//Creating cell, and index for each cell in win32Project

BOOL GetGameBoard(HWND hWnd, RECT \* pRect)

{

RECT rc;

if (GetClientRect(hWnd, &rc))

{

int width = rc.right - rc.left;

int height = rc.bottom - rc.top;

pRect->left = (width - CELL\_SIZE \* 5) / 2;

pRect->top = (height - CELL\_SIZE \* 5) / 2;

pRect->right = (pRect->left + CELL\_SIZE \* 5);

pRect->bottom = (pRect->top + CELL\_SIZE \* 5);

return TRUE;

}

SetRectEmpty(pRect);

return FALSE;

}

void Drawlines(HDC hdc, int x1, int y1, int x2, int y2)

{

MoveToEx(hdc, x1, y1, NULL);

LineTo(hdc, x2, y2);

}

int GetCellNumberFromPoint(HWND hwnd, int x, int y)

{

POINT pt = { x, y };

RECT rc;

if (GetGameBoard(hwnd, &rc))

{

if (PtInRect(&rc, pt))

{

//use clicks in the board

x = pt.x - rc.left;

y = pt.y - rc.top;

int column = x / CELL\_SIZE;

int row = y / CELL\_SIZE;

return (column + row \* 5);

}

}

//outside tic-tac-toe board

return -1;

}

BOOL GetCellRect(HWND hWnd, int index, RECT \*pRect)

{

RECT rcBoard;

SetRectEmpty(pRect);

if (index < 0 || index > 24)

return FALSE;

if (GetGameBoard(hWnd, &rcBoard))

{

int y = index / 5;

int x = index % 5;

pRect->left = rcBoard.left + x \* CELL\_SIZE + 10;

pRect->top = rcBoard.top + y \* CELL\_SIZE + 10;

pRect->right = pRect->left + CELL\_SIZE - 20;

pRect->bottom = pRect->top + CELL\_SIZE - 20;

return TRUE;

}

return FALSE;

}

//Creating gameboard

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hWnd, &ps);

RECT rc;

if (GetGameBoard(hWnd, &rc))

{

RECT rcClient;

if (GetClientRect(hWnd, &rcClient))

{

if (popup == 0)

{

int ret = MessageBox(hWnd, L"Do you want to go first?", L"Fisrt Move", MB\_YESNO | MB\_ICONQUESTION);

if (IDYES == ret)

{

playerTurn = 1;

}

else if (IDNO == ret)

{

if (mode == 0)

playerTurn = 2;

if (mode > 0)

playerTurn = 4;

}

popup++;

}

//display Player's name

SetBkMode(hdc, TRANSPARENT);

if (mode > 0)

{

SetTextColor(hdc, RGB(255, 0, 0));

TextOut(hdc, 18, 20, szPlayer1, ARRAYSIZE(szPlayer1));

SetTextColor(hdc, RGB(0, 255, 0));

TextOut(hdc, rcClient.right - 80, 20, szPlayer3, ARRAYSIZE(szPlayer3));

}

if (mode == 0)

{

SetTextColor(hdc, RGB(255, 0, 0));

TextOut(hdc, 18, 20, szPlayer1, ARRAYSIZE(szPlayer1));

SetTextColor(hdc, RGB(0, 0, 255));

TextOut(hdc, rcClient.right - 80, 20, szPlayer2, ARRAYSIZE(szPlayer2));

}

ShowTurn(hWnd, hdc);

}

//Remove boarders

FillRect(hdc, &rc, (HBRUSH)GetStockObject(WHITE\_BRUSH));

//Rectangle(hdc, rc.left, rc.top, rc.right, rc.bottom);

}

for (int i = 1; i < 5; ++i)

{

//vertical lines

Drawlines(hdc, rc.left + CELL\_SIZE \* i, rc.top, rc.left + CELL\_SIZE \*i, rc.bottom);

//Horizantal line

Drawlines(hdc, rc.left, rc.top + CELL\_SIZE \* i, rc.right, rc.top + CELL\_SIZE \* i);

}

for (int i = 0; i < 25; i++)

{

RECT rcCell;

//action stays after re-size page

if (0 != gameBoard[i] && GetCellRect(hWnd, i, &rcCell) && mode == 0)

{

FillRect(hdc, &rcCell, (gameBoard[i] == 2) ? hbr2 : hbr1);

}

else if (0 != gameBoard[i] && GetCellRect(hWnd, i, &rcCell) && mode > 0)

{

FillRect(hdc, &rcCell, (gameBoard[i] == 4) ? hbr3 : hbr1);

}

}

// Players move

void Moves(HWND hWnd, HDC hdc, int xPos, int yPos)

{

if (mode == 0)

{

//calling function for button

index = GetCellNumberFromPoint(hWnd, xPos, yPos);

//print out Value of cell clicked on

if (NULL != hdc)

{

/\*WCHAR temp[100];

wsprintf(temp, L"Index = %d", index);

TextOut(hdc, xPos, yPos, temp, lstrlen(temp));\*/

//get cell demensions

if (index != -1)

{

RECT rcCell;

//cell is taken no other player can take over

if ((0 == gameBoard[index]) && GetCellRect(hWnd, index, &rcCell))

{

gameBoard[index] = playerTurn;

FillRect(hdc, &rcCell, (playerTurn == 2) ? hbr2 : hbr1);

//FillRect(hdc, &rcCell, hbr1);

winner = Winner(wins);

if (winner == 1 || winner == 2)

{

MessageBox(hWnd, (winner == 1) ? L"Player 1 is the winnner!" : L"Player 2 is the winner!", L"You Win!", MB\_OK | MB\_ICONINFORMATION);

playerTurn = 0;

}

else if (3 == winner)

{

MessageBox(hWnd, L"No one wins!", L"It's a draw", MB\_OK | MB\_ICONEXCLAMATION);

playerTurn = 0;

}

else if (0 == winner)

{

playerTurn = (playerTurn == 1) ? 2 : 1;

}

ShowTurn(hWnd, hdc);

}

}

ReleaseDC(hWnd, hdc);

}

}

}