



浙江大学

Object-oriented programming Project 2

Game of Castle

Name: 沈吕可晟

Class: 数媒 1801

Student Id: 3180101044

Date: 2019-11-25

Content:

一、Project Introduction.....	3
二、Game of Castle.....	3
2.1 Project Displaying.....	3
2.1.1 A graph of my castle.....	3
2.1.2 My unique functions and interface display....	4
2.2 File constitution.....	8
2.3 Data Structures.....	9
2.4 Technical Details.....	10
三、Experiment Experiences Summary.....	11

一、Project Introduction

Adventure is a CLI game. The player has to explore in the castle with two levels and a lot of rooms. The task of the player is to find a room where the princess which is prinsoned and take her leave the castle. There are many types of rooms, and each type of room has different types of exits. Note that there's a monster in one of the rooms, which the exact location is randomly set. But once the player meets a monster, the game is over.

More details will be given in the part2.

二、Game of Castle

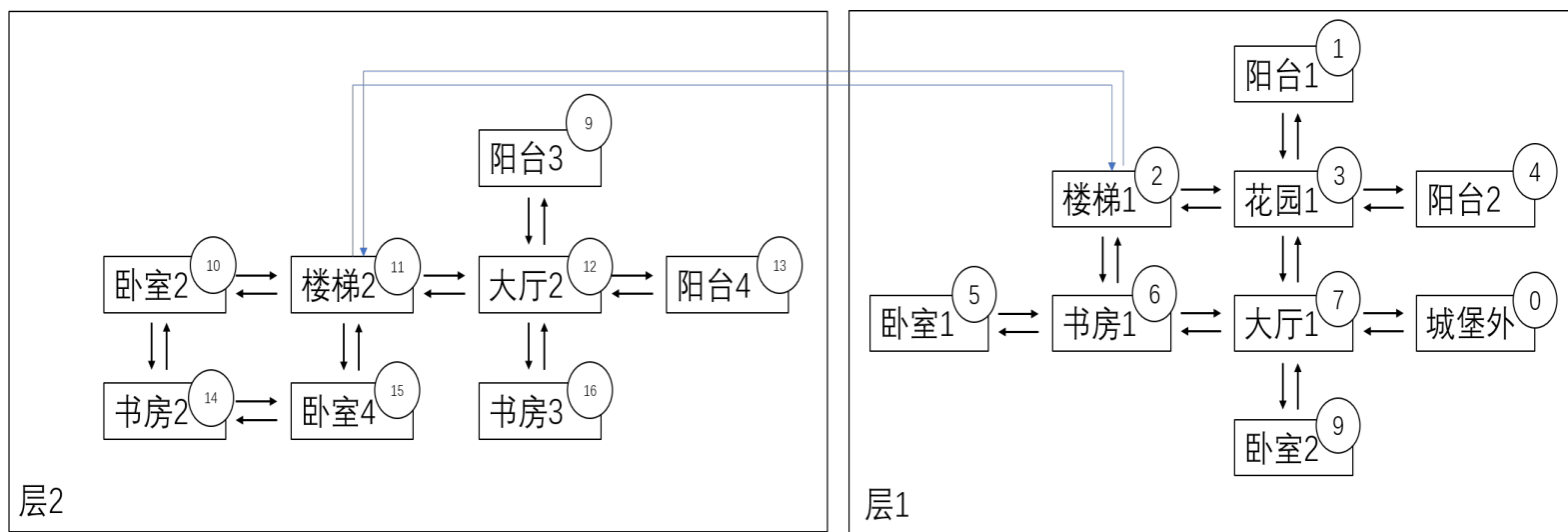
2.1 Project Displaying

2.1.1 A graph of the game

图例：



房间类型+类内编号

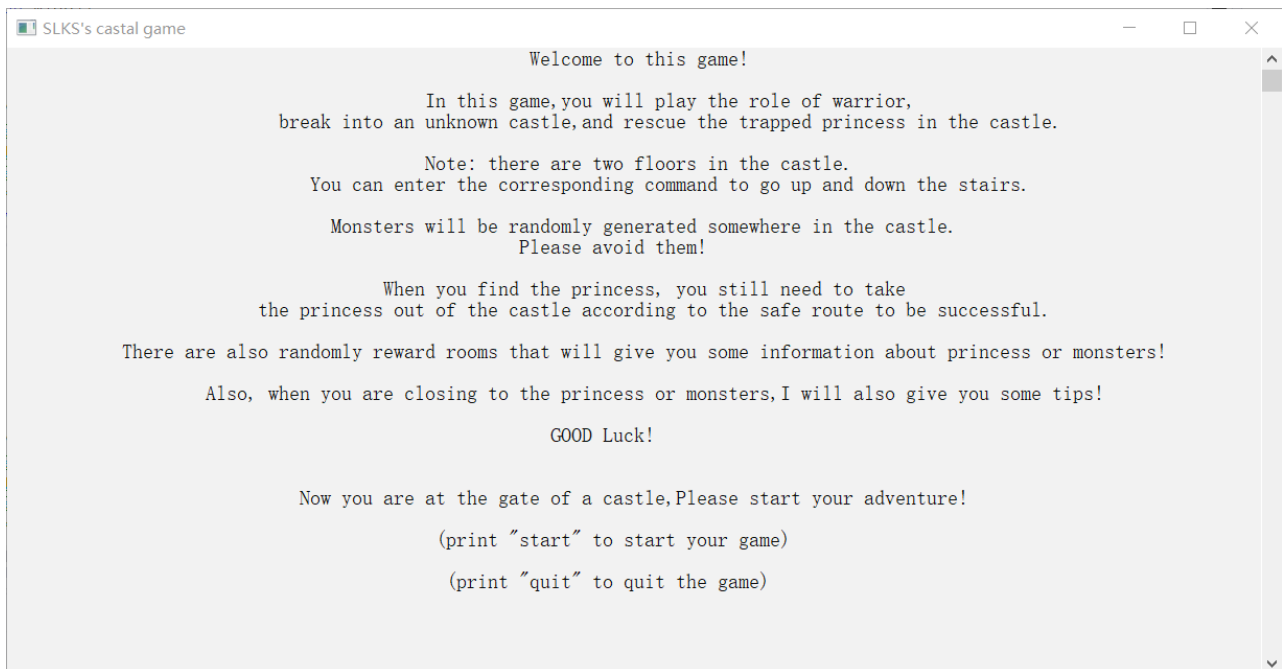


2.1.2 My unique functions and interface display

In my game, the castle has two layers, when coming to the stairs you can use command “up” or “down” to go the other layer. Before the game start, you can choose to start or quit. If you choose to start, you will be in the gate of the castle(the 8th id place). Then you can start your adventure. After finishing the game, whether you succeed or failed, you can choose to start again or not. If you choose start again, the game will refresh.

During the game, in my game, only find the princess is useless, because the princess is prisoned, you must find the key in the castle first. So, if you want to win the game, you should find the key first and find the princess, use the key to unlock the princess, and bring her out of the castle safely.

The first page in my program shows below:



```

Welcome to this game!

In this game,you will play the role of warrior,
break into an unknown castle,and rescue the trapped princess in the castle.

Note: there are two floors in the castle.
You can enter the corresponding command to go up and down the stairs.

Monsters will be randomly generated somewhere in the castle.
Please avoid them!

When you find the princess, you still need to take
the princess out of the castle according to the safe route to be successful.

There are also randomly reward rooms that will give you some information about princess or monsters!

Also, when you are closing to the princess or monsters,I will also give you some tips!

GOOD Luck!

Now you are at the gate of a castle,Please start your adventure!

(print "start" to start your game)
(print "quit" to quit the game)
  
```

【Unique function 1】（界面刷新功能）

The beauty of the game interface adopts a white background and black characters interface, which makes the user more comfortable when using, and can modify the font color to give different prompts to the player in the following text. At the same time, when entering a new room, the console information will be refreshed, only the current room information will be displayed, and the interface is more concise.

Game of Castle

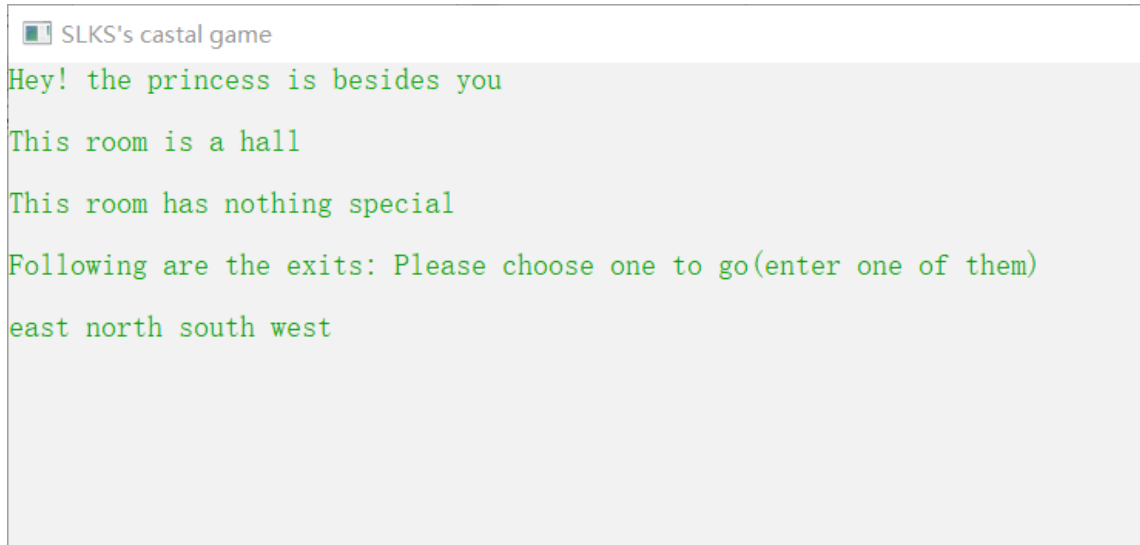
【Unique function 2】（提示功能）

Considering the randomness of the game, in order to enhance the playability of the game, I provide information prompt function.

When the princess is in the room around you, the font will turn green and prompt "" to remind you that victory is in front of you.

When the monster is in the surrounding room of your current room, the font will turn red and prompt "" to warn you that if you take the wrong step, you will be doomed.

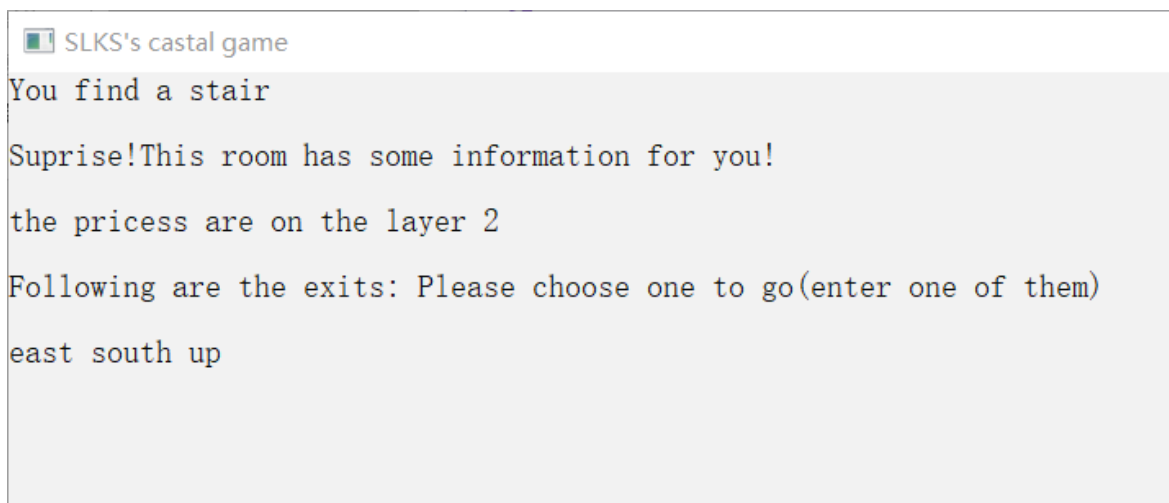
The specific interface screenshot is as follows:



【Unique function 3】（道具房间）

In order to increase the playability of the game, because the castle map is large, I randomly selected two rooms from all the rooms to generate props. One room will tell you the floor where the princess is, and the other room will tell you the floor where the monster is. They can better help you avoid the monster and find the princess.

The specific interface screenshot is as follows:



【Unique function 4】（钥匙机制）

For the integrity and verisimilitude of the game, I added the key mechanism. When the player does not find the key, if he enters the princess's room, he cannot rescue the princess. The game will also prompt players to find the key before they can come back to rescue the princess. If the player finds the key and comes back to the princess's room, he can save the princess from the castle.

```
SLKS's castal game
This room is a bedroom
You find princess! But the princess was imprisoned by a shackl
Please find the key first
Following are the exits: Please choose one to go(enter one of them)
north
```

```
SLKS's castal game
Hey! the princess is besides you
You find a stair
Oh! You find a key!It can be used to rescue the princess!
Following are the exits: Please choose one to go(enter one of them)
east south up
```

```
SLKS's castal game
You find a stair
You find princess! Oh! You have brought back the key!
You opend the lock with the key!
Now!Take the princess out of the castle!!
Following are the exits: Please choose one to go(enter one of them)
down east south west
```

【Unique function 5】（输入错误会请求重新输入）

If the player accidentally enters an unexpected command when entering the command (for example, if the player enters East when there is only one exit from West in the room, the game will prompt the player to re-enter the correct command)

```
SLKS's castal game
You find a stair
This stair has nothing special
Following are the exits: Please choose one to go(enter one of them)
down east south west
dounw
You have choose the wrong exits,please enter again correctly!
```

【Unique function 5】（支持选择是否重新开始游戏）

At the end of the game, whether or not the player has successfully saved the princess, the player can choose whether to restart the game or exit the game. If the wrong command is entered, the system will also prompt to re-enter

```
SLKS's castal game
You come to a garden
Oh! This room has a monster! You died!

Warrior,you have failed rescuing the princess! Don't be discouraged!
Do you want to play again?(print yes to go,print no to quit)
ysa
you have entered a undefined command! Please enter again
yes
```

2.2 File Constitution

Among them, there are seven. H files **represent** seven classes. Corresponding to 7. CPP files at the same time. There is also a .Cpp file where the main program is located.

In the main program. Cpp file, the file covers the following:

```
#include<iostream>
#include<stdlib.h>
#include<string>
#include<ctime>
#include<windows.h>
#include"Bedroom.h"
#include"Room.h"
#include"Study.h"
#include"Balcony.h"
#include"Stairs.h"
#include"Garden.h"
#include"Hall.h"
using namespace std;
```

The file coverage relationships in the derived classes .H and .CPP files are as follows: (take bedroom for example)

```
#ifndef BEDROOM_H
#define BEDROOM_H
#include"Room.h"
// class declaration
#endif
```

The file coverage relationships in the base class .H and .CPP files are as follows:

```
#ifndef ROOM_H
#define ROOM_H
#include<iostream>
#include<stdlib.h>
#include<string>
#include<ctime>
#include<windows.h>
#include<map>
using namespace std;
    //class declaration
#endif
```


2.3 Data structures

【Class definition description】

In my program, I have defined a base class and six derived classes, corresponding to seven different room types, and several objects of each type of room. In each category, the stored data and descriptions are as follows:

```
class Room{
    protected:
        map<string, Room*> way;
        int id;
        int layer;                //房间所在层数
        int monster;              //该房间有怪物设置为1，无怪物设置为0
        int princess;             //该房间有公主设置为1，无怪物设置为0
        int tool;                 //该房间是道具房间设置为1，无则设置为0
        int tool2;                //该房间是道具房间设置为1，无则设置为0
        int key;                  //该房间有钥匙设置为1，无则设置为0
}
```

【Global variables description】

At the same time, I defined several global variables to control the global information. Data and description are as follows

```
Room bornplace;                //玩家出生地(城堡外的gate)
Room* ptr[17];                 //其中ptr[0]指向城堡外
Room* now, last;               //玩家当前所在房间与玩家上一个房间
int flag;                      //判断玩家是否已经带着princess(1为已经带着princess, 0为还未找到princess)
int flag2;                    //判断上一局游戏胜利还是输了
int flagkey;                  //判断玩家是否找到了钥匙
int tool;                     //第一个提示房间所在位置（提示公主所在房间楼层）
int tool2;                    //第二个提示房间所在位置（提示怪物所在房间楼层）
int key;                      //钥匙所在房间
int prin;                     //公主所在房间
int mons;                     //怪物所在房间
Study study[3];               //三个书房对象
Bedroom bedroom[4];           //四个卧室对象
Balcony balcony[4];           //四个阳台对象
Garden garden;                //一个花园对象
Stairs stairs[2];             //两个楼梯对象
Hall hall[2];                 //两个大厅对象
```

2.4 Technical Details

The main program, main.cpp, is divided into three functions and one main function for writing. The main function architecture is as follows:

```
int main() {
    initilize();
    gameintroduction();
    gamestart();
}
```

The basic structure or function or pseudo code description of the other three functions are described as follows:

【initilize function】

This function is used to connect the defined rooms of various types, connect the rooms with their ID, input the exit information of each room, and randomly generate the room location where the monster, princess, props and keys are set. The overall structure of this function is as follows:

```
void initilize() {
    SetConsoleTitle("SLKS's castal game"); //设置控制台窗口标题
    system("color F0"); //设置控制台颜色
    system("cls"); //控制台清空
    seed = time(0); //生成随机数种子
    srand(seed); //利用随机数种子重置随机数

    Randomly generate the room number of monster and princess
    【for example:
        prin = rand()%16 + 1;
    】

    Remove the special cases.
    【for example:
        while(mons==2 || mons==11 || mons==7) {
            mons = rand()%16 + 1;
        }
    】

    Initialize the map, associate it with the PTR pointer array, and set the surrounding environment
    【for example:
        ptr[6] = &study[0];
        study[1].set_id(14);
        bornplace.set_out("west", ptr[7]);
        balcony[0].set_out("south", ptr[3]);
        stairs[0].set_out("east", ptr[3]);
    】

    Call the function to set the room where the monster princess is located
    【for example:
        for(i=1; i<=16; i++) {
            if(mons == i) {
                ptr[i]->set_monster();
            }
        }
    】
}
```

【gameintroduction function】

This function only does the function of game introduction, and outputs the game specific introduction to the first console page through “cout” function to introduce the game playing methods and rules for the players.

【gamestart function】

This function is the core function part of the game operation. Some core codes are described as follows: (details have been omitted)

```
void gamestart() {
    cout<<endl;
    string s;           //接收移动指令
    string temp;        //接收游戏结束后的指令
    string st;          //接收游戏最开始是start or quit
    cin>>st;
    if(st=="start"){    //如果玩家输入start，则开始游戏
        now = ptr[0];   //初始化now函数
        ptr[0]->print(); //打印出生点情况
        while(cin>>s){  //进入接受游戏指令
            last = now;  //记录当前房间
            now = now->get_mapinfo(s); //now跳转到，玩家指令输入后的下一个房间
            if(now == ptr[0] && flag==1){ //游戏获胜条件
                cout<<endl;
                cout<<"You have rescued the principle!Great!"<<endl;
                flag2 = 1;
                break;
            }
            if(now->monster_warning()){ //如果周围房间有怪物存在，进行提醒
                system("color F4");    //设置控制台字体颜色为红色
                cout<<"Warning!!monster are around you!\n"<<endl;;
            }
            if(now->princess_congradulate()){ //如果公主在周围房间，进行提醒
                system("color F2");    //设置控制台字体颜色为绿色
                cout<<"Hey! the princess is besides you\n"<<endl;
            }
            now->print(); //打印当前房间信息
            if(now->ifmonster()){ //如果碰上了怪物
                flag2 = 0;        //flag2置0 并跳出循环
                break;
            }
            if (now->ifkey())      flagkey = 1;
            if(now->ifprincess()&&flagkey) flag=1;//如果碰上了公主并且有钥匙，flag置1
        }
        //如果跳出循环后flag2为0，游戏失败，给出剧情
        //如果跳出循环后flag2为1，游戏成功，给出剧情
        //游戏结束后，接收玩家是否愿意重新游戏的指令
        if(temp=="yes"){ //如果玩家输入yes，则初始化界面，重新调用该函数
            initilize();
            gamestart();
        }
    }
}
```

三、Experiment Experiences Summary

For me, the difficulty of this project is not the realization of the Castle game function, but the design of the class and the relationship between the class, the design of the virtual function between the base class and the derived class.

At the same time, because there was less writing for multi file projects before, the processing of multi files and the mutual inclusion relationship between files were also confusing for me before this experiment.

However, after this experiment, I sorted out the basic form of multi file processing, and standardized the form of the definition of header file, which completed the experiment better.

Also, because this is a command control of input and output on the console, I also learned some simple console functions to make the page simpler and more comfortable.