# ­­Hero Era Battleground

By Jed

www.jed123.com

[“设计一个新英雄”大项目技术报告 1](#_Toc516390350)

[〇、 术语说明 2](#_Toc516390351)

[一、 需求分析 2](#_Toc516390352)

[这里介绍本次项目的需求，同时包含本次大项目的功能概述。 2](#_Toc516390353)

[ 英雄属性和行为 2](#_Toc516390354)

[ 装备属性 2](#_Toc516390355)

[ 战场属性 3](#_Toc516390356)

[二、 实现思路 3](#_Toc516390357)

[ 功能结构 3](#_Toc516390358)

[ 包含和继承关系 3](#_Toc516390359)

[三、 数据设计 4](#_Toc516390360)

[ 装备系列类 4](#_Toc516390361)

[ 英雄（Hero）类 4](#_Toc516390362)

[ 战场（Battleground）类 6](#_Toc516390363)

[ 简易UI（非类） 6](#_Toc516390364)

[ 常量（非类） 6](#_Toc516390365)

[四、 本文档涉及到的常量及数据计算方法 7](#_Toc516390366)

[ 伤害计算 7](#_Toc516390367)

[ 暴击率计算 7](#_Toc516390368)

[ 暴击加成计算 7](#_Toc516390369)

[ 攻击敌人获得金钱计算 7](#_Toc516390370)

[ 合成装备 7](#_Toc516390371)

[ 卖出装备收入计算 7](#_Toc516390372)

[ 得分计算 8](#_Toc516390373)

[五、 附录 8](#_Toc516390374)

[ 测试代码 8](#_Toc516390375)

[ 预置的装备库存 9](#_Toc516390376)

[ 游戏帮助（print\_help） 9](#_Toc516390377)

[ 更新日志 10](#_Toc516390378)

**请先阅读readme.md。**

## 术语说明

为避免歧义，特此说明本文档中部分容易混淆的术语。

1. **法术**：此词语与示例文档中的“魔法“一词完全等价，本文档中前者完全替换后者。
2. **基础攻击（防御）力**：是英雄自身的属性，默认为10。当英雄创建后，该值任何情况下都不会改变。也就是说英雄的基础攻击（防御）力与装备无关。
3. **物攻**：物理攻击，对玩家显示为Attack Damage，源码中相应函数或变量名为phyAtk。
4. **法攻**：法术攻击，与魔法攻击一词完全等价。对玩家显示为Ability Power，源码中响应函数或变量名为magAtk。
5. **物抗/法抗**：对玩家分别显示为Armor和Magic Defence，源码中响应函数或变量名为phyDef和magDef。
6. **合成装备**：作为名词，即为攻击型装备与防御型装备的综合体，故又名“**综合型装备**“；作为动词，即为花费额外金钱将两种装备合二为一的动作。详细定义见下文。
7. **购买装备**：购买装备即购买并穿戴，效果是花钱提升自身属性。不存在买了但是不穿戴的情况。
8. **卖出装备**：卖出装备即脱下装备并卖了换钱。不存在脱下但是不卖掉的情况。
9. **装备库存**：不同于英雄的装备栏。装备库存指的是预先定义好的若干装备，英雄购买装备时，实际上是从装备库存中购买的。

## 需求分析

### 这里介绍本次项目的需求，同时包含本次大项目的功能概述。

### 英雄属性和行为

基本**数值类属性**：生命值、基础攻击（物攻和法攻）、基础防御（物抗和法抗）、金钱数量、装备数量、暴击率；

附加**非数值属性**：装备栏。

**查询**：场外行为。以表格方式打印出英雄的属性、装备等。

**攻击**：主动行为。可以对其它英雄对象造成攻击，目的是减少对方的生命值。

**购买装备**：主动行为。减少金钱数量，给装备栏增加内容，目的是提升自己的攻击或防御力。

**合成装备**：主动行为。花额外的钱将两个装备合二为一，目的是减少装备总数（因为一个英雄最多同时穿戴三个装备）。

**卖出装备**：主动行为。将装备脱下并卖掉换钱，得到装备原价的60%。

**回血**：主动行为。花费金钱，增加自己的生命值。

**防御**：被动行为。当自己受到其它英雄攻击时触发，目的是减弱对方给自己数值属性造成的削减，下文有详细的计算方法。

**状态检查**：被动行为。每回合任何一方操作过后都对对方的存活/死亡状态进行检查。

### 装备属性

**价格**：购买装备所需要的金钱数。

**加成**：一个具体的装备应该能够提升英雄的攻击或防御力。当然，这里说的“提升”并不是直接增加英雄的基础攻击或防御力那个变量，而是在真正发生攻击行为时，使用“基础攻击或防御力+装备加成”这样的方式来进行计算。对应函数为get\_real\_phyAtk等四个。计算方法见“数据设计”板块。

### 战场属性

**回合数**：玩家英雄和敌人英雄个进行一次操作视为一回合，每个完整的回合内都要进行两次英雄状态检查。

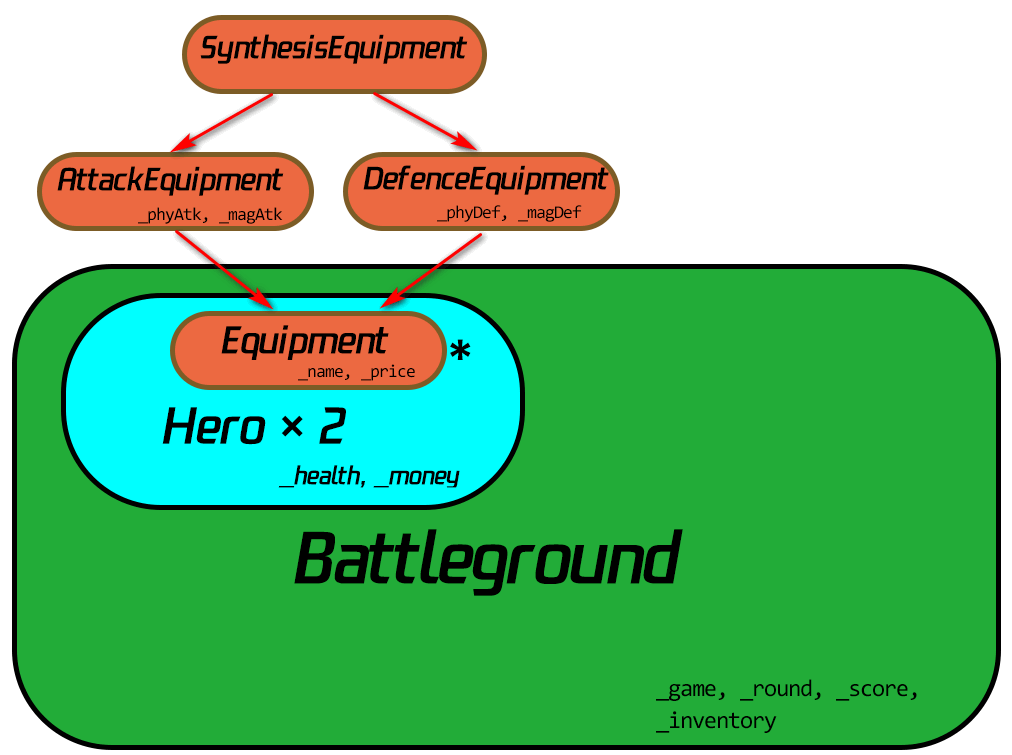
**局数**：一局可能包含多个回合。一局游戏中，如果有一方英雄死亡，那么该局结束。由玩家选择是否再来一局。

**装备库存**：内有预先定义好的装备。详细内容见本文档末尾附录。

## 实现思路

### 功能结构

### 包含和继承关系

****战场类（Battleground）中包含两个英雄，分别是由用户控制的**玩家英雄**和由电脑控制的**敌人英雄**，此外还有表示**装备库存**、**当前局数**、**当前回合数**、**玩家总得分**等成员变量。

一个英雄类（Hero）含有诸如生命值、各种攻击防御力的数据成员，同时还有含有指向装备的指针的vector用于储存该英雄已购买的装备。

装备类（Equipment）派生出攻击型装备类（AttackEquipment）和防御型装备类（DefenceEquipment），后二者又共同派生出综合型装备类（SynthesisEquipment）。英雄类中的vector储存的是Equipment\*，即指向基类的指针，由它可以调用任何派生装备（向上造型），在使用装备时，再用dynamic\_cast向下造型即可知道该装备是什么类型的装备。

图注说明：大框在小框外围表示**包含关系**，星号表示**以指针的方式包含**；红色箭头表示**继承关系**，由派生类指向基类。每个类右下角以下划线开头的单词是该类中较为重要的**数据成员**（并非所有成员都出现在了上图中）。

## 数据设计

### 装备系列类

装备系列类包含8个文件：Equipment.h(.cpp)、AttackEquipment.h(.cpp)、DefenceEquipment.h(.cpp)、SynthesisEquipment.h(.cpp)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 间接派生类 | 直接派生类 | 基类 | 成员 | 数据类型 | 备注 |
| **合**  **成**  **装**  **备** | **攻**  **击**  **装**  **备** | **装备** | \_name，装备名称 | string | 装备的名称 |
| \_price，装备价格 | price | 购买装备需要花费的金钱数 |
|  | \_phyAtk，物攻加成 | int | 物理攻击数值增量 |
| \_magAtk，法攻加成 | int | 法术攻击数值增量 |
| **防**  **御**  **装**  **备** | **装备[2]** | \_name，装备名称 | string | 装备的名称 |
| \_price，装备价格 | price | 购买装备需要花费的金钱数 |
|  | \_phyDef，物抗加成 | int | 物理防御数值增量 |
| \_magDef，法抗加成 | int | 法术防御数值增量 |

备注：

1. 合成装备的命名方法：一个合成装备（SynthesisEquipment）由一个攻击型装备（AttackEquipment）和一个防御型装备（DefenceEquipment）组合而成，其名称（\_name）格式如下：“**攻击型装备名-plus-防御型装备名**”，如：knife与shield合成为knife-plus-shield。此功能在构造函数中实现。
2. AttackEquipment类和DefenceEquipment继承Equipment的方式为虚继承（virtual public）。

### 英雄（Hero）类

包含文件：Hero.h(.cpp)

此类中成员较多，因此使用表格列出所有成员函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 成员函数 | 接收参数 | 返回值类型及含义 | 备注 |
| 构造函数 | | | |
| Hero | 共7个[2] | **-** |  |
| 基本操作 | | | |
| get\_name | void | string，英雄名称 |  |
| get\_health | void | int，英雄生命值 | 显示为“HP” |
| get\_phyAtk | void | int，英雄基础物理攻击力 | 基础能力随英雄而定，与装备完全无关 |
| get\_magAtk | void | int，英雄基础法术攻击力 |
| get\_phyDef | void | int，英雄基础物理防御力 |
| get\_magDef | void | int，英雄基础法术防御力 |
| get\_equipcnt | void | int，英雄已购买的装备个数 | 显示为“Equipment” |
| get\_money | void | int，英雄拥有的金钱数 | 显示为“MONEY” |
| get\_equipment | int num | Equipment\*，第num个装备的指针 | 0≤num≤2 |
| isalive | void | bool，判断英雄是否活着 | true 当且仅当HP＞0 |
| reset | 共4个 | void | 重置英雄状态 |
| 装备操作 | | | |
| buy\_equipment | Equipment\* | bool，金钱钱要足够够且现有装备数量小于3才返回true | 无论购买成功或失败，都打印反馈信息 |
| remove\_equipment[1] | int num | bool，num有效则返回true | 同上，打印反馈信息 |
| remove\_equipment[3] | Equipment\* | bool，通过指针删除装备 | 同上 |
| sell\_equipment | int num | bool，num有效则返回true | 得到装备价格的60% |
| syn\_equipment[5] | int x, int y | bool，合成成功则返回true | 合成攻击型装备和防御装备，额外花费15%金钱 |
| 战斗操作 | | | |
| get\_real\_phyAtk | void | int，总物理攻击力 | get\_real\_\* 系列成员函数获取英雄的总能力，包括基础能力和装备加成 |
| get\_real\_magAtk | void | int，总法术攻击力 |
| get\_real\_phyDef | void | int，总物理防御力 |
| get\_real\_magDef | void | int，总法术防御力 |
| get\_crit\_chance | void | int，暴击几率 |  |
| attack | Hero& | int，实际给对方造成的伤害 |  |
| crit[8] | Hero& | int，实际给对方造成的伤害 | attack时概率触发 |
| normal\_attack[7] | Hero& | int，实际给对方造成的伤害 |  |
| recover[9] | void | bool，回血成功则返回true | 见下方备注 |
| damage\_taken | int n | int，实际减少的生命值 |  |

备注：

1. 表中灰色的函数为private的，只允许被类中其他函数调用。
2. 出于数据合理性考虑，英雄初始生命值设置为100，物理、法术的攻击和防御力均为10，初始金钱数为800。
3. remove\_equipment与sell\_equipment不同。前者为private的，仅供其它成员函数调用；后者是public的，可以由玩家主动调用。remove\_equipment在程序中的作用是供sell\_equipment和syn\_equipment调用。
4. Hero类中**重载+=运算符**，myHero += knife效果等同于myhero.buy\_equipment( knife )；同理，**重载-=运算符**，myHero -= 0效果等同于myhero.sell\_equipment( 0 )。
5. syn\_equipment操作为一个英雄主动地合成已经在自己装备栏中的2个装备，目的是减少装备栏中的装备数量以供购买新装备。这2个装备必须是一个攻击型装备（AttackEquipment）和一个防御型装备（DefenceEquipment），**不能合成两个同类型的装备**。**合成装备需要额外花费两个子装备价格之和的15%**。

syn\_equipment函数是英雄的主动操作，实际运行时可由玩家决定。该成员函数的作用与main.cpp中不同，后者是直接构建一个既有攻击特性又有防御特性的“综合“型装备。

合成装备不允许再进行合成。

1. 英雄使用vector容器储存装备（Equipment\*），因此不需要“装备数量“这一成员变量，因为vector的增加、删除操作会自动改变其size()的大小。英雄类中仍保留get\_equipcnt函数方便使用，此函数直接返回vector的size()。
2. **伤害计算方法**：

C\_IMMUNE为常数，默认为24.0。

1. 每次attack对方的时候，有一定几率触发暴击，暴击可以造成1.5倍伤害。暴击几率的计算公式如下：暴击几率 = 剩余金钱数 / 2000，最多不超过100%。
2. 回血。英雄可以主动选择回血而不攻击，回血花费200金钱，恢复15生命值，生命值最多不超过100。如果生命值已经为100，那么除了花钱并浪费一回合，无任何效果。

### 战场（Battleground）类

包含文件：Battleground.h(.cpp)。

|  |  |  |
| --- | --- | --- |
| 数据成员 | 描述 | 备注 |
| Hero \_player | 玩家英雄 | 由玩家输入确定 |
| Hero \_enemy | 敌人英雄 | 系统随机指定 |
| int \_game | 游戏局数 |  |
| int \_round | 该局中的回合数 |  |
| int \_score | 玩家总得分 |  |
| vector<Equipment\*> \_inventory | 装备库存（所有可购买的） |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 成员函数 | 参数 | 返回值类型及含义 | 备注 |
| game\_loop | void | void，游戏开始后的多局循环 | 每局结束后询问是否再开一盘 |
| round\_loop | void | bool，true代表敌人死亡，false代表玩家死亡 | 一方英雄死亡，即一局结束 |
| get\_game | void | int，获取局数 |  |
| get\_round | void | int，获取局内的当前回合数 |
| get\_score | void | int，获取总得分 |
| get\_inventory | int num | Equipment\*，获取第num个装备 |
| print\_inventory | void | void，显示装备库存 |  |

### 简易UI（非类）

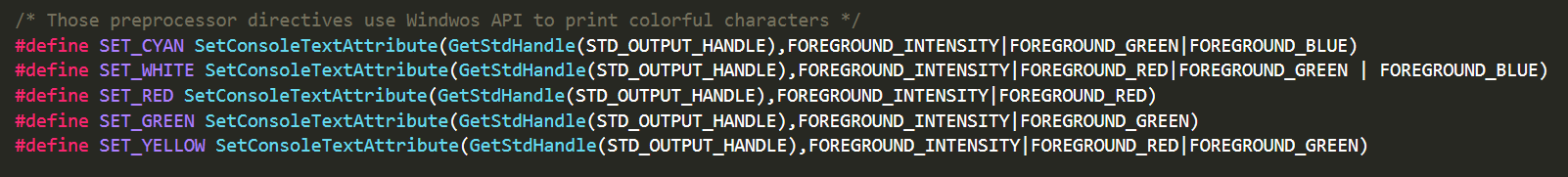
包含文件：ui.h(.cpp)。

|  |  |  |
| --- | --- | --- |
| 函数 | 描述 | 备注 |
| split\_line | 分割线 | 100个等号 |
| print\_introduction | 刚进入游戏的介绍画面 | 字面意思，见注释 |
| print\_help | 游戏说明 |
| print\_bye | 结束画面 |

### 常量（非类）

文件：macro.h。

定义了各种常量，宏名均为C\_开头。详见栏目（四）。除此之外，还有若干以SET\_开头的宏函数，如SET\_CYAN、SET\_WHITE等，是用来控制台显示颜色的。



## 本文档涉及到的常量及数据计算方法

本板块的内容大都在上文都出现过，此处作为汇总方便查阅。下文中C开头的变量为可变常量，均以宏的形式定义在macro.h中。

### 伤害计算

C\_IMMUNE为免伤比常数，为**24.0**.。

### 暴击率计算

C\_CRIT\_MONEY为暴击率上限金钱数，为**2000**。如果英雄金钱数≥2000，那么暴击率为100%。

### 暴击加成计算

C\_CRIT为暴击加成倍率，为**1.5**。

### 攻击敌人获得金钱计算

C\_RECEIVE为收入系数，为**5**。如造成对方10伤害，则自己收入50金钱。

### 合成装备

### 卖出装备收入计算

C\_SELL为卖出装备价格系数，为**0.6**。如果装备是合成装备，则有：

不包含合成装备额外花费的那15%的金钱。

### 得分计算

1. 每次造成敌人伤害，**造成了多少伤害就加多少分**；
2. 每回合得分**加100**；
3. 游戏结束时，得分加**200×局数**。

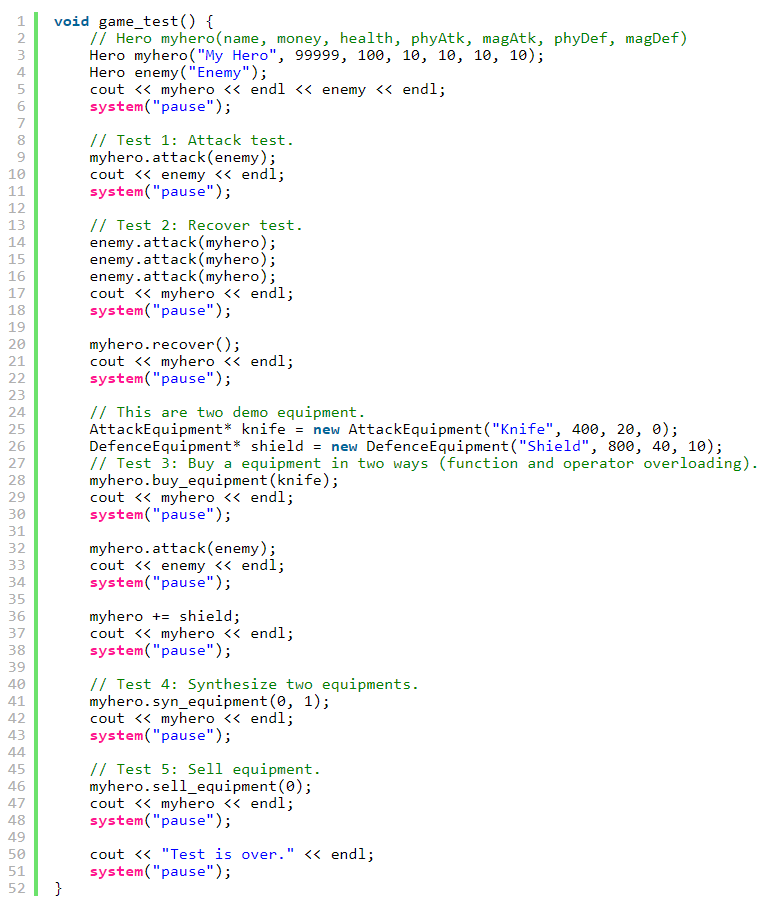
## 附录

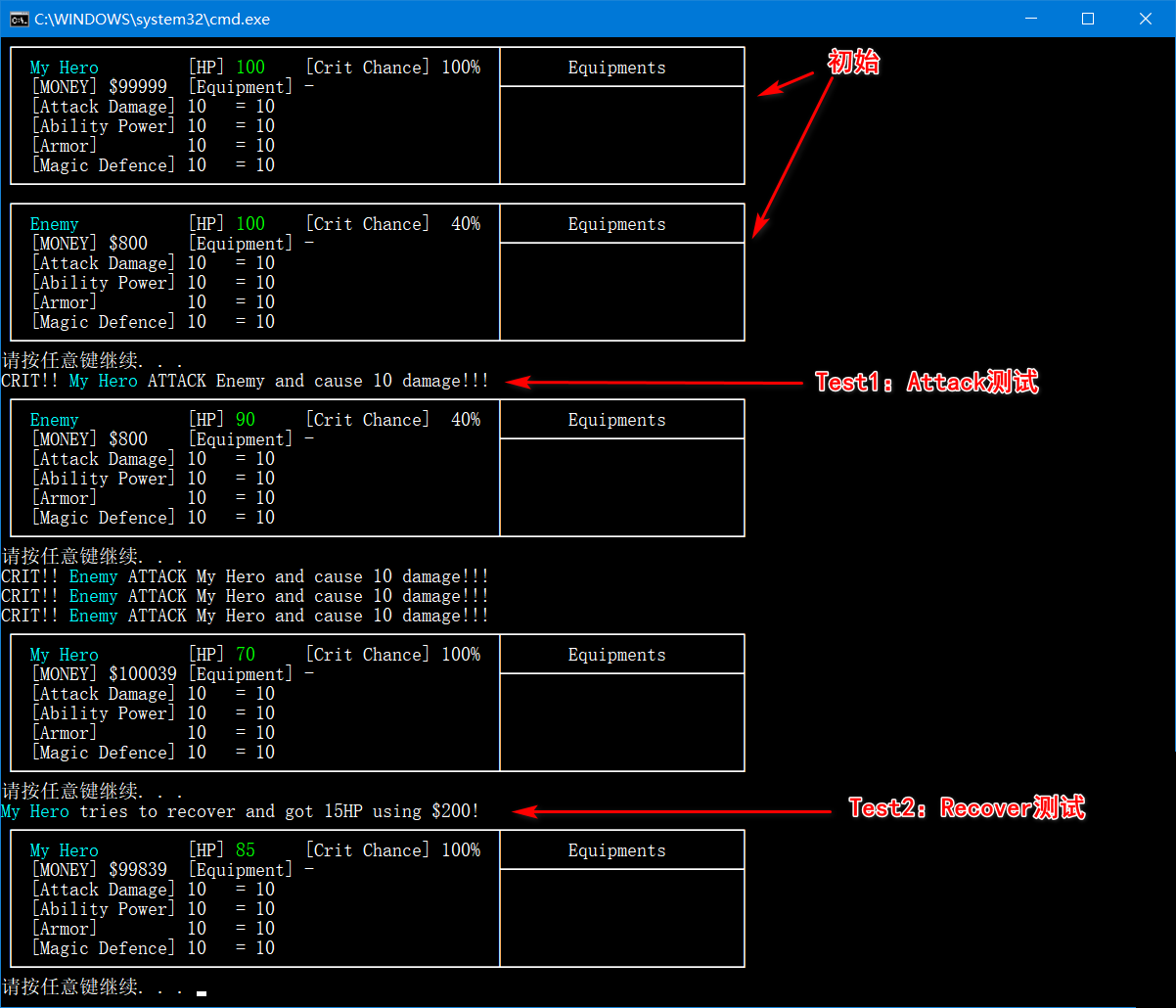
### 测试代码

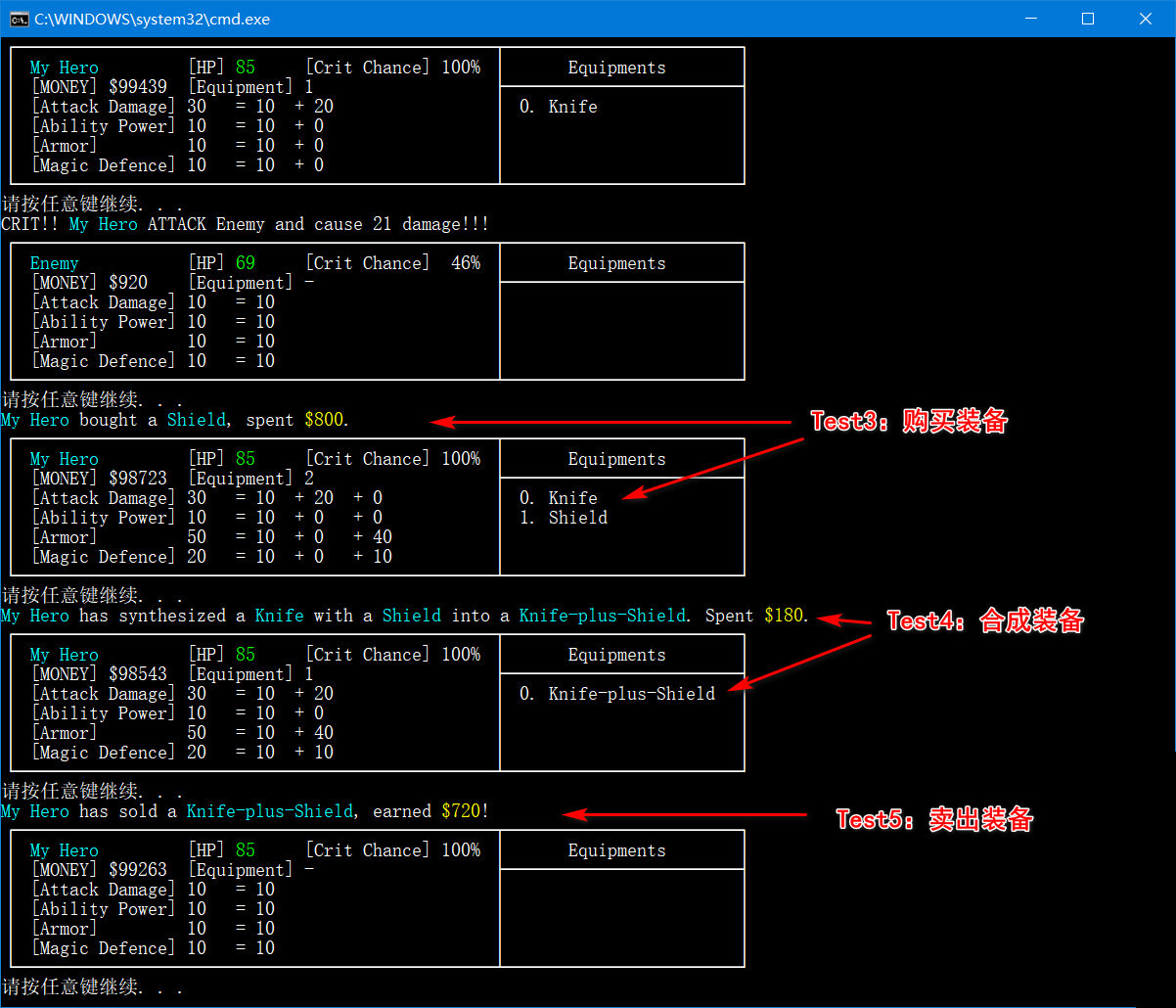
**为了方便调试，专门设计测试模式。**

game\_test函数用于手动测试代码，该函数位于main.cpp中。当通过-t参数运行程序时将进入测试模式，且不会显示任何UI界面。注意：game\_test与game\_main函数完全并列，在main函数中位于同一个if-else中。

在项目根目录下有testmode.bat，可以快速进入测试模式。



测试模式运行结果：



第一屏

第二屏

### 预置的装备库存

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类别 | 装备名称 | 价格 | Attack Damage | Ability Power | Armor | Magic Defence |
| Attack Equipment  攻击型装备 | Knife | 400 | 20 | 0 | - | - |
| Sword | 650 | 45 | 25 | - | - |
| Poison | 1000 | 30 | 50 | - | - |
| AK47 | 1400 | 90 | 10 |  |  |
| Defence Equipment  防御型装备 | Vest | 500 | - | - | 30 | 2 |
| Shield | 800 | - | - | 40 | 10 |
| Immunity | 1200 | - | - | 50 | 50 |
| Magnetic\_field | 1500 | - | - | 80 | 20 |
| Synthesis Equipment  综合型装备 | 尚未设计 |  |  |  |  |  |

后期版本可设计更多装备。

### 游戏帮助（print\_help）

In this game you will play a hero. You will participate in multiple games if you can keep alive. In each game, you will encounter an enemy hero, and you will conduct multiple rounds of fighting. At the beginning, both heroes have 100 HP. In each round, you and the enemy can each perform one operation. These operations include "attack", "recover", "buy equipment," "sell equipment," etc.

In this battle, if you defeat the enemy, then the game ends. You can choose to start a new game again and accumulate points. Of course you can also exit the game. However, if your hero dies, the game ends immediately and you will see your final score.

Enjoy it!

### 更新日志

**v1.0.3** (2018.06.09)

修复了Hero和Battleground中析构函数导致内存错误的问题。

**v1.0.2** (2018.06.07)

重新封装了SynthesisEquipment类，改用set方法。

**v1.0.1** (2018.06.03)

修复了购买装备时不合法输入导致程序崩溃的问题。

已知Bug：Sleep函数和\_getch函数导致的流缓冲问题。

描述：程序在Sleep的时候，按键操作仍被记录，程序恢复执行时这些输入直接被\_getch()读取，导致多于操作。例如，每次轮到玩家操作时，如果快速连续按很多次A，那么后续几个回合将自动进行。