Task 15 - Lab: SDL2 Concepts

The aim of this lab is to make sure you are familiar with SDL2 for use in later tasks. You will first need to investigate what SDL2 is and answer some questions. Next you will need install SDL2 for use with your IDE setup, and then create a basic program that uses SDL to show a window and respond to events.

Output: Lab Notes (what you find out) uploaded to Canvas as a PDF. Code in repo.

Note: Keep this task focused just like you do with Spikes. Learn about SDL2, demonstrate, and move on.

About SDL2:
Multimedia libraries provide an easy way for developers to avoid writing and testing low-level code for games, and to take advantage of the clever work of others to focus on what matters most for their own programs. However, all libraries and resources from other sources need to be understood and learnt for use to make the best long-term use of them. We can also learn a lot about how other developers think about multimedia concepts like images, sounds, video, files, threads, network connections, input devices and much more.

Lab Notes – Questions!
To increase your knowledge of SDL2, answer the following questions in your Lab Notes. You might need to answer some of these after you successfully create a working demo program.

Some *very* basic warm up questions …

1. What is the official URL for the SDL2 website?
2. What does “SDL” stand for?
3. What license is SDL2 released under?
4. What platforms does SDL2 support?
5. What language and standard (version) is SDL2 written in?
6. Where (URL) can you search the SDL API by Name for particular function or type details?

Some medium level questions:

7. SDL can be downloaded as “Source Code”, “Runtime Binaries” or “Development Libraries”. What is the difference?
8. For the different download options, which one do you personally want to use with your IDE setup? (If you use a different way to setup SDL, such as a package manager, state what you have used or will use.)
9. In simple terms, what are some of the differences between a multimedia library like SDL and a “game framework”? Think about what SDL is trying to provide compared to what a game framework tries to provide.

Answer these questions after you have completed the code demo;

10. For your particular IDE and setup, what settings are needed to make your project work with SDL2?
    - There will need to be libraries/references set. Make a clear note of what and how.
    - Even if you use a pre-setup SDL2 project, inspect the settings/properties and note where the library and/or reference details are set, and also check for relevant complier/linker options.
11. What tutorial(s)/internet resource(s) did you find most useful when creating your demo program?

Code Demo (the "Hello World" of SDL2):

Create a simple C++ SDL2 demonstration application that:

- uses SDL2 to create a window of size 600 wide by 800 high,
- sets the background colour of the window initially to green,
- will change the background to a new random colour when "r" is pressed, and
- closes (quits completely) in response to an exit message (provided via SDL).

Make sure you know how to shut-down SDL properly.

Lab Notes – Code Demo!

Include the essential parts of the code (or all if it's compact) in your lab notes document (a screenshot works for nice formatting), and a screenshot of the SDL window as evidence of the code working.

Planning Notes:

- Find a basic tutorial on SDL, read it and then recreate it.
- As already noted in the questions, you will need to first install SDL2 for your IDE and setup. Then you will need to make sure that your project knows where the lib and header files are, and also that your running application has a copy of the required SDL dll's so that it can load them at run time.
- Because you need to respond to events (provided to you by the SDL event system), there will need to be a loop of some kind.

Tip: Remember to tidy up/clean your code when you finish the lab, and commit that code to your repo along with your lab notes document.