allegro/4.4.2/allegro-4.4.2-msvc-10.0.zip; this library also works with MSVS 2012 and 2013.

   If you are using an earlier version, such as MS Visual Studio 2005, select the MSVC 8 archive

http://cdn.allegro.cc/file/library/allegro/4.4.2/allegro-4.4.2-msvc-8.0.zip for MS Visual Studio 2008 select the MSCV 9 archive http://cdn.allegro.cc/file/library/allegro/4.4.2/allegro-4.4.2-msvc-9.0.zip . (Note that the Microsoft Visual Studio product year name at times does not match the version number; hence the discrepancy above regarding which archive to use.)

Installation Choices

There are two possible methods to install and use the Allegro package with Visual Studio: Standalone, or Integrated.

Method 1: The Standalone method requires configuring each project you create to use specific Allegro directories and files; while it is a little more work than the Integrated method, it is the easiest in terms of troubleshooting the installation, avoiding problems due to newer Windows security settings with file locations, and using different versions of the Allegro library. This approach also allows you to use different Allegro libraries in future by making changes in your projects settings and selecting a different Allegro file location.

Method 2: The Integrated installation approach is taken in the textbook: to copy the files from the Allegro *bin*, *include*, and *library* folders into the respective folder locations of the Visual Studio package you are using, and make fewer project configuration changes. This method will require you to be familiar with the locations of the required Visual Studio directories on your system to copy appropriate files into.

Method 1 (Standalone)
(*setup for Dynamic Linking and using 'Release' configuration)

To install the Windows binaries so that the libraries will work with the installed MSVS:

1. Unzip the Allegro package you downloaded that works with your version of Visual Studio. It will create a folder based on the Allegro version. For example, if you used drive D: (and didn't specify a different folder to extract to), it will create the folder **D:\allegro-4.4.2-msvc**, under which you will find the subfolders **bin**, **include**, and **lib**. In the following steps, you will need to enter information on the locations of these folders, based on your installation; you may need to modify them accordingly.

2. When creating a new Visual Studio Project(s) to use the Allegro library: From the top menus, select **Project** -> **Properties** or press **Alt F7** to bring up the Property Pages.

3. Under **C/C++ -> General** for **Additional Include Directories**, add **D:\allegro-4.4.2-msvc\include** (This is the location on the hard drive where I extracted the Allegro archive file to in step 1 of this example).

4. Under **Linker -> General** for **Additional Library Directories**, add **D:\allegro-4.4.2-msvc\lib**

5. Under **Linker -> Input** for **Additional Dependencies**, add **allegro-4.4.2-monolith-md.lib** ***** For a Debug configuration replace the above .lib file with **allegro-4.4.2-monolith-md-debug.lib**

Method 2 (Integrated)

To install the Windows binaries to work with the installed MSVS:

1. Unzip the package into a temporary folder.
2. Locate the C++ folder within your installed MSVS folder.
3. Copy all the files in the include folder of your Allegro library unzipped to the temporary folder in the include folder of the C++.
4. Copy all the files in the lib folder of your Allegro library into the lib folder of the C++.
5. Copy all the files under bin to the system or system32 folder in the Windows folder of your computer.

To be sure your MSVS/MSVC can find all the files installed, you may need to restart your Windows to update all the paths.

Additional Settings

For general installation instructions, please also visit https://www.allegro.cc/files/ . In some cases, as former students have experienced, you may need to change the following settings to avoid errors:

Under **Project Properties** -> **C/C++** -> **General**, set SDL checks to **No(/sdl-)**

Under **C/C++** -> **Preprocessor Definitions**, add **_CRT_SECURE_NO_WARNINGS** (with the underscores as indicated)

If the executable of your game built by MSVS produces error messages indicating it cannot find some DLLs, you can simply copy the appropriate libraries into the folder with the generated .exe file. You will need to do this **with the corresponding runtime DLL** to match the link library you selected at build time (release/debug); e.g., allegro-4.4.2-monolith-md.dll if you used allegro-4.4.2-monolith-md.lib

Can you use Allegro 5 for your course study?

The short answer is yes, you could use Allegro 5 for COMP 369. But we strongly recommend that you don't. Here's why:

- You would still need to study the Allegro 4 routines, because the textbook uses Allegro 4 to teach game programming principles and essential technologies.

- For the games you need to develop for the assignment projects, if you use the textbook examples as a basis, you would have to write more code from scratch if you used Allegro 5. Because Allegro 5 is so different from Allegro 4, no sample codes from the textbook can be directly used and tested in an IDE (Dev-C++ or MSVC) with Allegro 5 installed. The syntax of the library routines has significantly changed, and the textbook examples will not function (unless you don't mind making all the needed changes to the example code).

- Moreover, 80% of the exam questions are based on Allegro 4.