**Final Code**

#include <FEHLCD.h>

#include <FEHIO.h>

#include <FEHUtility.h>

#include<LCDColors.h>

class Connect4 //declare class

{

public:

Connect4(); //constructor function

void BlankBoard(); //resets board to initial blank state

void PlayerBlue(); //handles blue player input and blue token placement

void PlayerRed(); //handles red player input and blue token placement

void WinState(); //checks to see if either player won

void GameLoop(); //loops for one iteration of the game

void Menu\_Function(); //Displays Menu, gives option for start or instructions

//and goes to blankboard

void Display\_Menu(); //Displays menu

void Check\_Touch(); //Checks if screen tapped AND Stores coordinates of tap

void Menu\_Button\_Check(); //Checks if coordinates of tap are on start or instructions button

void Display\_Instructions(); //Displays instructions

void Blank\_Board(); //Displays blank board

void Column\_Check(); //Check what column tap is in

void Instructions\_Button(); //what happens when instructions is clicked

void Slot\_Centers(); //calculates the center of each slot

void Find\_Slot(); //Find an open space in chosen column

void Fill\_Slot(); //Fill the lowest open space with a circle

//Change the corresponding array value

void BoardClear(); //Clears board array to reset game

void CreditDisplay(); //Displays credit screen after game

void DisplayStats(); //Displays statistics screen when clicked

private:

int board[5][5]; //declare virtual board array

int n; //looping variable n

//location of tap

float tapx;

float tapy;

//Button chosen on menu: 1 = start, 2 = instructions

int menu\_choice;

//Column chosen on board

int column\_index;

//arrays for centers of boxes

int x\_slot\_center[5];

int y\_slot\_center[5];

int row\_index;

int token\_type;

int red\_win\_count;

int blue\_win\_count;

};

Connect4 game; //declare object

int main(void) //main function

{

//Infinite while loop to keep game going

int TRUE=1;

while (TRUE) **//WHILE LOOP**

{

game.Slot\_Centers();

game.Menu\_Function(); //displays main menu

// while game

game.GameLoop(); //loop through player turns

game.CreditDisplay();

game.BoardClear();

}//end of infinite loop

}

Connect4::Connect4() //initializing constructor function

{

int i,j;

for (i=0;i<=4;i++) **//FOR LOOP**

{

for (j=0;j<=4;j++)

{

board[i][j]=0; //sets every array index to 0

}

}

//sets all class variables to 0

n=0;

tapx=0;

tapy=0;

menu\_choice=0;

token\_type=0;

blue\_win\_count = 0;

red\_win\_count = 0;

}

void Connect4::Slot\_Centers()

{

int i;

//Loop to calculate pixel centers of each slot on gameboard

//stores in array

for (i = 0; i < 5; i++)

{

x\_slot\_center[i] = 120 + i\*40;

y\_slot\_center[i] = 45 + i\*40;

}

}

void Connect4::Menu\_Function()

{

game.Display\_Menu(); //Display menu screen

//Loop to check which menu button is pressed and decides

//what screen to go to

int menu\_loop=1;

do **//DO WHILE LOOP**

{

game.Check\_Touch(); //Wait for tap

game.Menu\_Button\_Check(); //Check which button tapped

switch (menu\_choice) **//SWITCH CASE**

{

case 1:

game.BlankBoard(); //Display Blank Board

menu\_loop = 0; //End menu loop

break;

case 2:

game.Instructions\_Button(); //Display instructions and wait for tap

menu\_loop = 0;

break;

case 3: //Display stats and go back to menu

game.DisplayStats();

menu\_loop = 1;

default:

menu\_loop = 1; //Continue menu loop

break;

}

} while (menu\_loop==1);

}

void Connect4::Display\_Menu()

{

LCD.SetBackgroundColor(SPRINGGREEN); //Background spring green

LCD.Clear();

LCD.SetFontColor(RED); //Red Quad

LCD.WriteAt("Quad",80,40);

LCD.SetFontColor(BLUE); //Blue Connection

LCD.WriteAt("Connection",130,40);

LCD.SetFontColor(RED); //Red Start option

LCD.WriteAt("Start",130,100);

LCD.DrawRectangle(125,95,70,25);

LCD.SetFontColor(BLUE); //Blue Instructions option

LCD.WriteAt("Instructions",100,140);

LCD.DrawRectangle(95,135,150,25);

LCD.SetFontColor(BLUE);

LCD.DrawRectangle(1,215,65,25);

LCD.WriteAt("Stats",5,220); //Blue Stats option

LCD.SetFontColor(BLACK);

LCD.WriteAt("Written by Jake K.",70,200); //Black authors

LCD.WriteAt("& Michael A.",110,215);

}

void Connect4::Check\_Touch()

{

//Check if player taps screen and wait

float x, y;

while (!LCD.Touch(&x,&y)) {}

//stores in tap variables

while(LCD.Touch(&x,&y)){ tapx = x;

tapy = y;}

}

void Connect4::Menu\_Button\_Check()

{

//Check if start pressed

if ((tapy >= 95 && tapy <=125)&&(tapx >= 115 && tapx <=195))

{

menu\_choice = 1; //Start

}

//Check if instructions pressed

else if ((tapy >= 135 && tapy <=160)&&(tapx >= 95 && tapx <=245))

{

menu\_choice = 2; //Instructions

}

//Check if stats is pressed

else if ((tapy >= 215 && tapy <=240)&&(tapx >= 1 && tapx <=66))

{

menu\_choice=3; //stats

}

else

{

menu\_choice = 0; //Jump back to CheckTouch

}

}

void Connect4::Instructions\_Button()

{

//Display instructions screen, wait for touch, then start game

game.Display\_Instructions();

game.Check\_Touch();

game.BlankBoard();

}

void Connect4::Display\_Instructions()

{

//Writes instructions prompt to screen

LCD.Clear(SPRINGGREEN);

LCD.WriteLine("How to play:");

LCD.WriteLine("Each player takes a turn ");

LCD.WriteLine("placing a token in the slots.");

LCD.WriteLine("The tokens stack on top of eachother.");

LCD.WriteLine("The first player to get 4 in a row wins.");

LCD.WriteLine("Four in a row can be vertical, horizontal, or diagonal.");

LCD.SetFontColor(BLUE);

LCD.Write("Tap Anywhere to Start Game");

}

void Connect4::Column\_Check()

{

if (tapx>= 0 && tapx <140) //**IF STATEMENT** Left part of screen up to end of first column works

{

column\_index = 0;

}

else if (tapx>= 140 && tapx < 180) //**RELATIONAL OPERATORS** second column pixels

{

column\_index = 1;

}

else if (tapx>= 180 && tapx < 220) //**LOGICAL OPERATORS** third column pixels

{

column\_index = 2;

}

else if (tapx>= 220 && tapx < 260) //fourth column pixels

{

column\_index = 3;

}

else if (tapx>= 260 && tapx <=319) //Beginning of final column to end of screen works

{

column\_index = 4;

}

else

{

column\_index = 0; //Shouldn't ever be outside of the bounds but just in case

}

}

void Connect4::BlankBoard()

{

n=0; //resets game loop variable

//draws blank board and grid to screen

LCD.Clear(SPRINGGREEN);

LCD.SetFontColor(GRAY);

LCD.FillRectangle(100,25,200,200);

LCD.SetFontColor(SCARLET);

LCD.DrawHorizontalLine(25,100,300);

LCD.DrawHorizontalLine(65,100,300);

LCD.DrawHorizontalLine(105,100,300);

LCD.DrawHorizontalLine(145,100,300);

LCD.DrawHorizontalLine(185,100,300);

LCD.DrawHorizontalLine(225,100,300);

LCD.DrawVerticalLine(100,25,225);

LCD.DrawVerticalLine(140,25,225);

LCD.DrawVerticalLine(180,25,225);

LCD.DrawVerticalLine(220,25,225);

LCD.DrawVerticalLine(260,25,225);

LCD.DrawVerticalLine(300,25,225);

}

void Connect4::GameLoop()

{

while (n==0) //while loop for game

{

game.PlayerBlue(); //blue turn

game.WinState(); //check for win

if (n==0)

{

game.PlayerRed(); //red turn

game.WinState(); //check for win

}

}

}

void Connect4::PlayerBlue()

{

LCD.SetFontColor(BLACK);

LCD.WriteAt("Blue Turn",100,5); //write blue turn to top of screen

token\_type=1;

Check\_Touch(); //check for touch input

Column\_Check(); //find what column was touched

Find\_Slot(); //find what slot it corresponds to

LCD.SetFontColor(BLUE);

Fill\_Slot(); //fill slot and array index with appropriate token

}

void Connect4::PlayerRed()

{

LCD.SetFontColor(BLACK);

LCD.WriteAt("Red Turn ",100,5); //write red turn to top of screen

token\_type=2;

Check\_Touch(); //check for touch input

Column\_Check(); //find what column was touched

Find\_Slot(); //find what slot it corresponds to

LCD.SetFontColor(RED);

Fill\_Slot(); //fill slot and array index with appropriate token

}

void Connect4::WinState()

{

int i,j,count;

count=0;

n=0;

for (i=0;i<=4;i++) //player 1 horizontals and verticals

{

if (board[i][0]==1 && board[i][1]==1 && board[i][2]==1 && board[i][3]==1)

{

n=1;

}

else if (board[i][1]==1 && board[i][2]==1 && board[i][3]==1 && board[i][4]==1)

{

n=1;

}

else if (board[0][i]==1 && board[1][i]==1 && board[2][i]==1 && board[3][i]==1)

{

n=1;

}

else if (board[1][i]==1 && board[2][i]==1 && board[3][i]==1 && board[4][i]==1)

{

n=1;

}

}

for (i=0;i<=4;i++) //player 2 horizontals and verticals

{

if (board[i][0]==2 && board[i][1]==2 && board[i][2]==2 && board[i][3]==2)

{

n=2;

}

else if (board[i][1]==2 && board[i][2]==2 && board[i][3]==2 && board[i][4]==2)

{

n=2;

}

else if (board[0][i]==2 && board[1][i]==2 && board[2][i]==2 && board[3][i]==2)

{

n=2;

}

else if (board[1][i]==2 && board[2][i]==2 && board[3][i]==2 && board[4][i]==2)

{

n=2;

}

}

for (i=0;i<=1;i++) //player 1 main diagonals

{

if (board[i][i]==1 && board[i+1][i+1]==1 && board[i+2][i+2]==1 && board[i+3][i+3]==1)

{

n=1;

}

else if (board[i][4-i]==1 && board[i+1][3-i]==1 && board[i+2][2-i]==1 && board[i+3][1-i]==1)

{

n=1;

}

}

for (i=0;i<=1;i++) //player 2 main diagonals

{

if (board[i][i]==2 && board[i+1][i+1]==2 && board[i+2][i+2]==2 && board[i+3][i+3]==2)

{

n=2;

}

else if (board[i][4-i]==2 && board[i+1][3-i]==2 && board[i+2][2-i]==2 && board[i+3][1-i]==2)

{

n=2;

}

}

for (i=0;i<=1;i++) //player 1 minor diagonals

{

if (board[i][1-i]==1 && board[i+1][2-i]==1 && board[i+2][3-i]==1 && board[i+3][4-i]==1)

{

n=1;

}

else if (board[i][3+i]==1 && board[i+1][2+i]==1 && board[i+2][1+i]==1 && board[i+3][i]==1)

{

n=1;

}

}

for (i=0;i<=1;i++) //player 2 minor diagonals

{

if (board[i][1-i]==2 && board[i+1][2-i]==2 && board[i+2][3-i]==2 && board[i+3][4-i]==2)

{

n=2;

}

else if (board[i][3+i]==2 && board[i+1][2+i]==2 && board[i+2][1+i]==2 && board[i+3][i]==2)

{

n=2;

}

}

for (i=0;i<=4;i++) //check for full board

{

for (j=0;j<=4;j++)

{

if (board[i][j]==0)

{

count+=1;

}

}

}

if (count==0) //if full board

{

n=3;

}

if (n==1) //if blue winner

{

//display blue winner to screen

LCD.WriteAt("Blue Winner!",100,5);

LCD.WriteLine(" ");

LCD.WriteLine("BLUE!!!!");

LCD.WriteLine("BLUE!!!!");

LCD.WriteLine("BLUE!!!!");

LCD.WriteLine("BLUE!!!!");

LCD.WriteLine("BLUE!!!!");

LCD.WriteLine("BLUE!!!!");

LCD.WriteLine("BLUE!!!!");

LCD.WriteLine("BLUE!!!!");

LCD.WriteLine("BLUE!!!!");

LCD.WriteLine("BLUE!!!!");

LCD.WriteLine("BLUE!!!!");

//Add to appropriate statistics

blue\_win\_count += 1;

Sleep(4000);

}

if (n==2) //if red winner

{

//display red winner to screen

LCD.WriteAt("Red Winner!",100,5);

LCD.WriteLine(" ");

LCD.WriteLine("RED!!!!");

LCD.WriteLine("RED!!!!");

LCD.WriteLine("RED!!!!");

LCD.WriteLine("RED!!!!");

LCD.WriteLine("RED!!!!");

LCD.WriteLine("RED!!!!");

LCD.WriteLine("RED!!!!");

LCD.WriteLine("RED!!!!");

LCD.WriteLine("RED!!!!");

LCD.WriteLine("RED!!!!");

//Add to appropriate statistics

red\_win\_count += 1;

Sleep(4000);

}

if (n==3) //if tie

{

//display tie to screen

LCD.SetFontColor(BLACK);

LCD.WriteAt("TIE GAME!",100,5);

LCD.WriteLine(" ");

LCD.WriteLine("TIE!!!!!");

LCD.WriteLine("TIE!!!!!");

LCD.WriteLine("TIE!!!!!");

LCD.WriteLine("TIE!!!!!");

LCD.WriteLine("TIE!!!!!");

LCD.WriteLine("TIE!!!!!");

LCD.WriteLine("TIE!!!!!");

LCD.WriteLine("TIE!!!!!");

LCD.WriteLine("TIE!!!!!");

LCD.WriteLine("TIE!!!!!");

Sleep(4000);

}

}

void Connect4::Find\_Slot()

{

int i;

for (i = 4; i >=0; i--) //loop through chosen column rows

{

if (board[i][column\_index] == 0) //if empty

{

row\_index = i;

Fill\_Slot(); //fill slot

break;

}

}

}

void Connect4::Fill\_Slot()

{

//draw circle in empty slot

LCD.FillCircle(x\_slot\_center[column\_index], y\_slot\_center[row\_index], 20);

//fill array with appropriate token

board[row\_index][column\_index]=token\_type;

}

void Connect4::BoardClear()

{

int i,j;

for (i=0;i<=4;i++)

{

for (j=0;j<=4;j++)

{

board[i][j]=0; //sets every array index to 0

}

}

}

void Connect4::CreditDisplay()

{

//writes credits to screen

LCD.Clear(SPRINGGREEN);

LCD.SetFontColor(BLACK);

LCD.WriteLine("Thank you for playing!");

LCD.WriteLine("We hope you play again,");

LCD.WriteLine("-Michael and Jake");

LCD.WriteLine(" ");

LCD.WriteLine("Instructor: Phil Schlosser");

LCD.WriteLine(" ");

LCD.WriteLine("TAs: Alex Jacobs");

LCD.WriteLine(" Peter Schmitz");

LCD.WriteLine(" ");

LCD.WriteLine("Reference: u.osu.edu/fehproteus");

Sleep(3000);

}

void Connect4::DisplayStats()

{

//writes statistics to screen

LCD.Clear(SPRINGGREEN);

LCD.SetFontColor(BLUE);

LCD.Write("Blue Wins:");

LCD.Write(blue\_win\_count);

LCD.WriteLine(" ");

LCD.SetFontColor(RED);

LCD.Write("Red Wins:");

LCD.Write(red\_win\_count);

LCD.SetFontColor(BLUE);

LCD.WriteAt("Tap Anywhere to",70,150);

LCD.WriteAt("Return to Main Menu.",60,165);

game.Check\_Touch(); //checks for touch

game.Display\_Menu(); //goes back to menu

}