

Contents

Main form	1
Player Data	6

Main form

```
/*Author: Tyler Doye
 * 12002347
 *Name of app: starcraft 2 ladder
 */
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Collections;

namespace Starcraft2Ladder
{
    public partial class frmStarcraftData : Form
    {

        public frmStarcraftData()
        {
            InitializeComponent();
            cmbReportType.SelectedIndex = 1;
        }

        private void goToWebpageToolStripMenuItem_Click(object sender, EventArgs e)
        {
            // this willl mkae the menu button for go to website link you t othe
            website on click

            try
            {
                System.Diagnostics.Process.Start("http://us.battle.net/sc2/en/");
            }
            catch (Exception)
            {
                throw;
            }
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            // will make sure all the txtboxes are filled in if not they will be made
            red and the user will not be able to save the data until the form is complete
            Color redColor = Color.FromArgb(255, 0, 0);
            if (txtFirstName.Text.Equals(""))
```

```

{
    MessageBox.Show(lblFirstName.Text+" is Missing");
    txtFirstName.BackColor = redColor;
}
if (txtSurname.Text.Equals(""))
{
    MessageBox.Show(lblSurname.Text + "is Missing");
    txtSurname.BackColor = redColor;
}
if (txtAlias.Text.Equals(""))
{
    MessageBox.Show(lblAlias.Text + " is Missing");
    txtAlias.BackColor = redColor;
}
if (txtWins.Text.Equals(""))
{
    MessageBox.Show(lblWins.Text + " is Missing");
    txtWins.BackColor = redColor;
}
if (txtLosses.Text.Equals(""))
{
    MessageBox.Show(lblLosses.Text + " is Missing");
    txtLosses.BackColor = redColor;
}
if (txtApmAverage.Text.Equals(""))
{
    MessageBox.Show(lblAvgApm.Text + " is Missing");
    txtApmAverage.BackColor = redColor;
}
if (txtLeagueStanding.Text.Equals(""))
{
    MessageBox.Show(lblLeagueStanding.Text + " is Missing");
    txtLeagueStanding.BackColor = redColor;
}
else
{
    //making the variables and parsing from text box that will go in the
constructor
    Int32 wins = Int32.Parse(txtWins.Text);
    Int32 losses = Int32.Parse(txtLosses.Text);
    Int32 apm = Int32.Parse(txtApmAverage.Text);
    Int32 leagueStanding = Int32.Parse(txtLeagueStanding.Text);

    // these if statements will make a variable in for league (radio
buttons) that can be used in the constructor
    string league = string.Empty;
    if (rdbBronze.Checked)
    {
        league = "Bronze";
    }
    else if (rdbSilver.Checked)
    {
        league = "Silver";
    }
    else if (rdbGold.Checked)
    {
        league = "Gold";
    }
    else if (rdbPlatinum.Checked)
    {
        league = "Platiumum";
    }
}

```

```

        else if (rdbDiamond.Checked)
        {
            league = "Diamond";
        }
        else if (rdbMasters.Checked)
        {
            league = "Masters";
        }

        // starting the filestream and binary writer
        FileStream filStream = new FileStream("LeagueLadder.dat",
        FileMode.Append);
        BinaryWriter binWriter = new BinaryWriter(filStream);

        // writes the objects into a binary file
        binWriter.Write(league);
        binWriter.Write(txtFirstName.Text);
        binWriter.Write(txtSurname.Text);
        binWriter.Write(txtAlias.Text);
        binWriter.Write(wins);
        binWriter.Write(losses);
        binWriter.Write(apm);
        binWriter.Write(leagueStanding);

        binWriter.Close();
        filStream.Close();
    }
}

private void btnLoad_Click(object sender, EventArgs e)
{
    string report= string.Empty;
    string reportType=string.Empty;
    // this will make a report variable that will decide which form to create
    and sets the name of the report heading
    if (cmbReportType.Text.Equals("more than 100 wins"))
    {
        report= ("more_than_100_wins.txt");
        reportType = ("That have more than 100 wins");
    }
    if (cmbReportType.Text.Equals("By Favourable w/l"))
    {
        report = ("Favourable_wins.txt");
        reportType = ("That have more wins than Losses");
    }
    if (cmbReportType.Text.Equals("If in top 8"))
    {
        report = ("if_in_top_8.txt");
        reportType = ("That are in the top 8 of their league");
    }

    FileStream filStream = new FileStream("LeagueLadder.dat", FileMode.Open,
    FileAccess.Read);
    ArrayList playerList = new ArrayList();
    try
    {
        int i = 0;
        StreamWriter strmWriter = new StreamWriter(report, false);
        using (BinaryReader binReader = new BinaryReader(filStream))
        {

```

```

strmWriter.WriteLine("Report of Players"+ reportType );
while (binReader.PeekChar() != -1)
{
    //every loop will write a new line to the report
    //values are put into variables to easily minipulate
    string league = binReader.ReadString();
    string firstName = binReader.ReadString();
    string surname = binReader.ReadString();
    string alias = binReader.ReadString();
    Int32 wins = binReader.ReadInt32();
    Int32 losses = binReader.ReadInt32();
    Int32 apm = binReader.ReadInt32();
    Int32 leagueStanding = binReader.ReadInt32();

    PlayerData playerObject = new PlayerData(league, firstName,
surname, alias, wins, losses, apm, leagueStanding);

    playerList.Add(playerObject);

    // this will create the txtfile for players with more than 100
wins
    if (cmbReportType.Text.Equals("more than 100 wins"))
    {
        if (((PlayerData)playerList[i]).Wins > 100)
        {
            WriteToFile(strmWriter, playerList, i);
        }
    }
    // this will create a txt file for players who had a
favourable win to loss ratio
    if (cmbReportType.Text.Equals("By Favorouble w/l"))
    {
        if (((PlayerData)playerList[i]).Wins >
((PlayerData)playerList[i]).Losses)
        {
            WriteToFile(strmWriter, playerList, i);
        }
    }
    //this will create a txt file for players who are in the top 8
for their league
    if (cmbReportType.Text.Equals("If in top 8"))
    {
        if (((PlayerData)playerList[i]).LeagueStanding < 9)
        {
            WriteToFile(strmWriter, playerList, i);
        }
    }
    //increments the counter
    i++;
}
// will show the textfile with the specified report in
txtPlayerData.Show();
binReader.Close();
strmWriter.Close();
FileStream.Close();

//will check the report type selected and make the desired report

```

```

        if (cmbReportType.Text.Equals("more than 100 wins"))
        {
            WriteToForm(report);
        }
        if (cmbReportType.Text.Equals("By Favourable w/l"))
        {
            WriteToForm(report);
        }
        if (cmbReportType.Text.Equals("If in top 8"))
        {
            WriteToForm(report);
        }
    }
}
catch (IOException io)
{
    MessageBox.Show("invalid" + io.Message);
}
}

private void btnClear_Click(object sender, EventArgs e)
{
    //this will clear all the text boxes
    txtFirstName.Text = "";
    txtSurname.Text = "";
    txtAlias.Text = "";
    txtWins.Text = "";
    txtLosses.Text = "";
    txtApmAverage.Text = "";
    txtLeagueStanding.Text = "";
}

private void mnuSave_Click(object sender, EventArgs e)
{
}
//method for writeing to form
public void WriteToForm(string aReport)
{
    string inValue;
    try
    {
        using (StreamReader inFile = new StreamReader(aReport))
        {
            //will search through the file for each run and bring the text to
the textbox
            while ((inValue = inFile.ReadLine()) != null)
            {
                txtPlayerData.AppendText(inValue + "\n");
                txtPlayerData.SelectionStart = 0;
                txtPlayerData.ScrollToCaret();
            }
        }
    }
    catch (FileNotFoundException ex)
    {
    }
}

```

```

        MessageBox.Show("File not found " + ex.Message);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Unknown Error " + ex.Message);
    }
}
//method for the writing of the textfile and the formtting
public void WriteToFile(StreamWriter aStrmWriter,ArrayList aPlayerList,int i)
{
    try
    {
        // this will make the form look better by formatting the text into an
easily readable form
        aStrmWriter.WriteLine("-----");
        aStrmWriter.WriteLine(((PlayerData)aPlayerList[i]).Alias);
        aStrmWriter.WriteLine("-----");
        aStrmWriter.WriteLine("League: \t\t" +
((PlayerData)aPlayerList[i]).League + "\n" + "First Name: \t" +
((PlayerData)aPlayerList[i]).FirstName + "\n" + "Surname: \t" +
((PlayerData)aPlayerList[i]).Surname + "\n" + "Alias: \t\t" +
((PlayerData)aPlayerList[i]).Alias + "\n" + "Wins: \t\t" +
((PlayerData)aPlayerList[i]).Wins + "\n" + "Losses: \t\t" +
((PlayerData)aPlayerList[i]).Losses + "\n" + "Apm: \t\t" +
((PlayerData)aPlayerList[i]).Apm + "\n" + "League Standing: \t" +
((PlayerData)aPlayerList[i]).LeagueStanding);
    }
    catch (IOException io)
    {
        MessageBox.Show("IO Error" + io.Message);
    }
}
}
}

```

Player Data

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Starcraft2Ladder
{
    class PlayerData
    {
        // creating all the variables that will bew used in the tables
        string league = string.Empty;
        string firstName = string.Empty;
        string surname = string.Empty;
        string alias = string.Empty;
        Int32 wins=0;
        Int32 losses=0;
        Int32 apm=0;
        Int32 leagueStanding=0;
    }
}

```

```

public string League
{
    get { return league; }
}

public string FirstName
{
    get { return firstName; }
}

public string Surname
{
    get { return surname; }
}

public string Alias
{
    get { return alias; }
}

public int Wins
{
    get { return wins; }
}

public int Losses
{
    get { return losses; }
}

public int Apm
{
    get { return apm; }
}

public int LeagueStanding
{
    get { return leagueStanding; }
}

//constructor so that object of the playerdata can be made
public PlayerData(string aLeague, string aFirstName, string aSurname, string
aAlias,
    Int32 aWins, Int32 aLosses, Int32 aApm, Int32 aLeagueStanding)
{
    league = aLeague;
    firstName = aFirstName;
    surname = aSurname;
    alias = aAlias;
    wins = aWins;
    losses = aLosses;
    apm = aApm;
}

```

```
        leagueStanding = aLeagueStanding;
    }
}
```