**Gremlin v. Wizard: The Rematch**

**Part 2**

**CSC 5 46024**

**Nornubari Kanabolo**

**07/30/2014**

**Introduction**

In this game the wizard wants another chance of revenge on the gremlin for vanquishing him years ago. He travels the mountains searching for the gremlin and finally finds him at the peak of Mountain of Mordor. The wizard challenges him to battle to settle the score once and for all. However, the gremlin has grown more powerful than their last meeting and the wizard does not know if he can defeat him or not. The program decides whether or not the wizard(user) or gremlin attacks first. The user enters an option of what he wants to do and the program makes the gremlin retaliate and administer damage to the user. The user options are:

1 – Strong Attack

2 – Magic Attack

3 – Health Potion + Defensive Move

This continues until either the wizard or gremlin’s health hits 0. There are 4 functions that keep the score of the wizard and gremlin, project the map of the encounter on the screen, and repeat the program so the score accumulates.

**Summary**

Lines of code: 315

Comment Lines: 50

White spaces: 23

This project used most of the concepts and constructs we learned so far in the course. It uses a series of if, if-else, switch, while, and do while loops to control who goes first, what happens when a command is made, and how the gremlin or user wins the battle. It also used functions with call by references and values and a one dimensional array to display the map of battle. It was difficult to figure out where exactly to put the proper while, if else, and do-while statements and why they would go there. The array was difficult to code because I originally did not know how to display ASCII characters.

**Pseudo Code**

***Display map***

***Do***

***Initialize***

***If the user starts***

***Then the user chooses a move he wants to execute***

***Both the user and gremlin’s health are greater than 0***

***If the ghurt is less than 0***

***Then the gremlin takes no damage***

***If ghurt is greater than 0 then the user did damage to the gremlin***

***If ghp is less than 0***

***User wins***

***Output 1 point for user***

***If hurt is less than 0***

***User takes no damage***

***If hurt is greater than 0 then gremlin did damage to user***

***If hp>0 and hp<25***

***Then game prompts user to take health potion or else game ends***

***If hp is less than 1***

***Gremlin wins***

***Output 1 point for gremlin***

***Else gremlin starts***

***If hurt is less than 0***

***User takes no damage***

***If hurt is greater than 0 then gremlin did damage to user***

***If hp>0 and hp<25***

***Then game prompts user to take health potion or else game ends***

***If hp is less than 1***

***Gremlin wins***

***Output 1 point for gremlin***

***If user still alive***

***Then the user chooses a move he wants to execute***

***If the ghurt is less than 0***

***Then the gremlin takes no damage***

***If ghurt is greater than 0 then the user did damage to the gremlin***

***If ghp is less than 0***

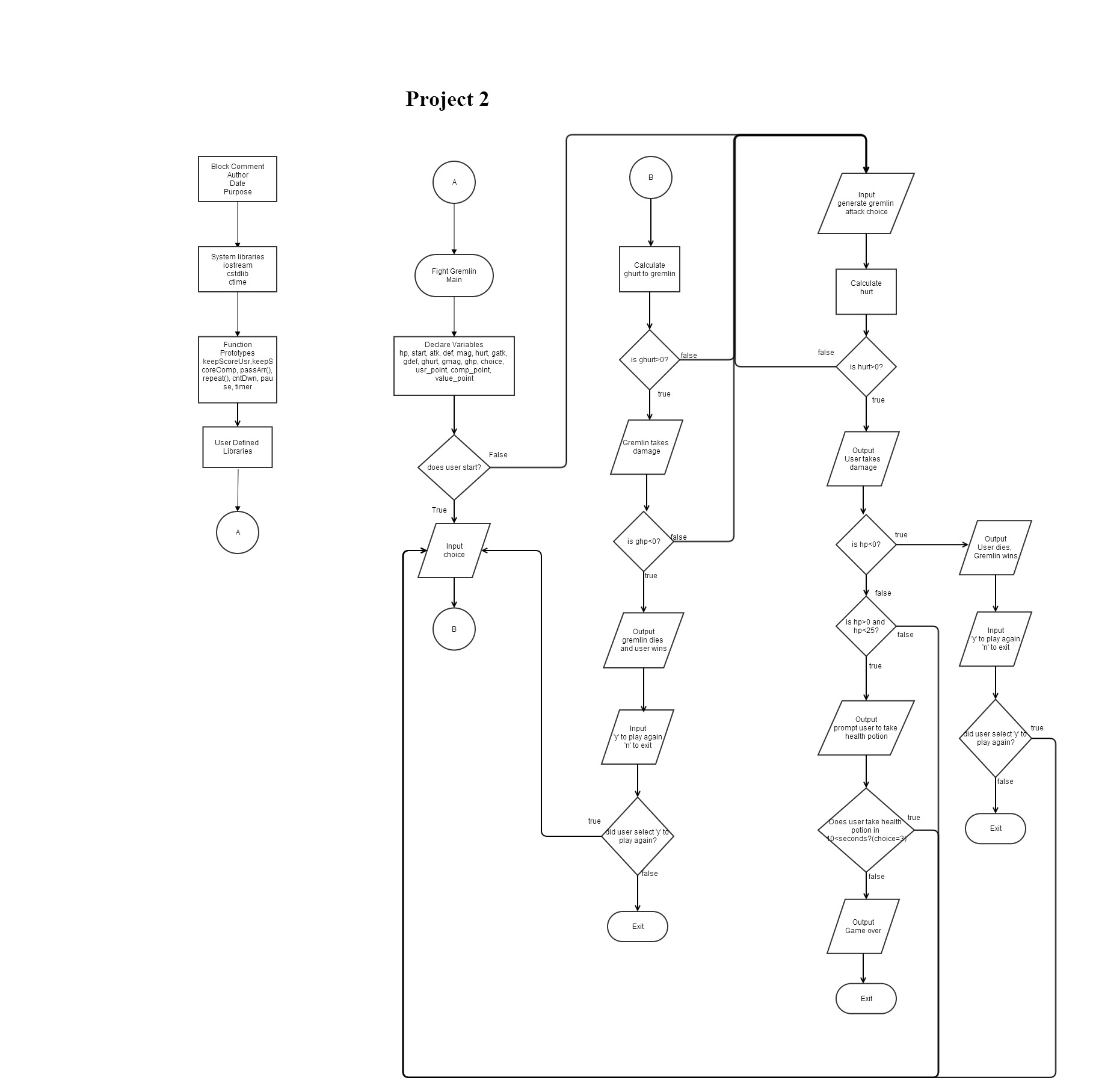
***User wins***

***Output 1 point for user***

***While the program repeats***

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Variable Name | Description | Location |
| Const in | SIZE |  |  |
| int | hp | User health points | If statement where calculation of damage to user is done |
|  | ghp | Gremlin health points | If statement where calculation of damage to gremlin is done |
|  | atk | User attack power | User menu choices of attack |
|  | gatk | Gremlin attack power | Gremlin menu choices of attack |
|  | mag | User magic power | User menu choices of attack |
|  | gmag | Gremlin magic power | Gremlin menu choices of attack |
|  | def | User defense power | User menu choices of attack |
|  | gdef | Gremlin defense power | Gremlin menu choices of attack |
|  | hurt | User damage taken | If statement where amount of damage taken by user is calculated |
|  | ghurt | Gremlin damage taken | If statement where amount of damage taken by gremlin is calculated |
|  | choice | Move execution choice | Menu choices for user and gremlin |
|  | Start | Determines who makes the first move | The first if statement of program |
|  | usr\_score | Reference value for user score | Function prototypes |
|  | comp\_score | Reference value for gremlin | Function prototypes |
|  | value\_usr | Reference value for user counter | Function prototypes |
|  | value\_comp | Reference value for gremlin counter | Function prototypes |
|  | usr\_point | User score value | Variable declaration and function calls after user win |
|  | comp\_point | Gremlin score value | Variable declaration and function calls after gremlin win |
|  | value\_point | Counter value | Variable declaration and function calls after user or gremlin win |
|  | flag | Timer settings value | In timer function near end of program |
|  | strt | Start time | In cntDwn function near end of program |
|  | freeze | Time output increment | In cntDwn function near end of program |
|  | now | Current time | In timer function |
|  | timer | Stopwatch for user decision | Where user is prompted to take health potion |
|  | secs | Number of seconds to pause execution | In timer function |
|  | stp | Stop time | In cntDwn function |
|  | wait | Wait time before user decision | In cntDwn function |
| static int | strt | Remember start time | In timer function |
|  | secs | Time elapsed | In timer function |
| char | choice | Repeat program | After while loop at end of main() |
| string | map | Lines of the map file | Right after variable declaration |

|  |  |  |
| --- | --- | --- |
| Chapter | Constructs/Syntax | Location |
| 2 | Equality and relational operators  (&&, ||, ==,>=,>,<=,<,-,+) | Where health, damage given, and taken are compared or calculation |
|  | bool | In the functions for the repetition of the program and timer |
|  | If | Where the damage to user and gremlin are determined |
|  | If else | If the user starts the program or the monster does |
|  | While | While either the user or gremlins health points is greater than 0 or less than 1 |
|  | Do-while | Executes the menu at least once before checking if health for either is diminished |
|  | for | Loop to output the map |
| 3 | Switch | The menu options for the user and gremlin |
|  | & call by reference | Functions where scores are calculated and returned |
| 7 | Array | Displaying the map array at the start of the game |



**Program Code**

*/\**

*\* File: main.cpp*

*\* Author: Nornubari Kanabolo*

*\* CSC 5 Project 2*

*\*/*

**#include <iostream>**

**#include <ctime>**

**#include <cstdlib>**

**#include <string>**

**#include <iomanip>**

**#include <fstream>**

**using** **namespace** std;

*//Function Prototypes*

*//Play again*

*//Keeping score*

*//user scores called by reference function*

**void** **keepScoreUsr**(**int&** usr\_score, **int&** comp\_score, **int&** value\_usr);

*//gremlin scores called by references function*

**void** **keepScoreComp**(**int&** comp\_score, **int&** usr\_score, **int&** value\_comp);

*//displays map function*

**void** **passArr**();

*//loops program so score is tallied*

**bool** **repeat**();

*//Countdown timer function*

**void** **cntDwn**(**int** strt,**int** freeze);

*//Pause function*

**void** **pause**(**int** secs);

**int** **timer**(**int=**0);

*//Execution*

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

*//Declare variables*

**int** choice, usr\_point**=**0, comp\_point**=**0, value\_point**=**0;

**int** hp, start, atk, def, mag, hurt, gatk, gdef, ghurt, gmag, ghp;

    atk **=** 12;*//user attack*

    def **=** 18;*//user defense*

    mag **=** 7;*//user magic*

    gatk **=** 14;*//gremlin attack*

    gdef **=** 18;*//gremlin defense*

    gmag **=** 7;*//gremlin magic*

    cout**<<**timer()**<<**" secs"**<<**endl;

**do**{

*//Random seed that determines who starts*

        start **=** rand()**%**2**+**1;

        hp **=** rand()**%**60**+**97;*//health points user starts with*

        ghp **=** rand()**%**40**+**98;*//health points gremlin starts with*

*//Project map*

        passArr();

*//Setup the random number seed*

        srand(**static\_cast<unsigned** **int>**(time(0)));

*//Display health for both characters*

        cout**<<**endl**<<**endl**<<**"Wizard health: "**<<**hp**<<**endl;

        cout**<<**"Gremlin health: "**<<**ghp**<<**endl**<<**endl;

**if** (start **==** 1)

        {

            cout**<<**"You attack first and swiftly!";

            cout**<<**endl;

*//User Menu*

**while** (hp **>** 0 **&&** ghp **>** 0) {

            cout**<<**"Choose what move you want to execute"**<<**endl;

            cout**<<**"1 - Strong Attack"**<<**endl;

            cout**<<**"2 - Magic Attack"**<<**endl;

            cout**<<**"3 - Health Potion + Defensive Move"**<<**endl;

**do**

            {

                cin**>>**choice;

            }**while**(choice**>**3 **||** choice**<**1);

**switch** (choice)

            {

**case** 1**:**

                atk **=** rand()**%**20**+**10;

                def **=** rand()**%**10**+**10;

                mag **=** rand()**%**5;

**break**;

**case** 2**:**

                atk **=** rand()**%**5**+**10;

                def **=** rand()**%**10**+**10;

                mag **=** rand()**%**15;

**break**;

**case** 3**:**

                atk **=** rand()**%**10**+**10;

                def **=** rand()**%**20**+**10;

                mag **=** rand()**%**5;

                hp **+=** 20;

**break**;

            }

*//Gremlin Menu decides which move it executes*

            choice **=** rand()**%**3;

**switch** (choice)

            {

**case** 1**:**

                gatk **=** rand()**%**20**+**10;

                gdef **=** rand()**%**10**+**10;

                gmag **=** rand()**%**5;

**break**;

**case** 2**:**

                gatk **=** rand()**%**5**+**10;

                gdef **=** rand()**%**10**+**10;

                gmag **=** rand()**%**15;

**break**;

**case** 3**:**

                gatk **=** rand()**%**10**+**10;

                gdef **=** rand()**%**20**+**10;

                gmag **=** rand()**%**5;

                ghp **+=** 40;

**break**;

            }

*//Damage to gremlin*

            ghurt **=** (atk **-** gmag) **-** (gdef**/**atk);

**if** (ghurt **<** 0)

            {

                ghurt **=** 0;

            }

            ghp **=** ghp **-** ghurt;

            cout**<<**"You did "**<<**ghurt**<<**" damage to the gremlin!";

            cout**<<**endl;

*//If user defeats gremlin*

**if** (ghp **<** 1)

            {

                cout**<<**"You destroyed the gremlin! You are victorious with "**<<**hp**<<**" hp to spare.";

                cout**<<**endl;

                keepScoreUsr(usr\_point, comp\_point, value\_point);*//displays user score*

            }

**if**(ghp**>**0) {

                cout**<<**"The gremlin now has "**<<**ghp**<<**" hp left.";

                cout**<<**endl;

            }

            hurt **=** (gatk **-** mag) **-** (def**/**gatk);

**if** (hurt **<** 0)

            {

                hurt **=** 0;

            }

            hp **=** hp **-** hurt;

**if** (hp**>**0 **&&** hp**<**25){

                cout**<<**"Low health! Use health potion before it's too late!\n";

                timer();

                 cout**<<**"Timer = "**<<**timer()**<<**" secs"**<<**endl;

            }**else**{

                timer(3);

                timer(1);

                timer(2);

            }

**if** (timer()**>**10){

                cout**<<**"Health potion not used in time. Game Over. You lose.\n";

            }

*//If gremlin defeats user*

**if** (hp **<** 1)

            {

                cout**<<**"You have been defeated. The gremlin lives with "**<<**ghp**<<**" hp remaining.";

                cout**<<**endl;

                keepScoreComp(comp\_point, usr\_point, value\_point);*//displays gremlin score*

            }

**if** (hp**>**0){

                cout**<<**"You now have "**<<**hp**<<**" hp left.\n"**<<**endl;

            }

            }

            }

*//The gremlin starts*

**else**

        {

            cout**<<**"Gremlin attacked first!"**<<**endl;

**while** (hp **>** 0 **&&** ghp **>** 0**&&**timer()**<=**10) {

                cout**<<**"Timer = "**<<**timer()**<<**" secs"**<<**endl;

            choice **=** rand()**%**3;

**switch** (choice)

            {

**case** 1**:**

                gatk **=** rand()**%**20**+**10;

                gdef **=** rand()**%**10**+**10;

                gmag **=** rand()**%**5;

**break**;

**case** 2**:**

                gatk **=** rand()**%**5**+**10;

                gdef **=** rand()**%**10**+**10;

                gmag **=** rand()**%**15;

**break**;

**case** 3**:**

                gatk **=** rand()**%**10**+**10;

                gdef **=** rand()**%**20**+**10;

                gmag **=** rand()**%**5;

                ghp **+=** 40;

                cout**<<**"Gremlin stole your health potion!\n";

**break**;

            }

*//Gremlin does damage to user*

            hurt **=** (gatk **-** mag) **-** (def**/**gatk);

**if** (hurt **<** 0)

            {

                hurt **=** 0;

            }

            hp **=** hp **-** hurt;

            cout**<<**"The gremlin hit you for "**<<**hurt**<<**" damage.";

            cout**<<**endl;

**if** (hp**>**0 **&&** hp**<**25){

                cout**<<**"Low health! Use health potion before it's too late!\n";

                timer();

                 cout**<<**"Timer = "**<<**timer()**<<**" secs"**<<**endl;

            }**else**{

                timer(3);

                timer(1);

                timer(2);

            }

*//If the gremlin kills the user*

**if** (hp **<** 1)

            {

                cout**<<**"You were killed. The gremlin still has "**<<**ghp**<<**" hp left.";

                cout**<<**endl;

                keepScoreComp(comp\_point, usr\_point, value\_point);

            }

**if** (hp**>**0){

                cout**<<**"You now have "**<<**hp**<<**" hp left.";

                cout**<<**endl;

            }

*//Next move for user if not killed*

            cout**<<**"Choose what move you want to execute"**<<**endl;

            cout**<<**"1 - Strong Attack"**<<**endl;

            cout**<<**"2 - Magic Attack"**<<**endl;

            cout**<<**"3 - Health Potion + Defensive Move"**<<**endl;

**do**{cin**>>**choice;}**while**(choice**>**3 **||** choice**<**1**&&**hp**>**0**&&**timer()**<=**10);

**switch** (choice)

            {

**case** 1**:**

                atk **=** rand()**%**20**+**10;

                def **=** rand()**%**10**+**10;

                mag **=** rand()**%**5;

**break**;

**case** 2**:**

                atk **=** rand()**%**5**+**10;

                def **=** rand()**%**10**+**10;

                mag **=** rand()**%**15;

**break**;

**case** 3**:**

                atk **=** rand()**%**10**+**10;

                def **=** rand()**%**20**+**10;

                mag **=** rand()**%**5;

                hp**+=**20;

**break**;

            }

*//User hurts gremlin*

            ghurt **=** (atk **-** gmag) **-** (gdef**/**atk);

**if** (ghurt **<** 0)

            {

                ghurt **=** 0;

            }

            ghp **=** ghp **-** ghurt;

**if** (timer()**>**10){

                cout**<<**"Health potion not used in time. Game Over. You lose.\n";

            }**else**{

            cout**<<**"You did "**<<**ghurt**<<**" damage to the gremlin!";

            cout**<<**endl;

            }

*//User wins*

**if** (ghp **<** 1)

            {

                cout**<<**"You destroyed the gremlin! You are victorious!";

                cout**<<**endl;

                keepScoreUsr(usr\_point, comp\_point, value\_point);

            }

**if**(ghp**>**0){

                cout**<<**"The gremlin now has "**<<**ghp**<<**" hp left."**<<**endl;

                cout**<<**endl;

            }

            }

        }

    }**while**(repeat());

**return** 0;

}

*//User score keeping function*

*//Score keeping function*

**void** **keepScoreUsr**(**int&** usr\_score, **int&** comp\_score, **int&** value\_usr)

{

    usr\_score**++**;

    usr\_score**=**usr\_score**+**value\_usr;

    cout**<<**endl**<<**"Your Score: "**<<**usr\_score**<<**endl;

    cout**<<**endl**<<**"Gremlin's Score: "**<<**comp\_score**<<**endl;

}

*//Gremlin score keeping function*

**void** **keepScoreComp**(**int&** comp\_score, **int&** usr\_score,**int&** value\_comp)

{

    comp\_score**++**;

    comp\_score**=**comp\_score**+**value\_comp;

    cout**<<**endl**<<**"Gremlin's Score: "**<<**comp\_score**<<**endl;

    cout**<<**endl**<<**"Your Score: "**<<**usr\_score**<<**endl;

}

*//Map displaying function*

**void** **passArr**(){

**const** **int** SIZE **=** 45;

    string map [SIZE];\

    fstream file;

    file.open("map.txt");

**for**(**int** i**=**0;i**<**SIZE;i**++**)

        getline(file,map[i]);

    file.close();

**for**(**int** i**=**0;i**<**SIZE;i**++**)

        cout**<<**map[i]**<<**endl;

}

*//Function to repeat program*

**bool** **repeat**(){

**char** choice;

**do**{

        cout**<<**"Do you want to continue [y/n]?"**<<**endl;

        cin**>>**choice;

    }**while**(choice**!=**'y'**&&**choice**!=**'n');

**if**(choice**==**'y'){

**return** true;

    }

**if**(choice**==**'n'){

**return** false;

    }

}

**int** **timer**(**int** flag){

*//Declare Variables*

**static** **int** strt**=**time(0);*//remember start time*

**static** **int** secs**=**time(0);*//seconds elapsed*

**static** **bool** stop**=**false;*//remember if timer stopped*

**int** now**=**time(0);*//current time*

**if**(stop**&&**flag**!=**2){ *//Stopped and not reset*

**return** secs;

    }**else** **if**(flag**==**0){ *//Normal timer*

        secs**=**now**-**strt;

    }**else** **if**(flag**==**3){ *//Reset*

        strt**=**time(0);

        secs**=**now**-**strt;

    }**else** **if**(flag**==**2){ *//Restart*

        stop**=**false;

        strt**=**now**-**secs;

    }**else**{ *//When stopped*

        stop**=**true;

        secs**=**now**-**strt;

    }

**return** secs;

}

**void** **cntDwn**(**int** strt,**int** freeze){

*//Perform the countdown*

**do**{

        cout**<<**strt**<<**" seconds remaining"**<<**endl;

        strt**-=**freeze;

        pause(freeze);

    }**while**(strt**>**0);

    cout**<<**strt**<<**" seconds"**<<**endl;

}

**void** **pause**(**int** secs){

*//Start the time*

**int** stp,wait,strt**=**time(0);

**do**{

        stp**=**time(0);

        wait**=**stp**-**strt;

    }**while**(wait**<=**secs);

}