



**Object-oriented Programming**

# **面向对象程序设计**

**User Manual & Project II Report**

**用户手册&大程报告**

3180102829 数字媒体技术 谢博洋

# 目录

- 一、 游戏介绍..... 3
  - 1. 游戏背景..... 3
  - 2. 环境设置..... 3
  - 3. 关卡设置..... 4
    - a. 城堡的层数、每层的格局，由用户输入信息决定。..... 4
    - b. 房间的种类：..... 4
    - c. 游戏玩法展示：..... 5
      - \* a/g 模式的提示..... 6
      - \* 正常行进的提示..... 7
      - \* 行进失败的提示..... 7
      - \* NORMAL ROOM 的提示..... 8
      - \* 普通状态下 SPECIAL ROOM 的提示..... 8
      - \* 上帝视角下 SPECIAL ROOM 的提示..... 9
      - \* 进入公主房的提示..... 11
      - \* 进入怪物所在房间的提示（上帝视角）..... 12
      - \* 回到大厅的提示..... 14
      - \* 成功的提示..... 15
  - 4. 隐藏攻略和提示..... 16
    - a. 计算 monster 的位置..... 16
    - b. 隐藏的面包..... 16
    - c. 血量的上限..... 16
- 二、 问题和解决过程..... 17
  - a. 实现阶段的问题..... 17
  - b. 试玩用户反馈..... 23
- 三、 感受..... 24

# 一、游戏介绍

## 1.游戏背景

---

从前有一个痴情的王子（?），他心爱的公主被怪兽掳去怪兽的城堡，王子痛定思痛，痛下决心，苦练剑术。数十年如一日，白发苍苍的他来到城堡脚下，决定与怪兽决一死战，抢回自己的公主。

但他已经年老体衰，不能硬拼，所以能躲避沉睡的怪兽为上策。在阴冷的城堡里，年老的王子四处游走，凭着气味猜测着怪兽的方向。他手中捏着自己的剑，寒光仿佛在斑驳的墙面上留下了痕迹。

在城堡里，有的房间堆满了尸骨，桌面上残留着几片面包，吃还是不吃？这是一个问题。更加令人懊恼的是，怪兽像是过分胸有成竹，在一些房间里为王子准备了随机恢复体力的药物 ... ..

## 2.环境设置

---

开发工具：CLion

运行环境：Win7 及以上

### 3. 关卡设置

---

a. 城堡的层数、每层的格局，由用户输入信息决定。

例如输入 3 4 4 即生成一个三层，每层四乘四大小的城堡。

b. 房间的种类：

\* HALL: 城堡的入口和出口。首次进入城堡，player 将被预置在 HALL，游戏中回到 HALL 时会询问玩家决定放弃并离开城堡还是路过 HALL 继续游走。

\*NORMAL ROOM: 堆满尸骨但有一些食物（bread）。选择吃掉，可以随机回血，如果不吃，面包将被保留。

\*SPECIAL ROOM: 怪兽的炫耀房间。有一道题目，有一定概率猜对，猜对能够获得固定的血。如果猜错，当前版本不会掉血。

\* NPC ROOM: 有公主/怪兽的房间。王子活着进去，怎么出来要分情况。

\*每个楼层的第一间房间 可以实现上楼下楼，即每个  $[i, 0, 0]$  房间可以接收上下楼指令。

### c. 游戏玩法展示:

#### \* 开始界面

提示初始化成功，表达欢迎，列出方向键（支持大小写通用）和警告，用户初始血槽状态是 60 点血。用户可以通过输入 a 来选择普通模式或者 g 来选择无敌上帝视角氪金模式。

```
Init Rooms :  !FINISHED!
Init Player :  !FINISHED!
=====
blood:  60 |  monster_distance: 0 end
floor: 0 x: 0 y: 0 end
=====

=====
|                                     |
|                               WELCOME                               |
|                                     |
|=====|
| *directions*                    |
| u: upstairs      d:downstairs   |
| l: left          r:right        |
| f: forward       b:backward     |
|=====|
| *modes*                    |
| normal(a)    god mode(g)      |
|=====|
| uppercase letters are allowed |
|=====|
|                               !CAUTION!                               |
| DO NOT TRY WRONG DIRECTIONS. |
|                               --monster                               |
|=====|
| input your choice with a or g |
|=====|
```

## \* a/g 模式的提示

```
a
you chose the normal mode.

now set your BUILDING. input the number of the floor, x and y.
it's suggested that numbers should not be larger than 10.
e.g. 4 5 6
the example above means a building with 4 floors ,and 5 X 6 = 30 rooms on EACH floor.
now set your BUILDING
=====
2 3 3
Fine! your building has 2 floors and 3 X 3rooms on each floor.
you can input a direction to start now. GOOD LUCK!

if you want to kill the monster, you need 400 blood!
Init Monster : !FINISHED!
Init Princess : !FINISHED!
```

```
g
please connect the programmer.
DONATE 5 RMB to get a god mode!
If finished donating, input the key:
clvusddfvuafvrgbjkr
WRONG KEY! CHECK AGAIN.
If finished donating, input the key:
707070
you chose the god mode.
now set your BUILDING. input the number of the floor, x and y.
it's suggested that numbers should not be larger than 10.
e.g. 4 5 6
the example above means a building with 4 floors ,and 5 X 6 = 30 rooms on EACH floor.
now set your BUILDING
=====
2 3 3
Fine! your building has 2 floors and 3 X 3rooms on each floor.
you can input a direction to start now. GOOD LUCK!

if you want to kill the monster, you need 40 blood!
Init Monster : !FINISHED!
Init Princess : !FINISHED!
princess position: 0 1 0 end
monster position: 1 2 2 end
```

可以看出，普通模式质朴真诚，氪金模式低调奢华。

**普通模式：**支持用户设定城堡规格；支持怪物血槽提示。

**氪金模式：**首先氪金之后会获得 key 密钥，输入错了会提示您一直输入。这一版的密码是 707070，通过密钥检查之后进入上帝视角，系统会提示本次公主和怪物所在房间号。

## \* 正常行进的提示

```
b
Current Situation:
=====
blood: 80 | monster_distance: 3 end
floor: 0 x: 0 y: 1 end
=====
```

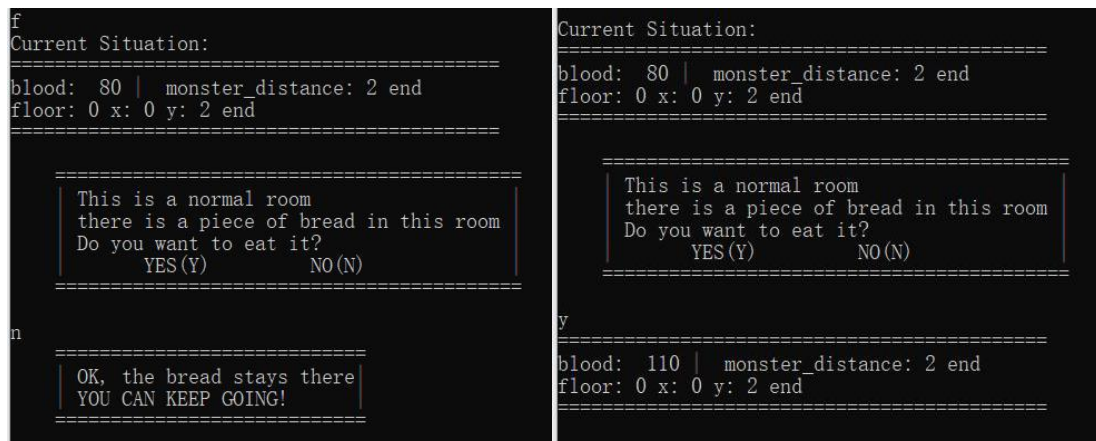
分别是：目前血槽状态 | 怪兽的折线距离，玩家目前的楼层数和位置信息。折线距离指王子到怪兽之间最短的可以通过直角或平角移动就能到达的距离值。例如，怪兽在一楼 2,2 房间，玩家在二楼 3,3 房间，折线距离就是：1+1+1=3.

## \* 行进失败的提示

```
f
=====
|      !CAUTION!      |
| NO ROOMS AHEAD     |
|      --monster     |
|=====|
Current Situation:
=====
blood: 110 | monster_distance: 2 end
floor: 0 x: 0 y: 2 end
=====
```

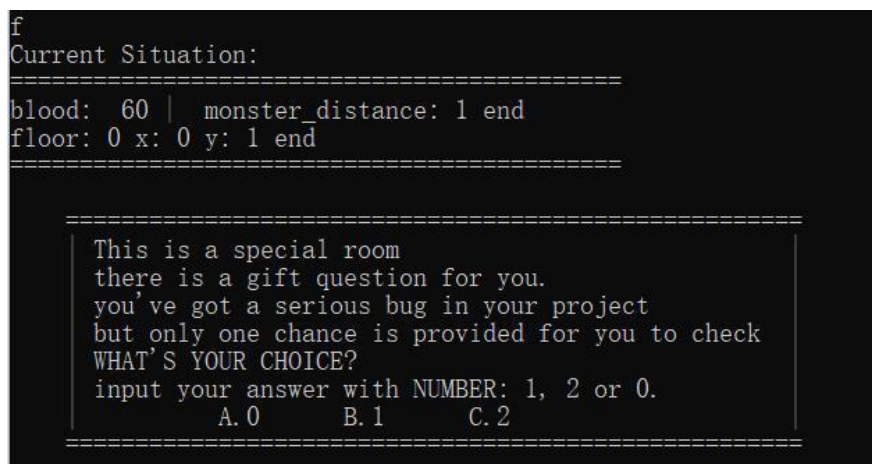
会提示相应的行进方向上已经没有房间，并重新输出当前玩家状态；或者提示不要尝试错误方向。是来自怪物的提示哦！

## \* NORMAL ROOM 的提示



会提示只是一个平平无奇的房间，但是有一些面包。不吃面包的话（左上图），面包会被保留，之后来到这个房间还是有面包吃的。吃掉面包可以增加血量，但一轮之内一个房间的面包吃完就没有了。

## \* 普通状态下 SPECIAL ROOM 的提示



询问你的选择。这个版本是简单的碰运气游戏（不完全碰运气，在攻略中有一些提示）所以答错不掉血。回答正确回血 20 点，回答错误弹出提示。



```

2
=====
| LUCKY YOU!                               |
| You successfully fixed the bug.          |
| Add 20 points of blood.                  |
=====
=====
blood: 80 | monster_distance: 1 end
floor: 0 x: 0 y: 1 end
=====

```

正确提示

```

2
=====
| PITY! you failed.                         |
| the project crashed.                     |
| YOU CAN KEEP GOING!                     |
=====

```

错误提示

### \* 上帝视角下 SPECIAL ROOM 的提示

下图展示上帝视角下的特殊房间。可以看到首先打印表格，表格的一行分别是：building 下标，房间类型，房间位置属性，房间关键值。对于普通房间，关键值是面包的回血点数；对于特殊房间，关键值是问题的答案。

**查表方式**是：找到自己当前所在房间号，对应到表中，获得关键值。

另外，氪金只保证上帝视角，但不保证无敌模式，如果王子应该死掉了还是要死掉的。

```

f
Current Situation:
=====
blood: 60 | monster_distance: 4 end
floor: 0 x: 0 y: 1 end
=====

0, 0, 0: n      0 0 0      key_point: 50
0, 0, 1: s      0 0 1      key_point: 0
0, 0, 2: n      0 0 2      key_point: 40
0, 1, 0: s      0 1 0      key_point: 1
0, 1, 1: n      0 1 1      key_point: 30
0, 1, 2: s      0 1 2      key_point: 1
0, 2, 0: n      0 2 0      key_point: 30
0, 2, 1: s      0 2 1      key_point: 2
0, 2, 2: n      0 2 2      key_point: 50
1, 0, 0: n      1 0 0      key_point: 50
1, 0, 1: s      1 0 1      key_point: 2
1, 0, 2: n      1 0 2      key_point: 50
1, 1, 0: s      1 1 0      key_point: 0
1, 1, 1: n      1 1 1      key_point: 30
1, 1, 2: s      1 1 2      key_point: 1
1, 2, 0: n      1 2 0      key_point: 50
1, 2, 1: s      1 2 1      key_point: 1
1, 2, 2: n      1 2 2      key_point: 20
=====

This is a special room
there is a gift question for you.
you've got a serious bug in your project
but only one chance is provided for you to check
WHAT'S YOUR CHOICE?
input your answer with NUMBER: 1, 2 or 0.
      A. 0      B. 1      C. 2
=====

0

=====
LUCKY YOU!
You successfully fixed the bug.
Add 20 points of blood.
=====

=====
blood: 80 | monster_distance: 4 end
floor: 0 x: 0 y: 1 end
=====

```

上帝视角的输出

### \* 进入公主房的提示

```
r  
Current Situation:  
=====
```

blood: 60	monster_distance: 3 end
floor: 0 x: 1 y: 0 end	

```
=====
```

```
=====
```

CONGRATULATIONS!

```
=====
```

```
you found the princess!  
but the princess need  
more blood points than you have!  
try to eat more bread!  
BUT on the way, I'm still waiting for u  
GOOD LUCK!
```

```
--monster
```

第一种是虽然找到了公主所在的房间，但是不能补充足够的体力给公主，所以解救失败，提示“try to eat more bread”。这时玩家可以去其他房间回血，等到血槽够了再回来解救公主。解救成功时显示：

```
b  
Current Situation:  
=====
```

blood:	110	monster_distance:	4 end
floor:	0 x:	1 y:	0 end

```
=====
```

```
=====
```

CONGRATULATIONS!

```
=====
```

you found the princess!  
and saved her with your blood points!  
now, you need to go back to the HALL(0, 0, 0)  
BUT on the way, I'm still waiting for u  
GOOD LUCK!

```
=====
```

```
--monster
```

## \* 进入怪物所在房间的提示（上帝视角）

```
f
Current Situation:
=====
blood: 120 | monster_distance: 0 end
floor: 2 x: 0 y: 1 end
=====

=====
YOU MEET THE MONSTER!
but your blood is more than the monster, so you killed this monster!
while at the same time, another monster has been created.
because you killed its friend, this monster is more tough to kill.
GOOD LUCK!
=====

delete successfully
0, 0, 0: n      0 0 0      key_point: 20
0, 0, 1: s      0 0 1      key_point: 0
0, 1, 0: s      0 1 0      key_point: 2
0, 1, 1: n      0 1 1      key_point: 40
1, 0, 0: n      1 0 0      key_point: 50
1, 0, 1: s      1 0 1      key_point: 1
1, 1, 0: s      1 1 0      key_point: 1
1, 1, 1: n      1 1 1      key_point: 40
2, 0, 0: n      2 0 0      key_point: 50
2, 0, 1: s      2 0 1      key_point: 0
2, 1, 0: s      2 1 0      key_point: 2
2, 1, 1: n      2 1 1      key_point: 20
3, 0, 0: n      3 0 0      key_point: 40
3, 0, 1: s      3 0 1      key_point: 2
3, 1, 0: s      3 1 0      key_point: 2
3, 1, 1: n      3 1 1      key_point: 50
Init Rooms : !FINISHED!
if you want to kill the monster, you need 80 blood!
Init Monster : !FINISHED!
Init Princess : !FINISHED!
Current Situation:
=====
blood: 40 | monster_distance: 4 end
floor: 2 x: 0 y: 1 end
=====

=====
blood: 40 | monster_distance: 4 end
floor: 2 x: 0 y: 1 end
=====
```

第一种情况，玩家的血槽比怪物的厚，提示“you killed the monster”。这头怪物被杀死之后，新的怪物会生成，玩家会掉相应点数的血。因为“you killed its friend”，所以新生成的怪兽血槽会比死掉的这头更厚，同时重新分配房间类型和面包/福利问题的奖励点数，而公主的位置不会改变。上帝视角下会打印出新的信息。

```
b
Current Situation:
=====
blood: 130 | monster_distance: 0 end
floor: 0 x: 1 y: 1 end
=====

=====
|                PITY!                |
=====
|                GAME * OVER            |
=====
| YOU LOOK REALLY TASTY                  |
| SO IS THE PRINCESS...                 |
| THANKS FOR GOOD DINNER                |
|                                     --monster |
=====

BYE! input any key to exit!
```

但是不可避免的会出现怪物血槽太厚的情况。这个时候王子就是时候一命呜呼了，同时怪物还会对王子和公主的味道进行一番评价... ..



## \* 回到大厅的提示

没有成功拯救公主的时候回到大厅，会询问玩家是否想要逃跑。玩家可以选择逃跑，也可以继续。

```
b
Current Situation:
=====
blood: 60 | monster_distance: 2 end
floor: 0 x: 0 y: 0 end
=====

=====
|                                     |
|                               HALL   |
|                                     |
=====
|                                     |
|    U WANT TO ESCAPE??????         |
|    YES (Y)           NO (N)       |
|                                     |
=====

=====
|                                     |
|                               RESTART |
|                                     |
=====
```

不逃跑

```
b  
Current Situation:  
=====
```

blood:	60		monster_distance:	2 end
floor:	0 x:	0 y:	0 end	

```
=====
```

| HALL |

U WANT TO ESCAPE??????

YES (Y)NO (N)

```
=====
```

y

PITY!

the princess dead!

```
=====
```

BYE! input any key to exit!

## 逃跑 (结束)

## \* 成功的提示

```

1
Current Situation:
=====
blood:   86 | monster_distance: 5 end
floor: 0 x: 0 y: 0 end
=====


=====
|                                |
|                   HALL        |
|                                |
=====
|                                |
|          CONGRATULATIONS!     |
|                                |
=====
| OK, FINE.                     |
| YOU WIN!                      |
| WISH YOU TWO HAPPINESS... ..  |
| I'm finding the next girl... .|
|                               |--monster
=====

BYE! input any key to exit!
```

同时满足了 1) 解救公主 2) 血槽状态 $>0$  3) 身处大厅三个条件的时候自动判定为成功，弹出框并结束游戏。

## 4. 隐藏攻略和提示

---

### a. 计算 monster 的位置

因为会给出与 monster 之间的折线距离，所以如果有兴趣的话可以通过观察移动时折线距离的变化来估算 monster 的可能位置。因为 princess 在沉睡，无法听到她的声音，所以游戏不提供公主房间位置的提示。

### b. 隐藏的面包

在成功解救公主以后，这个房间就会变成普通房间或者特殊房间（随机），此时可以离开后再次返回，以获得隐藏的面包。

### c. 血量的上限

默认血槽在 400 点的时候达到满血，不能再增加，但仍然可以执行吃面包的操作，但是会提示血槽已满不加血。



## 二、问题和解决过程

---

### a. 实现阶段的问题

---

#### 随机数问题：

C11 提供了新的随机数库<random.h>，在我使用原来的随机数函数的时候收到了警告，本着合格的程序员消灭所有警告的原则（？），查阅了一些文档，更新了随机数生成办法，在这里做一点记录：

原来的方法<sup>1</sup>：

**rand()：** 随机返回一个 0 到 RAND\_MAX 之间的整数值，RAND\_MAX 在 stdio.h 中定义为 2147483647，如果要获得某个范围内的值或者浮点数，对 rand（）结果进行除法或取余操作即可。

**srand()：** 设置随机数种子。rand 默认随机数种子为 1，也就是说这是一个“伪随机数列”，虽然这一组数据是随机的，但是每次都是这一组数据。通常的做法是 **srand(time(null))**，用系统时间作为这次 rand 的种子，这样每次发出来的芽就不一样了！

---

<sup>1</sup> <http://www.cplusplus.com/reference/cstdlib/rand/>

其实这个 埋下种子然后发芽 的过程是比较有局限性的，因为我们现在只用得到随机感受不到，但是如果分析时需要生成符合特殊分布的随机数（比如正态分布）就需要用户手动调整。

另外，time 返回的是距离 1970.01.01 零时的秒数，如果 rand 函数在 1s 内调用多次，那么产生的数据也是相同的。能不能用一个随机数做另一个随机数的种子呢？我认为是可以的。比如随机数列 A 作为 B 的种子，保证 A 和 B 不是每次都重新生成即可（否则一直使用的是一样的 A，也相当于没变）。

### C11 random 库<sup>2</sup>:

摘录一段 C++ 文档内容，很清晰的给出了思路：

This library allows to produce random numbers using combinations of generators and distributions:

**Generators:** Objects that generate uniformly distributed numbers.

**Distributions:** Objects that transform sequences of numbers generated by a generator into sequences of numbers that follow a specific random variable distribution, such as uniform, Normal or Binomial.

---

<sup>2</sup> <http://www.cplusplus.com/reference/random/>

其实也可以从种子的角度去理解，engine 就是一个种子，distribution 就是一个大棚，基本思路类似，只是这里可以直接选择不同的大棚，不必自己搭建了。

### 最后，在程序中的使用：

```
#include <random>
extern default_random_engine gen;
int a = 0, b = total_floor - 1, c = total_x - 1, d = total_y - 1;
uniform_int_distribution<int> dis1(a, b);
uniform_int_distribution<int> dis2(a, c);
uniform_int_distribution<int> dis3(a, d);
monster_floor = dis1(gen);
do {
    monster_x = dis2(gen);
    monster_y = dis3(gen);
} while (monster_x + monster_y == 0);
building[monster_floor][monster_x][monster_y]->set_monster(1);
cout << "    " << "Init Monster : !FINISHED!" << endl;
```

C++

这里为啥 gen 要搞成外部（全局）变量呢？和原来方法那部分随机数做随机数的种子是一样的！如果每次我都重新定义一下 gen，实际上得到的 gen 还是不变，相当于一个“伪随机数列”，所以至少引擎和分布函数生成的次数不能完全同步（不知道说明白没有...）。

## 多文件问题：

其实是一个比较小的问题。首先是头文件都能写什么，去年写 C 大程的时候没搞清楚多文件可头疼死我了 QAQ。先说说 C++ 头文件里面写什么吧？

```
#ifndef NAME_H
#define NAME_H
//你的代码写在这里
#endif //NAME_H
```

里面你可以写：你的类(声明成员和成员函数，但不必实现)、结构体，普通函数声明(**不必实现**)，外部变量，宏定义。btw，在自己写的头文件里包含<string>是常见的。

## 关于为什么不必实现

第一，预编译指令#include 的作用是将所包含的文件全文复制到#include 的位置，相当于是个展开为一个文件的宏（这里想到原来 getchar 也只是一个有关 getc 的宏）。

第二，C++ 允许多次声明，但只允许一次实现。比如 int foo(); 就是一次声明，而 int foo(){}就是一次实现。如果编译时有多个.cpp 文件中#include 了同一个含有函数实现的.h，这时候链接器就会在**多个目标文件中找到这个函数（和下一条的全局变量有关）**的实现，而这在 C++ 中是不允许的，此时就会引爆 LNK1169 错误:找到一个或多个重定义的符号。

因此为了让函数可以在各个.cpp 中共享，正确的做法就是在.h 中只声明函数，并在另一个.cpp 中实现这个函数。这样就不会冲突了。

**关于一些尝试：** 只是一些尝试。

1. 在头文件中定义宏可以在引用了这个头文件的.cpp/.h 文件中直接用——因为它是宏。
2. 但是全局变量不可以。全局变量在另一个.cpp 中定义，用的时候声明为外部变量。为什么呢？和头文件不必实现一个函数其实是类似的吧，因为这样编译器会报错说你在头文件的哪个地方 define 了好几次这个变量，因为你写在头文件里面，编译的时候所有引用了这个头文件的.cpp 相当于都有了这个变量的定义了，当然不行。

**其他的问题：简单报告**

### **1. 公主/怪物应当单独列出一个 class NPC 来吗？**

在目前版本属于“可以，但没必要”的操作，因为二者的属性太少了，访问对象、调用函数来获得 NPC 信息还不如一个全局变量来的方便。但 这样造成了风格不够 C++，封装不完善，其实对于游戏的发展维护是很有风险的。

## 2. 关于到达公主房之后消息弹框的问题

分为几个情况：进了公主房间，玩家属性“找到公主了吗”变成 1，会在判断里面导致消息弹框每步都弹出。后面又加了给公主加血的操作，所以就加了“救了公主了吗”这个属性，但是二者会导致消息弹框判断失误。现在的解决方法是添加第三个属性。。。很累赘了，后面有机会会想办法调整一下结构。

## 3. possible further development

唉，菜是原罪

说说还想改进的地方：

- a. 道具太少了！！！！！！内测玩家们表示 bread 道具丝毫没有吞咽的欲望；
- b. special room 加入真的题目（比如 oop 期中考试题）；
- c. 引入登录机制！储存玩家信息！
- d. 根据房间数改变 monster 的个数。现在它的血槽是根据房间数变化的了！

## b. 试玩用户反馈

---

**内测用户 1:** 哈哈哈哈哈太欢乐了，就是多点儿道具，然后怪兽打不过可以跑，有的时候多打一个回车会导致无限卡 bug 出不去房间，special room 的答案输入两位数会卡 bug，开头随便输个字母也能进上帝视角，结束的时候会闪退。

**道具问题:** 没时间写 QAQ

**怪兽打不过可以跑:** 那不是永远也死不了了，但是为了改进，我在开始提示了打败初级怪兽需要的最少血量。

**多打一个回车:** 已修复

**两位数答案:** 这个暂时不考虑，其他的容错也暂时不纳入考虑

**密匙检查:** 已修复

**闪退:** 已修复，需要用户输入一个指令后退出。

---

**内测用户 2:** 这个血怎么没上限啊。。。然后那个选 012 的我一直输入的 abc...

**血槽上限:** 已修复

**012 输入:** 已修复，加了提示

## 三、感受

面向对象，真香！

感觉这次写的很快乐

然后过程中遇到的一些有趣的问题

突然很想记录一下

于是也开始写自己的 blog 了

今天也是一只快乐的菜鸡！