# 中国象棋任务1

## 1.1 任务要点

选定棋子；显示时间。

## 1.2 任务内容

在界面上显示了步时和对局时间，并且可以选定棋子。

## 1.3 任务实现

时间函数

**流程图：**

获取鼠标坐标

判断所对应的二维数组坐标

对局时间

每步倒计时

显示标记图片

结束

结束

**实现效果：**





**核心代码：**

void game\_time()

{

int time\_start = fclock();

int temp;

for(; is\_run(); delay\_fps(60))

{

temp = fclock();

if(temp != time\_start)

{

time\_start = temp;

setbkmode(OPAQUE);

setbkcolor(RGB(0,255,0));

xyprintf(120, 160, "%2d:%2d:%2d",time\_start/3600,(time\_start-time\_start/3600\*3600)/60,time\_start-(time\_start-time\_start/3600\*3600)/60\*60);

xyprintf(120, 210, "%d sec",30-time\_start);

}

if(!abs(30-time\_start))

{

setbkmode(TRANSPARENT);

outtextxy(45,50,"时间到,你输了!");

return ;

}

}

}

void init\_chess()

{

for(int row=0; row<10; row++)

{

for(int col=0; col<9; col++)

{

switch(chart[row][col])

{

case 1:

putimage(34.45\*col+230,35.4\*row,imgredche);

break;

case 2:

putimage(34.45\*col+230,35.4\*row,imgblackche);

break;

case 3:

putimage(34.45\*col+230,35.4\*row,imgredma);

break;

case 4:

putimage(34.45\*col+230,35.4\*row,imgblackma);

break;

case 5:

putimage(34.45\*col+230,35.4\*row,imgredxiang);

break;

case 6:

putimage(34.45\*col+230,35.4\*row,imgblackxiang);

break;

case 7:

putimage(34.45\*col+230,35.4\*row,imgredshi);

break;

case 8:

putimage(34.45\*col+230,35.4\*row,imgblackshi);

break;

case 9:

putimage(34.45\*col+230,35.4\*row,imgredshuai);

putimage\_withalpha(NULL,imgmark,34.45\*col+230,35.4\*(row+1));

break;

case 10:

putimage(34.45\*col+230,35.4\*row,imgblackjiang);

break;

case 11:

putimage(34.45\*col+230,35.4\*row,imgredpao);

break;

case 12:

putimage(34.45\*col+230,35.4\*row,imgblackpao);

break;

case 13:

putimage(34.45\*col+230,35.4\*row,imgredbing);

break;

case 14:

putimage(34.45\*col+230,35.4\*row,imgblackzu);

break;

case 15:

putimage\_alphablend(NULL,imgredche, 34.45\*col+230,35.4\*row, 0x80);

break;

case 16:

putimage\_alphablend(NULL,imgblackche, 34.45\*col+230,35.4\*row, 0x80);

break;

case 17:

putimage\_alphablend(NULL,imgredma, 34.45\*col+230,35.4\*row, 0x80);

break;

case 18:

putimage\_alphablend(NULL,imgblackma, 34.45\*col+230,35.4\*row, 0x80);

break;

case 19:

putimage\_alphablend(NULL,imgredxiang, 34.45\*col+230,35.4\*row, 0x80);

break;

case 20:

putimage\_alphablend(NULL,imgblackxiang, 34.45\*col+230,35.4\*row, 0x80);

break;

case 21:

putimage\_alphablend(NULL,imgredshi, 34.45\*col+230,35.4\*row, 0x80);

break;

case 22:

putimage\_alphablend(NULL,imgblackshi, 34.45\*col+230,35.4\*row, 0x80);

break;

case 23:

putimage\_alphablend(NULL,imgredshuai, 34.45\*col+230,35.4\*row, 0x80);

break;

case 24:

putimage\_alphablend(NULL,imgblackjiang, 34.45\*col+230,35.4\*row, 0x80);

break;

case 25:

putimage\_alphablend(NULL,imgredpao, 34.45\*col+230,35.4\*row, 0x80);

break;

case 26:

putimage\_alphablend(NULL,imgblackpao, 34.45\*col+230,35.4\*row, 0x80);

break;

case 27:

putimage\_alphablend(NULL,imgredbing, 34.45\*col+230,35.4\*row, 0x80);

break;

case 28:

putimage\_alphablend(NULL,imgblackzu, 34.45\*col+230,35.4\*row, 0x80);

break;

}

}

}

}