**OOP Assignment2**

Due: Nov 11, 2015

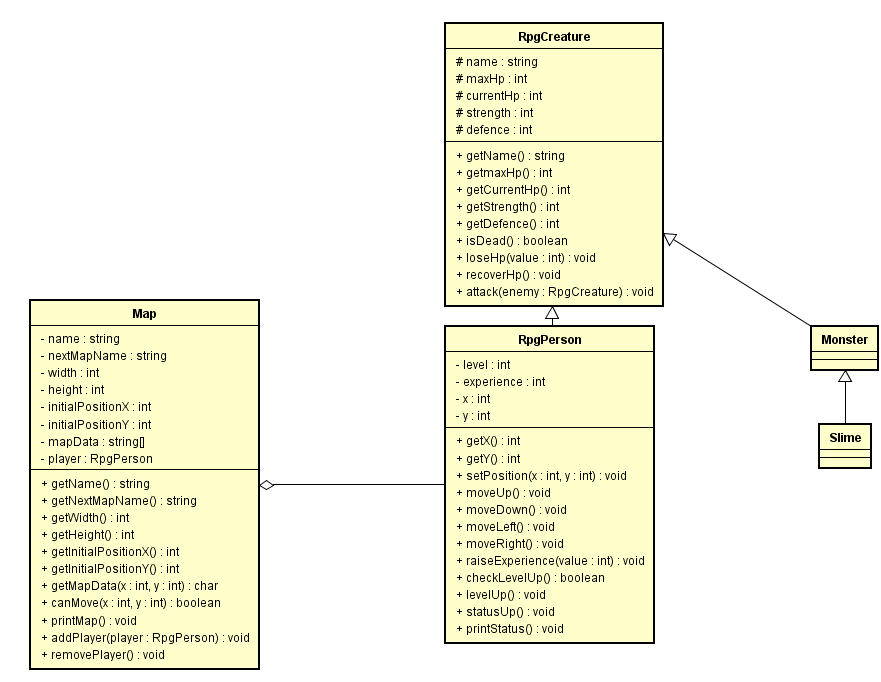
請務必填寫學號系級姓名、並繳交紙本，以免成績登錄錯誤。

學號系級姓名: 103703010 資科二 傅信穎

請完成下列項目後，將電子檔上傳至moodle。另外，請將填寫過的本份文件列印出來，依規定時間內繳交給助教(大仁104室，PerCom Lab張惟誠、盧威辰)或下週上課當場繳交給老師。未交紙本者最高以70分計算。

繳交規定：請將以下檔案及資料夾壓縮成zip、7z或rar，檔名為OOP\_assignment2\_學號\_姓名，例如：OOP\_assignment2\_103703000\_王小明.zip

1. Source資料夾，內含所有程式碼(.cpp及.h)
2. 本份文件(.doc或.docx)

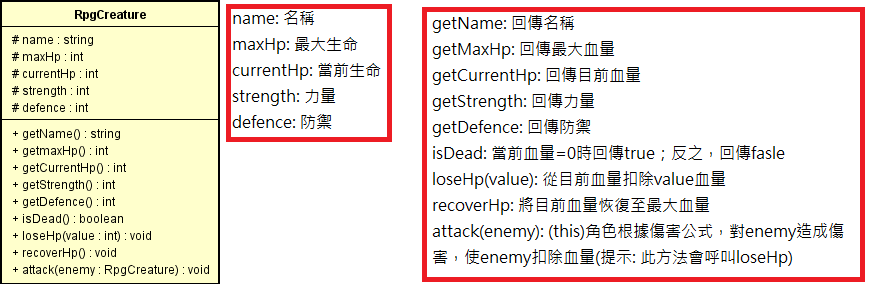
請根據以下類別圖，用C++實作簡易RPG系統

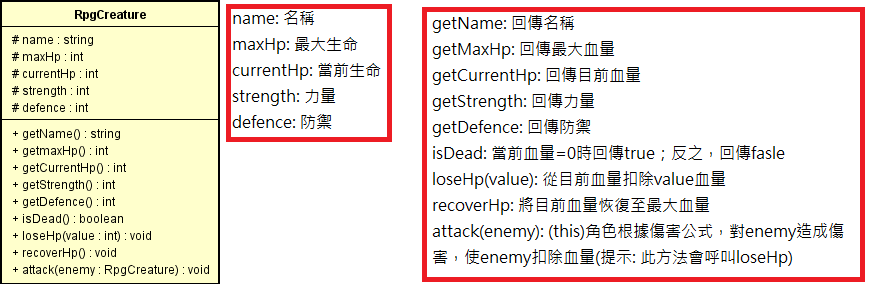
Map類別說明





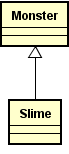
RpgCreature類別說明





Monster及Slime類別說明：

本次作業不強迫做出此兩類別，僅為了擴充性所繪製



RpgPerson類別說明

c



可至moodle下載操作範例影片(OOP\_assignment2\_demo.mp4)，詳細說明如下

* 程式操作流程：

1. 開始時，印出歡迎訊息，並要求使用者輸入角色名稱，格式如下

-----------------------------------------

| Welcome to oop assignment2 rpg. |

------------------------------------------

Enter your name:

1. 印出玩家角色當前地圖及操作說明，等待使用者輸入指令(指令請閱讀下方指令說明)，格式如下

\*\*\*\*\*

Use w/s/a/d to move up/down/left/right.

Enter i to show your status.

Enter exit to leave game.

\*\*\*\*\*

1. 接收指令後，進行相對應動作
2. 如果遊戲尚未結束，回流程2

* 指令說明：
  + 一般模式
    - w：角色在當前地圖往上移動一格(y座標-1)
      * 若角色上方一格為障礙物，角色無法移動(座標不變)，並印出下列訊息

Can't move up, try another direction.

* + - s：角色在當前地圖往下移動一格(y座標+1)
      * 若角色下方一格為障礙物，角色無法移動(座標不變)，並印出下列訊息

Can't move down, try another direction.

* + - a：角色在當前地圖往左移動一格(x座標-1)
      * 若角色左方一格為障礙物，角色無法移動(座標不變)，並印出下列訊息

Can't move left, try another direction.

* + - d：角色在當前地圖往右移動一格(x座標+1)
      * 若角色右方一格為障礙物，角色無法移動(座標不變)，並印出下列訊息

Can't move right, try another direction.

* + - i：印出角色基本資料，格式如下

|  |  |
| --- | --- |
| 說明 | 例 |
| Name: 角色名稱  Lv: 角色等級  Hp: 角色(目前等級)最大血量  Strength: 角色力量  Defence: 角色防禦  Exp: 角色目前經驗/升等所需經驗  Map: 所在地圖名稱 | Name: John  Lv: 2  Hp: 30  Strength: 8  Defence: 3  Exp: 0/2  Map: Omen |

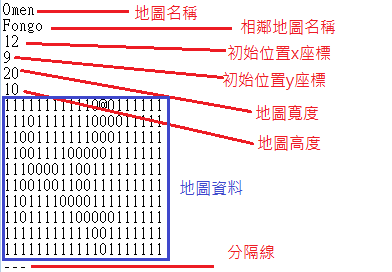
* + - exit：結束程式
    - 其他輸入：忽略並等待使用者重新輸入
* 角色如果成功移動，有機率(25%)遇到怪物，遇到怪物時進入戰鬥模式
  + 遇到怪物時，印出下列訊息

You meet a 怪物名稱!!! The fight starts!!!

* 例：

You meet a Slime!!! The fight starts!!!

* + 戰鬥模式（詳細內容見下方『戰鬥機制』）
    - f：戰鬥
    - e：逃跑
* 角色
  + 角色建立時，基本屬性如下
    - 血量(hp)：25
    - 等級(Lv)：1
    - 力量(strength)：7
    - 防禦(defence)：2
    - 所在地圖：Omen
  + 角色每提升一級
    - Hp+5
    - Strength+1
    - Defence+1
* 怪物
  + 本次作業只有一種怪物，資料如下
    - 名稱(name)：Slime
    - 血量(hp)：20
    - 力量(strength)：6
    - 防禦(defence)：3
* 地圖
  + 請用讀檔的方式將map.txt內的地圖資料加到RPG系統中，map.txt的格式如下(逗號代表換行)：「地圖名稱、相鄰地圖名稱、初始位置x座標、初始位置y座標、地圖寬度、地圖高度、地圖資料、分隔線」。



* + 座標軸定義：橫向為x軸，往右增加，往左減少；縱向為y軸，往上減少，往下增加。座標不會有負的。每一個字元為座標軸的一單位。
  + 地圖資料中，0表示可讓角色通行(空地)；1表示角色無法通行(障礙物)；#代表角色；@代表傳送點。
    - 注意：印出地圖時，0請用一格空格(“ “)取代
  + 當角色移動到傳送點上時，角色會移動至相鄰地圖的初始位置上；地圖上只會有一個傳送點。
* 戰鬥機制
  + 戰鬥
    - 進入戰鬥時，印出下列訊息

-----

Enter f to fight, e to escape.

-----

* + - 玩家的角色與怪物輪流攻擊，若選擇戰鬥(使用者輸入f)，玩家角色先展開攻擊；若選擇逃跑(使用者輸入e)且逃跑失敗，怪物先展開攻擊
    - 若輸入f和e之外的字串，印出下列訊息

Wrong input!

* + - 傷害計算方式為：

造成傷害值 ＝ 攻擊方力量(strength) – 被攻擊方防禦(defence)

（注意：不會小於零）

* + - 造成傷害時印出以下訊息

攻擊者 attacked 被攻擊者, caused 傷害值 damages!

被攻擊者’s hp: 被攻擊者剩餘血量

* + - * 例：

John attacked Slime, caused 13 damages!

Slime’s hp: 25

* + - 當有一方血量(hp)小於等於0時戰鬥結束
      * 角色死亡時，印出下列訊息，並結束程式

Your hero is dead. Game Over.

* + - * 怪物死亡時，印出下列訊息，並增加經驗值

You beat 被攻擊者, got 增加的經驗值 exp.

* + - * 例：

You beat Slime, add 1 exp.

* 贏得戰鬥後，可以隨機獲得1~2的經驗值
  + 逃跑
    - 進入戰鬥時，若選擇逃跑，有30%機率逃跑成功
    - 逃跑成功時印出下列訊息，結束戰鬥並回到地圖，不會增加經驗值

You successfully escaped from the fight!

* + - 逃跑失敗時印出下列訊息，戰鬥開始並由怪物先展開攻擊

Escape failed!!!

* + 戰鬥結束後，角色血量(hp)回復至(目前等級)最大血量，並印出下列訊息。

Hp recovered.

* 升級
  + 當角色經驗值(experience)到達指定數值，等級(level)上升一級(+1)並印出下列訊息

Level up!!! You are level 目前等級（升等之後）.

* + 經驗值公式
    - 升級所需經驗值＝角色目前等級
    - 例：

角色目前等級＝4，當角色經驗值(experience)＝4/4，角色等級上升一級＝>角色目前等級＝5，角色經驗值＝0/5，印出

Level up!!! You are level 5.

注意：

1. 下一次作業會以本次作業為基礎，增加更多的功能，因此請盡可能設計的好維護及好擴充
2. 本次作業需要現場demo及回答問題，時間約為五分鐘。
3. 嚴禁抄襲

評分標準：

(15%)現場demo時，是否能正確回答助教的問題

(15%)程式碼品質與結構

(5%)地圖是否用讀檔的方式加入RPG系統中

(8%)角色是否能正常的在地圖上移動

(2%)傳送點是否正常運作

(2%)升級是否正常運作

(8%)戰鬥系統是否正常運作

(15%)Map類別的屬性及方法是否正確

(15%)RpgCreature類別的屬性及方法是否正確

(15%)RpgPerson類別的屬性及方法是否正確

完成後請點選版面配置->雙欄，版面配置->窄邊界，字體大小12，中文字體為標楷體，英文字形為Times New Roman，將本次作業所有副檔名為cpp(或cc)的程式碼貼到以下的”答”下方，印出來繳交給助教。

答：

//Map.cpp

#include "Map.h"

Map::Map(){}

Map::Map(std::string \_name, std::string \_nextMapName, int \_initialPositionX, int \_initialPositionY, int \_width, int \_height, char \_mapData[11][21]) : name(\_name),nextMapName(\_nextMapName),initialPositionX(\_initialPositionX),initialPositionY(\_initialPositionY),width(\_width),height(\_height){

for(int i=0; i<10; i++){

for(int j=0; j<20; j++){

mapData[i][j] = \_mapData[i][j];

}

}

}

Map::~Map(){}

void Map::setMap(std::string \_name, std::string \_nextMapName, int \_initialPositionX, int \_initialPositionY, int \_width, int \_height, char \_mapData[11][21]){

this->name = \_name;

nextMapName = \_nextMapName;

initialPositionX = \_initialPositionX;

initialPositionY = \_initialPositionY;

width = \_width;

height = \_height;

for(int i=0; i<11; i++){

for(int j=0; j<21; j++){

mapData[i][j] = \_mapData[i][j];

}

}

}

std::string Map::getName(){

return name;

}

std::string Map::getNextMapName(){

return nextMapName;

}

int Map::getWidth(){

return width;

}

int Map::getHeight(){

return height;

}

int Map::getinitialPositionX(){

return initialPositionX;

}

int Map::getinitialPositionY(){

return initialPositionY;

}

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

return mapData[x][y];

}

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

if(getMapData(x,y)!='1' && (x<20 && x>=0) && (y<20 && y>=0))

return true;

else

return false;

}

void Map::printMap(int playerPositionX, int playerPositionY){

std::cout <<std::endl;

for(int i=0; i<10; i++){

for(int j=0; j<20; j++){

if(i==playerPositionX && j==playerPositionY){

std::cout <<'#';

}

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

std::cout <<' ';

}

else{

std::cout <<mapData[i][j];

}

}

std::cout <<std::endl;

}

}

void Map::addPlayer(RpgPerson& \_player){

this->player = &\_player;

}

void Map::removePlayer(){

this->player = NULL;

}

//Map.h

#ifndef MAP\_H\_INCLUDED

#define MAP\_H\_INCLUDED

#include "RpgPerson.h"

class Map{

public:

Map();

Map(std::string \_name, std::string \_nextMapName, int \_initialPositionX, int \_initialPositionY, int \_width, int \_height, char \_mapData[11][21]);

~Map();

void setMap(std::string \_name, std::string \_nextMapName, int \_initialPositionX, int \_initialPositionY, int \_width, int \_height, char \_mapData[11][21]);

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(int playerPositionX, int playerPositionY);

void addPlayer(RpgPerson& \_player);

void removePlayer();

private:

std::string name;

std::string nextMapName;

int initialPositionX;

int initialPositionY;

int width;

int height;

char mapData[11][21];

RpgPerson\* player;

};

#endif // MAP\_H\_INCLUDED

//Monster.h

#ifndef MONSTER\_H\_INCLUDED

#define MONSTER\_H\_INCLUDED

#include "RpgCreature.h"

class Monster: public RpgCreature{

public:

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

private:

};

#endif // MONSTER\_H\_INCLUDED

//Monster.cpp

#include "Monster.h"

Monster::Monster(std::string \_name, int \_maxHp, int \_strength, int \_defence):RpgCreature(\_name,\_maxHp,\_strength,\_defence){}

//RpgCreature.cpp

#include "RpgCreature.h"

RpgCreature::RpgCreature(){}

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

this->name = name;

}

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

this->name = name;

this->maxHp = maxHp;

this->strength = strength;

this->defence = defence;

currentHp = maxHp;

}

std::string RpgCreature::getName(){

return name;

}

int RpgCreature::getMaxHp(){

return maxHp;

}

int RpgCreature::getCurrentHp(){

return currentHp;

}

int RpgCreature::getStrength(){

return strength;

}

int RpgCreature::getDefence(){

return defence;

}

bool RpgCreature::isDead(){

if(currentHp<0 || currentHp==0)

return 1;

else

return 0;

}

void RpgCreature::loseHp(int value){

currentHp-=value;

}

void RpgCreature::recoverHp(){

currentHp = maxHp;

}

void RpgCreature::attack(RpgCreature& enemy){

int damage = strength - enemy.getDefence();

enemy.loseHp(damage);

}

void RpgCreature::setCurrentMap(int mapIndex){

currentMap = mapIndex;

}

int RpgCreature::getCurrentMap(){

return currentMap;

}

//RpgCreature.h

#ifndef RPGCREATURE\_H\_INCLUDED

#define RPGCREATURE\_H\_INCLUDED

#include <iostream>

#include <time.h>

#include <stdlib.h>

class RpgCreature{

public:

RpgCreature();

RpgCreature(std::string name);

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

std::string getName();

int getMaxHp();

int getCurrentHp();

int getStrength();

int getDefence();

bool isDead();

void loseHp(int value);

void recoverHp();

void attack(RpgCreature& enemy);

void setCurrentMap(int mapIndex);

int getCurrentMap();

protected:

std::string name;

int maxHp;

int currentHp;

int strength;

int defence;

int currentMap;

};

#endif // RPGCREATURE\_H\_INCLUDED

//RpgPerson.h

#ifndef RPGPERSON\_H\_INCLUDED

#define RPGPERSON\_H\_INCLUDED

#include <vector>

#include "RpgCreature.h"

class RpgPerson: public RpgCreature{

public:

RpgPerson(std::string \_name, int \_maxHp, int \_strength, int \_defence, int \_x, int \_y, int \_currentMap);

int getX();

int getY();

void setPosition(int x, int y);

void move(int xMove, int yMove);

void raiseExperience(int value);

bool checkLevelUp();

void levelUp();

void statusUp();

void printStatus();

int getLevel();

private:

int level;

int experience;

int x;

int y;

};

#endif // RPGPERSON\_H\_INCLUDED

//RpgPerson.cpp

#include "RpgPerson.h"

RpgPerson::RpgPerson(std::string \_name, int \_maxHp, int \_strength, int \_defence, int \_x, int \_y,int \_currentMap):RpgCreature(\_name,\_maxHp,\_strength,\_defence){

level=1;

experience=0;

x = \_x;

y = \_y;

currentMap = \_currentMap;

}

int RpgPerson::getX(){

return x;

}

int RpgPerson::getY(){

return y;

}

void RpgPerson::setPosition(int \_x, int \_y){

x = \_x;

y = \_y;

}

void RpgPerson::move(int xMove, int yMove){

x+=xMove;

y+=yMove;

}

void RpgPerson::raiseExperience(int value){

experience+=value;

}

bool RpgPerson::checkLevelUp(){

if(experience>=level)

return 1;

else

return 0;

}

void RpgPerson::levelUp(){

experience-=level;

level++;

}

void RpgPerson::statusUp(){

maxHp+=5;

strength+=1;

defence+=1;

}

void RpgPerson::printStatus(){

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

std::cout <<"Lv: " <<level <<std::endl;

std::cout <<"Hp: " <<maxHp <<std::endl;

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

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

std::cout <<"Exp: " <<experience <<"/" <<level <<std::endl;

}

int RpgPerson::getLevel(){

return level;

}

//rpg.cpp(main)

#include<iostream>

#include<fstream>

#include "Map.h"

#include "RpgCreature.h"

#include "RpgPerson.h"

#include "Monster.h"

#include<vector>

#include<string>

#include<cstdlib>

#include<ctime>

using namespace std;

void printOperationNotice(){

cout <<"\n\*\*\*\*\*\n" <<"Use w/s/a/d to move up/down/left/right.\n" <<"Enter i to show your status.\n" <<"Enter exit to leave game.\n" <<"\*\*\*\*\*\n";

}

void fight(RpgCreature &attacker, RpgCreature &victim){

while(!attacker.isDead() && !victim.isDead()){

attacker.attack(victim);

cout <<attacker.getName()<<" attacked "<<victim.getName()<<", caused " <<attacker.getStrength() <<" damages\n";

victim.attack(attacker);

cout <<victim.getName()<<" attacked "<<attacker.getName()<<", caused " <<victim.getStrength() <<" damages\n";

}

}

void playerWin(RpgPerson &player, Monster monster){

int exp = rand()%2+1;

cout <<"You beat a " <<monster.getName() <<",got " <<exp <<" exp.\n";

player.raiseExperience(exp);

if(player.checkLevelUp()){

player.levelUp();

cout <<"Level up!!!You are now level " <<player.getLevel() <<".\n";

player.statusUp();

cout <<"HP recovered.\n";

player.recoverHp();

}

}

int walk(Map map, RpgPerson &player, int xMove, int yMove){

if(map.canMove(player.getX()+xMove,player.getY()+yMove)){

player.move(xMove,yMove);

return 1;

}

else{

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

return 0;

}

}

int main(){

cout <<" -----------------------------------------\n" <<"| Welcome to oop assignment2 rpg. |\n" <<" ------------------------------------------\n";

ifstream fin("map.txt");

if(!fin){

cout <<"Failed to load map data\n" <<"The game is off\n";

return 0;

}

Map map[3]; //宣告map

for(int i=0; i<3; i++){ //吃地圖

string \_name,\_nextMapName,tmp;

int \_initialPositionX,\_initialPositionY,\_width,\_height;

char \_mapData[21][21];

getline(fin,\_name);

getline(fin,\_nextMapName);

fin >> \_initialPositionY;

fin >> \_initialPositionX;

fin >> \_width;

fin >> \_height;

for(int p=0; p<10; p++){

for(int q=0; q<20; q++){

fin.get(\_mapData[p][q]);

if(\_mapData[p][q]!='0' && \_mapData[p][q]!='1' && \_mapData[p][q]!='@')

q--;

}

}

getline(fin,tmp);

getline(fin,tmp);

map[i].setMap(\_name,\_nextMapName,\_initialPositionX,\_initialPositionY,\_width,\_height,\_mapData);

} // 結束吃地圖

fin.close();

cout <<"Enter your name:";//吃名字

string \_name,command,currentMap=map[0].getName();

cin >> \_name;

RpgPerson player1(\_name,25,7,2,map[0].getinitialPositionX(),map[0].getinitialPositionY(),0); //player實作

map[0].addPlayer(player1);

map[0].printMap(player1.getX(),player1.getY());//輸入名字後第一次印地圖

printOperationNotice();//輸入名字後第一次印指令

srand((unsigned)time(NULL));

while(cin >> command){

if(command == "exit")

return 0;

else if(command == "i"){

player1.printStatus();

cout <<"Map: " <<map[player1.getCurrentMap()].getName() <<endl<<endl;

}

else if(command =="w" || command =="s" || command == "a" || command == "d"){

if(command == "w"){

if(!walk(map[player1.getCurrentMap()],player1,-1,0))

continue;

}

else if(command == "s"){

if(!walk(map[player1.getCurrentMap()],player1,1,0))

continue;

}

else if(command == "a"){

if(!walk(map[player1.getCurrentMap()],player1,0,-1))

continue;

}

else{

if(!walk(map[player1.getCurrentMap()],player1,0,1))

continue;

}

if(!(rand()%4)){ //fight!

Monster monster("Slime",20,6,3);

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

cout<<"-----\n"<<"Enter f to fight,e to escape.\n"<<"-----\n";

string fight\_command;

while(cin>>fight\_command){

if(fight\_command == "f"){

fight(player1,monster);

if(player1.isDead()){

cout <<"You lose!Game over\n";

return 0;

}

else

playerWin(player1,monster);

break;

}

else if(fight\_command == "e"){

if( rand()%10<3 )

cout <<"You successfully escaped from the fight!\n";

else{

cout <<"Escape failed!!!\n";

fight(monster,player1);

if(player1.isDead()){

cout <<"You lose!Game over\n";

return 0;

}

else

playerWin(player1,monster);

}

break;

}

else{

cout <<"Wrong input!\n";

}

}

}

if(map[player1.getCurrentMap()].getMapData(player1.getX(),player1.getY()) == '@'){

for(int i=0; i<3; i++){

if(map[player1.getCurrentMap()].getNextMapName() == map[i].getName()){

cout <<"map i=" <<i <<endl;

map[player1.getCurrentMap()].removePlayer();

player1.setCurrentMap(i);

map[i].addPlayer(player1);

player1.setPosition(map[player1.getCurrentMap()].getinitialPositionX(),map[player1.getCurrentMap()].getinitialPositionY());

break;

}

}

}

}

else{

cout <<"Entered the wrong command! Please try again\n";

continue;

}

map[player1.getCurrentMap()].printMap(player1.getX(),player1.getY());

printOperationNotice();

}

}