Jose Jimenez

Project 2

Professor Lehr

CSC 17-A

Introduction

My project is a very basic game. It allows the user to name and age their hero. The user is then asked given the choice of going through either the left door or the right door. Each door consisting of a different tasks. The left asks the user a riddle and the right places the user in a fight with a dragon. Completing either task will result in the completion of the game. After solving the riddle the user will be asked for another decision one will let the player play a game of hangman and the other will start a mach of tic tac toe, lastly the game will start a game called jumble where the goal is too unscramble the word after completing it the game will end.

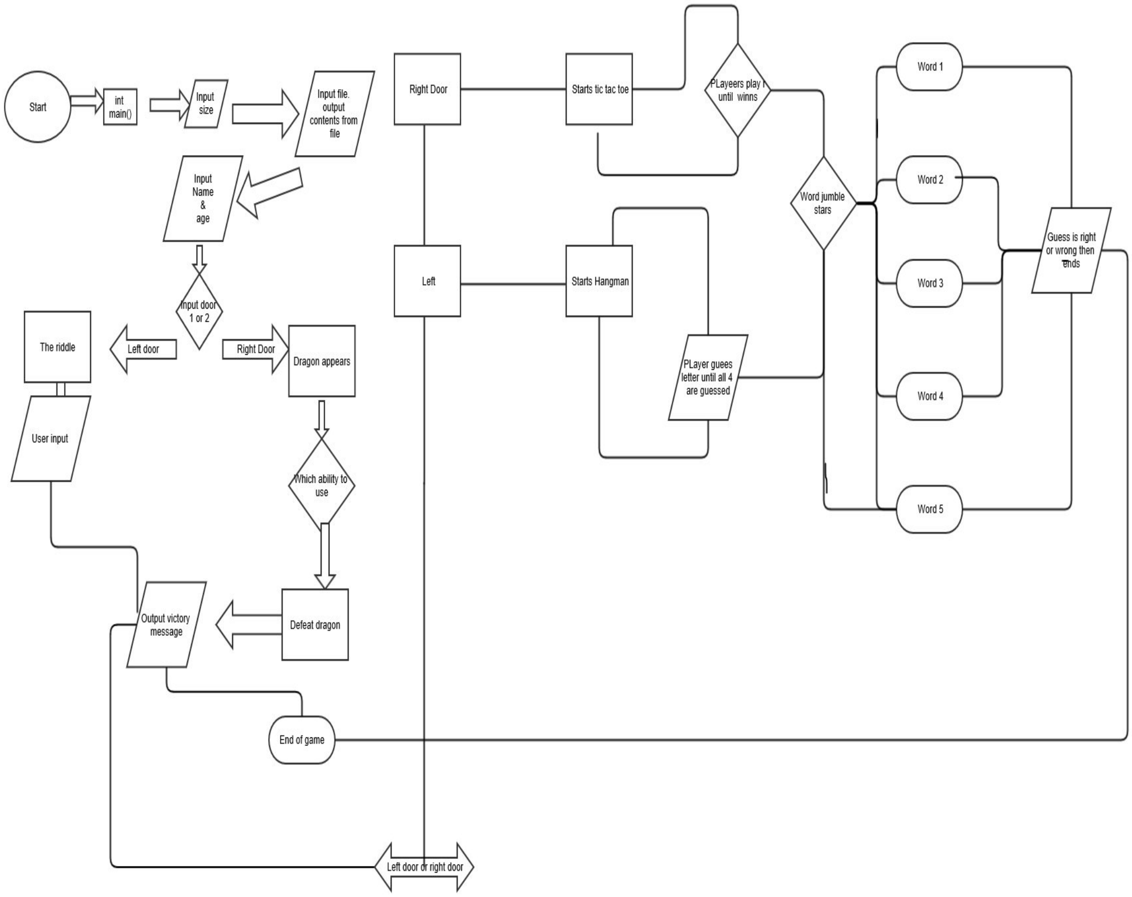
Summary

The program uses many of the same topics learned in the first half of the seamster. The first function uses the fstream operator. I also used 2 structures and called both of them into a function. The main function also has a pointer which created the dynamic array. Strings are constantly being used. The program is 1004 lines with about 35 variables(I only counted it once if it was used multiple times), 16 void functions, and 2 structs(each with 2 variables);It has one class with accessors and mutators. So it uses most of everything we have learned up to this point in the class. I was unable to use UML because gliffy charges users to use that feature.

Description

The main function of this program is to utilize concepts that we covered through the use of a small choice based game.

Flow Chart



Major Variables

Int		
String name	Located in struct Character and Enemy Called in main(), void create(Character &), stage1(Character), rdoor(Enemy &)	Used to find the name of the hero, inputted by the hero and name of the enemy.
Integer age	Struct Character, void create(Character)	Used to enter the name of the character
Integer health	Used in rdoor(Enemy &)	Used to determine the

		health of the Hero and the dragon. The dragons health is called from the Enemy struct but the heroes was created in the function
String rid1	Located in void ldoor()	Used for the user to answer the riddle
int *p	Located in main()	To determine the amount of times user wants to play the game and used to create dynamic array
String door	Located in void stage1(Character Hero)	Use makes choice in which door to go in.
int ability	Located in void rdoor(Enemy &)	User chooses how they wish to attack dragon number can from 1-4
int i	Used twice once in main in the for loop and again in void rdoor(Enemy &)	Used to determine how many times the user wished to play the game and which attack the enemy would use
Newname Newweight Newheight	Used for the class	Used to create the character in the class
String blankboard[]	Used for hangman	User fills the word when the correct word is guessed
String WORDS	Used for the jumblr game	Used to create 2d arrays for words and hints
numMoves	Used in tictactoe	Used to determine when the ame is tied

Concepts

Structures-

```
struct Enemy{
int health;
string name;
};

struct Character{
    string name;
    int age;
};
```

Used in the void create, rdoor, and stage() functions.

Pointer and dynamic arrays-

```
cin >> size;
    //Allocate the array
    int *p = new int[size];
```

Used in the main function.

Fstream-

```
ifstream intro;
    intro.open("Intro.txt");


    if(!intro.is_open()){
        exit(EXIT_FAILURE);}

    string intro2;

    while(!intro.eof()){
        cout << intro2 << " ";
        intro >> intro2;
    }
    //Close file
        intro.close();
```

Used in the void intro() function

Strings- Used everywhere in the program

Classes, accesors, mutators- All used in the header file in the class charact,

References

Gaddis 9th ed.

Code

```
//Jose Jimenez
//Game project
// 12/11/2013


//Libraries
#include <iostream>
#include <fstream>
#include <iomanip>
#include <ctime>
#include <string>
#include <cstdlib>

//Include the header file.
#include "Character.h"



using namespace std;

//Declare struct
struct Enemy{
int health;
string name;
};

struct Character{
    string name;
    int age;
};

//Function prototypes
void intro();
void create(Character &);
void stage1(Character);
void ldoor();
void rdoor(Enemy &);
void rstage2();
void lstage2();
void drawing(int position);
void guessword(string word);
void hangman();
void showBoard(char board[]);
bool checkForWinner(char board[]);
void tictac();
void jumble();





int main()
{
```

```
int size;
cout << "How many times do you want to play the game?"  << endl;

cin >> size;
//Allocate the array
int *p = new int[size];
//Begin loop

    for (int i =0; i<size; i++){


Character hero;
//Function calls
intro();

cout << endl;
cout << endl;

create(hero);

    int height = 0;
float weight = 0.0;
// Create the prof for player 1
string name;
cout << endl;
cout << "Enter your weight: " << endl;
cin >> weight;
cout << "Enter your height: " << endl;
cin >> height;
cout << "And once more enter your name" << endl;
cin>> name;

Charact classe(name, height, weight);
// Display the prof

cout << endl << "Character name: " << classe.getName() << endl
    << "Height: " << classe.getHeight() << endl
    << "Weight: " << classe.getWeight() << endl << endl;
cout << "Since some games require 2 people lets  create a second profile!" << endl;

cout << "Enter your weight: " << endl;
cin >> weight;
cout << "Enter your height: " << endl;
cin >> height;
cout << "And once more enter your name" << endl;
cin>> name;

cout << endl;

Charact hero2;

// Create hero 2

hero2.setName(name);
hero2.setWeight(weight);
hero2.setHeight(height);

stage1(hero);
```

```
        }
    string w;
    cout << "Enter any key to exit the game";

    //Delete allocated memory
    delete [] p;

    //Program ends here
    cin.ignore();
    getline(cin, w);
    return 0;
}

void intro()
{
    //Read into file
    ifstream intro;
    intro.open("Intro.txt");


    if(!intro.is_open()){
        exit(EXIT_FAILURE);}

    string intro2;

    while(!intro.eof()){
        cout << intro2 << " ";
        intro >> intro2;
    }
    //Close file
        intro.close();



}

void hangman(){


    int number=0;

     int count = 0;
     int position=0;

    string word;

    // OPen hangman word

    ifstream infile;
    infile.open("squa
        re.txt");


    getline (infile, word);

    guessword(word);
}
```

```
//Creates the users character
void create(Character &b)
{

    string name;
    cout << "Now that you ready to begin your journey lets create your hero!" << endl;
    cout << "Please enter the name of your hero - " ;

    //User input
    cin.ignore();
    getline(cin,b.name);
    cout << endl;




    cout << "Now how old is your hero? - ";
    cin >> b.age;


    cout << endl;
    if(b.age<21)
    {
        cout << "Oh still a minor are we? Well no matter, your never to young to become
a hero!" << endl;
    }

    else{
        cout << endl;
        cout << "Aren't you a bit to old to be a hero? hmmmm.....well no matter, lets
go anyways!" << endl;
    }
}

void stage1(Character hero)
{
    //call structure
    Enemy dragon;

    string door;
    string x;

    cout << endl;
    cout << "To continue hit enter" << endl;
    cin.ignore();
    getline(cin, x);

    cout << "So " << hero.name << " should we take the left or right door?" << endl
<< endl;
    cout << "Behind one lies a monster and the other a puzzle" << endl << endl;
    cout << "!!!Remember all decisions may affect the games outcome!!!" << endl << endl;
    cout << "Enter 1 to go through the left and 2 for the right." << endl;

    cin >> door;

    while(door != "1" && door !="2")
    {

        cout << "ummm please enter 1 or 2, try again." << endl;
        cin >> door;
    }
    if(door == "1"){
        ldoor();}
    else{

        rdoor(dragon);
    }

}

//Function for the left door
void ldoor()
{
    string rid1;

    cout << endl;
    cout << "Look at the ceiling their is writing on the wall. " << endl;

    cout << "What does it say?" << endl << endl;

    cout << "SOLVE THE RIDLLES AND THE DOOR WILL UNLOCK" << endl << endl;

    cout << "Riddle 1: Mary's father has 4 children;"
         << "three are named Nana, Nene, and Nini. So what is is the 4th child's name?"<<
endl;

    //The answer to the riddle is Mary
    cin >> rid1;

    //input validation;
     while(rid1 != "Mary")
     {
         cout << endl;
         cout << "BEEP! that is incorrect please try again" << endl;
         cout << "Remember capitalization counts." << endl;
         cin >> rid1;
         cout << endl;
     }

     cout << "Ding dong that is correct! You win! " << endl << endl;
     cout << endl;

     cout << "Congrats now we can move on to te next door" << endl;
     cout << endl;

     rstage2();
}

//Function for the right door
void rdoor(Enemy &d)
{
    int ability = 0;
    //Access the enemy structure
    d.health = 100;
```

```
d.name = "Red dragon";

int health = 100;
int i=0;

cout << "The " << d.name << " has appeared!" << endl;
cout << "Looks like we have no choice but to fight it. " << endl;

//Continue fidht until dragon dies
 while(d.health>0)
 {


     cout << "What attack should we use?" << endl << endl;
     cout << "Enter the number corresponding to the attack" << endl;


     cout << "1. Fireball"  << endl
          << "2. Water shot" << endl
          << "3.Heal" << endl
          << "4.Hurricane" << endl;
     cout << endl;

     //Choose what power to use
      cin >> ability;

      //Decrease dragons health accordingly
      if(ability == 1)
      {
          cout << "You attacked with fireball" << endl;
          d.health -= 25;

      }

      else if(ability == 2)
      {
           cout << "You attacked with Water Shot" << endl;
           d.health -= 20;
      }

        else if(ability == 3)
      {
           cout << "You healed yourself" << endl;
           health += 20;
      }
        else
      {
           cout << "You attacked with Hurricane" << endl;
           d.health -= 30;

      }

 //Exits the loop after you chose ability
 cout << "The enemy now has " << d.health <<  " HP" << endl << endl;

 //If dragon is dead then brak out of the
 if(d.health <= 0)
     {
             break;
         }

     //Enemy fights back
     if(i==0)
     {
         cout << "The enemy used slash!" << endl;
         health -= 10;
          cout <<"Your HP is now at " << health << endl << endl;
        i++;
     }

     //Counter to one and he will use this
     else if(i==1)
     {
       cout <<"The enemy used firebreath!" << endl << endl;
       health -= 15;

       cout <<"Your HP is now at " << health << endl << endl;
        i++;
     }
     else if(i==2)
     {
       cout <<"The enemy used Wind attack!" << endl << endl;
       health -= 5;

       cout <<"Your HP is now at " << health << endl << endl;
    i++;
     }
     else
     {
       cout <<"The enemy used fire tornado!" << endl << endl;
       health -= 15;

       //Tell user how much health he has after every turn
       cout <<"Your HP is now at " << health << endl << endl;
       i--;
     }



     }

     //Exits llop when dragon is uner 100 health
     cout << "You have slain the Red dragon and have emerged victorious" << endl;
}

void rstage2(){

    //choice for the next door
    char choice;

    cout << "Well we somehow made it out of that one still in one pice huh?" << endl;
    cout << "Look there is another sign. What does it say?" << endl;
    cout << "Choose a door" << endl;

    cout << endl;
    cout << " Both doors have diferent pictures on them" << endl;

    cout << endl;

    cout << "The one on the right has a picture of a man hanging" << endl;
    cout << "Thedoor on the right has a picture of a 3X3 board" << endl;

    cout << endl;

    cout << "Which door do you want to enter?" << endl;

    cout << endl;

    cout << "To go into the right door enter 'R' for the left door enter any other letter"
<< endl;


    //Enter what door to enter
    cin >> choice;

    //Determine what door

    if(choice == 'R'){

        cout << "You will now have to play a game of hangman" << endl;
        cout << endl;

        cout << "Guess all the words correctly and you will move on to the final game"
<< endl;

        //Plays hangman
        hangman();

    }
    else
    {
        //Plays tic tac
        tictac();
    }

    cout << endl;

    cout << "Congrats now you are on the final stage" << endl;

    //The final game is jumble

    jumble();
}

//Function for hangman
void guessword(string word)
     {

     // Initialize variables
     char letter;
     int position;

     //Creaye string for word
```

```
   string blankword[4];

   //Fill the array
   blankword[0] = "_";

   blankword[1] = "_ ";

   blankword[2] = "_ ";

   blankword[3] = "_ ";


   //Used as ounter
   int x = 1;
  int y=0;
  for (int i=1;i<8;i++)
  {
   cout << "What letter would you like to guess?";
   cin >>letter;

  position = word.find(letter);

  if (position > word.length()){

   cout<<letter<< " is not in the word "<<endl;

   // Next case
   drawing(x++);
  }

  //Use only get 8 attempts
if(x==8)
{

    string l;
    cout << "GAME OVER!" << endl;

  cin.ignore();

  getline(cin, l);

      exit(EXIT_FAILURE);
}

  else
  {
  cout<< letter << " is in the word"<<endl;

  //Creats new word after guess
  if(position==0)
  {
   blankword[0] = letter;

   cout << blankword[0]<<blankword[1]<<blankword[2]<<blankword[3];
  y++;
   i--;

   if (y==4){
```

```
  cout << "You win" << endl;
   cout << endl;
   cout<<blankword[0]<<blankword[1]<<blankword[2]<<blankword[3];
    cout << endl;
        break;
 }
 }


//Insert second letter
 else if(position==1)
 {

 blankword[1] = letter;

 cout<<blankword[0]<<blankword[1]<<blankword[2]<<blankword[3];
y++;
 i--;
 if (y==4){
  cout << "You win" << endl;
   cout << endl;
   cout<<blankword[0]<<blankword[1]<<blankword[2]<<blankword[3];
    cout << endl;
        break;
 }
 }


//Insert 3rd letter
 else if(position==2)
 {

 blankword[2] = letter;


 cout<<blankword[0]<<blankword[1]<<blankword[2]<<blankword[3];

 //counter to find when word was guessed
 y++;
 i--;
 if (y==4){
  cout << "You win" << endl;
   cout << endl;
   cout<<blankword[0]<<blankword[1]<<blankword[2]<<blankword[3];

    cout << endl;
        break;
 }
 }

 //Insert fourth letter
 else if (position==3)
 {
 blankword[3] = letter;
  cout << endl;

 cout<<blankword[0]<<blankword[1]<<blankword[2]<<blankword[3];
 y++;
```

```
i--;

if (y==4){
 cout << "You win" << endl;
 cout << endl;
  cout<<blankword[0]<<blankword[1]<<blankword[2]<<blankword[3];

       break;
}
}

}


}



}




void drawing(int position)
{

switch(position)
{

case 1:
cout << " ___________"<<endl;
cout << " | }"<<endl;
cout << " | " <<endl;
cout << "_|______________"<<endl;
++position;
break;

case 2:
cout << " ___________"<<endl;
cout << " | }"<<endl;
cout << " | \\ " <<endl;
cout << "_|______________"<<endl;
break;

case 3:
cout << " ___________"<<endl;
cout << " | }"<<endl;
cout << " | \\ O " <<endl;
cout << "_|______________"<<endl;
break;

case 4:
cout << " ___________"<<endl;
cout << " | }"<<endl;
cout << " | \\ O /" <<endl;
cout << "_|______________"<<endl;
break;
```

```
    case 5:
    cout << " ___________"<<endl;
    cout << " | }"<<endl;
    cout << " | \\ O /" <<endl;
    cout << " |  | "<<endl;
    cout << "_|______________"<<endl;
    break;

    case 6:
    cout << " ___________"<<endl;
    cout << " | }"<<endl;
    cout << " | \\ O /" <<endl;
    cout << " |  |"<<endl;
    cout << " |  / "<<endl;
    cout << "_|______________"<<endl;
    break;

    case 7:
    cout << " ___________"<<endl;
    cout << " | }"<<endl;
    cout << " | \\ 0 /" <<endl;
    cout << " |  |"<<endl;
    cout << " |  / \\ "<<endl;
    cout << "_|______________"<<endl;

    }
    }



    void showBoard(char board[])
    {
    cout << endl;

    //Loop that creates the board
    for (int i=0; i<9;i++)
    {
    cout << board[i]<< " ";

    // Create a 3x3 board
    if (((i+1) % 3) == 0)
        cout << endl;
    }
    cout << endl;
    }

    //Determine the wimnner
    bool checkForWinner(char board[])
    {
    //Initialize i
    int i;

for(i=0;i<9;i+=3)

if(board[i]==board[i+1]&&board[i]==board[i+2])
{
    cout<<board[i]<<" wins\n";
```

```
   return true;

   }
 //Determines who won
    for(i=0;i<3;i++)
    if(board[i] == board[i+3] && board[i]==board[i+6])

    {
        cout<<board[i]<<" wins\n";

       return true;
   }

    //Fill in empty space
   if(board[0]==board[4]&&board[0]==board[8])
   {
       cout<<board[0]<<" wins\n";

       return true;
   }

if(board[6]==board[4]&&board[6]==board[2])
{
    cout<<board[6]<<" wins\n" ;

    return true;
}

//If no one wins, the game ends at a tie
return false;
}

void tictac()
{
    //Initialze variables
    //error tic;
    bool gameover=false;

    char board[9];

   int i;

   int numMoves = 0;

   char whoseTurn = 'X';
   int move;

   //Game starts here
   cout << "Player X starts.\n";

   //Counter
   for (i=0; i<9; i++)
{
   board[i] = '1' + i;
}

   while (numMoves < 9 && !gameover)
```

```
{

   //Draw the board
   showBoard(board);

   cout << "Enter move: "<< endl;

   cin >> move;

   if ((move <1) || (move>9))
   {

   cout<< "Invalid move, try again." << endl;
   }

   else
   {
   move--;

   if((board[move]=='X') || (board[move]=='O'))
   {
   cout << "That space is taken. " << endl;
   }

   else
   {
   board[move] = whoseTurn;

   if (whoseTurn == 'X')

   whoseTurn = 'O';
   else

   whoseTurn = 'X';

   numMoves++;

   gameover=checkForWinner(board);

   }
   }
   }

   if(!gameover)

   cout<<"Tie Game\n";

   showBoard(board);

   cout << endl << "Game over!" << endl;

    system("pause");

    }

    // Word Jumble
   // The classic word jumble game where the player can ask for a hint
```

```
void jumble()
{
    //enumerators
enum fields {WORD, HINT, NUM_FIELDS};

const int NUM_WORDS = 5;

//Possible quesions
//with the hints
const string WORDS[NUM_WORDS][NUM_FIELDS] =
{
{"wall", "Do you feel you're banging your head against something?"},
{"glasses", "These might help you see the answer."},
{"Progrmming", "What do you do in this class?"},
{"persistent", "Keep at it."},
{"jumble", "It's what the game is all about."}
};

srand(time(0));

//Used to randomize the questions
int choice = (rand() % NUM_WORDS);

// word to guess
string theWord = WORDS[choice][WORD];

 // hint for word
string theHint = WORDS[choice][HINT];

// jumbled version of word
string jumble = theWord;
int length = jumble.size();

//randomize word
for (int i=0; i<length; ++i)
{

    //Jubles the word
int index1 = (rand() % length);

int index2 = (rand() % length);

char temp = jumble[index1];

jumble[index1] = jumble[index2];

jumble[index2] = temp;
}

//Introduction
cout << "Welcome to Word Jumble!";
cout << endl;

cout << "Unscramble the letters to make a word.";
cout << endl;

cout << "Enter 'hint' for a hint.";
cout << endl;
```

```cpp
cout << "Enter 'quit' to quit the game.";
cout << endl;

cout << "The jumble is: " << jumble;


//User guess
string guess;

cout << "\n\nYour guess: ";

cin >> guess;

// Game contiunes
while ((guess != theWord) && (guess != "quit"))
{
if (guess == "hint")
cout << theHint;

else
cout << "Sorry, that's not it.";

cout <<"\n\nYour guess: ";
cin >> guess;
}

//Winner
if (guess == theWord)
cout << "That's it! You guessed it!";

cout << endl;

cout << "Thanks for playing.";

cout << endl;

system("PAUSE");

cout << endl;

// End of final function
}
```