Task 15.P

1. The URL for SDL2 is: https://wiki.libsdl.org/SDL2/FrontPage
2. Simple DIrectMedia Layer 2.0.
3. The zlib license.
4. Windows, Mac OS X, iOS and Android, Linux/Unix, etc. (Oh my god, it supports Ouya).
5. Written in C, unsure about standard, but will assume C17?
6. https://wiki.libsdl.org/SDL2/CategoryAPI
7. The Source Code is the entire uncompiled code. Runtime libraries are the code compiled so that the necessary functions can be accessed during the running of a program, and the Development Libraries are libraries compiled in a way that works with IDEs and the functions can be referenced by code.
8. There is a version set up specifically for Visual Studio that I will be using. This is due to me using Visual Studio. That is all the insight I am currently able to share on this matter.
9. SDL2 is more focused on providing basic multimedia functionality, such as image, graphics and audio, as well as input and window management. A game framework, however, would be more focused on providing game-specific functionality, such as physics, collision, asset and agent management, scene management and so forth.
10. C/C++ or VC++ Directories need to have Include Directories and Library Directories properties altered. Linker > Input > Additional Dependencies also needs to reference SDL.lib and SDLmain.lib.
11. I mainly followed this tutorial, though I also included the DLL within the debug folder too and will be doing the same for release, should I need to: https://www.youtube.com/watch?v=13AEnd8XB70

Program Demo:

```cpp
int main(int argc, char* argv[])
{
    //Init, init check, error message.
    if (SDL_Init(SDL_INIT_VIDEO) != 0)
    {
        SDL_Log("SDL init failed: %s", SDL_GetError());
        return 1;
    }

    //Window creation, window check, error message.
    SDL_Window* window = SDL_CreateWindow("Task 15", SDL_WINDOWPOS_CENTERED, SDL_WINDOWPOS_CENTERED, 800, 600, SDL_WINDOW_SHOWN);
    if (!window)
    {
        SDL_Log("Window init failed: %s", SDL_GetError());
        SDL_Quit();
        return 2;
    }

    //Renderer creation, check and error message.
    SDL_Renderer* renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED);
    if (!window)
    {
        SDL_Log("Renderer init failed: %s", SDL_GetError());
        SDL_DestroyWindow(window);
        SDL_Quit();
        return 3;
    }

    bool running = true;
    SDL_Event event;

    //Set screen to green.
    SDL_SetRenderDrawColor(renderer, 0, 128, 0, 255);

    //While open, check if I've quit, then render.
    while (running)
    {
        while (SDL_PollEvent(&event))
        {
            if (event.type == SDL_QUIT)
                running = false;

            if (event.type == SDL_KEYDOWN && event.key.keysym.sym == SDLK_r)
            {
                SDL_SetRenderDrawColor(renderer, rand() % 256, rand() % 256, rand() % 256, 255);
            }
        }

        SDL_RenderClear(renderer);
        SDL_RenderPresent(renderer);
    }

    SDL_DestroyRenderer(renderer);
    SDL_DestroyWindow(window);
    SDL_Quit();
```