// 游戏类封装

// 游戏类Game : Run()

// 绘制棋盘类ChessBoard ：Draw()

#if 1

#include <iostream>

#include<graphics.h>

#include<vector>

using namespace std;

int \_gridSize=50; //格子的大小

int \_gridCount=8; //格子的个数

int \_winWidth=640; //窗口的宽度

int \_winHeigh=480; //窗口的高度

int \_pieceSize=\_gridSize/2-2; //棋子大小

//开始位置的x坐标

int \_xStart=(\_winWidth-\_gridSize\*\_gridCount)/2;

//开始位置的y坐标

int \_yStart=(\_winHeigh-\_gridSize\*\_gridCount)/2;

//定义枚举，表示棋盘格子中的内容

typedef enum{CP\_NOTHING,CP\_BLACK,CP\_WHITE}CHESS\_PIECE;

//存放棋盘的格子

vector<vector<CHESS\_PIECE> > \_layout;

//玩家

CHESS\_PIECE \_player;

//定义鼠标位置的结构体

struct POS

{

int row; // 行

int col; // 列

//操作符重载

bool operator==(const POS& pos)

{

return row==pos.row&&col==pos.col;

}

};

//棋盘布局

void Layout();

//绘制棋盘

void DrawChessBoard();

//绘制落子

void Draw();

//运行游戏

void Run();

//获取鼠标位置

bool GetMouseHit(POS& pos);

//落子下棋

void SetPieces(POS pos);

int main()

{

//初始化绘图环境

initgraph(640,480);

//设置背景颜色

setbkcolor(EGERGB(255,110,110));

//设置填充色

// setfillcolor(EGERGB(255,255,255));

//填充椭圆

// fillellipse(50,50,25,25);

//设置前景颜色

// setcolor(EGERGB(255,255,255));

//绘制直线

// line(10,60,300,60);

//设置字体

// setfont(30,0,"华文彩云");

//输出文字

// outtextxy(10,10,"棋盘");

//棋盘布局

Layout();

//绘制棋盘

DrawChessBoard();

//绘制落子

Draw();

//运行

Run();

getch();

//关闭绘图环境

closegraph();

return 0;

}

//获取鼠标位置

bool GetMouseHit(POS& pos)

{

mouse\_msg msg=getmouse();

if(msg.is\_left()&&msg.is\_down())

{

if(msg.x-\_xStart>0 && msg.y-\_yStart>0)

{

pos.row=(msg.y-\_yStart)/\_gridSize;

pos.col=(msg.x-\_xStart)/\_gridSize;

if(pos.row>=0 && pos.row<\_gridCount &&

pos.col>=0 && pos.col<\_gridCount)

{

return true;

}

}

}

return false;

}

//运行游戏

void Run()

{

POS pos;

for(;is\_run();delay\_fps(100))

{

//获取鼠标位置

while(!GetMouseHit(pos));

//落子下棋

SetPieces(pos);

//绘制棋盘

DrawChessBoard();

//绘制落子

Draw();

delay\_fps(100);

}

}

//落子下棋

void SetPieces(POS pos)

{

// 根据玩家 ，设置对应位置的棋子颜色

\_layout[pos.row][pos.col]=\_player;

// 切换玩家

\_player=(CHESS\_PIECE)(3-\_player);

}

//绘制落子

void Draw()

{

for(int i=0;i<\_gridCount;++i)

{

for(int j=0;j<\_gridCount;++j)

{

//如果是黑棋

if(\_layout[i][j]==CP\_BLACK)

{

//设置填充色为黑色

setfillcolor(EGERGB(0,0,0));

}

else if(\_layout[i][j]==CP\_WHITE)

{

//如果是白棋

//设置填充色为白色

setfillcolor(EGERGB(255,255,255));

}

else

{

continue;

}

//圆心的坐标

int x=\_xStart+j\*\_gridSize+ \_gridSize/2;

int y=\_yStart+i\*\_gridSize+ \_gridSize/2;

//绘制棋子

fillellipse(x,y,\_pieceSize,\_pieceSize);

}

}

}

//绘制棋盘

void DrawChessBoard()

{

//清空绘图设备

cleardevice();

int x1,x2,y1,y2;

//绘制水平线

for(int i=0;i<=\_gridCount;++i)

{

x1=\_xStart;

x2=\_winWidth-x1;

y1=y2=\_yStart+\_gridSize\*i;

line(x1,y1,x2,y2);

}

int x3,x4,y3,y4;

//绘制垂直线

for(int j=0;j<=\_gridCount;++j)

{

y3=\_yStart;

y4=\_winHeigh-y3;

x3=x4=\_xStart+\_gridSize\*j;

line(x3,y3,x4,y4);

}

}

//棋盘布局

void Layout()

{

//给容器分配内存

\_layout.resize(\_gridCount);

for(int i=0;i<\_gridCount;++i)

{

\_layout[i].resize(\_gridCount);

}

//初始化棋盘格子的内容

for(int i=0;i<\_gridCount;++i)

{

for(int j=0;j<\_gridCount;++j)

{

\_layout[i][j]=CP\_NOTHING;

}

}

//设置对应的位置为白棋

\_layout[3][3]=\_layout[4][4]=CP\_WHITE;

//设置对应的位置为黑棋

\_layout[3][4]=\_layout[4][3]=CP\_BLACK;

//设置玩家

\_player=CP\_BLACK;

}

#endif