

## Project 2

Title:  
A Robot,  
A Madman,  
and Too Many Burgers

a text based  
science fiction RPG

Course:  
CSC-17A  
Due Date:  
12/12/15  
Author:  
Fred Roybal IV

# Introduction

Title: A Robot, A Madman, and Too Many Burgers

This is a text based role-playing game where the user undertakes the role of an unnamed android aboard the starship "Iron Biscuit". The ship is then attacked by an alien invader and must be confronted.

The user will be given a menu of options when required along with the necessary info on those options.

Example:

Menu////////

1:Right Away!

2:I'll be right there!

3:Hold your horses tubby!

//////////

The game ends at either the player or enemies defeat.

## Summary

Size of Project: 588 lines including spacing

Number of Functions: 9 or 10 if main is included

Number of Variables: 28 in total

The project focused on providing examples of the subject matter of chapter 9 through 16 of the Gaddis textbook that was assigned for the class. Although due to technical malfunctions on my main computer and lack of backup, I was forced to make do with what time I had in light of this

## Summary

The inspiration for this game came in the form of just letting my mind wander and allow a scenario to develop. As the general idea of a science fiction game came about, I drew upon inspiration from the book "The Hitchhikers Guide to the Galaxy" and the game series "Mass Effect" to further flesh out the game.

# Flow Chart

# Pseudo Code

*Initialize*

*Load Libraries*

*Create Function Prototypes*

*Create Structure for Menus*

*Call Title Screen Function*

*Declare Variables*

*Prompt User to start the game and authenticate input via switch statement*

*If input is a letter C, continue game*

*Else exit program*

*Load Intro Text*

*Call Dialogue Function*

*Assign phrases to menu structure*

*Output Menu options*

*Wait for User Input*

*Output response based off previous input*

*Create array to store name of enemy*

*Output name of enemy*

*Initialize variables for both main character and enemy statistics*

*Do*

*If main character health below 0*

*return variable for Game Over*

*If main character health above 0*

*Prompt user for input on their turn*

*If user chooses to use to attack*

*execute basic attack code*

*If user brings up skill menu*

*execute a chosen skill*

*Else if user inputs an invalid option*

*skip player turn*

*If enemy health is below 0*

*return variable to continue game*

*If enemy health is above 0*

*enemy attacks player and deals damage*

*While the enemy is still up*

*End Program*

# Program

```
/*
 * File: Fight1.h
 * Author: Fred Roybal IV
 *
 * Created on December 12, 2015, 8:53 AM
 */

#ifndef FIGHT1_H
#define FIGHT1_H

class Fight1 {

private:
    short pHp;

public:
    Fight1();                //Constructor

    short setPhp(int);        //Enemy Health Points
    short getPhp() const;     //Retrieves Enemy Health

    Fight1();
    Fight1(const Fight1& orig);

    ~Fight1();                //Destructor
};

//Initialize starting enemy health

Fight1::Fight1(){

    pHp=70;

}

//Sets Player Health Points
short Fight1::setPhp(int x){

    pHp=x;

    if{php>71}{
        cout<<"ERROR: Enemy HP is above what it is supposed to be"<<endl;
        exit(EXIT_FAILURE);

    }

}
```

```

}

short Fight1::getPhp() const {

    return pHp;

}

#endif /* FIGHT1_H */

/*
 * File: Fight1.cpp
 * Author: User
 *
 * Created on December 12, 2015, 8:53 AM
 */

#include "Fight1.h"

Fight1::Fight1() {

}

Fight1::Fight1(const Fight1& orig) {

}

Fight1::~Fight1() {

}

/* File: main.cpp
 * Author: Fred Roybal IV
 * Created on October 19, 2015, 11:37 AM
 * Purpose: Project 1. A text based RPG set in space to show what was taught so
 * far in the class
 */

//System Libraries
#include <iostream>
#include <cstring>
#include <cstdlib>
#include <fstream>
#include <ctime>
using namespace std;

//User Libraries
#include <Fight1.h>

//Global Constants

```

```
const unsigned short CFTIRON=250;    //The approximate weight of a cubic ft of
                                     //iron on earth
```

```
//Function Prototypes
```

```
void TtlScrn ();                    //The title screen
void Continu ();                   //Used to pause text
void Talk1 ();                     //First Conversation in game
bool Battle1 (short, short, short, float); //Enemy Battle
unsigned short CritRt (short, float); //Used for an effect in Battle1
```

```
struct dMenu{                      //Used to store dialogue choices
    string item1;
    string item2;
    string item3;
};
```

```
int main(int argc, char** argv) {
    //Declare Variables
    srand(static_cast<unsigned short>(time(0))); //Used for random numbers
```

```
    bool isValid=false;            //Used to validate menu choices
    unsigned short mGen=0;          //Used to pick monster to fight
```

```
    char gamStrt=0;                //Used for Main Menu
```

```
    short gameOvr=0;               //Tests for game overs
```

```
    short rwdMenu=0;               //Used for Reward selection
```

```
TtlScrn();
cout<<endl;
```

```
cout<<"Type the letter C and press the enter key afterwards to continue"<<endl;
cout<<"Or type in a space or tab to stay here"<<endl;
cout<<"Or type anything else to exit"<<endl;
```

```
cin>>gamStrt;                     //Used for waiting until the user
cin.ignore();                      //is ready
cout<<endl;
```

```
switch (gamStrt){
    case 'C':
    case 'c':
        break;
    default:cout<<"See You Space Cowboy...";
        return 0;
}
```

```

//Output Intro to terminal
fstream textfile;
textfile.open ("Intro.txt");

string temp;
while(getline(textfile, temp)){
    cout<<temp;
}

textfile.close();

cout<<endl;
cout<<"Whenever you see the line below, type in anything but a space or tab to continue"<<endl;
Continu();

Talk1();

mGen=rand()%3+1;

char mName[10];
mName[1]='B';
mName[2]='u';
mName[3]='r';
mName[4]='g';
mName[5]='e';
mName[6]='r';

cout<<"At that exact moment, Joe was suddenly crushed by a giant alien";
cout<<" monstrosity only known as:"<<endl;

for(unsigned int index=1;index<=6;index++){
    cout<<mName[index];
}

cout<<endl;

Continu();
cout<<endl;

//Initiates battle function and checks for a "Game Over"
gameOvr=Battle1 (100,10,13,11);

//Ends game if main character is defeated

```



```

if(gameOvr==1){
    return 0;
}

cout<<"After the monster falls dead at your feet, you notice three"<<endl;
cout<<"parts of its body begin to shine. Your scanning function is"<<endl;
cout<<"picking up useful materials where the body is shining."<<endl;

Continu();

enum Reward1 { FILLER, FANG, HEART, BRAIN };

cout<<"Which body part do you harvest?"<<endl;

cout<<"Menu/////"<<endl;
    cout<<"1:Sesame Fang"<<endl;
    cout<<"2:Meaty Heart"<<endl;
    cout<<"3:Tomato Brain"<<endl;
    cout<<"/////////"<<endl;

    cin>>rdwMenu;

    if (rdwMenu==FANG){

        cout<<"You grabbed the Sesame Fang and put it in your chest compartment"<<endl;
        Continu();
        cout<<"All of a sudden, the reactor of the ship fails and causes the ship to explode"<<endl;
        cout<<"Your head and torso manage to survive the blast and miraculously the descent to
a"<<endl;
        cout<<"a nearby planet.";
        Continu();
        cout<<"Whats left of your chasis is recovered along with the body part of the monster you
defeated"<<endl;
        cout<<"You are hailed as a hero among robotic fry cooks everywhere and live a life of
luxury"<<endl;
        cout<<"relaxing at a robotic day spa until you finally cease to function"<<endl;

        cout<<"THE END"<<endl;
    }else if (rdwMenu==HEART){
        cout<<"You grabbed the Meaty Heart and put it in your chest compartment"<<endl;
        Continu();
        cout<<"All of a sudden, the reactor of the ship fails and causes the ship to explode"<<endl;
        cout<<"Your head and torso manage to survive the blast and miraculously the descent to
a"<<endl;
        cout<<"a nearby planet.";
        Continu();
        cout<<"Whats left of your chasis is recovered along with the body part of the monster you
defeated"<<endl;
        cout<<"You are hailed as a hero among robotic fry cooks everywhere and live a life of

```

```

luxury"<<endl;
    cout<<"relaxing at a robotic day spa until you finally cease to function"<<endl;

    cout<<"THE END"<<endl;
}else if (rwdMenu==BRAIN){
    cout<<"You grabbed the Tomato Brain and put it in your chest compartment"<<endl;

    Continu();
    cout<<"All of a sudden, the reactor of the ship fails and causes the ship to explode"<<endl;
    cout<<"Your head and torso manage to survive the blast and miraculously the descent to
a"<<endl;
    cout<<"a nearby planet.";
    Continu();
    cout<<"Whats left of your chasis is recovered along with the body part of the monster you
defeated"<<endl;
    cout<<"You are hailed as a hero among robotic fry cooks everywhere and live a life of
luxury"<<endl;
    cout<<"relaxing at a robotic day spa until you finally cease to function"<<endl;

    cout<<"THE END"<<endl;
}

```

```

    return 0;
}

```

```

//The Title Screen
void TtlScrn (){

```

```

    cout<<"          Fred Roybal IV Presents:          \n";
    cout<<"*****\n";
    cout<<"-----\n";
    cout<<"          A Robot, A Madman,          \n";
    cout<<"          and          \n";
    cout<<"          Too Many Hamburgers          \n";
    cout<<"-----\n";
    cout<<"*****\n";
    return;
}

```

```

//The function used to space out text and wait until the player is ready to continue

```

```

void Continu (){
    string verify;

    cout<<"Continue..."<<endl;

    cin>>verify;

```

```

    cout<<endl;
}

void Talk1(){

    short mSlct=0;                //Used for choosing menu options
    short *mPoint=&mSlct;

    bool isValid=false;

    cout<<endl;
    cout<<"Joey: \"Hey, hurry it up now wouldja!? I'm staavin' over here!\"<<endl;
    cout<<endl;
    cout<<"Whenever you see a menu like the one below, type in the number next to"<<endl;
    cout<<"the option to select it"<<endl;

    string mItem1= "1:Right Away!";
    string mItem2= "2:I'll be right there!";
    string mItem3= "3:Hold your horses tubby!";

    dMenu temp;
    temp.item1=mItem1;
    temp.item2=mItem2;
    temp.item3=mItem3;

    do{

        cout<<"Menu////<<endl;
        cout<<temp.item1<<endl;
        cout<<temp.item2<<endl;
        cout<<temp.item3<<endl;
        cout<<"////////<<endl;

        cin>>*mPoint;

        switch (*mPoint){
            case 1:cout<<"Good, and don't forget the pickles on my burger this";
                    cout<<" time yah?"<<endl;
                    isValid=true;
                    cin.ignore();
                    break;
            case 2:cout<<"Quit ya lolligaggin' and get a move on!"<<endl;
                    isValid=true;
                    cin.ignore();
                    break;
            case 3:cout<<"Ah putta sock in it tin can and get me mah breakfast."<<endl;
                    isValid=true;
                    cin.ignore();
                    break;
        }
    }
}

```

```

        default:cout<<"That answer isnt valid..."<<endl;
            cin.ignore();
            break;
    }

}while(isValid==false);

}

bool Battle1 (short mcHp, short mcStr, short mcMag, float mcCrit){

    //Declare Variables
    Fight1 fOne;          //Bring in class

    unsigned char mnstrUp=1;    //Checks if the enemy is still alive

    unsigned char Cmd;          //For tracking user input

    unsigned char trnOver=0;    //Used for tracking who's turn it is

    unsigned char ablMenu;      //Used for tracking which ability a user
                                //might choose locally

    short pHp=70;              //local Statistics of enemy
    unsigned short pStr=8;

    bool gameOvr;              //Tracks if the user is defeated

    cout<<"(You are in a battle!)"<<endl;
    cout<<"(The battle is over when either you or the enemy is defeated)"<<endl;

    Continu ();

    do{

        //Signals for when the player is defeated
        if(mcHp<=0){
            cout<<"The Burger covers you in cheese and eats you..."<<endl;
            Continu ();
            cout<<"You permanently malfunction and go to robot heaven.";
            Continu ();
            cout<<"GAME OVER"<<endl;
            return gameOvr=true;
            cout<<endl;
            ;
        }

        //List of the user's available actions

```

```

if(mcHp>0){
cout<<"Menu*****"<<endl;
cout<<"1: Attack"<<endl;
cout<<"2: Use Skill"<<endl;
cout<<"3: Make Sounds"<<endl;
cout<<"*****"<<endl;
cin>>Cmd;

switch (Cmd){
//Basic Attack command
case '1':
    cout<<"You attack the burger!"<<endl;
    cout<<"It has "<<endl;
    cout<<(fOne.setPhp(pHp)-=(rand()%39+1)+(mcStr))<<endl;
    cout<<"health left"<<endl;
    ablSlct++;
    break;
//Menu for skill select
case '2':
    do{
    cout<<"Menu*****"<<endl;
    cout<<"1: Calculated Servings "<<endl;
    cout<<"2: Spatula Hurricane"<<endl;
    cout<<"3: ACCESS THE MATRIX"<<endl;
    cout<<"*****"<<endl;
    cin>>ablMenu;

    switch (ablMenu){
        case '1':
            cout<<"You focus on hitting a weak point\n";
            (fOne.setPhp(pHp)((rand()%14+1),50))
            cout<<fOne.getPhp()<<endl;
            cout<<"The Burger has "<<pHp<<" health left.";
            ablSlct++;
            break;

        case '2':
            cout<<"You coil yourself rapidly to charge up, but you hurt";
            cout<<" yourself to deal more damage."<<endl;
            cout<<"The pudding has "<<endl;
            (fOne.setPhp(pHp)-=(rand()%34+20))
            cout<<fOne.getPhp()<<endl;
            cout<<" health left."<<endl;
            cout<<"You hurt yourself in the attack!"<<endl;
            cout<<"You have "<<(mcHp-=(rand()%4+1))<<" health left"<<endl;
            ablSlct++;
            break;

        case '3':

```

```

        cout<<"You focus and access the Matrix"<<endl;
        cout<<"Your strength has increased to ";
        cout<<(mcStr+=(rand()%4+(mcMag-3)))<<endl;
        ablSlct++;
        break;

        //Used for when invalid input is entered from skill menu
        default:
        cout<<"Your circuits hurt from trying to compute"<<endl;
        cout<<"(That isn't a valid response)"<<endl;

        Continu ();
        break;
    }

    }while (ablSlct==0);
    break;
//A comical action the user can do
case '3':
    cout<<"You fling robot swear words at the burger"<<endl;
    cout<<"The Burger just got dissed"<<endl;
    cout<<rand()%99+1;
    cout<<" style points gained"<<endl;
    ablSlct++;
    break;
//Used for when invalid input is entered from main menu
default:
    cout<<"Your circuits hurt from trying to compute"<<endl;
    cout<<"(That isn't a valid response)"<<endl;
    break;

}
//Checks if enemy is defeated
if(fOne.getPhp()<=0){
    --mnstrUp;
    cout<<"You defeated the Burger!!!"<<endl;
    Continu ();
    cout<<"You are now promoted to Cuisine Combatant!"<<endl;
    cout<<"The Burger dropped 1132 credits"<<endl;
    Continu ();
}
//Calculates and tracks damage to user when enemy attacks
if(fOne.getPhp()>0){
    cout<<"The Burger spits cheese at you!"<<endl;
    cout<<"You have "<<(mcHp=(rand()%19+1)+pStr)<<" left"<<endl;
    cout<<endl;
}
}

```

```

    }while(!mnstrUp==0);

    return gameOvr=false;
}

unsigned short CritRt (short base, float rate){

    unsigned short damage;
    unsigned short crit;
    float calc=rand()%99+1.5; //Randomly generates a number to decide if an
                             //attack critically strikes

    if (calc<=rate){
        //Output notification if succeeded
        cout<<"CRITICAL HIT!!! ";
        damage=base*2;
    }else{
        //Output normal damage if failed
        damage=base;
    }

    return damage;
}

```