Computer Programming Lab

Term Project

Semester and Section: BS (IT)-1B

Team Members:

Name: Maryam Qayyum **Enrolment No:** 01-135222-010

Name: Maria Saeed **Enrolment No:** 01-135222-044

Department of Computer Science

BUBBLE SHOOTER

MENU:

Menus are selection screens that are usually used to help in project progress. They give the viewer the choice to choose what to do, such as choosing whether to log in, register, or exit a game, or whether to enable or disable particular options or manage particular tools. There are various menu kinds, with the main menu serving as the player's official introduction at the start of the game being the most frequent. Since most programmes and games created outside of Scratch have menus, adding them might give a project a more polished appearance we add filling and structures in a menu .

contents:

login:

In order to login press 1 .

In order for the programme to recognise whose user is making the request and respond appropriately, logging in is fundamentally necessary. Enter your username and password to access the game if you have already logged in.

```
struct player(int, int) //structures
{
    int id;
    int password
}

void login()
{
    int count;
    string user, pass, u, p;
    system("cls");
    cout << "please enter the following details" << endl;
    cout << "PASSWORD :";
    cin >> player.id >> player.password; //structures

    ifstream input("database.txt"); //filling (read input from the file)
    while (input >> u >> p)
    {
        if (u == user && p == pass)

        {
            count = 1;
            system("cls");
        }
    }
    input.close(); //filling
    if (count == 1)
    {
        cout << "\nHello" << user << "\nLOGIN SUCESS\nWe're glad that you're here.\nThanks for logging in\n";
        cin.get();
```

Register:

In order to register press 2.

You must first register and establish an account in order to play the games after selecting the one that best meets your needs. The process is quite straightforward, and it might only take one minute. It costs nothing to create an account.

```
        }
    }
    input.close(); //filling
    if (count == 1)
    {
        cout << "\nHello" << user << "\nLOGIN SUCESS\nWe're glad that you're here.\nThanks for logging in\n";
        cin.get();
        cin.get();
        main();
    }
    else
    {
        cout << "\nLOGIN ERROR\nPlease check your username and password\n";
        main();
    }
}
struct player(string, int) //structure of player
{
    string name;
    int pass;
}


void registr()
{

    string reguser, regpass, ru, rp;
    system("cls");
    cout << "Enter the username :";
    cout << "\nEnter th password :";
    cin >> player.name >> player.pass; // structure

    ofstream reg("database.txt", ios::app); //read and write filling
    reg << reguser << ' ' << regpass << endl;
    system("cls");
    cout << "\nRegistration Sucessful\n";
    main();
```

29 0

Forgot password or username:

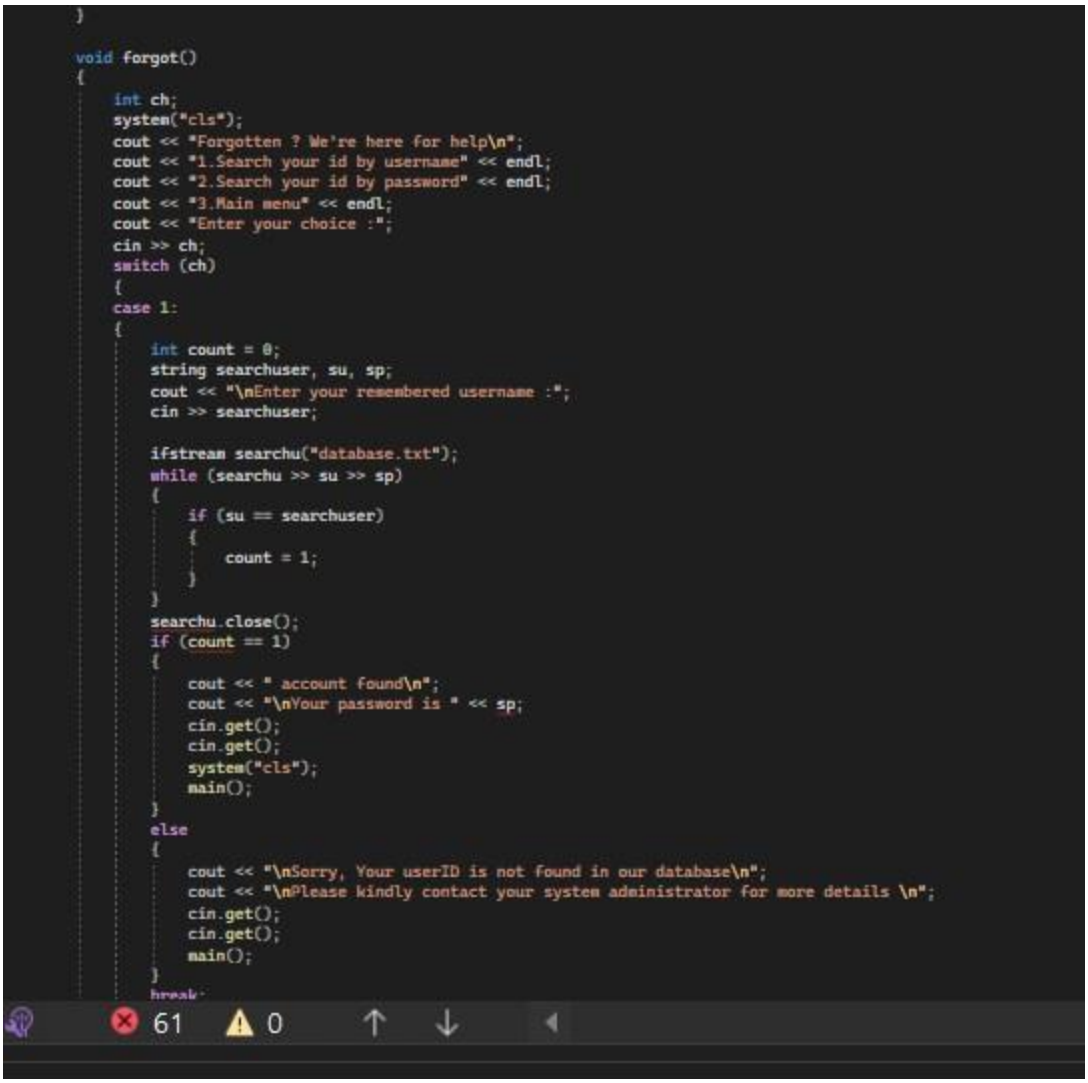
if you forgotten your password or username press 3.

What is the purpose of the forgot password feature? Almost all services do not recover or display the previously used password when a user clicks the "forgot password" button. Instead, you are either instructed to enter a new password or given a fresh temporary one. You now have a new password to access the account in both situations.

```
106     }
107 }
108 struct player(string, pass)
109 {
110     string name;
111     int id;
112 }
113
114
115 void forgot()
116 {
117     int ch;
118     system("cls");
119     cout << "Forgotten ? We're here for help\n";
120     cout << "1.Search your id by username" << endl;
121     cout << "2.Search your id by password" << endl;
122     cout << "3.Main menu" << endl;
123     cout << "Enter your choice :";
124     cin >> ch;
125     switch (ch)
126     {
127     case 1:
128     {
129         int count = 0;
130         string searchuser, su, sp;
131         cout << "\nEnter your remembered username :";
132         cin >> searchuser;
133
134         ifstream searchu("database.txt"); //filling
135         while (searchu >> su >> sp)
136         {
137             if (su == searchuser)
138             {
139                 count = 1;
140             }
```

```
    {
        cout << "\n\nHurray, account found\n";
        cout << "\nYour password is " << sp;
        cin.get();
        cin.get();
        system("cls");
        main();
    }
    else
    {
        cout << "\nSorry, Your userID is not found in our database\n";
        cout << "\nPlease kindly contact your system administrator for more details \n";
        cin.get();
        cin.get();
        main();
    }
    break;
}
case 2:
{
    int count = 0;
    string searchpass, su2, sp2;
    cout << "\nEnter the remembered password :";
    cin >> searchpass;

    ifstream searchp("database.txt"); //filling
    while (searchp >> su2 >> sp2)
    {
        if (sp2 == searchpass)
        {
            count = 1;
        }
    }
    searchp.close(); //filling
    if (count == 1)
    {
        cout << "\nYour password is found in the database \n";
```



```
}

void forgot()
{
    int ch;
    system("cls");
    cout << "Forgotten ? We're here for help\n";
    cout << "1.Search your id by username" << endl;
    cout << "2.Search your id by password" << endl;
    cout << "3.Main menu" << endl;
    cout << "Enter your choice :";
    cin >> ch;
    switch (ch)
    {
    case 1:
    {
        int count = 0;
        string searchuser, su, sp;
        cout << "\nEnter your remembered username :";
        cin >> searchuser;

        ifstream searchu("database.txt");
        while (searchu >> su >> sp)
        {
            if (su == searchuser)
            {
                count = 1;
            }
        }
        searchu.close();
        if (count == 1)
        {
            cout << " account found\n";
            cout << "\nYour password is " << sp;
            cin.get();
            cin.get();
            system("cls");
            main();
        }
        else
        {
            cout << "\nSorry, Your userID is not found in our database\n";
            cout << "\nPlease kindly contact your system administrator for more details \n";
            cin.get();
            cin.get();
            main();
        }
        break;
```

61 0

```
        cin.get();
        cin.get();
        system("cls");
        main();
    }
    else
    {
        cout << "\nSorry, Your userID is not found in our database\n";
        cout << "\nPlease kindly contact your system administrator for more details \n";
        cin.get();
        cin.get();
        main();
    }
    break;
}
case 2:
{
    int count = 0;
    string searchpass, su2, sp2;
    cout << "\nEnter the remembered password :";
    cin >> searchpass;

    ifstream searchp("database.txt");
    while (searchp >> su2 >> sp2)
    {
        if (sp2 == searchpass)
        {
            count = 1;
        }
    }
    searchp.close();
    if (count == 1)
    {
        cout << "\nYour password is found in the database \n";
        cout << "\nYour Id is : " << su2;
        cin.get();
        cin.get();
        system("cls");
        main();
    }
    else
    {
        cout << "Sorry, We cannot found your password in our database \n";
        cout << "\nkindly contact your administrator for more information\n";
        cin.get();
        cin.get();
        main();
    }

    break;
}

case 3:
{
    cin.get();
```

61 ⚠ 0

Exit :

To exit a game press 4.

This exit feature allows user to exit the game.

Functions and structures for the player for the player :

Another important function that we add is for the players. They can enter their names and start shooting

Generate enemy:

```
}
void genEnemy(int i)
{
    enemyX[i] = 3 + rand() % (70 - 10);
}
```

Draw enemy:

```
void drawEnemy(int ind)
{
    if (enemyflag[ind] = true)
    {
        gotoxy(enemyX[ind], enemyY[ind]);
        cout << " ** ";
        gotoxy(enemyX[ind], enemyY[ind] + 1);
        cout << " ** ";
        gotoxy(enemyX[ind], enemyY[ind] + 2);
        cout << " ** ";
        gotoxy(enemyX[ind], enemyY[ind] + 3);
        cout << " ** ";
    }
}
```

Generate bullet:

```
void genbullet()
{
    bullets[ind][0] = 22;
    bullets[ind][1] = shooterposition;
    bullets[ind][2] = 22;
    bullets[ind][3] = shooterposition + 4;
    ind++;
    if (ind == 0)
        ind = 0;
}
```

Move bullet:

```
void movebullet()
{

    for (int i = 0; i < 20; i++)
    {
        if
            (bullets[i][0] > 2)
            bullets[i][0]--;
        else
            bullets[i][0] = 0;

        if (bullets[i][2] > 2)
            bullets[i][2]--;
        else
            bullets[i][2] = 0;
    }
}
```

Draw bullet:

```
void drawbullets()
{
    for (int i = 0; i < 20; i++)
    {
        if (bullets[i][0] > 1)
        {
            gotoxy(bullets[i][1], bullets[i][0]); cout << " , ";
            gotoxy(bullets[i][3], bullets[i][2]); cout << " , ";
        }
    }
}
```

Draw shooter:

```
void drawshooter()
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            gotoxy(j + shooterposition, i + 22);
            cout << shooter[i][j];

        }


    }
}
```

---

collision:

```
int collision()
{
    if (enemyY[0] + 4 >= 23)
    {
        if (enemyX[0] + 4 - shooterposition >= 0 && enemyX[0] + 4 - shooterposition < 8)
        {
            return 1;
        }

        cout << collision();
    }
    return 0;
}
```
