**中北大学**

**程 序 设 计 基 础**

**课 内 项 目 说 明 书**

|  |  |  |  |
| --- | --- | --- | --- |
| **学生姓名:** | **高天琦** | **学 号：** | 1807004223 |
| **学 院:** | **大数据学院**  **（计算机科学与技术学院）** | | |
| **班 级:** | 18070042 | | |
| **题 目:** | **高天琦** | | |
| **指导教师** | **梁志剑** | | |

2019年 1月7日

**1．设计目的**

夯实C语言基础，制作一个自己喜欢的游戏。

**2．设计内容**

此游戏为大鱼吃小鱼游戏，玩家有三条命，并且用鼠标控制，吃到比自己小的鱼分数会增加，到达一定值会长大。

**3.本设计所采用的数据结构**

数据结构中包括刷新鱼的方向（鱼在左边刷新还是右边刷新），鱼的坐标，鱼的大小，鱼游动的速度，指向下一个单元的指针。

**4．功能模块详细设计**

**4.1游戏图片输出模块**

**4.1.1详细设计思想**

通过在死循环里以背景图玩家鱼与敌方鱼分别输出一次为一张图，来无限的打印图片，生成的下一张图片会把上一张图片覆盖掉。

**4.1.2 核心代码**

void game()

{

loadpcture();

loadsize();

h=NULL;

for(i=1;;i++)

{

BeginBatchDraw();

if(i==1000)

i=1;

fishsize();

mouse();

gameback();//输出游戏背景

if(i%100==0)

makeefish();

showefish();//输出敌方的鱼

efishcontrol();

controlsize();

controlscore();

outscore();

fisheat();

outfishlife();//输出玩家鱼

if(h!=NULL)

initgraphjudge();

if(protect==0)

{

protect=30;

if(a!=0)

controllife();

}

if(protect!=0)

protect--;

EndBatchDraw();

if(life<1)

gameover();

if(score>3000)

cow();

FlushBatchDraw();

Sleep(30);

}

}

void showefish()//输出敌方鱼

{

p=h;

while(p)

{

if(p->direction==0)

{

if(p->level==0||p->level==1||p->level==2)

{

putimage(p->fx,p->fy,&m131,NOTSRCERASE);

putimage(p->fx,p->fy,&m132,SRCINVERT);

}

else if(p->level==3||p->level==4)

{

putimage(p->fx,p->fy,&m151,NOTSRCERASE);

putimage(p->fx,p->fy,&m152,SRCINVERT);

}

else if(p->level==5||p->level==6)

{

putimage(p->fx,p->fy,&m141,NOTSRCERASE);

putimage(p->fx,p->fy,&m142,SRCINVERT);

}

else if(p->level==7)

{

putimage(p->fx,p->fy,&m161,NOTSRCERASE);

putimage(p->fx,p->fy,&m162,SRCINVERT);

}

else if(p->level==8)

{

putimage(p->fx,p->fy,&m121,NOTSRCERASE);

putimage(p->fx,p->fy,&m122,SRCINVERT);

}

}

else if(p->direction==1)

{

if(p->level==0||p->level==1||p->level==2)

{

putimage(p->fx,p->fy,&fm131,NOTSRCERASE);

putimage(p->fx,p->fy,&fm132,SRCINVERT);

}

else if(p->level==3||p->level==4)

{

putimage(p->fx,p->fy,&fm151,NOTSRCERASE);

putimage(p->fx,p->fy,&fm152,SRCINVERT);

}

else if(p->level==5||p->level==6)

{

putimage(p->fx,p->fy,&fm141,NOTSRCERASE);

putimage(p->fx,p->fy,&fm142,SRCINVERT);

}

else if(p->level==7)

{

putimage(p->fx,p->fy,&fm161,NOTSRCERASE);

putimage(p->fx,p->fy,&fm162,SRCINVERT);

}

else if(p->level==8)

{

putimage(p->fx,p->fy,&fm121,NOTSRCERASE);

putimage(p->fx,p->fy,&fm122,SRCINVERT);

}

}

p=p->next;

}

}

**4.2鱼大小控制模块**

**4.2.1详细设计思想**

每次循环中通过判断所得分数所在区间来决定玩家鱼的大小。

**4.2.2 核心代码**

void controlsize()

{

if(score<25)

fish1();

else if(score>=25&&score<75)

fish2();

else if(score>=75&&score<175)

fish3();

else if(score>=175&&score<275)

fish4();

else if(score>=275&&score<375)

fish5();

else if(score>=375&&score<475)

fish6();

else if(score>=475&&score<575)

fish7();

else if(score>=575&&score<675)

fish8();

else if(score>=675)

fish9();

}

**4.3大鱼吃小鱼判定模块**

**4.3.1详细设计思想**

通过链表遍历的方法和矩形判定，当矩形判定成功并且玩家鱼的宽像素比敌方鱼的宽像素大的时候这只敌方鱼会刷新它的所有属性从界面随机左右两边再次出来。

**4.3.2 核心代码**

void fisheat()

{

p3=h;

q3=h;

while(p3)

{

if(s.x+twidth>p3->fx&&s.x<p3->fx+p3->efishwidth&&s.y<p3->fy+p3->efishheigh&&s.y+theigh>p3->fy)

{

if(twidth>p3->efishwidth)

{

p3->fy=rand()%572;

if(p3->direction==0)

p3->fx=1000;

else if(p3->direction==1)

p3->fx=-400;

p3->level=rand()%9;

if(p3->level==0||p3->level==1||p3->level==2)

{

p3->efishwidth=35;

p3->efishheigh=29;

}

else if(p3->level==3||p3->level==4)

{

p3->efishwidth=90;

p3->efishheigh=72;

}

else if(p3->level==5||p3->level==6)

{

p3->efishwidth=200;

p3->efishheigh=156;

}

else if(p3->level==7)

{

p3->efishwidth=300;

p3->efishheigh=202;

}

else if(p3->level==8)

{

p3->efishwidth=400;

p3->efishheigh=207;

}

}

}

p3=p3->next;

}

}

**4.4玩家鱼生命控制模块**

**4.4.1详细设计思想**

当玩家鱼与敌方鱼碰撞时，如果敌方鱼的宽像素比玩家鱼的大，那么玩家的生命会减一。

如果生命值减一的时候玩家鱼会有一定的无敌时间，防止生命值减少过快。无敌通过定义一个变量int protect=0当生命减少时protect=30，控制当protect不等于0的时候每次循环protect减一。从而实现无敌时间的控制。

**4.4.2 核心代码**

void controllife()

{

p5=h;

while(p5)

{

if(s.x+twidth>p5->fx&&s.x<p5->fx+p5->efishwidth&&s.y<p5->fy+p5->efishheigh&&s.y+theigh>p5->fy)

{

if(twidth<p5->efishwidth)

life--;

}

p5=p5->next;

}

}

void game()

{

loadpcture();

loadsize();

h=NULL;

for(i=1;;i++)

{

BeginBatchDraw();

if(i==1000)

i=1;

fishsize();

mouse();

gameback();

if(i%100==0)

makeefish();

showefish();

efishcontrol();

controlsize();

controlscore();

outscore();

fisheat();

outfishlife();

if(h!=NULL)

initgraphjudge();

if(protect==0)//无敌的控制

{

protect=30;

if(a!=0)

controllife();

}

if(protect!=0)//无敌的控制

protect--;

EndBatchDraw();

if(life<1)

gameover();

if(score>3000)

cow();

FlushBatchDraw();

Sleep(30);

}

}

**4.5游戏初始界面的控制模块**

**4.5.1详细设计思想**

通过输出初始菜单来控游戏的开始与结束，当输入1的时候游戏开始，输入2是歪瓜验证如果外挂输入正确玩家鱼会成为无敌状态。

**4.5.2 核心代码**

int main()

{

initgraph(1000,572);

mciSendString("open C:\\Users\\10614\\Desktop\\大鱼吃小鱼项目图片\\影视原声-猪八戒背媳妇.mp3 alias bgm",NULL,0,NULL);

mciSendString("play bgm repeat",NULL,0,NULL);

startshow();

control();

return 0;

}

void control()

{

while(1)

{

char a=getch();

switch(a)

{

case '1':game();break;

case '2':hack();break;

case '3':shezhi();break;

case '4':return ;break;

default:break;

}

}

}

**4.6鱼分数控制模块**

**4.6.1详细设计思想**

Int socre=0；通过数组遍历的方法判定如果玩家鱼与敌方的鱼碰撞，并且玩家的鱼大鱼敌方的鱼，那么玩家的分数会得到增加。

**4.6.2 核心代码**

void controlscore()

{

p1=h;

while(p1)

{

if(s.x+twidth>p1->fx&&s.x<p1->fx+p1->efishwidth&&s.y<p1->fy+p1->efishheigh&&s.y+theigh>p1->fy)

{

if(twidth>p1->efishwidth)

{

if(p1->level==0||p1->level==1||p1->level==2)

score=score+5;

else if(p1->level==3||p1->level==4)

score=score+10;

else if(p1->level==5||p1->level==6)

score=score+20;

else if(p1->level==7)

score=score+40;

else if(p1->level==8)

score=score+50;

}

}

p1=p1->next;

}

}

**5．心得体会详细**

做项目的前期有太多的函数都不知道，做这个项目的过程本来就是一个学习的过程，这个项目做完之后我觉得我对C语言有了一个新的认识。在制作游戏的过程各种各样并且意想不到的bug都能出现，搞得我头脑发昏，曾有很多次想放弃转而做一个比较简单的项目。但是我还是坚持了下来。当游戏做成功的时候我又是喜悦，还有一种激动的想哭的感觉。做项目的过程中遇到的各种困难磨炼了我的意志，让我感受到了坚持啊的重要性。在接下来的时间我会把我的项目不断的完善，让我的游戏功能更多bug更少。