**Jacob Trubey**

**CSC-17A**

**Section 43950**

**Spring 2015**

**Project 1**

**Mercenary Mission**

**Introduction: Mercenary Mission**

Mercenary Mission is a turn-based/text-based strategy game involving money and mercenary warfare. You start with the options of viewing a tutorial on the game, starting the game, or viewing the list of high scores.

**Viewing The Tutorial**

Viewing the tutorial is much like viewing this write-up, it breaks down the details of the game by section and tells the user how to play.

**Starting The Game**

*Choosing an amount of mercenaries to train*

When starting the game, you are given $200,000 which you start using by picking an amount of mercenaries to train with. This is the only time during the game that you will have this option, so you will have to apply some thought or strategy here.

*Increasing Abilities/Resuscitating Mercenaries*

Whatever funds you have left over from training your team of mercenaries can then be used to increase each of your mercenaries’ abilities. These abilities include ‘attack’, which is a determining factor in dealing a lethal attack to an enemy (based on the idea of good equipment and the ability to use iteffectively), and ‘defense’, which is a determining factor on whether or not an enemy can deal a lethal attack to your mercenary (also based on the idea of equipment and personal skill). Increasing these statistics costs money, but each have their own perks in a battle sequence, and it is up to you to determine which is more worth your money for given circumstances. Another option in the ability increasing function is the option to resuscitate any injured mercenaries, but this option is naturally only useable if you have injured mercenaries.

*Choosing a job*

There are 1 of 3 types of jobs to choose from, in descending order of difficulty from lowest to highest. Each job has a risk rating that demonstrates this difficulty (low, moderate, high). Beginning players will be most likely unable to survive the moderate difficulty level, let alone the high level, so it would be a good idea to spend some time taking low difficulty level jobs of escorting dignitaries and using your monetary rewards to increase your mercenaries abilities until you feel that they are ready for the more difficult mission.

*Engaging A Mission/Battle Sequence*

The battle sequence starts when a mission is accepted and engaged. This sequence involves a random generation of enemies (The possible amounts depending on the difficultly of the mission) and then more random generation determines which of your mercenaries are personally engaged by an enemy, and even further random generation uses variations on your mercenaries base attack score, vs. the enemy’s randomly generated defense, and the enemy’s randomly generated attack score, based on your mercenaries base defense. A successful hit is all that is needed to defeat either the enemy or your mercenary (keeping in mind unsuccessful attacks are evaded/blocked by body armor, or considered negligible, and that successful attacks or those inflicted by firearms or other equally lethal weapons with enough damage to defeat the opponent). After the two attack each other simultaneously, it is then determined how much of an attack score was given by both opponents and who, if any, was defeated as a result. If all of your mercenaries are defeated before defeating all of the enemies, you fail the mission with no reward. If all of the enemies are defeated before all of your mercenaries are, then the mission is a success and your reward is earned. If you defeat the last enemy while that enemy simultaneously defeats your last mercenary, you successfully complete the mission and receive the reward, but are then unable to accept any other jobs until you resuscitate one of your mercenaries. Should you have insufficient funds to resuscitate a mercenary and all of your mercenaries are injured, it is up to you to quit the game from the “Job List” menu as your only other option is to continue to go through the other menu options to no avail.

*Saving Your Score*

After quitting the game you have the option to save your score. If you choose yes, you are then prompted to enter your initials which will precede the high score you earned from playing the game. Your high score is based on money earned, services purchased, and mission ready (non-injured) mercenaries upon quitting. This score will then be saved to a file named “score”, which will keep a log of all previously saved scores on your computer.

**Viewing High Scores**

Selecting the option to “View High Scores” allows you to view the file called “scores” which contains the initials and accompanying high scores of previously played games.

**Variables Used**

|  |  |  |
| --- | --- | --- |
| **Type** | **Variable Name** | **Purpose** |
| int | inN | Initial menu choice |
| int | funds | Money earned in game |
| int | numMrc | Initial number of mercenaries chosen |
| struct Mercenary | mercenaries | Array of structures, each for individual mercenaries and their statistics (name, attack, defense, stable) |
| bool | misRed | Determines if any of your mercenaries are mission ready |
| const int | SIZE | Size of character array for player’s initials |
| char | Initials | Saves player’s initials for high score |
| int | highScore | Saves player’s high score |
| char | incChc | Choice to increase abilities |
| int | mrcChc | Choose which mercenary to improve |
| char | conChc | Confirm previous choice |
| char | ablChc | Choose which ability to improve |
| int | jobChc | Choose which job to accept |
| char | qitChc | Choice to quit the game |
| char | savChc | Choice to save high score |
| string | name | Name of mercenary in structure |
| int | attack | attack of mercenary in structure |
| int | defense | defense of mercenary in structure |
| bool | stable | Health stability of mercenary in structure |
| unsigned | seed | Random number generator |
| int | numEnm | Number of enemies encountered |
| int | mrcEng | Mercenary engaged by enemy |
| int | mrcHit | Mercenary’s generated Hit |
| int | enmHit | Enemy’s generated Hit |
| int | enmAtk | Enemy’s base attack score |
| int | enmDfn | Enemy’s base defense score |

**Project Code: Structure Header File “mercenary.h”**

/\*

\* File: mercenary.h

\* Author: Jacob Trubey

\*

\* Created on May 4, 2015, 7:11 PM

\*/

#include <iostream>

#include <cstdlib>

#include <string>

using namespace std;

#ifndef MERCENARY\_H

#define MERCENARY\_H

struct Mercenary{

string name;

int attack;

int defense;

bool stable;

};

#endif /\* MERCENARY\_H \*/

**Project Code: Main Source File “main.cpp”**

/\*

\* File: main.cpp

\* Author: Jacob Trubey

\*

\* Created on April 23, 2015, 4:20 PM

\*/

//Library includes Here!!!

#include "mercenary.h"

#include <iostream>

#include <iomanip>

#include <string>

#include <cstdlib>

#include <ctime>

#include <fstream>

using namespace std;

//Global Constants Here!!!

//Function Prototypes Here!!!

void Menu();

int getN();

void def(int);

void viewTutorial();

void startGame();

void viewScores();

void amtMrc(int \*, int \*);

void trainMrc(int \*, int \*, Mercenary \*);

void incAbl(int \*, int \*, Mercenary \*);

void chooseJob(int \*, int \*, bool \*, Mercenary \*);

void startJob1(int \*, int \*, bool \*, Mercenary \*);

void startJob2(int \*, int \*, bool \*, Mercenary \*);

void startJob3(int \*, int \*, bool \*, Mercenary \*);

void saveScore(int \*, int \*, Mercenary \*);

//Begin Execution Here!!!

int main(int argv,char \*argc[]){

int inN;

do{

Menu();

inN=getN();

switch(inN){

case 1: viewTutorial();break;

case 2: startGame();break;

case 3: viewScores();break;

default: def(inN);}

}while(inN>=1&&inN<=3);

return 0;

}

void Menu(){

cout<<"Welcome to Mercenary Mission!"<<endl;

cout<<"(a turn based strategy game of)"<<endl;

cout<<"(money and hired mercenaries)"<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

cout<<"Please type the number associated with the option from the following menu list."<<endl<<endl;

cout<<"1) View Tutorial"<<endl;

cout<<"2) Start Game"<<endl;

cout<<"3) View High Scores"<<endl;

cout<<"(type anything else to exit)"<<endl<<endl;

}

int getN(){

int inN;

cin>>inN;

cout<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

return inN;

}

void viewTutorial(){

cout<<"Tutorial"<<endl<<endl;

}

void startGame(){

int funds = 200000;

int numMrc;

Mercenary \*mercenaries;

bool misRed = true;

amtMrc(&funds, &numMrc);

mercenaries = new Mercenary[numMrc];

trainMrc(&funds, &numMrc, mercenaries);

incAbl(&funds, &numMrc, mercenaries);

chooseJob(&funds, &numMrc, &misRed, mercenaries);

saveScore(&funds,&numMrc, mercenaries);

cout<<"Thank you for playing."<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

delete [] mercenaries;

mercenaries = 0;

}

void viewScores(){

const int SIZE = 3;

char initials[SIZE] = {' ',' ',' '};

int highScore = 0;

fstream file;

file.open("scores.txt", ios::in | ios::binary);

cout<<"High Scores"<<endl<<endl;

file.read(initials, sizeof(initials));

file.read(reinterpret\_cast<char \*>(&highScore), sizeof(highScore));

while (!file.eof()){

for(int count = 0; count < SIZE; count++)

cout<<initials[count];

cout<<" "<<highScore<<endl;

file.read(initials, sizeof(initials));

file.read(reinterpret\_cast<char \*>(&highScore), sizeof(highScore));

}

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

file.close();

}

void amtMrc(int \*funds, int \*numMrc){

cout<<"Your current funds are $"<<\*funds<<"."<<endl;

cout<<"(each mercenary costs $20000 to train)"<<endl;

cout<<"How many mercenaries do you want to train?"<<endl;

cin>>\*numMrc;

cout<<endl;

if (\*numMrc<1 || \*numMrc>10){

do{

cout<<"That is an invalid amount."<<endl;

cout<<"Please enter a valid amount of mercenaries to train."<<endl;

cin>>\*numMrc;

cout<<endl;

}while(\*numMrc<1 || \*numMrc>10);

}

}

void trainMrc(int \*funds, int \*numMrc, Mercenary \*mercenaries){

cin.ignore();

for(int count = 0; count < \*numMrc; count++){

cout<<"Please enter mercenary number "<<count+1<<"'s name."<<endl;

getline(cin, (\*(mercenaries + count)).name);

cout<<endl;

(\*(mercenaries + count)).defense = 5;

(\*(mercenaries + count)).attack = 3;

(\*(mercenaries + count)).stable = true;

}

cout<<"Training "<<\*numMrc<<" mercenaries..."<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

cout<<"Your team of mercenaries is as follows:"<<endl<<endl;

for(int count = 0; count < \*numMrc; count++){

cout<<(\*(mercenaries + count)).name<<endl;

cout<<"defense: "<<(\*(mercenaries + count)).defense<<endl;

cout<<"attack: "<<(\*(mercenaries + count)).attack<<endl;

if ((\*(mercenaries + count)).stable == true)

cout<<"health status: mission ready"<<endl<<endl;

else

cout<<"health status: injured"<<endl<<endl;

}

\*funds = \*funds - 20000\*(\*numMrc);

cout<<"Your current funds are $"<<\*funds<<"."<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

void incAbl(int \*funds, int \*numMrc, Mercenary \*mercenaries){

char incChc; //choice to increase abilities of a mercenary

int mrcChc; //choice of mercenary to increase abilities of

char conChc; //confirm choice of which mercenary's abilities to increase

char ablChc; //choice of ability to increase

cout<<"Would you like to improve the equipment of any of your mercenaries or resuscitate them?"<<endl;

cout<<"(improving defensive equipment increases a mercenary's defense and costs $20000 per mercenary)"<<endl;

cout<<"{improving offensive equipment increases a mercenary's attack and costs $50000 per mercenary)"<<endl;

cout<<"(resuscitation changes a mercenary's status from injured to mission ready and costs $100000 per mercenary)"<<endl;

cout<<"Please enter Y for yes, or anything else for no."<<endl;

cin>>incChc;

cout<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

while(incChc == 'Y'){

cout<<"Which mercenary would you like to equip or resuscitate?"<<endl;

cout<<"Enter number from list associated with mercenary."<<endl<<endl;

for(int count = 0; count < \*numMrc; count++){

cout<<count + 1<<") "<<(\*(mercenaries + count)).name<<endl;

cout<<"defense: "<<(\*(mercenaries + count)).defense<<endl;

cout<<"attack: "<<(\*(mercenaries + count)).attack<<endl;

if ((\*(mercenaries + count)).stable == true)

cout<<"health status: mission ready"<<endl<<endl;

else

cout<<"health status: injured"<<endl<<endl;

}

cin>>mrcChc;

cout<<endl;

if (mrcChc<1 || mrcChc>(\*numMrc)){

do{

cout<<"That is an invalid entry."<<endl;

cout<<"Please enter a valid entry from the list of mercenaries."<<endl;

cin>>mrcChc;

cout<<endl;

}while(mrcChc<1 || mrcChc>(\*numMrc));

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

cout<<"You've chosen to increase the abilities of "<<(\*(mercenaries + mrcChc - 1)).name<<". Is this correct?"<<endl;

cout<<"Please enter Y for yes, or anything else for no."<<endl;

cin>>conChc;

cout<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

while(conChc == 'Y'){

cout<<"What would you like to improve for "<<(\*(mercenaries + mrcChc - 1)).name<<"?"<<endl;

cout<<"(please enter the letter associated with the option from the list, or anything else to go back)"<<endl<<endl;

cout<<"Your current funds are $"<<\*funds<<"."<<endl<<endl;

cout<<"D) Improve Defensive Equipment: $20000"<<endl;

cout<<"O) Improve Offensive Equipment: $50000"<<endl;

cout<<"R) Resuscitate: $100000"<<endl;

cin>>ablChc;

cout<<endl;

if (ablChc == 'D'){

if (\*funds < 20000){

cout<<"You have insufficient funds."<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

else{

(\*(mercenaries + mrcChc - 1)).defense++;

\*funds = \*funds - 20000;

cout<<(\*(mercenaries + mrcChc - 1)).name<<"'s defense is now "<<(\*(mercenaries + mrcChc - 1)).defense<<"."<<endl;

cout<<"Your current funds are $"<<\*funds<<"."<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

}

else if (ablChc == 'O'){

if (\*funds < 50000){

cout<<"You have insufficient funds."<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

else{

(\*(mercenaries + mrcChc - 1)).attack++;

\*funds = \*funds - 50000;

cout<<(\*(mercenaries + mrcChc - 1)).name<<"'s attack is now "<<(\*(mercenaries + mrcChc - 1)).attack<<"."<<endl;

cout<<"Your current funds are $"<<\*funds<<"."<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

}

else if (ablChc == 'R'){

if ((\*(mercenaries + mrcChc - 1)).stable == true){

cout<<(\*(mercenaries + mrcChc - 1)).name<<" is already mission ready."<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

else if (\*funds < 100000){

cout<<"You have insufficient funds."<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

else{

(\*(mercenaries + mrcChc - 1)).stable = true;

\*funds = \*funds - 100000;

cout<<(\*(mercenaries + mrcChc - 1)).name<<" has been resuscitated back to stable health."<<endl;

cout<<"Your current funds are $"<<\*funds<<"."<<endl<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

}

conChc = 'N';

}

cout<<"Would you like to improve the equipment of any of your mercenaries or resuscitate them?"<<endl;

cout<<"{improving defensive equipment increases a mercenary's defense and costs $20000 per mercenary)"<<endl;

cout<<"(improving offensive equipment increases a mercenary's attack and costs $50000 per mercenary)"<<endl;

cout<<"(resuscitation changes a mercenary's status from injured to mission ready and costs $100000 per mercenary)"<<endl;

cout<<"Please enter Y for yes, or anything else for no."<<endl;

cin>>incChc;

cout<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

}

void chooseJob(int \*funds, int \*numMrc, bool \*misRed, Mercenary \*mercenaries){

int jobChc; //job to be chosen for mercenary team

char conChc; //confirm choice of job

char qitChc = 'Y';

do{

cout<<"Please choose from the list of available mercenary missions."<<endl;

cout<<"(enter the number associated with the option from the list)"<<endl;

cout<<"(entering any number not on the list will exit the job menu)"<<endl<<endl;

cout<<"1) Escort Dignitary."<<endl;

cout<<"Risk Factor: low"<<endl;

cout<<"Payment: $20000."<<endl<<endl;

cout<<"2) Apprehend Crime Lord."<<endl;

cout<<"Risk Factor: medium"<<endl;

cout<<"Payment: $100000."<<endl<<endl;

cout<<"3) Infiltrate Foreign Military Complex."<<endl;

cout<<"Risk Factor: high"<<endl;

cout<<"Payment: $1000000."<<endl<<endl;

cin>>jobChc;

cout<<endl;

while(jobChc > 0 && jobChc < 4){

if(jobChc == 1){

cout<<"Escort Dignitary: $20000"<<endl<<endl;

cout<<"A wealthy dignitary is looking for security for hire."<<endl;

cout<<"As he is valuable enough to be concerned with his safety,"<<endl;

cout<<"but not in a governmental position to qualify for"<<endl;

cout<<"formal military protection, he has turned to our"<<endl;

cout<<"employers for assistance on the matter. Your job is"<<endl;

cout<<"to escort the client to his destination unharmed. Our"<<endl;

cout<<"intel has determined the risk factor of this mission to"<<endl;

cout<<"be of a low rating, though some conflict may be likely."<<endl<<endl;

cout<<"Would you like to take this job?"<<endl;

cout<<"(please enter Y for yes, or anything else to decline)"<<endl;

cin>>conChc;

cout<<endl;

if(conChc == 'Y'){

\*misRed = false; //mission ready boolean

for(int count = 0; count < \*numMrc; count++){

if ((\*(mercenaries + count)).stable == true)

\*misRed = true;

}

if(\*misRed == false)

cout<<"You have no mission ready mercenaries. You cannot accept any jobs at this time."<<endl<<endl;

else{

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

startJob1(funds, numMrc, misRed, mercenaries);

}

}

incAbl(funds, numMrc, mercenaries);

}

else if(jobChc == 2){

cout<<"Apprehend Crime Lord: $100000"<<endl<<endl;

cout<<"An organized crime lord has been under surveillance of"<<endl;

cout<<"federal law enforcement for some time, but due to legal"<<endl;

cout<<"constraints, they have not been able to get close enough"<<endl;

cout<<"to him to expose his activity and try him for his crimes."<<endl;

cout<<"This is where you come in. Your mission is to enter his"<<endl;

cout<<"compound where you will find the necessary information to"<<endl;

cout<<"expose and apprehend him. Our intel has determined the"<<endl;

cout<<"risk factor for this mission to be of a moderate rating."<<endl;

cout<<"As the compound is well guarded, conflict is more than likely."<<endl<<endl;

cout<<"Would you like to take this job?"<<endl;

cout<<"(please choose Y for yes, or anything else to decline)"<<endl;

cin>>conChc;

cout<<endl;

if(conChc == 'Y'){

\*misRed = false; //mission ready boolean

for(int count = 0; count < \*numMrc; count++){

if ((\*(mercenaries + count)).stable == true)

\*misRed = true;

}

if(\*misRed == false)

cout<<"You have no mission ready mercenaries. You cannot accept any jobs at this time."<<endl<<endl;

else{

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

startJob2(funds, numMrc, misRed, mercenaries);

}

}

incAbl(funds, numMrc, mercenaries);

}

else if(jobChc == 3){

cout<<"Infiltrate Foreign Military Complex: $1000000"<<endl<<endl;

cout<<"Federal Intel has determined that a foreign military power"<<endl;

cout<<"threatens the safety of the nation, but foreign policy"<<endl;

cout<<"currently prohibits government forces from intervening."<<endl;

cout<<"Our employers have been called upon to eliminate this threat."<<endl;

cout<<"Your mission is to infiltrate the foreign military complex"<<endl;

cout<<"in question and neutralize this threat. Our intel has"<<endl;

cout<<"determined that the risk factor for this mission is high, due"<<endl;

cout<<"to heavily armed military forces, making conflict almost certain."<<endl<<endl;

cout<<"Would you like to take this job?"<<endl;

cout<<"(please choose Y for yes, or anything else to decline)"<<endl;

cin>>conChc;

cout<<endl;

if(conChc == 'Y'){

\*misRed = false; //mission ready boolean

for(int count = 0; count < \*numMrc; count++){

if ((\*(mercenaries + count)).stable == true)

\*misRed = true;

}

if(\*misRed == false)

cout<<"You have no mission ready mercenaries. You cannot accept any jobs at this time."<<endl<<endl;

else{

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

startJob3(funds, numMrc, misRed, mercenaries);

}

}

incAbl(funds, numMrc, mercenaries);

}

cout<<"Please choose from the list of available mercenary missions."<<endl;

cout<<"(enter the number associated with the option from the list)"<<endl;

cout<<"(entering a number not on the list will exit the job menu)"<<endl<<endl;

cout<<"1) Escort Dignitary."<<endl;

cout<<"Risk Factor: low"<<endl;

cout<<"Payment: $20000."<<endl<<endl;

cout<<"2) Apprehend Crime Lord."<<endl;

cout<<"Risk Factor: medium"<<endl;

cout<<"Payment: $100000."<<endl<<endl;

cout<<"3) Infiltrate Foreign Military Complex."<<endl;

cout<<"Risk Factor: high"<<endl;

cout<<"Payment: $1000000."<<endl<<endl;

cin>>jobChc;

cout<<endl;

}

cout<<"Do you wish to exit the game?"<<endl;

cout<<"(please enter Y for yes, or anything else to decline)"<<endl;

cin>>qitChc;

cout<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}while(qitChc != 'Y');

}

void startJob1(int \*funds, int \*numMrc, bool \*misRed, Mercenary \*mercenaries){

cout<<"Job accepted. Engaging mission..."<<endl<<endl;

unsigned seed = time(0);

srand(seed);

int numEnm;

int mrcEng;

int mrcHit;

int enmHit;

int enmAtk;

int enmDfn;

numEnm = 0 + rand() % 6;

cout<<"Your mercenaries encounter "<<numEnm<<" hired thugs..."<<endl<<endl;

cin.ignore();

cout<<"(please press enter to continue)"<<endl;

cin.get();

cout<<endl;

if(numEnm == 0){

cout<<"The dignitary made it to his destination unharmed. Mission accomplished!"<<endl<<endl;

\*funds = \*funds + 20000;

cout<<"Your funds are now $"<<\*funds<<"."<<endl<<endl;

}

while ((numEnm > 0) && (\*misRed == true)){

do{ //check to make sure mercenary engaged is not injured

mrcEng = (0 + rand () % (\*numMrc));

}while((\*(mercenaries + mrcEng)).stable == false);

enmAtk = 1 + rand () % 3;

enmDfn = 1 + rand() % 10;

cout<<"A hired thug attacks "<<(\*(mercenaries + mrcEng)).name<<"!"<<endl;

mrcHit = (0 + rand() % 4)\*((\*(mercenaries + mrcEng)).attack);

enmHit = (0 + rand() % 4)\*(enmAtk);

cout<<(\*(mercenaries + mrcEng)).name<<" deals "<<mrcHit<<" points of damage to the thug..."<<endl;

cout<<"The thug deals "<<enmHit<<" points of damage to "<<(\*(mercenaries + mrcEng)).name<<"..."<<endl;

if (mrcHit > enmDfn){

numEnm = numEnm - 1;

cout<<(\*(mercenaries + mrcEng)).name<<" defeated the hired thug!"<<endl;

}

if (enmHit > (\*(mercenaries + mrcEng)).defense){

(\*(mercenaries + mrcEng)).stable = false;

cout<<(\*(mercenaries + mrcEng)).name<<" is now injured and won't be mission ready until resuscitated!"<<endl<<endl;

}

cout<<numEnm<<" thugs remain..."<<endl<<endl;

cout<<"(please press enter to continue)"<<endl;

cin.get();

cout<<endl;

\*misRed = false; //mission ready boolean

for(int count = 0; count < \*numMrc; count++){

if ((\*(mercenaries + count)).stable == true)

\*misRed = true;

}

if (\*misRed == false && numEnm == 0){

cout<<"All of the hired thugs have been defeated, but all of your mercenaries are injured."<<endl;

cout<<"You have completed the mission and the dignitary made it to his destination unharmed,"<<endl;

cout<<"but are unable to accept any other jobs until one of your mercenaries are resuscitated."<<endl<<endl;

\*funds = \*funds + 20000;

cout<<"Your funds are now $"<<\*funds<<"."<<endl<<endl;

}

else if(\*misRed == false)

cout<<"All of your mercenaries are injured. You have failed the mission..."<<endl<<endl;

else if(numEnm == 0){

cout<<"All of the hired thugs have been defeated! The dignitary has made it"<<endl;

cout<<"to his destination unharmed. Mission accomplished!"<<endl<<endl;

\*funds = \*funds + 20000;

cout<<"Your funds are now $"<<\*funds<<"."<<endl<<endl;

}

}

cout<<"(please press enter to continue)"<<endl;

cin.get();

cout<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

void startJob2(int \*funds, int \*numMrc, bool \*misRed, Mercenary \*mercenaries){

cout<<"Job accepted. Engaging mission..."<<endl<<endl;

unsigned seed = time(0);

srand(seed);

int numEnm;

int mrcEng;

int mrcHit;

int enmHit;

int enmAtk;

int enmDfn;

numEnm = 0 + rand() % 11;

cout<<"Your mercenaries encounter "<<numEnm<<" compound guards..."<<endl<<endl;

cin.ignore();

cout<<"(please press enter to continue)"<<endl;

cin.get();

cout<<endl;

if(numEnm == 0){

cout<<"The crime lord has been apprehended along with the"<<endl;

cout<<"evidence necessary for his incarceration. Mission accomplished!"<<endl<<endl;

\*funds = \*funds + 50000;

cout<<"Your funds are now $"<<\*funds<<"."<<endl<<endl;

}

while ((numEnm > 0) && (\*misRed == true)){

do{ //check to make sure mercenary engaged is not injured

mrcEng = (0 + rand () % (\*numMrc));

}while((\*(mercenaries + mrcEng)).stable == false);

enmAtk = 1 + rand () % 5;

enmDfn = 1 + rand() % 20;

cout<<"A guard attacks "<<(\*(mercenaries + mrcEng)).name<<"!"<<endl;

mrcHit = (0 + rand() % 6)\*((\*(mercenaries + mrcEng)).attack);

enmHit = (0 + rand() % 6)\*(enmAtk);

cout<<(\*(mercenaries + mrcEng)).name<<" deals "<<mrcHit<<" points of damage to the guard..."<<endl;

cout<<"The guard deals "<<enmHit<<" points of damage to "<<(\*(mercenaries + mrcEng)).name<<"..."<<endl;

if (mrcHit > enmDfn){

numEnm = numEnm - 1;

cout<<(\*(mercenaries + mrcEng)).name<<" defeated the compound gaurd!"<<endl;

}

if (enmHit > (\*(mercenaries + mrcEng)).defense){

(\*(mercenaries + mrcEng)).stable = false;

cout<<(\*(mercenaries + mrcEng)).name<<" is now injured and won't be mission ready until resuscitated!"<<endl<<endl;

}

cout<<numEnm<<" guards remain..."<<endl<<endl;

cout<<"(please press enter to continue)"<<endl;

cin.get();

cout<<endl;

\*misRed = false; //mission ready boolean

for(int count = 0; count < \*numMrc; count++){

if ((\*(mercenaries + count)).stable == true)

\*misRed = true;

}

if (\*misRed == false && numEnm == 0){

cout<<"All of the compound guards have been defeated, but all of your mercenaries are injured."<<endl;

cout<<"You have completed the mission and the crime lord has been apprehended along with the,"<<endl;

cout<<"evidence necessary for his incarceration, but you are unable to accept any other jobs until"<<endl;

cout<<"one of your mercenaries are resuscitated."<<endl<<endl;

\*funds = \*funds + 50000;

cout<<"Your funds are now $"<<\*funds<<"."<<endl<<endl;

}

else if(\*misRed == false)

cout<<"All of your mercenaries are injured. You have failed the mission..."<<endl<<endl;

else if(numEnm == 0){

cout<<"All of the compound guards have been defeated! The crime lord has been apprehended"<<endl;

cout<<"along with the evidence necessary for his incarceration. Mission accomplished!"<<endl<<endl;

\*funds = \*funds + 50000;

cout<<"Your funds are now $"<<\*funds<<"."<<endl<<endl;

}

}

cout<<"(please press enter to continue)"<<endl;

cin.get();

cout<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

void startJob3(int \*funds, int \*numMrc, bool \*misRed, Mercenary \*mercenaries){

cout<<"Job accepted. Engaging mission..."<<endl<<endl;

unsigned seed = time(0);

srand(seed);

int numEnm;

int mrcEng;

int mrcHit;

int enmHit;

int enmAtk;

int enmDfn;

numEnm = 0 + rand() % 21;

cout<<"Your mercenaries encounter "<<numEnm<<" military soldiers..."<<endl<<endl;

cin.ignore();

cout<<"(please press enter to continue)"<<endl;

cin.get();

cout<<endl;

if(numEnm == 0){

cout<<"The military complex has been successfully infiltrated and"<<endl;

cout<<"its threat has been neutralized. Mission accomplished!"<<endl<<endl;

\*funds = \*funds + 100000;

cout<<"Your funds are now $"<<\*funds<<"."<<endl<<endl;

}

while ((numEnm > 0) && (\*misRed == true)){

do{ //check to make sure mercenary engaged is not injured

mrcEng = (0 + rand () % (\*numMrc));

}while((\*(mercenaries + mrcEng)).stable == false);

enmAtk = 1 + rand () % 10;

enmDfn = 1 + rand() % 40;

cout<<"A soldier attacks "<<(\*(mercenaries + mrcEng)).name<<"!"<<endl;

mrcHit = (0 + rand() % 11)\*((\*(mercenaries + mrcEng)).attack);

enmHit = (0 + rand() % 11)\*(enmAtk);

cout<<(\*(mercenaries + mrcEng)).name<<" deals "<<mrcHit<<" points of damage to the soldier..."<<endl;

cout<<"The soldier deals "<<enmHit<<" points of damage to "<<mercenaries[mrcEng].name<<"..."<<endl;

if (mrcHit > enmDfn){

numEnm = numEnm - 1;

cout<<(\*(mercenaries + mrcEng)).name<<" defeated the military soldier!"<<endl;

}

if (enmHit > (\*(mercenaries + mrcEng)).defense){

(\*(mercenaries + mrcEng)).stable = false;

cout<<(\*(mercenaries + mrcEng)).name<<" is now injured and won't be mission ready until resuscitated!"<<endl<<endl;

}

cout<<numEnm<<" soldiers remain..."<<endl<<endl;

cout<<"(please press enter to continue)"<<endl;

cin.get();

cout<<endl;

\*misRed = false; //mission ready boolean

for(int count = 0; count < \*numMrc; count++){

if ((\*(mercenaries + count)).stable == true)

\*misRed = true;

}

if (\*misRed == false && numEnm == 0){

cout<<"All of the military soldiers have been defeated, but all of your mercenaries are injured."<<endl;

cout<<"You have completed the mission and the military complex has been infiltrated and its threat has been"<<endl;

cout<<"neutralized, but you are unable to accept any other jobs until one of your mercenaries are resuscitated."<<endl<<endl;

\*funds = \*funds + 100000;

cout<<"Your funds are now $"<<\*funds<<"."<<endl<<endl;

}

else if(\*misRed == false)

cout<<"All of your mercenaries are injured. You have failed the mission..."<<endl<<endl;

else if(numEnm == 0){

cout<<"All of the military soldiers have been defeated! The military complex has been"<<endl;

cout<<"successfully infiltrated and its threat has been neutralized. Mission accomplished!"<<endl<<endl;

\*funds = \*funds + 100000;

cout<<"Your funds are now $"<<\*funds<<"."<<endl<<endl;

}

}

cout<<"(please press enter to continue)"<<endl;

cin.get();

cout<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

}

void saveScore(int \*funds, int \*numMrc, Mercenary \*mercenaries){

char savChc = 'Y';

const int SIZE = 3;

char initials[SIZE] = {' ',' ',' '};

int highScore = 0;

fstream file;

file.open("scores.txt", ios::out | ios::binary | ios::app);

cout<<"Would you like to save your score?"<<endl;

cout<<"(please enter Y for yes, or anything else to decline)"<<endl;

cin>>savChc;

cout<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

if(savChc == 'Y'){

cout<<"(please enter your initials to be saved with your high score)"<<endl;

for(int count = 0; count < SIZE; count++)

cin>>initials[count];

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

cout<<endl<<endl;

for(int count = 0; count < \*numMrc; count++){

highScore = highScore + 20000\*((\*(mercenaries + count)).defense - 5);

highScore = highScore + 20000\*((\*(mercenaries + count)).attack - 3);

if ((\*(mercenaries + count)).stable == true)

highScore = highScore + 20000;

}

highScore = highScore + \*funds;

file.write(initials, sizeof(initials));

file.write(reinterpret\_cast<char \*>(&highScore), sizeof(highScore));

cout<<endl<<endl;

}

file.close();

}

void def(int inN){

cout<<"Exiting the program..."<<endl;

}