Max Guidry

10-5-16

Text Based Adventure Game

**1.0 Requirements Documentation**

**1.1 Description of Problem**

**Name:** Text Based Adventure Game

**Problem Statement:** Create a fictional interactive world where a player can traverse using text commands in the console. This has to be done using Donray’s String Class

**Problem Specification:** To create a Text Based Adventure Game to be played in the console where the player must input commands to move and interact with the world around. These responses must be manipulated using Donray’s String Class. The player needs to be able to move around the world using commands like “move north”.

**1.2 Input Information**

The player enters in commands that are executed if acceptable commands.

**1.3 Output Information**

The console prints information onto the screen about the room you are in.

**1.4 User Interface**

Not Applicable.

**2.0 System Architecture**

**Member Functions in the class:**

**Room Class:**

**PROTOTYPE:** void openDoor(Room)

**DESCRIPTION:** opens doors that were locked because of an enemy in the room

**PARAMETERS:** takes in a variable of type Room that the player is in

**PRECONDITION:** player must be in a room

**POST CONDITION:** doors that were locked are now open

**VISIBILITY:** public

**PROTOTYPE:** void lockDoors()

**DESCRIPTION:** locks doors of the room that the player is in

**PARAMETERS:** Not applicable

**PRECONDITION:** must be in a room with an enemy in it

**POST CONDITION:** all doors are locked in the room

**VISIBILITY:** Public

**PROTOTYPE:** void PrintInfo()

**DESCRIPTION:** prints the info about the room to the screen for the player to read

**PARAMETERS:** Not applicable

**PRECONDITION:** Not applicable

**POST CONDITION:** Not applicable

**VISIBILITY:** Public

**PROTOTYPE:** void printDatBoi()

**DESCRIPTION:** prints Dat Boi to the screen

**PARAMETERS:** Not applicable

**PRECONDITION:** Not applicable

**POST CONDITION:** Not applicable

**VISIBILITY:** Public

**PROTOTYPE:** void printPepe()

**DESCRIPTION:** prints Pepe to the screen

**PARAMETERS:** Not applicable

**PRECONDITION:** Not applicable

**POST CONDITION:** Not applicable

**VISIBILITY:** Public

**PROTOTYPE:** void printDoge()

**DESCRIPTION:** prints Doge to the screen

**PARAMETERS:** Not applicable

**PRECONDITION:** Not applicable

**POST CONDITION:** Not applicable

**VISIBILITY:** Public

**PROTOTYPE:** void printHarambe()

**DESCRIPTION:** prints Harambe to the screen

**PARAMETERS:** Not applicable

**PRECONDITION:** Not applicable

**POST CONDITION:** Not applicable

**VISIBILITY:** Public

**PROTOTYPE:** void printBoss()

**DESCRIPTION:** prints Boss to the screen

**PARAMETERS:** Not applicable

**PRECONDITION:** Not applicable

**POST CONDITION:** Not applicable

**VISIBILITY:** Public

**PLAYER CPP FILE:**

**PROTOTYPE:** int CheckResponse(MyString,Room[][5])

**DESCRIPTION:**  takes in the player’s command and checks what is said in it then decides what to do.

**PARAMETERS:** takes in a MyString which is the response of the player and takes in an array of Room which is the map that you are playing on.

**PRECONDITION:** player must type in something

**POST CONDITION:** something should have happened depends on what the response is

**VISIBILITY:** public

**PROTOTYPE:** char \* Response()

**DESCRIPTION:**  gets the players commands

**PARAMETERS:** Not applicable

**PRECONDITION:** Not applicable

**POST CONDITION:** returns the players response

**VISIBILITY:** Public

**PROTOTYPE:** int moveNorth()

**DESCRIPTION:**  decreases the player m\_positionY variable by 1

**PARAMETERS:** Not applicable

**PRECONDITION:** player must command to move north

**POST CONDITION:** Player should have moved north and function returns the position in the Y direction

**VISIBILITY:** Public

**PROTOTYPE:** int moveSouth()

**DESCRIPTION:**  increases the player m\_positionY variable by 1

**PARAMETERS:** Not applicable

**PRECONDITION:** player must command to move south

**POST CONDITION:** Player should have moved south and function returns the position in the Y direction

**VISIBILITY:** Public

**PROTOTYPE:** int moveEast()

**DESCRIPTION:**  increases the player m\_positionX variable by 1

**PARAMETERS:** Not applicable

**PRECONDITION:** player must command to move east

**POST CONDITION:** Player should have moved east and function returns the position in the X direction

**VISIBILITY:** Public

**PROTOTYPE:** int moveWest()

**DESCRIPTION:**  decreases the player m\_positionX variable by 1

**PARAMETERS:** Not applicable

**PRECONDITION:** player must command to move west

**POST CONDITION:** Player should have moved west and function returns the position in the X direction

**VISIBILITY:** Public

**PROTOTYPE:** void setPrevious(int,int)

**DESCRIPTION:**  keeps track of the player’s previous position so the player can command “go back” if they want to return to the previous room

**PARAMETERS:**  Takes in the X and Y positions of the player

**PRECONDITION:** Player must exist and be in a room

**POST CONDITION:** function sets the new previous positions after the player moves

**VISIBILITY:** public

**Member Variables:**

**Room Class:**

**bool** m\_EastDoor-variable for if the east door exists

**bool** m\_NorthDoor-variable for if the north door exists

**bool** m\_SouthDoor-variable for if the south door exists

**bool** m\_WestDoor-variable for if the west door exists

**bool** m\_Empty-variable for if the room is empty or not

**bool** m\_HasBossKey-variable for if the room has a boss key or not

**bool** m\_HasTreasureKey-variable for if the room has a treasure key or not

**bool** m\_HasSword-variable for if the room has a sword in it or not

**bool** m\_HasEnemy-variable for if the room has an enemy or not

**bool** m\_HasDeathButton-variable for if the room has a death button in it or not

**bool** m\_HasTrapHole-variable for if the room has a trap hole or not

**bool** m\_HiddenDoor-variable for if the room has a hidden door or not

**bool** m\_HasBoss-variable for if the room has a boss in it or not

**const** **char** **\*** m\_Text**-** stores the room information in it

**Player Class:**

**int** m\_positionX-variable to hold the value of the players X postion

**int** m\_positionY-variable to hold the value of the players Y postion

**bool** m\_hasSword-variable for if the player has the sword or not

**bool** m\_hasTreasure-varable for if the player has the treasure key

**bool** m\_hasBossKey-for if the player has the key to the boss room

**ROOMS CPP FILE:**

#include"Rooms.h"

#include<iostream>

#include<Windows.h>

Room::Room(bool EastDoor, bool WestDoor, bool NorthDoor, bool SouthDoor, bool Empty, const char \* Text, bool Key, bool BossKey, bool Sword, bool Enemy, bool RedButton, bool TrapHole)

{

m\_EastDoor = EastDoor;

m\_WestDoor = WestDoor;

m\_NorthDoor = NorthDoor;

m\_SouthDoor = SouthDoor;

m\_Empty = Empty;

m\_Text = Text;

m\_HasTreasureKey = Key;

m\_HasBossKey = BossKey;

m\_HasSword = Sword;

m\_HasEnemy = Enemy;

m\_HasDeathButton = RedButton;

m\_HasTrapHole = TrapHole;

}

Room::Room(bool EastDoor, bool WestDoor, bool NorthDoor, bool SouthDoor, bool Empty, const char \* Text, bool Key, bool BossKey, bool Sword, bool Enemy, bool RedButton, bool TrapHole, bool HiddenDoor, bool Boss)

{

m\_EastDoor = EastDoor;

m\_WestDoor = WestDoor;

m\_NorthDoor = NorthDoor;

m\_SouthDoor = SouthDoor;

m\_Empty = Empty;

m\_Text = Text;

m\_HasTreasureKey = Key;

m\_HasBossKey = BossKey;

m\_HasSword = Sword;

m\_HasEnemy = Enemy;

m\_HasDeathButton = RedButton;

m\_HasTrapHole = TrapHole;

m\_HiddenDoor = HiddenDoor;

m\_HasBoss = Boss;

}

void Room::PrintInfo()

{

for (int i = 0; m\_Text[i] != '\0'; i++)

{

std::cout << m\_Text[i];

Sleep(18);

}

std::cout << std::endl;

}

void Room::printDatBoi()

{

std::cout << "\n\nMMMMMMMMMMMMMMMMMMMN` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` ``..-://+o/` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` .sssssosshNho. `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` :mmddhyysyyys- `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` -dNNNNmdhysss. `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `+ddddhhysos+` ";

std::cout << "`NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` .yhhyyysooos/` ";

std::cout << "`NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `.shhhyyyyysosy/ `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` ``..-/osyyhsyhyyyyyyysos: `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `.:/+osyyyyso:`.yyysssyyyysoo+-` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` .-//+ss++++/-` :hysssyyyyssyh- `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `-/o/` ";

std::cout << ":hyyyyyyyyyyyh` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` .hhhhhhhhhhyoo- `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `+hdddhhhhdddyos+` ";

std::cout << "`NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `+yyhdmddddmmmdsoso:. `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `oyhhhdmysdmNNNy/yso++/` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `ohddmdyoyddmmmNNs.oys+++. `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `/hddy/.ooosy/////-``-++++/` ";

std::cout << "`NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `+ydy-` -: `````.ooo. `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` -syh-`` :- ```.++oo. `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` ./+oo.`` .+:` ````:ooso` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `///+``-/symdoo/. ````.+oss/ `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` .//++hmNmmNysmNNh/.``.+osy. ";

std::cout << "`NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` -odmmy:.oh-s--yNmh-..oss+ `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `+mds.::`/y-o`./-ymd--oss` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` ";

std::cout << "+mhhs` :-.s-/`/`.:hNh:sy- `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` .+mdyoo--`:/s/::`..`-NNss/ `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` .yNy++o:-:-ohso-.`-::dNys` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` dms..-:++shdyo/-:-.`ymy/ `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `dms..-:::+odms//+o/-dms: `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` ";

std::cout << "dmh-.``-::osso-::--+dmso/:-` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` +Nd/`.-`-/:/o-/-.--hNh``` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` ";

std::cout << "`dNd/` .:`/.::./-`yNd. `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` ``..``....://////////:+mNd+::`./`-/`-odmh. `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` ``.:/+:/+sdNmmdsss/+sydNms` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` `-/+/:--/sdmNNNNNNNNdy+:-..` `NMMMMMMMMMMMMMMMMMMM\nMMMMMMMMMMMMMMMMMMMN` .::-....-..:+++/:-.` `NMMMMMMMMMMMMMMMMMMM\n\n\n O SHIT WADDUP\n\nWhat will you do?\n";

}

void Room::printPepe()

{

std::cout << "..................................-:::::-...........................................................\n";

std::cout << "............................-+osyysssssssyyso+/...............-:/+++++++/:..........................\n";

std::cout << ".........................:ossooooooooooooooooosyso/......-/+ssssooooooooosys+-......................\n";

std::cout << "......................-+ssoooooooooooooooooooooooosyo/+ossoooooooooooooooooooyo-....................\n";

std::cout << ".....................oyooooooooooooooooooooooooooooooyyooooooooooooooooooooooooy/...................\n";

std::cout << ".................../ysoooooooooooooooossssssooooooooooshooooooooooooooooooooooooy+..................\n";

std::cout << ".................-ssoooooooooooosyyyyyssssssyyyyyysooooshooooooooooooooooooooooooy:.................\n";

std::cout << "................:ysooooooooosyyysooooooooooooooooosyyysoysoossssyssssssssssyssssooy.................\n";

std::cout << "...............:yooooooooooyysoooooooooooooooooooooooosyhhyyssssosoooooooooossssyyyso/:-............\n";

std::cout << "..............:yooooooooooooooooooooooooooooooooooooooooosysoooooooooooooooooooooooooosss+-.........\n";

std::cout << ".............:yoooooooooooooooooooooooooosssyyyyyyyyyyyyyyyhhysoooooooosssssssyyyyysssssssso:.......\n";

std::cout << ".............yooooooooooooooooooooooosyyysssssyyyyyyyyyyssssssyyssssyyyssssyyyyyyyyyyyyyssyyy+:.....\n";

std::cout << ".........:+oyyooooooooooooooooooossyysssyyyyssso++++oooossssyyyyhyyyyyyyysssssssssyssssyyyyyyyss....\n";

std::cout << "......-+sssohooooooooooooooossyyyssssyyso+++sshmmmhs:```.-:/+osysssssooso++oyhsdNNNmh/.-:/+yssyh+-..\n";

std::cout << "...../ysoooshooooooooooooosssysssyyyso:.`.hNMosMNddMNy` ``.:ssoo+:.` :mMN+dMmydMNs .:/+oyh-.\n";

std::cout << "....+yoooooyyoooooooooooohsssssssso:. oMMMmdMd::mMMo .y / .dMMMdmMy:oMMM: `-y-.\n";

std::cout << "..-oyooooooyooooooooooooosyyyyyyy-` yMMMmdMMNNMMMy /- `NMMMdNMMNMMMM/ +/.\n";

std::cout << "..oyooooooooooooooooooooooooooosso/-` oMMMMMMMMMMMN: `- + hMMMMMMMMMMMh` `.:++-.\n";

std::cout << ".+yooooooooooooooooooooooooooooossooo+:-.`omNMMMMMNms- `..:+ssh/---.```.odNMMMMMNdo:--:/+ossh:-..\n";

std::cout << "+yoooooooooooooooooooooooooooooossysssssso++osyysso/::/+ooooososyyssssoooooooshhhhhyssssssssso+:....\n";

std::cout << "yoooooooooooooooooooooooooooooooooosssyyysssyyyyyyyyssssssssyyyssoooooooooooooooooooooooooss-.......\n";

std::cout << "soooooooooooooooooooooooooooooooooooooooossssssssssssssssyysooooooooooooooooooooooooooosss/.........\n";

std::cout << "sooooooooooooooooooooooooooooooooooooooooooooooooooossyysssoooooooosyysssooossssssssys/:-...........\n";

std::cout << "sooooooooooooooooooooooooooooooooooooooooooooosssyyysssoooooooooooooosssyhhyyyysssssss+-............\n";

std::cout << "soooooooooooooooooooooooooooooooooooooooooooosysssoooooooooooooooooooooooossooooooooooss+-..........\n";

std::cout << "soooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooss:.........\n";

std::cout << "soooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooy-........\n";

std::cout << "sooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooosy........\n";

std::cout << "sooooooooooooooooooooooooooooooooossoooooooooooooooooooooooooooooooooooooooooooooooooooooooy-.......\n";

std::cout << "sooooooooooooooooooooooooooossyyyyyyyyyyyysssssssooooooooooooooooooooooooooooooooooooooosssys+......\n";

std::cout << "sooooooooooooooooooooooooosyysooooooooooosssssssyyyyysssssssooooooooooooooooooooosssssyyysssoy/.....\n";

std::cout << "sooooooooooooooooooooooooshsoooossssssssssssoooooooossssyssyyyyyyyyyyyssssyyyyyyyyssssooooossy:.....\n";

std::cout << "soooooooooooooooooooooooohsooooossssssssyyyyyyyssssssssooooooooooossssssssssoooooooooossssss/.......\n";

std::cout << "sooooooooooooooooooooooooyysoooooooooooooooooossssyyyyyyyyyyyyysssssssssssssssyyyyyyyyyyyh:.........\n";

std::cout << "sooooooooooooooooooooysooosyyyyyyyyyyyyyysssssssooooooooooosssssssssssssssssssssssoooooooy+.........\n";

std::cout << "yoooooooooooooooooooooyyooooooooooooooooosssyyyyyyyysssssssssssssooooooooooooooooooooossys..........\n";

std::cout << ":ssoooooooooooooooooooosyyyssoooooooooooooooooooooossssyyyyyyyyyyyyyyyyyyyyyyyyyyysssso+:...........\n";

std::cout << "-:hyssooooooooooooooooooooossoooooooooooooooooooooooooooooooooooooooooooooooossss/..................\n";

std::cout << "hhhhyyyyssooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooss+-....................\n";

std::cout << "hyyyhhyysyyyyyssoooooooooooooooooooooooooooooooooooooooooooooooooooooooosso+-.......................\n";

std::cout << "hyyyyyyyhhyyssssyyyyyyyysssooooooooooooooooooooooooooooooooooooooossss+/-...........................\n";

std::cout << "hyyyyyyyyyyyhhhyyssoooossssyyyyyyyyyyyyyyyysssssssssssssyyyyyyyhhh+.................................\n";

std::cout << "hyyyyyyyyyyyyyyyyhhhhyyyyysssssoooooooooooosssssssssssssssossyyhyyhs/...............................\n";

std::cout << "hyyyyyyyyyyyyyyyyyyyyyyyyyyhhhhhhhhhhhhyyyyyyyyyyyyyyyyhhhhhhyyyyyyyhhs+-...........................\n";

std::cout << "hyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyhs:.........................\n";

std::cout << "hyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyhs........................\n";

}

void Room::printDoge()

{

std::cout << "::::::::::::::::::::////////:::::/++//::::-------........................--.......-.......--........\n";

std::cout << ":::::::::::::::::/::///:::::---::/+oo+/:::--..................................-::://:-..............\n";

std::cout << "::::::::://::::///:::::::------::/+ooo+/::--................................-:/+++ooo+-.............\n";

std::cout << "::::::://////:///::::----------::/+ooooo+/:-..............................--:/++++osyso-........-...\n";

std::cout << "//://////::::::::-------------:://++oooo+++/----........................--:/++++++osssso--......---.\n";

std::cout << "::::::::----------------------://+++++++++++///::::------..-........-.--//+oo+++ooooosss+-..........\n";

std::cout << "----------------------------:///+o+////////////////////////////:::::::/+++osoooosssssssss:..........\n";

std::cout << "-------------:----:--:::::::////::::///////////////////////////+/++o++++osysooosssyyssoos+-.........\n";

std::cout << "--------:::::::://///+++++//::::::::///////+++++++++/////////////++oooossysooosyhhhhhysoso:.........\n";

std::cout << "::::://////++++oo++++++//::::::::::://///+++++++/+++++//////////++/++oyyysoooooshdddhysoss/-........\n";

std::cout << "++++++oooooooooo+///:::::::::::::::///++++++++////++++////++/////+///++osssooooshdmdhyoooo/-........\n";

std::cout << "ooooooooooo++++/::::::::::::::::::///++++++++/++////+/////++/++/+///////+osoosydmmmdhsoooo/--.-.....\n";

std::cout << "oooooooo++++//::::::-::::::-----:://++o+++++++++////////////+++/+/////////+ooydmmddhso++oo:.........\n";

std::cout << "oooo+++++++/::::::-:::::::::::::://+oo+++////////////////////+//+///////////+sshdhysoooooo:--.......\n";

std::cout << "ooo+++++++/::::::--::://///::::://+++++++//:::://///++////////////////////////++ssooooshso/--.......\n";

std::cout << "o+++++++//:--------:/osso++/::://++++o++/:::::::////++++/////////////////////////+ooooooo++:--......\n";

std::cout << "++++////::---....--/yhymmds+://///+++++/:::::/:::////+++++++//////////////////:////+ooossso/:--.....\n";

std::cout << "/::::::::--......--++hdmmdh+////://///////:///+oo++/++++oo++++//////////////////:://+oossso+/:--....\n";

std::cout << ":---------......--:oohhdmh+/:::::/:::////+++shhhmNmhs+++++++++////////////////////////+oyyso/:--....\n";

std::cout << "---------......--::+oyys+::::::::::://++oshmh++yhmNNNdso+++///////////////////////////++osso+/:--...\n";

std::cout << "--------.......-::::::::::::::-::::::/+oshmms+hmNNNmdds++///////////////////////////////+oso+/:--...\n";

std::cout << "--------.......--::---:::::::::::::://+++ohddyyddddyo//:/::::::::::::::::::::///////////++o++//-....\n";

std::cout << "-:::---.........----::::::::::::::::://++///++ooo+/////:::-------::::::::::::::::::::////++++//:-...\n";

std::cout << ":::::--.......-:/ooooooo++/::::-::::::////////////////::::------::/:::::::::::::::::://///++++/:--..\n";

std::cout << "::/::---.....:yddmmmdddddhyo/:----::::://::::/://////:::::------------::::::::::::::::::////////--..\n";

std::cout << ":://:-----.../dNNmNNNMMNNmddo:------:::::::::---:::::::::::----------::::::::::::::::::::///////:-..\n";

std::cout << "//::-..-..--:+ydmmmNNMMMNNmho/:-:::-::::::::--------::::::::::::::::::::::----::::::::::::///////--.\n";

std::cout << "/:::-.------/+hmmmNmmNNNNmhy+/:::::/::/::::::::::-----:::::::::::::::::::--:::::::::::::::///////:-.\n";

std::cout << "//::------::/oyhdmNmmmmddhso+///:::::://::::::::::::---:::::::::::::::::::::::::::::::////+++////:--\n";

std::cout << "//::-..---:/+syhhdmdddddyys+++/+/////////::::::::::::::::::::::::::::::::::::::::::///////+++/////:-\n";

std::cout << "//:-------::+syyhhdhhhhhhyso++//////////////:/:::::::///:::::::::::::::::::://///////////+++++////::\n";

std::cout << "//::--.---:::/ydmNNNNNddhhyso++++/+///+++++////////////:/://////:::::::::///////////////++++++/////:\n";

std::cout << "//:----------:oyhddmmmmmNNmdyo++++oosyyhys++/////////:::::://:::::::::://////:::::::///++++++++++//:\n";

std::cout << "-----.------::/+oyhdddddddddmmmhdddddhyys+/////////:::::/:::::::::://///:::::::::::////++++++++++//:\n";

std::cout << ".....--------:::/+syssssssssyhhysssssoo+////://////:::://:::::::::////::::::::::::::////++++/++++//:\n";

std::cout << ".....--------:::::/+ooooo++++++++o+++//////////////////:/::::::::::::::::::::::::::://+++++++++++///\n";

std::cout << "......------:::::::://///////////////////////////////////::::::::::::/::::::::::::///++++++++++++///\n";

std::cout << "......------:::::::::::://////////////////////////::///::::::::::::::::::::::://////+++++++++oo+++//\n";

std::cout << "......------:::::::::::::::://////////////////////::::://////::/::::::::::/:///////++++++++ooooo+++/\n";

std::cout << "-.....--::-:::::::::::::::://///////////////////:::::::::////:::::::::://////////+++++++++oooooo++++\n";

std::cout << "--.-----::::::::::::::://////////////////////////:/:::::::::::::::::::://///////+++++++oooooooo++++/\n";

std::cout << "--------:::::::::::::::::://////////////////////////::::::::::::::////////////++++++++ooooooo++++++/\n";

std::cout << "--------:::::::::::::::////////////////////////////:::::::::::::://////////////+++++ooooooo++++++///\n";

std::cout << "--------:::::::://:::::::///////////////////////////://:::://///////////////////++++oooooo++///////:\n";

std::cout << "---------::::::::///////://///////////+///////////////////////////////////////+++++ooo++++/////////:\n";

std::cout << "------------:::::://////////////////////++//+++++++++++++///////////////////////+++++++//////:::::::\n";

std::cout << "------------::::::::///+++++++++++++++++++++++++++++++////////////////////////+++++////:::::::::::::\n";

std::cout << "-------------::::::://////+++++++++++++++++++++++///////://///::////////////////::::::::::::::::::/:\n";

std::cout << "------..------:::::://////////+++++++++//++//////:::::://///::////::::///://::::::::::::::::::::::/:\n";

}

void Room::printHarambe()

{

std::cout << "NKNKKKKKKKNNNKKKKk!|!!!!)k ')]K@NNKKKkkkkk!!)kkk))kNKNNKKNKkkkk)KKNNKkk)KKNNKk\n";

std::cout << "kKkkNNNKKKKKNKKKNk!!| yk)kkL |>]KNNNKMMM)kkkk>||||>))KKKKKKKNKNNMkkNKKKk>)kNNKKK\n";

std::cout << "kkkkkNNKKNKMkkKKKWk!!)kkNM .|>kKMNNKWkw|kkk!!!!!!!!)Y)KKNNKKKKkkkkkMkk!)kKkkMkN\n";

std::cout << "kkkkkNNKKKKKKKKKNKkkkkkkk ,{ \)MMM^|M(NKNNNw!!!!|!!!!|))k)KNKkkkk)kkkk!|)NKKk!|)\n";

std::cout << "kkkkkkkkkkkNKKKKKKkkkkkM`<M` /!|!|||/|Y)|)NMQKW|||!!!!|!!!!)kKkkkk|||Y|||)KMkk|||\n";

std::cout << "kkk)kkkkkkkNNKKKKKkkkkk,))M<`|^||^ ! `',/w,y(kUWw!!!!!!!!))kKKkkvvv,;|||)kkk!|||\n";

std::cout << ")kkkkkkNKKNKKNKKKKKkkk]KNM|)M^ | ` ,wWNNKNMNKKWK(MNW!v!!!!))NKKkkkNKKNk!)k||||||\n";

std::cout << "kKNKKKNKKKKKKKKKKKKKK@K/M)@M| | '|MMMY^'`''\"/`^)k%kMMU/k)k)kNNKNKNKKK@NK!)kKWw,,,\n";

std::cout << "NKKKKK@@@@K@@@@@@@@N@Mk$U@M|/kk(|| \"` 'YMKMNWNW>))KK@@@@@@@@@NK)NNKKNKKK\n";

std::cout << "NNN@@@@@@@@@@@@@@@@0NYk@@^/;Kkx|| '` , ,,xwyyK(kKk)WkNKK@@@@@@@@@NK)KKKNNKKK\n";

std::cout << "NNNN@@@@@@NNNNNNNNNNMK@@WKMMM\y><., , ,wWN)NNkKNNKMMM)M('\*NKKKNN@@@@@@@NN]KNNNNN@@\n";

std::cout << "NNKN@@@@@@@NNKKMM]N@)RKKKMjM(k/KM)KKNNW@W^^\*```` ``')N@NN@@@@@@@@@@@@@@@@@@@\n";

std::cout << "@@@@@@@@@@@@@@@@@@WNKN)ML /)MKKN`(` |!`K@KNN)KK@@@@@@@@@N@@@@N\n";

std::cout << "@@@@@@@@@@@@@@@@NKNMMNMW`,XKM/r^' . ,w,,vK` |^. ]KMMNWM)ypXN@N@@@NKNNNN\n";

std::cout << "NNNNNNNNNNN@@@@KMyM]MYMM!)NM^\ ~>~,r\KW@]kMw .,' KNK(<)MM|kk.|NNNNNNKMkM\n";

std::cout << "!||||))kk)MKK)yWNNNHMMMyw)NN ' ! ')XMNKw@) ,., !)NkW^k> || ^'!|k/KW/)kKk|\n";

std::cout << "kkkk/)kkWW]@MN)K)kU@)Aw)k%NNV, )@N%N0M`]W.KNKkW, | `'`'^ | )vNKN@UKMk\n";

std::cout << "KKK@@NKKKK@N@$KNWMk^k(%\kMMKMM:'/ ,!. ]W ` ,@N\` )|(|kk>~.' ` ]k\*MKNNMN\n";

std::cout << "@@@@@@@@@&@@KM]Q@@KWY^ | '``v|M``' ``` >WU\"%@$@@@Nk|]w '`'``\'` '|)N@@@@@\n";

std::cout << "@@@@@@@@@BNNN@M`| !>`|,w, `: `')Kk` `\"|M | ,,!|| .' '. '|0@@@@\n";

std::cout << "@@@@@@@@@@K@M|'/.!!/k;L)k.v ` ' ''``,awmvv\*` ' \"NNKN\n";

std::cout << "KKN@@@@@@MMM|!`|/ k<\) || V\ | \". `` ! ]K@N\n";

std::cout << "MNNM)]0M)Vk```/|||`>||^`)|,! ,k|. : $@@@\n";

std::cout << ")k)]@`|| M,,>K || `\!| !`,^'|' ' ,'| .'. |@@@\n";

std::cout << "KNNMY/!:`,,(/M /k))Y|\^''|'' ,| ,>!|' NK^\n";

}

void Room::printBoss()

{

std::cout << ",,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,\n";

std::cout << "BBBBBBBBBBBBBBBB@@@@@@@@@@B@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@BBBBBBBBB@@@@@@@@@@BBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB@@BBBB##BBNN@BBBB##BBBB@BBBBB#BBBBBBBBBB######BB@@BBBBBBB@@BBBB##BB&@BBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB@@BB @B&KBB[ ]B` BBBR` \"#B` @BB@@BBBBBB@@BBP ]BB@@BBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB@@B% @B&KBB[ |BL $P g$BB, ;gggg@BB@@BBBBBBB@@BB @BB@@BBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBB@@BB% BB&RBB[ |BL ` ,@BBBBBBN '^^\"$BBN@BBBBB@@N@BBM ]BB$@BBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBB@@BBK BB$@BBK |BL #BBBBBBL @BB@@BBBB@@BBBB@ #% $BB$@BBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBB@@BBM ]BBB@#B$ |BL , \"#BBBBBW ]BBBBBBB&@@BB@@BB $BB@@BBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBB@@BB @B$ ]BL BW `##BB[ ``````BB$@BB&@BB$ gggw \"BB$@BBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBB@@BB$ |BBK @ BBBg ,@BBM BB&@BBB@@BB BBBBL $BB@BBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBB@@B###@BBBBBBBB$gg##Bg##BBBBBB%0BBBM###B#####BBBBB@@@@@@BBB##BBBBB###BBB@@BBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB&@@@@@@@@@@@@@###@@@@@@@N@@BBBB#P^ -aggw, `^###BB@@&@@@@@@@@@@@@@@@@BBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBB@BBBBB@@@BBB#\"` ``` `@P`, , )BBB@@BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBN@BB#^, @BB#w, , , @BBBb w ` \"#BB@@BBBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB@@BBB^ gBBBBP##PP\"\"\"``""\"\*###g, , , @BB@@BBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB@@BBB, #BBBBB g&###BB``""\*w, `#BB$]BB@@BBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB@@BB` BBBBBBBB#BBBBBBBBBB#@w,@N `#B BB&@BBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB&@BB BBBBBBBBBBBBBBBBBBBBBBBBBBggB BB$@BBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB@@BB& ]BBBBBBBBBBBBBBBBBBBBBBBBBBB@ ]BB@@BBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB&@BBW ]BBBBBBBBBBBBBBBBBBBBBBBBBB` @BB@@BBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB@@BB% %BBBBBBBBBBBBBBBBBBBBBBBM @BB@@BBBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB&@@BBBw \"#BBBB#P^BBBBBBBBBBB#` @BB@@BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB@@@@@BBBBB#@ p BBBB@###F` gBBBN@BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBBBB@@@BBB#F^``, gg@$B%gw, , , , , ~ggBBBB@@@BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBBBB@@BBBP` ug$BBBBBBBBBBBB #BBBB#@@@@@@BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBBB@@@BB@, @BBBBBBBBBBBBBBBB[ Bk 5BBB@@@BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBBB@N@BBM gBBBBBBBBBBBBBBBBBB@ ]BB, TBBB$@@@BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBBB@NBB#` @BBBBBBBBBBBBBBBBBBBBL BBBB, \"#BBB@@@@@@BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBBBB@@BB@, BBBBBBBBBBBBBBBBBBBBBB% ]BBBBBg, \"##BBBB@@@@@@BBBBBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBBB@N@BBP / BBBBBBBBBBBBBBBBBBBBBBBBW $BBBBBBBBg, `\*##BBBB&@@@BBBBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBB@NBB#` 4BBBBBBBBBBBBBBBBBBBBBBBBBB BBBBBBBBBBBB%g, \"T#BBB@@BBBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBBBBB@@BB `#BBBBBBBBBBBBBBBBBBBBBBBB@ ]BBBBBBBBBBBBBBB, \"#BB@@BBBBBBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBB@@@@N@B$ ], Bg@BBBBBBBBBBBBBBBBBBBBBBBBK $BBBBBBBBBBBBB@,gBw, , BBN@@@@@BBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBB@N@$BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBL,@BBBBBBBBBBBBBBBBBBBBBBBBBBB@@BBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBB@@BB\"``` `\"T#BBBBBB#P`` `\"\"``BBBB#P^`` ``\"@BB#R^``` ```$BB@@BBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBB@@B$ ,gg, \"BBB#` ]BM BP @BB@BBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB@@BBL ]##M ,BBK g@BBB#L N e#BBBBBBB#@@@ <@BBBBBBB##0BB@@BBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB@@BBL '#BB 4BBBBBB[ @ ``""T##BBB `""^P#BBBB@@BBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB@@BBL ,gw, `# $BBBBBB` BBg,. )BB@w, `#BB@@BBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB@@BBM $BBBB` | #BBBR` @BBB#BBBB##@m @BB#BBBB###w |BB@@BBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB@@BB @B ,@BBBN gBBK aBBB@BBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB@@BB ,g#BBBB@w ,g@BBBB$, ,,wg#BBBB, , , g#BBB@@@BBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBB@@@BBBBBBBBBBBBB#@@@@@BBBBBBBBBB@@@@BBBBBBBBBBBB##@@@BBBBBBBBBBBBB#@@@@BBBBBBBBBBBBBBB\n";

std::cout << "BBBBBBBBBBBBBBBB$&&@@@@@@@@@&@@BBBB$&@@@@@@@@@BBB@$&$@@@@@@&@@@BBBB$&&@@@@@@@&@$BBBBBBBBBBBBBBBBBBBB\n";

}

void Room::lockDoors()

{

m\_EastDoor = false;

m\_WestDoor = false;

m\_NorthDoor = false;

m\_SouthDoor = false;

}

void Room::openDoor(Room Original)

{

(Original.m\_EastDoor == true) ? m\_EastDoor = true : m\_EastDoor = false;

(Original.m\_WestDoor == true) ? m\_WestDoor = true : m\_WestDoor = false;

(Original.m\_NorthDoor == true) ? m\_NorthDoor = true : m\_NorthDoor = false;

(Original.m\_SouthDoor == true) ? m\_SouthDoor = true : m\_SouthDoor = false;

}

**PLAYER CPP FILE:**

#include "Player.h"

#include<iostream>

#include<Windows.h>

Player::Player(bool Sword, bool Key, bool BossKey, int x, int y)

{

m\_hasSword = Sword;

m\_hasTreasure = Key;

m\_hasBossKey = BossKey;

m\_positionX = x;

m\_positionY = y;

}

int Player::moveWest()

{

m\_positionX--;

return m\_positionX;

}

int Player::moveEast()

{

m\_positionX++;

return m\_positionX;

}

int Player::moveSouth()

{

m\_positionY++;

return m\_positionY;

}

int Player::moveNorth()

{

m\_positionY--;

return m\_positionY;

}

void Player::setPrevious(int tempX, int tempY)

{

m\_PreviousX = tempX;

m\_PreviousY = tempY;

}

char \* Player::Response()

{

char PlayerResponse[255];

std::cin.getline(PlayerResponse, 255);

system("cls");

return PlayerResponse;

}

int Player::CheckResponse(MyString Answer, Room Rooms[][5])

{

if (Answer.compare("help") == true)

{

std::cout << "To move type in \"move\" and then the direction you want to move... ex-> \"move east\"\n";

std::cout << "At any time if you want to return to the previous room type \"go back\"\n";

std::cout << "If you encounter an enemy type\"fight\" to attack it.\n";

std::cout << "Other words that could be useful to try at certain times are \"grab\" \"pickup\" \"check\" \"press\" \"go down\" \"drink\"\n\n";

Rooms[m\_positionY][m\_positionX].PrintInfo();

CheckResponse(Response(), Rooms);

}

else if (Answer.findSubString("move") == true)

{

if (Answer.findSubString("east") == true)

{

if (Rooms[m\_positionY][m\_positionX].m\_EastDoor == false)

{

std::cout << "Are you retarded? Try again." << std::endl;

CheckResponse(Response(), Rooms);

}

else

{

moveEast();

}

}

else if (Answer.findSubString("south") == true)

{

if (Rooms[m\_positionY][m\_positionX].m\_SouthDoor == false)

{

std::cout << "Are you retarded? Try again." << std::endl;

CheckResponse(Response(), Rooms);

}

else

{

moveSouth();

}

}

else if (Answer.findSubString("west") == true)

{

if (Rooms[m\_positionY][m\_positionX].m\_WestDoor == false)

{

std::cout << "Are you retarded? Try again." << std::endl;

CheckResponse(Response(), Rooms);

}

else

{

moveWest();

}

}

else if (Answer.findSubString("north") == true)

{

if (Rooms[m\_positionY][m\_positionX].m\_NorthDoor == false)

{

std::cout << "Are you retarded? Try again." << std::endl;

CheckResponse(Response(), Rooms);

}

else

{

moveNorth();

}

}

}

else if (Answer.findSubString("grab") == true || Answer.findSubString("pickup") == true)

{

if (Answer.findSubString("key") && Rooms[m\_positionY][m\_positionX].m\_HasTreasureKey == true)

{

std::cout << "You picked up the treasure key." << std::endl;

m\_hasTreasure = true;

}

else if (Answer.findSubString("sword") && Rooms[m\_positionY][m\_positionX].m\_HasSword == true)

{

std::cout << "You picked up the shining shortsword." << std::endl;

m\_hasSword = true;

}

else if (Answer.findSubString("key") && Rooms[m\_positionY][m\_positionX].m\_HasBossKey == true)

{

std::cout << "You picked up the key to the boss room." << std::endl;

m\_hasBossKey = true;

Rooms[3][2].m\_SouthDoor = true;

}

}

else if (Answer.findSubString("fight") && (Rooms[m\_positionY][m\_positionX].m\_HasEnemy == true || Rooms[m\_positionY][m\_positionX].m\_HasBoss == true))

{

if (Rooms[m\_positionY][m\_positionX].m\_HasBoss == true && m\_hasSword == true)

{

Rooms[m\_positionY][m\_positionX].m\_HasBoss = false;

return 1;

}

if (m\_hasSword == false)

{

std::cout << "You charged forward and were decimated.\n\nGAME OVER" << std::endl;

Sleep(2000);

return -1;

}

else

{

std::cout << "You stabbed the creature in the chest and killed it.\nWhat will you do?";

Rooms[m\_positionY][m\_positionX].m\_HasEnemy = false;

Rooms[m\_positionY][m\_positionX].openDoor(Rooms[m\_positionY][m\_positionX]);

}

}

else if (Answer.findSubString("press") && Rooms[m\_positionY][m\_positionX].m\_HasDeathButton == true)

{

std::cout << "You pressed the button and spikes shot out from in front of you impaling you in 50 different places...\n\nGAMEOVER" << std::endl;

Sleep(2000);

return -1;

}

else if ((Answer.findSubString("go down") || Answer.findSubString("jump down") || Answer.findSubString("down")) && Rooms[m\_positionY][m\_positionX].m\_HasTrapHole == true)

{

std::cout << "You went down into the hole and landed on a bed of spikes...\n\nGAME OVER" << std::endl;

Sleep(2000);

return -1;

}

else if ((Answer.findSubString("drink") || Answer.findSubString("eat")) && m\_positionY == 4 && m\_positionX == 4)

{

std::cout << "You drank the strange liquid and you were teleported to a random room." << std::endl;

do

{

m\_positionX = rand() % 5;

m\_positionY = rand() % 5;

} while (Rooms[m\_positionY][m\_positionX].m\_Empty == false);

}

else if ((Answer.findSubString("check") || Answer.findSubString("open")) && Rooms[m\_positionY][m\_positionX].m\_HiddenDoor == true)

{

if (Answer.findSubString("crack"))

{

std::cout << "You managed to open a secret door.\n";

moveSouth();

}

}

else if (Answer.findSubString("go back") || Answer.findSubString("back"))

{

m\_positionX = m\_PreviousX;

m\_positionY = m\_PreviousY;

}

else if (Answer.findSubString("open") && m\_positionY == 1 && m\_positionX == 1)

{

if (m\_hasTreasure == true)

{

std::cout << "You opened the treasure chest and it was completely empty...\n";

}

else

{

std::cout << "You need the key.\n";

}

}

else

{

std::cout << "Are you retarded? Try again." << std::endl;

CheckResponse(Response(), Rooms);

}

return 0;}

**READ ME:**

You can access the files with this link: https://github.com/MaxGuidry/Text-Based-Adventure

Click the button that says “Clone or download” button listed on the top right in this page, then click “Download ZIP” so that you can access each individual file.

The String-Class folder contains the source and header files for my game.

The Dungeon Crawler zip folder contains the executable if you would like to just play the game.

If you would like to access the files without downloading the ZIP:

An executable to start the program and test it is also enclosed in the repository.

To access this executable click on the file named “Dungeon Crawler.zip” and download that file using the download button near the history and delete button.

The assessment documentation is enclosed in the repository.

When the game is started up you can follow the instructions given to you by the game.