Project v 1.2

Squire of Zir

Zach Stewart

CSC5 Intro to C++

46024

7/29/2014

V 1.2 notes will be in italics.

**Introduction:** This program is like a game described to me by my step-dad when computers games were only text based and very simple. This game is very self-explanatory and lays out what it wants in the text. The program lays out a scenario for the user and allows a little immersion into it by using the users name in the story. The user follows the onscreen prompts from the program and goes on a quest. There is a fight sequence in the game that the user terminates when they want and also a trap along the way. There are 5 win/lose conditions in the game.

**Notes:** There are many things that would have made this game smoother with more knowledge of C++. I would like to revisit this project later on during my C++ education. I want to make this game better and develop more options to it and make the UI even better, but as it stands I don’t know enough about C++ and must forgo my unhappiness with it for now. I had a bit of trouble with ending my code at the trap sequence, so instead I added a damage modifier. *The game was modified to include functions, arrays, and file input output. This was to make the code simpler. I also tried using std::tuple to get two variables back out from my functions, but I was unable to use them. I’m not sure if the issue was with Net beans or cywin, but either way it didn’t work so I changed my function into two functions. This made the game easy to win, but I was able to keep my win/loss conditions workable.*

**Rules:** The rules are very simple and strait forward. Follow the onscreen prompts as they appear and follow the story. There are no tricks in the program. Just choose what you would like to do in the game and watch the story develop.

**Approach:** My approach to this program was to think back to how my step-dad described the old text-based rpg games he would play when computers were first being developed for home usage. I thought about what I’ve learned already in C++ and thought about how I would use everything I’d learned already. I then focused on developing the story and plugging in what I’ve learned in areas that seemed right to use them. *I added in function, arrays, and file input output as I learned more about them.*

**Variable List:** const int - TOWER = 1 – this is used to hold the choice by the user for what quest they want to complete.

Const int - CAVE = 2 – this is used to hold the choice by the user for what quest they want to complete.

Int - choice , endFight – these are used to determine the users choice during the quest choosing sequence, and if the user wants to end the fight.

Short - hp = 100, mp = 30 – these hold the amount of health points and mana points the user still has left.

Float - dragonHp = 150, ogreHp = 150 – these hold the amount of health points the dragon and ogre have.

Char - chest, fight, door – these hold the results of the trap sequences and the fight sequence .

String – name – this hold the users name for use in the game.

Int - sword = rand() % 10 + 1, spell = rand() % 25 + 15 – these hold the amount of damage you do in the fight sequence.

Int - sword = rand() % 12 + 1, spell = rand() % 25 + 15 – these hold the amount of damage you do in the fight sequence.

Int - dragonDmg = rand() % 12 + 1 – this holds the amount of damage the dragon does in the fight sequence.

Int - ogreDmg = rand() % 13 + 1 – this holds the amount of damage the ogre does in the fight sequence.

**Topics Covered:** do while loop, if else, if else if, switch, char, float, short, int, iostream/iomanip/cstdlib/ctime/string/fstream, ==, <=, != , >=, ++, menu, &&, function, *string array, file input/output*

**Flowchart:**

**Code:**

/\*

\* File: main.cpp

\* Author: Stewart, Zach

\* Class: 46024

\*

\* Created on July 29, 2014, 7:00 PM

\*/

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <string>

#include <iomanip>

#include <fstream>

using namespace std;

/\*

\*

\*/

ofstream outputFile;

int main() {

outputFile.open("game.txt");

//seed random function here

unsigned seed = time (0);

srand(seed);

//set up function prototypes

int dragonFight(float, short);

int ogreFight(float, short);

int dragon(short, short);

int ogre(short, short);

void chest();

void door();

int output();

//set variables

const int TOWER = 1;

const int CAVE = 2;

const int NAMES = 3;

string name[NAMES];

float dragonHp = 150, ogreHp = 150;

short hp = 100, mp = 30;

int choice;

//Explaining what's happening in the game and making a story

//Use 2 dimensional array to get names of characters

cout << "You are a squire in the kingdom of Zir.\n";

cout << "Enter your name.\n";

getline(cin, name[0]);

cout << "Who is the king you serve?\n";

getline(cin, name[1]);

cout << name[0] << ", you are going on a quest to become a knight sent by ";

cout << name[1] << "\n";

cout << "You also have a special ability that allows you to ";

cout << "regen 2 HP per round you fight.\n";

cout << "Who is the knight you squire for?\n";

getline(cin, name[2]);

cout << "Before going " << name[2]<< " gives you a sword and shield.\n";

cout << name[0] << ", your mission is to slay either the dragon or the ogre\n";

cout << name[0] << ", you are in the forest and find the ogre's cave and the";

cout << "dragon's tower.\n";

cout << name[0] << ", do you choose to enter the cave or the tower.\n";

//Display quest options

cout << "1. Tower\n";

cout << "2. Cave\n";

cin >> choice;

//Make choice where to go

if (choice == TOWER)

{

cout << name[0] << " has chosen to enter the tower and fight the dragon.\n";

//Run trap function

chest();

cout << "You finally arrive at the lair of the dragon. \n";

cout << "Prepare yourself for battle.\n\n";

//Run fight function

dragonHp = dragonFight(dragonHp, mp);

hp = dragon(hp, mp);

//Run end of game sequence

if (hp <= 0 && dragonHp <= 0)

{

cout << name[0] << " and the dragon both have died.\n";

cout << "Although you died, you slain the dragon.\n";

cout << name[0] << " is posthumously knighted and celebrated";

cout << " throughout the kingdom. GAMEOVER.\n";

}

else if (hp >= 0 && dragonHp >= 0)

{

cout << name[0] << " has fled from the battle with the dragon.\n";

cout << "You are banished from the kingdom and must live the ";

cout << "rest of your life in disgrace.\n";

cout << "GAME OVER " << name[0] << " Coward of the Kingdom.\n";

}

else if (hp <= 0)

{

cout << name[0] << " has died on your quest.\n";

cout << "The kingdom mourns your death.\n";

cout << "GAME OVER";

}

else if (dragonHp <= 0)

{

cout << name[0] << " has slain the dragon.\n";

cout << "You return victoriously to the kingdom with the ";

cout << "dragon's head as proof of your victory.\n";

cout << "King Mark knights you and the kingdoms cheers.\n";

cout << "CONGRADULATIONS SIR " << name[0] << " YOU WIN!\n";

}

}

else

{

cout << name[0] << " has chosen to enter the cave and fight the ogre.\n";

//Run trap function

door();

cout << "You finally arrive at the den of the ogre. \n";

cout << "Prepare yourself for battle.\n\n";

//Start fight function

ogreHp = ogreFight(ogreHp, mp);

hp = ogre(hp, mp);

//Run end of game sequence

if (hp <= 0 && ogreHp <= 0)

{

cout << name[0] << " and the dragon both have died.\n";

cout << "Although you died, you slaid the dragon.\n";

cout << name[0] << " is posthumously knighted and celebrated ";

cout << "throughout the kingdom. GAMEOVER.\n";

}

else if (hp >= 0 && ogreHp >= 0)

{

cout << name[0] << " has fled from the battle with the ogre.\n";

cout << "You are banished from the kingdom and must live the ";

cout << "rest of your life in disgrace.\n";

cout << "GAME OVER " << name[0] << " Coward of the Kingdom.\n";

}

if (hp <= 0)

{

cout << name[0] << " has died on your quest.\n";

cout << "The kingdom mourns your death.\n";

cout << "GAME OVER";

}

else if (ogreHp <= 0)

{

cout << name[0] << " has slain the ogre.\n";

cout << "You return victoriously to the kingdom with the ogre's ";

cout << "head as proof of your victory.\n";

cout << "King Mark knights you and the kingdoms cheers.\n";

cout << "CONGRADULATIONS SIR " << name[0] << " YOU WIN!";

}

}

outputFile << "You are a squire in the kingdom of Zir.\n";

outputFile << "Enter your name.\n";

outputFile << "Who is the king you serve?\n";

outputFile << name[0] << ", you are going on a quest to become a knight sent by ";

outputFile << name[1] << "\n";

outputFile << "You also have a special ability that allows you to ";

outputFile << "regen 2 HP per round you fight.\n";

outputFile << "Who is the knight you squire for?\n";

outputFile << "Before going " << name[2]<< " gives you a sword and shield.\n";

outputFile << name[0] << ", your mission is to slay either the dragon or the ogre\n";

outputFile << name[0] << ", you are in the forest and find the ogre's cave and the";

outputFile << "dragon's tower.\n";

outputFile << name[0] << ", do you choose to enter the cave or the tower.\n";

outputFile << "1. Tower\n";

outputFile << "2. Cave\n";

outputFile << name[0] << " has chosen to enter the tower and fight the dragon.\n";

outputFile << "You finally arrive at the lair of the dragon. \n";

outputFile << "Prepare yourself for battle.\n\n";

outputFile << name[0] << " and the dragon both have died.\n";

outputFile << "Although you died, you slain the dragon.\n";

outputFile << name[0] << " is posthumously knighted and celebrated";

outputFile << " throughout the kingdom. GAMEOVER.\n";

outputFile << name[0] << " has fled from the battle with the dragon.\n";

outputFile << "You are banished from the kingdom and must live the ";

outputFile << "rest of your life in disgrace.\n";

outputFile << "GAME OVER " << name[0] << " Coward of the Kingdom.\n";

outputFile << name[0] << " has died on your quest.\n";

outputFile << "The kingdom mourns your death.\n";

outputFile << "GAME OVER";

outputFile << name[0] << " has slain the dragon.\n";

outputFile << "You return victoriously to the kingdom with the ";

outputFile << "dragon's head as proof of your victory.\n";

outputFile << "King Mark knights you and the kingdoms cheers.\n";

outputFile << "CONGRADULATIONS SIR " << name[0] << " YOU WIN!\n";

outputFile << name[0] << " has chosen to enter the cave and fight the ogre.\n";

outputFile << "You finally arrive at the den of the ogre. \n";

outputFile << "Prepare yourself for battle.\n\n";

outputFile << name[0] << " and the dragon both have died.\n";

outputFile << "Although you died, you slaid the dragon.\n";

outputFile << name[0] << " is posthumously knighted and celebrated ";

outputFile << "throughout the kingdom. GAMEOVER.\n";

outputFile << name[0] << " has fled from the battle with the ogre.\n";

outputFile << "You are banished from the kingdom and must live the ";

outputFile << "rest of your life in disgrace.\n";

outputFile << "GAME OVER " << name[0] << " Coward of the Kingdom.\n";

outputFile << name[0] << " has died on your quest.\n";

outputFile << "The kingdom mourns your death.\n";

outputFile << "GAME OVER";

outputFile << name[0] << " has slain the ogre.\n";

outputFile << "You return victoriously to the kingdom with the ogre's ";

outputFile << "head as proof of your victory.\n";

outputFile << "King Mark knights you and the kingdoms cheers.\n";

outputFile << "CONGRADULATIONS SIR " << name[0] << " YOU WIN!";

outputFile << "As you ascend the tower you come across a chest do you open it?\n";

outputFile << "Y/N\n";

outputFile << "You triggered a trap but it was inactive. Let ";

outputFile << "this be a warning for next time.\n\n";

outputFile << "You ignore the booby trapped chest and move ";

outputFile << "on.\n\n";

outputFile << "That is not a valid option. Enter you choice ";

outputFile << "again.\n\n";

outputFile << "You triggered a trap but it was ";

outputFile << "inactive. Let this be a warning ";

outputFile << "for next time.\n\n";

outputFile << "You ignore the booby trapped ";

outputFile <<"chest and move on.\n\n";

outputFile << "As you enter the cave you come across a door marked with Treasure";

outputFile << "Room do you open it?\n";

outputFile << "Y/N\n";

outputFile << "You triggered a trap but it was inactive. ";

outputFile << "Let this be a warning for next time.\n\n";

outputFile << "You ignore the booby trapped room and move ";

outputFile << "on.\n\n";

outputFile << "That is not a valid option. Enter you choice ";

outputFile << "again.\n\n";

outputFile << "You triggered a trap but it was ";

outputFile << "inactive. Let this be a warning ";

outputFile << "for next time.\n\n";

outputFile << "You ignore the booby trapped ";

outputFile <<"door and move on.\n\n";

outputFile << left << "1. Fight\t\t2. Magic (Fireball/10 MP)\n";

outputFile << "You do not have enough Mana.\n";

outputFile << "You must fight with your sword.\n";

outputFile << "That's not a valid option. Please enter ";

outputFile << "your choice again.\n";

outputFile << "You do not have enough";

outputFile << " Mana.\n";

outputFile << "You must fight with your";

outputFile << " sword.\n";

outputFile << "\n\n";

outputFile << "The dragon still has " << dragonHp << " HP left.\n\n";

outputFile << "Do you want to keep fighting? 1. Yes or 2. No.\n\n";

outputFile << "You still have " << hp << " left and " << mp << " left.\n";

outputFile << "Do you want to keep fighting? 1. Yes or 2. No.\n\n";

outputFile << left << "1. Fight\t\t2. Magic (Iceball/10 MP)\n";

outputFile << "You do not have enough Mana.\n";

outputFile << "You must fight with your sword.\n";

outputFile << "That's not a valid option. Please enter ";

outputFile << "your choice again.\n";

outputFile << "You do not have enough";

outputFile << "Mana.\n";

outputFile << "You must fight with your";

outputFile << "sword.\n";

outputFile << "The ogre still has " << ogreHp << " HP left.\n\n";

outputFile << "Do you want to keep fighting? 1. Yes or 2. No.\n\n";

outputFile << "You still have " << hp << " HP and " << mp << " MP left.\n";

outputFile << "Do you want to keep fighting? 1. Yes or 2. No.\n\n";

outputFile.close();

return 0;

}

//Trap sequence

void chest()

{

char chest;

cout << "As you ascend the tower you come across a chest do you open it?\n";

cout << "Y/N\n";

cin >> chest;

switch (chest)

{

case 'y':

case 'Y': cout << "You triggered a trap but it was inactive. Let ";

cout << "this be a warning for next time.\n\n";

break;

case 'n':

case 'N': cout << "You ignore the booby trapped chest and move ";

cout << "on.\n\n";

break;

default: cout << "That is not a valid option. Enter you choice ";

cout << "again.\n\n";

cin >> chest;

switch (chest)

{

case 'y':

case 'Y': cout << "You triggered a trap but it was ";

cout << "inactive. Let this be a warning ";

cout << "for next time.\n\n";

break;

case 'n':

case 'N': cout << "You ignore the booby trapped ";

cout <<"chest and move on.\n\n";

}

}

return;

}

//Trap sequence

void door()

{

char door;

cout << "As you enter the cave you come across a door marked with Treasure";

cout << "Room do you open it?\n";

cout << "Y/N\n";

cin >> door;

switch (door)

{

case 'y':

case 'Y': cout << "You triggered a trap but it was inactive. ";

cout << "Let this be a warning for next time.\n\n";

break;

case 'n':

case 'N': cout << "You ignore the booby trapped room and move ";

cout << "on.\n\n";

break;

default: cout << "That is not a valid option. Enter you choice ";

cout << "again.\n\n";

cin >> door;

switch (door)

{

case 'y':

case 'Y': cout << "You triggered a trap but it was ";

cout << "inactive. Let this be a warning ";

cout << "for next time.\n\n";

break;

case 'n':

case 'N': cout << "You ignore the booby trapped ";

cout <<"door and move on.\n\n";

}

}

return;

}

//You doing damage sequence

int dragonFight(float dragonHp = 150, short mp = 30)

{

char fight;

int endFight;

do

{

int sword = rand() % 10 + 1, spell = rand() % 25 + 15;

cout << left << "1. Fight\t\t2. Magic (Fireball/10 MP)\n";

cin >> fight;

//Decide how to fight

switch (fight)

{

case '1': cout << "You do " << sword << " damage.\n";

dragonHp = dragonHp - sword;

outputFile << "You do " << sword << " damage.\n";

break;

case '2': if (mp <=1)

{

cout << "You do not have enough Mana.\n";

cout << "You must fight with your sword.\n";

cout << "You do " << sword << " damage.";

cout << "\n\n";

outputFile << "You do " << sword << " damage.";

outputFile << "\n\n";

}

else

{

cout << "You do " << spell << " damage.\n\n";

outputFile << "You do " << spell << " damage.\n\n";

dragonHp = dragonHp - spell;

mp = mp - 10;

}

break;

default: cout << "That's not a valid option. Please enter ";

cout << "your choice again.\n";

cin >> fight;

switch (fight)

{

case '1': cout << "You do " << sword << "";

cout << "damage.\n\n";

outputFile << "You do " << sword << "";

outputFile << "damage.\n\n";

dragonHp = dragonHp - sword;

break;

case '2': if (mp <=1)

{

cout << "You do not have enough";

cout << " Mana.\n";

cout << "You must fight with your";

cout << " sword.\n";

cout << "You do " << sword << " ";

cout << "damage.";

cout << "\n\n";

}

else

{

cout << "You do " << spell << " ";

cout << "damage.\n\n";

outputFile << "You do " << spell << " ";

outputFile << "damage.\n\n";

dragonHp = dragonHp - spell;

mp = mp - 10;

}

}

}

cout << "The dragon still has " << dragonHp << " HP left.\n\n";

cout << "Do you want to keep fighting? 1. Yes or 2. No.\n\n";

cin >> endFight;

} while (endFight != 2);

return dragonHp;

}

//Damage done to you sequence

int dragon(short hp = 100, short mp = 30)

{

int endFight;

int dragonDmg = rand() % 12 + 1;

do

{

cout << "The dragon does " << dragonDmg << " damage to you.\n";

outputFile << "The dragon does " << dragonDmg << " damage to you.\n";

hp = hp - dragonDmg;

hp++;

cout << "You still have " << hp << " left and " << mp << " left.\n";

cout << "Do you want to keep fighting? 1. Yes or 2. No.\n\n";

cin >> endFight;

} while (endFight != 2);

return hp;

}

//You fighting sequence

int ogreFight(float ogreHp = 150, short mp = 30)

{

char fight;

int endFight;

do

{

int sword = rand() % 12 + 1, spell = rand() % 25 + 15;

cout << left << "1. Fight\t\t2. Magic (Iceball/10 MP)\n";

cin >> fight;

//Decide how to fight

switch (fight)

{

case '1': cout << "You do " << sword << " damage.\n";

outputFile << "You do " << sword << " damage.\n";

ogreHp = ogreHp - sword;

break;

case '2': if (mp <=1)

{

cout << "You do not have enough Mana.\n";

cout << "You must fight with your sword.\n";

cout << "You do " << sword << " damage.\n\n";

outputFile << "You do " << sword << " damage.\n\n";

ogreHp = ogreHp - sword;

}

else

{

cout << "You do " << spell << " damage.\n\n";

outputFile << "You do " << spell << " damage.\n\n";

ogreHp = ogreHp - spell;

mp = mp - 10;

}

break;

default: cout << "That's not a valid option. Please enter ";

cout << "your choice again.\n";

cin >> fight;

switch (fight)

{

case '1': cout << "You do " << sword << " ";

cout << "damage.\n\n";

outputFile << "You do " << sword << " ";

outputFile << "damage.\n\n";

ogreHp = ogreHp - sword;

break;

case '2': if (mp <=1)

{

cout << "You do not have enough";

cout << "Mana.\n";

cout << "You must fight with your";

cout << "sword.\n";

cout << "You do " << sword << " ";

cout << "damage.\n\n";

outputFile << "You do " << sword << " ";

outputFile << "damage.\n\n";

ogreHp = ogreHp - sword;

}

else

{

cout << "You do " << spell << "";

cout << " damage.\n\n";

outputFile << "You do " << spell << "";

outputFile << " damage.\n\n";

ogreHp = ogreHp - spell;

mp = mp - 10;

}

}

}

cout << "The ogre still has " << ogreHp << " HP left.\n\n";

cout << "Do you want to keep fighting? 1. Yes or 2. No.\n\n";

cin >> endFight;

} while (endFight != 2);

return ogreHp;

}

//Damage done to you sequence

int ogre(short hp = 100, short mp = 100)

{

char endFight;

int ogreDmg = rand() % 13 + 1;

do

{

cout << "The ogre does " << ogreDmg << " damage to you.\n";

outputFile << "The ogre does " << ogreDmg << " damage to you.\n";

hp = hp - ogreDmg;

hp++;

cout << "You still have " << hp << " HP and " << mp << " MP left.\n";

cout << "Do you want to keep fighting? 1. Yes or 2. No.\n\n";

cin >> endFight;

} while (endFight != 2);

return hp;

}