**Project 2**

**<Guessing Password>**

CSC-5

Name: Bochi Lin

Date: 12/11/16

**Introduction**

Title: Password Guessing Game

Assume three players are locked three bombs in their bodies. The passwords of these bombs are the same. This game is to get points from guessing password. The bomb with the lowest point will explode. If a player guesses the right password, the bomb will explode as well. The only way to survive is to get as many points as you can. However, if the three players have three same scores or their total score is equal or equal to 15, they all survive. In round 1 and round 2, each player should read the file after he or she input guessing since there will be a hint for the player, and he or she should keep it as a secret.

**Summary**

Project size: 361 lines

The number of variables: 23

The number of methods: 30

**Problems during coding**

I want to do a brand new program which is different from the first project to challenge myself. Thus, it took me about two weeks to finish this project. Although I have experience doing project, I still had troubles while I was using new concepts to code.

First, I don’t want to make a game only for guessing the right password, but mainly to get points while players are guessing. I spent a few days to come up this idea.

Second, since players are guessing one by one in each round, it is impossible to output hints using “cout” because I want them to keep the hint as a secret. I tried using functions and delete array methods to do it but it didn’t make sense. At last, I used output file, which is from chapter 5 to show hints to players.

**System libraries**

|  |
| --- |
| #include <cstdlib> |
| #include<iostream> |
| #include<ctime> |
| #include<fstream> |
| #include<string> |
| #include<cmath> |

**Global Constants**

|  |  |  |
| --- | --- | --- |
| const int | ROW=3 | Row of 2D array |
| const int | COL=3 | Column of 2D array |

**Variables**

|  |  |  |
| --- | --- | --- |
| int | ranNumber | Random number for each digit |
| int | score=0 | Scores for each player are 0 |
| int | Password[SIZE1]  Password2[SIZE1] | Two identical password arrays |
| int | choice2 | Choice in round 2 |
| int | Number | The number players need to guess in round 2 |
| int | digit1=0  digit2=0  digit3=0  digit4=0 | In round 1:  First digit  Second digit  Third digit  Fourth digit |
| int | score1[SIZE] = {}  score2[SIZE] = {}  score3[SIZE] = {} | Score array for p1  Score array for p2  Score array for p3 |
| int | sumScore[ROW][COL] | 2D array  for the total score |
| const int | SIZE=3 | Size of nameArray |
|  | SIZE1=4 | Size of Password/Password2 |
| ofstream | outputfile | Output to file |
| char | answer | Answer A/B in round 2 |
| string | nameArray[SIZE] | Array of players’ names |

**Function Prototypes**

|  |
| --- |
| void Introduce( ); |
| void sortpass ( int [ ], int) |
| void prntScore ( int [ ] , int ) |
| void outputFile( int [ ] , int [ ] ) |
| bool rightPass( int , int , int , int , int [ ] ) |
| bool linearSrch (int [ ] , int , int , int , int , int ) |

**Concepts**

|  |  |  |  |
| --- | --- | --- | --- |
| **concept** | **type** | **code** | **location** |
| Cout object | cout | Cout<<”Please enter the … | 51 |
| #include | #include | #include<iostream> | 8 |
| variables | int,char | char answer | 44 |
| comment | //statement | //Declare variables | 28 |
| Cin object | Cin | Cin>>digit1 | 69 |
| Type casting | Static\_cast<int> | Static\_cast<int>(time(0)) | 57 |
| Making decision | Relational operator | ==,> | 218 |
|  | If statement | if(digit1==Password[1]) | 151 |
|  | If/else if | If(Number….) …  Else if(Number!=…) … | 267 |
|  | If/else | If(sum1>sum3) cout…  Else cout… | 263 |
|  | If/else if/else | If(Number==Password[0])  Else if (Number!=P……)  Else score=0 | 170 |
|  | Menu | Cout<< “do you want…”  Cout<< “1…;2….” | 134 |
|  | Switch statement | Switch(choice2) | 140 |
| Loops and files | For loop | For(int i=0;i<SIZE;i++) | 50 |
|  | Nested loops | For(…){  For(…..)  } | 282 |
|  | Using files | #include <fstream>  ofstream ouputfile | 11  41 |
| Functions | Function Prototypes | void sortpass(int [ ],int ); | 21 |
|  | Sending data into a function | sortpass(Password, SIZE1); | 127 |
|  | Passing data by value | linearSrch(Password,SIZE1,  digit1,…) | 77 |
|  | Returning boolean | bool linearSrch  (int a [ ], ...)  return true | 295  298 |
| Arrays | Array initialization | String nameArray[SIZE] | 31 |
|  | Arrays as function arguments | linearSrch(Password[ ]……) | 77 |
|  | Two-dimensional array | sumScore[ROW][COL] | 40 |
| Searching and sorting | Linear search | linearSrch(int a[], int n, int num1, int num2, int num3, int num4) | 295 |
|  | Selection sort | sortpass(int a[ ], int n) | 280 |

**Flowchart**

<https://www.gliffy.com/go/publish/11593167>

**Pseudo-code:**

System libraries

Global constants

Function prototypes

Declare variables

Call Introduce ( ) to introduce the whole game

Ask players to enter their names

Input names into name array

Set a random number

Get a four-digit random password and store it into two identical arrays

Ask each player to enter his/her guess

Input digits

If player guess the right password

Display game over, please start over

End if

If players’ digits contained in the password array

Output file, and tell players which number is right and which digit in the right password

Add score to players, and store into score array for players

If player’s score is 0

Output hint file , tell player the right first digit

End if

Display each score of each player

End if

Display all scores of players in round 1

Call function sortpass to sort the password from low to high

If the lowest number is not 0, ask players to guess the lowest number

If the lowest number is 0, ask players to guess the number in second digit

Display a menu for players to fill in a blank or do multiple choice

If player choose to fill in a blank

If right, add 5 points

If wrong, add 1 point

Output points to file

End if

If player choose to do a multiple choice

If right, add 3 points

If wrong, add 0 points

Output points to file

End if

Display all scores in round 1 and round 2

Now the password is back into the original order

Display the third digit of the password

Ask each player to enter their guessing, and round 3 begins

If player guess the right password

Display Game over

End if

else

If the second digit is in range +- 3

Add 1 point

If second digit is right

Add 3 points

End if

If the fourth digit is in range +-3

Add 2 points

End if

End else

Add scores to each array

Display all scores in round 1, round 2, and round 3

Calculate the final score

Display the result of the game

**Code**

/\*

\* File: main.cpp

\* Author: Bochi Lin

\* Created on December 4, 2016, 12:12 PM

\* Purpose: Project 2 - CSC5 - 48102

\*/

#include <iostream>//I/O

#include <string>

#include <cstdlib> //Random

#include <fstream> //I/O file

#include <cmath> //pow

#include <ctime>

using namespace std;

//Global Constants

const int ROW=3;

const int COL=3;//Sizes of the two dimensional array

//Function Prototypes

void Introduce();

void sortpass(int [], int);

bool linearSrch(int [], int, int, int, int, int);

void prntScore(int [], int);

void outputFile(int [], int []);

bool rightPass(int,int,int,int,int []);

int main(int argc, char\*\* argv) {

//Declare variables

const int SIZE = 3;

const int SIZE1 = 4;

string nameArray[SIZE];

int ranNumber;

int digit1 = 0, digit2 = 0, digit3 = 0, digit4 = 0;

int Password[SIZE1] = {};

int Password2[SIZE1]={};

int score = 0;

int score1[SIZE] = {};

int score2[SIZE] = {};

int score3[SIZE] = {};

int sumScore[ROW][COL];

ofstream outputfile;

int choice2;

int Number;

char answer;

//Introduce the game

Introduce();

//Input information

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

cout << "Please enter the name of ";

cout << "player " << i + 1 << ": ";

getline(cin, nameArray[i]);

}

//Set a random number

srand(static\_cast<int> (time(0)));

//Get a four-digit random password here

for (int i = 0; i < SIZE1; i++) {

ranNumber = rand() % 9;

Password[i] = ranNumber;

Password2[i]=Password[i]; //It is the same array as Password

}

cout << endl;

//Input information

for (int j = 0; j < SIZE; j++) {

cout<<"Please use space to separate each digit."<<endl;

cout << "Player " << nameArray[j] << ", please enter your guess: " << endl;

cin >> digit1 >> digit2 >> digit3>>digit4;//Enter digits one by one

if(rightPass(digit1,digit2,digit3,digit4,Password2)){

cout<<"You guess the right password. Please start over."<<endl;

cout<<"Game over.";//If the player guess the right password, run finish

return 0;

}

//Round 1 begins here

if (linearSrch(Password, SIZE1, digit1, digit2, digit3, digit4)){//Call a bool function here

for (int i = 0; i < SIZE1; i++) {

if (digit1 == Password[i]) {

outputfile.open("newfile.h"); //Open file

outputfile << digit1 << " is right, and it is at digit " << i + 1 << endl;

outputfile.close();//close file

}

if (digit2 == Password[i]) {

outputfile.open("newfile.h");

outputfile << digit2 << " is right, and it is at digit " << i + 1 << endl;

outputfile.close();

}

if (digit3 == Password[i]) {

outputfile.open("newfile.h");

outputfile << digit3 << " is right, and it is at digit " << i + 1 << endl;

outputfile.close();

}

if (digit4 == Password[i]) {

outputfile.open("newfile.h");

outputfile << digit4 << " is right, and it is at digit " << i + 1 << endl;

outputfile.close();

}

}

score += 1; //If it is right, add one point

} //Else, the score is still 0

cout << "You get " << score << " point in this round." << endl;//Display the score

if (j==0) {

score1[0] = score;//Add score for the first player

if (score==0) outputFile(score1, Password);//If the score is 0, call function

}

if (j==1) {

score2[0] = score;//Add score for the second player

if (score==0) outputFile(score2, Password);//If the score is 0, call function

}

if (j==2) {

score3[0] = score;//Add score for the second player

if (score==0) outputFile(score3, Password);//if the score is 0, call function

}

score = 0; //Initialize score to 0 again

}

//Here are the grades for the first round

cout<<"\*\*\*\*\* GRADES \*\*\*\*\*"<<endl;

cout << "round 1" << endl;

prntScore(score1, SIZE);

prntScore(score2, SIZE);

prntScore(score3, SIZE);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

//Round 2 begins

sortpass(Password, SIZE1); //Call a function for using selection sort to sort password array

cout << endl;

cout << "Now. The password is in a new order from low to high." << endl;

if(Password[0]!=0) cout<<"Please guess the lowest number."<<endl;

else cout<<"Please guess the number in the second digit."<<endl; //If the lowest is 0, guess the second digit

for (int j = 0; j < SIZE; j++) {

//Display menu here

cout<<"\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"Player "<<nameArray[j]<<endl;

cout<<"Do you want to guess the number on your own or choose an answer?"<<endl;

cout<<"1.On your own(5 points if correct/1 point if wrong); 2.choose answer(2 points if correct)"<<endl;

cout<<"Please enter your choice: ";

cin>>choice2;//player enter their choice

switch(choice2){

case 1:{

cout<<"You chose to guess this number on your own."<<endl;

cin>>digit1;//Enter the number

if(Password[0]!=0){//if the first digit is not 0

if(digit1==Password[0]){

score+=5; //If the number is right, add 5 points

}

else score+=1;//if the number is not right, add 1 point

}

else{//If the first digit is 0

if(digit1==Password[1]){//The number player entered is compare to the second digit

score+=5;//If the number is right, add 5 points

}

else score+=1;//if the number is not right, add 1 point

}

break;

}

case 2:{

cout<<"You chose to guess this number on an multiple choice."<<endl;

Number = rand() % 9;//Get a random number here

cout<<"Is "<<Number<<" the number?"<<endl;

cout<<"A.Yes; B.No"<<endl;

cin>>answer;//Player enter his/her choice

if(Password[0]!=0){

if(Number==Password[0]&&answer=='A') score+=3; //If the choice is correct

else if(Number!=Password[0]&&answer=='B') score+=3; //then a 3 points

else score=0;//If the choice is not correct,score is 0

}

if(Password[0]==0){//The same idea as below, but is compare to the second digit

if(Number==Password[1]&&answer=='A') score+=3;

else if(Number!=Password[1]&&answer=='B') score+=3;

else score=0;

}

break;

}

}

outputfile.open("newfile.h");//Open file

outputfile<<"You get "<<score<<" points in this round."<<endl;//output file

outputfile.close();//close the file

//Add scores to each arrays of each player

if (j==0) score1[1] = score;

if (j==1) score2[1] = score;

if (j==2) score3[1] = score;

score=0; //Initialize score to 0 again

}

//Display the grades here

cout<<endl;

cout<<"\*\*\*\*\*\*\*\*\* GRADES \*\*\*\*\*\*\*\*\*"<<endl;

cout << "round 1 round 2" << endl;

prntScore(score1, SIZE);

prntScore(score2, SIZE);

prntScore(score3, SIZE);//Here are the scores for round1 and round2

cout<<endl;

//Round 3 begins here

cout<<"\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"Now. The password is back into the original order."<<endl;//Use array Password2

cout<<"Hint: The third digit of the password is "<<Password2[2]<<endl;//Show hints to players

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

cout<<"Player "<<nameArray[i]<<", please enter your guess:"<<endl;

cin>>digit1>>digit2>>digit3>>digit4;

if(rightPass(digit1,digit2,digit3,digit4,Password2)){

cout<<"Game over."; //Players can not guess the right password immediately

return 0;

}

else{

if(pow(digit2-Password2[1],2)<=9){//If the range is equal to or less than 3

score+=1; //Score add 1

if(digit2==Password2[1]) //If the second digit is correct

score+=3; //Score add 3

}

if(pow(digit4-Password2[3],2)<=9){//If the range is equal to or less than 3

score+=2; //Score add 2

}

}

//Add scores to each array

if (i==0) score1[2] = score;

if (i==1) score2[2] = score;

if (i==2) score3[2] = score;

score=0;//Initialize score to 0

}

//Display all three grades

cout<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\* GRADES \*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" round 1 round 2 round3" << endl;

prntScore(score1, SIZE);

prntScore(score2, SIZE);

prntScore(score3, SIZE);//Here are the scores for round1 and round2

cout<<endl;

//Calculate the final score

int total=0,sum1=0,sum2=0,sum3=0;

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

for(int k=0;k<COL;k++){

if(i==0){

sumScore[i][k]=score1[k];

sum1+=sumScore[i][k]; //final score for player1

}

if(i==1){

sumScore[i][k]=score2[k];

sum2+=sumScore[i][k]; //final score for player2

}

if(i==2){

sumScore[i][k]=score3[k];

sum3+=sumScore[i][k]; //final score for player3

}

total+=sumScore[i][k];

}

}

//Display the result here

if((sum1==sum2&&sum2==sum3)||total>=10) {

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*\* You all saved, the bomb has been disposed \*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

else{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

if(sum3<sum1&&sum3<sum2) cout<<"Player "<<nameArray[2]<<" dies with the lowest score."<<endl;

if(sum2<sum1&&sum2<sum3) cout<<"Player "<<nameArray[1]<<" dies with the lowest score."<<endl;

if(sum1<sum2&&sum1<sum3) cout<<"Player "<<nameArray[0]<<" dies with the lowest score."<<endl;

if(sum1==sum2){

if(sum2>sum3) cout<<"Player "<<nameArray[2]<<" dies with the lowest score."<<endl;

else cout<<"Player "<<nameArray[2]<<" survives with the highest score."<<endl;

}

if(sum1==sum3){

if(sum3>sum2) cout<<"Player "<<nameArray[1]<<" dies with the lowest score."<<endl;

else cout<<"Player "<<nameArray[1]<<" survives with the highest score."<<endl;

}

if(sum3==sum2){

if(sum2>sum1) cout<<"Player "<<nameArray[0]<<" dies with the lowest score."<<endl;

else cout<<"Player "<<nameArray[0]<<" survives with the highest score."<<endl;

}

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

//Program ends

return 0;

}

//Selection sort to sort password digits from low to high

void sortpass(int a[], int n) {

int startScan, minIndex, minValue;

for (startScan = 0; startScan < n; startScan++) {

minIndex = startScan;

minValue = a[startScan];

for (int index = startScan + 1; index < n; index++) {

if (a[index] < minValue) {

minValue = a[index];

minIndex = index;

}

}

a[minIndex] = a[startScan];

a[startScan] = minValue;

}

}

//Linear search to find the right digit

bool linearSrch(int a[], int n, int num1, int num2, int num3, int num4) {

fstream outputfile;

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

if (num1 == a[i]) return true;

if (num2 == a[i]) return true;

if (num3 == a[i]) return true;

if (num4 == a[i]) return true;

}

return false;

}

//Print out the scores that players get

void prntScore(int a[], int n) {

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

cout <<" " <<a[i] <<" ";

}

cout << endl;

}

//Output hints to file called "newfile.h" for each player

void outputFile(int a[], int b[]) {

ofstream outputfile;

if (a[0] == 0) {

outputfile.open("newfile.h");

outputfile << "Here is a hint for you: " << endl;

outputfile << "The first digit of the password is " << b[0] << endl;

outputfile.close();

}

if (a[0] == 0) {

outputfile.open("newfile.h");

outputfile << "Here is a hint for you: " << endl;

outputfile << "The first digit of the password is " << b[0] << endl;

outputfile.close();

}

if (a[0] == 0) {

outputfile.open("newfile.h");

outputfile << "Here is a hint for you: " << endl;

outputfile << "The first digit of the password is " << b[0] << endl;

outputfile.close();

}

}

//Determine whether the player enter the right password or not

bool rightPass(int a,int b,int c,int d,int e[]){

if(a==e[0]){

if(b==e[1]){

if(c==e[2]){

if(d==e[3]) return true;

}

}

}

return false;

}

//Introduce the whole game here

void Introduce(){

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\*\* Welcome to the Passwords Guessing Game \*\*\n";

cout<<"\*\* This game is to find password for the bomb \*\*\n";

cout<<"\*\* Assume you three players have bombs lock your bodies \*\*\n";

cout<<"\*\* The passwords are the same \*\*\n";

cout<<"\*\* You should try your best to get points in three rounds \*\*\n";

cout<<"\*\* If you get the right password, don't think you are lucky \*\*\n";

cout<<"\*\* Because the bomb will explode \*\*\n";

cout<<"\*\* What you have to do is getting points \*\*\n";

cout<<"\*\* The bomb with the lowest points will explode \*\*\n";

cout<<"\*\* Make sure to check file after you input in round 1 and 2 \*\*\n";

cout<<"\*\*If the total point is >=15 or scores are same you all survived\*\*\n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

}