**Main**

#include <iostream>

#include <string>

#include <vector>

#include <fstream>

#include <sstream>

//--------------------------

#include "Map.h"

#include "RpgCreature.h"

//--------------------------

#include "RpgPerson.h"

#include "RpgPerson\_SwordMan.h"

#include "RpgPerson\_Thief.h"

#include "RpgPerson\_Magician.h"

//--------------------------

#include "Monster.h"

#include "Monster\_Slime.h"

#include "Monster\_Troll.h"

#include "Monster\_Dragon.h"

//-------------------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-------------------------

using namespace std;

//////////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////

class SaveData{

public:

SaveData(string name,int level,string job,int currentHp,int maxHp,

int strength,int defence,int cirts,int coin,int experience,string mapName,int x,int y)

{

this->name=name;

this->level=level;

this->job=job;

this->currentHp=currentHp;

this->maxHp=maxHp;

this->strength=strength;

this->defence=defence;

this->cirts=cirts;

this->coin=coin;

this->experience=experience;

this->mapName=mapName;

this->x=x;

this->y=y;

}

~SaveData(){}

string getName()

{

return name;

}

int getLevel()

{

return level;

}

string getJob()

{

return job;

}

int getCurrentHp()

{

return currentHp;

}

int getMaxHp()

{

return maxHp;

}

int getStrength()

{

return strength;

}

int getDefence()

{

return defence;

}

int getCirts()

{

return cirts;

}

int getCoin()

{

return coin;

}

int getExperience()

{

return experience;

}

string getMapName()

{

return mapName;

}

int getX()

{

return x;

}

int getY()

{

return y;

}

///////////////////////////////////////////

void setName(string name)

{

this->name=name;

}

void setLevel(int level)

{

this->level=level;

}

void setJob(string job)

{

this->job=job;

}

void setCurrentHp(int currentHp)

{

this->currentHp=currentHp;

}

void setMaxHp(int maxHp)

{

this->maxHp=maxHp;

}

void setStrength(int strength)

{

this->strength=strength;

}

void setDefence(int defence)

{

this->defence=defence;

}

void setCirts(int cirts)

{

this->cirts=cirts;

}

void setCoin(int coin)

{

this->coin=coin;

}

void setExperience(int experience)

{

this->experience=experience;

}

void setmapName(string mapName)

{

this->mapName=mapName;

}

void setX(int x)

{

this->x=x;

}

void setY(int y)

{

this->y=y;

}

private:

string name,job,mapName;

int level,currentHp,maxHp,strength,defence,cirts,coin,experience,x,y;

};

//////////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////

vector<string> mapData;

//void meetMonster(RpgPerson &player);

void meetMonster(RpgPerson &player, vector<Map> game\_Map, int j);

int main()

{

cout<<"-----------------------------------------"<<endl;

cout<<"| Welcome to oop assignment4 rpg. |"<<endl;

cout<<"------------------------------------------"<<endl;

cout<<"| 1. New game |"<<endl;

cout<<"| 2. Load game |"<<endl;

cout<<"| 3. Exit |"<<endl;

cout<<"------------------------------------------"<<endl;

cout<<"Enter your choice:";

RpgPerson \*player\_role; //Upcasting //Polymorphism

//////////////////////////////////////

/\*--------------------

Map

--------------------\*/

int j = 0;// mapNumber

vector<Map> game\_Map; //map

string line;

string name,nextMapName;

int initialPositionX,initialPositionY,width,height,order=0;

vector<string> mapData; //string可直接用string[][]

vector<string> monsterName;

ifstream mapFile ("map.txt");

while(mapFile>>name>>nextMapName>>initialPositionX>>initialPositionY>>width>>height)

{

string line;

for(int i=0;i<height;i++)

{

mapFile >> line;

mapData.push\_back(line);

}

while(1)

{

mapFile >> line;

if(line=="---")

{

break;

}

else

{

monsterName.push\_back(line);

}

}

game\_Map.push\_back(Map(name,nextMapName,initialPositionX,initialPositionY,width,height,mapData,monsterName));

monsterName.clear();

mapData.clear();

}

/////////////////////////////////////

/////////////////////////////////////

////////////////////////////////////

int mode;

cin>>mode;

switch(mode)

{

case 1:

{

//////////////////////////////////////

/\*--------------------

New player

--------------------\*/

//////////////////////////////////////

//RpgPerson \*player\_role; //Upcasting //Polymorphism

cout<<"Enter your name:";

string player\_name;

cin>>player\_name;

cout<<"Enter your Job (1-3) 1.SwordMan 2.Thief 3.Magician:";

int character; //choose character寫在RPGPerson class裡

cin>>character;

cout<<endl;

switch(character)

{

case 1:

{

player\_role = new RpgPerson\_SwordMan(player\_name,1,"SwordMan",100,100,7,2,20,0,0,

game\_Map[0].getinitialPositionX(),game\_Map[0].getinitialPositionY());

}

break;

case 2:

{

player\_role = new RpgPerson\_Thief(player\_name,1,"Thief",100,100,7,2,20,0,0,

game\_Map[0].getinitialPositionX(),game\_Map[0].getinitialPositionY());

}

break;

case 3:

{

player\_role = new RpgPerson\_Magician(player\_name,1,"Magician",100,100,7,2,20,0,0,

game\_Map[0].getinitialPositionX(),game\_Map[0].getinitialPositionY());

}

break;

default:

{

cout<<"Wrong Input"<<endl;

}

}

j=0;

}

break;

case 2:

{

///////////////////////////////////////

//---------Open importSaveFile/------//

//////////////////////////////////////

ifstream importSaveFile("save.txt");

string name,job,mapName;

int level,currentHp,maxHp,strength,defence,cirts,coin,experience,x,y;

///////////////////////////////////////

//////////Reand and Print Data/////////

///////////////////////////////////////

vector<SaveData> saveDatas;

while(importSaveFile>>name>>level>>job>>currentHp>>maxHp>>strength>>defence>>cirts>>coin>>experience>>mapName>>x>>y)

{

saveDatas.push\_back(SaveData(name,level,job,currentHp,maxHp,strength,defence,cirts,

coin,experience,mapName,x,y));

}

cout<<"You have "<<saveDatas.size()<<" records."<<endl;

for(int i=0;i<saveDatas.size();i++)

{

cout<<"---------------record "<<i<<" -----------------"<<endl;

cout<<"Name: "<<saveDatas[i].getName()<<endl;

cout<<"Job: "<<saveDatas[i].getJob()<<endl;

cout<<"Level: "<<saveDatas[i].getLevel()<<endl;

cout<<"-----------------------------------------------"<<endl;

}

cout<<"Load record (Enter 0, 1, 2, .....)"<<endl;

///////////////////////////////////////

////////////chooseDataNumber///////////

///////////////////////////////////////

while(1)

{

int chooseDataNumber;

cin>>chooseDataNumber;

if(chooseDataNumber<=saveDatas.size()){

///////////////////////////////////////

///////////////////Job/////////////////

///////////////////////////////////////

//need to be cout!

if(saveDatas[chooseDataNumber].getJob()=="SwordMan")

{

player\_role = new RpgPerson\_SwordMan(saveDatas[chooseDataNumber].getName(),

saveDatas[chooseDataNumber].getLevel(),

saveDatas[chooseDataNumber].getJob(),

saveDatas[chooseDataNumber].getCurrentHp(),

saveDatas[chooseDataNumber].getMaxHp(),

saveDatas[chooseDataNumber].getStrength(),

saveDatas[chooseDataNumber].getDefence(),

saveDatas[chooseDataNumber].getCirts(),

saveDatas[chooseDataNumber].getCoin(),

saveDatas[chooseDataNumber].getExperience(),

saveDatas[chooseDataNumber].getX(),

saveDatas[chooseDataNumber].getY());

}

else if(saveDatas[chooseDataNumber].getJob()=="Thief")

{

player\_role = new RpgPerson\_Thief(saveDatas[chooseDataNumber].getName(),

saveDatas[chooseDataNumber].getLevel(),

saveDatas[chooseDataNumber].getJob(),

saveDatas[chooseDataNumber].getCurrentHp(),

saveDatas[chooseDataNumber].getMaxHp(),

saveDatas[chooseDataNumber].getStrength(),

saveDatas[chooseDataNumber].getDefence(),

saveDatas[chooseDataNumber].getCirts(),

saveDatas[chooseDataNumber].getCoin(),

saveDatas[chooseDataNumber].getExperience(),

saveDatas[chooseDataNumber].getX(),

saveDatas[chooseDataNumber].getY());

}

else if(saveDatas[chooseDataNumber].getJob()=="Magician")

{

player\_role = new RpgPerson\_Magician(saveDatas[chooseDataNumber].getName(),

saveDatas[chooseDataNumber].getLevel(),

saveDatas[chooseDataNumber].getJob(),

saveDatas[chooseDataNumber].getCurrentHp(),

saveDatas[chooseDataNumber].getMaxHp(),

saveDatas[chooseDataNumber].getStrength(),

saveDatas[chooseDataNumber].getDefence(),

saveDatas[chooseDataNumber].getCirts(),

saveDatas[chooseDataNumber].getCoin(),

saveDatas[chooseDataNumber].getExperience(),

saveDatas[chooseDataNumber].getX(),

saveDatas[chooseDataNumber].getY());

}

else{}

///////////////////////////////////////

///////////////Exist Map///////////////

///////////////////////////////////////

if(saveDatas[chooseDataNumber].getMapName()=="Omen")

{

j=0;

}

else if(saveDatas[chooseDataNumber].getMapName()=="Tinda")

{

j=1;

}

else if(saveDatas[chooseDataNumber].getMapName()=="Fongo")

{

j=2;

}

else{}

break;

}

else{

cout<<"You don't have this data"<<endl;

}

}

saveDatas.clear();

}

break;

case 3:

{

exit(0);

}

break;

default:

{

cout<<"Wrong Input"<<endl;

exit(0);

}

}

RpgPerson &player = \*player\_role; //RpgPerson player = \*player\_role;

game\_Map[j].addPlayer(player); //\*player 是傳值進去??

game\_Map[j].printMap();

/////////////////////////////////////////////////////////////////////////////////////

//-------------------------------------------Move----------------------------------//

/////////////////////////////////////////////////////////////////////////////////////

//int j = 0;

while(1)

{

cout<<endl;

cout<<"\*\*\*\*\*"<<endl;

cout<<"Use w/s/a/d to move up/down/left/right."<<endl;

cout<<"Enter i to show your status."<<endl;

cout<<"Enter h to recover your hp (need 5 coins)."<<endl;

cout<<"Enter save to save game."<<endl;

cout<<"Enter exit to leave game."<<endl;

cout<<"\*\*\*\*\*"<<endl;

string input;

cin>>input;

if(input=="w")

{

if(game\_Map[j].canMove(player.getX(),player.getY()-1)==true && (game\_Map[j].getMapData(player.getX(),player.getY()-1)!='@'))

{

game\_Map[j].removePlayer(player);//刪#

player.moveUp();

game\_Map[j].addPlayer(player);//#

//meetMonster(player);

meetMonster(player,game\_Map,j);

game\_Map[j].printMap();

}

else if((game\_Map[j].canMove(player.getX(),player.getY()-1)==true) && (game\_Map[j].getMapData(player.getX(),player.getY()-1)=='@'))

{

game\_Map[j].removePlayer(player);//刪#

if(game\_Map[j].getNextMapName()=="Fongo"){

j=2;

}

else if(game\_Map[j].getNextMapName()=="Omen"){

j=0;

}

else if(game\_Map[j].getNextMapName()=="Tinda"){

j=1;

}

else{}

player.setPosition(game\_Map[j].getinitialPositionX(),game\_Map[j].getinitialPositionY());

game\_Map[j].addPlayer(player);

game\_Map[j].printMap(); //wrong!

}

else

{

cout<<"Can't move up, try another direction."<<endl;

cout<<endl;

game\_Map[j].printMap();

}

}

////////////////////////////////////////////////////////////////////////////////

else if(input=="s")

{

if(game\_Map[j].canMove(player.getX(), player.getY()+1)==true && (game\_Map[j].getMapData(player.getX(),player.getY()+1)!='@'))

{

game\_Map[j].removePlayer(player);//刪#

player.moveDown();

game\_Map[j].addPlayer(player);//加#

//meetMonster(player);

meetMonster(player,game\_Map,j);

game\_Map[j].printMap();

}

else if((game\_Map[j].canMove(player.getX(),player.getY()+1)==true) && (game\_Map[j].getMapData(player.getX(),player.getY()+1)=='@'))

{

game\_Map[j].removePlayer(player);//刪#

if(game\_Map[j].getNextMapName()=="Fongo"){

j=2;

}

else if(game\_Map[j].getNextMapName()=="Omen"){

j=0;

}

else if(game\_Map[j].getNextMapName()=="Tinda"){

j=1;

}

else{}

player.setPosition(game\_Map[j].getinitialPositionX(),game\_Map[j].getinitialPositionY());

game\_Map[j].addPlayer(player);

game\_Map[j].printMap();

}

else

{

cout<<"Can't move down, try another direction."<<endl;

cout<<endl;

game\_Map[j].printMap();

}

}

/////////////////////////////////////////////////////////////////////////////

else if(input=="a")

{

if(game\_Map[j].canMove(player.getX()-1, player.getY())==true && (game\_Map[j].getMapData(player.getX()-1,player.getY())!='@'))

{

game\_Map[j].removePlayer(player);//刪#

player.moveLeft();

game\_Map[j].addPlayer(player);//加#

//meetMonster(player);

meetMonster(player,game\_Map,j);

game\_Map[j].printMap();

}

else if((game\_Map[j].canMove(player.getX()-1,player.getY())==true) && (game\_Map[j].getMapData(player.getX()-1,player.getY())=='@'))

{

game\_Map[j].removePlayer(player);//刪#

if(game\_Map[j].getNextMapName()=="Fongo"){

j=2;

}

else if(game\_Map[j].getNextMapName()=="Omen"){

j=0;

}

else if(game\_Map[j].getNextMapName()=="Tinda"){

j=1;

}

else{}

player.setPosition(game\_Map[j].getinitialPositionX(),game\_Map[j].getinitialPositionY());

game\_Map[j].addPlayer(player);

game\_Map[j].printMap();

}

else

{

cout<<"Can't move left, try another direction."<<endl;

cout<<endl;

game\_Map[j].printMap();

}

}

//////////////////////////////////////////////////////////////////////////////////////

else if(input=="d")

{

if(game\_Map[j].canMove(player.getX()+1, player.getY())==true && (game\_Map[j].getMapData(player.getX()+1,player.getY())!='@'))

{

game\_Map[j].removePlayer(player);//刪#

player.moveRight();

game\_Map[j].addPlayer(player);//加#

//meetMonster(player);

meetMonster(player,game\_Map,j);

game\_Map[j].printMap();

}

else if((game\_Map[j].canMove(player.getX()+1,player.getY())==true) && (game\_Map[j].getMapData(player.getX()+1,player.getY())=='@'))

{

game\_Map[j].removePlayer(player);//刪#

if(game\_Map[j].getNextMapName()=="Fongo"){

j=2;

}

else if(game\_Map[j].getNextMapName()=="Omen"){

j=0;

}

else if(game\_Map[j].getNextMapName()=="Tinda"){

j=1;

}

else{}

player.setPosition(game\_Map[j].getinitialPositionX(),game\_Map[j].getinitialPositionY());

game\_Map[j].addPlayer(player);

game\_Map[j].printMap();

}

else

{

cout<<"Can't move Right, try another direction."<<endl;

cout<<endl;

game\_Map[j].printMap();

}

}

///////////////////////////////////////////////////////////////////////////

else if(input=="i")

{

player.printStatus();

cout<<"Map:"<<game\_Map[j].getName()<<endl;

}

///////////////////////////////////////////////////////////////////////////

else if(input=="h")

{

cout<<"Use 5 coins to recover your hp. (yes/no)"<<endl;

string input;

cin>>input;

if(input=="yes")

{

if(player.RpgCreature::getCoins()<5)

{

cout<<"You don't have enough coins. "<<"("<<player.RpgCreature::getCoins()<<"/5)"<<endl;

}

else

{

player.RpgCreature::reduceCoins();

player.RpgCreature::recoverHp();

}

}

else if(input=="no"){}

else{}

}

///////////////////////////////////////////////////////////////////////////

else if(input=="save")

{

//--------------------Read Data--------------------------//

fstream importSaveFile("save.txt",ios::in|ios::app);

string name,job,mapName;

int level,currentHp,maxHp,strength,defence,cirts,coin,experience,x,y;

vector<SaveData> saveDatas;

while(importSaveFile>>name>>level>>job>>currentHp>>maxHp>>strength>>defence>>cirts>>coin>>experience>>mapName>>x>>y)

{

saveDatas.push\_back(SaveData(name,level,job,currentHp,maxHp,strength,defence,cirts,coin,experience,mapName,x,y));

//cout<<name<<level<<job<<currentHp<<maxHp<<strength<<defence<<cirts<<coin<<experience<<mapName<<x<<y<<endl;

}

importSaveFile.close();

//---------------------show choice----------------------//

cout<<"You have "<<saveDatas.size()<<" records."<<endl;

for(int i=0;i<saveDatas.size();i++)

{

cout<<"---------------record "<<i<<" -----------------"<<endl;

cout<<"Name: "<<saveDatas[i].getName()<<endl;

cout<<"Job: "<<saveDatas[i].getJob()<<endl;

cout<<"Level: "<<saveDatas[i].getLevel()<<endl;

cout<<"-----------------------------------------------"<<endl;

}

cout<<"Load record (Enter 0, 1, 2, .....)"<<endl;

//--------------------Change/Overwrite Data--------------------------//

int record;

cin>>record;

//--------------------------Overwrite Data--------------------------//

if(record<saveDatas.size())

{

cout<<"Overwrite record "<<record<<" ? (yes/no)"<<endl;

string yes\_or\_no;

cin>>yes\_or\_no;

if(yes\_or\_no=="yes")

{

fstream exportFile("save.txt",ios::out|ios::trunc);

saveDatas[record].setName(player.getName());

saveDatas[record].setLevel(player.getLevel());

saveDatas[record].setJob(player.getJob());

saveDatas[record].setCurrentHp(player.getCurrentHp());

saveDatas[record].setMaxHp(player.getMaxHp());

saveDatas[record].setStrength(player.getStrength());

saveDatas[record].setDefence(player.getDefence());

saveDatas[record].setCirts(player.getCrits());

saveDatas[record].setCoin(player.getCoins());

saveDatas[record].setExperience(player.getExperience());

saveDatas[record].setmapName(game\_Map[j].getName());

saveDatas[record].setX(player.getX());

saveDatas[record].setY(player.getY());

for(int i=0;i<saveDatas.size();i++)

{

exportFile<<saveDatas[i].getName()<<endl;

exportFile<<saveDatas[i].getLevel()<<endl;

exportFile<<saveDatas[i].getJob()<<endl;

exportFile<<saveDatas[i].getCurrentHp()<<endl;

exportFile<<saveDatas[i].getMaxHp()<<endl;

exportFile<<saveDatas[i].getStrength()<<endl;

exportFile<<saveDatas[i].getDefence()<<endl;

exportFile<<saveDatas[i].getCirts()<<endl;

exportFile<<saveDatas[i].getCoin()<<endl;

exportFile<<saveDatas[i].getMapName()<<endl;

exportFile<<saveDatas[i].getExperience()<<endl;

exportFile<<saveDatas[i].getX()<<endl;

exportFile<<saveDatas[i].getY()<<endl;

}

exportFile.close();

}

else{}

}

//--------------------Add Data--------------------------//

else

{

fstream exportFile("save.txt",ios::out|ios::app);

exportFile<<name<<endl;

exportFile<<level<<endl;

exportFile<<job<<endl;

exportFile<<currentHp<<endl;

exportFile<<maxHp<<endl;

exportFile<<strength<<endl;

exportFile<<defence<<endl;

exportFile<<cirts<<endl;

exportFile<<coin<<endl;

exportFile<<experience<<endl;

exportFile<<mapName<<endl;

exportFile<<x<<endl;

exportFile<<y<<endl;

exportFile.close();

}

saveDatas.clear();

}

///////////////////////////////////////////////////////////////////////////

else if(input=="exit")

{

break;

}

///////////////////////////////////////////////////////////////////////////

else

{

cout<<"Wrong input"<<endl;

cout<<endl;

game\_Map[j].printMap();

}

}

return 0;

}

//meetMonster移到副程式//

//////////////////////////////////////////////////////////////////////////////////////////////////

///////////////////////////Encounter a monster////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////////////////////////////

//void meetMonster(RpgPerson &player){

void meetMonster(RpgPerson &player,vector<Map> game\_Map,int j){ //monster name

int meet\_monster\_prabobility;

/\* initialize random seed: \*/

srand (time(NULL));

/\* generate secret number between 1 and 4: \*/

meet\_monster\_prabobility = rand() % 4 + 1;

if(meet\_monster\_prabobility == 1)

{

/////////////////////////////////////////////////////

Monster \*chooseMonster;

if(game\_Map[j].randomMonster()== "Slime")//"Slime" //problem: game\_Map[j].randomMonster()

{

chooseMonster = new Monster\_Slime("Slime", 30, 10, 3, 2, 20, 1);

}

else if(game\_Map[j].randomMonster()== "Troll") //"Troll"

{

chooseMonster = new Monster\_Troll("Troll", 35, 20, 5, 10, 35, 5);

}

else if(game\_Map[j].randomMonster()== "Dragon") //"Dragon"

{

chooseMonster = new Monster\_Dragon("Dragon", 150, 20, 30, 25, 1, 15);

}

else{}

Monster &monster = \*chooseMonster;

cout<<"########"<<monster.RpgCreature::getName()<<"######"<<endl;////////////////////////////////

cout<<"########"<<monster.RpgCreature::getCoins()<<"######"<<endl;////////////////////////////////

cout<<"########"<<monster.RpgCreature::getExperience()<<"######"<<endl;////////////////////////////////

//I can use chooseMonster but I have to use monster->getName()

/////////////////////////////////////////////////////////

//Monster monster;

//monster=Monster("Slime",30,10,3,20);

////Monster monster("Slime",20,6,3); //和上面兩行有何不同?Ask!

cout<<endl;

cout<<"You meet a "<<monster.RpgCreature::getName()<<"!!! The fight starts!!!"<<endl;

cout<<endl;

bool is\_Effective\_Input=false;

do{

string attack\_skill\_escape;

cout<<"------------------------------"<<endl;

cout<<"Enter '1' to attack, '2' to use skill, '3' to escape."<<endl;

cout<<"------------------------------"<<endl;

cin>>attack\_skill\_escape;

if(attack\_skill\_escape=="1")

{

is\_Effective\_Input=true;

//////////////////////////////

//player attack monster///////

/////////////////////////////

player.attack(monster);

if(monster.RpgCreature::isDead()== true)

{

//int value;

//srand (time(NULL));

//value = rand() % 2 + 1;

player.raiseExperience(monster.RpgCreature::getExperience());

player.raiseCoin(monster.RpgCreature::getCoins());

cout<<"You beat "<<monster.RpgCreature::getName()<<" ,got "

<<monster.RpgCreature::getExperience()<<" exp."<<endl;

if(player.checkLevelUp()==true)

{

player.levelUp();

player.statusUp();

cout<<"Level up!!! You are level ";

cout<<player.getLevel()<<endl;

}

else{}

//player.RpgCreature::recoverHp();

//cout<<"Hp recovered."<<endl;

cout<<endl;

break;

}

else

{

cout<<monster.RpgCreature::getName()<<"'s hp:"<<" "

<<monster.RpgCreature::getCurrentHp()<<endl;

}

//////////////////////////////

//monster attack player///////

/////////////////////////////

monster.attack(player);

if(player.RpgCreature::isDead()== true)

{

int value;

srand (time(NULL));

value = rand() % 2 + 1;

//player.raiseExperience(value);

player.decreaseExperience(value);

cout<<"Your hero is dead. Lost "<<value<<" exp."<<endl;

player.RpgCreature::recoverHp();

cout<<"Hp recovered."<<endl;

cout<<endl;

break;

}

else

{

cout<<player.RpgCreature::getName()<<"'s hp:"<<" "

<<player.RpgCreature::getCurrentHp()<<endl;

}

}

else if(attack\_skill\_escape=="2")

{

is\_Effective\_Input=true;

//////////////////////////////

//player Choose Skills////////

/////////////////////////////

player.printfSkill();

cout<<"Choose Skill:";

bool is\_input\_right=true;

do{

int choose;

cin>>choose;

if(player.checkSkill(choose,player)==true)

{

switch(choose)

{

case 1:

{

player.attackSkill(monster);

}

break;

case 2:

{

player.healSkill(player);

}

break;

case 3:

{

player.speacialSkill(monster);

}

break;

default: //It's won't be excute Think!!!

{

cout<<"Have no this choose"<<endl;

cout<<"Choose another one:";

is\_input\_right=false;

}

}

is\_input\_right=true;

}

else

{

cout<<"Cannot use this skill! Your Level is too low"<<endl;

cout<<"Choose another one:";

is\_input\_right=false;

}

}while(is\_input\_right==false);

//////////////Judge monster\_isDead()

//////////////

if(monster.RpgCreature::isDead()== true)

{

//int value;

//srand (time(NULL));

//value = rand() % 2 + 1;

player.raiseExperience(monster.RpgCreature::getExperience());

player.raiseCoin(monster.RpgCreature::getCoins());

cout<<"You beat "<<monster.RpgCreature::getName()<<" ,got "

<<monster.RpgCreature::getExperience()<<" exp."<<endl;

if(player.checkLevelUp()==true)

{

player.levelUp();

player.statusUp();

cout<<"Level up!!! You are level ";

cout<<player.getLevel()<<endl;

}

else{}

//player.RpgCreature::recoverHp();

//cout<<"Hp recovered."<<endl;

cout<<endl;

break;

}

else

{

cout<<monster.RpgCreature::getName()<<"'s hp:"<<" "

<<monster.RpgCreature::getCurrentHp()<<endl;

}

//////////////////////////////

//monster attack player///////

/////////////////////////////

monster.attack(player);

if(player.RpgCreature::isDead()== true)

{

int value;

srand (time(NULL));

value = rand() % 2 + 1;

//player.raiseExperience(value);

player.decreaseExperience(value);

cout<<"Your hero is dead."<<" Lost "<<value<<" exp."<<endl;

player.RpgCreature::recoverHp();

cout<<"Hp recovered."<<endl;

cout<<endl;

break;

}

else

{

cout<<player.RpgCreature::getName()<<"'s hp:"<<" "

<<player.RpgCreature::getCurrentHp()<<endl;

}

}

/////////////////////////////

//////Escape/////////////////

/////////////////////////////

else if(attack\_skill\_escape=="3")

{

is\_Effective\_Input=true;

int escape\_prabobility;

srand (time(NULL));

meet\_monster\_prabobility = rand() % 3 + 1;

if(meet\_monster\_prabobility == 1)

{

cout<<"You successfully escaped from the fight!"<<endl;

break;

}

else

{

cout<<"Escape failed!!!"<<endl;

//////////////////////////////

//Monster attack player///////

/////////////////////////////

monster.attack(player);

if(player.RpgCreature::isDead()== true)

{

int value;

srand (time(NULL));

value = rand() % 2 + 1;

//player.raiseExperience(value);

player.decreaseExperience(value);

cout<<"Your hero is dead."<<" Lost "<<value<<" exp."<<endl;

player.RpgCreature::recoverHp();

cout<<"Hp recovered."<<endl;

cout<<endl;

break;

}

else

{

cout<<player.RpgCreature::getName()<<"'s hp:"<<" "

<<player.RpgCreature::getCurrentHp()<<endl;

}

}

}

else

{

cout<<"Wrong Input"<<endl;

is\_Effective\_Input=true;

}

}while(is\_Effective\_Input==true);

}

else{}

}

///////////////////////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////////////////////////

**Map.cpp**

#include "Map.h"

//--------------------------

//--------------------------

Map::Map(){}

Map::Map(std::string name, std::string nextMapName, int initialPositionX, int initialPositionY, int width,

int height, std::vector<std::string> mapData, std::vector<std::string>monsterName){

this->name = name;

this->nextMapName = nextMapName;

this->initialPositionX = initialPositionX;

this->initialPositionY = initialPositionY;

this->width = width;

this->height = height;

this->mapData = mapData;

this->monsterName = monsterName;

for(int i=0; i<mapData.size();i++)

for(int j=0; j<mapData[i].size();j++)

if(mapData[i][j] == '0')

this->mapData[i][j] = ' ';

}

Map::~Map(){}

std::string Map::getName(){

return this->name;

}

std::string Map::getNextMapName(){

return this->nextMapName;

}

int Map::getWidth(){

return this->width;

}

int Map::getHeight(){

return this-> height;

}

int Map::getinitialPositionX(){

return this-> initialPositionX;

}

int Map::getinitialPositionY(){

return this-> initialPositionY;

}

char Map::getMapData(int x, int y){

return this->mapData[y][x];

}

bool Map::canMove(int x, int y){

if(0 <= y && y <= height-1 && 0 <= x && x <= width-1){ //<height-1;< width-1

if(mapData[y][x] == '1')

return false;

else

return true;

}

else

return false;

}

void Map::printMap(){

for(int i=0;i<mapData.size();i++)

{

cout << mapData[i] <<endl;

}

}

void Map::addPlayer(RpgPerson& player){

mapData[player.getY()][player.getX()] = '#';

}

void Map::removePlayer(RpgPerson& player){

mapData[player.getY()][player.getX()] = ' ';

}

std::string Map::randomMonster(){

return monsterName[rand()%(this->monsterName.size())];

}

**Map.h**

#ifndef MAP\_H\_INCLUDED

#define MAP\_H\_INCLUDED

#include <vector>

#include "RpgPerson.h"

//----------------------------

#include<string>

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-----------------

using namespace std;

class Map{

public:

Map();

Map(std::string name, std::string nextMapName, int initialPositionX, int initialPositionY, int width,

int height,std::vector<std::string> mapData, std::vector<std::string>monsterName);

~Map();

std::string getName();

std::string getNextMapName();

int getWidth();

int getHeight();

int getinitialPositionX();

int getinitialPositionY();

char getMapData(int x, int y);

bool canMove(int x, int y);

void printMap(); //check

void addPlayer(RpgPerson& player);

void removePlayer(RpgPerson& player);

string randomMonster();

private:

std::string name;

std::string nextMapName;

int width;

int height;

int initialPositionX;

int initialPositionY;

std::vector<std::string> mapData;

RpgPerson\* player;

std::vector<std::string> monsterName;

};

#endif // MAP\_H\_INCLUDED

**Monster.cpp**

#include "Monster.h"

Monster::Monster(){}

Monster::Monster(std::string name, int maxHp, int strength, int defence, int experience, int cirts, int coins){

//is that right??

this->name=name;

this->maxHp=maxHp;

this->currentHp=maxHp; //this->currentHp=maxHp;因為原本沒有，要用creature時會用到currentHp

this->strength=strength;

this->defence=defence;

this->cirts=cirts;

this->coins=coins;

this->experience=experience;

}

**Monster.h**

#ifndef MONSTER\_H\_INCLUDED

#define MONSTER\_H\_INCLUDED

#include "RpgCreature.h"

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-----------------

class Monster: public RpgCreature{

public:

Monster(); //I added.

//~Monster();

Monster(std::string name, int maxHp, int strength, int defence, int experience, int cirts, int coins);

private:

};

#endif // MONSTER\_H\_INCLUDED

**Monster.Dragon.cpp**

#include "Monster\_Dragon.h"

Monster\_Dragon::Monster\_Dragon(){}

Monster\_Dragon::Monster\_Dragon(std::string name, int maxHp, int strength, int defence, int experience, int cirts, int coins){

//is that right??

this->name=name;

//this->maxHp=maxHp;

this->currentHp=maxHp; //this->currentHp=maxHp;因為原本沒有，要用creature時會用到currentHp

this->strength=strength;

this->defence=defence;

this->cirts=cirts;

this->coins=coins;

this->experience=experience;

}

**Monster.Dragon.h**

#ifndef MONSTER\_DRAGON\_H\_INCLUDED

#define MONSTER\_DRAGON\_H\_INCLUDED

//----------------------------------

#include "RpgCreature.h"

#include "Monster.h"

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-----------------

class Monster\_Dragon: public Monster{

public:

Monster\_Dragon(); //I added.

//~Monster\_Dragon();

Monster\_Dragon(std::string name, int maxHp, int strength, int defence, int experience, int cirts, int coins);

private:

};

#endif // MONSTER\_DRAGON\_H\_INCLUDED

**Monster\_Slime.cpp**

#include "Monster\_Slime.h"

Monster\_Slime::Monster\_Slime(){}

Monster\_Slime::Monster\_Slime(std::string name, int maxHp, int strength, int defence, int experience, int cirts, int coins){

//is that right??

this->name=name;

this->maxHp=maxHp;

this->currentHp=maxHp; //this->currentHp=maxHp;因為原本沒有，要用creature時會用到currentHp

this->strength=strength;

this->defence=defence;

this->cirts=cirts;

this->coins=coins;

this->experience=experience;

}

**Monster\_Slime.h**

#ifndef MONSTER\_SLIME\_H\_INCLUDED

#define MONSTER\_SLIME\_H\_INCLUDED

//----------------------------------

#include "RpgCreature.h"

#include "Monster.h"

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-----------------

class Monster\_Slime: public Monster{

public:

Monster\_Slime(); //I added.

//~Monster\_Slime();

Monster\_Slime(std::string name, int maxHp, int strength, int defence, int experience, int cirts, int coins);

private:

};

#endif // MONSTER\_SLIME\_H\_INCLUDED

**Monster\_Troll.cpp**

#include "Monster\_Troll.h"

Monster\_Troll::Monster\_Troll(){}

Monster\_Troll::Monster\_Troll(std::string name, int maxHp, int strength, int defence, int experience, int cirts, int coins){

//is that right??

this->name=name;

this->maxHp=maxHp;

this->currentHp=maxHp; //this->currentHp=maxHp;因為原本沒有，要用creature時會用到currentHp

this->strength=strength;

this->defence=defence;

this->cirts=cirts;

this->coins=coins;

this->experience=experience;

}

**Monster\_Troll.h**

#ifndef MONSTER\_TROLL\_H\_INCLUDED

#define MONSTER\_TROLL\_H\_INCLUDED

//----------------------------------

#include "RpgCreature.h"

#include "Monster.h"

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-----------------

class Monster\_Troll: public Monster{

public:

Monster\_Troll(); //I added.

//~Monster\_Troll();

Monster\_Troll(std::string name, int maxHp, int strength, int defence, int experience, int cirts, int coins);

private:

};

#endif // MONSTER\_TROLL\_H\_INCLUDED

**RpgCreature.cpp**

#include "RpgCreature.h"

//-----------------------

#include "Map.h"

//-----------------------

RpgCreature::RpgCreature(){

}

RpgCreature::RpgCreature(std::string name){

this->name = name;

}

RpgCreature::RpgCreature(std::string name, int maxHp, int strength, int defence,int cirts,int coin,int experience){

this->name = name;

this->maxHp = maxHp;

this->strength = strength;

this->defence = defence;

this->cirts = cirts; //每種職業不一樣 (初始值20)

this->coins = coins;

this->experience = experience;

}

std::string RpgCreature::getName(){

return this->name;

}

int RpgCreature::getMaxHp(){

return this->maxHp;

}

int RpgCreature::getCurrentHp(){

return this->currentHp;

}

int RpgCreature::getStrength(){

return this->strength;

}

int RpgCreature::getDefence(){

return this->defence;

}

int RpgCreature::getCrits(){

return this->cirts;

}

void RpgCreature::changeStrength(int point){

this->strength = point;

}

void RpgCreature::changeDefence(int point){

this->defence = point;

}

void RpgCreature::changeCirts(int point){

this->cirts = point;

}

bool RpgCreature::isDead(){

if(currentHp <= 0)

{

return true;

}

else return false;

}

void RpgCreature::loseHp(int losehp){

this->currentHp -= losehp;

}

void RpgCreature::recoverHp(){

this->currentHp = maxHp;

}

void RpgCreature::attack(RpgCreature& enemy){

//think and check!//

/\* initialize random seed: \*/

srand (time(NULL));

int drift\_damage;

/\* generate secret number between -3 and 3: \*/

drift\_damage = rand() % 6 - 3;

int losehp;

losehp=this->strength-enemy.getDefence()+drift\_damage;

cout<<name<<" "<<"attack, caused"<<" "<<losehp<<" "<<"damages!"<<endl;

if(enemy.getDefence()>=(losehp))

{

enemy.loseHp(1);

}

else

{

double chance\_of\_cirts,sample;

sample=rand();

chance\_of\_cirts=(sample+cirts)/sample;

if(chance\_of\_cirts>=1)

{

enemy.loseHp(losehp);

}

else

{

enemy.loseHp(losehp\*2);

}

}

}

void RpgCreature::reduceCoins(){

this->coins=this->coins-5;

}

int RpgCreature::getCoins(){

return this->coins;

}

int RpgCreature::getExperience(){

return this->experience;

}

**RpgCreature.h**

#ifndef RPGCREATURE\_H\_INCLUDED

#define RPGCREATURE\_H\_INCLUDED

#include <iostream>

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-----------------

using namespace std;

class RpgCreature{

public:

RpgCreature();

RpgCreature(std::string name);

RpgCreature(std::string name, int maxHp, int strength, int defence,int cirts,int coin,int experience);

std::string getName();

int getMaxHp();

int getCurrentHp();

int getStrength();

int getDefence();

bool isDead();

void loseHp(int losehp);

void recoverHp();

void attack(RpgCreature& enemy);

int getCrits();//add cirts

int getExperience();

void changeStrength(int point);

void changeDefence(int point);

void changeCirts(int point);

void reduceCoins();

void decreaseCoins();

int getCoins();

protected:

std::string name;

int maxHp;

int currentHp;

int strength;

int defence;

int cirts;// add cirts

int coins;

int experience;

};

#endif // RPGCREATURE\_H\_INCLUDED

**RpgPerson.cpp**

#include "RpgPerson.h"

//------------------------

#include "Map.h"

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//------------------------

RpgPerson::RpgPerson(){}

RpgPerson::RpgPerson(std::string name, int maxHp, int strength, int defence, int cirts,int coins,

int experience, int x, int y){

this->name=name;

this->maxHp=maxHp;

this->currentHp=maxHp; //this->currentHp=maxHp;因為原本沒有，要用creature時會用到currentHp

this->strength=strength;

this->defence=defence;

this->x=x;

this->y=y;

this->cirts = cirts;

this->coins = coins;

this->experience = experience;

//this->experience=0;

//this->experience=0; //I add

//this->level=3; //I add coz there is 1 in .h.

//this->cirts=20; //每種職業不一樣 (初始值20)

//為什麼要寫成 this->name=name;等...因為初始值才能直接設定

}

int RpgPerson::getX(){

return this->x;

}

int RpgPerson::getY(){

return this->y;

}

void RpgPerson::setPosition(int x, int y){

this->x=x;

this->y=y;

}

void RpgPerson::moveUp(){

this->y=this->y-1;

}

void RpgPerson::moveDown(){

this->y=this->y+1;

}

void RpgPerson::moveLeft(){

this->x=this->x-1;

}

void RpgPerson::moveRight(){

this->x=x+1;

}

void RpgPerson::raiseExperience(int experience){

this->experience= this->experience+experience;

}

void RpgPerson::raiseCoin(int coins){

this->coins= this->coins+coins;

}

void RpgPerson::decreaseExperience(int experience){

if(((this->experience)-experience) <= 0)

{

this->experience=0;

}

else

{

this->experience=(this->experience)-experience;

}

}

bool RpgPerson::checkLevelUp(){

if(experience>7\*(0.5+level/2))

{

return true;

}

else

{

return false;

}

}

void RpgPerson::levelUp(){

this->experience=this->experience%level;

this->level=this->level+1;//提升一級

}

int RpgPerson::getLevel(){

return this->level;

}

//void RpgPerson::statusUp(){ //Abstract?? -->"player.statusup()" called this one

// this->maxHp=this->maxHp+5; //Hp+5

// this->strength=this->strength+1; //Strength+1

// this->defence=this->defence+1; //defence+1

//}

string RpgPerson::getJob(){ // I add

return this->job;

}

void RpgPerson::printStatus(){

cout<<"Name:"<<name<<endl;

cout<<"LV:"<<getLevel()<<endl;

cout<<"Job:"<<job<<endl;

cout<<"Hp:"<<currentHp<<"/"<<maxHp<<endl;

cout<<"Strength:"<<strength<<endl;

cout<<"Defence:"<<defence<<endl;

cout<<"Crits:"<<cirts<<endl;

cout<<"Coins:"<<coins<<endl;

cout<<"Exp:"<<experience<<"/"<<7\*(0.5+level/2)<<endl;

//cout<<"Map:"<<Map::getName()<<endl;

}

void RpgPerson::attackSkill(RpgCreature& monster){

int losehp;

losehp = monster.getCurrentHp()/2;

monster.loseHp(losehp); //Is this okay? class Creature method is used here?

}

void RpgPerson::healSkill(RpgPerson& player)

{

player.currentHp = player.currentHp+(player.getMaxHp() - player.currentHp)\*8/10; //player.currentHp is okay?

if(player.currentHp>=getMaxHp())

{

this->currentHp=player.getMaxHp();

}

else

{

this->currentHp = player.currentHp;

}

}

bool RpgPerson::checkSkill(int choose,RpgPerson& player){

if(choose==1 && player.getLevel()>=1)

{

return true;

}

else if(choose==2 && player.getLevel()>=2)

{

return true;

}

else if(choose==3 && player.getLevel()>=3)

{

return true;

}

else

{

return false;

}

}

**RpgPerson.h**

#ifndef RPGPERSON\_H\_INCLUDED

#define RPGPERSON\_H\_INCLUDED

#include "RpgCreature.h"

//----------------------

#include <vector>

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-----------------

class RpgPerson: public RpgCreature{

public:

RpgPerson();

RpgPerson(std::string name, int maxHp, int strength, int defence, int cirts,int coins,

int experience, int x, int y);

int getX();

int getY();

void setPosition(int x, int y);

void moveUp();

void moveDown();

void moveLeft();

void moveRight();

void raiseExperience(int experience);

void decreaseExperience(int experience);

void raiseCoin(int coins);

bool checkLevelUp();

void levelUp();

int getLevel();

virtual void statusUp(){};//or use statusUp()=0; // I change?? is okay? //Statusup()加入cirts up!-->移到各個角色中 (haven't done)

void printStatus();

string getJob(); //I add

void attackSkill(RpgCreature& monster);

void healSkill(RpgPerson& player); //

virtual void speacialSkill(RpgCreature& monster){};

virtual void printfSkill(){};

bool checkSkill(int choose,RpgPerson& player);

//private:

protected://change private:-->protected:

int level;

//int experience;

int x;

int y;

//int value;

string job;// I add

};

#endif // RPGPERSON\_H\_INCLUDED

**RpgPerson\_Magician.cpp**

#include "RpgPerson\_Magician.h"

RpgPerson\_Magician::RpgPerson\_Magician(){}

RpgPerson\_Magician::RpgPerson\_Magician(std::string name, int level, string job,int currentHp, int maxHp, int strength, int defence,

int cirts,int coins,int experience, int x, int y){

this->name=name;

this->maxHp=maxHp;

this->currentHp=maxHp;

this->strength=strength;

this->defence=defence;

this->x=x;

this->y=y;

this->experience=experience;

this->level=level;

this->coins=coins;

this->cirts=cirts; //每種職業不一樣 (初始值20)

this->job="Magician"; // I add

}

void RpgPerson\_Magician::statusUp(){ //+virtual才可以!!!!!!!!!!!!!

this->maxHp=this->maxHp+5; //Hp+5

this->strength=this->strength+2; //Strength+2

this->defence=this->defence+1; //defence+1

this->cirts=this->cirts+0; //cirts+0

}

void RpgPerson\_Magician::speacialSkill(RpgCreature& monster){ //how to programming would be better(won't change this->defence)

int point;

if(monster.getCrits()<=0){

monster.changeCirts(0);

}

else{

point=monster.getCrits()-rand()%10;

monster.changeCirts(point);

}

}

void RpgPerson\_Magician::printfSkill(){

cout<<"<1>"<<"attack\_skill:"<<"Everytime, Monster will lose 50% Hp."<<endl;

cout<<"<2>"<<"heal\_skill:"<<"Everytime, Player will recover 80% Hp of losing hp."<<endl;

cout<<"<3>"<<"speacial\_skill:"<<"Everytime, Monster's cirts will be decreased:0~9 "<<endl;

}

//void RpgPerson\_Magician::setJob(){ // I add

// this->Job="Magician";

//}

**RpgPerson\_Magician.h**

#ifndef Magician\_H\_INCLUDED

#define Magician\_H\_INCLUDED

//----------------------------------

#include "RpgCreature.h"

#include "RpgPerson.h"

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-----------------

class RpgPerson\_Magician: public RpgPerson{

public:

RpgPerson\_Magician();

RpgPerson\_Magician(std::string name, int level, string job,int currentHp, int maxHp, int strength, int defence,

int cirts,int coins,int experience, int x, int y); //add cirts

//void statusUp(); //add cirts (Magician)

//////////////////////////////////////I add (Could I use function? Don't implement in Class?)

void statusUp();

virtual void speacialSkill(RpgCreature& monster);

virtual void printfSkill();

};

#endif // Magician\_H\_INCLUDED

**RpgPerson\_SwordMan.cpp**

#include "RpgPerson\_SwordMan.h"

RpgPerson\_SwordMan::RpgPerson\_SwordMan(){}

RpgPerson\_SwordMan::RpgPerson\_SwordMan(std::string name, int level, string job,int currentHp, int maxHp, int strength, int defence,

int cirts,int coins,int experience, int x, int y){

this->name=name;

this->maxHp=maxHp;

this->currentHp=maxHp;

this->strength=strength;

this->defence=defence;

this->x=x;

this->y=y;

this->experience=experience;

this->coins=coins;

this->level=level;

this->cirts=cirts; //每種職業不一樣 (初始值20)

this->job="SwordMan"; // I add

}

void RpgPerson\_SwordMan::statusUp(){

this->maxHp=this->maxHp+5; //Hp+5

this->strength=this->strength+1; //Strength+1

this->defence=this->defence+2; //defence+2

this->cirts=this->cirts+0; //cirts+0

}

void RpgPerson\_SwordMan::speacialSkill(RpgCreature& monster){

int point;

if(monster.getDefence()<=0){

monster.changeDefence(0);

}

else{

point=monster.getDefence()-rand()%2;

monster.changeDefence(point);

}

}

void RpgPerson\_SwordMan::printfSkill(){

cout<<"<1>"<<"attack\_skill:"<<"Everytime, Monster will lose 50% Hp."<<endl;

cout<<"<2>"<<"heal\_skill:"<<"Everytime, Player will recover 80% Hp of losing hp."<<endl;

cout<<"<3>"<<"speacial\_skill:"<<"Everytime, Monster's defence will be decreased:0~2 "<<endl;

}

//void RpgPerson\_SwordMan::setJob(){ // I add

// this->Job="SwordMan";

//}

**RpgPerson\_SwordMan.h**

#ifndef SWORDMAN\_H\_INCLUDED

#define SWORDMAN\_H\_INCLUDED

//----------------------------------

#include "RpgCreature.h"

#include "RpgPerson.h"

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-----------------

class RpgPerson\_SwordMan: public RpgPerson{

public:

RpgPerson\_SwordMan();

RpgPerson\_SwordMan(std::string name, int level, string job,int currentHp, int maxHp, int strength, int defence,

int cirts,int coins,int experience, int x, int y); //add cirts

//void statusUp(); //add cirts (SwordMan)

/////////////////////////////////////////////I add (Could I use function? Don't implement in Class?)

void statusUp();

void speacialSkill(RpgCreature& monster);

void printfSkill();

};

#endif // SWORDMAN\_H\_INCLUDED

**RpgPerson\_Thief.cpp**

#include "RpgPerson\_Thief.h"

RpgPerson\_Thief::RpgPerson\_Thief(){}

RpgPerson\_Thief::RpgPerson\_Thief(std::string name, int level, string job,int currentHp, int maxHp, int strength, int defence,

int cirts,int coins,int experience, int x, int y){

this->name=name;

this->maxHp=maxHp;

this->currentHp=maxHp;

this->strength=strength;

this->defence=defence;

this->x=x;

this->y=y;

this->experience=experience;

this->coins=coins;

this->level=level;

this->cirts=cirts; //每種職業不一樣 (初始值20)

this->job="Thief"; // I add

}

void RpgPerson\_Thief::statusUp(){

this->maxHp=this->maxHp+5; //Hp+5

this->strength=this->strength+1; //Strength+1

this->defence=this->defence+1; //defence+1

this->cirts=this->cirts+5; //cirts+5

}

void RpgPerson\_Thief::speacialSkill(RpgCreature& monster){

int point;

if(monster.getStrength()<=0){

monster.changeStrength(0);

}

else{

point=monster.getStrength()-rand()%5;

monster.changeStrength(point);

}

}

void RpgPerson\_Thief::printfSkill(){

cout<<"<1>"<<"attack\_skill:"<<"Everytime, Monster will lose 50% Hp."<<endl;

cout<<"<2>"<<"heal\_skill:"<<"Everytime, Player will recover 80% Hp of losing hp."<<endl;

cout<<"<3>"<<"speacial\_skill:"<<"Everytime, Monster's strength will be decreased:0~5"<<endl;

}

//void RpgPerson\_Thief::setJob(){ // I add

// this->Job="Thief";

//}

**RpgPerson\_Thief.h**

#ifndef THIEF\_H\_INCLUDED

#define THIEF\_H\_INCLUDED

//----------------------------------

#include "RpgCreature.h"

#include "RpgPerson.h"

//-----------------<Random>

#include<time.h>

#include<stdlib.h>

#include<stdio.h>

//-----------------

class RpgPerson\_Thief: public RpgPerson{

public:

RpgPerson\_Thief();

RpgPerson\_Thief(std::string name, int level, string job,int currentHp, int maxHp, int strength, int defence,

int cirts,int coins,int experience, int x, int y);//add cirts

//void statusUp(); //add cirts (Thief)

/////////////////////////////////////////I add (Could I use function? Don't implement in Class?)

void statusUp();

void speacialSkill(RpgCreature& monster);

void printfSkill();

};

#endif // THIEF\_H\_INCLUDED