

POTTER 2077

灵感源于《达拉崩吧》、《霍格沃兹之遗》的打字游戏

第七小组

洪佳荣 薛全笑 高翊航



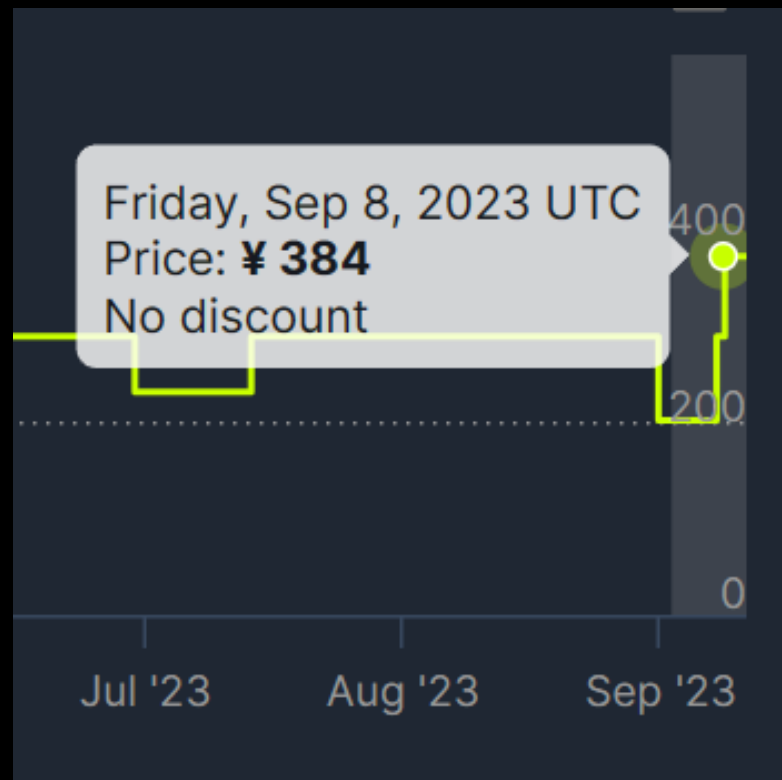
勇者斗恶龙


达拉崩吧

蒙达鲁克硫斯伯古比奇巴勒城
昆图库塔卡提考特苏瓦西拉松
达拉崩吧斑得贝迪卜多比鲁翁
昆特牌提琴烤蛋挞苏打马拉松
公主米娅莫拉苏娜丹妮谢莉红

浩然·达拉崩吧斑得
贝迪卜多比鲁翁·米
娅莫拉苏娜丹妮谢莉
红·迪菲特·昆图库塔
卡提考特苏瓦西拉
松·蒙达鲁克硫斯伯
古比奇巴勒·王







POTTER 2077

游戏背景

- 故事发生在未来的魔法世界。
- 2077年，魔法社会已经发生了巨大的变化。魔法与高科技融合，成为一个充满危险与奇迹的新时代。
- 哈利·波特的孙子，亚历克斯·波特 (Alex Potter) 是一名年轻有为的魔法学徒，在这个世界中展开他的冒险之旅——


```
Assets
├──.avatar
├──.credit
├──.credit_logo
├──.fightScene
├──.help
├──.logo
├──.map
├──.map_path
├──.monsterDescription
├──.welcomePage
└──.wordlistAttack

Monster
├──Scene
│   ├──剧情流程.txt
│   ├──AfterFight
│   │   └──战斗后的剧情
│   ├──Background
│   │   └──到达一个新地点时的故事背景
│   ├──BeforeFight
│   │   └──战斗前的剧情
│   ├──Default
│   │   └──地点默认剧情
│   └──Other
│       └──其余特殊剧情
└──src
    ├──Backpack.h
    ├──Behavior.h
    ├──Creature.h
    ├──Data.h
    ├──FightScene.h
    ├──Interface.h
    ├──Item.h
    ├──Location.h
    ├──Main.cpp
    ├──Map.h
    ├──Monster.h
    ├──Player.h
    └──Word.h
```

外部资源文件

Assets

所有源码

src

项目文件总览

```
Assets
├──.avatar
├──.credit
├──.credit_logo
├──.fightScene
├──.help
├──.logo
├──.map
├──.map_path
├──.monsterDescription
├──.welcomePage
└──.wordlistAttack

Monster
├──Scene
│   ├──剧情流程.txt
│   ├──AfterFight
│   │   └──战斗后的剧情
│   ├──Background
│   │   └──到达一个新地点时的故事背景
│   ├──BeforeFight
│   │   └──战斗前的剧情
│   ├──Default
│   │   └──地点默认剧情
│   └──Other
│       └──其余特殊剧情
└──src
    ├──Backpack.h
    ├──Behavior.h
    ├──Creature.h
    ├──Data.h
    ├──FightScene.h
    ├──Interface.h
    ├──Item.h
    ├──Location.h
    ├──Main.cpp
    ├──Map.h
    ├──Monster.h
    ├──Player.h
    └──Word.h
```

Assets

外部资源文件

素材

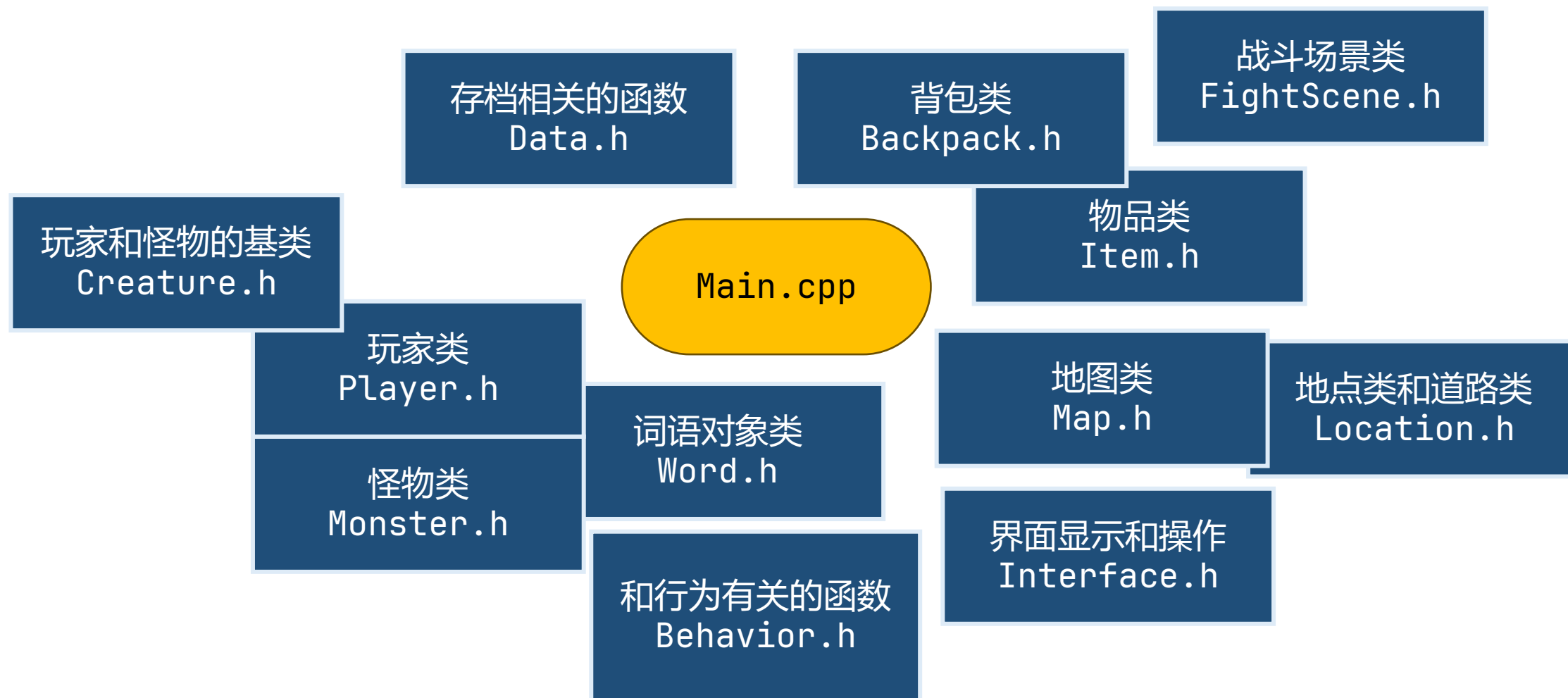
数值

剧情文本

项目文件总览

src

所有源码



```
Assets
├──.avatar
├──.credit
├──.credit_logo
├──.fightScene
├──.help
├──.logo
├──.map
├──.map_path
├──.monsterDescription
├──.welcomePage
└──.wordlistAttack

Monster
├──Scene
│   ├──剧情流程.txt
│   ├──AfterFight
│   │   └──战斗后的剧情
│   ├──Background
│   │   └──到达一个新地点时的故事背景
│   ├──BeforeFight
│   │   └──战斗前的剧情
│   ├──Default
│   │   └──地点默认剧情
│   └──Other
│       └──其余特殊剧情
└──src
    ├──Backpack.h
    ├──Behavior.h
    ├──Creature.h
    ├──Data.h
    ├──FightScene.h
    ├──Interface.h
    ├──Item.h
    ├──Location.h
    ├──Main.cpp
    ├──Map.h
    ├──Monster.h
    ├──Player.h
    └──Word.h
```

外部资源文件

Assets

通过文件流读取

所有源码

src

项目文件总览

成员分工

- 组长：洪佳荣
 - 负责界面设计、存档、玩家行为、地图、进度管理
 - 参与了战斗场景的思路设计和使用道具的实现
- 组员：薛全笑
 - 负责玩家、怪物的完善，负责战斗的实现，包括打字判定和实时改色等
- 组员：高翊航
 - 负责剧情细节和数值的编写，协助完善了战斗场景

- Pulse
- Contributors
- Community
- Community Standards
- Traffic
- Commits
- Code frequency
- Dependency graph
- Network
- Forks
- People

August 11, 2023 – September 11, 2023

Period: 1 month

Overview

58 Active pull requests

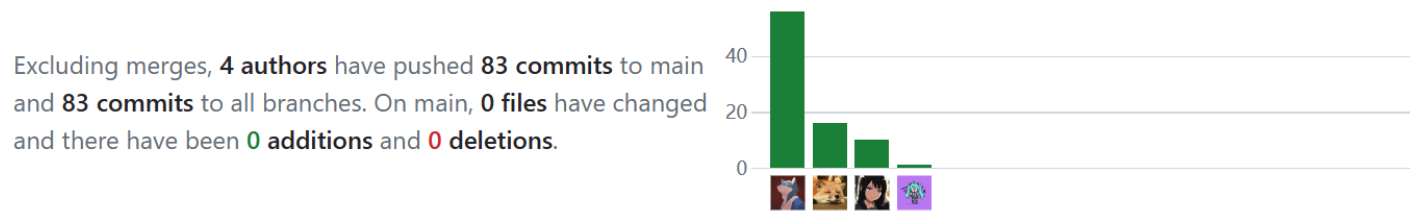
1 Active issue

58 Merged pull requests

0 Open pull requests

1 Closed issue

0 New issues



58 Pull requests merged by 4 people

- 修复了很多bug
#61 merged 2 hours ago
- 修复了很多bug
#60 merged 14 hours ago
- 修复了很多bug
#59 merged 14 hours ago
- 修复了一些bug

玩家创建

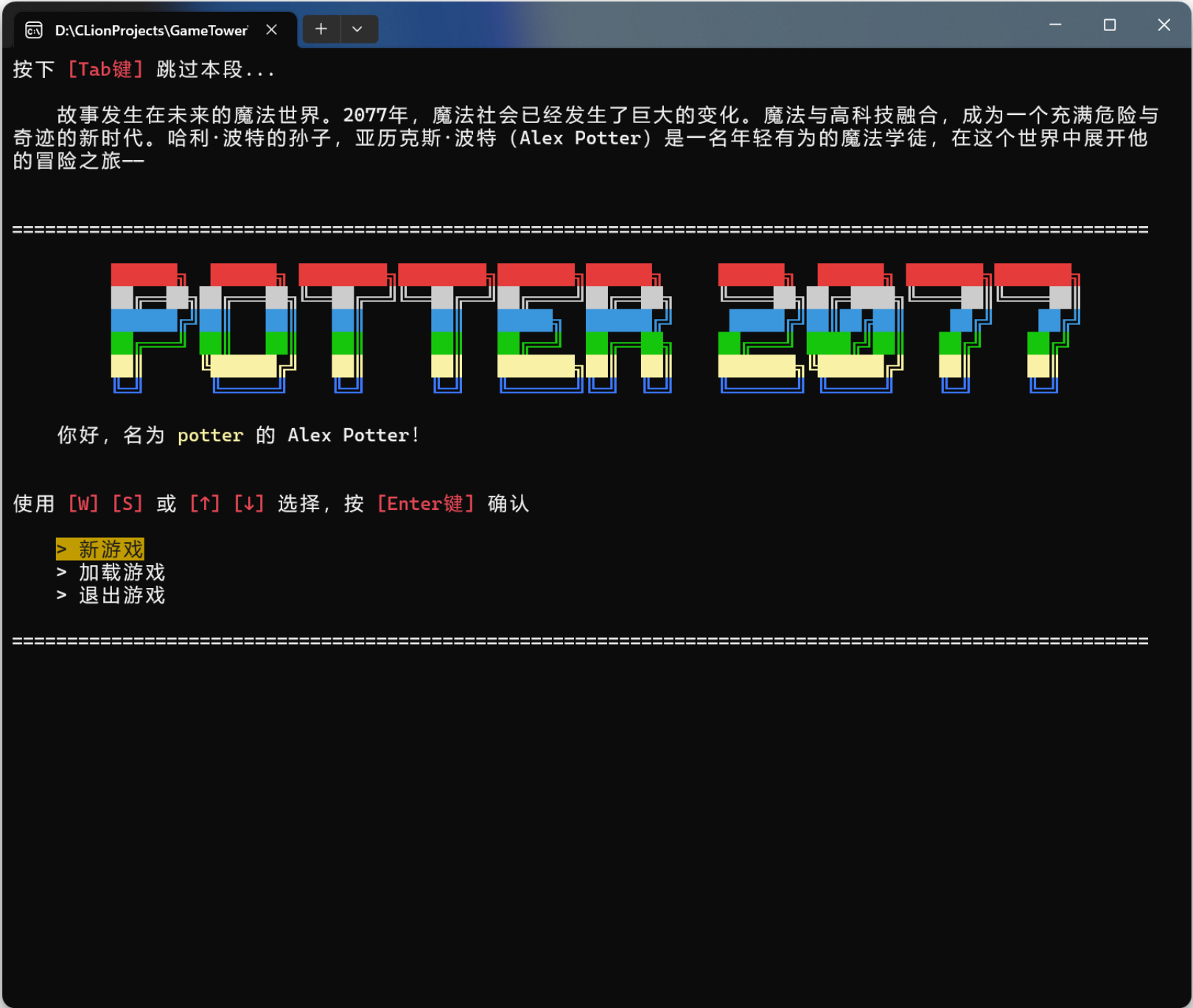
必须输入用户名
作为存档文件的名称

新游戏

允许使用同一名字
会给出是否覆盖提示

加载游戏

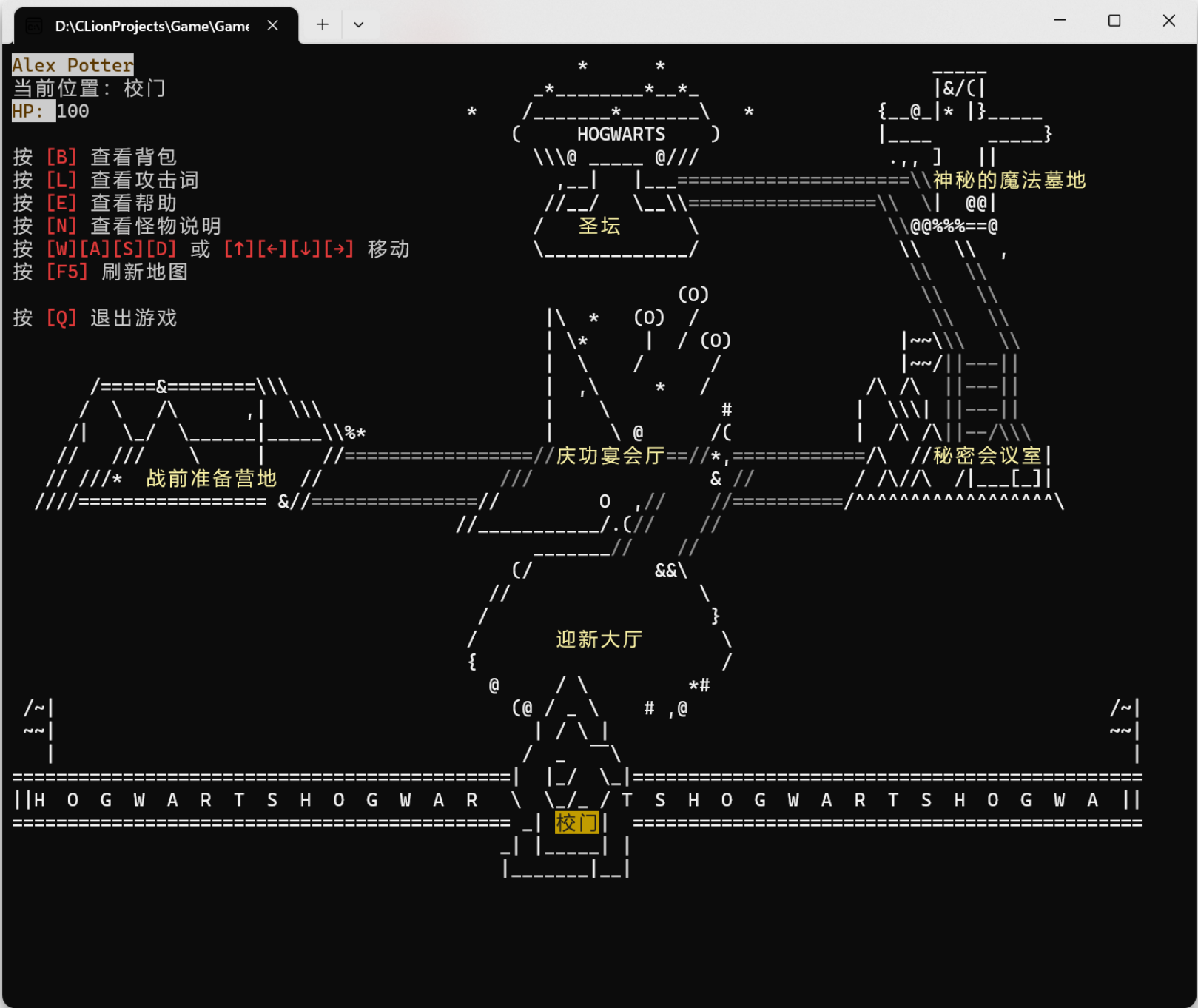
若无当前用户名存档
会创建新存档



玩家属性

HP 血量
背包 存放物品
攻击词 和伤害有关

隐藏属性 当前阶段



Alex Potter

当前位置： 校门

HP: 100

按 [B] 查看背包

按 [L] 查看攻击词

按 [E] 查看帮助

按 [N] 查看怪物说明

按 [W][A][S][D] 或 [↑][←][↓][→] 移动

按 [F5] 刷新地图

按 [Q] 退出游戏

300词

Word类

有effect和length两个基础属性
战斗时用到其color和currLength属性

```
D:\CLionProjects\Game\Game x + v
[13] descendo
[14] homenum
[15] revelio
[16] partis
[17] temporus
[18] ventuso
[19] aquaralis
[20] incendaria
[21] ventifero
[22] gelacius
[23] aerolitus
[24] pyrosium
[25] silvarum
[26] nocturnis
[27] solaris
[28] nox
[29] imperio
[30] crucio
[31] attacca
[32] expulso
```

背包

使用

`vector<pair<Item, int>>`

作为存储容器

```
D:\CLionProjects\Game\Game x + v
[0] LittleHealPotion 存量: 6
    回血10, 冷却50个时间刻
[1] MagicAttack 存量: 5
    造成10点伤害, 冷却50个时间刻
请按任意键继续. . .
```

物品

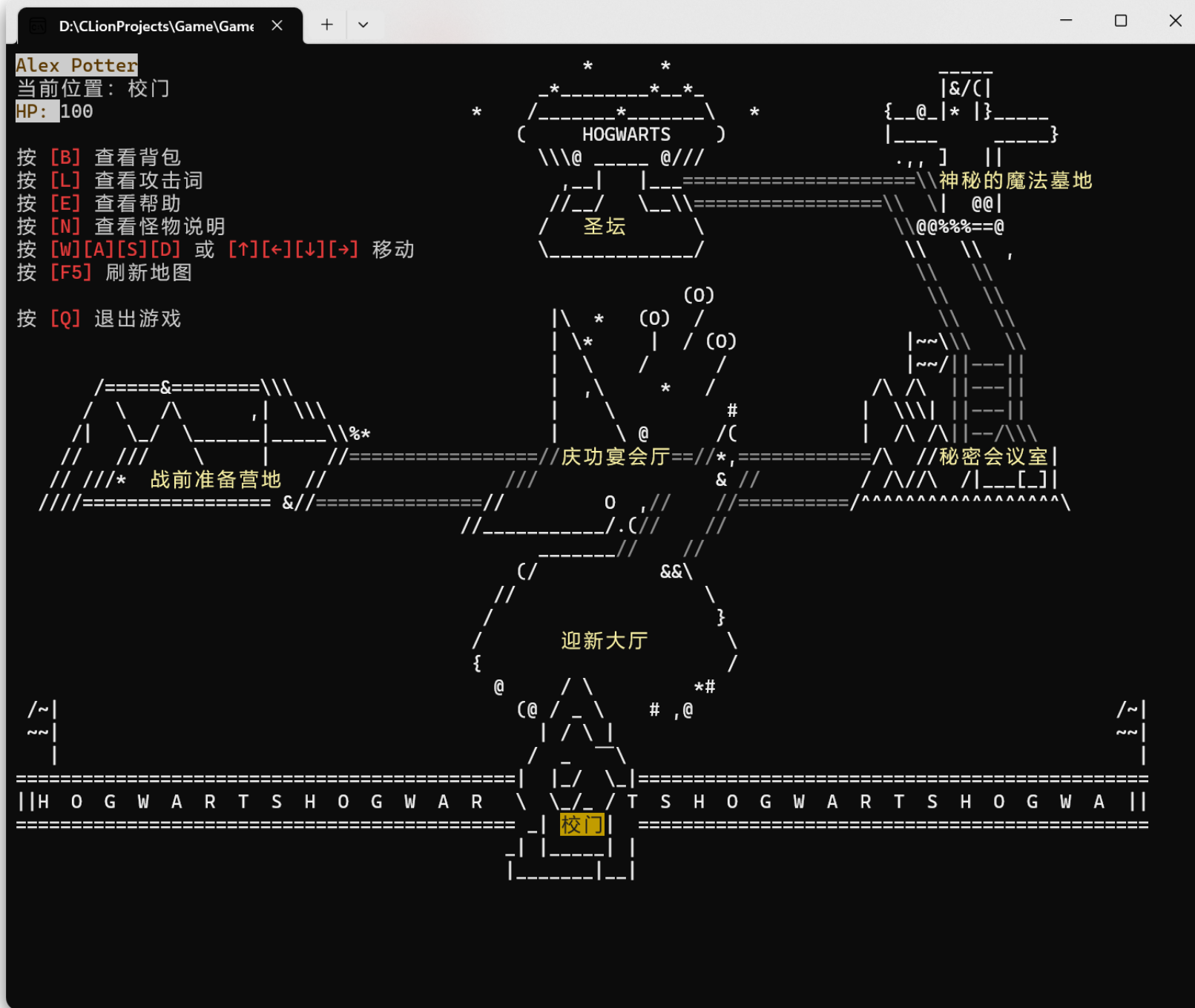
Item类, 有effect和cd两个属性

地图

由道路和地点组成
还有屏障区域

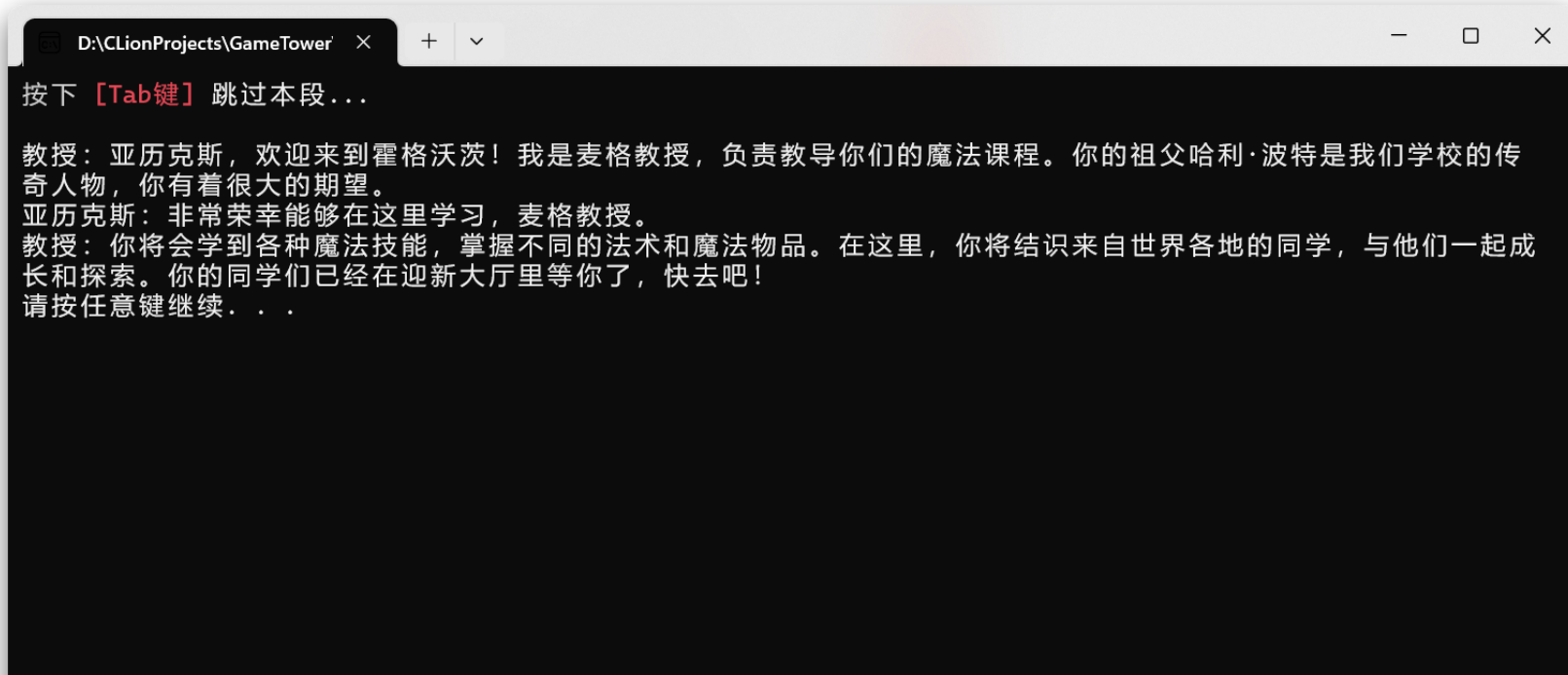
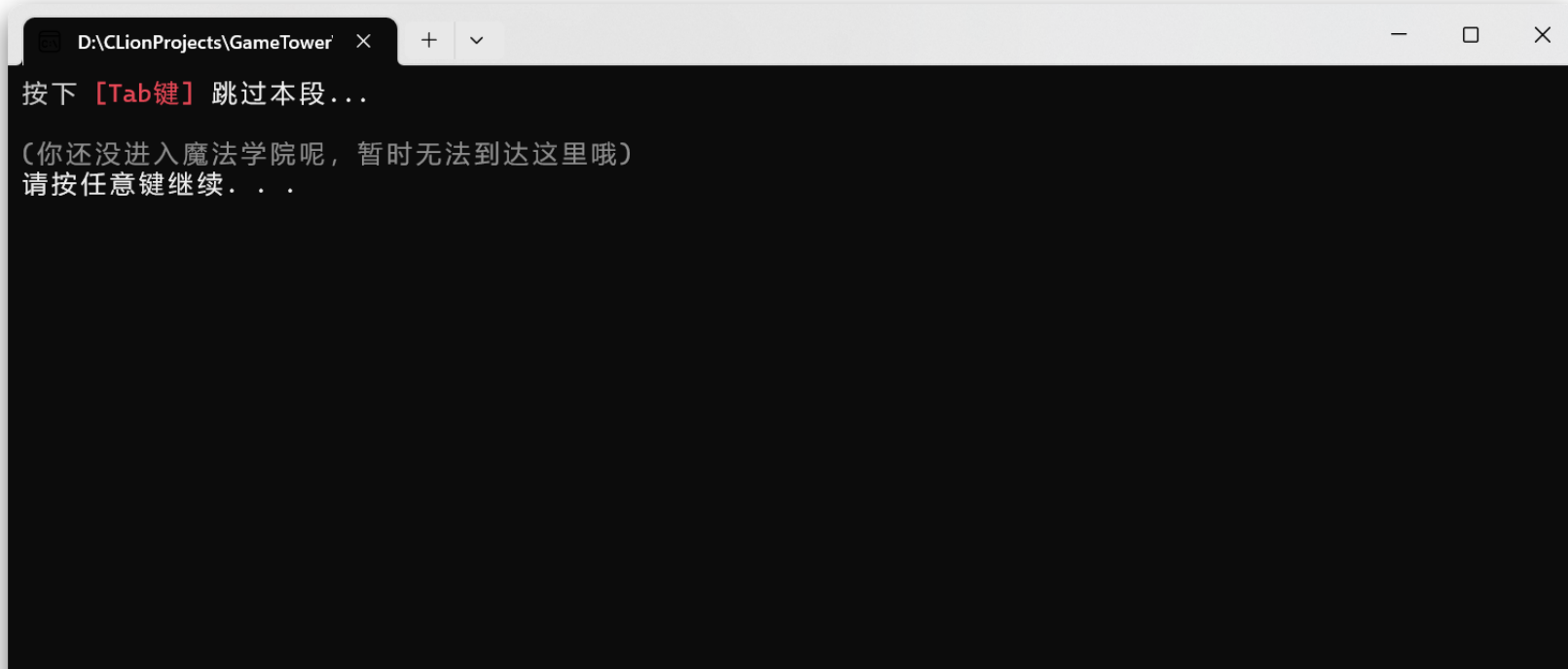
进度

进入未解锁区域
和已解锁区域
和目的地点
三者提示词不同

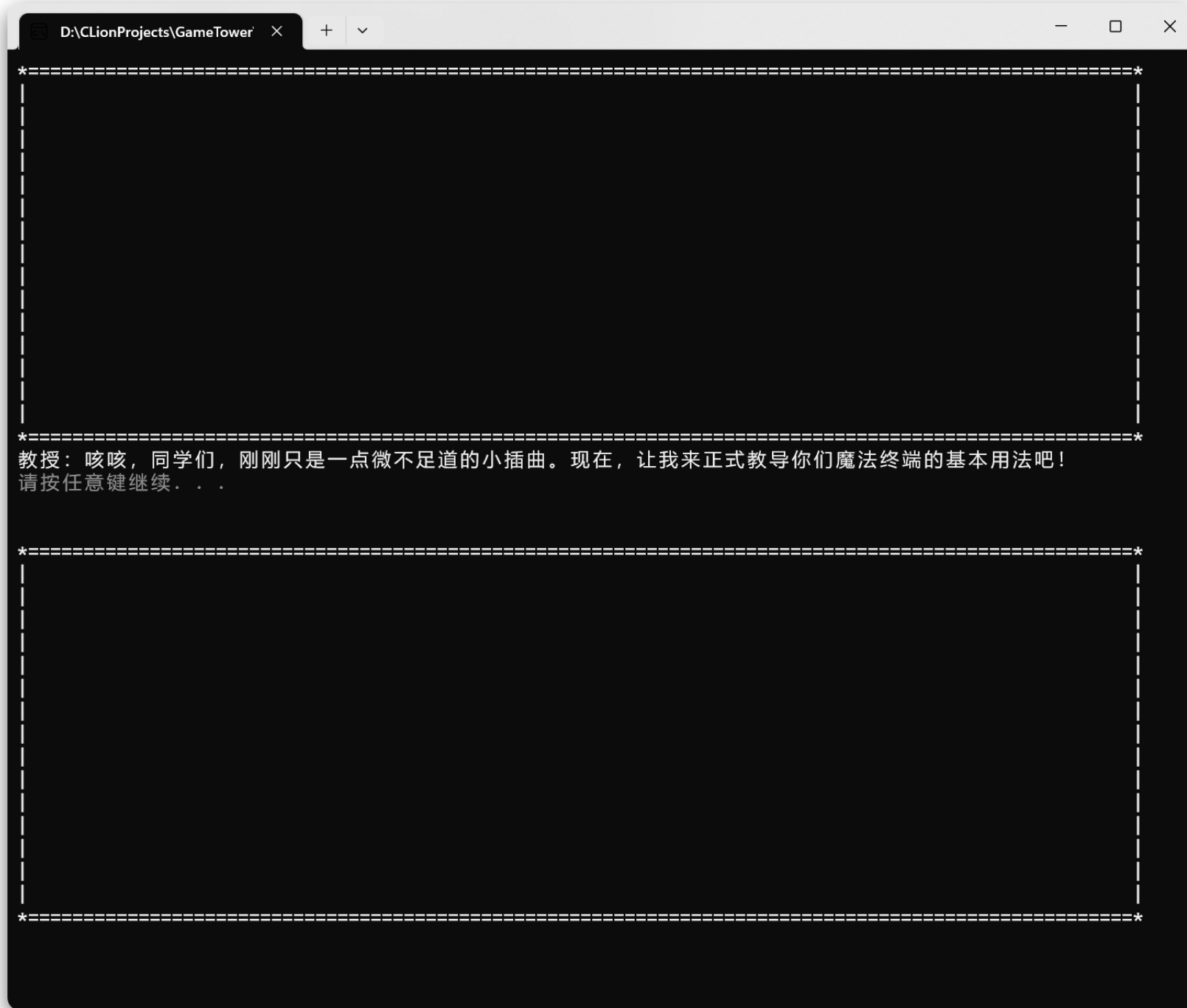


剧情

读取外部文件



战斗场景

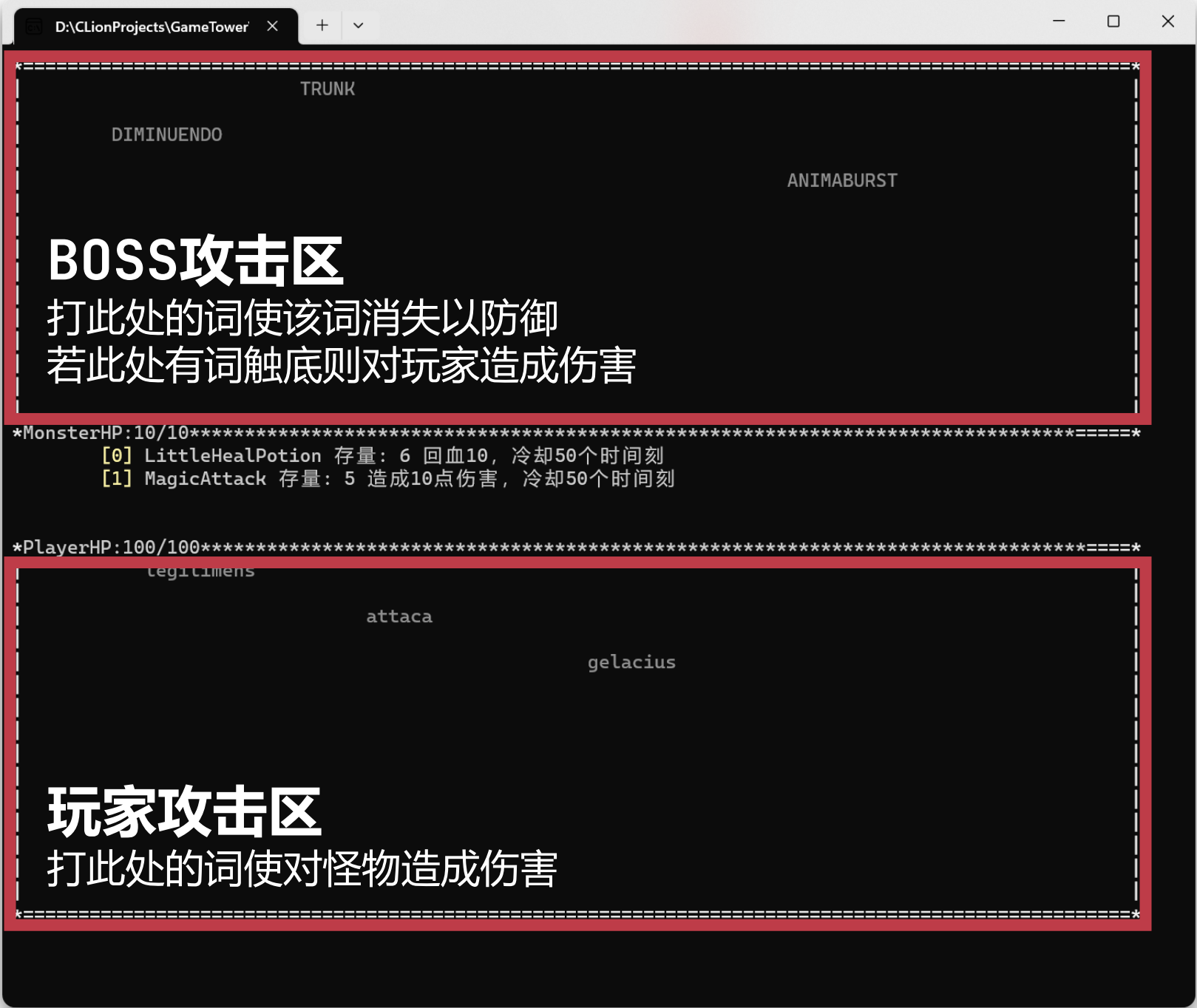


怪物属性

HP 血量
攻击词
伤害

物品区

打出物品前的编号
使用物品

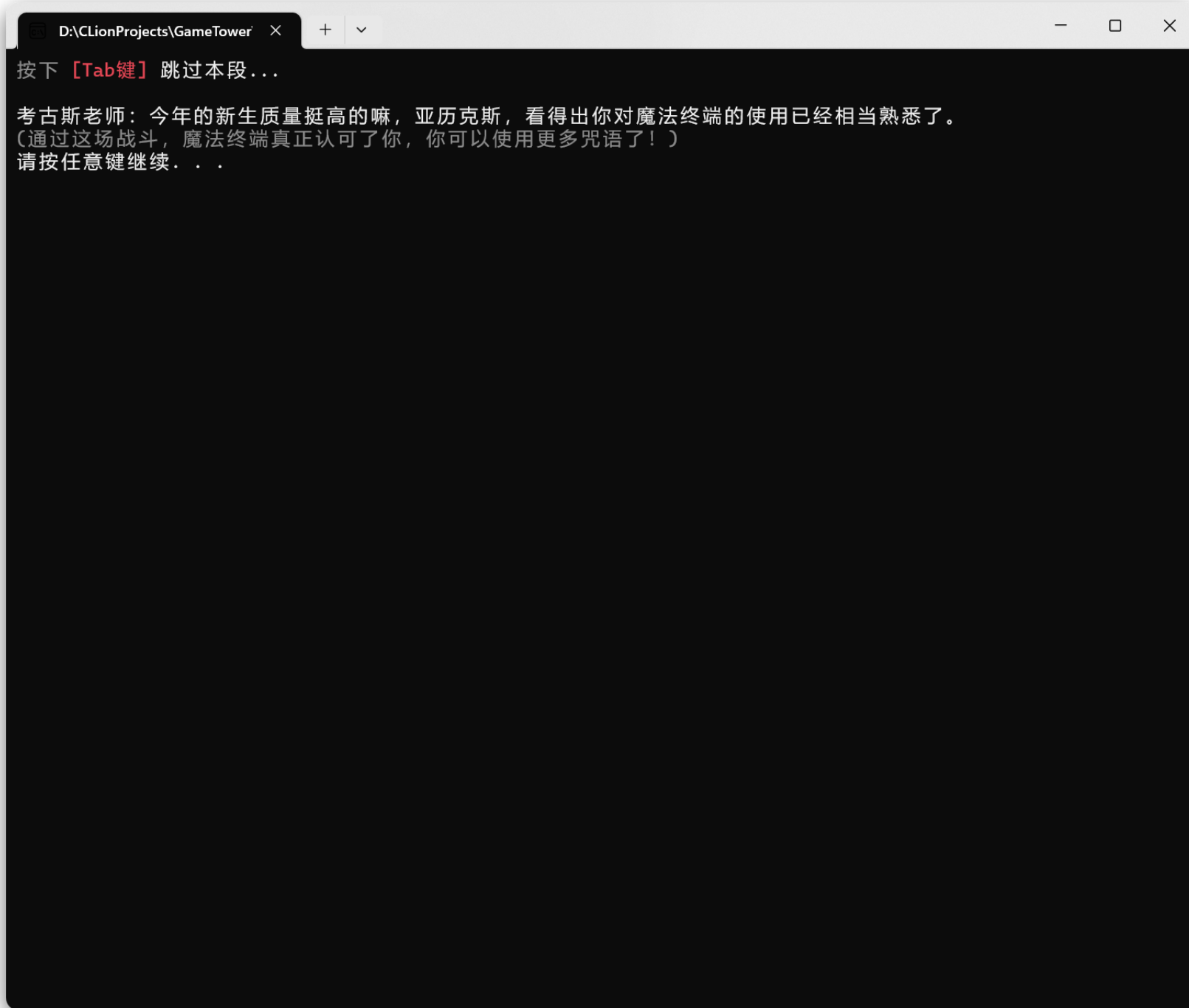


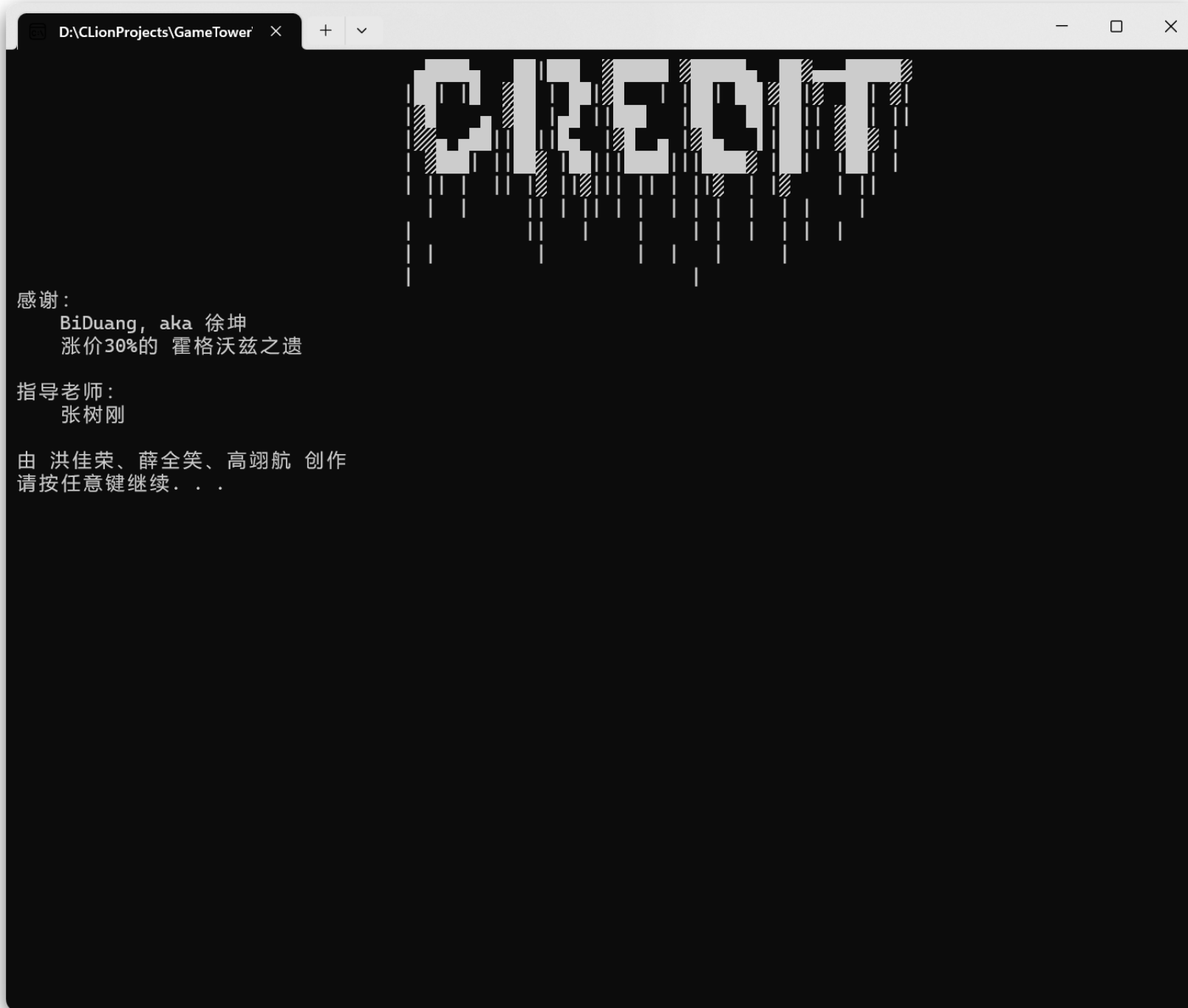
BOSS血条
玩家血条

5

阶段系统

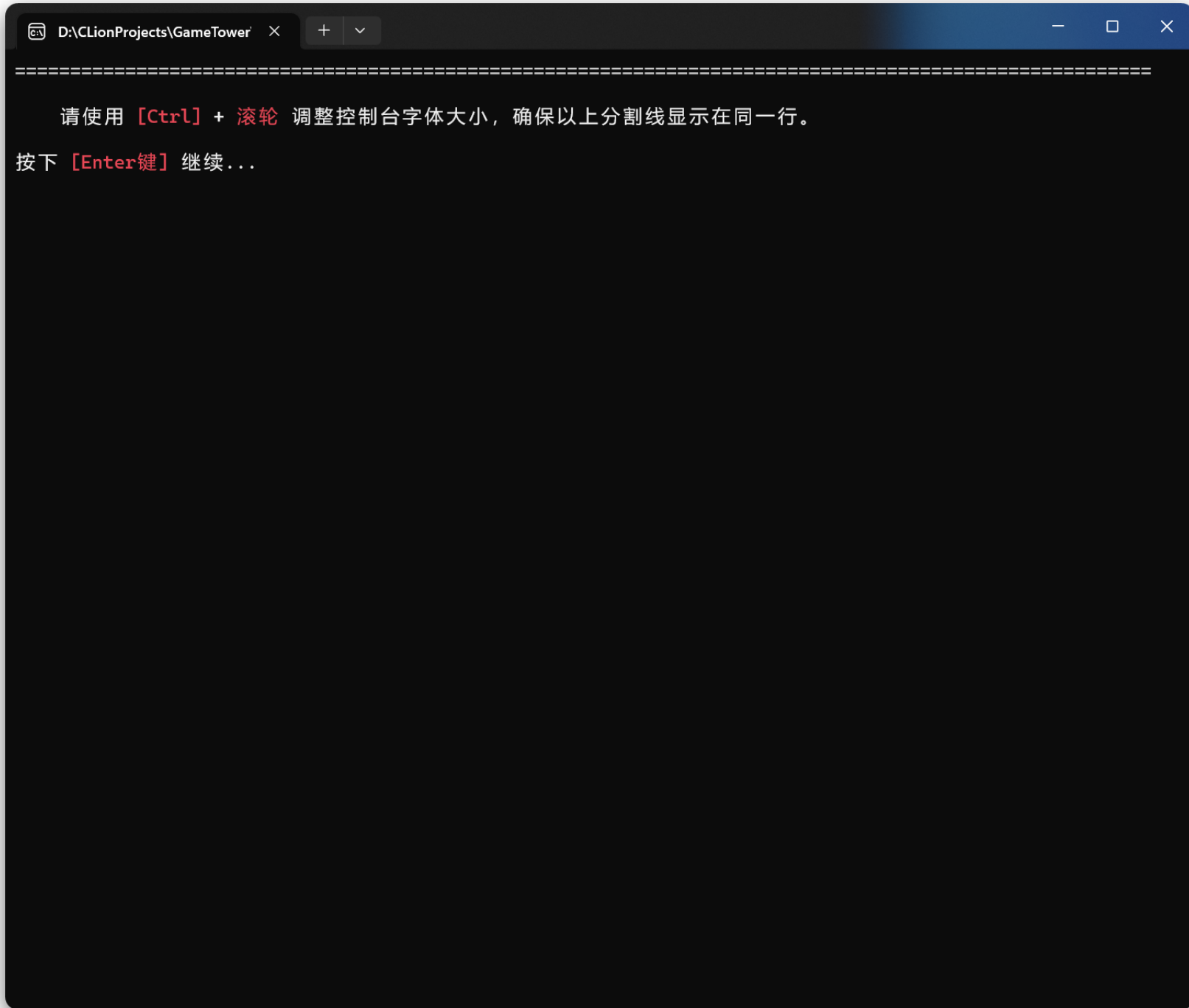
每打过一个BOSS
进度前进一阶段







POTTER 2077, 启动



窗口大小

- 程序开启时已自动调整好窗口大小和窗口位置，但因不同电脑的默认缩放比例不同导致DPI不同，仍然可能会显示错位。
- 提示玩家调整窗口字体大小。



按下[Tab]跳过

```
bool flagTab = false;
for (char c: msg) {
    cout << c;
    if (!flagTab) Sleep(5);
    if (kbhit()) {
        char c;
        c = getch();
        if (c == '\t') {
            flagTab = true;
        }
    }
}
```

按下 **[Tab键]** 跳过本段...

故事发生在未来的魔法世界。2077年，魔法社会已经发生了巨大的变化。魔法与高科技融合，成为一个充满危险与奇迹的新时代。哈利·波特的孙子，亚历克斯·波特 (Alex Potter) 是一名年轻有为的魔法学徒，在这个世界中展开他的冒险之旅——



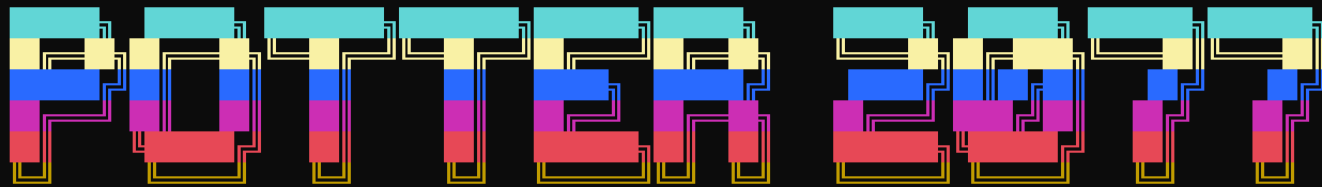
按下 **[Enter键]** 继续...

LOGO部分

```
PosControl::setPos(x, y);  
ifstream logoFile("./Assets/.logo");  
string logo;  
while (getline(logoFile, logo)) {  
    int color = randInt(1, 15);  
    SetConsoleTextAttribute(GetStdHandle  
        (STD_OUTPUT_HANDLE), color);  
    Sleep(50);  
    cout << space << logo << endl;  
}  
logoFile.close();
```

按下 **[Tab键]** 跳过本段...

故事发生在未来的魔法世界。2077年，魔法社会已经发生了巨大的变化。魔法与高科技融合，成为一个充满危险与奇迹的新时代。哈利·波特的孙子，亚历克斯·波特 (Alex Potter) 是一名年轻有为的魔法学徒，在这个世界中展开他的冒险之旅——



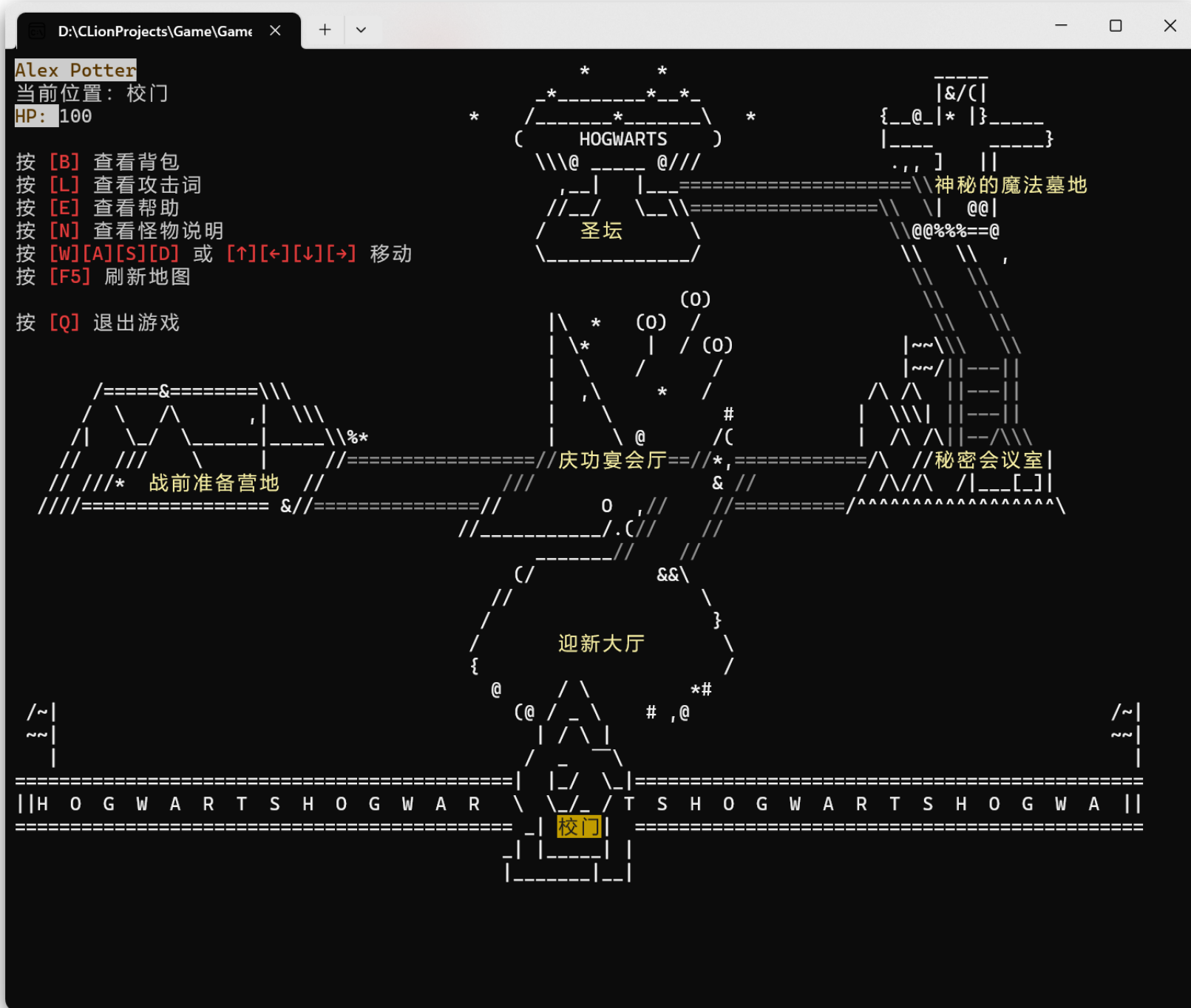
你好，名为 Jiarong 的 Alex Potter!

使用 **[W]** **[S]** 或 **[↑]** **[↓]** 选择，按 **[Enter键]** 确认

- > 新游戏
- > 加载游戏
- > 退出游戏

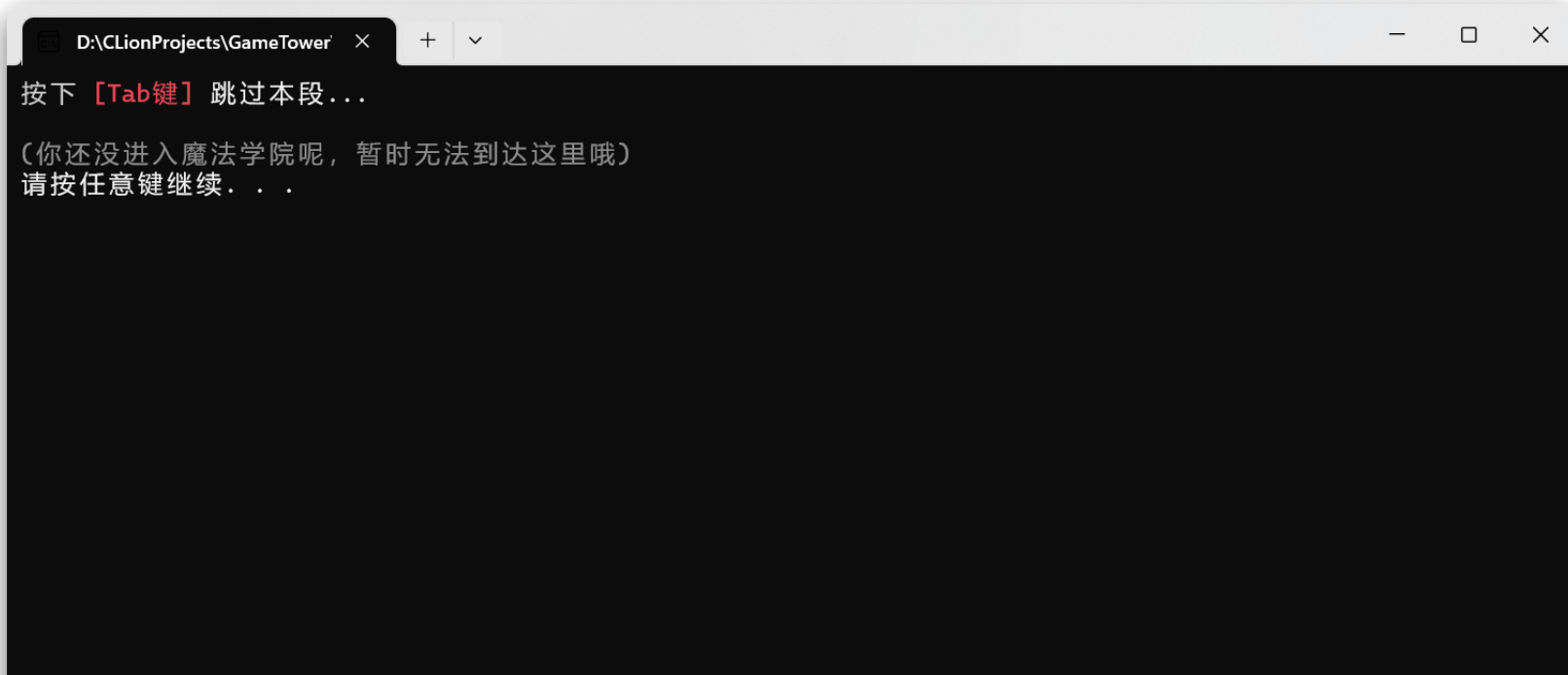
菜单部分

```
c = getch();
switch (c) {
    case 'w':
    case 'W':
    case 72: // 72是上箭头
        prevChoice = choice;
        choice = (choice + length -
1) % length;
        break;
    case 's':
    case 'S':
    case 80: // 80是下箭头
        prevChoice = choice;
        choice = (choice + 1) %
length;
        break;
    case '\r':
    case '\n':
        return choice;
    default:
        prevChoice = choice;
        break;
}
```



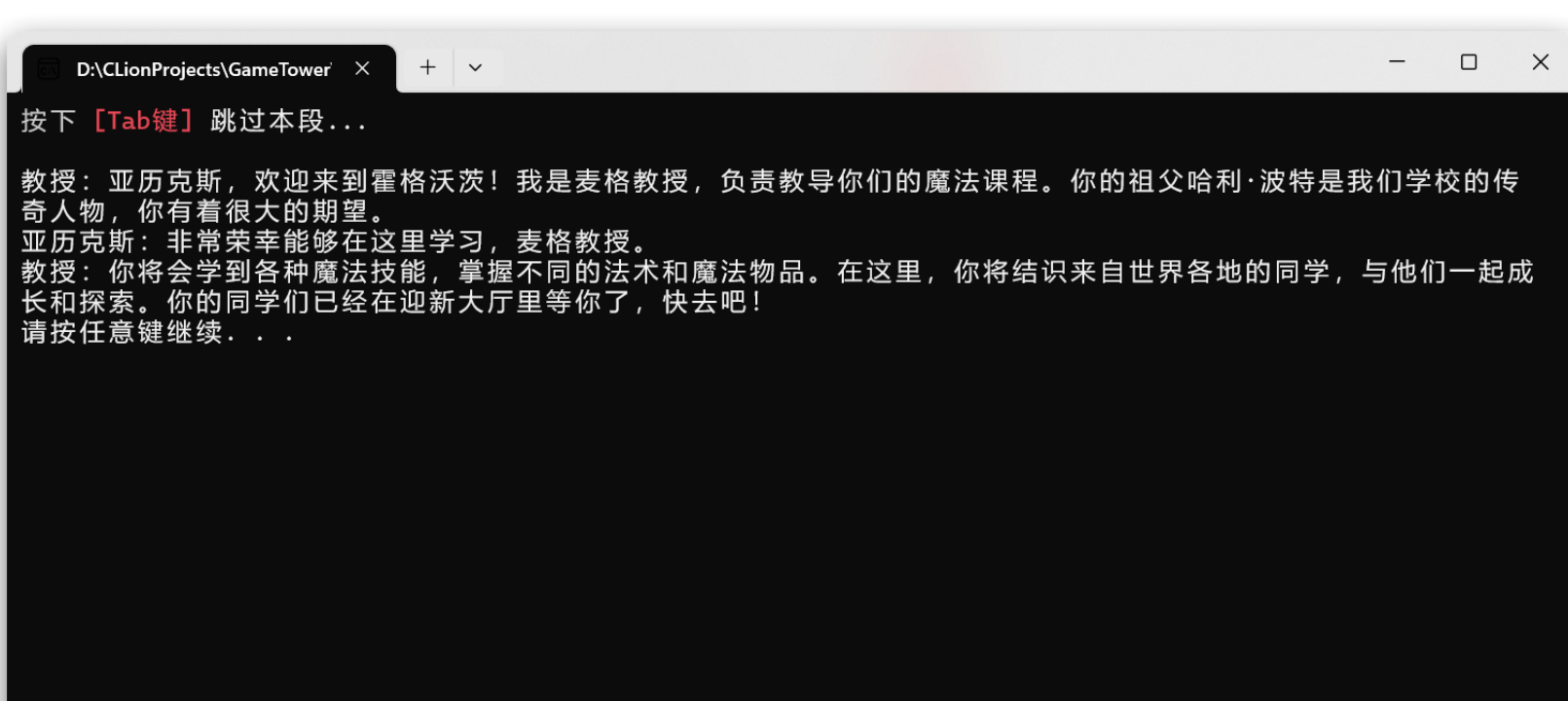
位置移动

- 控制地图显示在同一控制台窗口
- 当前位置高亮显示
- 染色



剧情显示

- 复杂，需要特别判定的地方多



- [0] 先进入大门;
如果在进入大门之前去之后任何地方都触发Other/进门前.txt
- [1] 去迎新大厅;
如果去迎新大厅之前去了任何地方都触发Other/进迎新大厅前.txt
- [2] 去庆功宴会厅读剧情后, 移动到秘密会议室;
如果移动到任何地方都触发Other/探险前.txt
- [3] 去神秘的魔法墓地
如果去神秘的魔法墓地之前去任何地方都会触发Other/战斗意志.txt
- [4] 去庆功宴会厅, 触发 Other/决战信息.txt
此时如果去其他地方都触发各地的default

在 下一步 之前任何时候去 战前准备营地都触发 Other/营地未建立.txt
- [4] 去战前准备营地, 触发 Other/战前准备.txt,

此时如果去其他除了圣坛的地方都触发各地的default。但是如果去了圣坛, 则触发 Other/去看看吧.txt
- [4] 去圣坛, 触发 BeforeFight/圣坛.txt --> AfterFight/圣坛.txt
游戏流程到此结束, 此时如果再来圣坛触发 Other/决战后的圣坛 , 如果去其他地方则触发default


```

if (getIsLocked(pos.line, pos.column)) {

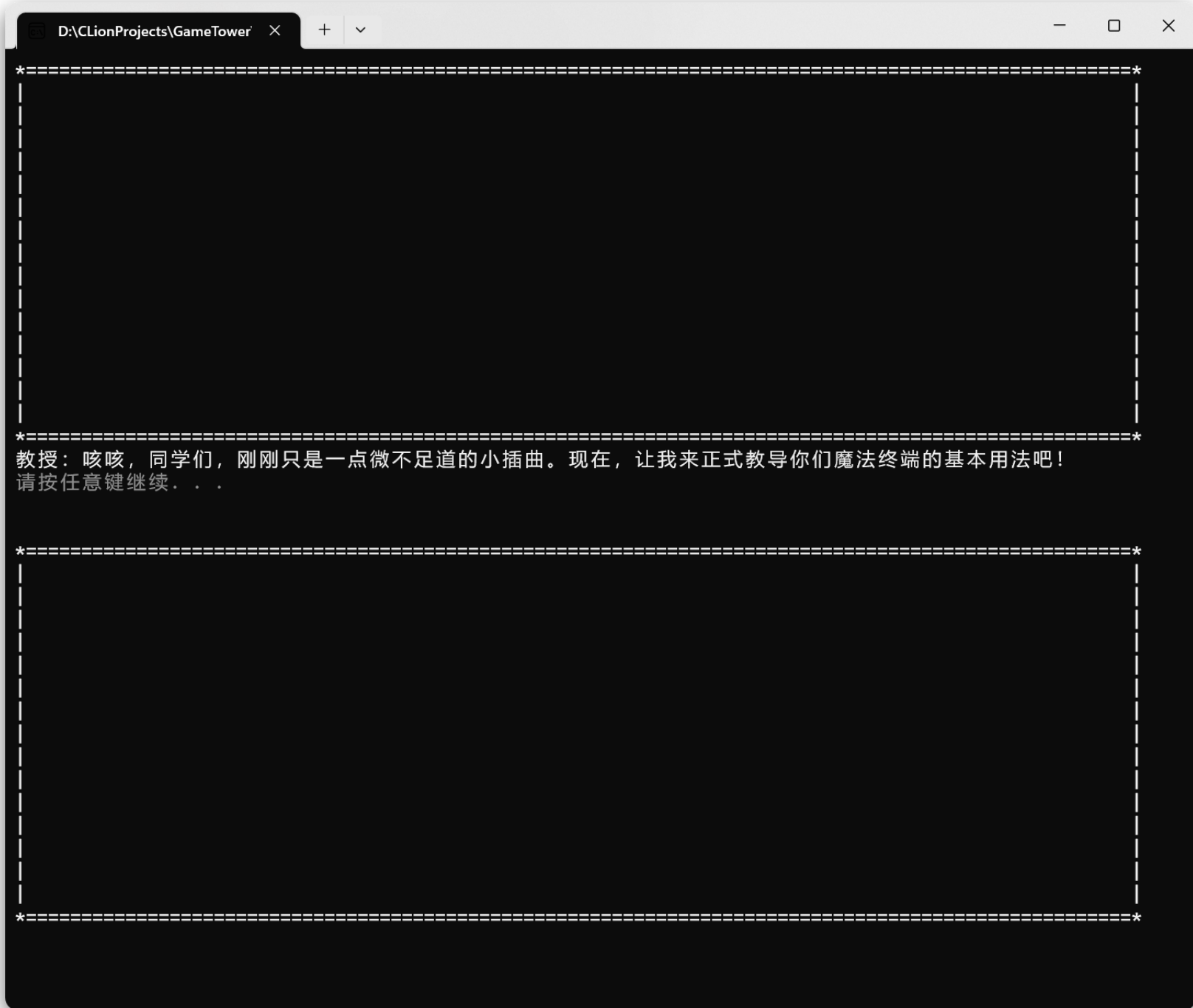
    if (map.getProgress() < 3 && pos.line == 4 && pos.column == 2) {
        printMsg( msgDir: "./Assets/Scene/Other/营地未建立.txt");
    } else if (map.getProgress() == 4 || map.getProgress() == 5) {
        printMsg( msgDir: "./Assets/Scene/Default/" + getPlaceName(pos.line, pos.column) + ".txt");
    } else {
        printMsg( msgDir: map.getDefaultMsgDir());
    }
    system( Command: "pause");
} else {
    if (ifHasDone(pos.line, pos.column)) {
        printMsg( msgDir: "./Assets/Scene/Default/" + getPlaceName(pos.line, pos.column) + ".txt");
        system( Command: "pause");
    } else {
        if (map.getProgress() == 2 && pos.line == 4 && pos.column == 4) {
            printMsg( msgDir: "./Assets/Scene/Background/" + getPlaceName(pos.line, pos.column) + ".txt");
            system( Command: "pause");
            system( Command: "cls");
            lastPos = { .line: 4, .column: 4}; // 庆功宴会厅
            pos = { .line: 4, .column: 6}; // 秘密会议室
            printMsg( msgDir: "./Assets/Scene/Background/" + getPlaceName(pos.line, pos.column) + ".txt");
        } else if (map.getProgress() == 2 && pos.line == 4 && pos.column == 6) {
            lastPos = { .line: 4, .column: 4};
            pos = { .line: 4, .column: 4};
            printMsg( msgDir: "./Assets/Scene/Background/" + getPlaceName(pos.line, pos.column) + ".txt");
            system( Command: "pause");
            system( Command: "cls");
            pos = { .line: 4, .column: 6};
            printMsg( msgDir: "./Assets/Scene/Background/" + getPlaceName(pos.line, pos.column) + ".txt");
        } else if (map.getProgress() == 3 && pos.line != 2 && pos.column != 6) {
            printMsg( msgDir: "./Assets/Scene/Other/战斗意志.txt");
        } else if (map.getProgress() == 4 && pos.line == 4 && pos.column == 4) {
            if (getIsLocked(2,4))
                printMsg( msgDir: "./Assets/Scene/Other/决战信息.txt");
            else printMsg( msgDir: "./Assets/Scene/Default/" + getPlaceName(pos.line, pos.column) + ".txt");
        } else if (map.getProgress() < 4 && pos.line == 2 && pos.column == 4) {
            printMsg( msgDir: "./Assets/Scene/Other/去看看吧.txt");
        } else printMsg( msgDir: "./Assets/Scene/Background/" + getPlaceName(pos.line, pos.column) + ".txt");
        system( Command: "pause");

        if (map.getProgress() == 4 && pos.line == 4 && pos.column == 2) {
            player.getBackpack().progress4();
            dynamic_cast<Place*>(map.getLocation( line: 4, column: 2))->setHasDone(true);
            dynamic_cast<Place*>(map.getLocation( line: 2, column: 4))->setIsLocked(false);
            dynamic_cast<Place*>(map.getLocation( line: 4, column: 4))->setIsLocked(true);
        }

        if ((pos.line == 4 && pos.column == 2) ||
            (pos.line == 4 && pos.column == 4));
    }
}

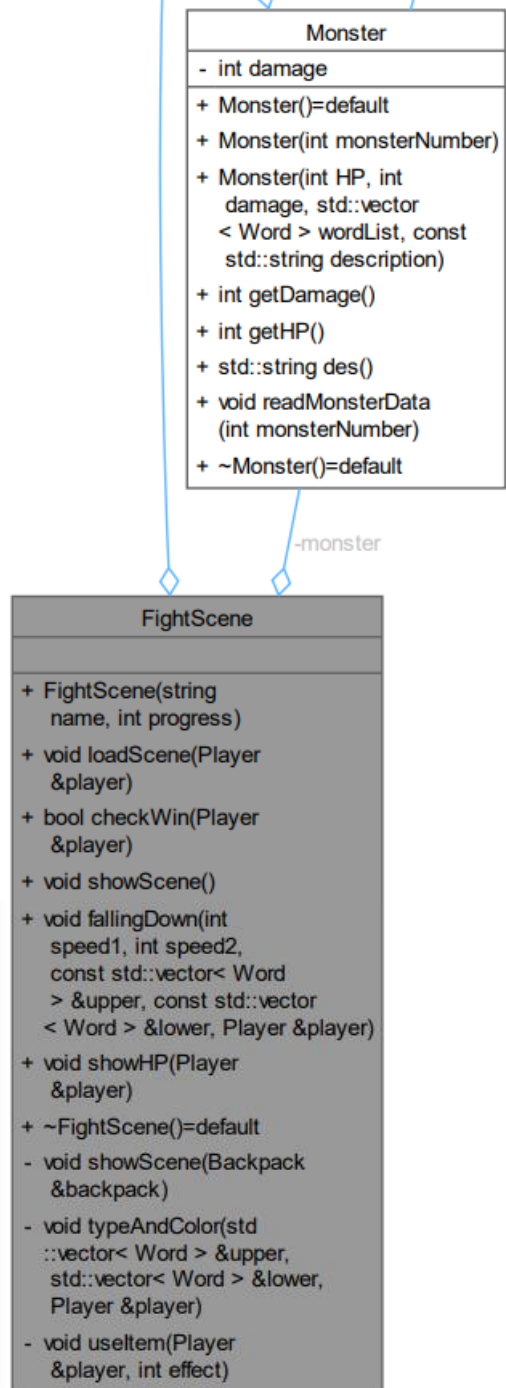
```

- 约 **60** 行代码特判
- 通过结合各个地点的
isLocked标签和
hasDone标签来实现复
杂的剧情控制

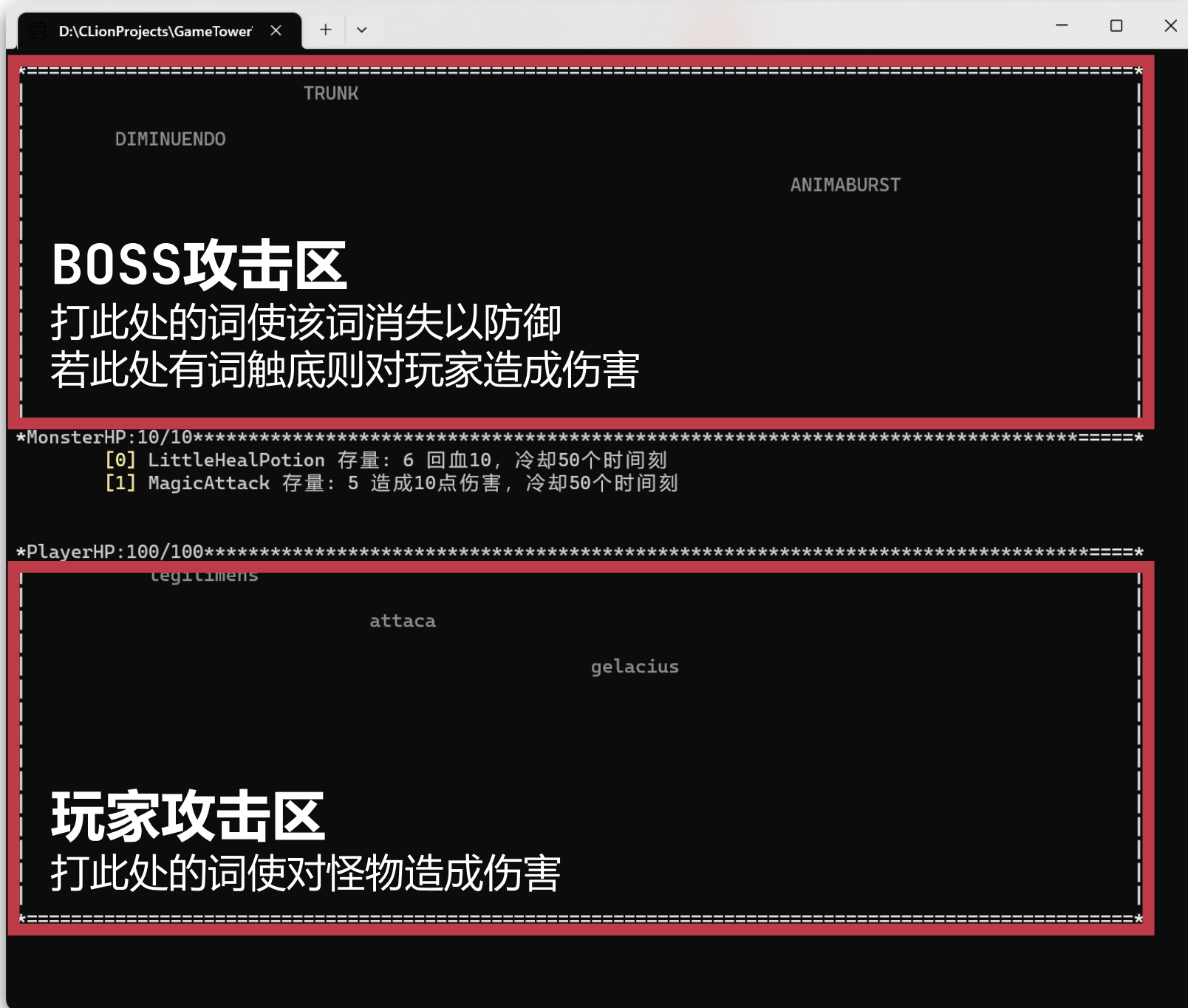


创建战斗场景

- 避免额外的资源开销并为了方便数值修改，战斗场景独立于地图上的地点创建，当需要战斗时才创建



- 传入当前阶段序号
- 构造怪物（即Boss）
- 打印战斗界面
- 输出战斗前剧情
- 战斗
- 输出战斗后剧情



物品区

打出物品前的编号
使用物品

BOSS血条
玩家血条

BOSS攻击区

打此处的词使该词消失以防御
若此处有词触底则对玩家造成伤害

```
std::vector<Word> upper, lower; // 核心单词表, 直接源自monster和player, 分别代表上下  
upper = monster.deliverWord();  
lower = player.deliverWord();
```

```
showHP(& player); // 显示初始血量
```

玩家攻击区

打此处的词使对怪物造成伤害

Word1

Word2

Word3

Word4

Word5

玩家攻击区

打此处的词使对怪物造成伤害

```
while (cur < upper.size()) {//这是第一部分，词还没有完全进入shown区域内。
```

```
    if (shownUpper[shownUpper.size() - 1].getState() == 0 && shownUpper[shownUpper.size() - 1].get  
        player.getDamaged( damage: monster.getDamage());
```

```
    showHP( &: player);    词是否触底，通过词语的state判定
```

```
}//人物扣血的判定
```

```
for (int i = shownUpper.size() - 1; i > 0; i--) {
```

```
    shownUpper[i] = shownUpper[i - 1];
```

```
    shownLower[i] = shownLower[i - 1];
```

```
}//替换词，向下一行
```

使用offset控制加词频率

```
offset = (offset + 1) % 2;//offset在0和1之间变换，用来实现每两回合加入一个单词的设定
```

```
if (offset) {
```

```
    shownUpper[0] = upper[cur];
```

```
    shownLower[0] = lower[cur];
```

```
    cur++;
```

```
}
```

```
for (int i = offset; i < shownUpper.size(); i += 2) {//有重复加入的bug，所以把应该是白板的地方重新塞
```

```
    shownUpper[i] = blank;
```

```
    shownLower[i] = blank;
```

```
}
```

```
for (int i = 0; i < shownUpper.size(); i++) {
```

```
    PosControl::setPos( x: i + 1, y: 1);
```

```
    shownUpper[i].putWord();
```

控制光标坐标到边框内
输出词语

~ 打此处的词使对怪物造成伤害

Word1

Word2

Word3

Word4

Word5

玩家攻击区

打此处的词使对怪物造成伤害

Word1

Word2

Word3

Word4

Word5

玩家攻击区

打此处的词使对怪物造成伤害

```
int t = 100; 下落速度
char next = ' '; //next是下一个键盘敲得字符
int itemClock[10] = {0}; 物品cd钟
```

```
while (t-- >= 0) {
```

```
    Sleep( dwMilliseconds: 5); 进一步控制每次下落前检测的tick长度
```

```
    next = ' ';
```

```
    if (_kbhit()) { //get 键盘字符
```

```
        next = _getch();
```

```
    }
```

```
    if (next <= 'z' && next >= 'a') //小写字母, 对应下面
```

检测当前打了什么字母, 比对字母和那一个词的状态
再进一步判定是否将打出长度+1或清0

```
        bool existWrong = 0, existUnfinished = 0; //两个状态量, 本次循环中是否存在打错, 是否存在打了一半的词
```

```
        for (int i = lower.size() - 1; i >= 0; i--) { //未打完的字具有优先性, 所以先循环一遍找有没有未打完的
```

```
            int cur = lower[i].getCur(); //这里的cur是词的状态, 表示当前打了几个字符, 通常为0;
```

```
            if (!lower[i].getState() && cur != 0) { //如果有一个词没打完, cur也不是0, 说明找到了未打完的词
```

```
                existUnfinished = 1; //改状态
```

```
                if (lower[i].getString()[cur] == next) { //如果下一位和键盘敲得匹配成功
```

```
                    lower[i].changeColor( pos: lower[i].getCur(), co: 5); //染色
```

```
                    lower[i].changeCur( cur: cur + 1); //进一位
```

```
                    updatePrint(upper, lower); //重新输出, 更新显示的颜色
```

```
                    int state = deeper( &t, &lower[i], &upper, &lower); //调用deeper, 对当前单
```

```
                    if (state == -1) { //deeper进行时打错字了
```

```
                        existWrong = 1; //存在错误
```

```
                        existUnfinished = 0; //打错了回到解放前, 所以不存在没有完成的词了
```

```
                        break;
```

```
void Word::putWord() {
```

```
    for (int i = 0; i < position; i++) std::cout << ' ';
```

```
    if (isFinished) {
```

```
        for (int i = 0; i < length; i++) {
```

```
            cout << ' ';
```

```
        }
```

```
        return;
```

```
    for (int i = 0; i < length; i++) {
```

```
        SetConsoleTextAttribute( hConsoleOutput: GetStdHandle( nStdHandle: STD_OUTPUT_HANDLE), wAttributes
```

```
        cout << word[i];
```

```
    }
```

```
    SetConsoleTextAttribute( hConsoleOutput: GetStdHandle( nStdHandle: STD_OUTPUT_HANDLE), wAttributes: 8)
```

如果词被打完了就输出空格清空

否则按打出长度染色

玩家攻击区

打此处的词使对怪物造成伤害

```

void Map::setProgress(int progress) {
    currProgress = std::min(5,progress);
    // set the progress (unlock the places)
    switch (progress) { // 倒序放置，当导入新进度时，前面的场景也能解锁
        case 4:

            dynamic_cast<Place*>(locations[4][2])->setIsLocked(false);
            dynamic_cast<Place*>(locations[2][6])->setHasDone(true);
            dynamic_cast<Place*>(locations[4][4])->setIsLocked(true);
            dynamic_cast<Place*>(locations[4][4])->setHasDone(true);

        case 3:
            dynamic_cast<Place*>(locations[2][6])->setIsLocked(false);
            dynamic_cast<Place*>(locations[4][4])->setHasDone(false);
            dynamic_cast<Place*>(locations[4][6])->setHasDone(true);

        case 2:
            dynamic_cast<Place*>(locations[4][4])->setIsLocked(false);
            dynamic_cast<Place*>(locations[4][6])->setIsLocked(false);
            dynamic_cast<Place*>(locations[6][4])->setHasDone(true);

        case 1:
            dynamic_cast<Place*>(locations[6][4])->setIsLocked(false);
            dynamic_cast<Place*>(locations[8][4])->setHasDone(true);

        case 0:
            dynamic_cast<Place*>(locations[8][4])->setIsLocked(false);

```

```

        void setHasDone(int line, int column) {
            dynamic_cast<Place*>(locations[line][column])->setHasDone(true);
            if (line == 2 && column == 4) dynamic_cast<Place*>(locations[4][4])->setHasDone(true);
            setProgress(currProgress + 1);
        }

```

```

// set the default message directory, using "break"
switch (progress) {
    case 0:
        currDefaultMsgDir = "../Assets/Scene/Other/进门大厅.txt";
        break;

    case 1:
        currDefaultMsgDir = "../Assets/Scene/Other/进迎新大厅前.txt";

```

按下 **[Tab键]** 跳过本段...

亚历克斯：大家都在都在忙碌。

罗德·韦斯莱：是啊，在战前准备营地，每个人都有要忙的事。

亚历克斯：走吧，我们去领东西。

〔来到物资供应处〕

赫尔姆斯：嘿，小子们，我就知道你们也会来！

亚历克斯：赫尔姆斯先生！

赫尔姆斯：来，拿上你们的東西，好好准备战斗吧！

〔你获得了 生命药水*3 蠕虫病毒.exe*2〕

*生命药水：使用后 HP+50，冷却时间20s

*蠕虫病毒.exe：使用后 敌方HP-80，冷却时间30s

赫尔姆斯：对了，还有这个〔眨眼〕

〔你获得了 最终咒语 已自动同步到魔法终端。你获得了最强的攻击咒语和防御咒语！〕

现在前往 圣坛 迎接战斗吧！

请按任意键继续...

```
D:\CLionProjects\游戏\Game1 x + v - □ ×  
[15] revelio 伤害: 1  
[16] partis 伤害: 1  
[17] temporus 伤害: 1  
[18] ventuso 伤害: 1  
[19] aquaralis 伤害: 1  
[20] incendaria 伤害: 1  
[21] ventifero 伤害: 1  
[22] gelacius 伤害: 1  
[23] aerolitus 伤害: 2  
[24] pyrosium 伤害: 1  
[25] silvarum 伤害: 1  
[26] nocturnis 伤害: 1  
[27] solaris 伤害: 1  
[28] engorgio 伤害: 3  
[29] diminuendo 伤害: 3  
[30] morsmordre 伤害: 3  
[31] waddiwasi 伤害: 3  
[32] diffindo 伤害: 3  
[33] relashio 伤害: 3  
[34] locomotor 伤害: 3  
[35] trunk 伤害: 3  
[36] aguamenti 伤害: 3  
[37] erecto 伤害: 3  
[38] deprimio 伤害: 3  
[39] immobulus 伤害: 3  
[40] luminara 伤害: 3  
[41] umbrosus 伤害: 3  
[42] aetherium 伤害: 10  
[43] spiritus 伤害: 3  
[44] luminosus 伤害: 3  
[45] astralis 伤害: 17  
[46] florifera 伤害: 3  
[47] elementum 伤害: 3  
[48] animosus 伤害: 12  
[49] silvestris 伤害: 3  
[50] quietus 伤害: 3  
[51] portus 伤害: 3  
[52] specialis 伤害: 3  
[53] revelio 伤害: 3  
[54] tergeo 伤害: 7  
[55] erecto 伤害: 3
```

[26] nocturnis 伤害: 1
[27] solaris 伤害: 1
[28] engorgio 伤害: 3
[29] diminuendo 伤害: 3
[30] morsmordre 伤害: 3
[31] waddiwasi 伤害: 3
[32] diffindo 伤害: 3
[33] relashio 伤害: 3
[34] locomotor 伤害: 3
[35] trunk 伤害: 3
[36] aguamenti 伤害: 3
[37] erecto 伤害: 3
[38] deprimio 伤害: 3
[39] immobilus 伤害: 3
[40] luminara 伤害: 3
[41] umbrosus 伤害: 3
[42] aetherium 伤害: 10
[43] spiritus 伤害: 3
[44] luminosus 伤害: 3
[45] astralis 伤害: 17
[46] florifera 伤害: 3
[47] elementum 伤害: 3
[48] animosus 伤害: 12
[49] silvestris 伤害: 3
[50] quietus 伤害: 3
[51] portus 伤害: 3
[52] specialis 伤害: 3
[53] revelio 伤害: 3
[54] tergeo 伤害: 7



感谢：

BiDuang, aka 徐坤
涨价30%的 霍格沃兹之遗

指导老师：

张树刚

由 洪佳荣、薛全笑、高翊航 创作