**C++第四次实验报告**

1. **设计程序的思路**

本次实验的目的是要实现一个技能的两个攻击目标分摊伤害的效果。该程序用到了重载函数、引用方面的知识。重载函数只要函数参数的类型不同，或者参数的个数不同，或者二者兼而有之，两个或两个以上的函数就可以使用相同的函数名；在该程序中使用对象引用作为函数参数不但具有对象指针用作函数参数的优点，而且用对象引用作函数参数将更简单、更直接，如：

#include <iostream>

using namespace std;

class gy

{

public:

gy();

gy( int xl,int mf,int gj,int sd,int gs,int fq,int mk,int wk,int hl,int hx);

void showd();

void beidongjineng();

void at();

void lunhui1();

void lunhui2();

void fangyu();

int d1;

int d2;

int d3;

int d4;

int d5;

int d6;

int d7;

int d8;

int d9;

int d10;

};

1. **程序的主体代码**

**gy.cpp**

#include <iostream>

#include "gy.h"

using namespace std;

gy::gy( int hp,int mp,int gj,int sd,int gs,int fq,int mk,int wk,int hmp,int hhp)//英雄属性

{

P1=hp;//HP

P2=mp;//MP

P3=gj;//攻击

P4=sd;//速度

P5=gs;//攻速

P6=fq;//法强

P7=mk;//魔抗

P8=wk;//物抗

P9=hmp;//MP回复

P10=hhp;//HP回复

}

void gy::showd()

{

cout<<"HP:";cout<<P1 <<endl;

cout<<"MP:";cout<< P2 <<endl;

cout<<"攻击:";cout<< P3<<endl;

cout<<"速度:";cout<< P4<<endl;

cout<<"攻速:";cout<< P5 <<endl;

cout<<"法强:";cout<< P6<<endl;

cout<<"魔抗:";cout<< P7<<endl;

cout<<"物抗:";cout<< P8<<endl;

cout<<"恢复MP:";cout<< P9 <<endl;

cout<<"恢复HP:";cout<< P10 <<endl;

}

void gy::jineng()

{

P5 =P5+10;

P8 =P8+10;

P2 =P2-10;

}

void gy::at()

{

P1= P1-50;

}

void gy::lunhui1 ()

{

P2=P2-100;

}

void gy::lunhui2 ()

{

P1=P1-100;

}

void gy::fangyu ()

{

P7=P7\*2;

P8=P8\*2;

}

**gy.h**

#include <iostream>

using namespace std;

class gy

{

public:

gy();

gy( int hp,int mp,int gj,int sd,int gs,int fq,int mk,int wk,int hmp,int hhp);

void showd();

void beidongjineng();

void at();

void lunhui1();

void lunhui2();

void fangyu();

int P1;

int P2;

int P3;

int P4;

int P5;

int P6;

int P7;

int P8;

int P9;

int P10;

};

void solo();

**源.cpp**

#include <iostream>

#include "gy.h"

using namespace std;

int main()

{cout<<"欢迎来到王者荣耀"<<endl;

cout<<"开始游戏，您的英雄为 “轮回之风 ”\nid为幻影之雪。"<<endl;

int xz;

cout<<"1.调查看英雄属性："<<endl;

cout<<"2.查看技能轮回之风的介绍："<<endl;

cout<<"3.攻击敌人项羽："<<endl;

cout<<"4.攻击敌人墨子和项羽："<<endl;

cout<<"5.查看自身状态："<<endl;

cout<<"6.退出游戏："<<endl;

cin>>xz;

gy xiangyu(100,100,0,0,0,0,0,0,0,0);//初始化敌人项羽

gy mozi(100,100,0,0,0,0,0,0,0,0);//初始化敌人墨子

gy guanyu(100,100,100,100,100,100,100,100,1,1);//英雄关羽初始化

while(xz!=6)

{

if(xz==5)

{ guanyu.showd ();}

else

switch (xz)

{

case 1:cout <<"查看英雄属性"<<endl;break ;

case 2:cout <<"轮回之风：专属技能，发出一道龙卷风，\n对敌人造成200%攻击 ，\n该技能会消耗1/2的蓝量"<<endl;break ;

case 3: {

int xz2;

cout<<"1.震慑之吼："<<endl;

cout<<"2.狂暴双A："<<endl;

cout<<"3.轮回之风："<<endl;

cout<<"4.惩戒之雷："<<endl;

cin>>xz2 ;

switch (xz2)

{

case 1: guanyu.beidongjineng() ; break ;

case 2: {xiangyu .at ();if(xiangyu.d1 ==0) {cout <<"项羽被击杀"<<endl;};} ; break ;

case 3: {guanyu.lunhui1 ();xiangyu.lunhui2(); if(xiangyu.d1 ==0){cout <<"项羽被击杀"<<endl;};} ; break ;//lunhui12分别减少项羽hp和关羽mp

case 4:guanyu.fangyu ();break ; }};break ;

case 4:{int xz3;

cout <<"1.直接轮回之风"<<endl;

cin>>xz3;

switch (xz3 )

{

case 1:{cout<<"你被打飞了，哇咔咔咔咔！！！"<<endl;guanyu .lunhui1 ();xiangyu.at ();mozi .at (); if(xiangyu.d1 ==0){cout <<"项羽嗝儿屁了"<<endl;}if(mozi .d1 ==0){cout <<"墨子被干掉了"<<endl;} } ; break;

}

};break ;

}

cin>>xz;

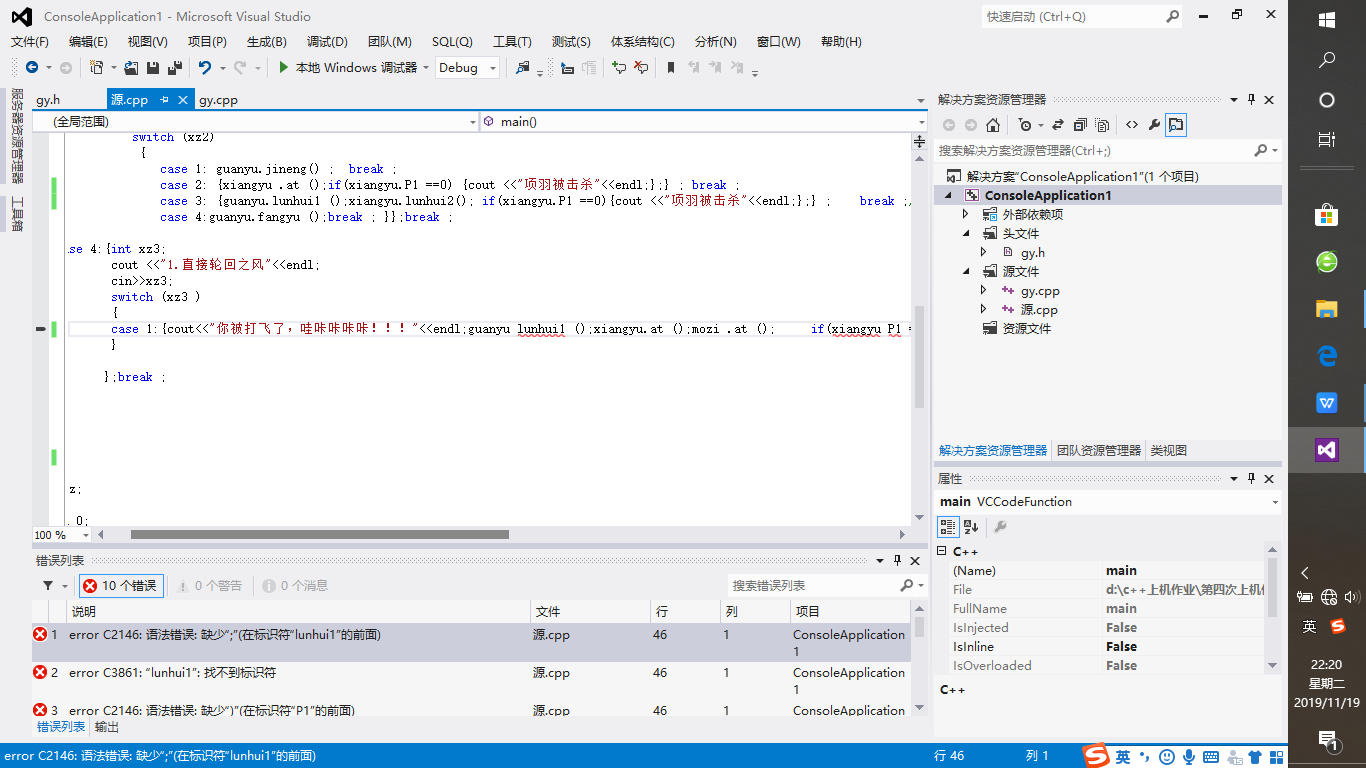
}

return 0;

}

因 **gy.h** 中的定义条件太繁琐，以本人的水平分开定义很容易出错，所以将它们都定义在public（公有权限）中，如此比较方便编程，也不容易出错。

1. **程序执行时遇到的错误**



乍一看10个错觉得很慌，但双击第一个错仔细查看光标处的语句发现其实也就guanyu lunhui1 和 xiangyu P1 处在 guanyu 和 xiangyu 后面少了个符号 . 改为guanyu . lunhui1 和 xiangyu . P1 与上面的函数语句

switch (xz2)

{

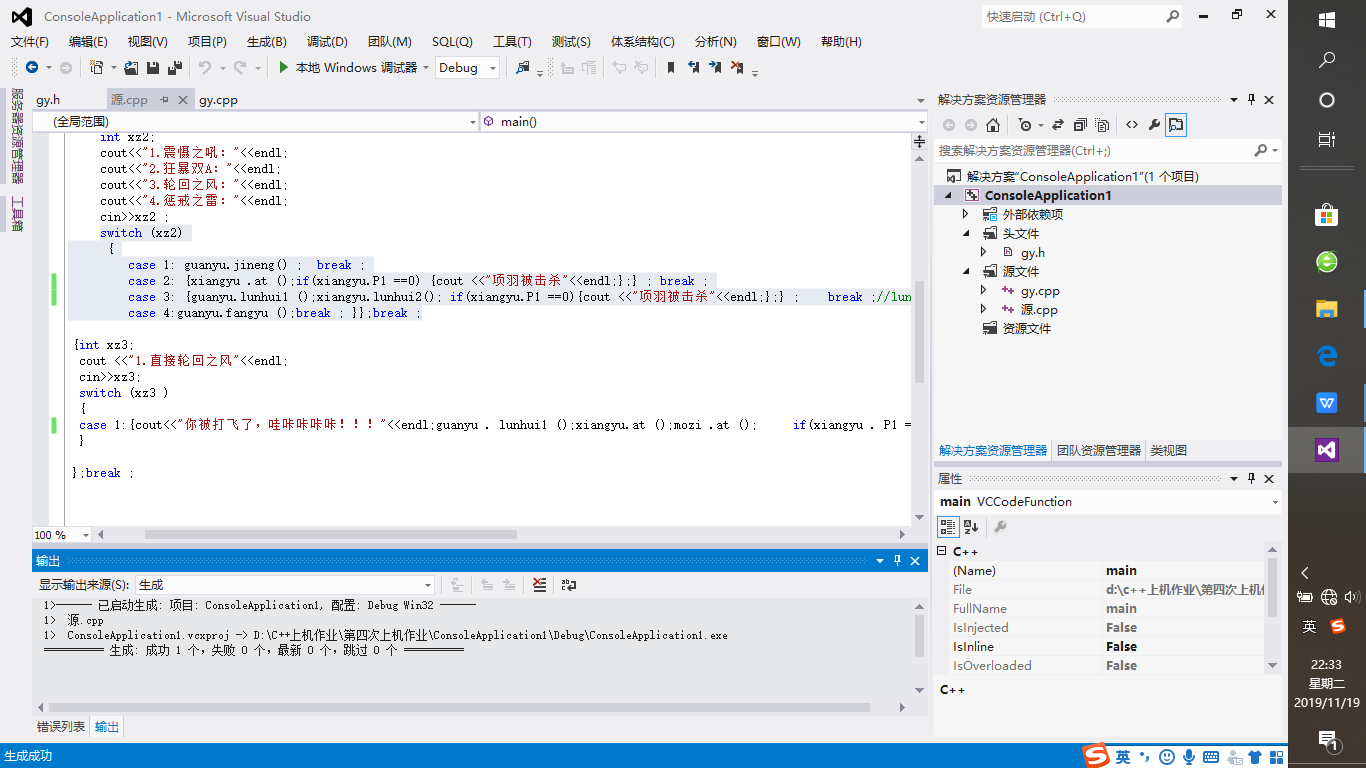
case 1: guanyu.beidongjineng() ; break ;

case 2: {xiangyu .at ();if(xiangyu.P1 ==0) {cout <<"项羽被击杀"<<endl;};} ; break ;

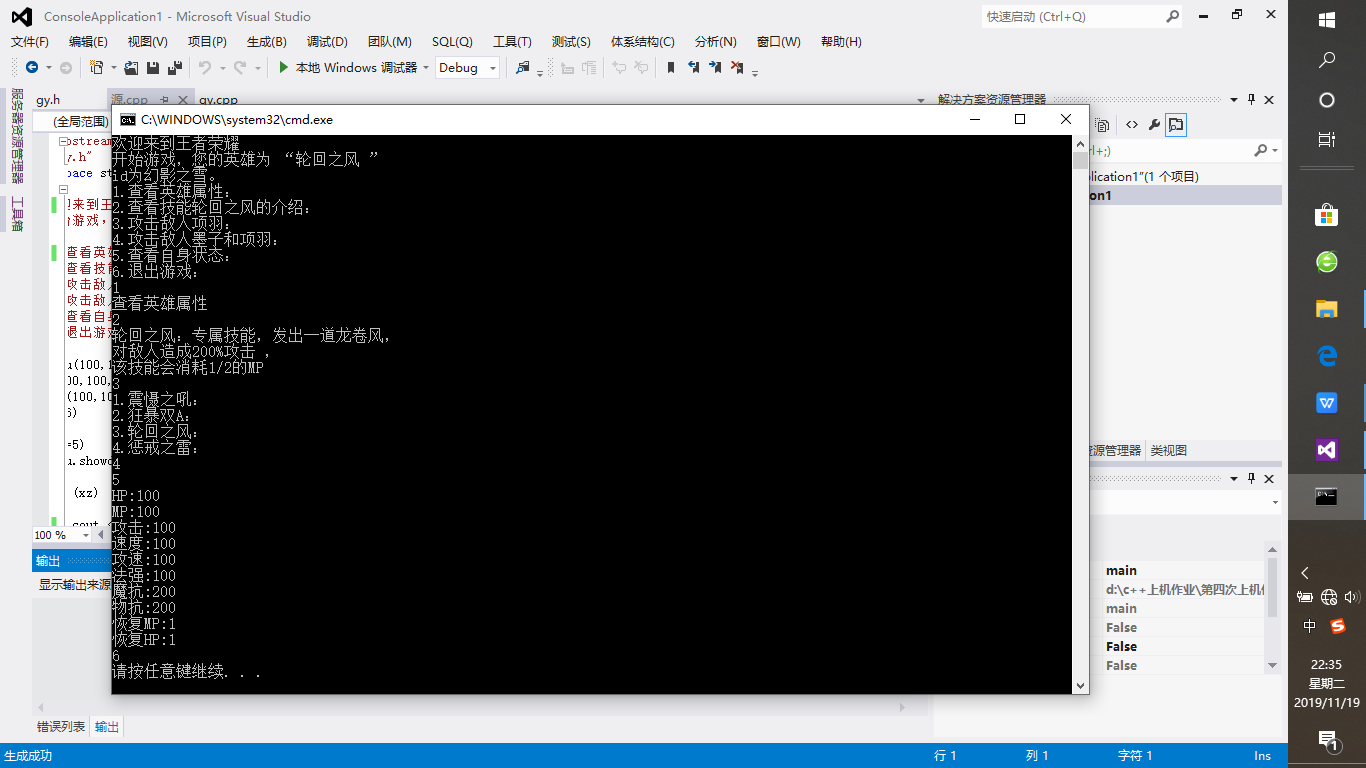
case 3: {guanyu.lunhui1 ();xiangyu.lunhui2(); if(xiangyu.P1 ==0){cout <<"项羽被击杀"<<endl;};} ; break ;//lunhui12分别减少项羽hp和关羽mp

case 4:guanyu.fangyu ();break ; }};break ;

中的格式对应，程序正常运行。



程序正常运行后，输出结果如图所示：



根据图中结果所示，该程序实现一个技能的两个攻击目标分摊伤害的效果，本次实验是成功的。

1. **总结**

**经过为期12周的C++上机实验课的学习和黄老师的教导，本人初步了解了如何用VS2012软件通过编写程序实现想要的结果，C++语言的学习是计算机程序设计的基础和重要组成部分，在黄老师的带领下我初步领略了计算机程序设计中编写程序的艰难和回头查找就、解决程序语句中错误时的痛苦（好多错误是真的找不到，总在一些很不起眼的地方、老是在一些很令人无语的地方出错）。平时上着阿元老师的理论课时觉得这些东西应该很简单，真正自己上机操作之后才发现根本不是那么回事，平时的理论和手上的实际操作对不起来。综合来说要想学好C++不仅仅需要学好课本的知识，更关键的是要多操作，有些东西只有操作后才明白（并不是说课本知识不重要，二者缺一不可）。**