**BGS NATIONAL PUBLIC SCHOOL**

**COMPUTER SCIENCE INVESTIGATORY PROJECT**

**2018-19**



**ORION- A SPACE ADVENTURE**

OR ON

****

NAME : ANANYA R BURLI

CLASS : XII A

EXAM ROLL NO :

**CERTIFICATE**

This is to certify that Ananya R Burli of class XII A, exam number has successfully completed the Computer Science Investigatory Project ‘ORION-A SPACE ADVENTURE’ under my guidance during the year 2018-19 in the partial fulfilment of the Computer Science practical examination conducted by CBSE.

**SIGNATURE OF THE PRINCIPAL**

**SIGNATURE OF THE SIGNATURE OF THE**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

‘Thank you’ is the best prayer that one can say.

Thus, I wish to take this opportunity to express my heartfelt gratitude to each and every one who has played a role in the completion of this project. Firstly, I would like to thank the Academic Director of BGS NPS, Sri SA Nair and our beloved Principal Ms. Sreekala G Kumar for their blessings and constant support.

Secondly, I would like to thank my Computer Science teacher Ms. Babitha Mathews for her ceaseless efforts and the sea of knowledge she had to offer. I am thankful to BGS National Public School for providing their laboratory facilities to complete the project.

A big thank you to my friends for helping me with the little things and always standing by my side. Most importantly, I wish to thank my parents for being the pillars of strength and motivating me to do this project with their ever-encouraging words.

THANK YOU.

**INDEX**

|  |  |  |
| --- | --- | --- |
| SL  NO. | TOPIC | PAGE  NO. |
| 1. | INTRODUCTION | 5 |
| 2. | PROJECT SELECTION | 6 |
| 3. | WORKING ENVIRONMENT | 7 |
| 4. | HEADER FILES | 8 |
| 5. | DATA DICTIONARY | 9 |
| 6. | SOURCE CODE | 12 |
| 7. | PROJECT LOG | 44 |
| 8. | SAMPLE OUTPUT | 46 |
| 9. | BIBLIOGRAPHY | 49 |

**INTRODUCTION**

ORION- A SPACE ADVENTURE, is a space themed game conceived in C++. It embraces the player as an astronaut who must save the galaxy from extra-terrestrial forces that are threatening to destroy his/her home. In order to do so, the user must shoot the bullets fired at his/her battleship and answer space trivia at different stages of the game. The game enables the user to create an account by choosing a unique player number and password for authentication and identification. The player can also login via the same. The game allows modification of the user’s player number if needed. The user can also view high scores.

Overall, Orion is a game that appeals to its players to indulge in a thrilling yet informative experience.

**PROJECT SELECTION**

I have selected this space-themed game for my investigatory project as it is very compelling, both to create as well as play. Orion is a game that not only allows the player to have fun while playing, but also taps their minds and encourages them to think and develop a hunger to learn more and more about the multifaceted universe that we a part of. Designing this game was a thoroughly enjoyable task. C++ games are intriguing, but at the same time they are difficult to design. This makes the choice of a game, for one’s investigatory project rather unconventional and unexpected. Hence, the choice for my Computer Science Investigatory Project.

**WORKING ENVIRONMENT**

Optimum Requirements –

* + - * Operating System – Windows XP/7/8/8.1/10
      * Processor – Must be clocked over 1.5 GHz
      * Graphics Driver – Intel Integrated
      * Graphics or equivalent or greater
      * RAM – 1 GB or more.
      * Hard Disk – 1 GB or more.
      * A C++ compiler is necessary.

**HEADER FILES**

|  |  |
| --- | --- |
| #include<iostream.h> | Defines input and output stream (for cout and cin) |
| #include<conio.h> | Provides console input and output |
| #include<fstream.h> | Defines new input/output stream between the program and file |
| #include<stdio.h> | For gets(),puts() |
| #include<string.h> | For strcpy(),strcmp(),strlen() |
| #include<dos.h> | For delay() |
| #include<stdlib.h> | For exit() |
| #include<graphics.h> | For a graphical user interface |
| #include<ctype.h> | For isdigit() |

**DATA** **DICTIONARY**

1. STRUCTURES/CLASSES USED :

|  |  |  |
| --- | --- | --- |
| **NAME** | **TYPE** | **PURPOSE** |
| Option | Structure | To display options for the space trivia |
| login | Class | To allow the user to login, if they have an existing account, or create a new account |
| HighScore | Class | To arrange the scores in descending order and display the highest scores |
| Trivia | Class | To display appropriate trivia questions and check if the user’s answer is correct, in order to move forward in the game |

1. MEMBER FUNCTIONS USED :

|  |  |  |
| --- | --- | --- |
| **CLASS** | **MEMBER FUNCTION** | **PURPOSE** |
| Login | Uniquepno() | To check if the ID number entered by a new user is unique |
| Uniquepswd() | To check if the password entered by a new user is unique |
| Login\_get() | Accepts the details of the user |
| Change\_pno() | To enter a new ID number |
| Retpno() | Returns the player number of the corresponding user |
| Retname() | Returns the username of the corresponding user |
| Retpswd() | Returns the password of the corresponding user |
| Retname() | Returns the username of the user |
| Trivia | Trivia\_Get() | To accept the questions from the programmer, to be written onto the file |
| Trivia\_put() | To display the appropriate questions |
| Retans() | To return the correct option number for the appropriate question |
| Retqno() | To return the question number in order to display them in the right order |
| HighScore | Change1() | To copy the new ID number onto the ‘highscore’ file |
| Assign() | To assign the data members for the corresponding object |
| Put\_sc() | Displays the score and ID number |
| Retscr() | To return the score of the corresponding user |
| Retpno1() | Returns the ID number of the current user |
| Change\_pno1() | To change the ID number on the leaderboard |

1. DATA MEMBERS :

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **DATA MEMBER** | **DATA TYPE** | **PURPOSE** |
| Login | Pno | Int | ID number |
| Name | Char[] | Username |
| Pswd | Char[] | Password |
| Trivia | Q | Char[] | Question |
| Qno | Int | Question number |
| Ans | Int | Answer (correct option number) |
| Highscore | Score | Int | Score of the user |
| Pno1 | Int | ID number of the user |

1. GLOBAL FUNCTIONS :

|  |  |
| --- | --- |
| **NAME** | **PURPOSE** |
| Ink() | To display information in the graphical window |
| Ins() | To display information in the graphical window |
| Rules() | Display the rules of the game |
| Start\_loop() | Display options to sign up/login in the graphical window |
| Start() | Display the options to sign up or login |
| Modify1() | To change the ID number in ‘highscore.txt’ |
| Modify() | To change the ID number in ‘login.txt’ |
| Log\_in() | Allows the user to log in to their existing account |
| signup() | Allows the user to sign up to create a new account |
| Globput\_sc() | Displays the leader board |
| Desc() | Writes scores of players onto ‘highscore.txt’ in descending order |
| Del() | Deletes score from ‘highscore.txt’ |
| Compare() | To view the leader board |
| High() | Displays options to save one’s score, display it, etc. |
| Welcome() | Displays the welcome page |
| Bar() | Displays the user’s battleship for the game |
| Check() | Checks if the player’s answer to the trivia questions are correct |
| Trivia\_dis() | Displays the trivia questions |
| Options\_assign() | Assigns options to the trivia questions |
| Table() | Displays options to exit, play, etc. during the game |
| Options\_dis() | Displays options to the trivia questions |
| Up() | Allows player to shoot bullets |
| Status() | Shows the present status of the player |
| Down() | Displays the obstacles fired by the enemy |
| Main() | The main function of the program that allows its execution |

**SOURCE CODE**

#include<stdio.h>

#include<dos.h>

#include<ctype.h>

#include<string.h>

#include<conio.h>

#include<stdlib.h>

#include<iostream.h>

#include<fstream.h>

#include<graphics.h>

void ink(char name[20],int pno) //IN DO-WHILE LOOP

{

char a2[20];

itoa(pno,a2,10);

cleardevice();

outtextxy(10,10,"GAMER, ENTER YOUR USERNAME : ");

outtextxy(250,10,name);

outtextxy(10,35,"GAMER, ENTER YOUR ID NUMBER: ");

outtextxy(250,35,a2);

outtextxy(10,50, "Create an alphanumeric password of your choice : ");

}

void ins() //IN DO-WHILE LOOP

{

cleardevice();

rectangle(0,0,getmaxx()-1,getmaxy()-1);

outtextxy(0,10,"ENTER 1 TO SAVE YOUR SCORE, 2 TO DISPLAY, 3 TO DELETE, ANY OTHER INTEGER TO EXIT : ");

}

//GLOBAL VARIABLES

int glob=0, var=0,HELLO=0,flag=0,globb=0;

xp[20],yp[20];

char pass[50],pass1[50];

const int sz=5;

int xx[sz],yy[sz]; void desc(); void log\_in(); void signup();

int ctr=1;

static scr=0;

int fr=0;

struct Option //STRUCTURE OF OPTIONS

{

char o1[100],o2[100],o3[100];

}ob[11];

void rules() //RULES OF THE GAME

{

ofstream fout("rules.txt",ios::out);

fout<<" YEAR 2050.";

fout<<"FELLOW ASTRONAUT,WELCOME TO THE ORION! AN EXTRA-TERRESTRIAL FORCE IS THREATENING TO DESTROY OUR GALAXY. THE BRAVEST OF THE BRAVE MUST PROTECT US. AFTER MUCH DELIBERATION,YOU HAVE BEEN CHOSEN FOR THE MISSION. HERE ARE YOUR INSTRUCTIONS: \n";

fout<<"1)IF YOU ARE A VETERAN, LOG IN. IF THIS IS YOUR DEBUT FLIGHT, CREATE A PROFILE (SIGN UP).";

fout<<" YOUR MISSION IS TO TAKE DOWN AS MANY ENEMY SHUTTLES AS YOU CAN. BEWARE: IF A BULLET HITS 'THE ORION', YOU DIE.";

fout<<"3)USE THE RIGHT AND LEFT ARROW KEYS TO NAVIGATE. USE THE SPACE BAR TO SHOOT.";

fout<<"4)YOU MAY ENCOUNTER SPACE EMERGENCIES. USE YOUR INTELLECTUAL PROWESS AND ANSWER QUESTIONS IN ORDER TO PROCEED.";

fout<<"\n\nPRO TIP: THE ENEMY ARE GOOD AT HIDING THEMSELVES. SHOOT IN THE DIRECTION OF BULLETS FROM THE ENEMY SHUTTLE IN ORDER TO TRACE AND DESTROY THEM. ";

fout.close();

ifstream fin("rules.txt",ios::in);

cout<<"\n\n";

for(int i=0;i<80;i++)

cout<<"~";

char ch1;

while(!fin.eof())

{

fin.get(ch1);

cout<<ch1;

delay(80);

if(ch1=='.'||ch1=='?')

{

cout<<"\n\n";

delay(1000);} }

fin.close();

cout<<endl<<endl; }

void start\_loop() //IN DO-WHILE LOOP

{

rectangle(0,0,getmaxx()-1,getmaxy()-1);

outtextxy(10,getmaxy()/2,"Greetings, astronaut! Sign up to create a new profile. Log in if you have an ");

outtextxy(10,getmaxy()/2+15,"existing ID");

outtextxy(10,getmaxy()/2+30,"1. LOG IN ");

outtextxy(10,getmaxy()/2+45,"2. SIGN UP ");

outtextxy(10,getmaxy()/2+60,"3. EXIT ");

}

void start() //SIGN UP OR LOG IN

{

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TC\\BGI");

setcolor(LIGHTCYAN);

rectangle(0,0,getmaxx()-1,getmaxy()-1);

outtextxy(10,getmaxy()/2,"Greetings, astronaut! Sign up to create a new profile. Log in if you have an ");

outtextxy(10,getmaxy()/2+15,"existing ID");

outtextxy(10,getmaxy()/2+30,"1. LOG IN ");

outtextxy(10,getmaxy()/2+45,"2. SIGN UP ");

outtextxy(10,getmaxy()/2+60,"3. EXIT ");

int a;

label:

char c;

int pos=0,e=0;

char a1[10];

strcpy(a1,"");

do{

c=getch();

switch(c)

{

case 8:

if(pos)

{

pos--;

a1[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a1[pos]=c;

pos++;

a1[pos]=0; } }

cleardevice();

start\_loop();

outtextxy(10,getmaxy()/2+75,a1);

}

while(!e);

a=atoi(a1);

cleardevice();

switch(a)

{

case 1: log\_in(); break;

case 2: signup(); break;

case 3: exit(0);

default: cout<<"Error. Please choose between options 1, 2 and 3 only. "<<endl; goto label;

}

closegraph();

}

class login //CLASS 1

{ char password[5];

int pno; char name[20];

public:

int uniquepno(int pn) //UNIQUE PNO

{

int count=0;

login obj;

ifstream fin("login.txt",ios::in|ios::binary);

while(fin.read((char\*)&obj,sizeof(obj)))

{

if(obj.retpno()==pn)

count++;

}

return count;

}

int uniquepswd(char ps[]) //UNIQUE PSWD

{

int count=0;

login obj;

ifstream fin("login.txt",ios::in|ios::binary);

while(fin.read((char\*)&obj,sizeof(obj)))

{

if(strcmp(obj.retpswd(),ps)==0)

count++;

}

return count;

}

void login\_get() //GET DETAILS TO SIGN UP

{

cleardevice();

int pos=0,e=0;

char a[10];

char name1[20];

outtextxy(10,10,"GAMER, ENTER YOUR USERNAME : ");

strcpy(a,"");

do{

c=getch();

switch(c)

{

case 8:

if(pos)

{

pos--;

a[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a[pos]=c;

pos++;

a[pos]=0;

}

}

cleardevice();

outtextxy(10,10,"GAMER, ENTER YOUR USERNAME : ");

outtextxy(250,10,a);

}

while(!e);

strcpy(name,a);

strcpy(a,"");

outtextxy(10,35,"GAMER, ENTER YOUR ID NUMBER (2-digit): ");

pos=0,e=0;

do{

f:

c=getch();

if((!isdigit(c))&&(c!=13)&&(c!=8))

goto f;

switch(c)

{

case 8:

if(pos)

{

pos--;

a[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a[pos]=c;

pos++;

a[pos]=0;

}

}

lk:

cleardevice();

outtextxy(10,10,"GAMER, ENTER YOUR USERNAME : ");

outtextxy(250,10,name);

outtextxy(10,35,"GAMER, ENTER YOUR ID NUMBER (2-digit): ");

outtextxy(315,35,a);

}while(!e);

pno=atoi(a);

if(pno/100)

{

pos=e=0;

strcpy(a,"");

goto lk;

}

loop1:

flag=uniquepno(pno);

if(flag!=0)

{

cleardevice();

outtextxy(1,10,"SORRY!! YOUR ID NUMBER HAS ALREADY BEEN CHOSEN, GO FOR ANOTHER NUMBER");

strcpy(a,"");

outtextxy(10,35,"GAMER, ENTER YOUR ID NUMBER (2-digit): ");

pos=0,e=0;

do

{

gh:

c=getch();

if((!(isdigit(c)))&&(c!=13)&&(c!=8))

goto gh;

switch(c)

{

case 8:

if(pos)

{

pos--;

a[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a[pos]=c;

pos++;

a[pos]=0;

}

}

fgh:

cleardevice();

outtextxy(10,10,"GAMER, ENTER YOUR USERNAME : ");

outtextxy(250,10,name);

outtextxy(10,35,"GAMER, ENTER YOUR ID NUMBER (2-digit): ");

outtextxy(315,35,a);

}

while(!e);

pno=atoi(a);

if(pno/100)

{

e=pos=0;

strcpy(a,"");

goto fgh;

}

flag=0;

goto loop1; }

globb=pno;

loopyloop:

strcpy(name1,name);

outtextxy(10,50, "Create an alphanumeric password of your choice : ");

pos=0,e=0;

strcpy(a,"");

strcpy(password,"");

do{

c=getch();

switch(c)

{

case 8:

if(pos)

{

pos--;

password[pos]=0;

a[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

password[pos]=c;

a[pos]='\*';

pos++;

password[pos]=0;

a[pos]=0;

}

}

cleardevice();

ink(name1,pno);

outtextxy(10,50, "Create an alphanumeric password of your choice : ");

outtextxy(10,65,a);

}

while(!e);

cout<<endl;

delay(1000);

getch();

int chck=uniquepswd(password);

if(chck!=0)

{

cleardevice();

outtextxy(0,20,"Password is not unique. Please create a new password. "); goto loopyloop;

getch();

}

}

void change\_pno() //GETS NEW PNO

{

char a[20],c;

strcpy(a,"");

int pos=0,e=0;

do{

v:

c=getch();

if((!(isdigit(c)))&&(c!=13)&&(c!=8))

goto v;

switch(c)

{

case 8:

if(pos)

{

pos--;

a[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a[pos]=c;

pos++;

a[pos]=0;

}

}

fghi:

cleardevice();

outtextxy(10,10,"Enter your new ID Number (2-digit) : ");

outtextxy(350,10,a);

}

while(!e);

pno=atoi(a);

if(pno/100)

{e=pos=0;

strcpy(a,"");

goto fghi;

}

}

int retpno() //RETURN PNO

{return pno; }

char\* retpswd() //RETURN PSWD

{return password;}

char\* retname() //RETURN NAME

{return name;} };

class HighScore //CLASS 2

{

int score; int pno1;

public:

void change1(int j) //NEW PNO1

{ pno1=j;}

void assign() //SCORE IN H.S. FILE

{

score = scr;

pno1=globb;

}

void put\_sc() //DISPLAY SCORES

{cout<<"\t\t "<<pno1<<"\t\t \t \t"<<score<<endl;}

int retscr() //RETURN SCORE

{return score;}

int retpno1() //RETURN PNO1

{return pno1;}

void change\_pno1(int y)

{pno1=y;}

};

void modify1( int y, int b) //CHANGE IN H.S.

{

HighScore objs;

fstream f("highscore.txt",ios::in|ios::out|ios::binary);

while(f.read((char\*)&objs,sizeof(objs)))

{

if(objs.retpno1()==b)

{

int g = f.tellg();

f.seekp(g-sizeof(objs),ios::beg);

objs.change\_pno1(y);

f.write((char\*)&objs,sizeof(objs));

}}

f.close(); }

void modify(char \* psw) // TO MODIFY THE ID NUMBER IF THE USER KNOWS THE PASSWORD BUT FORGOT THE ID NUMBER

{

login obj; int ctr=0,flag=0;

fstream fin("login.txt",ios::in|ios::out|ios::binary);

while(fin.read((char\*)&obj,sizeof(obj)))

{

if(strcmp(psw,obj.retpswd())==0)

{ int b=obj.retpno();

int p=fin.tellg();

fin.seekp(p-sizeof(obj),ios::beg);

outtextxy(10,10,"Enter your new ID Number (2-digit) : ");

obj.change\_pno();

int y=obj.retpno();

::globb=y;

modify1(y,b);

fin.write((char\*)&obj,sizeof(obj));

cleardevice();

outtextxy(10,10,"Modification successful. Please wait..."); delay(1000);

outtextxy(10,35,"Press any key to continue: "); getch();

break;

}}

fin.close(); }

void log\_in() //LOG IN

{ int cou1=0,cou2=0;

login obj;

char username[20],password[5]; int playerno,flg=0;

label2:

ifstream fin("login.txt",ios::in|ios::binary);

outtextxy(getmaxx()/2-100,getmaxy()/2,"ID NUMBER : ");

char c;

int pos=0,e=0;

char a1[10],a2[20],a3[20];

strcpy(a1,"");

do{

c=getch();

switch(c)

{

case 8:

if(pos)

{

pos--;

a1[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a1[pos]=c;

pos++;

a1[pos]=0;

}

}

cleardevice();

outtextxy(getmaxx()/2-100,getmaxy()/2,"ID NUMBER : ");

rectangle(getmaxx()/2-10,getmaxy()/2-10,getmaxx()/2+130,getmaxy()/2+10);

outtextxy(getmaxx()/2,getmaxy()/2,a1);

}

while(!e);

playerno=atoi(a1);

globb=playerno;

outtextxy(getmaxx()/2-100,getmaxy()/2+30,"PASSWORD: ");

strcpy(a1,"");

strcpy(a3,"");

e=0,pos=0;

do{

c=getch();

switch(c)

{

case 8:

if(pos)

{

pos--;

a3[pos]=0;

a1[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a1[pos]=c;

a3[pos]='\*';

pos++;

a1[pos]=0;

a3[pos]=0;

}

}

cleardevice();

outtextxy(getmaxx()/2-100,getmaxy()/2,"ID NUMBER : ");

outtextxy(getmaxx()/2-100,getmaxy()/2+30,"PASSWORD: ");

rectangle(getmaxx()/2-10,getmaxy()/2-10,getmaxx()/2+130,getmaxy()/2+10);

itoa(playerno,a2,10);

outtextxy(getmaxx()/2,getmaxy()/2,a2);

rectangle(getmaxx()/2-10,getmaxy()/2+20,getmaxx()/2+130,getmaxy()/2+40);

outtextxy(getmaxx()/2,getmaxy()/2+30,a3);

}

while(!e);

strcpy(password,a1);

while(fin.read((char\*)&obj,sizeof(obj)))

{

if(strcmp(password,obj.retpswd())==0)

{ cou1++;

if(playerno==obj.retpno())

{ cou2++;

getch();

cleardevice(); delay(100);

settextjustify(CENTER\_TEXT,CENTER\_TEXT);

outtextxy(getmaxx()/2,getmaxy()/2,"LOG IN SUCCESSFUL. PLEASE WAIT...");

delay(1000);

settextjustify(LEFT\_TEXT,TOP\_TEXT);

}}}

if(cou1!=0&&cou2==0)

{ cleardevice();

outtextxy(10,10,"Hmm. You seem to be an imposter.");

outtextxy(10,35," The details you have entered do not match the ones on your profile. ");

outtextxy(10,50,"Would you like to sign up as a new user instead?");

outtextxy(10,65,"1. Sign up");

outtextxy(10,80,"2. Forgot ID Number ");

int n;

repeat1:

strcpy(a1,"");

pos=e=0;

do{

c=getch();

switch(c)

{

case 8:

if(pos)

{

pos--;

a1[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a1[pos]=c;

pos++;

a1[pos]=0;

}

}

cleardevice();

outtextxy(10,10,"Hmm. You seem to be an imposter.");

outtextxy(10,35," The details you have entered do not match the ones on your profile. ");

outtextxy(10,50,"Would you like to sign up as a new user instead?");

outtextxy(10,65,"1. Sign up");

outtextxy(10,80,"2. Forgot ID Number ");

outtextxy(10,95,a1);

}

while(!e);

n=atoi(a1);

cleardevice();

switch(n)

{

case 1: signup(); break;

case 2: modify(password); break;

default: outtextxy(10,10,"Error. Please choose between options 1 and 2 only."); goto repeat1;

}}

if(cou1==0)

{

cleardevice();

outtextxy(10,10,"Sorry. Profile does not exist. ");

outtextxy(10,35," Please reenter details (press 1) or create a new profile (press 2): ");

int no;

repeat2:

strcpy(a1,"");

pos=e=0;

do{

c=getch();

switch(c)

{

case 8:

if(pos)

{

pos--;

a1[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a1[pos]=c;

pos++;

a1[pos]=0;

}}

cleardevice();

outtextxy(10,10,"Sorry. Profile does not exist. ");

outtextxy(10,35," Please reenter details (press 1) or create a new profile (press 2): ");

outtextxy(10,50,a1);

}

while(!e);

no=atoi(a1);

fin.close();

switch(no)

{ case 1: goto label2;

case 2: signup(); flg++; break;

default: cleardevice();

outtextxy(10,10,"Error. Please choose between options 1 and 2 only. "); goto repeat2;

}}}

void signup() //WRITE ONTO LOGIN FILE

{

login obj;

ofstream fout("login.txt",ios::app|ios::binary);

obj.login\_get();

fout.write((char\*)&obj,sizeof(obj));

fout.close();

}

class trivia //CLASS 3

{

char q[100];

int ans,qno;

public:

void trivia\_put() //DISPLAY Q

{

puts(q);

cout<<" ";

}

int retans() //RETURN ANS

{return ans;}

int retqno() //RETURN QNO

{return qno;}};

void globput\_sc(char \* passw) //LEADERBOARD

{

HighScore obj1; login obj; char username[50]; int luckyno,counter=0;

obj1.assign();

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

ifstream fin("login.txt",ios::in|ios::binary);

while(fin.read((char\*)&obj,sizeof(obj)))

{

if(strcmp(obj.retpswd(),passw)==0)

{strcpy(username,obj.retname());

luckyno=obj.retpno();} }

cout<<" Player no: "<<globb<<endl;

cout<<"Wotcher astronaut "<<username<<"! Your score is: "<<obj1.retscr()<<endl;

ifstream fin1("highscore.txt",ios::in|ios::binary);

cout<<"\n\n LEADERBOARD:\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout<<"\t\tPLAYER NO.\t\t\tSCORE \n";

while(fin1.read((char\*)&obj1,sizeof(obj1)))

{

if(counter<8)

{obj1.put\_sc(); counter++; cout<<endl; }}

fin1.close();

fin.close();

}

void desc() //WRITE ONTO H.S. FILE

{

HighScore obj,obj1;

obj1.assign();

int flag3=0,flag1=0;

int num=obj1.retscr();

ifstream fin("highscore.txt",ios::in|ios::binary);

ofstream fout("temp6.txt",ios::out|ios::binary);

while(fin.read((char\*)&obj,sizeof(obj)))

{

flag3=1;

if(num<=obj.retscr())

fout.write((char\*)&obj,sizeof(obj));

else

{

fout.write((char\*)&obj1,sizeof(obj1));

flag1++;

fout.write((char\*)&obj,sizeof(obj));

while(fin.read((char\*)&obj,sizeof(obj)))

fout.write((char\*)&obj,sizeof(obj));

} }

if(flag3==0||flag1==0)

fout.write((char\*)&obj1,sizeof(obj1));

fin.close();

fout.close();

remove("highscore.txt");

rename("temp6.txt","highscore.txt"); }

void del() //DEL RECORD

{

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TC\\BGI");

setcolor(LIGHTCYAN);

cleardevice();

char a1[20],a3[20];

HighScore obj6; int q=0;

int k;

ifstream fin("highscore.txt",ios::in|ios::binary);

ofstream fout("temp5.txt",ios::out|ios::binary);

outtextxy(10,10,"Enter your password to make changes: ");

strcpy(a1,"");

strcpy(a3,"");

char c;

int e=0,pos=0;

do{

c=getch();

switch(c)

{

case 8:

if(pos)

{

pos--;

a3[pos]=0;

a1[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a1[pos]=c;

a3[pos]='\*';

pos++;

a1[pos]=0;

a3[pos]=0;

}}

cleardevice();

outtextxy(10,10,"Enter your password to make changes: ");

outtextxy(315,10,a3);

}

while(!e);

strcpy(pass1,a1);

if(strcmp(pass,pass1)==0)

{

cleardevice();

outtextxy(10,10,"ENTER YOUR ID NUMBER : ");

strcpy(a1,"");

char c;

int e=0,pos=0;

do{

c=getch();

switch(c)

{

case 8:

if(pos)

{

pos--;

a1[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(c>=' ' && c<='~')

{

a1[pos]=c;

pos++;

a1[pos]=0; } }

cleardevice();

outtextxy(10,10,"ENTER YOUR ID NUMBER : ");

outtextxy(250,10,a1); }

while(!e);

k=atoi(a1);

if(k==globb)

{ while(fin.read((char\*)&obj6,sizeof(obj6)))

{

if(k!=obj6.retpno1())

fout.write((char\*)&obj6,sizeof(obj6));

else

q++;

}

if(q==0)

outtextxy(10,35,"Sorry fellow astronaut! We're afraid this player no. does not exist!");

}

else

{ outtextxy(10,60,"Kindly delete your own record."); getch();

return;} }

else

outtextxy(10,35,"Record irremovable. ");

fin.close();

fout.close();

remove("highscore.txt");

rename("temp5.txt","highscore.txt"); }

void compare() //VIEW LEADERBOARD

{

cleardevice();

int f=0;

login obj;

ifstream fin("login.txt",ios::in|ios::binary);

outtextxy(10,10,"Enter your password to view high scores: ");

char psw[50];

char ck;

int e=0,pos=0;

strcpy(psw,"");

do{

ck=getch();

switch(ck)

{

case 8:

if(pos)

{

pos--;

psw[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(ck>=' ' && ck<='~')

{

psw[pos]=ck;

pos++;

psw[pos]=0; } }

cleardevice();

outtextxy(10,10,"Enter your password to view high scores: ");

outtextxy(400,10,psw);

}

while(!e);

strcpy(pass,psw);

while(fin.read((char\*)&obj,sizeof(obj)))

{

if(strcmp(psw,obj.retpswd())==0)

{globput\_sc(psw); f++;}

}

if(f==0)

{

cleardevice();

outtextxy(10,10,"User data corresponding to password entered does not exist.");

}

fin.close();

getch();

cleardevice();

}

void high() //PAGE AFTER GAME

{

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TC\\BGI");

setcolor(LIGHTCYAN);

for(int sld=0; sld<80; sld++) //TO CLEAR THE SCREEN FOR THE TRIVIA

cout<<"\n";

char ck;

int pos=0,e=0;

char a[10];

char k;

char k1[10];

int c;

loop2:

pos=0,e=0;

cleardevice();

outtextxy(0,10,"ENTER 1 TO SAVE YOUR SCORE, 2 TO DISPLAY, 3 TO DELETE, ANY OTHER INTEGER TO EXIT : ");

rectangle(0,0,getmaxx()-1,getmaxy()-1);

strcpy(a,"");

do{

ck=getch();

rectangle(0,0,getmaxx()-1,getmaxy()-1);

switch(ck)

{

case 8:

if(pos)

{

pos--;

a[pos]=0;

}

break;

case 13:

e=1;

break;

default:

if(ck>=' ' && ck<='~')

{

a[pos]=ck;

pos++;

a[pos]=0; }

}

cleardevice();

rectangle(0,0,getmaxx()-1,getmaxy()-1);

ins();

outtextxy(10,30,a);

}

while(!e);

c=atoi(a);

switch(c)

{

case 1:

desc(); goto loop2;

case 2:

compare(); goto loop2;

case 3:

del(); goto loop2;

default:exit(1); } }

void welcome() //OPENING PAGE

{

for(int g=0; g<3; g++)

{

int h1,h2;

clrscr();

if(g%2==0)

{

textcolor(MAGENTA);

h1=random(100);

h2=random(34);

gotoxy(h1,h2);

cprintf("WELCOME");

delay(400);

clrscr();

delay(400);

h1=random(100);

h2=random(34);

gotoxy(h1,h2);

cprintf("TO ");

delay(400);

clrscr();

delay(400);

h1=random(100);

h2=random(34);

gotoxy(h1,h2);

delay(400);

cprintf("THE ORION ");

delay(400);

}

else

{

delay(400);

textcolor(YELLOW);

clrscr();

delay(400);

h1=random(100);

h2=random(34);

gotoxy(h1,h2);

cprintf("WELCOME ");

delay(400);

clrscr();

delay(400);

h1=random(100);

h2=random(34);

gotoxy(h1,h2);

cprintf("TO ");

delay(400);

clrscr();

delay(400);

h1=random(100);

h2=random(34);

gotoxy(h1,h2);

cprintf("THE ORION ");

delay(800); } } }

void bar(int z) //ROCKET

{

window(z+1,22,70,25);

textcolor(10);

cprintf(" Û ");

window(z+1,23,70,25);

cprintf(" Þ");

textcolor(9);

cprintf("Û");

textcolor(10);

cprintf("Ý ");

window(z+1,24,70,25);

cprintf(" Þ");

textcolor(9);

cprintf("ÛßÛ");

textcolor(10);

cprintf("Ý ");

}

void check() //CHECKS IF THE ANSWER ENTERED IS RIGHT

{

int n; char x;

trivia obj;

cout<<"ENTER YOUR ANSWER (OPTION NUMBER) : " ;

cin>>n;

ifstream fin("trivia.txt",ios::in|ios::binary);

while(fin.read((char\*)&obj,sizeof(obj)))

{

if(ctr==obj.retqno())

{

if(n==obj.retans())

cout<<"\n\nWELL DONE, CREW! YOU MAY CONTINUE THE GAME.\n\n "<<endl;

else

{

cout<<" \n\nW R O N G A N S W E R."<<endl<<" G A M E O V E R :((";

getch();

high();

} } }

fin.close();

}

void trivia\_dis() //DISPLAYS APPROPRIATE QUESTIONS USING A COUNTER

{

trivia obj;

ifstream fin("trivia.txt",ios::in|ios::binary);

while(fin.read((char\*)&obj,sizeof(obj)))

if(ctr==obj.retqno())

obj.trivia\_put();

fin.close();

}

void options\_assign(int var) //DISPLAYS THE OPTIONS TO THE QUESTIONS

{

trivia obj;

strcpy(ob[0].o1,"1. PLUTO");

strcpy(ob[0].o2,"2. MERCURY");

strcpy(ob[0].o3,"3. NEPTUNE");

strcpy(ob[1].o1,"1. 1000 YEARS OLD");

strcpy(ob[1].o2,"2. 13 BILLION YEARS OLD");

strcpy(ob[1].o3,"3. 5 MILLION YEARS OLD ");

strcpy(ob[2].o1,"1. THE SUN");

strcpy(ob[2].o2,"2. A SUPERNOVA");

strcpy(ob[2].o3,"3. A BLACK HOLE");

strcpy(ob[3].o1,"1. 181.5");

strcpy(ob[3].o2,"2. 181");

strcpy(ob[3].o3,"3. 181.99");

strcpy(ob[4].o1,"1. THE SUN");

strcpy(ob[4].o2,"2. POLARIS ");

strcpy(ob[4].o3,"3. VY CANIS MAJORIS");

strcpy(ob[5].o1,"1. 1.3 MILLION");

strcpy(ob[5].o2,"2. 100 BILLION");

strcpy(ob[5].o3,"3. 500,000");

strcpy(ob[6].o1,"1. JUPITER");

strcpy(ob[6].o2,"2. MARS");

strcpy(ob[6].o3,"3. VENUS");

strcpy(ob[7].o1,"1. TO GET THEM ADJUSTED TO SPACE CONDITIONS ");

strcpy(ob[7].o2,"2. TO PREVENT EXTRATERRESTRIAL GERMS THAT THEY MAY BE CARRYING FROM BEING SPREAD");

strcpy(ob[7].o3,"3. TO PREVENT TERRORISM");

strcpy(ob[8].o1,"1. IT IS THE BIGGEST SPACE SHUTTLE YET ");

strcpy(ob[8].o2,"2. IT CAN TRAVEL AT THE SPEED OF LIGHT ");

strcpy(ob[8].o3,"3. IT CAN HOST 4 ASTRONAUTS ");

strcpy(ob[9].o1,"1. DEIMOS ");

strcpy(ob[9].o2,"2. TITAN ");

strcpy(ob[9].o3,"3. PHOBOS ");

strcpy(ob[10].o1,"1. PLUTO ");

strcpy(ob[10].o2,"2. URANUS ");

strcpy(ob[10].o3,"3. NEPTUNE ");

cout<<endl;

if(ctr)

{

puts(ob[var].o1); cout<<endl;

puts(ob[var].o2); cout<<endl;

puts(ob[var].o3); cout<<endl;

} }

void table() //DISPLAYS THE TABLE ON THE RIGHT

{

\_setcursortype(\_NOCURSOR);

textcolor(15);

for(int i=0;i<18;i++)

window(66,3,78,25);

textcolor(11);

cprintf("Press X ");

cprintf("to Exit\_\_\_\_\_ \n");

textcolor(14);

cprintf("Press P to ");

cprintf("Pause Game\_\_ \n");

textcolor(11);

cprintf("Press Space ");

cprintf("to Shoot\_\_\_\_ \n");

textcolor(14);

cprintf("Press Left, ");

cprintf("Right to Move\n");

textcolor(11);

cprintf("Press H: Save score ");

cprintf("& End game \n");

textcolor(WHITE);

cprintf(" Score : ");

if(!HELLO)

cprintf(" 0 ");

HELLO++; }

void options\_dis() //DISPLAYS QUESTIONS AND OPTIONS

{

getch();

if(glob<11)

{

for(int j=0; j<2; j++)

{

trivia\_dis();

options\_assign(var);

var++;

check();

ctr++;

glob++;

} } }

void up() //BULLETS GO UP

{

int flag=0,k;

for(int i=0;i<sz && xx[i]!=0 ;i++)

{

for(k=0;k<20 ;k++)

{

if(xx[i]==xp[k] && yy[i]==yp[k]&& xp[k]!=0)

{

xx[i]=0;

window(74,20,80,24);

scr++;

textcolor(15);

cout<<scr;

if(scr%5==0&&scr!=0)

{

delay(100);

for(int sld=0; sld<80; sld++) //TO CLEAR THE SCREEN FOR THE TRIVIA

cout<<"\n";

options\_dis();

for(sld=0; sld<80; sld++) //TO CLEAR THE SCREEN FOR THE TRIVIA

cout<<"\n";

delay(2000);

table(); } } } }

textcolor(10);

for(i=0;i<20;i++)

{

if(xp[i]!=0)

{

flag++;

window(xp[i],yp[i],xp[i]+1,yp[i]+1);

if(yp[i]!=1)

cprintf("%c",4);

window(xp[i],yp[i]+1,xp[i]+1,yp[i]+2);

if(yp[i]!=21)

cprintf(" ");

yp[i]--;

delay(22/flag);

}

if(yp[i]==0)

xp[i]=0;

}

textcolor(WHITE); }

void status(int x) //SHOWS THE STATUS OF THE PLAYER

{

window(67,21,79,23);

if(x==1)

{

textcolor(LIGHTGREEN);

cprintf(" Playing.. ");

}

else if(x==2)

cprintf(" Paused ");

else

{

textcolor(LIGHTRED);

cprintf(" Stopped ");

} }

void down() //OBSTACLES COMING DOWN

{

int x,i=0;

yy[i]=2;

if(random(10)==1){

x=random(52)+7;

textcolor(12);

for(i=0;i<sz;i++)

{

if(xx[i]==0)

{

xx[i]=x;

yy[i]=2;

break;

}

}

for(i=0;i<sz;i++)

{

if(random(2)){

window(xx[i],yy[i],xx[i],yy[i]+2);

cprintf("V");

if(yy[i]>=20)

{

status(3);

getch();

delay(1500);

getch();

high(); }

yy[i]++; }

}

textcolor(WHITE);

delay(300); } }

void main()

{

clrscr();

welcome(); rules();

clrscr();

start();

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TC\\BGI");

cleardevice();

setcolor(YELLOW);

settextstyle(0,HORIZ\_DIR,3);

outtextxy(getmaxx()/2-150,getmaxy()/2,"Brace yourself..."); delay(1000);

cleardevice();

outtextxy(getmaxx()/2-100,getmaxy()/2+15,"T minus "); delay(1000);

cleardevice();

outtextxy(getmaxx()/2-100,getmaxy()/2+45,"5..."); delay(1000);

cleardevice();

outtextxy(getmaxx()/2-100,getmaxy()/2+45,"4..."); delay(1000);

cleardevice();

outtextxy(getmaxx()/2-100,getmaxy()/2+45,"3..."); delay(1000);

cleardevice();

outtextxy(getmaxx()/2-100,getmaxy()/2+45,"2..."); delay(1000);

cleardevice();

outtextxy(getmaxx()/2-100,getmaxy()/2+45,"1..."); delay(1000);

cleardevice();

outtextxy(getmaxx()/2-150,getmaxy()/2,"BLAST OFF!!!!");delay(1000);

cleardevice();

closegraph();

for(int i=0; i<80;i++)

cout<<"\n";

char ch;

table();

int x=0;

for(i=0;i<20;i++)

{ xp[i]=0; }

for(i=0;i<sz;i++)

xx[i]=0;

bar(x);

while(1)

{

status(1);

while(!kbhit())

{

up();

down();

}

ch=getch();

switch(tolower(ch))

{

case 'p' : status(2);

delay(200);

getch();

case ' ' :

delay(80);

for(int i=0;i<20;i++)

{

if(xp[i]==0)

{

xp[i]=x+5;

yp[i]=20;

break; } }

break;

case 0 : ch=getch();

if(ch==75 && x>0)

{

x-=1;

bar(x);

}

else if(ch==77 && x<55)

{

x+=1;

bar(x);

}

break;

case 'x' : exit(0);

case 'h' : high(); }

}

}

**PROJECT LOG**

1. 9/7/2018

 Brainstorming: decided to design a space shooting game for the project

1. 11/7/2018

 Started the project- idea implementation methodology

 Decided the game name as ‘ORION’

 Browsed the internet for inspiration

 Decided the layout of the game, classes that need to be used for it

1. 12/7/2018

 Created a text file “Rules.txt” and wrote the rules of the game onto it

1. 19/7/2018

 Worked on the graphics for the game

1. 26/7/2018

Worked on the graphics for the game

1. 2/8/2018

Worked on the graphics for the game

1. 3/8/2018

Worked on the graphics for the game

1. 9/8/2018

 Created the welcome page to be displayed at the start of the game

1. 16/8/2018

 Decided to include space trivia as part of the game and browsed the net to come up with interesting questions related to the universe

1. 23/8/2018

 Created a structure Option to list the options for the array of questions

Created class ‘Trivia’

Created file “trivia.txt” and wrote the questions onto it

Created functions check(),opt(), dis()

1. 30/8/2018

 Created class HighScore and file “highscore.txt”

Created functions desc(),compare()

1. 6/9/2018

Corrected the logical and syntax errors

1. 15/9/2018

Created del() to work on deletion of records

1. 21/9/2018

 Created high() to allow users to make score entries, exit the game, view scores, etc.

1. 27/9/2018

 Corrected logical and syntax errors

1. 15/10/2018

 Created class ‘Login’ and its member functions, data members

1. 16/10/2018

 Created modify() to allow user to modify his player number

1. 25/10/2018

Created put2() to display scores

1. 2/11/2018

Corrected logical errors

1. 12/11/2018

 Corrected binary file errors

1. 16/11/2018

 Corrected binary file errors

1. 22/11/2018

 Debugged the program

1. 23/11/2018

 Improved the aesthetics of the project

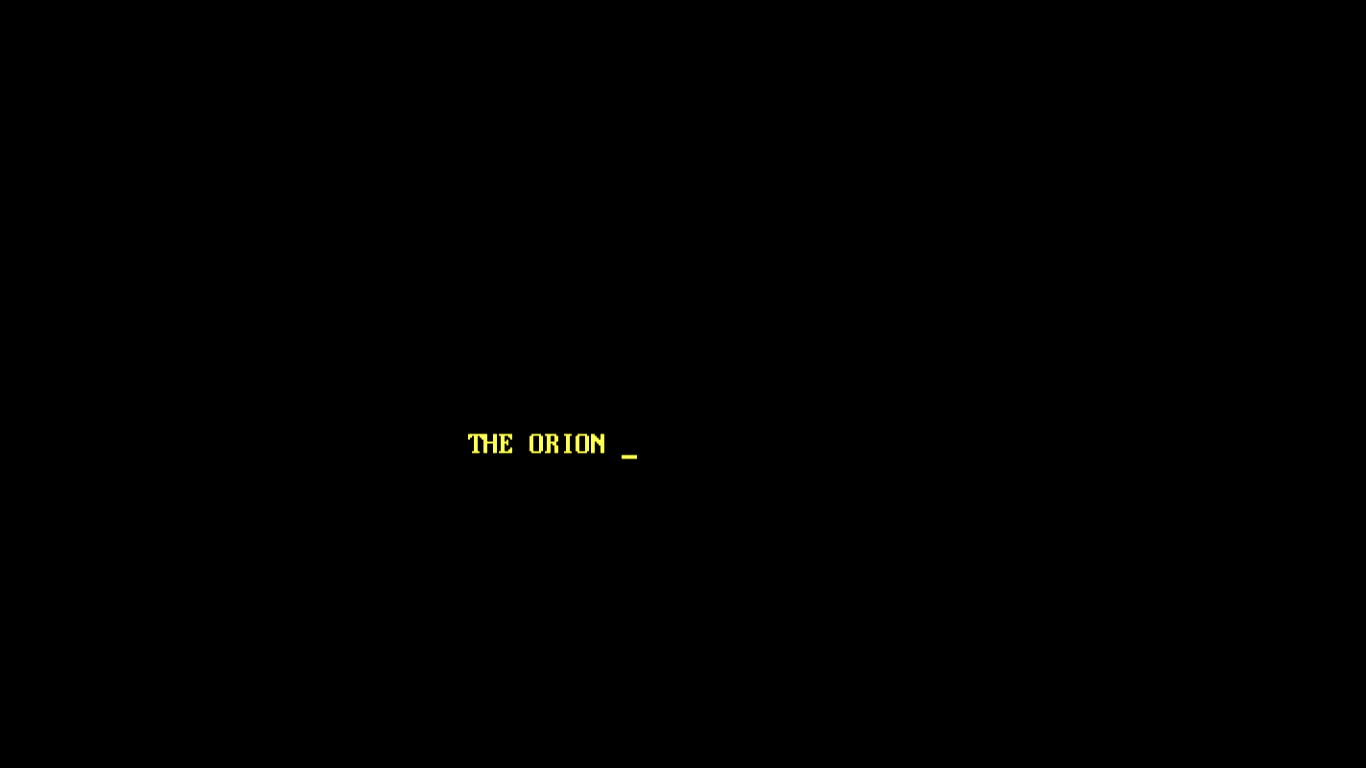
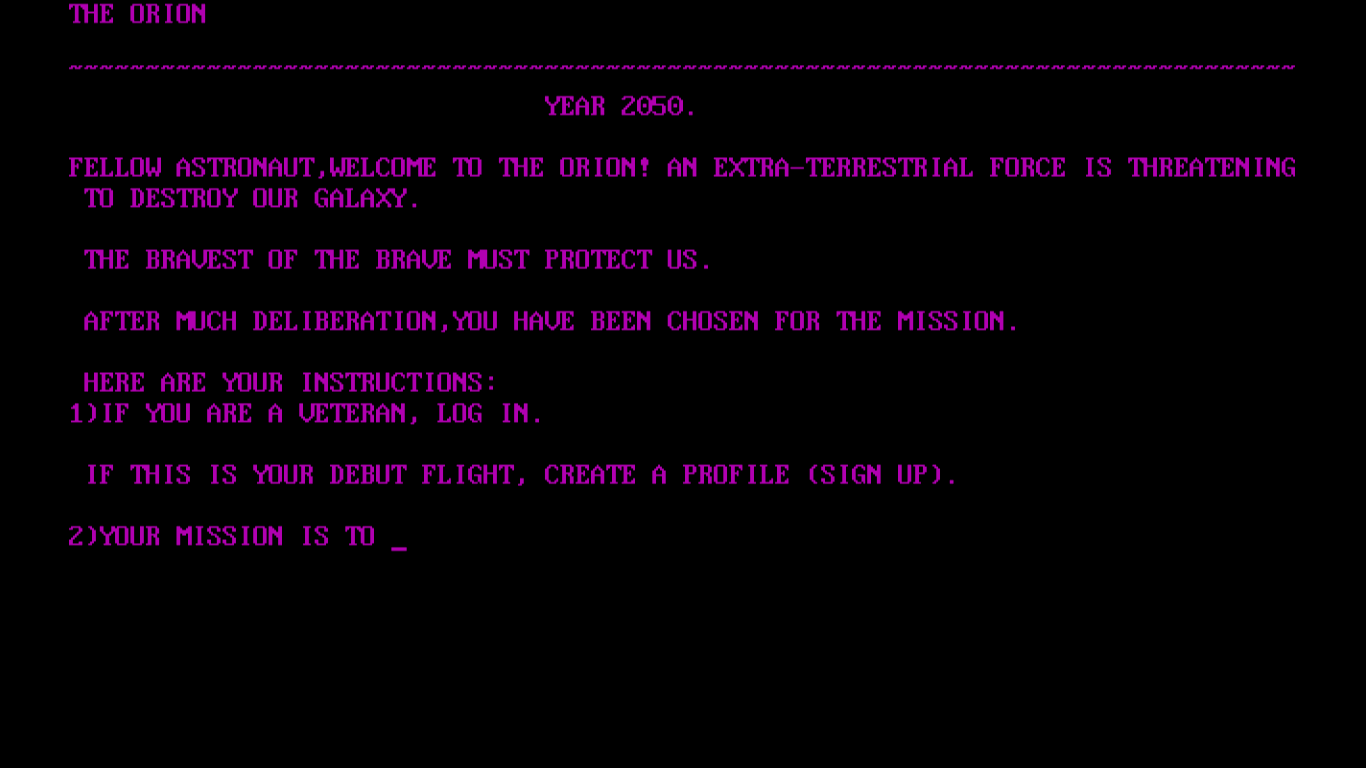
1. 29/11/2018

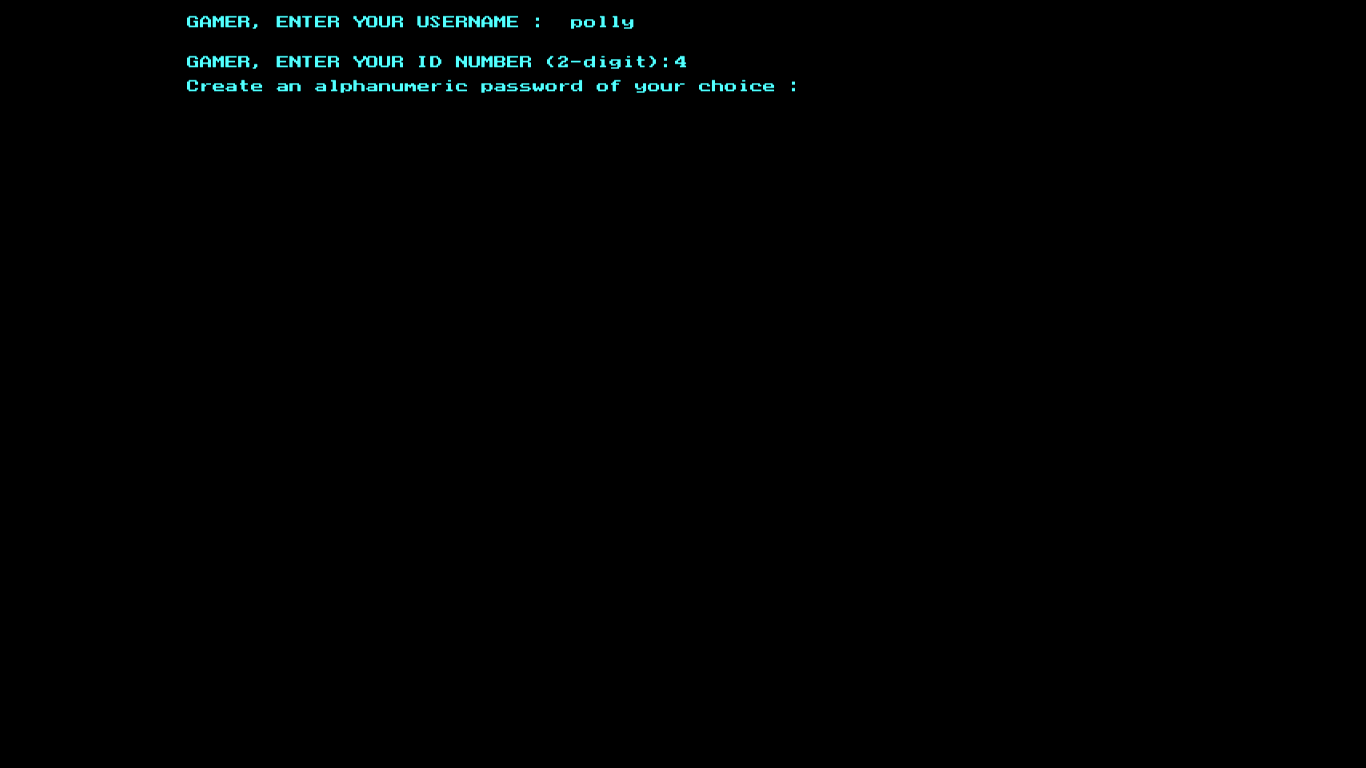
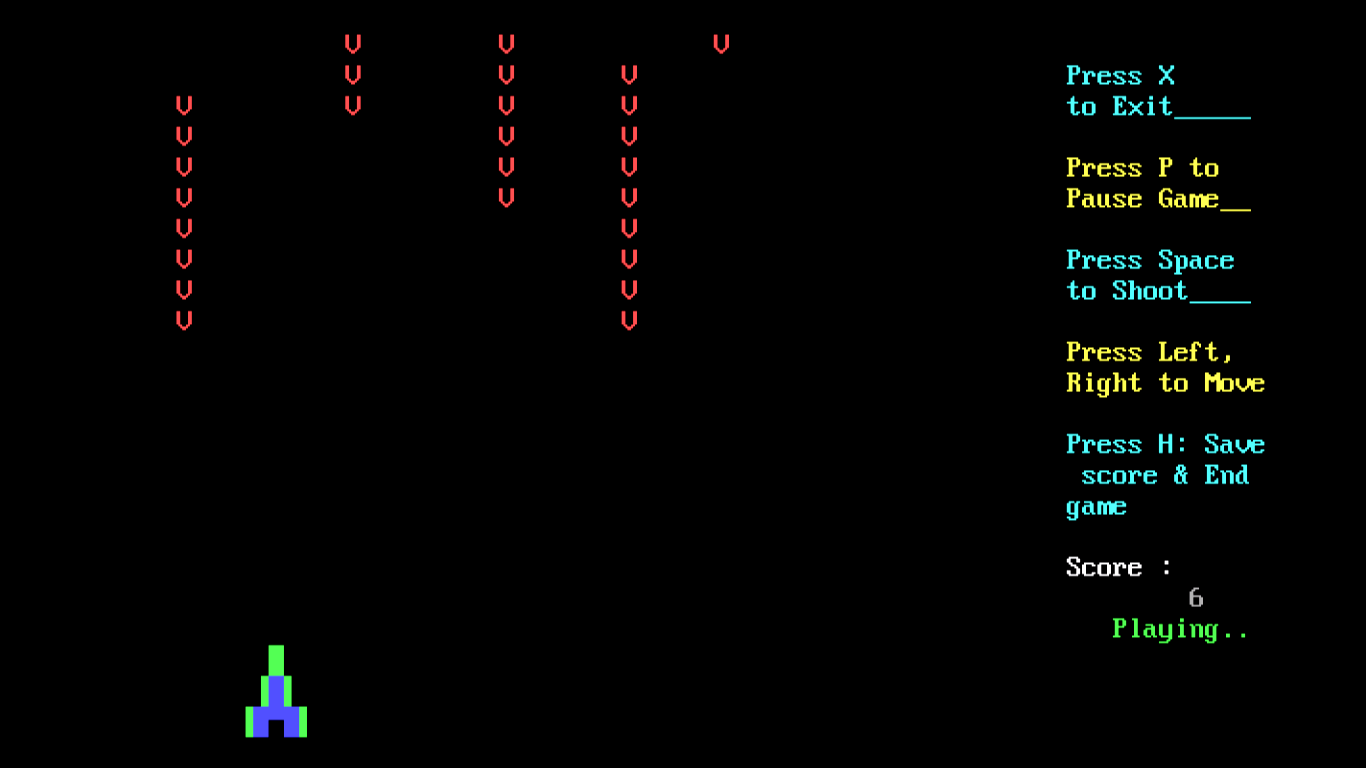
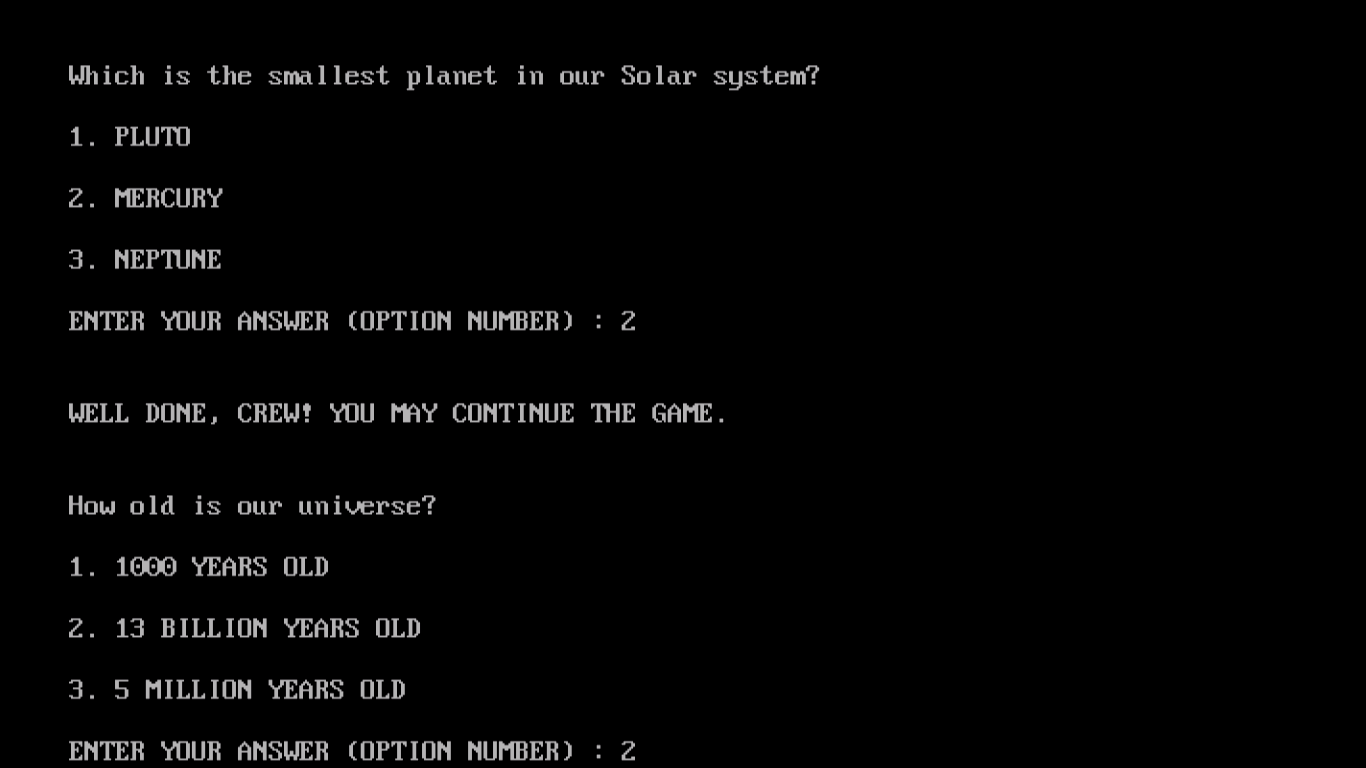
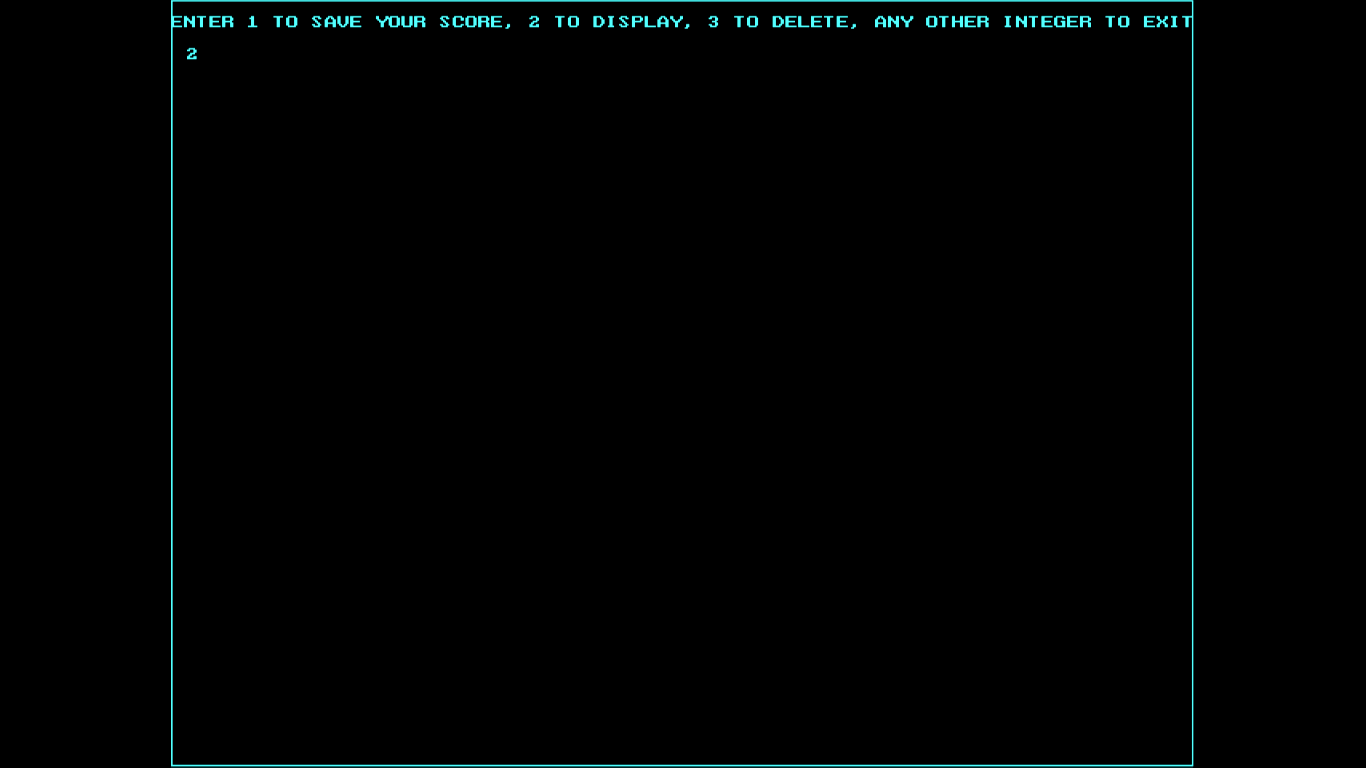
 Final touches

1. 3/12/2018

Submission

**SAMPLE OUTPUT**

**BIBLIOGRAPHY**

https://en.wikipedia.org

https://www.youtube.com

‘Computer Science with C++’:Sumita Arora