**实验报告三**

**程序设计思路：**实现一个技能的基本功能，我这里选择的是诸葛亮的技能。我选择二技能突进，每次突进10码的距离，可以突击3次。因此我需要写一个函数当英雄位置信息传递给函数时，英雄前进10码，并且记录次数，累计三次后进入冷却。我需要定义一个全局变量用来记录次数，函数将位置信息加10。在程序执行完后调用析构函数清除。

**程序代码：**

#include<iostream>

using namespace std;

class Hero

{

public:

Hero();

Hero(int num,int hurt,string s,int hero\_blood,

float hero\_orign\_x,float hero\_orign\_y,int x1,int x2,int x3);

Hero(const Hero & name);

~Hero();

void changename(string myname);

int use\_skill1(int num);

Hero &operator=(const Hero &rhs);

void showskill();

int use\_skill2(float hero\_orign\_x);

private:

int skill1;

int skill2;

int skill3;

int orign\_x;

int hurt;

int num;

string hero\_name;

};

头文件

#include<iostream>

#include"Hero.h"

using namespace std;

string sunwukong;

int skill;

int num=2;

int aim1=5000;

int aim2=3000;

int dis = 30;

int main()

{

int times = 3;

Hero Zhugeliang;

Zhugeliang.showskill();

cout << "回合开始" << endl;

cout << "1：技能一 2：技能二 3：技能三" << endl;

cout << "当前距离目标30" << endl;

cin >>skill;

if (skill == 1)

{

aim1=aim1-Zhugeliang.use\_skill1(num);

aim2 = aim2- Zhugeliang.use\_skill1(num);

cout << "目标一生命值" << aim1;

cout << "目标二生命值" << aim2 << endl;

}

if (skill == 2)

{

dis = Zhugeliang.use\_skill2(dis);

times--;

cout << "当前距离目标:" <<dis;

cout << " 剩余次数:" <<times<<endl;

}

while (1)

{

cin >> skill;

if (skill == 2)

{

dis = Zhugeliang.use\_skill2(dis);

times--;

cout << "当前距离目标:" << dis;

cout << " 剩余次数:" <<times<<endl;

}

if (times <= 0)

{

cout << "技能进入冷却" << endl;

break;

}

}

while (1)

{

cout << "按1释放技能:东风破" << endl;

cin >> skill;

aim1 = aim1 - Zhugeliang.use\_skill1(num);

aim2 = aim2 - Zhugeliang.use\_skill1(num);

cout << "目标一生命值" << aim1;

cout << " 目标二生命值" << aim2 << endl;

if (aim2 <= 0)

{

cout << "你击杀了一名敌人" << endl;

break;

}

}

while(1)

{

cout << "按1继续释放技能" << endl;

cin >> skill;

aim1 = aim1 - Zhugeliang.use\_skill1(1);

cout << " 目标二生命值" << aim1 << endl;

if (aim1 <= 0)

{

cout << "来犯之敌已尽数诛灭！" << endl;

break;

}

}

return 0;

}

主函数

#include"Hero.h"

Hero::Hero()

{

cout <<"this is a defult consturctor"<<endl;

cout << "英雄：诸葛亮" << endl;

cout << "技能一：Dongfengpo" << endl;

cout << "技能二：Tujin" << endl;

cout << "技能三：Yuanqidan" << endl;

skill1=1000;

skill2=500;

skill3=2000;

}

Hero::Hero(int num,int hurt,string s,int hero\_blood,

float hero\_orign\_x,float hero\_orign\_y,int x1,int x2,int x3)

{

cout <<"this is a overload consturctor"<<endl;

skill1=x1;

skill2=x2;

skill3=x3;

}

Hero::Hero(const Hero & name)

{

cout <<"this is a copy consturctor"<<endl;

skill1=name.skill1;

skill2=name.skill2;

skill3=name.skill3;

}

Hero ::~Hero()

{

cout <<"this is a desturctor"<<endl;

}

Hero & Hero::operator=(const Hero & rhs)

{

if (this==&rhs)

{

return \*this;

}

this->skill1=rhs.skill1;

this->skill2=rhs.skill2;

this->skill3=rhs.skill3;

return \*this;

}

void Hero::showskill()

{

cout <<"技能一"<<skill1<<endl;

cout <<"技能二"<<skill2<<endl;

cout <<"技能三"<<skill3<<endl;

}

int Hero::use\_skill1(int num)

{

if (num == 2)

{

return 500;

}

if (num == 1)

{

return 1000;

}

}

int Hero::use\_skill2(float hero\_orign\_x)

{

return hero\_orign\_x - 10;

}

**产生的错误：**在编写程序的时候出现了各种各样的错误，但很多都是低级错误诸如未加“;”或者变量未定义。这些错误没有什么价值，多调试几次就能把错误找出来。在头文件.h的声明中一定要和.cpp形式一致，不然会报错。在写.cpp文件中写子函数时也一定要在.h文件中预先定义。

**解决办法：**在程序中设置断点，逐个排查，看看程序到底哪一步出了问题。之前用来记录技能次数的变量是局部变量，不能达到我想要的结果，用全局变量就OK了。

**思想感悟：**这种开放性的作业确实难对付，抄都没地方抄，不过确实也能激发同学们的兴趣，我都有种把王者荣耀用回合制的卡牌游戏表现出来的冲动了。在一开始写程序的时候确实手忙脚乱，有种无从下手的感觉，但思绪理清楚之后也没这么难了。.h文件中的声明与.cpp中的定义形式一一对应，将自己想要完成的效果在.cpp中用函数的形式复现。当程序出现错误时，可以在程序中增添断点，单步执行查找错误。

**兴趣创新：**将王者荣耀的英雄诸葛亮改成横冲直撞式，我在里面还写了一个东风破的程序，当命中单个敌人时造成1000伤害，当命中两个敌人时，对两个敌人造成500伤害。