# 软件说明

## 游戏使用说明

这款游戏讲述了一个小冰块误入险境，面对着各种危险，需要玩家操纵小冰块躲避障碍获取更高的分数。打开游戏后，会来到一个主界面，可以通过键盘选择查看帮助、开始游戏、关于。其中查看帮助界面可以看到详细的操作信息，关于界面有作者的详细信息。请注意使用**英文输入法**。进入游戏后，可以使用鼠标或者键盘进行操作。使用WSAD来控制小冰块上下左右移动，或者通过鼠标右键设定小冰块需要到达的地点，小冰块会沿直线走到此地点。如果按住鼠标右键不放，小冰块会跟随鼠标右键的地点。小冰块总共有3点生命值，生命值到0点时小冰块死亡，进入结算界面。得分数会随着存活时间增长而增加。随着分数的提高，难度会逐渐增大，火球会更加精准。当小冰块遇到火球时会扣除1点生命值。小冰块遇到喷发的岩浆池时会扣除1点生命值，岩浆池在暗淡的时候不会对小冰块造成伤害。小冰块遇到蓝心时会回复1点生命值。如果小冰块离开了屏幕，则会进入彩蛋界面，此时再离开一次屏幕则会回到游戏。

## 源码编译说明

使用MinGW编译，g++版本为g++ (x86\_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0。先使用命令gcc -c \*.c和g++ -c \*.cpp将所有C以及C++代码编译为目标文件。再使用g++ \*.o -mwindows -lgdi32 -lole32 -loleaut32 -luuid -lwinmm -lmsimg32 -o game.exe 编译出可执行文件，名字为game.exe。其中-mwindows用来隐藏命令窗口，-lgdi32 -lole32 -loleaut32 -luuid -lwinmm -lmsimg32用来链接需要的库文件。

# 系统设计

首先要创建类，经过分析可以得到以下的表格。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类名 | 基类 | 火球 | 岩浆 | 蓝心 | 冰块 |
| 继承自 |  | 基类 | 基类 | 火球 | 基类 |
| 变量 | 坐标、长宽、图片 | 速度、方向 | 时间 |  | 速度、方向、目标点、得分、生命 |
| 函数 | 返回坐标、移动坐标、更新状态、画图 | 返回方向、加速、更新状态 | 返回时间 | 更新状态 | 更新状态、移动坐标、重置、增加和返回得分、降低速度、加减生命 |

将游戏中的各个对象封装成了一个一个的类，其中火球继承自基类，蓝心继承自火球，冰块继承自基类，岩浆继承自基类。它们各自包含了特有的参数和函数，也使用或者覆盖了父类的函数。

因为每个类都有坐标、长宽、图片的参数，所以基类有这些变量，同时也需要返回和移动坐标，随着时间更改状态。其中移动坐标和更改状态是其虚函数，因为其他类有不同的修改方式。

火球继承自基类，有自己的速度和方向变量，在创建火球时便会随机给一个方向，随着难度提升会逐渐朝向玩家，速度也会进行变化。同时要有返回参数的函数，以及自己的更新状态的函数，以及增加速度的函数。更新状态通过三角函数、方向、速度。

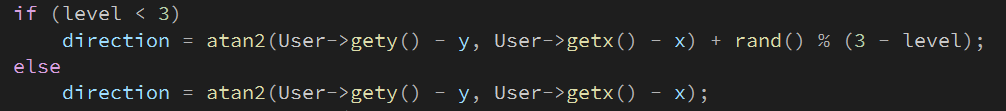
岩浆因为是固定的，所以直接继承自基类。岩浆的喷发需要一定的时间，因此需要记录创建岩浆的时间，通过创建时间和现在时间进行对比确认岩浆是否喷发，所以增加了时间变量。相应的在函数里面增加了返回时间。

蓝心作为恢复生命的道具，与火球有着相似的性质，因此继承自火球。

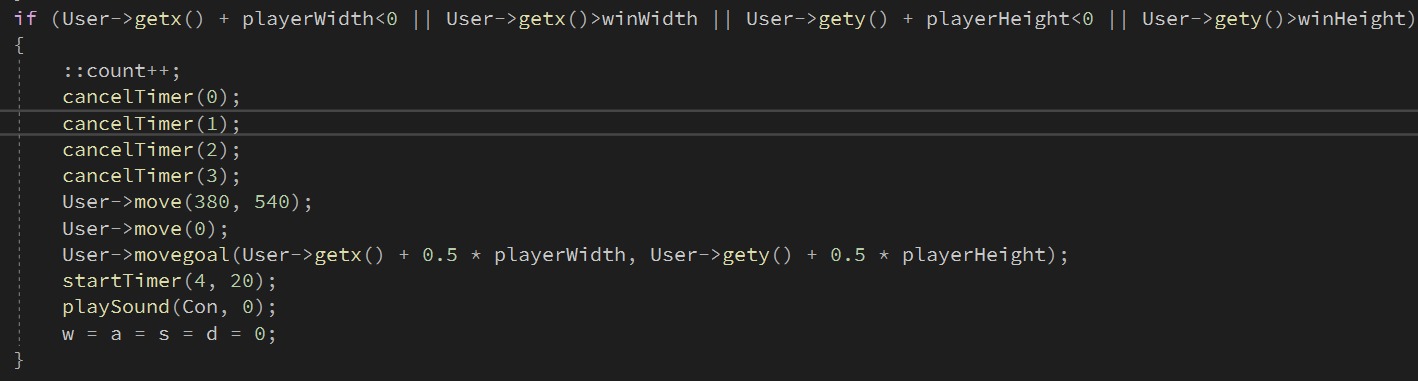
冰块作为玩家，有着得分、生命等变量，也因此有了关于这些的函数。因为有2套对于冰块的控制方法，因此设立全局变量iskeyboard进行判断。

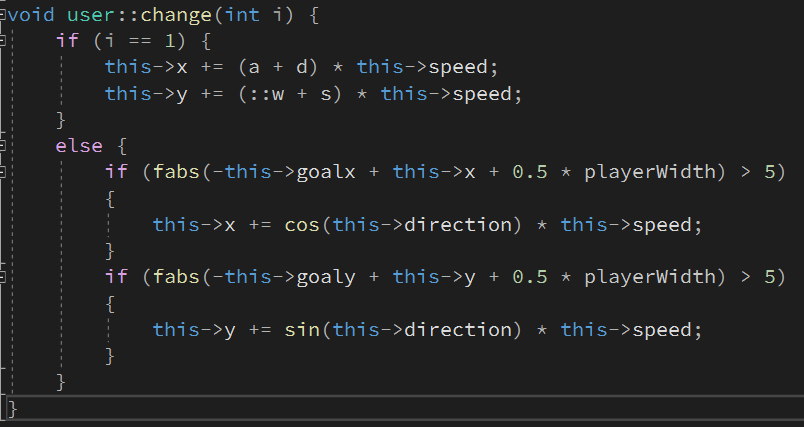
为了函数的简洁，单独设置不同名称的函数disp1等进行不同菜单界面的切换，设置不同的函数KeyEvent1等进行不同界面的按键获取。

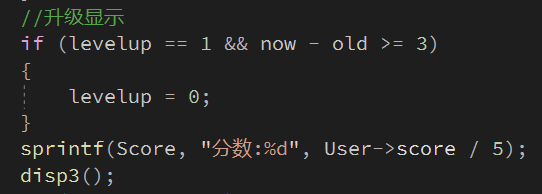
# 程序实现

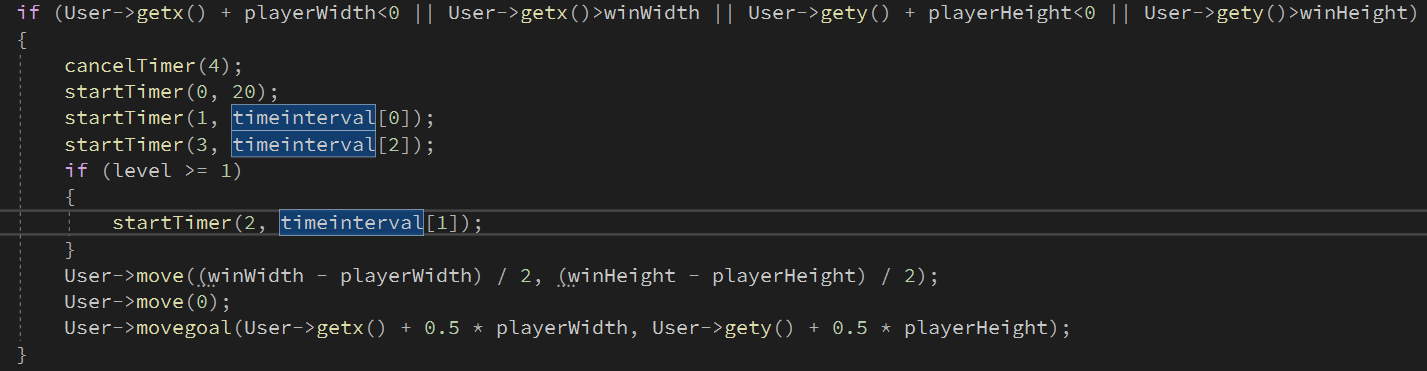
在创建火球的函数中，在随机生成了火球的出现位置后，使用如下代码

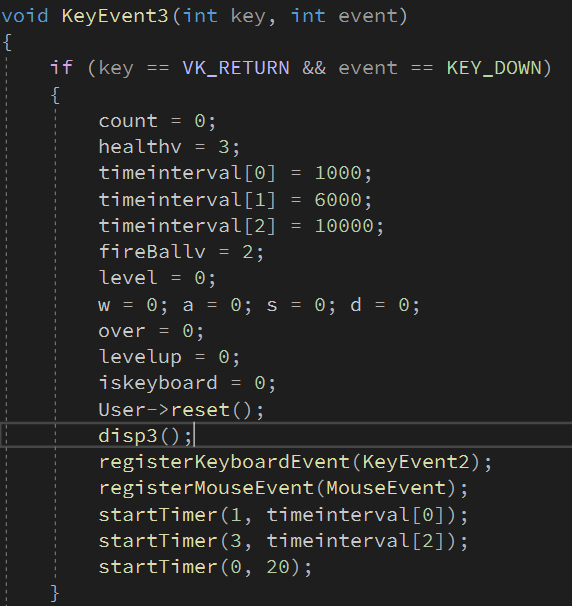
根据等级判断火球的精准度，在低等级时为了给玩家熟悉操作时间，额外增加了偏移值，在高等级时让火球直接朝向玩家当前位置，增加难度。

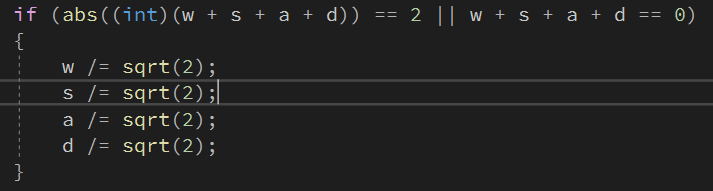
hitAdelete()函数（main.cpp第127行）用于碰撞检测和改变。它遍历存放障碍物的数组，如果障碍物存在，则判断是否与玩家碰撞，如果碰撞则执行相应的函数。例如火球是扣除玩家生命值，播放受伤的音效，删除此火球。如果障碍物到了屏幕的外面，则调用删除函数删除障碍物。如果检测到玩家离开了屏幕，则触发彩蛋功能，彩蛋计数器加1，同时取消游戏的定时器，将玩家的位置移动到特定地点，将目标位置和方向重置，防止玩家自动移动。同时播放音乐。

冰块的更新状态函数需要传入参数判断此时的控制方法，其中在用鼠标控制的方法中，为了防止冰块在一个单位游戏时间中远离目标位置造成的反复抖动现象，增加了限制条件，允许冰块中心与实际目标位置有5个像素点的差距。W变量因为有重名现象，因此使用::来限定范围。

在定时器函数中（timeevent.cpp），如果在游戏中，就先将所有障碍物以及玩家的位置更新，其中检测岩浆存在时间，若超过则删除岩浆。然后调用hitAdelete()函数进行碰撞检测，再增加玩家的分数，判断玩家的分数是否到达升级条件，满足则进行难度的增加。同时进行升级图标的显示，持续3秒。然后输出分数和更新画面。再检测玩家是否死亡。

其余的定时器为简单的创建障碍物，最后一个定时器用于彩蛋界面。在离开彩蛋界面的时候，要对玩家的各参数进行重置，同时开启游戏中必要的定时器。

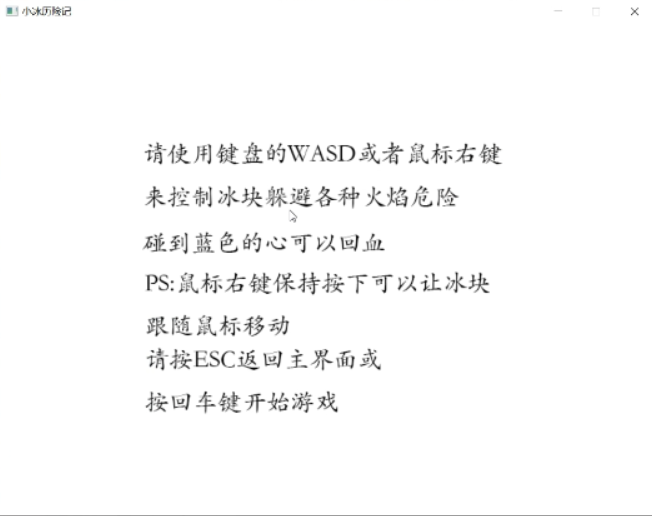
在游戏结束后的界面，可以选择重新开始或者返回主菜单。此时需要将参数重置回初始状态。这在KeyEvent3中完成。

玩家有两套控制方案，为了识别玩家使用的方案，在MouseEvent以及KeyEvent中需要随时进行调整。在KeyEvent中，如果玩家同时按下两个按键，会导致速度的加快。因此设置以下判断条件进行速度的调节。

在MouseEvent中，要针对玩家选择按住右键以及按下右键的方案进行判断，增加flag标志。当玩家按住右键时，检测鼠标移动的事件，以此更改玩家的位置。当玩家按下右键并且弹起时，直接将冰块的目标位置改为鼠标当前的位置。

# 测试报告

打开程序

按下H查看帮助，正常

按下ESC返回主菜单，正常

按下A查看关于，正常

按下ESC返回菜单，正常

在菜单处按下回车键，正常进入游戏

查看右键按住操作功能，正常

查看右键单击操作功能，正常

使用WASD操作，正常。

触碰火球后正常掉血，同时播放受伤的音效

升级图标显示正常

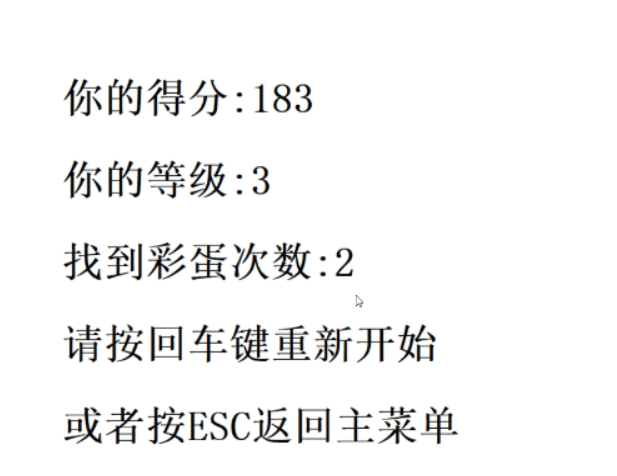
第一阶段的岩浆生成正常，并且走上去不会掉血

第二阶段岩浆变化正常，并且走上去会掉血

触碰蓝心检测回血情况，正常

菜单界面进入正常，并且不操作不会乱动。

离开彩蛋界面游戏照常运行

死亡后正确结算分数

在结算界面可以正常重新开始游戏，也可以正常回到主菜单。在帮助界面和关于界面也可以直接进入游戏。

# 实验心得

在调试的过程中，我发现一旦使用按键操作冰块，冰块就会极快的往下走，进入彩蛋界面。我开始对此很疑惑，然后对改变冰块位置的函数打了断点，发现控制移动方向的W变量和自己的宽度变量名字相同，在类中默认使用类中的变量名，才导致了这个问题，在更改为::w后，程序正常运行。

在刚开始编写类的时候，我对于各种类之间的关系没有理解清楚，所以对于一些可以继承使用的函数，以及是否需要虚函数等问题不清楚，造成了编程时的反复更改，这种更改浪费时间。因此我后来自己思考了各个类的关系后，效率大幅度提升。

在使用鼠标操作冰块的设计中，我没有考虑到冰块无法精准的移动到目标位置，因此判断的条件过于精确，造成了冰块在目标区域附近反复抽搐的问题，在增加了5个像素点的误差后冰块可以正常移动。

附：评分标准

# 软件说明（20分）

## 游戏使用说明

详细介绍游戏的基本功能、打开方式、使用方法等信息。一个未接触过本实验的用户阅读完游戏使用说明后，能够轻松上手玩游戏，并了解游戏全部功能。

A挡：8~10分

B档：5~7分

C档：0~4分

## 源码编译说明

详细描述源代码的编译方法，如何从提交的实验代码（\*.cpp、\*.c、\*.h、……）编译出最终的可执行程序（\*.exe）。如果使用图形化编译器（如VS），要有关键步骤的截图，把创建工程、编译、运行等过程描述清楚。如果使用命令行编译器（如MinGW），要把每一步使用的命令、参数、输入与输出等信息描述清楚。一个有基本编程经验的用户根据源码编译说明中的步骤，能够轻松上手把源代码编译成最终的可执行程序。

A挡：8~10分

B档：5~7分

C档：0~4分

# 系统设计（20分）

用文字、图形、表格、伪代码等方式，对程序各项功能的设计思想进行清楚的描述。重点描述各项功能涉及的类、类的成员、不同类之间的关系等，充分体现出封装、继承和多态的思想。注意：关键在于对设计思想进行描述，不得大量拷贝粘贴源代码。

A挡：16~20分

B档：10~15分

C档：0~9分

注：连续拷贝粘贴源代码，每超出1页扣1分。

# 程序实现（30分）

用文字对程序各项功能涉及的关键文件、函数、代码等进行说明，详细分析它们是“做什么”和“怎么做”的。注意，关键在于对实现细节进行说明，不得大量拷贝粘贴源代码。

A挡：16~20分

B档：10~15分

C档：0~9分

注：连续拷贝粘贴源代码，每超出1页扣1分。

# 测试报告（20分）

对程序各项功能进行详细的测试，测试内容覆盖到全部功能点。对测试过程、测试方法和测试结果进行详细的描述，图文并茂。对测试结果进行分析，得出程序是否实现了预期功能的结论。

A挡：16~20分

B档：10~15分

C档：0~9分

# 实验心得 （10分）

描述实验过程中的问题和收获。着重描写你在实验过程中遇到的问题，你是如何解决这些问题的，目前还存在哪些问题。不仅仅限于编程上的问题，在分析、设计、测试过程中遇到的问题都算。你从解决问题的过程中学习到了什么。

A挡：8~10分

B档：5~7分

C档：0~4分