Project 1
<Munchkin Game>

CIS-5
Name: Ian Langley
Date: 2/6/20

Introduction

Title: Munchkin

Munchkin is a card game based around the classic rpg troupe of leveling up. 2+ players take turns “Kicking down the door”, revealing cards with monsters, curses, or other types of fantasy concepts. The goal of the game is to be the first player to reach level ten, primarily done through the defeat of monsters. After each monster is fought, they will provide a level and a number of treasures scaling based on the difficulty of the monster itself. In the full card based title, players can help and impede each other in their adventures resulting in a game more of social politics and backstabbing than dungeon delving.

Summary

Project size: 247 Lines For Main
449 Including unused function Prototypes
The number of variables: 26
The number of method: 0

Description

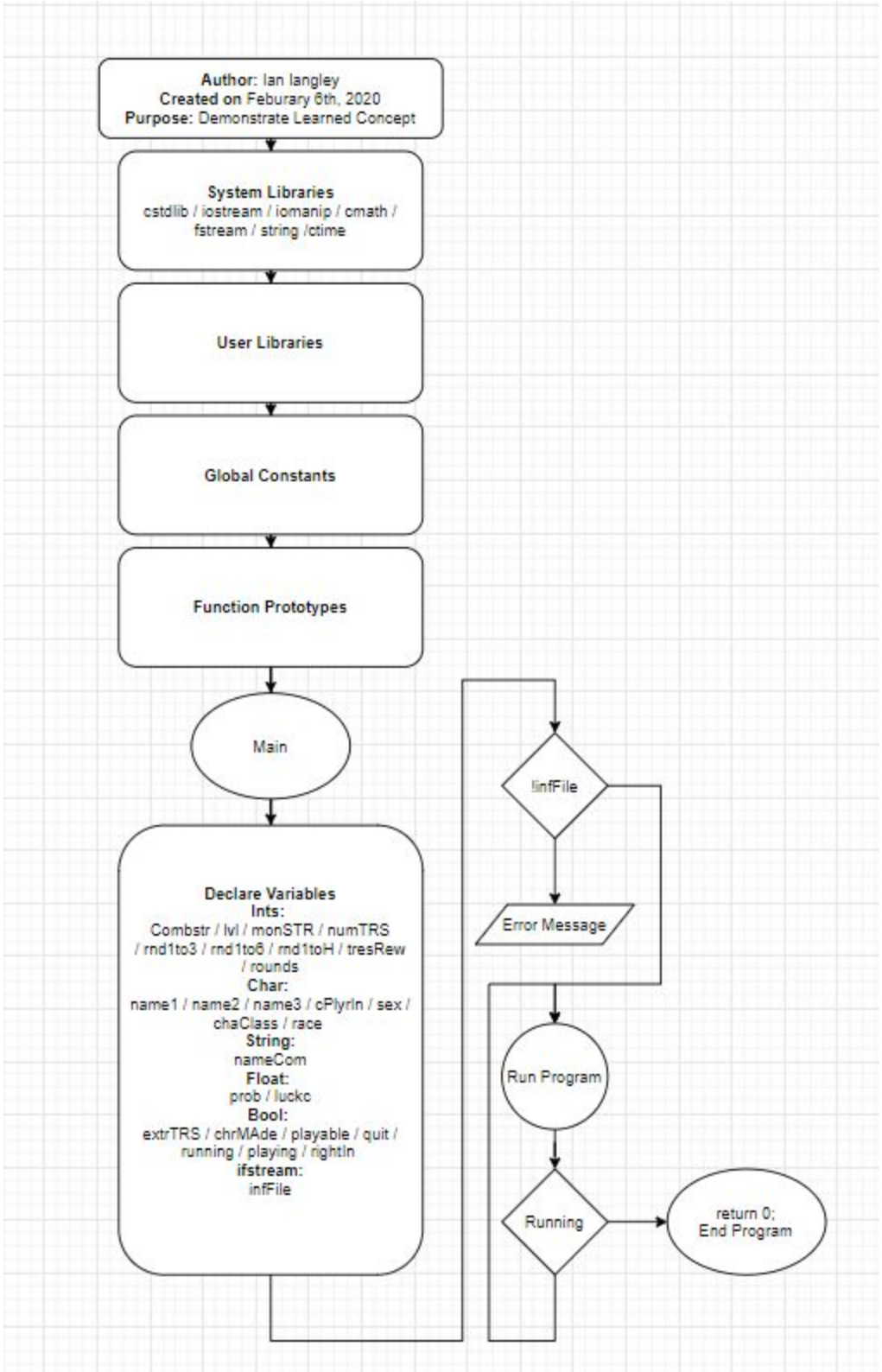
The main point of this program is to demonstrate the topics learned in the last five sections of this class. The program itself allows players to play an incredibly simplified version of the above described title. The program encompases the main feeling of the game, whilst using the topics learned, and allowing room for improvements, resulting in a great learning tool for myself.

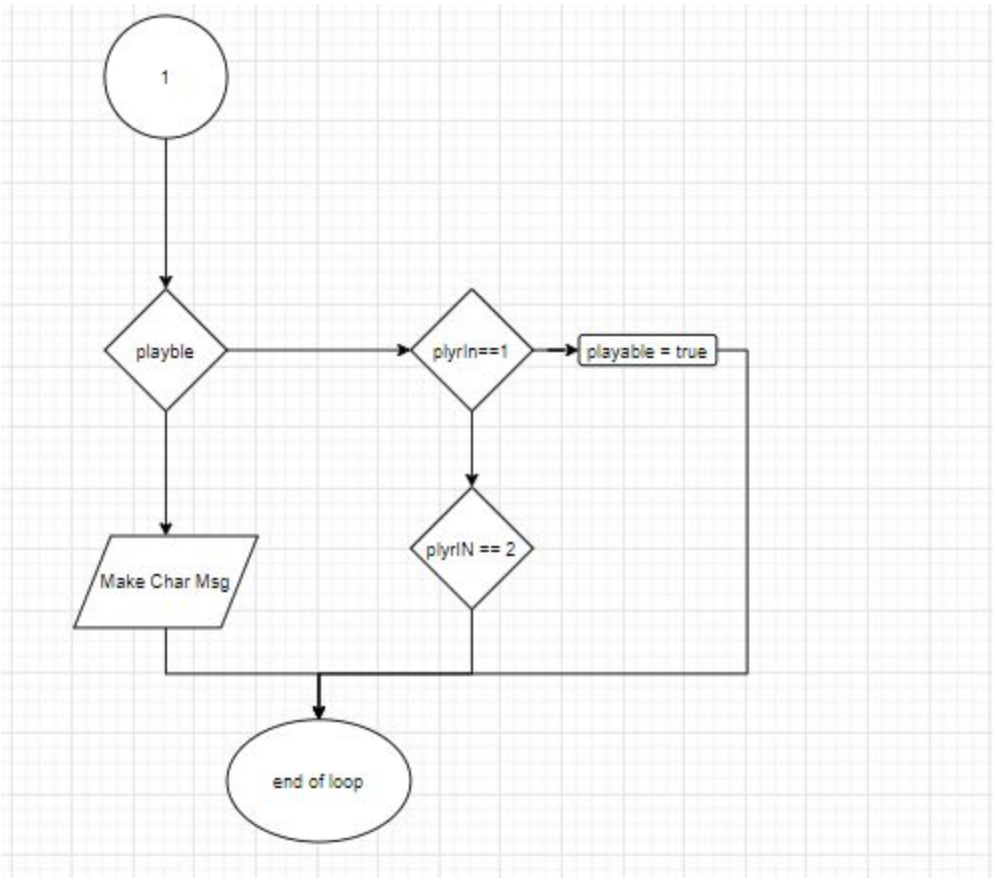
The project also includes several function prototypes that I used to create the initial structure of the program. These were late removed due to the no function creation requirement, but in future versions of this program, I will transition the majority of loops into these functions, allowing the main to be a more clear and precise style that is easy to read and follow.

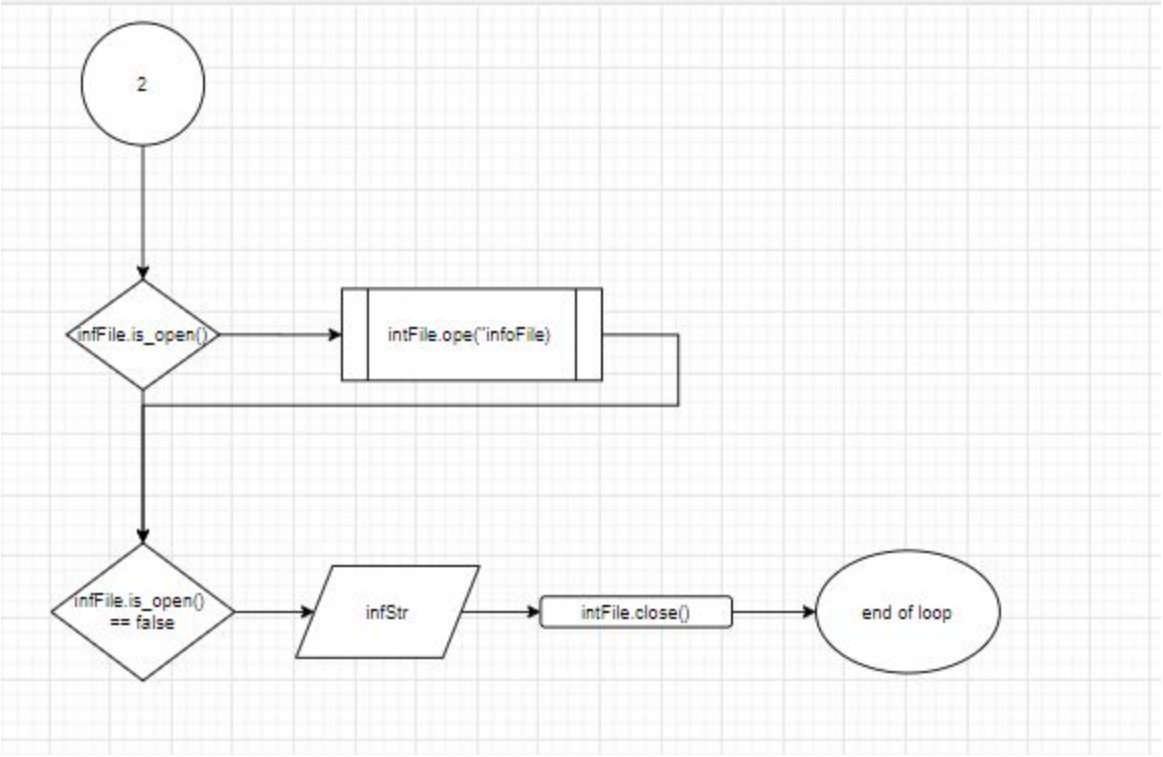
Version Documentation:(In order of creation)

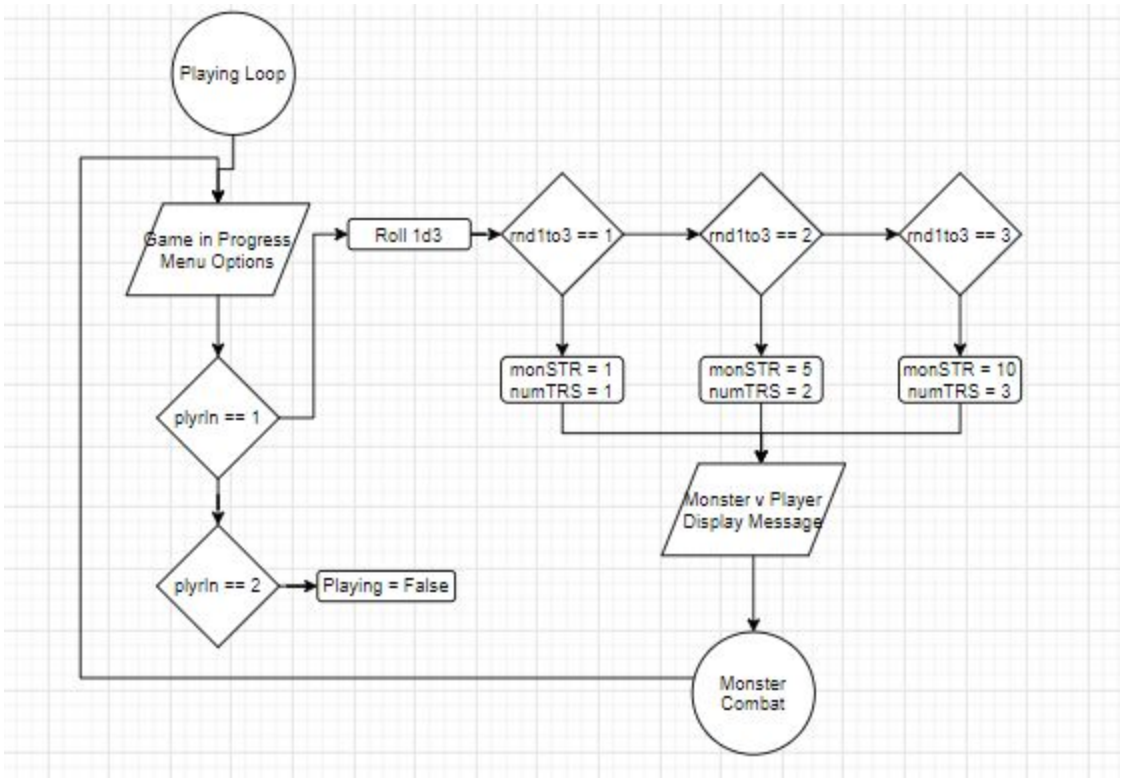
Version 1: Contains the main menu. Written in functions to later be translated
Combat Version 1: Contains the main combat loop. Would later be iterated on.
Version 2: Merges previous two versions. Allows for the player to initiate a game, no win condition yet.
Version 3: Adds final touches to combat and win condition. Can now effectively play the game. Diversifying the gameplay itself to be touched on in future versions.

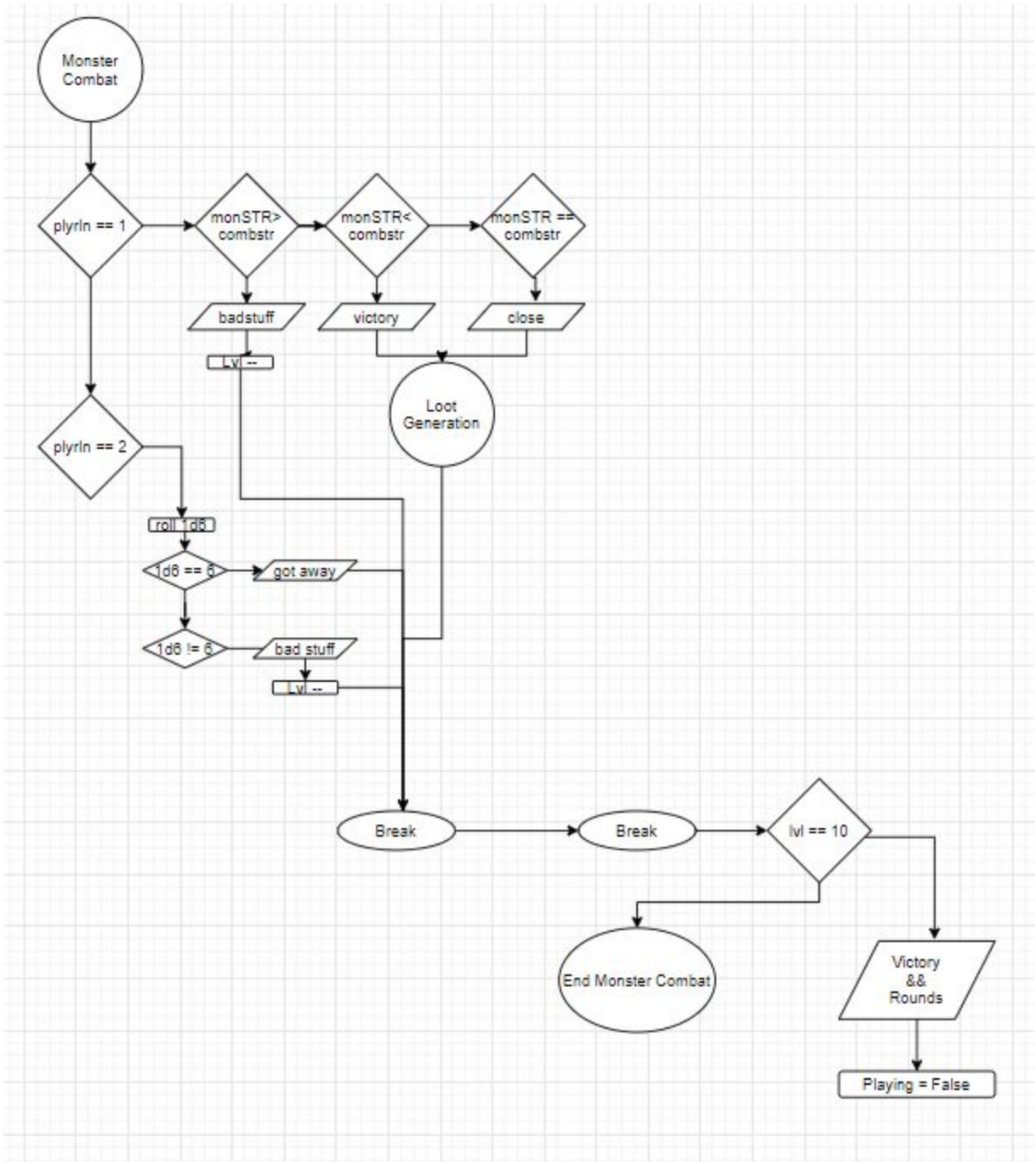
FlowChart

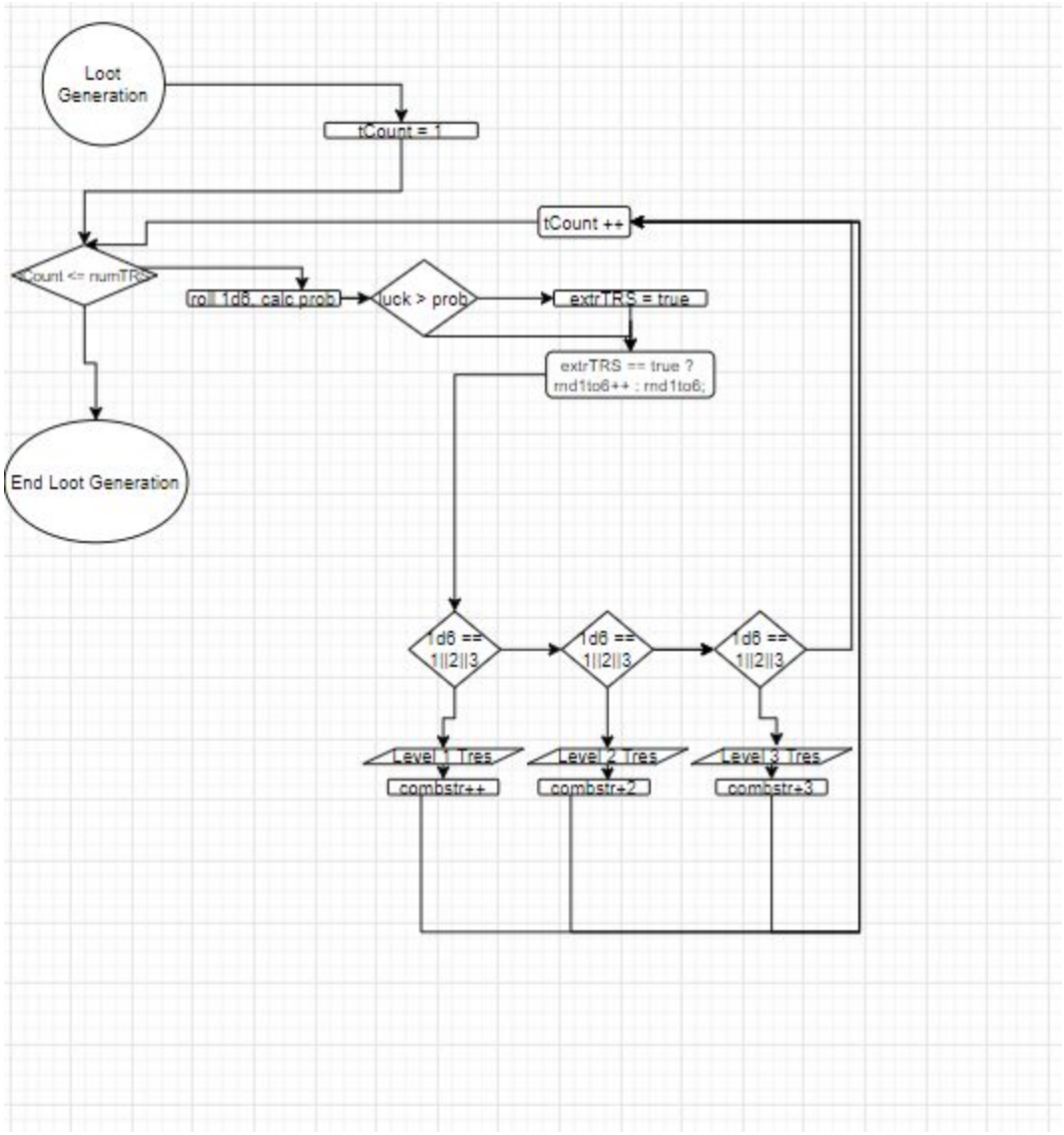

Author: Ian langley
Created on Feburary 8th, 2020
Purpose: Demonstrate Learned Concept
System Libraries
cstdlib / iostream / iomanip / cmath / fstream / string /ctime
User Libraries
Global Constants
Function Prototypes
Main
Declare Variables
Ints:
Combstr / lvl / monSTR / numTRS / rnd1to3 / rnd1to6 / rnd1toH / tresRew / rounds
Char:
name1 / name2 / name3 / cPlyrIn / sex / chaClass / race
String:
nameCom
Float:
prob / luckc
Bool:
extrTRS / chrMAde / playable / quit / running / playing / rightIn
ifstream:
infFile
!infFile
Error Message
Run Program
Running
return 0;
End Program


1
playble
plyrIn==1
playable = true
Make Char Msg
plyrIN == 2
end of loop


2
infFile.is_open()
intFile.ope("infoFile)
infFile.is_open()
== false
infStr
intFile.close()
end of loop


Playing Loop
Game in Progress
Menu Options
Roll 1d3
rnd1to3 == 1
rnd1to3 == 2
rnd1to3 == 3
monSTR = 1
numTRS = 1
monSTR = 5
numTRS = 2
monSTR = 10
numTRS = 3
plyrIn == 1
Monster v Player
Display Message
plyrIn == 2
Playing = False
Monster
Combat


Monster Combat
plyrIn == 1
monSTR> combstr
monSTR< combstr
monSTR == combstr
badstuff
victory
close
Lvl --
Loot Generation
plyrIn == 2
roll 1d6
1d6 == 6
got away
1d6 != 6
bad stuff
Lvl --
Break
Break
lvl == 10
End Monster Combat
Victory && Rounds
Playing = False


Loot Generation
tCount = 1
tCount ++
tCount <= numTRS
roll 1d8, calc prob
luck > prob
extrTRS = true
extrTRS == true ? rnd1to6++ : rnd1to6;
End Loot Generation
1d8 == 1||2||3
1d8 == 1||2||3
1d8 == 1||2||3
Level_1_Tres
Level_2_Tres
Level_3_Tres
combstr++
combstr+2
combstr+3

Pseudo Code

```
Main{
        Initialize Variables
        Open Info.txt file
        if(file returns null){
                Return error message}
        Do{
                While not playing and running{
                Display welcome message and options
                menu(1): Play game
                        If character not made, return to menu
                        If character made confirm input and play
                menu(2): Set game to create character
                menu(3): Read out info file
                menu(4); Quit game, use confirmation
                if(to create character){
                        Input name as 3 Chars
                        Turn to String
                                Use confirm loop on name
                        Establish character sex
                                Use confirm loop on sex
                        Initialize class and race
                        }
                }
                While playing and running{
                Display Game Menu Message
                Display Options
                Option(1): Kick down the door
                        Generate Monster with respective treasure
                        Give Choice to run/fight
                                If fight, compare power
                                        If greater or equal, win lvlup, give treasure
                                        If lower, loose level
                                If run, roll 1d6
                                        If 6, run away
                                        If other, loose level
                Option(2): Quit Game
                        Returns to main menu

        }
        While(running)
        Return 0;
```

Major Variables

Type	Variable Name	Descrition	Location(Declaration)
Int	**Combstr**	Shows the strength of the player	**48**
	lvl	The level and win condition	**48**
	monSTR	Shows the strength of the monster	**55**
	numTRS	The number of treasures rewarded	**56**
	rnd1to3	Value of random int between 1 and 3	**57**
	rnd1to6	Value of random int between 1 and 6	**58**
	rnd1toH	Value of random int between 1 and 100	**59**
	tresRew		**60**
	rounds	Counts number of rounds to win	**61**
Char	**name1**	Names first character	**52**
	name2	Names second character	**52**
	name3	Names third character	**52**
	cPlyrIn	Player input with character	**65**
	sex	Player sex as character	**49**
	chaClass	Player class as character	**49**
	race	Player race as character	**49**
String	**nameCom**	Combination of all name char to string	**53**
Float	**prob**		**62**
	luck	Player luck, chance to get extra treasure	**50**
Bool	**extrTRS**	If player got extra treasure	**63**

	chrMade	If character was made	**68**
	playable	If game is playable	**69**
	quit	If player wants to quit	**70**
	running	If game is running	**71**
	playing	If user is playing	**69**
	rightIn	If input is right	**72**
ifstream	**infFile**	File containing munchkin info	**73**

C++ Concepts

Chapter	New syntax and keywords	First Location (Line/Line used)
2	**cout**	84
	libraries	26
	variables	48
	Identifiers	48
	Integers	48
	Characters	52
	Strings	53
	Floats	50
	Bools	63
	Sizeof	**********
	Variables <= 7 char	48-73(All variable declarations)
	scope	**********
	Arithmetic Operators	232
	Comments	36
	Named Constants	50
	Programing Style	**********
3	**cin**	88
	Math Expression	221
	Mixing data types	**********
	Overflow/underflow	**********
	Type casting	236
	Multiple assignment	152

	Formatting output	84
	Strings	127
	Math library	221
	Hand tracing	**********
4	**Relational Operators**	215
	If	222
	if-else	92&&103
	Nesting	258
	if-else-if	124&&161
	Flags	289
	Logical operators	82
	Validating user input	161-169
	Conditional Operator	238
	Switch	96
5	**Increment/Decrement**	214
	While	132
	Do-while	80-298
	For Loop	219
	Files input/output both	75
	No breaks in loops	*******

Reference

1. Text Book
2. C++ Library Documentations
3. Provided Class Lab examples
    a. Shoot-Out Lab

Example Inputs and Outputs

```
          Welcome to Munchkin, the dungeon crawling 'card' game!




           1. Play Game       2. Create Character                 3. Info        4. Quit
2

What is your name? Use only 3 letters: IAN
Your name is IAN?
Y/N :    N
What is your name? Use only 3 characters:ian
Your name is ian?
Y/N :    N
What is your name? Use only 3 characters:14N
Your name is 14N?
Y/N :    Y
Is your character Male or Female?
M/F :    M
```

```
          Welcome to Munchkin, the dungeon crawling 'card' game!




           1. Play Game       2. Create Character                 3. Info        4. Quit
1

Are you sure you want to start the game?
    1. Yes     2. No
2


          Welcome to Munchkin, the dungeon crawling 'card' game!
```

```
          Welcome to Munchkin, the dungeon crawling 'card' game!



           1. Play Game       2. Create Character                3. Info        4. Quit
3

**********************************************************************************************************************************************************

In Munchkin, you're a fantasy hero kicking down doors in a dungeon, looking for monsters to bash and Treasures to grab.
You level up each time you kill a monster.
Reach Level 10, and you win!
But watch out - other players mights stab you in the back at any moment!

On your turn, you kick down the door by drawing and revealing the top Door card.
Compare your Level and bonuses against the monster; if your total is bigger or equal, you win!
The fight isn't over until you decide to Run Away.

**********************************************************************************************************************************************************
```

```
          Welcome to Munchkin, the dungeon crawling 'card' game!




           1. Play Game       2. Create Character                  3. Info        4. Quit
1

Are you sure you want to start the game?
    1. Yes     2. No
1
GAME START

Game in Progress, Reach Level 10 or Quit
14N Level: 1



    1. Kick down the door             2. Quit Game
1
Monster is a strength level of : 1
You are a strength level of : 1
Would you like to fight or run away?
1. Fight 2. Run
1

That was close! Level UP!
Here is some treasure!
 You got some new gear! + 2 to your fighting level!


Game in Progress, Reach Level 10 or Quit
14N Level: 2



    1. Kick down the door             2. Quit Game
1
Monster is a strength level of : 10
You are a strength level of : 3
Would you like to fight or run away?
1. Fight 2. Run
2

You ran away!


Game in Progress, Reach Level 10 or Quit
14N Level: 2
```

Program

```
/*
 * File:   main.cpp
 * Author: Ian Langley
 *
 * This version of Munchkin will only have the main menu, allowing for the installation of the game play loop later on.
 * Munchkin is card game in which players race to level 10 by battling monsters in the dungeon. Each turn, a player can
 * "Kick down the door" revealing a monster, curse, or buff card. The players then compare a numerical value related to them
 * to the monster's level, and if they are higher, they win. In traditional munchkin there are ways to manipulate other players
 * such as adding bonuses to monster, or helping in the fight for money. This version of munchkin will demonstrate the core
 * game play loop with a single player, working essentially as a race against time.
 *
 * Created on February 3, 2020, 11:03 AM
 */
/*
 * Future Tasks:
 * 1. Create a 2D array to generate a large quantity of monsters;
 * 2. Create a 2D array for equipment;
 * 3. Port all loops to Void Functions
 * 4. Add multiple players;
 */

//Muchkin Game

//Included Libraries
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <cmath>
#include <cstdlib>
#include <fstream>
#include <string>
```

```
#include <ctime>
//Global Variables

//Created Functions **********THESE NEED TO BE UPDATED: DO NOT USE YET(FEB 5 2020)
void mainMenu(bool& playing, bool& playable, bool& quit, bool& chrSel);
void quitCon(bool& quit, bool& running);
void chrCreat(char& sex, char& chaClass, char& race, bool& playable, bool& chrSel, char& name1, char& name2, char& name3);
void kickDTD(int& plyrIn, int& combstr, bool& playing, char& name1, char& name2, char& name3, int& lvl, float& luck);
int genTRES(float luck);
using namespace std;

int main(int argc, char** argv)
{
   srand(static_cast<unsigned int> (time(0)));
   //Player Character Variables
   int combstr, lvl = 1;
   char sex, chaClass, race;
   const float luck = 1.0f / 10.0f;
   bool alive;
   char name1, name2, name3;
   string nameCom("");
   //Game System Variables
   int monSTR,
         numTRS,
         rnd1to3,
         rnd1to6,
         rnd1toH,
         tresRew,
         rounds = 1;
   float prob;
   bool extrTRS = false;
   //Menu Variables
   char cPlyrIn;
   string infStr;
   int plyrIn;
   bool chrMade = false,
```

```
        playable = false, playing = false,
        quit = false, chrSel = false,
        running = true,
        rightIn = false;
    ifstream infFile;

    infFile.open("infoFile.txt");
    if (!infFile) {
       cout << "Error opening file. \n";
    }
    //Game running
    do {
       //Main
Menu**************************************************************************
************************
       while (playing == false && running == true) {
          //Main Menu
Display***********************************************************************
*****************************
          cout << setw(65) << endl << endl << "Welcome to Munchkin, the dungeon crawling
'card' game!" << setw(25) << " " << endl;
          cout << endl << endl << endl << endl << endl;
          cout << setw(25) << "1. Play Game " << setw(25) << " 2. Create Character" << setw(25)
<< "3. Info"
                << setw(15) << "4. Quit" << endl;
          cin >> plyrIn;
          cout << endl;
          switch (plyrIn) {
             // Starts the gamplay loop
          case 1:
             if (playable == true) {
                cout << "Are you sure you want to start the game?" << endl;
                cout << setw(10) << "1. Yes" << setw(10) << "2. No" << endl;
                cin >> plyrIn;
                //Validates input, allows for cancelation.
                switch (plyrIn) {
                case 1:cout << "GAME START";
                   playing = true;
                   break;
```

```
            case 2: break;
            }
         }
         else {
            cout << "You have yet to make a character! Go to character creation to start!" <<
endl;
         }
         break;
         // Sets loop to include character creation
      case 2: chrSel = true;
         break;
         //Reads out information on Munchkin
      case 3:
         if (infFile.is_open() == false) {
            infFile.open("infoFile.txt");
         }
         if (infFile.is_open()) {
            while (getline(infFile, infStr)) {
               cout << infStr << endl;
            }
            infFile.close();
         }
         break;
         //Sets game to quit state, results in quit confirmation
      case 4: quit = true;
         break;
      }
      //Char Creation
**************************************************************************
**************************
      if (chrSel == true) {
         cout << "What is your name? Use only 3 letters: ";
         cin >> name1 >> name2 >> name3;
         nameCom += name1;
         nameCom += name2;
         nameCom += name3;
         cout << "Your name is " << nameCom << "?" << endl << "Y/N :    ";
         cin >>cPlyrIn;
         while (cPlyrIn == 'N' || cPlyrIn == 'n') {
```

```
            cout << "What is your name? Use only 3 characters:";
            cin >> name1 >> name2 >> name3;
            nameCom = "";
            nameCom += name1;
            nameCom += name2;
            nameCom += name3;
            cout << "Your name is " << nameCom << "?" << endl << "Y/N :    ";
            cin >> cPlyrIn;
         }
         do {
            cout << "Is your character Male or Female?" << endl;
            cout << "M/F :    ";
            cin >> cPlyrIn;

            if (cPlyrIn == 'M' || cPlyrIn == 'm' || cPlyrIn == 'F' || cPlyrIn == 'f') {
               rightIn = true;
            }
         }
         while (rightIn == false);
         combstr = lvl = 1;
         sex = cPlyrIn;
         chaClass = 'D';
         race = 'H';
         playable = true;
         chrSel = false;
         //Quit
Game**********************************************************************
**********************************
            quitCon(quit, running);
      }
         //Quit Confirmation
      else if (quit == true) {
         cout << "Are you sure you want to quit?";
         cout << setw(10) << "1. Yes" << setw(10) << "2. No" << endl;
         cin >> plyrIn;
         if (plyrIn == 1) {
            running = false;
         }
         else {
```

```
            quit = false;
         }
      }
   }
   //Game
Playing**************************************************************************
*************************
   while (playing == true) {
      cout << endl << endl << setw(25) << "Game in Progress, Reach Level 10 or Quit" <<
endl;
      cout << nameCom << " Level: " << lvl << endl;
      cout << endl;
      cout << endl;
      cout << endl;
      cout << setw(25) << "1. Kick down the door" << setw(25) << "2. Quit Game" << endl;
      cin >> plyrIn;
      switch (plyrIn) {
      case 1: //Generate Monster
         rnd1to3 = rand() % 3 + 1;
         if (rnd1to3 == 1) {
            monSTR = 1;
            numTRS = 1;
         }
         else if (rnd1to3 == 2) {
            monSTR = 5;
            numTRS = 2;
         }
         else if (rnd1to3 == 3) {
            monSTR = 10;
            numTRS = 3;
         }
         //Display Monster Level
         cout << "Monster is a strength level of : " << monSTR << endl;
         //Display player combat power
         cout << "You are a strength level of : " << combstr << endl;
         //Give option to run or fight;
         cout << "Would you like to fight or run away?" << endl << "1. Fight 2. Run" << endl;
         cin >> plyrIn;
         cout << endl;
```

```
            switch (plyrIn) {
               //Fight monster
            case 1:
               //Monster Stronger
               if (monSTR > combstr) {
                  //Nullifiy low level punishment
                  if (lvl = 1) {
                     cout << "You are too low level for bad stuff to happen!";
                  }
                  //Allows or punishment over level 1
                  if (lvl > 1) {
                     cout << "BAD STUFF HAPPENS" << endl;
                     cout << "You lost a level!";
                     lvl--;
                  }
               }
               //Player Stronger
               if (monSTR < combstr) {
                  cout << "You beat them! Level UP!" << endl << "Here is some treasure!" <<
endl;
                  //Generates treasure based on rolled monster
                  for (int tCount = 1; tCount <= numTRS; tCount++) {
                     //Random luck +1 treasure. Adds a touch of variety
                     rnd1to6 = rand() % 6 + 1;
                     prob = 1.0f * static_cast<int>(rand()) / (pow(2, 31) - 1);
                     if (luck > prob) {
                        extrTRS = true;
                     }
                     extrTRS == true ? rnd1to6++ : rnd1to6;
                     //Three gear tiers. Allows for additional rewards later.
                     if (rnd1to6 == 1 || rnd1to6 == 2 || rnd1to6 == 3) {
                        cout << " You got some new gear! + 1 to your fighting level!" << endl;
                        combstr += 1;
                     }
                     if (rnd1to6 == 4 || rnd1to6 == 5) {
                        cout << " You got some new gear! + 2 to your fighting level!" << endl;
                        combstr += 2;
                     }
                     if (rnd1to6 > 5) {
```

```
                cout << " You got some new gear! + 3 to your fighting  level!" << endl;
                combstr += 3;
              }
            }
            lvl++;
          }
          //Equal Strength
          if (monSTR == combstr) {
            cout << "That was close! Level UP!" << endl << "Here is some treasure!" <<
endl;
            for (int tCount = 1; tCount <= numTRS; tCount++) {
              //Random luck +1 treasure. Adds a touch of variety
              rnd1to6 = rand() % 6 + 1;
              prob = 1.0f * rand() / (pow(2, 31) - 1);
              if (luck > prob) {
                extrTRS = true;
              }
              extrTRS == true ? rnd1to6++ : rnd1to6;
              //Random luck +1 treasure. Adds a touch of variety
              if (rnd1to6 == 1 || rnd1to6 == 2 || rnd1to6 == 3) {
                cout << " You got some new gear! + 1 to your fighting level!" << endl;
                combstr += 1;
              }
              if (rnd1to6 == 4 || rnd1to6 == 5) {
                cout << " You got some new gear! + 2 to your fighting level!" << endl;
                combstr += 2;
              }
              if (rnd1to6 > 5) {
                cout << " You got some new gear! + 3 to your fighting  level!" << endl;
                combstr += 3;
              }
            }
            lvl++;
          }
          break;
          //Running away
        case 2:
          //Rolling a 1d6
          rnd1to6 = rand() % 6 + 1;
```

```
            //Successful Run
            if (rnd1to6 == 6) {
               cout << "You ran away!" << endl;
            }
            //Unsuccessful run
            if (rnd1to6 != 6) {
               //Low level punishment nullifiation;
               if (lvl = 1) {
                  cout << "You are too low level for bad stuff to happen!";
               }
               //Punishment over level 1
               if (lvl > 1) {
                  cout << "You were caught! BAD STUFF HAPPENS" << endl;
                  cout << "You lost a level!";
                  lvl--;
               }
            }
            break;
         }
         break;
         //Playing quit button
      case 2: playing = false;
         break;
      }
      //Increasing round counter
      rounds++;
      //Win condition hecker
      if (lvl == 10) {
         cout << setfill('*') << setw(50) << endl << endl;
         cout << "You won the game!";
         cout << "It took " << rounds << " rounds!";
         playing = false;
         cout << setfill('*') << setw(50) << endl << endl;
         setfill(' ');
      }
   }
}
while (running == true);
return 0;
```

```
}
//9 Lines of comments

void mainMenu(bool& playing, bool& playable, bool& quit, bool& chrSel)
{
   bool chrMade;
   int plyrIn;
   cout << setw(75) << endl << endl << "Welcome to Munchkin, the dungeon crawling 'card'
game!" << endl;
   cout << endl << endl << endl << endl << endl;
   cout << setw(25) << "1. Play Game " << setw(25) << " 2. Create Character" << setw(25) <<
"3. Quit" << endl;
   cin >> plyrIn;
   switch (plyrIn) {
   case 1:
      if (playable == true) {
         cout << "Are you sure you want to start the game?" << endl;
         cout << setw(10) << "1. Yes" << setw(10) << "2. No" << endl;
         cin >> plyrIn;
         switch (plyrIn) {
         case 1:cout << "GAME START";
            playing = true;
            break;
         case 2: break;
         }
      }
      else {
         cout << "You have yet to make a character! Go to character creation to start!" << endl;
      }
      break;
   case 2: chrSel = true;
      break;
   case 3: quit = true;
      break;
   }
}

void chrCreat(char& sex, char& chaClass, char& race, bool& playable, bool& chrSel, char&
name1, char& name2, char& name3)
```

```
{
   char cPlyrIn;
   bool rightIn = false;
   cout << "What is your name? Use only 3 letters: ";
   cin >> name1 >> name2 >> name3;
   cout << "'Your name is " << name1 << name2 << name3 << "? y/n";
   cin >>cPlyrIn;
   while (cPlyrIn == 'N' || cPlyrIn == 'n') {
      cout << "What is your name? Use only 3 characters:";
      cin >> name1 >> name2 >> name3;
      cout << "Your name is " << name1 << name2 << name3 << "? y/n";
      cin >> cPlyrIn;
   }
   do {
      cout << "Is your character Male or Female?" << endl;
      cout << "M/F :    ";
      cin >> cPlyrIn;

      if (cPlyrIn == 'M' || cPlyrIn == 'm' || cPlyrIn == 'F' || cPlyrIn == 'f') {
         rightIn = true;
      }
   }
   while (rightIn == false);
   sex = cPlyrIn;
   chaClass = 'D';
   race = 'H';
   playable = true;
   chrSel = false;
}

void quitCon(bool& quit, bool& running)
{
   int plyrIn;
   if (quit == true) {
      cout << "Are you sure you want to quit?";
      cout << setw(10) << "1. Yes" << setw(10) << "2. No" << endl;
      cin >> plyrIn;
      if (plyrIn == 1) {
         running = false;
```

```
        }
        else {
            quit = false;
        }
    }
}

void kickDTD(int& plyrIn, int& combstr, bool& playing, char& name1, char& name2, char&
name3, int& lvl, float& luck)
{
    srand(static_cast<unsigned int> (time(0)));
    int monSTR,
            numTRS;
    int rnd1to3;
    int rnd1to6;
    cout << setw(25) << "Game in Progress, Reach Level 10 or Quit" << endl;
    cout << name1 << name2 << name3 << " Level: " << lvl << endl;
    cout << endl;
    cout << endl;
    cout << endl;
    cout << setw(25) << "1. Kick down the door" << setw(25) << "2. Quit Game" << endl;
    cin >> plyrIn;
    switch (plyrIn) {
    case 1: //Generate Monster
        rnd1to3 = rand() % 3 + 1;
        if (rnd1to3 == 1) {
            monSTR = 1;
            numTRS = 1;
        }
        else if (rnd1to3 == 2) {
            monSTR = 5;
            numTRS = 2;
        }
        else if (rnd1to3 == 3) {
            monSTR = 10;
            numTRS = 3;
        }
        //Display Monster Level
        cout << "Monster is a strength level of : " << monSTR << endl;
```

```
    //Display player combat power
    cout << "You are a strength level of : " << combstr << endl;
    //Give option to run or fight;
    cout << "Would you like to fight or run away?" << endl << "1. Fight 2. Run" << endl;
    cin >> plyrIn;
    switch (plyrIn) {
    case 1:
       if (monSTR > combstr) {
          cout << "BAD STUFF HAPPENS" << endl;
       }
       if (monSTR == combstr) {
          cout << "That was close!" << endl << "Here is some treasure!" << endl;
          for (int tCount = 1; tCount == numTRS; tCount++) {
             genTRES(luck);
          }
       }
       if (monSTR < combstr) {
          cout << "You beat them!" << endl << "Here is some treasure!" << endl;
          for (int tCount = 1; tCount = numTRS; tCount++) {
             genTRES(luck);
          }
       }
       break;
    case 2:
       rnd1to6 = rand() % 6 + 1;
       if (rnd1to6 == 6) {
          cout << "You ran away!" << endl;
       }
       if (rnd1to6 != 6) {
          cout << "You were caught! BAD STUFF HAPPENS";
       }
       break;
    }
    break;
case 2: playing = false;
    break;
}
```

```
}

int genTRES(float luck)
{
   bool extrTRS = false;
   float prob;
   int tresRew;
   int rnd1to6;
   int rnd1toH;
   srand(static_cast<unsigned int> (time(0)));
   //generate number, higher being better
   rnd1to6 = rand() % 6 + 1;
   prob = 1.0f * rand() / (pow(2, 31) - 1);
   if (luck > prob) {
      extrTRS = true;
   }
   extrTRS == true ? rnd1to6++ : rnd1to6;
   if (rnd1to6 == 1 || rnd1to6 == 2 || rnd1to6 == 3) {
      cout << " You got some new gear! + 1 to your fighting level!" << endl;
   }
   if (rnd1to6 == 4 || rnd1to6 == 5) {
      cout << " You got some new gear! + 2 to your fighting level!" << endl;
   }
   if (rnd1to6 > 5) {
      cout << " You got some new gear! + 3 to your fighting  level!" << endl;
   }

}
```