

基于 C 语言的 Flappy Bird 游戏的设计与开发

刘沁沁 段华琼

四川大学锦城学院 计算机与软件学院 四川 成都 611371

【摘要】Flappy bird 是一款经典的小游戏,本文介绍基于 C 语言的 Flappy bird 游戏的开发。本文首先简单介绍了 Flappy bird 游戏,然后介绍了游戏的功能结构以及工作流程,最后重点介绍了游戏部分功能设计及实现。

【关键词】游戏设计;Flappy bird;C 语言

1 引言

Flappy bird 作为一款适合全年龄段娱乐的小型益智游戏,大受群众的喜爱和欢迎。其简单有趣,久玩不腻,不仅适合人们休息的时候消遣,还可以锻炼人的思维能力,训练人的注意力。

C 语言是一种面向过程的编程工具,它在使用少量代码并不需要任何环境支持的情况下也可以运行高效的程序。本文基于 C 语言设计了一款 Flappy bird 小游戏,使这个小游戏可以在简单的条件下在计算机中使用,希望通过这个简单的游戏使大家更透彻的了解 C 语言,也为广大游戏开发爱好者提供更多的灵感和新思路。

2 游戏功能设计

(1) 游戏简介

Flappy bird 不同于普遍的闯关类游戏,它是一个无剧情无结局的积分制游戏。对游戏的功能介绍如下:

● 开始游戏时小鸟自动向前进行位移并且自动下落,玩家通过敲击空格键控制小鸟向上跳跃来躲避并跃过障碍物。

● 小鸟每通过一个障碍物积分增加一分,小鸟通过障碍物后障碍物自动消失。

● 障碍物的出现频率是固定的,但出现障碍物的长短是随机的。

● 游戏过程中如果想要进行存档则敲击 F1 键,存档后自动进入游戏欢迎界面,如果想要返回游戏,则敲击 F2 读档继续游戏。

● 游戏过程中如果想要暂停游戏则敲击 F3 键进入游戏暂停界面,此时如果想要继续游戏则敲击 F4 键回到游戏进行界面继续游戏。

● 一旦小鸟碰到障碍物或并未通过障碍物则判定游戏失败,游戏失败的同时进行积分统计,在游戏的界面下方会显示历史最高积分。

● 想要在游戏进程中直接结束游戏可以敲击 ESC 键返回游戏欢迎界面。

(2) 系统功能结构

Flappy bird 游戏拥有四个游戏界面,分别是游戏欢迎界面、游戏主窗体、游戏说明界面以及游戏结束界面,系统功能结构如图 1 所示:

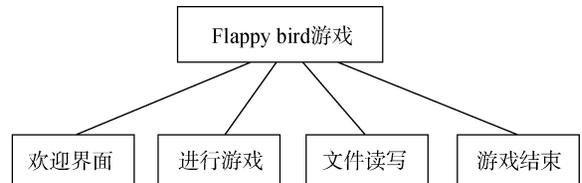


图 1 系统功能结构

① 欢迎界面包括以下功能:

- 设置欢迎字符画、语句
- 设置菜单选项

② 游戏模块包括以下功能:

- 绘制地图、小鸟和障碍物
- 判断小鸟是否撞到障碍物
- 设置小鸟的自动向前位移和自由下落
- 设置键盘按键功能
- 设置小鸟碰到障碍物的情况

③ 文件读写模块包括以下功能:

- 最高分的写入和读取最高分
- 游戏进程的写入和读取游戏进程

④ 游戏结束模块包括以下功能:

- 小鸟撞到障碍物
- 敲击 ESC 键

(3) 业务流程

游戏的业务流程图如图 2 所示。这个完整的业务流程图清晰说明了游戏操作过程。

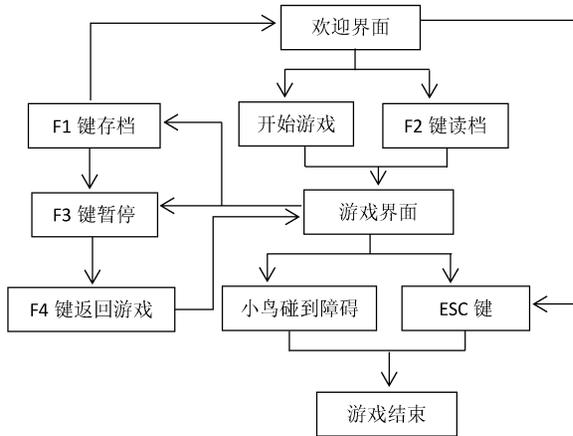


图 2 游戏流程图

3. 游戏具体实现

(1) 开始界面设计

游戏需要设计一个对玩家相对友好的开始界面,界面应该包括游戏的相关规则介绍和各种鼓励语句,以提起玩家对游戏的兴趣。本游戏开始界面的设计主要依靠简单的 printf 输出语句,简洁明了,以下是详细的程序代码:

```
void welcome()
{
    printf(" * * * * * \n");
    printf("欢迎来到 Flappy bird 游戏! \n");
    printf("请玩家通过空格键控制小鸟的跳跃,使小鸟跃过障碍物,如果碰到障碍物则游戏失败,在此祝玩家游戏愉快! \n");
    printf(" * * * * * \n");
}
```

(2) 暂停页面设置

玩家暂停游戏的时候会进入暂停页面,在这个页面里玩家可以继续游戏也可以选择结束,这里涉及存档和读档的设计过程。暂停游戏相当于一个短时间的存档操作,当玩家选择继续游戏的同时就是在进行读档操作,这种存档相当于是一次性的。设计中为了增强玩家的游戏体验,可以加入一个 Sleep() 函数来增加游戏延迟准备时间,使玩家能够拥有充分的准备时间来进行下一步的游戏。设计的每一步都要有对玩家是否有正确输入的判定,当然在设

计暂停操作的同时也需要加入一些界面细节润色,比如插入图片让玩家知道此时暂停成功而非系统卡顿。以下是大概的程序代码:

```
void pauseMenu()
{
    while(1)
    {
        loadImage(&pause,"F:\\暂停时的图片.jpg");//打开暂停菜单的背景图片
        putimage(0,0,&pause);
        FlushBatchDraw();
        char input;
        if(kbhit())//判断是否有鼠标输入
        {
            input=getch();
            if(input=='1')//开始新游戏
            {
                gameStatus=1;//不进入主菜单,直接开始新游戏
                main();
            }
            else if(input=='2')//继续游戏
            {
                startup();//从头开始初始化后读取存档,继续游戏
                readRecordFile();//读取存档
                show();
                Sleep(1400);//预备开始阶段延时
                while(1)//游戏循环执行
                {
                    show();//显示画面
                    updateWithoutInput();
                    updateWithInput();
                }
            }
            else if(input=='3')//回主菜单
            {
                gameStatus=0;
                startup();
                putimage(0,0,&img_bk); //打开预备开始阶段背景画面
                FlushBatchDraw();
            }
        }
    }
}
```

```

show();
Sleep(1400); //预备开始阶段延时
break;
}
else if(input == '4') //退出游戏
{
gameStatus = 1;
exit(0);
}
}
}
}
}

```

(3) 存档和读档设计

存档和读档是本游戏的特点之一,相对比其他很多小型游戏里都没有存档功能,本游戏的功能更加完整。存档时将游戏的核心数据储存到计算机中,需要读档时又可以调用已存储的数据继续游戏。以下程序的目的是将数据储存在电脑硬盘中的游戏记录里,也就是将小鸟、障碍物等游戏界面元素的位置数据储存下来,其中运用到了指针变量的知识点。以下是详细的程序代码:

```

void readRecordFile()
{
FILE * fp;
fp = fopen("F:\\游戏记录.txt", "r");
//游戏核心数据读出
fscanf(fp, "%d %d %d %d %d %d",
&.huaji_x, &.huaji_y, &.bar_x, &.shang_y, &.xia
_y, &.score);
fclose(fp);
}
//游戏数据写入存档,保存游戏记录
void writeRecordFile()
{
FILE * fp;
fp = fopen("F:\\游戏记录.txt", "w+");
//游戏核心数据写入
fprintf(fp, "%d %d %d %d %d %d",
huaji_x, huaji_y, bar_x, shang_y, xia_y, score);
fclose(fp);
}

```

(4) 游戏主窗体设计

游戏主窗体的设计是本游戏的重点,包括小鸟坐标、障碍坐标的设置,游戏数据的显示和背景音乐的设置,这里首先导入各个元素的图片对页面进行细化。Flappy bird 的一大游戏特点在于小鸟跃过的障碍是随机出现的,对玩家来说是一个挑战,所以在初始化小鸟及障碍物坐标时,运用到系统时间产生伪随机种子来得到随机位置的障碍,并设定障碍随机出现的坐标范围,以防止游戏玩到中途出现无法通过障碍的无解情况。以下是详细的程序代码:

```

void startup()
{
score = 0; //初始化分数为 0
initgraph(Width, High); //定义游戏界面大小
loadimage(&.img_bk, "F:\\背景图片.jpg");
loadimage(&.huaji1, "F:\\小鸟 1.jpg");
loadimage(&.huaji2, "F:\\小鸟 2.jpg");
loadimage(&.shang1, "F:\\上障碍 1.jpg");
loadimage(&.shang2, "F:\\上障碍 2.jpg");
loadimage(&.xia1, "F:\\下障碍 1.jpg");
loadimage(&.xia2, "F:\\下障碍 2.jpg"); //打
开一系列游戏界面的图片
mciSendString("open F:\\背景音乐 1. mp3
alias bkmusic_1", NULL, 0, NULL);
//打开背景音乐
mciSendString("play bkmusic_1 repeat",
NULL, 0, NULL); //音乐的播放
srand(time(NULL));
//利用系统时间产生伪随机种子,以为了得到
随机位置的障碍
huaji_x = 24; //初始化小鸟坐标
huaji_y = rand() % 150 + 250;
bar_x = 450; //初始化障碍的坐标
shang_y = rand() % 400 - 550;
while(xia_y - shang_y > 730 || xia_y -
shang_y < 670)
//上下障碍随机出现的坐标范围
{
shang_y = rand() % 400 - 550;
xia_y = rand() % 300 + 150;
}
while(gameStatus == 0) //状态为 0 则
进入主菜单

```

```

{
Sleep(200);
startMenu();//主菜单界面
FlushBatchDraw();
}
}

```

(5)设置背景音乐

玩游戏如果没有背景音乐会少了很多乐趣,所以在游戏进行中最好设置背景音乐让玩家获得更好的游戏体验,放松心情。设置背景音乐时用到了引用电脑文件地址的函数,以下是详细的程序代码:

```

mciSendString("close bkmusic_1",NULL,0,
NULL);//关闭主菜单背景音乐
mciSendString("open F:\\背景 音乐 2. mp3
alias bkmusic_2",NULL,0,NULL);
//打开游戏背景音乐
mciSendString("play bkmusic_2 repeat",
NULL,0,NULL);//播放
mciSendString("close beginmusic",NULL,0,
NULL);
mciSendString("open F:\\开始. mp3 alias be-
ginmusic ",NULL,0,NULL);
//播放一次开始时的预备音效
mciSendString("play beginmusic",NULL,0,
NULL);

```

(6)设计小鸟动作

本游戏中,小鸟是可以灵活地移动的,所以设计小鸟的移动方式是这个游戏的重点之一。其过程是首先判断键盘是否有输入,其次通过赋值 bird_y 来确定小鸟的跳动高度,然后通过函数 putimage 设置小鸟每跳动一次的动作变化,动作变化的效果图可以导入计算机中设置好的图片,其中通过 mciSendString 函数加入动作音效,同时运用 if 判断语句设置小鸟的自动下落。最后使用一个 if-else 判断语句设置障碍的自动移动、出现和消失,由于设置了屏幕的大小,所以可以设定当长度超过 140 时,进行最左边障碍的清除和最右边障碍的自动生成。以下是详细的程序代码:

```

void updateWithInput()
{
char input;
if (kbhit())//判断是否有键盘输入

```

```

{
input = getch();
if (input=='\`')
{
loadimage(&tiao1,"F:\\我跳 1. jpg");
loadimage(&tiao2,"F:\\我跳 2. jpg");
putimage(huaji_x, huaji_y, &tiao1,NOTSR-
CERASE);
//每跳一下小鸟的表情就变一次
putimage(huaji_x, huaji_y, &tiao2, SRCINVE-
RT);
huaji_y-=30;//点一下空格小鸟就会上跳的
高度
mciSendString("close jpmusic", NULL, 0,
NULL);
mciSendString("open F:\\跳. mp3 alias jpmu-
sic ",NULL,0,NULL);
//按一下空格播放一次音效
mciSendString("play jpmusic", NULL, 0,
NULL);
}
else if(input==27) //27 为 esc 键的 ascll 码
{
writeRecordFile();//存档
mciSendString("close bkmusic_2",NULL,0,
NULL);
//关闭游戏时的背景音乐
pauseMenu();//打开暂停界面
}
}
void updateWithoutInput()
{
if (huaji_y<High) //自动下落
huaji_y++;
if (bar_x>-140) //障碍向左移动,140 为
整个障碍的宽度
bar_x--;
else //新的障碍从屏幕右边出现
{
bar_x = Width;
shang_y = rand()%300 - 450;
while(xia_y - shang_y > 730 || xia_y -

```

```
shang_y<670)
```

```
{
    shang_y = rand()%300 - 450;
    xia_y = rand()%300 + 150;
}
```

(7)游戏结束

本游戏随着小鸟的死亡而结束,而结游戏结束是通过判断小鸟是否碰到障碍来进行的。判断方法很简单,只要判断小鸟的坐标是否“碰到”障碍的坐标就可以,即判断一个时点小鸟的坐标是否与属于障碍的坐标有相同的地方,如果有,那就判断游戏结束,这个过程通过 if 语句实现,实现判定的同时可以增加音乐方面的润色,在结束的同时停止播放背景音乐,并通过导入电脑中下载好的尖叫音效来提醒玩家游戏结束。以下是详细的程序代码:

```
if (((bar_x<50 && bar_x>-100) &&
(huaji_y+40>xia_y || huaji_y<shang_y+585))
```

```
|| (huaji_y>550 || huaji_y<0 ))//判断失败
{
    mciSendString("close bkmusic_2",NULL,0,
NULL); //关闭背景音乐
    mciSendString("close overmusic_1",NULL,
0,NULL);
    mciSendString("open F:\\尖叫.mp3 alias
overmusic_1",NULL,0,NULL);
    //播放触碰危险边缘的尖叫音效
    mciSendString("play overmusic_1",NULL,0,
NULL);
};
```

4 结束语

本文基于 C 语言实现了一个 Flappy bird 小游戏。本游戏沿用了传统 Flappy bird 游戏的规则设定,更注重游戏的拓展性和操作体验,增加了更多的细节设计。希望本文能够对广大 C 语言学习者以及游戏开发爱好者提供一些新思路。

【参考文献】

- [1]谭浩强. C 语言程序设计 [M]. 北京:清华大学出版社,2018 年
- [2]明日科技. C 语言项目开发全程实录 [M]. 北京:清华大学出版社,2018 年
- [3]谢昊,管乡镇. 基于 JAVA 平台的 flappy+bird 游戏设计[J]. 电脑知识与技术,2018 年
- [4]脚本之家. C 语言实现 flappy bird 游戏.
https://blog.csdn.net/qq_40685101/article/details/80275112
- [5]CSDN. 课程设计—用 C 语言编码一个小游戏(flappy bird).
https://blog.csdn.net/qq_43605229/article/details/88948520