

河海大学

物联网工程学院

课程设计报告

**C语言程序设计**

**——小鸡弹跳设计报告**

|  |  |
| --- | --- |
| **授课班号：** | c0601128-c01 |
| **课程名称：** | 程序设计基础 |
| **专业年级：** | 计算机科学与技术2021级本科 |
| **姓 名：** | 杨婧蕙 |
| **学 号：** | 2162810203 |
| **学年学期：** | 2021-2022学年第一学期 |
| **指导老师：** | 童晶 |

打印日期：2022年1月8日

摘要：本次游戏设计以同名手机游戏（小鸡弹跳）为参考对象制作而成，通过EasyX插件和C语言编程实现。玩家可通过鼠标交互实现多种操作进行游戏，丰富的游戏效果可给玩家带来丰富的游戏体验。本文详细地介绍了该游戏的设计思路、游戏框架与主要函数功能描述、分步骤实现方法与核心代码以及参与游戏设计人员的体会总结。

关键词：小鸡弹跳、游戏设计、C语言、面向对象编程、EasyX

1. 设计思路

本游戏以同名手机游戏（小鸡弹跳）为参考对象，通过EasyX插件和C语言编程实现。玩家可通过鼠标交互查看游戏说明或开始游戏，开始游戏后点击鼠标左键实现小鸡身体高度增加，从而绕过随机生成的多种障碍物，小鸡安全跑完全程则游戏胜利，头部撞到障碍物则游戏失败。此外，本游戏还添加了游戏进度条、音效、背景变换等效果，丰富了玩家的游戏体验。

1. 功能描述

本游戏的主体功能为点击鼠标实现小鸡身体数量的增加和被撞身体随砖块的移动。次要功能为砖块类型的随机出现以及砖块位置的改变。主要函数功能介绍如下：

2.1 主函数

主函数可体现本游戏的游戏框架。首先初始化游戏数据，之后利用while循环更新绘制、和输入无关的更新、和输入有关的更新这三个函数，从而实现画面的不断更新。

2.2 初始化

初始化函数分为startup()函数以及startupBlock()函数。将砖块的初始化从主体初始化中分离，便于不断初始化砖块数据，以达到在同一次程序运行中随机出现不同类型砖块的效果。

2.21 startup()函数

1.导入并循环播放音乐（mciSendString()）。

2.导入所有图片、背景（loadimage），设置各个图像的初始坐标。

3.新开一个画面（inigraph()）。

2.22 startupBlock()函数

1.应用枚举类型和随机函数，随机出现不同形式的砖块：n个砖块横向 组合，阶梯形砖块，悬浮砖块，落地长砖块等。

2.开始批量绘制（BeginBatchDraw()）。

2.3 显示

显示函数分为show()函数和showStartGame()函数，show()函数中调用showStartGame()函数：

2.31 display

设置全局变量display表示游戏当前状态，display=0表示处于游戏开始界面，display=1表示处于游戏状态，display=2表示处于游戏说明界面。

2.32 show()函数

1. 主函数中调用show()函数，该函数可与2.5与输入有关的更新进行交互。开始执行程序时默认display=0，显示游戏开始界面。玩家可通过鼠标交互进行操作，若玩家选择开始游戏（display=1）则调用showStartGame()函数从而显示游戏进行画面。

2.开始按键、游戏说明按键的发光显示。

3. 批量绘制图像（FlushBatchDraw()）。

2.33 showStartGame()函数

1. 显示小鸡、砖块、背景等图像。

2. 在屏幕上方显示小鸡移动的距离，并根据距离显示进度条。

3. 显示游戏失败界面和游戏胜利界面。

4.接收2.5与输入有关的更新中bodynum（身体数量）的值，从而绘制出相应数量的身体。

2.4 与输入无关的更新

与输入无关的更新使用updateWithoutInput()函数，对游戏中各个类元素的状态进行更新并由此判断游戏是否失败或胜利：

1. 调用背景、云朵、砖块等类中的update()函数，进行图像位置自动更新。
2. 进行砖块和身体碰撞的判断及被碰身体随砖块移动的数据更新。
3. 身体某部位被砖块推走后，进行上方身体和头部下落数据的更新。
4. 通过给全局变量finddd赋值，对所有砖块类型与身体碰撞的结果进行分类讨论。
5. 游戏胜利与游戏失败的判定。

2.5 与输入有关的更新

与输入有关的更新使用updateWithInput()函数，实现鼠标点击与其他各部分（如2.3 显示中show()函数）的交互作用，实现游戏中与输入有关的数据和图片的更新及点击音效的播放。

1. 分步骤实现方法

3.1 砖块移动，小鸡出现

绘制窗口，利用"EasyXPng.h"头文件导入素材，运用面向对象（class）设置多个类别并搭建大致框架。定义砖块图像横纵坐标x、y与速度v，并在与输入无关的更新中不断更新图像坐标数据，利用while循环实现砖块的移动。

3.2 每按左键增加一个身体

在与输入有关的更新中加入鼠标交互信息，使chickenhead.n（头部图片纵坐标）和chickenbody.b（身体图片纵坐标）的数据发生相应变化，达到每按左键增加一个身体的效果。代码如下：

void updateWithInput()//和输入有关的更新

{

MOUSEMSG m;//定义鼠标消息

if(MouseHit())

{

m=GetMouseMsg();//获得鼠标消息

if(m.uMsg==WM\_LBUTTONDOWN)//如果按下鼠标左键

{

chickenbody.b=chickenhead.n+gao;//更新身体的纵坐标

chickenhead.n=chickenhead.n-gao;//头部向上移动一个身体的距离

bodynum++;

}

}

}

3.3 砖块推走身体，上方头部和身体下落（重点&难点）

这部分的判断条件较为严苛，逻辑稍错一点就会出现很离谱的效果。经过反复修改，我们确定如下实现思路：先在画面固定位置导入全部身体图像但并不显示，引入bodynum变量统计玩家点击屏幕的次数并显示对应数量的身体图片，这样可以更便捷地实现小鸡身体随鼠标输入的增加；将每个砖块和身体看做独立的个体，依次遍历，找出与砖块碰撞的身体使其随砖块向左移动，并引入pushNum变量实现其上方的身体及头部的精准下落。核心代码如下：

1.判断身体是否与砖块碰撞

for (int i=0;i<blockNum;i++)//遍历所有砖块

{

for(int j=0;j<numMax;j++)//遍历所有身体,找出与砖块相碰的身体

{

if(chickenbody[j].b>=block[i].y-20&&chickenbody[j].b<=block[i].y+20)//找出与每个砖块距离最近的身体

{

if(chickenbody[j].a+gao==block[i].x)//如果身体右侧和砖块左侧的横坐标恰好相等（即判断身体与砖块相撞）

{

chickenbody[j].a=chickenbody[j].a-1;//该身体被砖块推走，向左移动，速度与砖块相同

}

}

}

}

2.某身体被砖块推走后，其上方身体及头部的下落

for (int i=0;i<blockNum;i++)//遍历所有砖块

{

for(int j=0;j<numMax;j++)//遍历所有身体

{

if(chickenbody[j].a==block[i].x+gao)//如果该砖块离开头身列（小鸡头部和未被推走的身体所在列）

{

if(chickenbody[j].b<block[i].y)//找出砖块上面的身体

{

if(block[blockNum-1].x<chickenbody[j].a)//如果该砖块为每组砖块最后一个

{

chickenbody[j].b=chickenbody[j].b+gao\*pushNum;//身体下落

find1=1;//该变量赋值为1，表示头部应下落

}

}

}

}

}

if(find1==1)//如果头部应下落

chickenhead.n=chickenhead.n+gao\*pushNum;//头部下降对应高度

3.4 随机生成不同类型的砖块（难点）

我们的预期目标是在程序的一次运行中可以随机出现不同类型砖块，但是后来发现程序每次运行只会随机生成一种类型的砖块。究其原因，是程序每次运行只会初始化一种砖块的数据。因此我们想到将砖块数据初始化从主体初始化中分离的方法，当上一组砖块从屏幕中消失后，重新初始化砖块数据并绘制对应类型的砖块图像，从而实现在程序的一次运行中随机出现不同类型砖块的效果。代码如下：

1.砖块初始化部分，生成随机类型的砖块

void startupBlock()

{

int type=rand()%5;

if(type==n\_32Block)

n32Block();

else if(type==n\_Lengthwise)

nLengthwise();

else if(type==three\_Next)

threeNext();

else if(type==n\_Broadwise)

nBroadwise();

else if(type==n\_6Block)

n6Block();

BeginBatchDraw();// 开始批量绘制

}

2.砖块从页面消失，重新初始化

if(block1.x2==-gao)

{

startupBlock();//重新生成随机类型的砖块

}

3.5 游戏的完善

游戏的完善也应用了一些巧妙的方法。例如：

1. 为实现背景图片向左移动的连贯显示，我们将背景图片的左右两端处理为相同图像；
2. 查阅资料，应用“mciSendString(\_T("play jpmusic wait"),NULL,0,NULL); ”添加“wait”，播放完歌曲再返回，成功解决音乐播放不良问题。
3. 参考“王国保卫战”的血条制作方法，应用putimage的图片部分显示，制作了游戏进度条，代码如下：

putimage(50,100,300,110,&im\_progressbar,-nowSecond\*3/10+300,0);

putimagePng(nowSecond\*3/10,68,&im\_progressbarHead);

1. 体会与总结

刚开始游戏选题时其实遇到了很多困难，经过重重筛选才选择了小鸡弹跳这一款又好玩看起来又不那么难的游戏。刚开始的时候确实是头脑一片空白，毫无头绪，感觉无从下手，但在参考童老师的火柴人的无尽冒险游戏过后，我们也能从无到有，从简单框架到添加功能，使游戏初具雏形。经过一个多月无数次熬夜苦战，我们组俩人也多次在灵感枯竭时又突然想法乍现，在无数个自己和别组同学发现的bug斗智斗勇中获胜，终于写完了小鸡弹跳。我们还添加了可爱的音效，变幻的背景等让游戏变得更有趣。  
 在做游戏的过程中，真的是快乐与痛苦并存。看似容易敲的代码实则难以实现，我们甚至花了半个多月死磕其中一个bug，这个过程确实是痛苦的，但是，当我们真正解决了问题的那一刻，是多么的欣喜若狂！是多么的舒畅爽快！那些抓耳挠腮的时光也顿时变得有意义起来。  
 这次期末大作业真的让我收获很多，不仅仅有编程能力的提升，更有全局观，逻辑思维方面更深入的理解。创作游戏时，框架思维贯穿始终，每一个功能都有它应该放置的位置，否则很有可能牵一发而动全身。从无到有地编写出一款游戏，我能感到满满的成就感，也能感到浓浓的内驱力，促使我谨听童老师教诲，深入学习，勤加探索，以做出最满意的成果！