**面向对象程序设计（C++）**

**课程设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 贪吃蛇游戏 | | |
| 日 期 | 2019年8月1日 至 2019年 9月 7 日 | | |
| 专 业 | 计算机科学与工程 | 班 级 | 计1802 |
| 姓 名 | 寇凯淇 | 学 号 | 20184446 |

报告成绩

|  |
| --- |
|  |

**报告**撰写的内容与要求

1. 项目简介：介绍本次课程设计选题的目的、意义、任务概况、本人在项目中的分工等内容。
2. 项目内容：系统的设计与实现的全面描述，介绍系统整体结构、系统框架图、UML类图、程序流程图、重点难点分析及解决方案、调试难点及解决方法、系统交互界面及结果展示等。本部分内容应以记叙或[白描手法](http://www.so.com/s?q=%E7%99%BD%E6%8F%8F%E6%89%8B%E6%B3%95&ie=utf-8&src=wenda_link" \t "_blank)为基调，在完整叙述的基础上，对自己认为有重要意义或需要研究解决的问题进行重点叙述，其它内容则可简述。
3. 总结或体会：对课设效果进行综合评价，着重介绍自身的收获与体会，内容较多时可列出[小标题](http://www.so.com/s?q=%E5%B0%8F%E6%A0%87%E9%A2%98&ie=utf-8&src=wenda_link" \t "_blank)，逐一列举。总结或体会的最后部分，应针对实习中发现的自身不足，简要地提出今后学习，努力的方向。
4. 报告正文一律采用计算机排版、A4纸**双面**打印，正文字体为**小四号**宋体，**1.35倍**行距，正文页数不低于10页，不超过30页（其中代码总计不超过3页，**中文字数**不少于4000字）。要求语句通顺、论述严谨、规范、正确。
5. 请注意封面页、扉页、评语页等的打印及装订顺序。分别为①封面页（其背面为扉页“实习报告的内容与要求”）、②目录、③正文、④其他附件（如有）。

# 项目简介--------------------------------------------------------------2页

1. 选题目的：----------------------------------------------------------------------------2页
2. 选题意义：----------------------------------------------------------------------------2页
3. 分工：-----------------------------------------------------------------------------------2页

# 项目内容------------------------------------------------------------2~8页

1. 系统的设计与实现的全面描述：---------------------------------------------------2页
2. 系统框架图：----------------------------------------------------------------------------3页
3. UML类图：-------------------------------------------------------------------------------4页
4. 程序流程图：----------------------------------------------------------------------------5页
5. 重点难点分析：----------------------------------------------------------------------5~6页
6. 解决方案：-------------------------------------------------------------------------------7页
7. 调试难点：-------------------------------------------------------------------------------7页
8. 解决方法：-------------------------------------------------------------------------------7页
9. 系统交互及结果界面：------------------------------------------------------------8~10页

# 总结与体会

1. 课设效果综合评价：------------------------------------------------------------------10页
2. 收获与体会：----------------------------------------------------------------------10~11页
3. 自身不足及今后努力方向：-------------------------------------------------------------11页

# 一、项目简介

1、选题目的：

贪吃蛇是家喻户晓的益智类小游戏，选择这个题目一是为了将自己的所学知识加以运用，二是一直以来贪吃蛇这个游戏就深深地吸引着我，它的制作方法对于以前的我而言都是很神秘的。我希望通过自己的所学知识把它剖析开来，真正的了解它的本质和精髓。虽然我的编程能力不是很强，但是我有信心，在这次学习中我将从实践和实际的项目中提高自己的编程能力。因此在小组讨论后共同决定选择了这个题目。

2、选题意义：

提高自身编程能力，通过开发游戏提高自己对编程的兴趣

1. 任务概况：

设计并实现贪吃蛇游戏系统；有欢迎与登录界面、说明界面和游戏界面。欢迎界面显示游戏的名称和游戏菜单，选择不同的菜单可以实现开始游戏、阅读游戏说明、按键说明及退出游戏功能。设计游戏地图、绘制蛇身及随机出现食物。实现贪吃蛇功能，即控制蛇的方向、设置蛇撞墙的情况。显示游戏积分、执行步数及所用时间。实现在失败界面进行选择（重玩一局或退出游戏）

5、分工：负责QT界面设计编写，地图类，豆子类，判定，文件读写。

# 二、项目内容

1、系统的设计与实现的全面描述：

本游戏包括启动器部分（QT），游戏部分（控制台）

启动器界面（QT）：

1. 主选单
2. 模式、皮肤、难度选单

（部分代码）

connect(&start,QPushButton::pressed,this,&MainWidget::Choose); connect(&rank,QPushButton::pressed,this,&MainWidget::ReadFile\_Rank); connect(&about,QPushButton::pressed,this,&MainWidget::ReadFile\_About);

connect(&rankback,QPushButton::pressed,this,&MainWidget::RankBack); connect(&aboutback,QPushButton::pressed,this,&MainWidget::AboutBack);

connect(&exit,QPushButton::pressed,this,&MainWidget::close);

// /选择模式 connect(&Mode1,QPushButton::pressed,this,&MainWidget::ChooseMode); connect(&Mode2,QPushButton::pressed,this,&MainWidget::ChooseMode); connect(&Mode3,QPushButton::pressed,this,&MainWidget::ChooseMode); connect(&back1,QPushButton::pressed,this,&MainWidget::BacktoMain)； // /选择难度 connect(&eazy,QPushButton::pressed,this,&MainWidget::ChooseDifficulty); connect(&normal,QPushButton::pressed,this,&MainWidget::ChooseDifficulty); connect(&hard,QPushButton::pressed,this,&MainWidget::ChooseDifficulty);

游戏界面（控制台）：

1）程序运行界面（显示界面）

2）操作界面（右侧）（包括信息统计与游戏选单）

①继续游戏

②重新开始

③结束游戏

游戏设置：

1. 蛇的控制（WASD，方向键）
2. 模式设置（经典模式、普通模式、街机模式）
3. 难度设置（低中高）
4. 积分（每次吃掉一个食物增加5分，并显示出来）

使用easyX来完成图案的绘制，使用QT制作界面。

（部分代码）

class Snake

{

private:

int len;

list<pos> body;

pos head;

int speed;

direction dir;

int blood;

int maxblood;

int past;

bool ifnext;//下一步是否有效,0无效,1有效

pos next;//下一步位置

int speedpast;

public:

bool havebean;

Snake(pos \_head, int \_blood, int \_maxblood, list<pos> &\_body, int \_speed, int \_len = 3, direction \_dir = d);

bool ifmove();//是否移动,移动时修改ifnext并记录下一格

//以下函数需要在ifmove（）使用后使用

void correctmove();//修正下一步超出地图的现象，修正返回1，否则返回0

void move();//将蛇在地图矩阵上进行修改

bool died(int key = 0);//判断是否死亡，返回1死亡，返回0存活

//bool eatfood();//判断是否吃到食物，吃到返回1，没吃到返回0

//void addlenth();//增加长度，在身体链表中增加一个头部

void eatfood();//吃豆子后改变蛇的数学

direction change\_direction();//从键盘上接受按键，改变方向

direction getdirection();//返回蛇现在的方向

void changespeed(int ss = 0);//改变速度，未做完

bool ifblood();//判断是否撞到可摧毁墙，并扣除血量。如果撞到返回1，否则返回0

int getblood();//返回血量

int addblood(int i = 1);//增加血量

pos gethead();

pos gettail();

pos getfirstbody();

list<pos> getbody();

pos getnext();

void changeblood(int bb);//改变血量

void changemaxblood(int bb);//改变最大血量

};

class Bean{

private:

pos BEAN;

public:

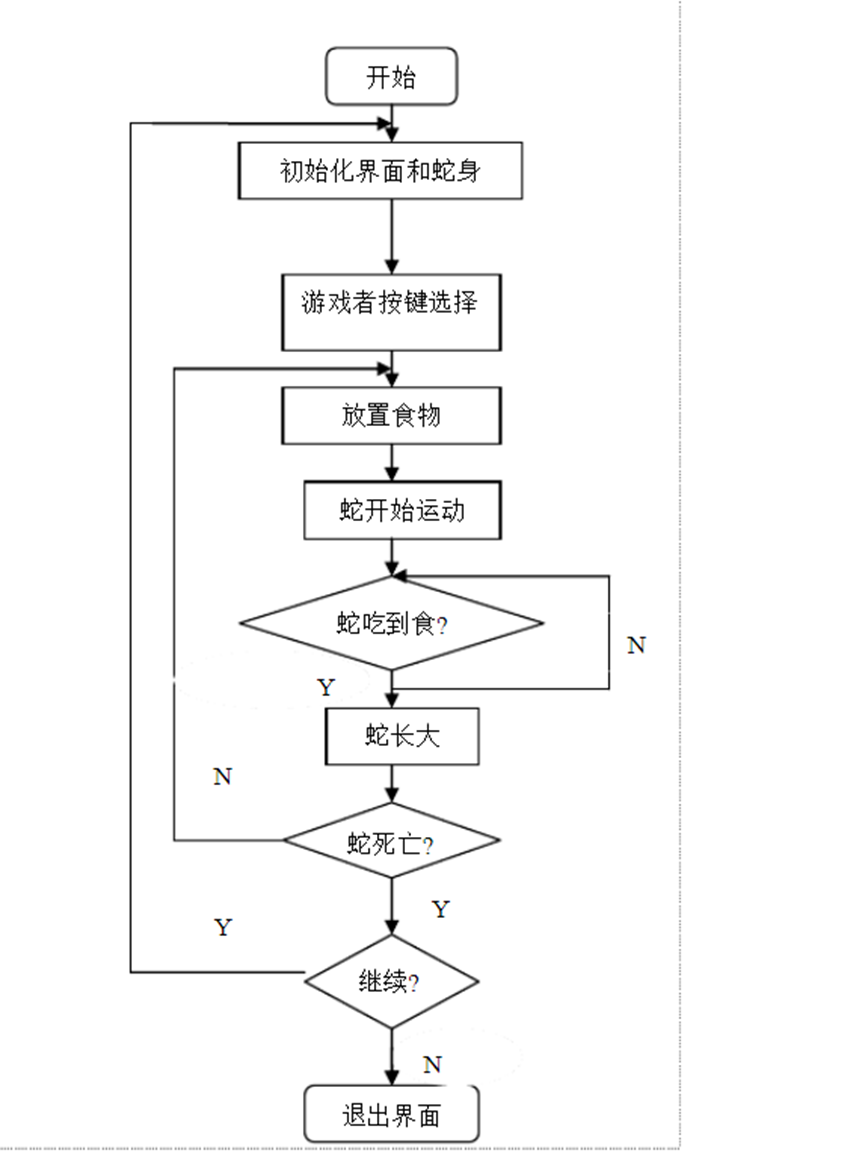
void Create();

void Print();

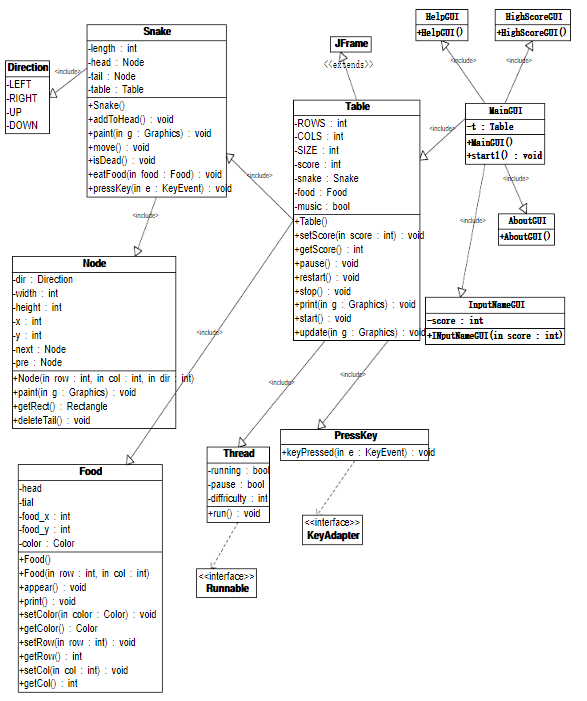
int nature=0; //0 普通 1 血+1

};

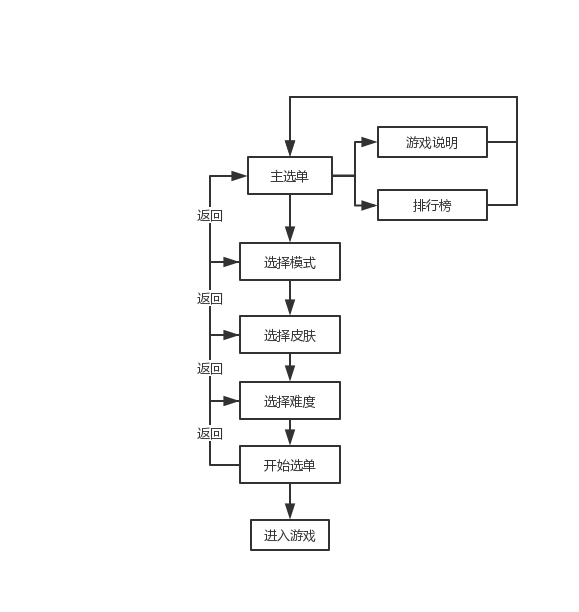
2、系统框架图：

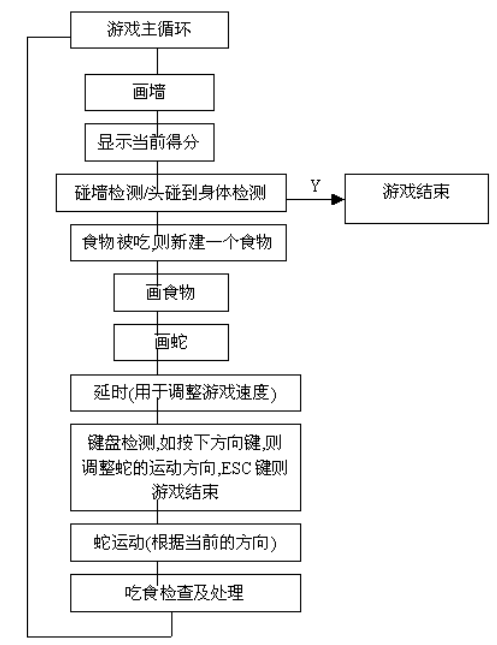


3、UML类图：



4、程序流程图：





5、重点难点分析：

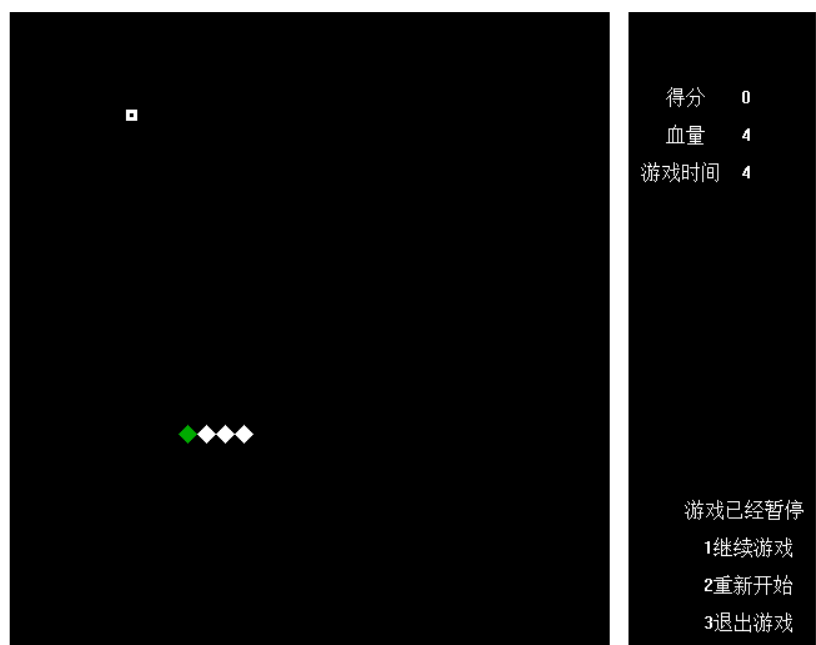
游戏界面，贪吃蛇的游戏界面包括墙的绘制，蛇身体的绘制，蛇移动范围的限制，食物的控制遗迹以及积分注释界面的绘制，其中难点是蛇身体的绘制、死亡后刷新像素点避免蛇的尸体残留、在蛇撞到墙上的时候避免蛇的身体覆盖墙体导致墙体消失等。这个程序的关键是表示蛇的图形以及蛇的移动。用一个小矩形表示蛇的一节身体，身体每长一节，增加一个矩形块。移动时必须从蛇头开始，所以蛇不能向相反方向移动，也就是蛇尾不能改作蛇头。如果不按任何键，蛇自行在当前方向上前移，当游戏者按了有效的方向键后，蛇头朝着指定的方向移动，一步移动一节身体，所以当按了有效的方向键后，先确定蛇头的位置，然后蛇身体随着蛇头移动，图形的实现是从蛇头的新位置开始画出蛇，这时原来蛇的位置和新蛇的位置差一个单位，所以看起来蛇会多一节身体，所以将蛇的最后一节用背景色覆盖。食物的出现和消失也是画矩形块和覆盖矩形块。

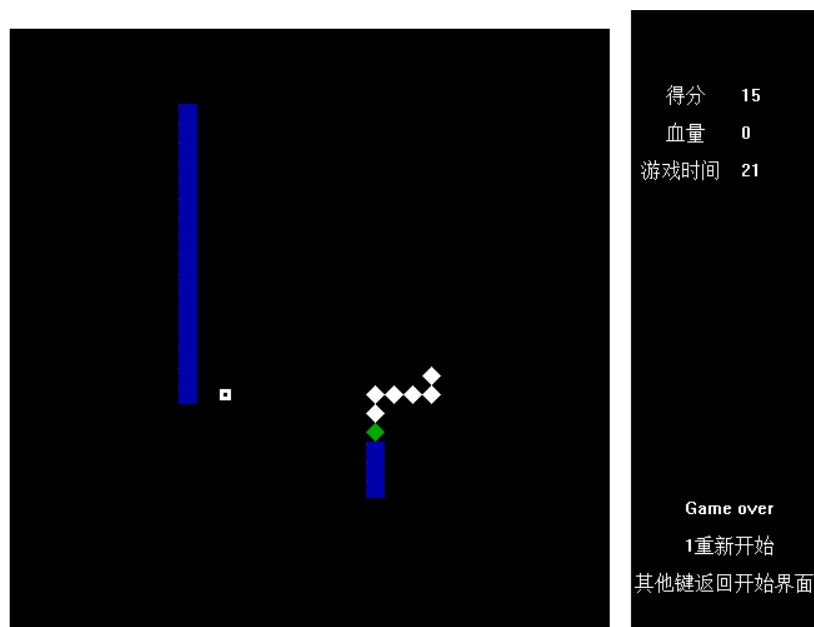
6、解决方案：用easyX绘制地图、蛇身、蛇头、以及食物，在蛇死亡后刷新界面确保不会出现bug，最终选择给地图的每个15\*15的像素点设置属性并依照属性填充颜色、形状，这样只要弄清楚填充颜色和清屏的先后顺序就避免了大部分覆盖问题。蛇的移动最后决定通过蛇头移动并判定是否吃到豆子，如果吃到则蛇尾不动，吃不到则蛇尾消失来体现移动，并依此设置了难度选择、地图选择、血量选择等。

7、调试难点：在优化游戏可玩性时选择为蛇增加血量并增加可以摧毁但是扣血的墙体，并设计了加血道具完善游戏性，但实践途中由于改变蛇的类的成员产生了问题。调试时链接动态符号库失败也困扰了我们很久

8、解决方法：采用友元的知识改变一部分私有成员使得蛇的更多属性可以被改变，同时通过改变编译器设置也解决了链接动态符库失败的问题

9、系统交互及结果界面：









# 三、总结与体会

1、课设效果综合评价：

游戏已经具备所需基本功能，能够进行正常的选单，游戏，记录，排名等功能，达到初步预期，在经典模式的贪吃蛇下，添加了血量与回血的小机制，增加了游戏的可游玩性。游戏程序语言C++，除了控制台程序外，添加了QT工具制作的选单界面以优化程序及界面。

2、收获与体会：

通过此次课程设计，我再次复习了所学习的C++语言知识并增加了了其应用的经验，是一次宝贵的经历，他填补了我在课堂上学习的漏洞，让我更加深刻的认识学习C++：1)在课程设计初期对程序的整体规划要有大体思路，不能走一步看一步，会增加执行的困难程度，使代码杂乱无章，缺少条理2)程序是由一个个类、一个个函数组成的，要合理划分，灵活利用C++所具备的多态重载等功能的优势，减少代码过度重复，减少工作量3)小组之间合作完工的要点在于加强相互之间的沟通，清楚互相之间的代码关联，统一按照初期计划进行并依照出现的问题合理及时沟通调整，避免小组成员之间代码冲突巨大，造成返工4)对于程序代码一定要标注好注释与信息，便于自己回顾与他人读解5）学会在网络上寻找信息与方法

3、自身不足及今后努力方向：

平时编程练手次数较少，编程能力较弱，编程的时候思绪跟不上，反应慢，算法掌握较少，课堂外的知识储备不足，这些都是我这次课程设计结束后所暴露出的问题。在之后要勤奋练习代码编程，保证思绪的敏捷与知识的长时间储备努力提高自己的编程能力，积极向他人学习，补充自己，完善自己。

**附录：**

(个人负责的代码)

//MAINWIDGET.H

#ifndef MAINWIDGET\_H

#define MAINWIDGET\_H

#include <QWidget>

#include <QPushButton>

#include <QProcess>

#include <QString>

#include <QLabel>

#include <QFont>

class MainWidget : public QWidget

{

Q\_OBJECT

public:

MainWidget(QWidget \*parent = 0);

~*MainWidget*();

/\*----------------函数------------------\*/

void Choose();

void ChooseMode();

void ChooseDifficulty();

//void ChooseMap();

void ChooseSkin();

void StartGame();

void OpenRank();

void OpenAbout();

void BacktoMain();

void BacktoMode();

void BacktoDifficulty();

//void BacktoMap();

void BacktoSkin();

void GameBack();

void RankBack();

void AboutBack();

void WriteFile\_Arg(QString arg);

void ReadFile\_About();

void ReadFile\_Rank();

/\*----------------程序------------------\*/

void OpenGame();

public slots:

/\*----------------槽函数----------------\*/

//void start\_game();

void Open();

void Finished(int);

private:

/\*----------------按钮------------------\*/

//界面1---主界面

QPushButton start;

QPushButton about;

QPushButton rank;

QPushButton exit;

//界面2---选择模式

QPushButton Mode1,Mode2,Mode3,back1;

//界面3---选择难度

QPushButton eazy,normal,hard,back2;

//界面4---选择地图

//QPushButton Map1,Map2,Map3,back3;

//界面5---选择皮肤

QPushButton Skin1,Skin2,Skin3,back4;

//界面6---开始！

QPushButton startGame,back5;

//界面7---排行榜

QPushButton rankback;

//界面8---游戏说明

QPushButton aboutback;

/\*----------------程序------------------\*/

QProcess \*Game\_exe=new QProcess(this);

/\*----------------文本------------------\*/

QLabel about\_txt;

QLabel rank\_txt;

};

#endif // MAINWIDGET\_H

//MAIN.H

#include "mainwidget.h"

#include <QApplication>

#include <QProcess>

#include <QPushButton>

#include <QDebug>

#include <QDir>

int main(int argc, char \*argv[])

{

QString path=argv[0];

qDebug()<<path;

//QDir::setCurrent(snakeapp.applicationDirPath());//添加这句 立马可以

QApplication snakeapp(argc, argv);

MainWidget mainwindow; //主窗口

mainwindow.setWindowTitle("贪吃蛇2019.7.22qt版"); //表替

mainwindow.setFixedSize(400,400); //设置窗口大小

mainwindow.show(); //打开窗口

snakeapp.exec();

return 733;

}

//MAINWIDGET.CPP

#include "mainwidget.h"

#include <QPushButton>

#include <QDebug>

#include <iostream>

#include <QString>

#include <string>

#include <time.h>

#include <windows.h>

#include <QProcess>

#include <QFile>

#include <QCoreApplication>

#include <QFont>

#include <windows.h>

using namespace std;

QString mode\_code="0";

QString skin\_code="0";

QString speed\_code="0";

MainWidget::MainWidget(QWidget \*parent)

: QWidget(parent)

{

/\*----------------按钮------------------\*/

start.setParent(this); //设置父类

about.setParent(this);

exit.setParent(this);

rank.setParent(this);

Mode1.setParent(this);

Mode2.setParent(this);

Mode3.setParent(this);

back1.setParent(this);

eazy.setParent(this);

normal.setParent(this);

hard.setParent(this);

back2.setParent(this);

Skin1.setParent(this);

Skin2.setParent(this);

Skin3.setParent(this);

back4.setParent(this);

startGame.setParent(this);

back5.setParent(this);

rankback.setParent(this);

aboutback.setParent(this);

Mode1.hide(); //隐藏按钮

Mode2.hide();

Mode3.hide();

back1.hide();

eazy.hide();

normal.hide();

hard.hide();

back2.hide();

Skin1.hide();

Skin2.hide();

Skin3.hide();

back4.hide();

startGame.hide();

back5.hide();

rankback.hide();

aboutback.hide();

start.setText("开始游戏"); //设置按钮名称

about.setText("游戏说明");

exit.setText("退出游戏");

rank.setText("排行榜");

Mode1.setText("经典模式");

Mode2.setText("普通模式");

Mode3.setText("街机模式");

back1.setText("返回");

eazy.setText("简单");

normal.setText("正常");

hard.setText("困难");

back2.setText("返回");

Skin1.setText("皮肤1");

Skin2.setText("皮肤2");

Skin3.setText("皮肤3");

back4.setText("返回");

startGame.setText("开始！");

back5.setText("返回");

rankback.setText("返回");

aboutback.setText("返回");

start.move(150,75); //定位

rank.move(150,150);

about.move(150,225);

exit.move(150,300);

Mode1.move(150,75); //定位

Mode2.move(150,150);

Mode3.move(150,225);

back1.move(150,300);

eazy.move(150,75); //定位

normal.move(150,150);

hard.move(150,225);

back2.move(150,300);

/\*Map1.move(150,75); //定位

Map2.move(150,150);

Map3.move(150,225);

back3.move(150,300);\*/

Skin1.move(150,75); //定位

Skin2.move(150,150);

Skin3.move(150,225);

back4.move(150,300);

startGame.move(150,150); //定位

back5.move(150,225);

rankback.move(0,0); //定位

aboutback.move(0,0);

connect(&start,QPushButton::pressed,this,&MainWidget::Choose); connect(&rank,QPushButton::pressed,this,&MainWidget::ReadFile\_Rank); connect(&about,QPushButton::pressed,this,&MainWidget::ReadFile\_About); connect(&rankback,QPushButton::pressed,this,&MainWidget::RankBack); connect(&aboutback,QPushButton::pressed,this,&MainWidget::AboutBack);

connect(&exit,QPushButton::pressed,this,&MainWidget::close);

// /选择模式 connect(&Mode1,QPushButton::pressed,this,&MainWidget::ChooseMode); connect(&Mode2,QPushButton::pressed,this,&MainWidget::ChooseMode); connect(&Mode3,QPushButton::pressed,this,&MainWidget::ChooseMode); connect(&back1,QPushButton::pressed,this,&MainWidget::BacktoMain);

// /选择难度 connect(&eazy,QPushButton::pressed,this,&MainWidget::ChooseDifficulty); connect(&normal,QPushButton::pressed,this,&MainWidget::ChooseDifficulty); connect(&hard,QPushButton::pressed,this,&MainWidget::ChooseDifficulty); connect(&back2,QPushButton::pressed,this,&MainWidget::BacktoMode);

// /选择皮肤 connect(&Skin1,QPushButton::pressed,this,&MainWidget::ChooseSkin); connect(&Skin2,QPushButton::pressed,this,&MainWidget::ChooseSkin); connect(&Skin3,QPushButton::pressed,this,&MainWidget::ChooseSkin); connect(&back4,QPushButton::pressed,this,&MainWidget::BacktoDifficulty);

// /选择开始 connect(&startGame,QPushButton::pressed,this,&MainWidget::OpenGame); connect(&back5,QPushButton::pressed,this,&MainWidget::BacktoSkin);

/\*----------------程序------------------\*/

connect(Game\_exe,SIGNAL(started()),this,SLOT(Open()));

connect(Game\_exe,SIGNAL(finished(int)),this,SLOT(Finished(int)));

}

/\*----------------按钮------------------\*/

void MainWidget::Choose()

{

start.hide();

rank.hide();

about.hide();

exit.hide();

Mode1.show();

Mode2.show();

Mode3.show();

back1.show();

}

void MainWidget::ChooseMode() //判断选择的模式

{

int str1,str2,str3;

str1=Mode1.isDown();

str2=Mode2.isDown();

str3=Mode3.isDown();

qDebug()<<"Mode:"<<str1<<str2<<str3;

if(str1==1)

mode\_code="1";

else if(str2==1)

mode\_code="2";

else if(str3==1)

mode\_code="3";

qDebug()<<mode\_code;

eazy.show();

normal.show();

hard.show();

back2.show();

Mode1.hide();

Mode2.hide();

Mode3.hide();

back1.hide();

}

void MainWidget::ChooseDifficulty() //判断选择的难度

{

int str1,str2,str3;

str1=eazy.isDown();

str2=normal.isDown();

str3=hard.isDown();

qDebug()<<"Dif:"<<str1<<str2<<str3;

if(str1==1)

speed\_code="1";

else if(str2==1)

speed\_code="2";

else if(str3==1)

speed\_code="3";

qDebug()<<speed\_code;

eazy.hide();

normal.hide();

hard.hide();

back2.hide();

Skin1.show();

Skin2.show();

Skin3.show();

back4.show();

}

void MainWidget::ChooseSkin() //判断选择的皮肤

{

int str1,str2,str3;

str1=Skin1.isDown();

str2=Skin2.isDown();

str3=Skin3.isDown();

qDebug()<<"Skin:"<<str1<<str2<<str3;

if(str1==1)

skin\_code="1";

else if(str2==1)

skin\_code="2";

else if(str3==1)

skin\_code="3";

qDebug()<<skin\_code;

Skin1.hide();

Skin2.hide();

Skin3.hide();

back4.hide();

startGame.show();

back5.show();

}

void MainWidget::OpenRank(){}

void MainWidget::BacktoMain() //返回按钮

{

start.show();

rank.show();

about.show();

exit.show();

Mode1.hide();

Mode2.hide();

Mode3.hide();

back1.hide();

}

void MainWidget::BacktoMode()

{

eazy.hide();

normal.hide();

hard.hide();

back2.hide();

Mode1.show();

Mode2.show();

Mode3.show();

back1.show();

}

void MainWidget::BacktoDifficulty()

{

eazy.show();

normal.show();

hard.show();

back2.show();

Skin1.hide();

Skin2.hide();

Skin3.hide();

back4.hide();

}

void MainWidget::BacktoSkin()

{

Skin1.show();

Skin2.show();

Skin3.show();

back4.show();

startGame.hide();

back5.hide();

}

//void MainWidget::GameBack(){}

void MainWidget::AboutBack()

{

start.show();

rank.show();

about.show();

exit.show();

aboutback.hide();

about\_txt.hide();

}

void MainWidget::RankBack()

{

start.show();

rank.show();

about.show();

exit.show();

rankback.hide();

rank\_txt.hide();

}

/\*----------------程序------------------\*/

void MainWidget::ReadFile\_About() //文件打开

{

QFont font\_about;

font\_about.setPointSize(13);

about\_txt.setParent(this);

start.hide();

about.hide();

exit.hide();

rank.hide();

about\_txt.move(0,28);

about\_txt.setFont(font\_about);

QFile file("./txt/About.txt");

file.*open*(QIODevice::ReadOnly | QIODevice::Text);

QByteArray t = file.readAll();

QString txt;

txt=QString(t);

this->about\_txt.setText(txt.toUtf8());

about\_txt.show();

file.*close*();

aboutback.show();

}

void MainWidget::ReadFile\_Rank()

{

QFont font\_rank;

font\_rank.setPointSize(15);

rank\_txt.setParent(this);

start.hide();

about.hide();

exit.hide();

rank.hide();

rank\_txt.move(0,28);

rank\_txt.setFont(font\_rank);

QFile file("./txt/Rank.txt");

file.*open*(QIODevice::ReadOnly | QIODevice::Text);

QByteArray t = file.readAll();

QString txt;

txt=QString(t);

this->rank\_txt.setText(txt.toUtf8());

rank\_txt.show();

file.*close*();

rankback.show();

}

void MainWidget::WriteFile\_Arg(QString arg)

{

QFile file("./txt/arg.txt");

file.*open*(QIODevice::WriteOnly | QIODevice::Text);

file.write(arg.toUtf8());

file.*close*();

}

void MainWidget::OpenGame()

{

QString argg;

argg=mode\_code+"\n"+skin\_code+"\n"+speed\_code;

this->WriteFile\_Arg(argg);

time\_t \_temp=clock();

while(clock()-\_temp<1000);

Game\_exe->startDetached("./exe/Snake.exe");

Game\_exe->waitForFinished(500);

close();

qDebug()<<Game\_exe->exitCode();

// int t=time(NULL);

// while((time(NULL)-10)<=t);

qDebug()<<"game should be opened";

}

void MainWidget::Open() //槽函数信号函数

{

qDebug()<<"game has been opened!!!";

}

void MainWidget::Finished(int exitcode)

{

qDebug()<<exitcode;

if(exitcode==1319)

{

qDebug()<<"game has been closed!!!"<<endl<<exitcode;

this->show();

}

}

MainWidget::~*MainWidget*()

{

}

//BEAN.H

#pragma once

#ifndef BEAN\_H

#define BEAN\_H

#include "all\_head.h"

extern COLORREF bean[rate][rate];

class Bean{

private:

pos BEAN;

public:

void Create();

void Print();

int nature=0; //0 普通 1 血+1 2

};

extern Bean beanfood;

#endif

//MAP.H

#pragma once

#ifndef MAP\_H

#define MAP\_H

#include "all\_head.h"

//-----------//

const int mapsize = 34;

const int rate = 15;

//0表示空白，1蛇的身体，2蛇的头部，3豆子，4可摧毁墙，5不可摧毁墙

extern COLORREF snakebody[rate][rate];

extern COLORREF snakehead[rate][rate];

extern COLORREF bean[rate][rate];

extern COLORREF wall\_distructible[rate][rate];

extern COLORREF wall\_indistrutible[rate][rate];

extern COLORREF black[rate][rate];

class Map {

public:

int mapp[mapsize][mapsize];

void show();

};

extern Map map[3];

extern pos p\_next, p\_head, p\_tail;

///////////////////函数////////////////////////

void delay(int t);

void Print\_snake(pos p\_next, pos p\_head);

void Print\_snake(pos p\_next, pos p\_head, pos p\_tail);

void Judge(Snake &snake);

void pause();

struct Rank {

string name;

int score = 0;

int time = 0;

};

void SignInRank();

#endif

//BEAN.CPP

#include "all\_head.h"

using namespace std;

Bean beanfood;

int Random()

{

int r;

//srand((int)time(0));

r = rand() % 34;

return r;

}

void Bean::Print()

{

print(bean, BEAN);

}

void Bean::Create()

{

if (total.score % 25 == 0 && total.score != 0)

{

nature = 1;

}

BEAN.x = Random();

BEAN.y = Random();

if (map[1].mapp[BEAN.x][BEAN.y] == 0)

{

map[1].mapp[BEAN.x][BEAN.y] = 3;

Print();

}

else

{

Create();

}

}

//MAP.CPP

#include "all\_head.h"

using namespace std;

COLORREF snakebody[rate][rate];

COLORREF snakehead[rate][rate];

COLORREF bean[rate][rate];

COLORREF wall\_distructible[rate][rate];

COLORREF wall\_indistrutible[rate][rate];

COLORREF black[rate][rate];

Map map[3];

int t = 0;

void Map::show() //void print(COLORREF photo[][\_proportion], pos p)//在p点打印图案

{

clock\_t t1,t2;

cleardevice();

BeginBatchDraw();

for (int i = 0; i < 34; i++)

{

for (int j = 0; j < 34; j++)

{

if (mapp[i][j] == 0)

{

continue;

}

else if (mapp[i][j] == 1)

{

pos p;

p.x = i;

p.y = j;

print(snakebody, p);

}

else if (mapp[i][j] == 2)

{

pos p;

p.x = i;

p.y = j;

print(snakehead, p);

}

else if (mapp[i][j] == 3)

{

pos p;

p.x = i;

p.y = j;

print(bean, p);

}

else if (mapp[i][j] == 4)

{

pos p;

p.x = i;

p.y = j;

print(wall\_distructible, p);

}

else if (mapp[i][j] == 5)

{

pos p;

p.x = i;

p.y = j;

print(wall\_indistrutible, p);

}

}

}

t1 = clock();

EndBatchDraw();

t2 = clock();

cout << t2 - t1 << endl;

}

void delay(int t)

{

while (t--);

}

pos p\_next, p\_head, p\_tail;

void Print\_snake(pos p\_next, pos p\_head)

{

//eatbean

map[1].mapp[p\_next.x][p\_next.y] = 2;

map[1].mapp[p\_head.x][p\_head.y] = 1;

print(snakehead, p\_next);

print(snakebody, p\_head);

}

void Print\_snake(pos p\_next, pos p\_head, pos p\_tail)

{

map[1].mapp[p\_next.x][p\_next.y] = 2;

map[1].mapp[p\_head.x][p\_head.y] = 1;

map[1].mapp[p\_tail.x][p\_tail.y] = 0;

print(snakehead, p\_next);

print(snakebody, p\_head);

print(black, p\_tail);

}

void Judge(Snake &snake)

{

void SignInRank();

if (map[1].mapp[p\_next.x][p\_next.y] == 0)

{

Print\_snake(p\_next, p\_head, p\_tail);

return;

}

else if (map[1].mapp[p\_next.x][p\_next.y] == 5|| map[1].mapp[p\_next.x][p\_next.y] ==1)

{

gameover();

}

else if (map[1].mapp[p\_next.x][p\_next.y] == 3)

{

if (beanfood.nature == 1)

{

snake.addblood();

beanfood.nature = 0;

}

Print\_snake(p\_next, p\_head);

beanfood.Create();

snake.havebean = 1;

total.addsocre();

total.print(snake);

return;

}

else if (map[1].mapp[p\_next.x][p\_next.y] == 4)

{

snake.ifblood();

total.print(snake);

Print\_snake(p\_next, p\_head, p\_tail);

if (snake.died() == 1)

gameover();

}

}

string TCHARTurnString(TCHAR \*STR)

{

int iLen = WideCharToMultiByte(CP\_ACP, 0, STR, -1, NULL, 0, NULL, NULL);

char\* chRtn = new char[iLen \* sizeof(char)];

WideCharToMultiByte(CP\_ACP, 0, STR, -1, chRtn, iLen, NULL, NULL);

string str(chRtn);

return str;

}

void SignInRank()

{

Rank rankFile[6];

TCHAR name[16];

InputBox(name, 16, \_T("请输入用户名(英文字符15个以内)"));

//outtext(name);

//closegraph();

string user\_name;

user\_name = TCHARTurnString(name);

rankFile[5].name = user\_name;

rankFile[5].score = total.score;

rankFile[5].time = total.alltime;

ifstream infile("./txt/Rank.txt", ios::in);

char title[6];

infile >> title;

//cout << title;

for (int i = 1; i <= 5; i++)

{

int n;

char No[5];

char useless[3][10];

string name;

int score, time;

infile >> No >> n >> useless[0] >> name >> useless[1] >> score >> useless[2] >> time;

//cout << No << n << useless[0] << name << useless[1] << score << useless[2] << time;

rankFile[n-1].name = name;

rankFile[n - 1].score = score;

rankFile[n - 1].time = time;

}

infile.close();

for (int i = 4; i >= 0; i--)

{

if (rankFile[i + 1].score < rankFile[i].score)

{

break;

}

else if (rankFile[i + 1].score == rankFile[i].score)

{

if (rankFile[i + 1].time >= rankFile[i].time)

{

break;

}

else

{

swap(rankFile[i + 1], rankFile[i]);

}

}

else

{

swap(rankFile[i + 1], rankFile[i]);

}

}

ofstream outfile("./txt/Rank.txt", ios::out);

outfile << title << endl;

for (int i = 0; i < 5; i ++)

{

outfile << "No " << i + 1 << endl;

outfile << "Name: " << rankFile[i].name << " Score: " << rankFile[i].score << " Time: " << rankFile[i].time << endl;

}

outfile.close();

}