

The Devilish Rogue: A Dungeons and Dragons Scenario

D&D 5th Edition

RYAN JOHNSON

Project 2

CSC-5

Section 45561

**Table of Contents**



**Introduction......................................................................................................................3**

**Rules and Objective..........................................................................................................3**

**Concept Checklist..............................................................................................................4**

**Flow Charts to Clarify Ideas...............................................................................................7**

**The Code.........................................................................................................................11**

**Introduction**

For my project, I decided to write a short Dungeons and Dragons scenario of a level 5 Rogue-Thief on his/her own in an inn for an evening. I used the Dungeons and Dragons 5th Edition Dungeon Master's Guide as a reference for the rules as this is the version of the game I play with my friends. This undertaking ended up being far more complex than I first thought it would be. There is so much simple math that happens automatically on both sides of the DM screen that I took for granted how simply 5th edition plays. The next factor that possibly caused un-due complexity was that I wanted to write my program in a way that it could be expanded in the future for new characters and classes. In addition to the possibility for expansion I wrote the ability scores in a way that they could easily be modified and even though the adventurer in this scenario will only use the dexterity and charisma modifiers, the rest of the stats are filled out. My hope was to make it possible to level up and adjust ability scores. I was advised that this undertaking was quite complex for an intro C++ project and that all of this would be much easier with objects and classes but I trudged on diligently as ever. Of course, I scaled back my original plan once I got to about the seven-hundredth line, I even scrapped a whole branch of the scenario in the 11th hour of the project.

I have easily spent 40 hours on this project and I am quite proud of the finished product. The scenario includes money changing hands, gambling, a simple game within the game, deception checks, sleight of hand checks, and finally and most complexly combat.

**Rules and Strategy**

In this scenario, you will be taking on the role of a thief. A good thief is cunning, dexterous, and cold hearted. An expert in deception and sleight of hand, you use your skills to dupe people and take what you want.

The goal of the game is to acquire as much gold as possible. Losing a fight will get you thrown out of the inn and end your evening, getting caught doing questionable deeds will result in the same outcome. Combat will close of some doors so avoid it if possible, while your skill in combat will most likely secure your safety, a fight could shut you of from acquiring more coin. Buying beer for your soon to be victims will make things easier but you must weigh the cost against the possible gain. Once you have had your fill of thieving and deceiving you can call it a night and go upstairs to you room to count your coins.

A comprehensive source of the Dungeons and Dragons 5th Edition rules can be found here:

<https://www.5thsrd.org/>

**Concepts Checklist**

**Chapter 2**

|  |  |  |
| --- | --- | --- |
| **Section** | **Topic** | **Location in Code** |
| **2** | **cout<<** | **147** |
| **3** | **Libraries:** | **\*\*\*\*** |
|  | **iostream** | **147** |
|  | **iomanip** | **553** |
|  | **cmath** | **185** |
|  | **cstdlib** | **771** |
|  | **fstream** | **460** |
|  | **string** | **149** |
|  | **ctime** | **65** |
| **4** | **variables/literals** | **112,113** |
| **5** | **Identifiers** | **301** |
| **6** | **Integers** | **184** |
| **7** | **Characters** | **171** |
| **8** | **Strings** | **149** |
| **9** | **Floats** | **551** |
| **10** | **Bools** | **172** |
| **11** | **Sizeof\*\*\*\*** | **\*\*\*\*** |
| **12** | **Variables of 7 characters or less** | **68 through 135** |
| **13** | **Scope \*\*\*\* No Global Variables** | **\*\*\*\*** |
| **14** | **Arithmetic operators** | **551** |
| **15** | **Comments 20%+** | **552** |
| **16** | **Named Constants** | **70** |
| **17** | **Program Style \*\*\*\*Emulated** | **\*\*\*\*** |

**Chapter 3**

|  |  |  |
| --- | --- | --- |
| **Section** | **Topic** | **Location in Code** |
| **1** | **cin** | **148** |
| **2** | **Math Expression** | **786** |
| **3** | **Mixing Data types** | **\*\*\*\*** |
| **4** | **Overflow/Underflow** | **\*\*\*\*** |
| **5** | **Type Casting** | **542** |
| **6** | **Multiple Assignment** | **\*\*\*\*** |
| **7** | **Formatting output** | **553** |
| **8** | **Strings** | **149** |
| **9** | **Math Library** | **185** |
| **10** | **Hand Tracing** | **\*\*\*\*** |

**Chapter 4**

|  |  |  |
| --- | --- | --- |
| **Section** | **Topic** | **Location in Code** |
| **1** | **Relational Operators** | **178** |
| **2** | **If** | **172** |
| **4** | **If-Else** | **201,203** |
| **5** | **Nesting** | **207** |
| **6** | **If-Else-If** | **415,424,430** |
| **7** | **Flags** | **\*\*\*\*** |
| **8** | **Logical Operators** | **219** |
| **11** | **Validating User Input** | **520** |
| **13** | **Conditional Operator** | **219** |
| **14** | **Switch** | **170 through 230** |

**Chapter 5**

|  |  |  |
| --- | --- | --- |
| **Section** | **Topic** | **Location in Code** |
| **1** | **Increment/Decrement** | **545,845** |
| **2** | **While** | **520** |
| **5** | **Do-While** | **167 through 231** |
| **6** | **For Loop** | **770** |
| **11** | **File I/O** | **253,243** |
| **12** | **No Breaks in Loops** | **\*\*\*\*** |

**Chapter 6**

|  |  |  |
| --- | --- | --- |
| **Section** | **Topic** | **Location in Code** |
| **3** | **Function Prototypes** | **23 through 54** |
| **5** | **Passing by Value** | **826** |
| **8** | **Returning values from functions** | **828** |
| **9** | **Returning a boolean** | **\*\*\*\*** |
| **10** | **No Global Variables Allowed** | **\*\*\*\*** |
| **11** | **Static Local** | **801** |
| **12** | **Default Arguments** | **58** |
| **13** | **Reference Parameters** | **475** |
| **14** | **Overloading functions** | **28,29** |
| **15** | **Exit Functions** | **\*\*\*\*** |

**Chapter 7**

|  |  |  |
| --- | --- | --- |
| **Section** | **Topic** | **Location in Code** |
| **4** | **Array Initialization** | **256 and 257** |
| **6** | **Processing Arrays** | **262** |
| **7** | **Parallel Arrays** | **262** |
| **8** | **Arrays as a function argument** | **281** |
| **9** | **2-D Arrays** |  |
| **12** | **STL Vector** |  |

**Chapter 8**

|  |  |  |
| --- | --- | --- |
| **Section** | **Topic** | **Location in Code** |
| **1** | **Linear and Binary Search** | **281** |
| **3** | **Bubble and Selection Sort** | **272** |
| **5** | **Searching/Sorting Vectors** | **\*\*\*\*** |

**Clarifying the Code :With visual representation 2**

|  |  |
| --- | --- |
| **-End of Combat function with if{}-else-if{}-else{}:**   * **If player runs away victory=false and health will be >0, so we output the ran away message.** * **If combat was lost health =0 and victory=false. Output combat loss message and half gold.** * **If victory = true, output combat win message and recover some of the gold that was at stake during gambling round. Display message.** * **Reach end of function and leave gambling scenario.** |  |

|  |  |
| --- | --- |
|  | **-independent if {statement}**   * **If combat is required due to a development in the gambling scenario gold sum because negative to trigger the combat function.** * **If gold is negative we go into the combat function which alters variables as necessary and leads to the combat resolution function.** * **Combat resolution function assesses the outcome of combat based on the victory bool and current value of health.** * **If gold is >0 when leaving the gambling function then we hit the end of the initial switch and make the next decision of what to do in the common room.** |

|  |  |
| --- | --- |
| **-The Love Fool Scenario:**  **A switch**   * **Choosing an action determines which branch of the scenario you go down.** * **Action 1 buys a beer for the fool** * **Action 2 attempts to steal the engagement ring.** * **Action 3 is to simply walk away and let things unfold without intervention.** * **This whole function is in a do while loop.** |  |

|  |  |
| --- | --- |
|  | **-do{}while**   * **This branch loops two functions as long as the ring is still in play and the player still has gold to buy a drink.** * **The first function outputs a message based on how many beers have been bought for the fool.** * **The we go into the function show on the previous flowchart where the player chooses what to do with the fool.** * **As long as the ring stays on the fool the loop continues.** |

|  |  |
| --- | --- |
| **-Rolling dice:**  **For loops{}**   * **When this function is called the number of dice to be rolled is indicated.** * **The total of the die roll is initialized to 0.** * **While fewer dice have been rolled than is required the function will loop and the total will continue to add random number ranged between 1 and 6.** * **The function returns the total rolled.** |  |

|  |  |
| --- | --- |
|  | **While{}**   * **This is an example of input validation.** * **The user inputs a wager.** * **The wager must be:**   + **< = the amount of gold they have.**   + **< = the amount of money available from the table.**   + **> 0.**   + **Of the correct data type ie not causing an error.** * **If it is not then a message is displayed and a new wager is prompted.** * **Once and acceptable wager has been entered the game proceeds.** |

**Pulling Back the Curtain: My Code**

|  |  |
| --- | --- |
| /\* |  |
|  | \* File: main.cpp |
|  | \* Author: Ryan Johnson |
|  | \* Created on July 25, 2017, 12:16 PM |
|  | \* Purpose: Project 2: The Devilish Rogue w/ Arrays |
|  | \*/ |
|  |  |
|  | //System Libraries |
|  | #include <iostream> //Input - Output Library |
|  | #include <ctime> //Time |
|  | #include <cstdlib> //Random Function |
|  | #include <fstream> //File I/O |
|  | #include <iomanip> //Format Library |
|  | #include <cmath> //Math Library |
|  | #include <sstream> //Stream Library |
|  | using namespace std; //Name-space under which system libraries exist |
|  |  |
|  | //User Libraries |
|  |  |
|  | //Global Constants |
|  |  |
|  | //Function Prototypes |
|  | int d4(int);//Roll 4-sided die x times |
|  | int d6(int);//Roll 6-sided die x times |
|  | int d8(int);//Roll 8-sided die x times |
|  | int d10(int);//Roll 10-sided die x times |
|  | int d12(int);//Roll 12-sided die x times |
|  | int d20(int);//Roll 20-sided die x times |
|  | int d20();//Roll a 20-sided die. |
|  | bool check(int,int,int);//Set check difficulty,add bonuses, test for success |
|  | int abltmod(int);//determine modifier from ability scores |
|  | int dmg(int,int,int);//Roll + Ability Modifier + Bonus |
|  | int rapier(); |
|  | int dagger(); |
|  | int fists(); |
|  | int combat(int&,int&,int,int&,int,int,int,int&,int,int,bool&,bool&); |
|  | int heroTrn(int&,int&,int,int&,int,int,int,int&,int,int,bool&,bool&); |
|  | int enmyTrn(int&,int&,int,int&,int,int,int,int&,int,int,bool&,bool&); |
|  | int hlthpot(int&,int&,int&); |
|  | bool guesNum(int&,float&,int,int,int&);//Guess the number |
|  | void scorSrt(int[],string[],int,string,int);//Sorting High Scores |
|  | void play (int&,int&,int&,char&);//gold,health,potions,action |
|  | void lvfool(int&,int&,int,int,int,int&,int,int,int&,int&,bool,bool&,string); |
|  | //gold,health,max health,armor,finesse weapon,potions,dexterity mod, |
|  | //sleight of hand,beers,ring,helped,victory,helper name |
|  | void drnkCnt (int&,int&);//gold,beers |
|  | void buybeer (int&,int&);//gold,beers |
|  | void getRing(int&,int&,int,int,int,int&,int,int,int&,int&,bool,bool&,string); |
|  | //gold,health,max health,armor,finesse weapon,potions,dexterity mod, |
|  | //sleight of hand,beers,ring,helped,victory,helper name |
|  | void wlkaway(int&,int,int&); |
|  | void cmbtend(int&,int,int,bool&,bool&,bool&,string); |
|  | void border(); |
|  | void end(int,string); |
|  |  |
|  | //Function with Default argument |
|  | //Determine player initiative in combat |
|  | int intiatv(int roll,int abltmod=0){ |
|  | roll+abltmod; |
|  | } |
|  |  |
|  | //Execution begins here |
|  | int main(int argc, char\*\* argv) { |
|  | //Setting the random number seed |
|  | srand(static\_cast<unsigned int>(time(0))); |
|  |  |
|  | //Declare and Initialize variables |
|  | char action=0; |
|  | //Character Ability Scores |
|  | const int str=10,//Strength |
|  | dex=16,//Dexterity |
|  | con=10,//Constitution |
|  | intl=14,//Intellect |
|  | wis=13,//Wisdom |
|  | cha=14;//Charisma |
|  |  |
|  | //Ability Modifiers |
|  | int strmod=abltmod(str); |
|  | int dexmod=abltmod(dex); |
|  | int conmod=abltmod(con); |
|  | int intmod=abltmod(intl); |
|  | int wismod=abltmod(wis); |
|  | int chamod=abltmod(cha); |
|  |  |
|  | //Character Proficiencies |
|  | int profbns=3;//Proficiency Bonus |
|  | int athltcs=profbns\*1, //Athletics |
|  | acrbtcs=profbns\*1, //Acrobatics |
|  | sltohnd=profbns\*2, //Sleight of Hand (double proficiency) |
|  | stlth=profbns\*1, //Stealth |
|  | arcana=profbns\*0, //Arcana |
|  | history=profbns\*0, //History |
|  | invstgt=profbns\*0, //Investigate |
|  | nature=profbns\*0, //Nature |
|  | relgion=profbns\*0, //Religion |
|  | anmlhnd=profbns\*0, //Animal Handling |
|  | insight=profbns\*0, //Insight |
|  | medicn=profbns\*0, //Medicine |
|  | prcptn=profbns\*1, //Perception |
|  | survl=profbns\*0, //Survival |
|  | decptn=profbns\*2, //Deception (double proficiency) |
|  | intmdtn=profbns\*0, //Intimidation |
|  | prfmnc=profbns\*0, //Performance |
|  | persn=profbns\*2, //Persuasion (double proficiency) |
|  | fweapon=profbns\*1, //Finesse Weapon |
|  | rweapon=profbns\*1, //Ranged Weapon |
|  | mweapon=profbns\*1, //Marshal Weapon |
|  | gweapon=profbns\*0; //Great/2-hand Weapon |
|  |  |
|  |  |
|  | //Character Features |
|  | const int maxhlth=20;//Character's Health |
|  | int health=maxhlth;//Character's current health |
|  | int gold=50;//Starting gold |
|  | int armor=16;//Character's Armor Class |
|  | string name;//Character's name |
|  | string helper;//Name of contracted help |
|  |  |
|  | //Enemy Features |
|  | int mobhlth=0; |
|  | int mobAC=0; |
|  | int cmbtmod=0; |
|  |  |
|  | //Environmental Features |
|  | bool helped=false; |
|  | int beers=0;//Beers drank |
|  | int potions=3;//Health potions in inventory |
|  | float table=500;//Available gold in the Inn |
|  | int ring=1;//Limits the occurrence of the ring theft |
|  | bool victory=true;//Condition set by combat resolution |
|  | int wager=0; |
|  | bool fight=false;//Prevents player from returning to the gambling table |
|  | //After fighting. |
|  | //I/O Files |
|  | string instrng=""; |
|  |  |
|  | //Introduction |
|  | border(); |
|  | ifstream intro; |
|  | intro.open("intro.txt"); |
|  | while(!intro.eof()){ |
|  | getline(intro, instrng); |
|  | cout<<instrng<<endl; |
|  | } |
|  | intro.close(); |
|  | border(); |
|  | cout<<setw(100)<<"What name will you be going by this evening? (no spaces)"<<endl; |
|  | getline(cin,name); |
|  | cout<<setw(100)<<"Okay '"+name+"' and what is the name of your friend here? (no spaces)"<<endl; |
|  | getline(cin,helper); |
|  | cout<<setw(100)<<"Great well then Masters "+name+" and "+helper+" I will"<<endl; |
|  | cout<<setw(100)<<"set your things in your room upstairs."<<endl; |
|  | cout<<setw(100)<<"Please make yourselves at home in our humble common room"<<endl; |
|  | border(); |
|  | //Begin Play |
|  | ifstream cmnroom; |
|  | cmnroom.open("cmnroom.txt");//Description of common room |
|  | while(!cmnroom.eof()) |
|  | { |
|  | getline(cmnroom,instrng); |
|  | cout <<instrng<<endl; |
|  | } |
|  | cmnroom.close(); |
|  | border(); |
|  | cout<<endl; |
|  |  |
|  | do{ |
|  | play (gold,health,potions,action);//The main decision to get to each branch |
|  |  |
|  | switch(action){ |
|  | case '1':{//Gambling |
|  | if(fight==true){ |
|  | cout<<"There is no one gambling, the fight made " |
|  | "everyone move on to new things for the evening." |
|  | <<endl; |
|  | table=0; |
|  | } |
|  | if (table<=0&&fight==false){//Prevents taking money from house that isn't there |
|  | cout<<setw(100)<<"There is no money at the table, " |
|  | "haven't you done" |
|  | " enough?"<<endl; |
|  | } |
|  | guesNum(gold,table,chamod,decptn,wager); |
|  | if (gold<0){//You are fighting the gambler (Health=20,mod=1,AC=14) |
|  | gold=abs(gold);//Return gold to positive if negative |
|  | gold--;//resets gold to true value. |
|  | mobhlth=20; |
|  | cmbtmod=1; |
|  | mobAC=14; |
|  | combat(gold,health,maxhlth,potions,dexmod,armor, |
|  | fweapon,mobhlth,cmbtmod,mobAC,victory,helped); |
|  | //Player's Gold, Current Health, Max Health, # of Potions, |
|  | //Combat Modifier, and Armor Class; |
|  | //Mob's Health, Combat Modifier, and Armor Class; |
|  | cmbtend(gold,health,wager,victory,helped,fight,helper); |
|  | }break; |
|  | } |
|  | case '2':{//Step up to the bar |
|  | beers=0; |
|  |  |
|  | if (ring<1){ |
|  | cout<<"There is nothing of interest at the bar."<<endl; |
|  | }else{ |
|  | border(); |
|  | ifstream loverboy; |
|  | loverboy.open("loverboy.txt"); |
|  | while(!loverboy.eof()) |
|  | { |
|  | getline(loverboy,instrng); |
|  | cout <<instrng<<endl; |
|  | } |
|  | loverboy.close(); |
|  | border(); |
|  | do{ |
|  | drnkCnt (gold,beers); |
|  | lvfool(gold,health,maxhlth,armor,fweapon,potions, |
|  | dexmod,sltohnd,beers,ring,helped,victory, |
|  | helper); |
|  | }while (ring>=1&&gold>0); |
|  | } |
|  |  |
|  | }break; |
|  | default:{ |
|  | cout<<"After a long night you decide you've had you fill of the" |
|  | " inn."<<endl; |
|  | cout<<"You gather your things and head up stairs to" |
|  | " your room."<<endl; |
|  | victory=false;//Ends Game |
|  | } |
|  | } |
|  | }while(health>0&&victory==true); |
|  | end(gold,name); |
|  |  |
|  | //Exit stage right! |
|  | return 0; |
|  | } |
|  | //End Display |
|  |  |
|  | void end(int gold,string name){ |
|  | cout<<"You end the night with "<<gold<<" gold."<<endl; |
|  | border(); |
|  | cout<<setw(50)<<"GAME OVER"<<endl; |
|  | ofstream scorBrd; |
|  | scorBrd.open ("ScorBrd.txt",ios::app); |
|  | scorBrd <<setw(20)<<name<<setw(20)<<gold<<endl; |
|  | scorBrd.close(); |
|  | cout<<endl; |
|  |  |
|  | //Sort and display leader board |
|  | border(); |
|  | cout<<setw(51)<<"High Scores"<<endl; |
|  | cout<<setw(33)<<"Name"<<setw(30)<<"Gold Total"<<endl; |
|  | ifstream showBrd; |
|  | int SIZE=100; |
|  | string names[SIZE]={}; |
|  | int scores[SIZE]={}; |
|  | int scrCnt[SIZE]={}; |
|  | showBrd.open("ScorBrd.txt"); |
|  |  |
|  | //Fill Array from file |
|  | for (int i=0;i<SIZE;i++){ |
|  | showBrd>>names[i]; |
|  | showBrd>>scores[i]; |
|  | } |
|  | showBrd.close(); |
|  | //Sort and Display Leader Board |
|  | scorSrt(scores,names,SIZE,name,gold); |
|  | for(int i=0;i<10;i++){ |
|  | scrCnt[i]=i+1; |
|  | cout<<setw(2)<<scrCnt[i]; |
|  | cout<<setw(35)<<names[i]; |
|  | cout<<setw(25)<<scores[i]; |
|  | if (names[i]==name&&scores[i]==gold){ |
|  | cout<<" \*\*\*\*\*NEW HIGH SCORE\*\*\*\*\*"; |
|  | } |
|  | cout<<endl; |
|  | } |
|  | border(); |
|  | } |
|  | //Sorting High scores |
|  | void scorSrt(int a[],string b[],int n,string,int){ |
|  | for(int i=0;i<n-1;i++){ |
|  | for(int j=i+1;j<n;j++){ |
|  | if(a[i]<a[j]){ |
|  | int temp1=0; |
|  | string temp2=""; |
|  | temp1=a[i]; |
|  | a[i]=a[j]; |
|  | a[j]=temp1; |
|  | temp2=b[i]; |
|  | b[i]=b[j]; |
|  | b[j]=temp2; |
|  |  |
|  | } |
|  | } |
|  | } |
|  | } |
|  | //Play |
|  | void play (int &gold,int &health,int &potions,char &action){//gold,health,potions,victory,fight |
|  | cout<<endl; |
|  | cout<<"|Health : "<<health<<"|Gold : "<<gold; |
|  | cout<<"|Potions : "<<potions<<endl; |
|  | cout<<setw(100)<<"What would you like to do?"<<endl; |
|  | cout<<setw(100)<<"Look for a game for gambling, though what" |
|  | " you do could hardly be called gambling. - PRESS 1"<<endl; |
|  | cout<<setw(100)<<"Step up to the bar and buy a drink. - Press 2"; |
|  | cout<<endl; |
|  | cout<<setw(100)<<"Call it a night. - Press 3"; |
|  | cout<<endl; |
|  | cout<<"Your choice of ACTION : "; |
|  | cin>>action; |
|  | while(action>'3'||action<'1'){ |
|  | cout<<"Try again."<<endl; |
|  | cout<<"Your choice of ACTION : "; |
|  | cin>>action; |
|  | } |
|  | border(); |
|  | } |
|  |  |
|  | //Scenarios: |
|  |  |
|  | //The Love Fool Scenario |
|  | void lvfool(int &gold,int &health,int maxhlth,int armor,int fweapon, |
|  | int &potions,int dexmod,int sltohnd,int &beers,int &ring, |
|  | bool helped,bool &victory,string helper){ |
|  | char action='0'; |
|  | cout<<"|Health : "<<health<<"|Gold : "<<gold; |
|  | cout<<"|Potions : "<<potions<<endl; |
|  | cout<<setw(100)<<"Would you like to buy a round of drinks for 5g?" |
|  | " - PRESS 1"<<endl; |
|  | cout<<setw(100)<<"Attempt to swipe the ring? - PRESS 2"<<endl; |
|  | cout<<setw(100)<<"Or do you want to just walk away?" |
|  | " - PRESS 3"<<endl; |
|  | cin>>action; |
|  | switch (action){ |
|  | case '1':{ |
|  | buybeer(gold,beers); |
|  | }break; |
|  | case '2':{ |
|  | getRing(gold,health,maxhlth,armor,fweapon,potions, |
|  | dexmod,sltohnd,beers,ring,helped,victory,helper); |
|  | }break; |
|  | default:{ |
|  | wlkaway(gold,beers,ring); |
|  | }break; |
|  | } |
|  | } |
|  | void buybeer (int &gold, int &beers){ |
|  | if (beers==4){ |
|  | cout<<setw(100)<<"The bartender looks at you scornfully and " |
|  | "declines to pour your friend another drink." |
|  | <<endl; |
|  | }else{ |
|  | beers++; |
|  | gold-=5; |
|  | } |
|  | } |
|  | void drnkCnt (int &gold,int &beers){ |
|  | switch (beers){ |
|  | case 0:{ |
|  | cout<<setw(100)<<"Perhaps a nice cold ale" |
|  | " will help calm his nerves."<<endl;break; |
|  | } |
|  | case 1:{ |
|  | cout<<"You buy a round for the two of you."<<endl; |
|  | cout<<setw(100)<<"He seems to calm down and be more at ease."<<endl; |
|  | break; |
|  | } |
|  | case 2:{ |
|  | cout<<"You buy another round."<<endl; |
|  | cout<<setw(100)<<"He's a little tipsy but no cause for worry."<<endl; |
|  | break; |
|  | } |
|  | case 3:{ |
|  | cout<<"How about another! "<<endl; |
|  | cout<<setw(100)<<"'A terst!' he shouts!'Tuh th must butufl wemn " |
|  | "IN th werld!'"<<endl; |
|  | cout<<setw(100)<<"Okay, he's drunk now and starting" |
|  | " to attract attention."<<endl;break; |
|  | } |
|  | case 4:{ |
|  | cout<<"One more couldn't hurt..."<<endl; |
|  | cout<<setw(100)<<"Oh that did it! He passed out. As he falls"<<endl; |
|  | cout<<setw(100)<<"to the ground you pick him up and set him " |
|  | "back on the bar."<<endl; |
|  | cout<<setw(100)<<"Of course you also nicked the ring " |
|  | "in the process, hopefully"<<endl; |
|  | cout<<setw(100)<<"its worth the gold in beer you spent."<<endl; |
|  | break; |
|  | } |
|  | default: |
|  | cout<<"He's still passed out."<<endl; |
|  | } |
|  | } |
|  | void getRing(int &gold,int &health,int maxhlth,int armor,int fweapon, |
|  | int &potions,int dexmod,int sltohnd,int &beers,int &ring, |
|  | bool helped,bool &victory,string helper){ |
|  | if(ring<=0){ |
|  | cout<<"You already grabbed it, don't linger"<<endl; |
|  | } |
|  | if(check(20-beers\*2,dexmod,sltohnd)==true){ |
|  | ring--; |
|  | int worth=d20()\*5; |
|  | cout<<"The ring was worth "<<worth<<" gold."<<endl; |
|  | gold+=worth; |
|  | }else{//Combat!!!You're fighting Lover-Boy (health=14,mod=0,ac=12) |
|  | int mobhlth=14; |
|  | int cmbtmod=0; |
|  | int mobAC=12; |
|  | combat(gold,health,maxhlth,potions,dexmod, |
|  | armor,fweapon,mobhlth,cmbtmod,mobAC,victory,helped); |
|  | //Player's Gold, Current Health, Max Health, # of Potions, |
|  | //Combat Modifier, and Armor Class; |
|  | //Mob's Health, Combat Modifier, and Armor Class; |
|  | if (victory==false&&health>0){ |
|  | cout<<"Running from a fight is not necessarily a bad decision but it" |
|  | " will end your night at least you "<<endl; |
|  | cout<<"kept the gold on your person. You cannot go back into the in " |
|  | "for the shame would be too much."<<endl; |
|  | cout<<"You eventually put together a workable disguise and return to" |
|  | " the room you rented."<<endl; |
|  | ring--; |
|  | } |
|  | else if (health<=0&&victory==false){ |
|  | cout<<"You lost the fight and were thrown out of the inn"<<endl; |
|  | cout<<"You lost half of your money in the exchange."<<endl; |
|  | gold/=2; |
|  | ring--; |
|  | } |
|  | else { |
|  | cout<<"You beat the kid up for catching you in the act " |
|  | "this is not what we would call a high moment for you."<<endl;; |
|  | ring--; |
|  | int worth=d20()\*5; |
|  | cout<<"The ring was worth "<<worth<<" gold."<<endl; |
|  | gold+=worth; |
|  | cout<<"What just occurred is not exactly a mystery to the patrons " |
|  | "and you are 'asked' to leave."<<endl; |
|  | victory=false; |
|  | } |
|  | if (helped==true){ |
|  | gold-=10; |
|  | cout<<"You pay "+helper+" 10g."<<endl; |
|  | } |
|  | helped=false; |
|  | } |
|  | } |
|  |  |
|  | void wlkaway(int &gold,int beers, int &ring){ |
|  | if (beers>=4){ |
|  | cout<<setw(100)<<"The smart thing to do now is remove " |
|  | "yourself from the situation"<<endl; |
|  | cout<<setw(100)<<"and avoid further notice."<<endl; |
|  | ring--; |
|  | int worth=d20()\*5; |
|  | cout<<"The ring was worth "<<worth<<" gold."<<endl; |
|  | gold+=worth; |
|  | }else{ |
|  | border(); |
|  | string instrng=""; |
|  | ifstream proposal; |
|  | proposal.open("loverboy.txt"); |
|  | while(!proposal.eof()) |
|  | { |
|  | getline(proposal,instrng); |
|  | cout <<instrng<<endl; |
|  | } |
|  | proposal.close(); |
|  | border(); |
|  | ring--; |
|  | } |
|  | } |
|  |  |
|  | //Gambling Games |
|  | bool guesNum(int &gold,float &table,int chamod,int decptn,int& wager){ |
|  | //Character's Gold, Possible gold at the table |
|  | char stay=1; |
|  | int beer=0; |
|  | int wins=0,losses=0; |
|  | char num; |
|  | char action='0'; |
|  | if(table<=0){ |
|  | cout<<setw(100)<<"..."<<endl; |
|  | }else if (gold<=0){ |
|  | cout<<setw(100)<<"You can't gamble if you don't have anything to " |
|  | "wager, move along."<<endl; |
|  | }else{ |
|  | border(); |
|  | string instrng=""; |
|  | ifstream gambling; |
|  | gambling.open("gambling.txt"); |
|  | while(!gambling.eof()) |
|  | { |
|  | getline(gambling,instrng); |
|  | cout <<instrng<<endl; |
|  | } |
|  | gambling.close(); |
|  | border(); |
|  | do{ |
|  | cout<<"|Table : "<<table<<"|Gold : "<<gold<<"|Beers : "<<beer<<" |Wins : "<<wins<<" |Losses : " |
|  | <<losses<<"|"<<endl; |
|  | cout<<setw(100)<<"Would you like to buy a round for the table for 5g?"<<endl; |
|  | cout<<setw(100)<<"Yes - PRESS 1"<<endl; |
|  | cout<<setw(100)<<"No - PRESS 2"<<endl; |
|  | cout<<"Buy Round? "; |
|  | cin>>action; |
|  | if (action=='1'){ |
|  | cout<<"You buy a round of ale for the table."<<endl; |
|  | cout<<setw(100)<<"The table of risk takers thank you kindly and" |
|  | " urge you on to the next game."<<endl; |
|  | beer++; |
|  | gold-=5; |
|  | }else{ |
|  | cout<<"You politely decline the suggestion and things continue as" |
|  | " they were."<<endl; |
|  | } |
|  | cout<<setw(100)<<"Make a wager"<<endl; |
|  | cout<<"Wager : "; |
|  | cin>>wager; |
|  | while(!(wager<=table)||!(wager<=gold)||!(wager>0)||(cin.fail())){ |
|  | cin.clear();//clears error thrown by cin.fail |
|  | cin.ignore();//allows acceptance of next entry |
|  | wager=0; |
|  | cout<<setw(100)<<"That's not a real wager. " |
|  | "Try a different wager."<<endl; |
|  | cout<<"Wager : "; |
|  | cin>>wager; |
|  | } |
|  | gold-=wager;//Player Ante |
|  | table-=wager;//Table Ante |
|  | cout<<setw(100)<<"Okay pal, now pick a number between 1-3 and lets" |
|  | " see if we can guess it"<<endl; |
|  | cout<<"Your number : "; |
|  | cin>>num; |
|  | while(num>'3'||num<'1'){ |
|  | cout<<setw(100)<<"Come on, pick a number between 1 and 3, this isn't the " |
|  | "part you get to cheat on."<<endl; |
|  | cout<<"Your number : "; |
|  | cin>>num; |
|  | } |
|  | int guess=rand()%3+1; |
|  | cout<<setw(98)<<"Is your number "<<static\_cast<char>(guess+48)<<"?"<<endl; |
|  | if(!(num==static\_cast<char>(guess+48))){ |
|  | gold+=wager\*2; |
|  | wins++; |
|  | cout<<"No."<<endl; |
|  | cout<<"You won honestly...neat! Now you have "<<gold; |
|  | cout<<" gold."<<endl; |
|  | }else{ |
|  | cout<<setw(100)<<"Do you want to tell the truth or lie?"<<endl; |
|  | float dfclty=((10-beer+wins-losses)\*(1+wager/(table+1))); |
|  | cout<<setw(98)<<"Check Difficulty is : "<<fixed; |
|  | cout<<setprecision(0)<<dfclty<<endl; |
|  | cout<<setw(100)<<"Lie - PRESS 1"<<endl; |
|  | cout<<setw(100)<<"Tell Truth - PRESS 2"<<endl; |
|  | cin>>action; |
|  | if (action=='2'){ |
|  | cout<<"You admit the truth and share with them that they" |
|  | " guessed correctly, your pockets are lighter"<<endl; |
|  | cout<<"but so is you hearth for all that will get you."<<endl; |
|  | table+=wager\*2; |
|  | losses++; |
|  | cout<<"Now you have "<<gold<<" gold."<<endl; |
|  | }else{ |
|  | cout<<"You decided to lie, of course you did. Simple games require" |
|  | " simple lies, you respond with a simple 'No'"<<endl; |
|  | /\*Difficulty of deception check rises when a larger percentage |
|  | \* of the table's money is at stake and when you have won several |
|  | \* times. Losing occasionally lowers their suspicion, but so does |
|  | \* beer. |
|  | \*/ |
|  | if (check(dfclty,chamod,decptn)==true){ |
|  | cout<<"The gamblers are disappointed but they believe you."<<endl; |
|  | gold+=wager\*2; |
|  | wins++; |
|  | cout<<"Now you have "<<gold<<" gold."<<endl; |
|  | }else{ //Combat!!! |
|  | cout<<setw(100)<<"As convincing as you typically are they don't " |
|  | "buy it. The biggest of the bunch"<<endl; |
|  | cout<<setw(100)<<"flips the table over and attacks you."<<endl; |
|  | gold++;//Prevents gold from being '0' which would |
|  | //avoid combat |
|  | gold\*=(-1);//Makes gold value negative so that combat |
|  | //triggers. |
|  | } |
|  |  |
|  | } |
|  |  |
|  | } |
|  | if (table<=0){ |
|  | cout<<setw(100)<<"'Well that's it for us!'"<<endl; |
|  | cout<<setw(100)<<"There is no more gold to be won here," |
|  | "time to move on."<<endl; |
|  | } |
|  | if (gold>0&&table>0){ |
|  | cout<<setw(100)<<"Do you want to stay at the table?"<<endl; |
|  | cout<<setw(100)<<"Stay - PRESS 1"<<endl; |
|  | cout<<setw(100)<<"Leave - PRESS 2"<<endl; |
|  | cin>>stay; |
|  | } |
|  | }while (stay=='1'&&gold>0&&table>0); |
|  | } |
|  | return gold,table,wager; |
|  | } |
|  | void cmbtend(int &gold,int health,int wager,bool &victory,bool &helped,bool &fight, |
|  | string helper){ |
|  | if (victory==false&&health>0){ |
|  | cout<<"Running from a fight is not necessarily a bad decision but it" |
|  | " will end your night at least you "<<endl; |
|  | cout<<"kept the gold on your person. You cannot go back into the in " |
|  | "for the shame would be too much."<<endl; |
|  | cout<<"You eventually put together a workable disguise and return to" |
|  | " the room you rented."<<endl; |
|  | } |
|  | else if (health<=0&&victory==false){ |
|  | cout<<"You lost the fight and were thrown out " |
|  | "of the inn"<<endl; |
|  | cout<<"You lost half of your money in the " |
|  | "exchange."<<endl; |
|  | gold/=2; |
|  | }else { |
|  | cout<<"You defeated the suspicious gamblers. " |
|  | "Collect what gold you can find on the" |
|  | " ground and go about your business."<<endl; |
|  | cout<<"Fights over gambling are common and " |
|  | "while everyone is a bit wary of you," |
|  | " no one is giving you trouble."<<endl; |
|  | int find=(wager\*2-wager/d20());//Randomize gold recovered just the a DM would |
|  | gold+=find; |
|  | cout<<"You find "<<find<<" gold and now have " |
|  | <<gold<<" total."<<endl; |
|  | } |
|  | if (helped==true){ |
|  | gold-=10; |
|  | cout<<"You pay "+helper+" 10g."<<endl; |
|  | } |
|  | fight=true; |
|  | helped=false; |
|  | } |
|  | //If Combat is necessary |
|  | int combat(int& gold,int& health,int maxhlth,int& potions,int abltmod,int plyrAC,int weapon, |
|  | int &mobhlth,int cbtmod,int mobAC,bool &victory,bool &helped){ |
|  | //roll initiative |
|  |  |
|  | char move='0'; |
|  | int pInitv=intiatv(d20(),abltmod);//player initiative plus dexterity |
|  | int mInitv=intiatv(d20());//mob initiative (no dex bonus because inn mobs are not special |
|  | if (pInitv>mInitv){//Player goes first |
|  | do{ |
|  | heroTrn(gold,health,maxhlth,potions,abltmod,plyrAC,weapon,mobhlth,cbtmod,mobAC,victory,helped); |
|  | enmyTrn(gold,health,maxhlth,potions,abltmod,plyrAC,weapon,mobhlth,cbtmod,mobAC,victory,helped); |
|  | }while (health>0&&mobhlth>0); |
|  | }else{ |
|  | do{ |
|  | enmyTrn(gold,health,maxhlth,potions,abltmod,plyrAC,weapon,mobhlth,cbtmod,mobAC,victory,helped); |
|  | heroTrn(gold,health,maxhlth,potions,abltmod,plyrAC,weapon,mobhlth,cbtmod,mobAC,victory,helped); |
|  | if (health<=0)victory=false; |
|  | }while (health>0&&mobhlth>0&&victory==true); |
|  | } |
|  | abs(gold); |
|  | return gold,health,potions,victory; |
|  | } |
|  | int heroTrn(int &gold,int &health,int maxhlth,int& potions,int abltmod,int plyrAC,int weapon, |
|  | int &mobhlth,int cbtmod,int mobAC,bool &victory,bool &helped){ |
|  | char move='0'; |
|  | //Player Move |
|  | if(health>0){ |
|  | cout<<"COMBAT"; |
|  | cout<<"|Health : "<<health<<"|Gold : "<<gold; |
|  | cout<<"|Potions : "<<potions<<"|Enemy's health : "; |
|  | cout<<mobhlth<<endl; |
|  | cout<<setw(94)<<"Make a decision."<<endl; |
|  | cout<<setw(100)<<"Attack with my sword - PRESS 1"<<endl; |
|  | cout<<setw(100)<<"Attack with my dagger - PRESS 2"<<endl; |
|  | cout<<setw(100)<<"Run away! - PRESS 3"<<endl; |
|  | cout<<setw(100)<<"Signal to your helper for assistance - PRESS 4"<<endl; |
|  | cout<<setw(100)<<"Use a health Potion - PRESS 5"<<endl; |
|  | cin>>move; |
|  | switch (move){ |
|  | case '1':{//Sword Attack |
|  | if(check(mobAC,abltmod,weapon)==true){ |
|  | int total=0; |
|  | if (helped==true){ |
|  | total=d6(3)+//Sneak Attack |
|  | d4(1);//Helper's Damage |
|  | } |
|  | int damage=dmg(total,abltmod,rapier()); |
|  | cout<<"You hit dealing "<<damage<<" damage."<<endl; |
|  | mobhlth-=damage; |
|  | }else cout<<"You missed"<<endl; |
|  | break; |
|  | } |
|  | case '2':{//Dagger Attack |
|  | if(check(mobAC,abltmod,weapon)==true){ |
|  | int total=0; |
|  | if (helped==true){ |
|  | total=d6(3)+//Sneak Attack |
|  | d4(1)+2;//Helper's Damage |
|  | } |
|  | int damage=dmg(total,abltmod,rapier()); |
|  | cout<<"You hit dealing "<<damage<<" damage."<<endl; |
|  | mobhlth-=damage; |
|  | }else cout<<"You missed"<<endl; |
|  | break; |
|  | } |
|  | case '3':{//Run away little girl, run away |
|  | cout<<"You run out of the inn and everybody laughs at you"; |
|  | cout<<endl; |
|  | victory=false; |
|  | break; |
|  | } |
|  | case '4':{//Call for help |
|  | cout<<"You execute the agreed upon signal to call " |
|  | "for assistance"<<endl; |
|  | if(helped==true){cout<<"but your helper is already " |
|  | "involved, quit wasting time!"<<endl; |
|  | }else {cout<<"and your helper joins the fight."<<endl; |
|  | helped=true; |
|  | }break; |
|  | } |
|  | case '5':{//Use Health Potion |
|  | if(potions>=1){ |
|  | cout<<"You drink a health potion to regain some strength"; |
|  | cout<<endl; |
|  | hlthpot(health,maxhlth,potions); |
|  | cout<<"Your health is now "<<health<<endl; |
|  | }else{ |
|  | cout<<"You have no more potions."<<endl; |
|  | }break; |
|  | } |
|  | default: cout<<"You failed to make a real decision."<<endl; |
|  | } |
|  | }//End Player's move |
|  |  |
|  | } |
|  | int enmyTrn(int &gold,int &health,int maxhlth,int& potions,int abltmod,int plyrAC,int weapon, |
|  | int &mobhlth,int cbtmod,int mobAC,bool &victory,bool &helped){ |
|  | //Enemy's Move |
|  | if(mobhlth>0){ |
|  | if(rand()%10==0){ |
|  | cout<<"Your enemy picks up a chair and attempts to hit you with it"<<endl; |
|  | if(check(plyrAC,cbtmod,0)==true){ |
|  | int total=0; |
|  | int damage=dmg(total,cbtmod,d8(1)+3); |
|  | cout<<"You are hit and receive "<<damage<<" damage."<<endl; |
|  | health-=damage; |
|  | }else cout<<"Your enemy missed"<<endl; |
|  | }else{ |
|  | cout<<"Your enemy attempts to punch you..."; |
|  | if(check(plyrAC,cbtmod,0)==true){ |
|  | int total=0; |
|  | int damage=dmg(total,cbtmod,rapier()); |
|  | cout<<"you are hit and receive "<<damage<<" damage."<<endl; |
|  | health-=damage; |
|  | }else cout<<"but misses."<<endl; |
|  | } |
|  | } |
|  | } |
|  | //Mechanics and Rules |
|  | int d4(int nDice){ |
|  | int total=0; |
|  | for(int i=1;i<=nDice;i++){ |
|  | int die=rand()%4+1;//[1,4] |
|  | total+=die; |
|  | } |
|  | return total; |
|  | } |
|  | int d6(int nDice){ |
|  | int total=0; |
|  | for(int i=1;i<=nDice;i++){ |
|  | int die=rand()%6+1;//[1,6] |
|  | total+=die; |
|  | } |
|  | return total; |
|  | } |
|  | int d8(int nDice){ |
|  | int total=0; |
|  | for(int i=1;i<=nDice;i++){ |
|  | int die=rand()%8+1;//[1,8] |
|  | total+=die; |
|  | } |
|  | return total; |
|  | } |
|  | int d10(int nDice){ |
|  | int total=0; |
|  | for(int i=1;i<=nDice;i++){ |
|  | int die=rand()%10+1;//[1,10] |
|  | total+=die; |
|  | } |
|  | return total; |
|  | } |
|  | int d12(int nDice){ |
|  | int total=0; |
|  | for(int i=1;i<=nDice;i++){ |
|  | int die=rand()%12+1;//[1,12] |
|  | total+=die; |
|  | } |
|  | return total; |
|  | } |
|  | int d20(int nDice){ |
|  | int total=0; |
|  | for(int i=1;i<=nDice;i++){ |
|  | int die=rand()%20+1;//[1,20] |
|  | if (die==20){cout<<"CRITICAL ROLL!"<<endl;} |
|  | total+=die; |
|  | } |
|  | return total; |
|  | } |
|  | int d20(){ |
|  | return rand()%20+1;//[1,20] |
|  | } |
|  | bool check(int dfclty,int abltmod,int prof){ |
|  | bool check; |
|  | int roll=0; |
|  | roll=d20()+abltmod+prof; |
|  | cout<<endl; |
|  | if (roll==20){check=true; |
|  | } |
|  | else {(roll>=dfclty)?check=true:check=false;} |
|  | return check; |
|  | } |
|  | int abltmod(int ablty){ |
|  | int mod=(ablty-10)/2; |
|  | return mod; |
|  | } |
|  | int dmg(int roll,int mod,int bonus=0){ |
|  | int dmg=roll+mod+bonus; |
|  | return dmg; |
|  | } |
|  | int rapier(){ |
|  | return d8(1); |
|  | } |
|  | int dagger(){ |
|  | return d4(1); |
|  | } |
|  | int shrtbow(){ |
|  | return d6(1); |
|  | } |
|  | int fists(){ |
|  | return d4(1); |
|  | } |
|  | int hlthpot(int& health,int& maxhlth,int& potions){ |
|  | health+=d4(4); |
|  | if (health>=maxhlth)health=maxhlth; |
|  | potions--; |
|  | cout<<"You have "<<potions<<" left!"<<endl; |
|  | return health; |
|  | } |
|  | void border(){ |
|  | cout<<endl; |
|  | cout<<"/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/" |
|  | "\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/"<<endl; |
|  | cout<<endl; |
|  | } |

**For a clearer look at my code with proper formatting go to my** [**github**](https://github.com/arewhyain25/JohnsonRyan_CSC5_Summer2017/blob/master/Project_2_DungeonsAndDragons_v2.0/main.cpp)**.**