class Block

{

public:

char BLOCKS[10][10];//方块的类型

int sit[10][10];//四个方块的位置

int x;

int y;//确定一个重心

void moveup();//方块的变形

int num;//块ID

int NextNum;//下一块ID

Block ();//构造函数

bool detect();//检测是否与边界碰撞

};

bool Block::detect(int flag){

//获得当前操作的目标状态：目标坐标，目标方块类型。

int nextX, nextY, nextType;

// 0,1,2,3对应上下左右

switch(flag)

{

case 0: nextX = x; nextY = y; nextType = NEXT[type]; break;

case 1: nextX = x+1; nextY = y; nextType = type; break;

case 2: nextX = x; nextY = y-1; nextType = type; break;

case 3: nextX = x; nextY = y+1; nextType = type; break;

}

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

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

{

//tx，ty表示i，j所对应的地图的实际位置。

int tx = nextX+i;

int ty = nextY+j;

//边界处理

if(tx < 0 || tx > board->height+1 || ty < 0 || ty > board->width+1)

continue;

//如果块与地图墙相重合，则发生碰撞

if(BLOCKS[nextType][i][j] != '#' && (ty == 0 || ty == board->width+1))

return false;

if(BLOCKS[nextType][i][j] != '#' && board->map[tx][ty] != '#')

return false;

}

return true;

}

Block::Block(Board \*board){

this->board = board;

x = -3;

y = 4;

//随机数确定第一个块的类型

srand((unsigned) time(NULL));

type = rand()%19;

nextType = rand()%19;

//读入方块类型

freopen("block.bin", "rb", stdin);

for(int i=0; i<19; i++)

{

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

cin>>BLOCKS[i][j];

}

//next数组模拟指向，用于方块旋转功能

for(int i=0; i<3; i++){

for(int j=0; j<3; j++)

NEXT[i\*4+j] = i\*4+j+1;

NEXT[i\*4+3] = i\*4;

}

for(int i=12; i<18; i+=2){

NEXT[i] = i+1;

NEXT[i+1] = i;

}

NEXT[18] = 18;

//设置每种方块对应的颜色

COLOR[1] = Qt::blue;

COLOR[2] = Qt::darkBlue;

COLOR[3] = Qt::yellow;

COLOR[4] = Qt::magenta;

COLOR[5] = Qt::darkMagenta;

COLOR[6] = Qt::cyan;

COLOR[7] = Qt::red;

}