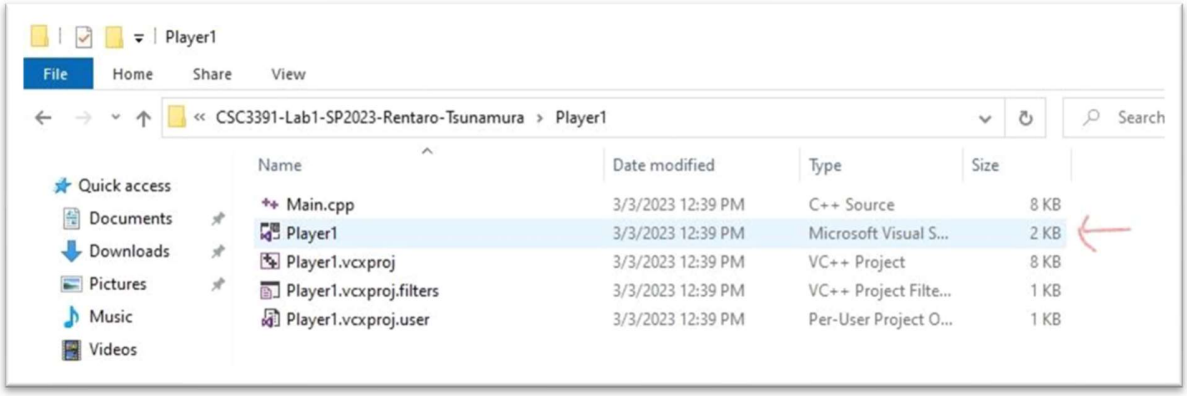
Rentaro Tsunamura
Dr. Bahram Khailili
CSC 3391
4 Mar 2023

Lab1 Hex Game

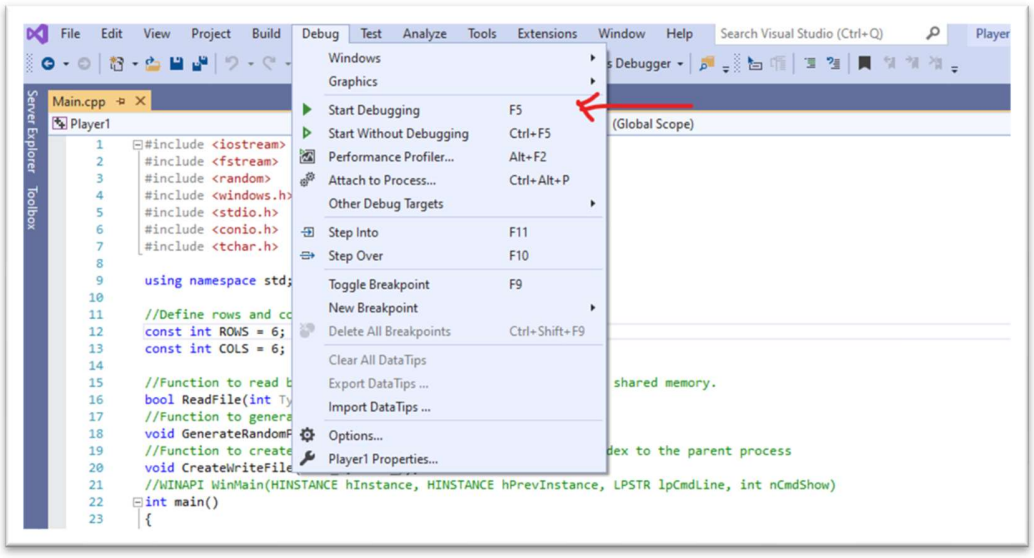
In this Lab Assignment, I created the Hex Game using C/C++. If you want to play this game, you need to have Visual Studio 2019 or Visual Studio 2022, Integrated Development Environment (IDE). In addition, you need to download a library called DX Library to run my code. This library was created in Japan and is only available in the Japanese language, so I will explain in detail how to install it later. There are several steps required to set up the environment for this game, so please follow the instructions below.

Set up

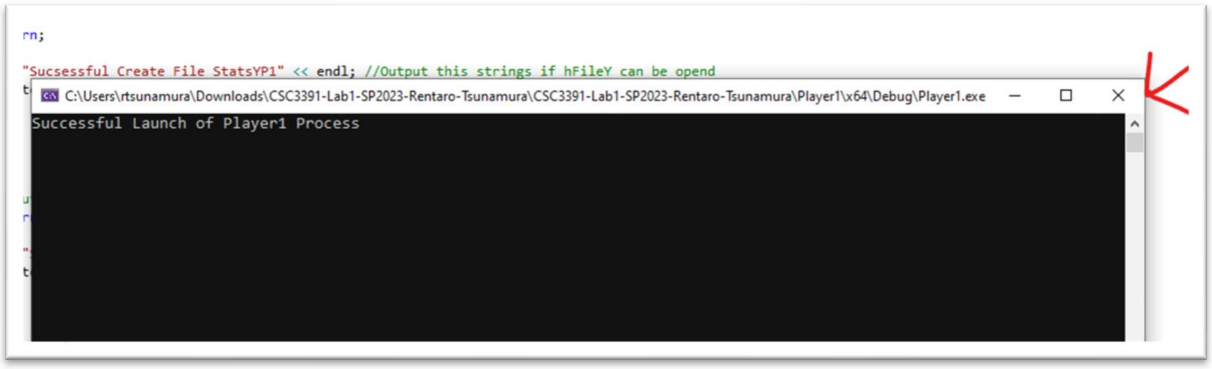
1. First, please extract my game file CSC3391-Lab1-SP2023-Rentaro-Tsunamura in any way you want. After unzipping, select the file CSC3391-Lab1-SP2023-Rentaro-Tsunamura->Player1 and open Player1 in Visual Studio. Please refer to the image below.



2. Once Visual Studio is open, select Debug->Start Debugging and run Player1.



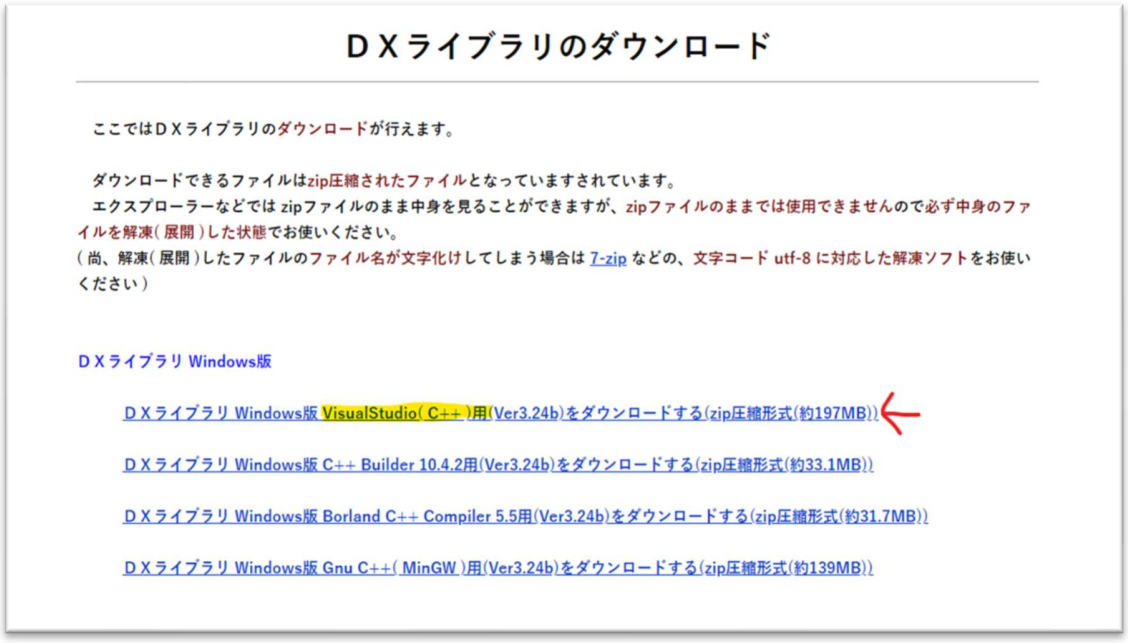
3. If the execution is successful, you will see the following text. Then press the X in the upper right corner of the console window to terminate the execution.



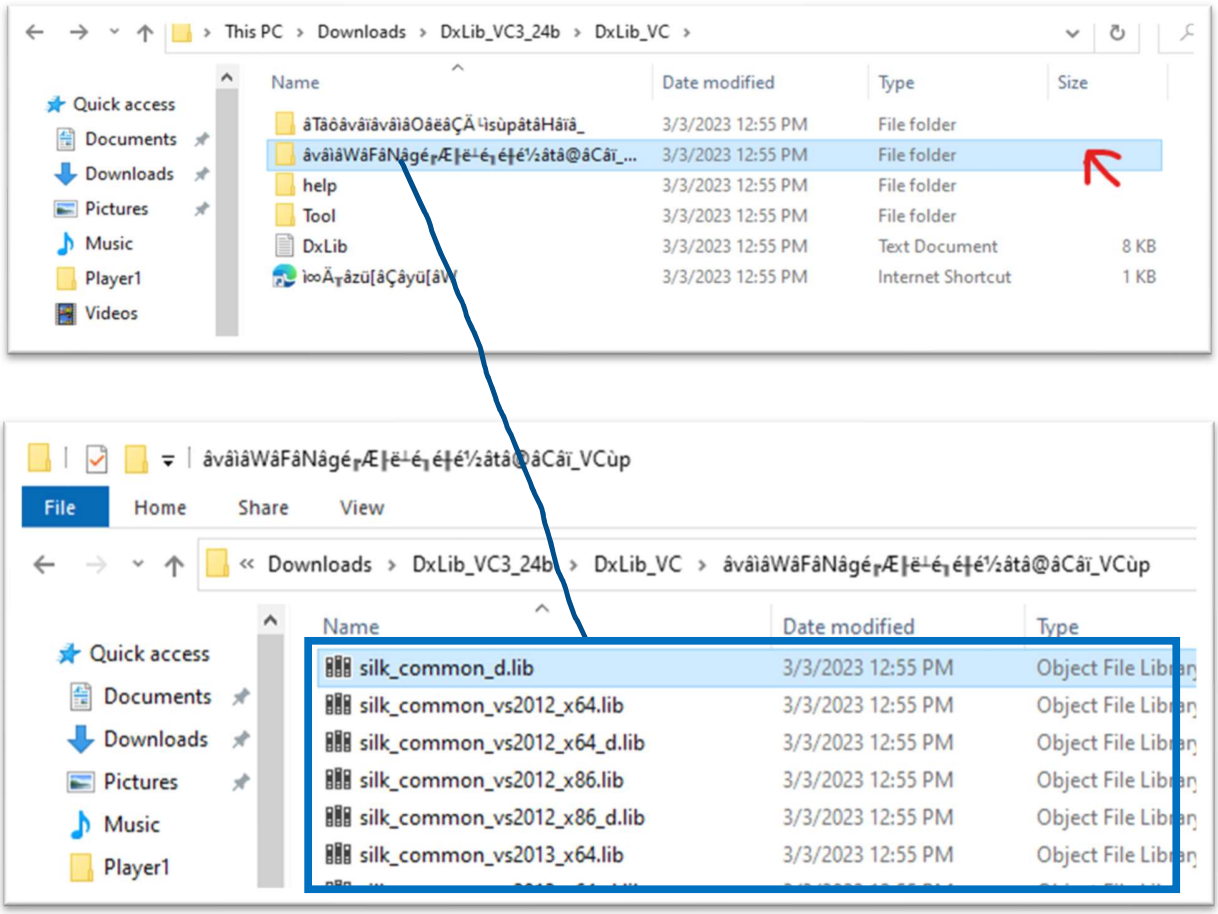
4. Repeat the same process with the Player2 folder.

5. After Player1 and Player2 are completed, go to the following site to download the DX library.
Link: https://dxlib.xsrv.jp/dxdload.html

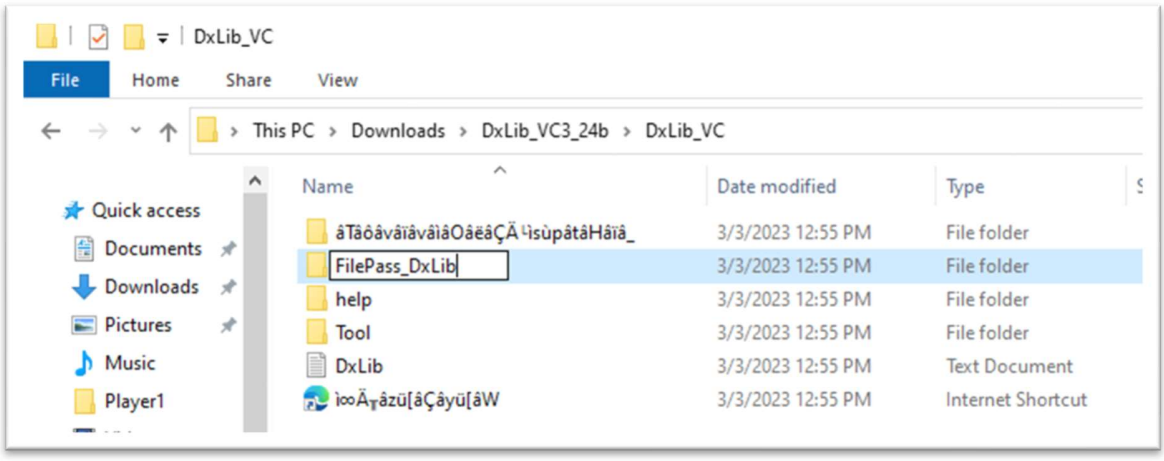
6. Download the file in the image below from the website that you jumped from the link in the previous section. The landmarks are the words" Windows" and "VisualStudio (C++)." You can download it by clicking.



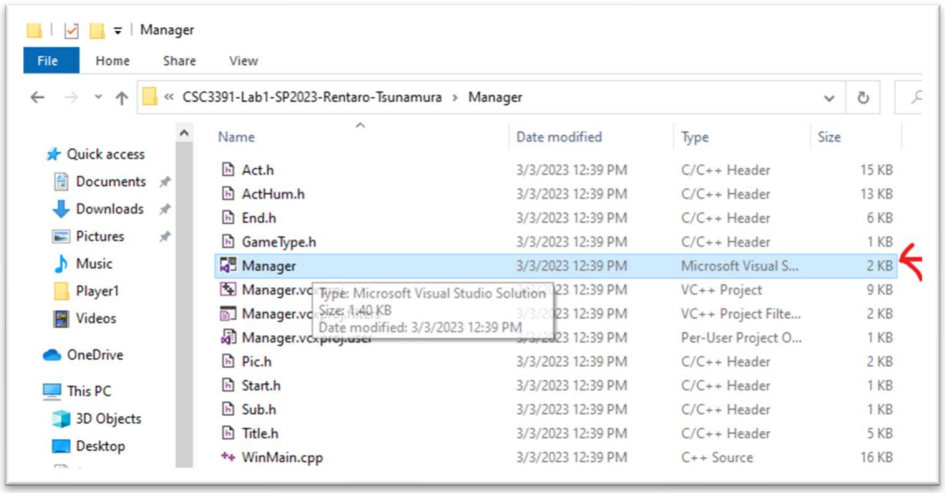
7. After downloading the DxLib_VC3_24 zip1 file, extract the zip file. If you go to DxLib_VC3_24b-> Dxlib_VC you will see some broken characters like below image. Find one of the two garbled files that contains more ".lib"



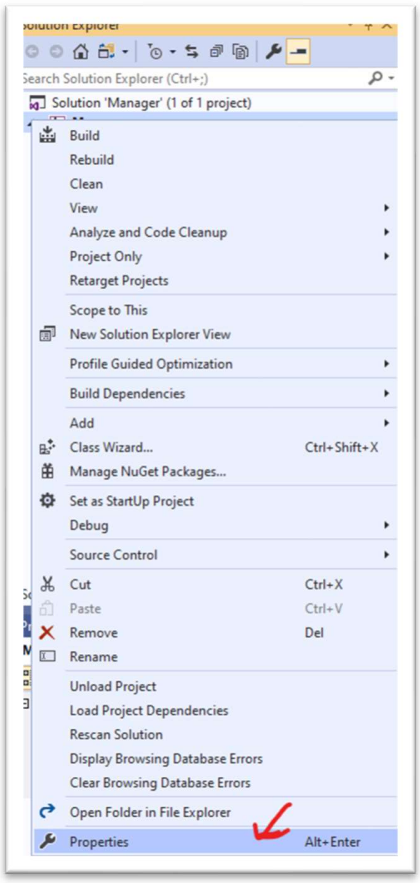
8. Set the file found in step 7 to "FilePass_DxLib" for clarity.



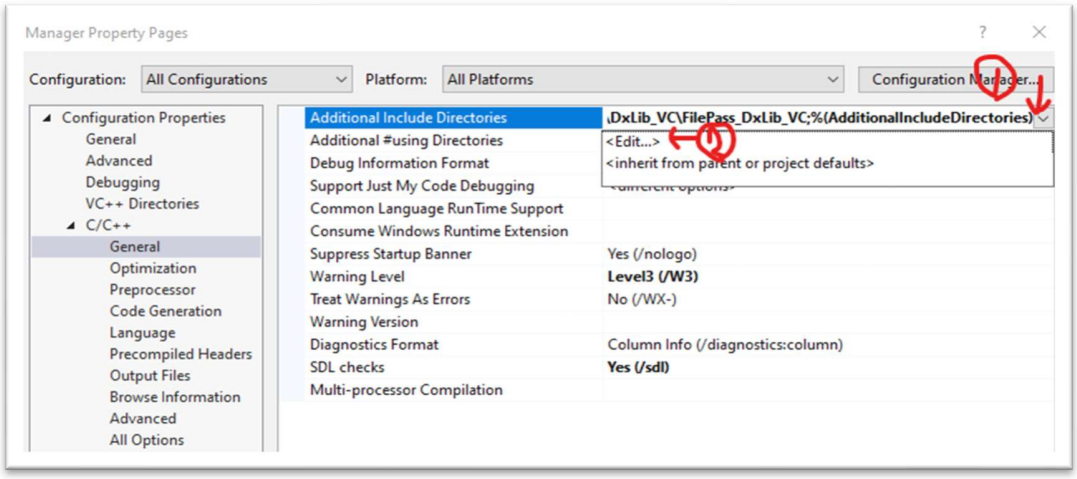
9. Go to 3391-Lab1-SP2023-Rentaro-Tsunamura->Manager in the first game file you downloaded. Then launch the Manager project in Visual Studio.



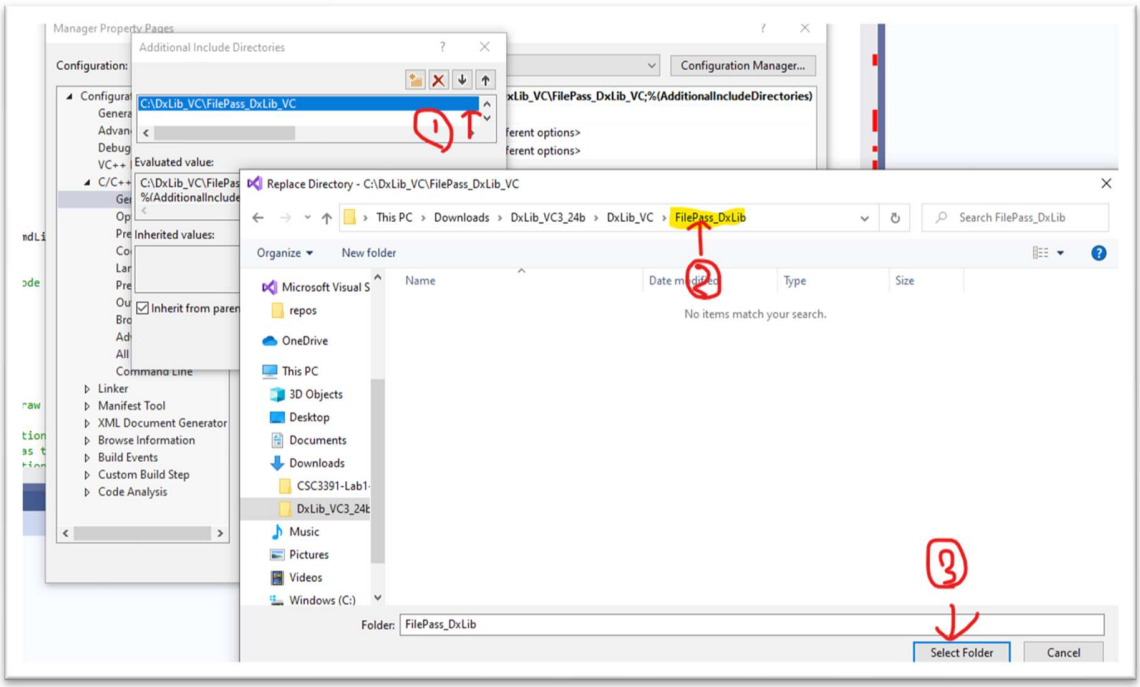
10. We will set up the library. First select Solution Manager->Properties.



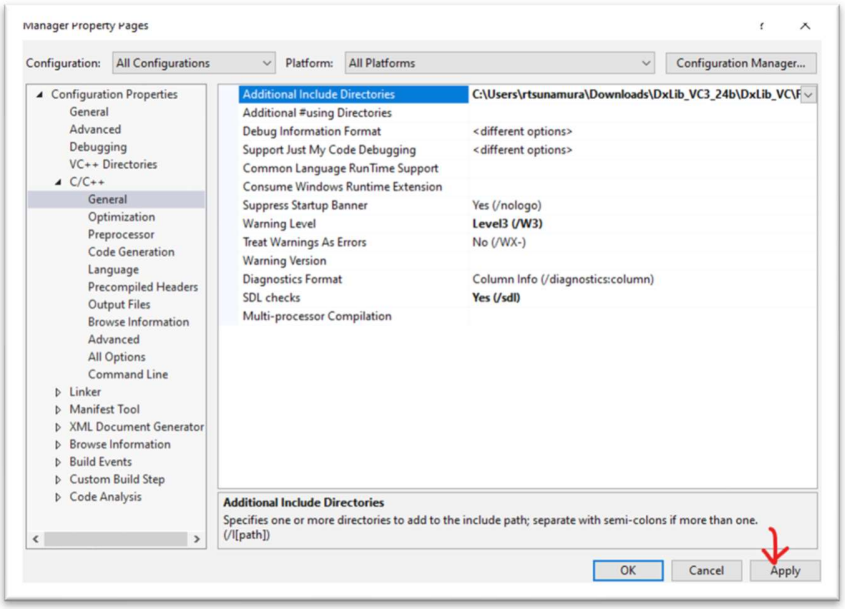
11. Select Configuration Properties->C/C++->General. In General, click on the right edge of Additional Include Directories and select <Edit>.



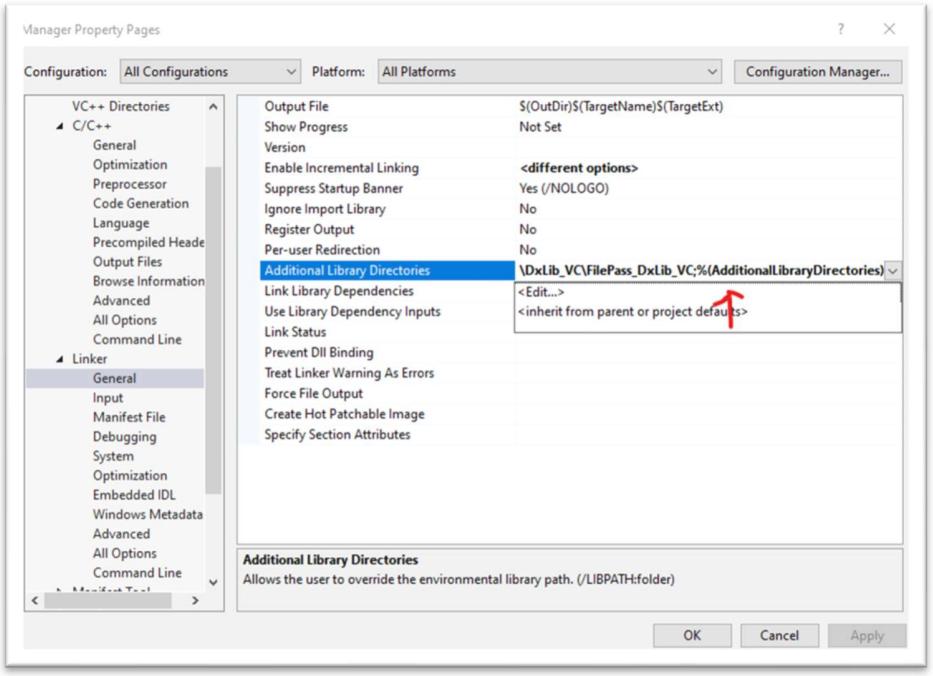
12. Now add the file path of the library you just downloaded to the Additional Include Directory. The file to add is DxLib_VC3_24b-> Dxlib_VC->FilePass_DxLib.



13. After setting up the library, press “Apply”.



14. Next, go to Configuration Properties->Linker->General. Then, click on the right edge of Additional Library Directories and select <Edit>. After that, as before, set the file path for the additional library and select "Apply" at the end.



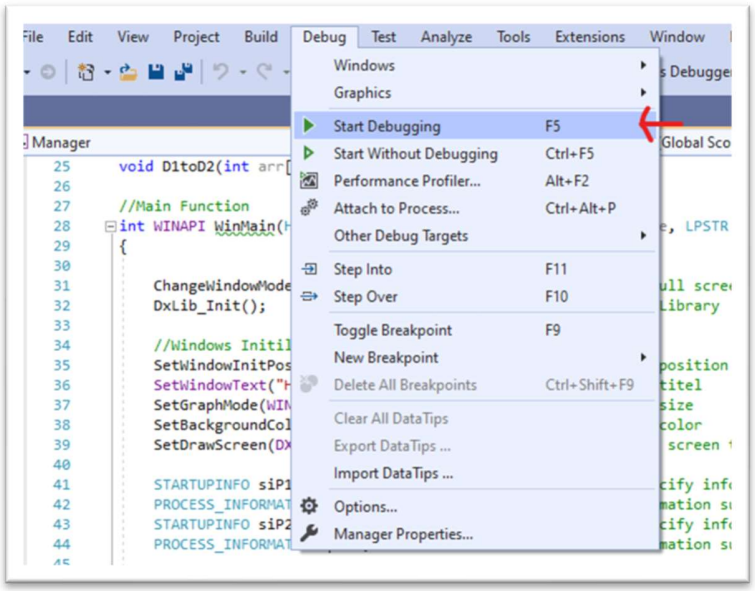
15. Check the location in the middle of WinMain.cpp where other processes should be run. You probably do not need to change it but check it just in case. If it is wrong, set the locations of the "Player1.exe" and "Player2.exe" files correctly.

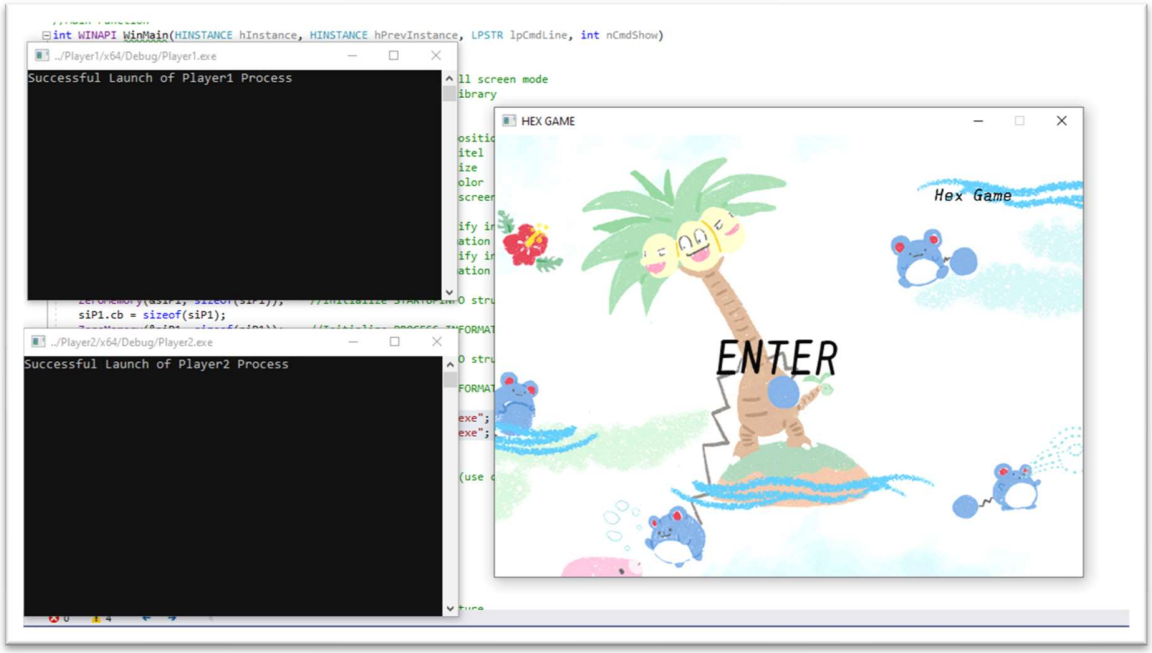
```
//Main Function
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
{

    ChangeWindowMode(TRUE);                    //Switch to full screen mode
    DxLib_Init();                              //Initialize Library

    //Windows Initilize
    SetWindowInitPosition(WIN_POS_X, WIN_POS_Y); //Background position
    SetWindowText("HEX GAME");                 //Background titel
    SetGraphMode(WIN_MAX_X, WIN_POS_Y, 32);    //Background size
    SetBackgroundColor(255, 255, 255);         //Background color
    SetDrawScreen(DX_SCREEN_BACK);             //Selects the screen to draw on

    STARTUPINFO siP1;                          //Used to specify information for creating a process
    PROCESS_INFORMATION piP1;                  //Holds information such as thread handles and thread IDs.
    STARTUPINFO siP2;                          //Used to specify information for creating a process
    PROCESS_INFORMATION piP2;                  //Holds information such as thread handles and thread IDs.

    ZeroMemory(&siP1, sizeof(siP1));    //Initialize STARTUPINFO structure
    siP1.cb = sizeof(siP1);
    ZeroMemory(&piP1, sizeof(piP1));    //Initialize PROCESS_INFORMATION structure

    ZeroMemory(&siP2, sizeof(siP2));    //Initialize STARTUPINFO structure
    siP1.cb = sizeof(siP1);
    ZeroMemory(&piP2, sizeof(piP2));    //Initialize PROCESS_INFORMATION structure

    LPCTSTR lpApplicationName1 = "../Player1/x64/Debug/Player1.exe";  //An executable file to start a new process
    LPCTSTR lpApplicationName2 = "../Player2/x64/Debug/Player2.exe";  //An executable file to start a new process
```

16. If the above is wrong, then the location where the file is created in the CreataeWriteFile function in Main.cpp of Plyaer1 and Player2 may also be wrong. Rewrite it correctly in the location where WinMain.cpp and other headers in Manager are located.

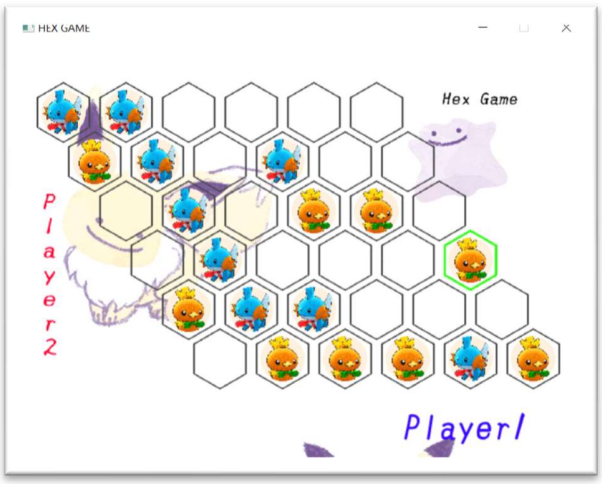
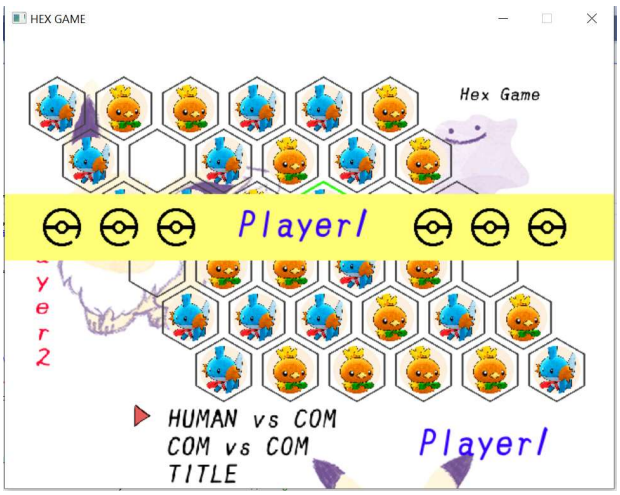
```
/Function to create and write a file to send the generated index to th
oid CreateWriteFile(int _x, int _y) {
    DWORD dwBytesWritten;
    LPCSTR  lpFileName1 = "../Manager/StatsXP1.txt";  //Specify the loc
    LPCSTR  lpFileName2 = "../Manager/StatsYP1.txt";  //that will write
    //Declare variables to send data
    char charData1[16];
    char charData2[16];
    //Convert data type from int to char
    sprintf_s(charData1, "%d", _x);
    sprintf_s(charData2, "%d", _y);
    HANDLE hFileX = CreateFileA(lpFileName1,           //The name of th
                                GENERIC_WRITE,         //The requested
```

17. Debugging and execution of the Manager project to launch the Manager process. Finally, when the two processes and the graphics of the game are launched, you have succeeded. If you have some trouble to set up the library or something, contact me rtsunamura@txwes.edu. Thanks for the long setup.





Manipulate

	This is the title screen. The user can select the type of game (Computer vs Computer or Human vs Computer) or exit here. The cursor can be moved by pressing W or S on the keyboard. W moves the cursor up and S moves the cursor down. If you decide on the next action, set the cursor and press Enter.
	This is the game screen. Player 1 connects pieces from top to bottom or bottom to top, and Player 2 connects pieces from right to left or left to right. In a Human vs. Computer game, Player 1 is you and must move the cursor. The cursor can be moved right with the D button, left with the A button, up with the W button, and down with the S button. If you decide where you want to place a piece, press Enter.
	After the game is over, the winner is displayed. You can choose to play the game again or return to the title. If you want to end the game, return to the title once and then select Exit. W moves the cursor up and S moves the cursor down. If you decide on the next action, set the cursor and press Enter.

Bugs

There are no specific bugs in the game. However, a bug that will be expected in the future is that the processing speed will slow down as the number of games is played without Exit. I think that launching three processes at the same time is not a small burden for a PC. It is also not impossible that the game may not run if another program is using the shared memory first.

Limitations

The limitation of this game or this project is that it is not possible to decide whether Player1 or Player2 places the piece first. Player1 must always place piece first. Selections can only be made with the keyboard and not with the mouse cursor.

Reference

Code	Link
The basic implementation of WinMain.cpp, Pic.h, Act.h, ActHum.h, and Sub.h in my Manager Process was based on the code in this YouTube video.	https://www.youtube.com/watch?v=Gs-UiGJ2eWc
I referred to this site to create the Player1 or Player2 process in WinMain.cpp.	https://learn.microsoft.com/en-us/windows/win32/procthread/creating-processes --- https://stackoverflow.com/questions/42531/how-do-i-call-createprocess-in-c-to-launch-a-windows-executable
I referred to this website to terminate the child process stop in WinMain.cpp.	https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-terminateprocess
I referred to this website to create the shared memory in WinMain.cpp in Manager Project and Main.cpp in Player1 and Player2 Project.	https://learn.microsoft.com/en-us/windows/win32/memory/creating-named-shared-memory --- https://stackoverflow.com/questions/15461478/how-to-use-shared-memory-in-windows ---

	http://web-ext.u-aizu.ac.jp/course/osrtk/exercise/ex/ex07/ex07.html --- windows - C++ Read from shared memory - Stack Overflow --- https://stackoverflow.com/questions/2681061/memcpy-what-should-the-value-of-the-size-parameter-be --- https://www.geeksforgeeks.org/difference-between-sizeofint-and-sizeofint-in-c-c/
I referred to this site to create files, read files, and write files.	http://www.cs.rpi.edu/courses/fall01/os/CreateFile.html --- https://learn.microsoft.com/en-us/windows/win32/fileio/opening-a-file-for-reading-or-writing --- https://stackoverflow.com/questions/4010709/readfile-win32-api --- C++ Program For char to int Conversion - GeeksforGeeks --- http://winapi.freetechsecrets.com/win32/WIN32CreateFileMapping.htm --- https://learn.microsoft.com/en-us/windows/win32/api/memoryapi/nf-memoryapi-mapviewoffile --- https://learn.microsoft.com/en-us/cpp/c-runtime-library/reference/sprintf-s-sprintf-s-l-swprintf-s-swprintf-s-l?view=msvc-170
I referred to this website to generate random values from 0 to 5 in the Player1 and Player2 projects.	https://cplusplus.com/forum/beginner/250575/

Summary

I had never made a game before, so this assignment was very difficult for me. It was especially hard when I implemented InterProcess Communication using shared memory, which many times did not execute well. However, I was happy when the game was finally completed. My implementation in this assignment is to create three processes, Manager, Player1, and Player2 processes. The Manager sends the board data to Player1 or Player2 using shared memory. Shared memory is created only during each player's turn, and child processes cannot access it during another player's turn. The child process randomly generates an index of locations where no pieces have been placed yet from the board information and sends index to the parent process. At that time, a file is created, and the index information is written to the file. At the end of the game, the Player process is terminated first, and then the Manager Process is terminated. I also put a lot of effort into the graphics and used a library called the DX library. This library is often used to create games, and it is very easy to create a GUI. lease enjoy the game!