using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace Project

{

public partial class Game2 : Form

{

private List<Artist> myArtistData = new List<Artist>();

private List<string[]> questionArtist = new List<string[]>();

private int count = 0;

public Game2()

{

InitializeComponent();

// Creates and gets question

Game2Controller myGame2Controller = new Game2Controller();

myGame2Controller.GenerateSongInfo();

myGame2Controller.GenerateQuestionOptions();

myArtistData = myGame2Controller.Game1RawInfo;

questionArtist = myGame2Controller.QuestionOptions;

}

private void textControl(int num)

{

Artist myArtist = new Artist();

myArtist = myArtistData[num];

// change Artist

if (myArtist.Song.Count() == 2)

{

Random random = new Random();

int randomnumber = random.Next(0, 2);

songNameLabel.Text = Convert.ToString(myArtist.Song[randomnumber]);

}

else

{

songNameLabel.Text = Convert.ToString(myArtist.Song[0]);

}

string[] tempArr = new string[3];

tempArr = questionArtist.ElementAt(num);

Random Randomnum = new Random();

int RandomNumb = Randomnum.Next(0, 3);

if (RandomNumb == 0)

{

radioButton1.Text = tempArr[0];

radioButton2.Text = tempArr[1];

radioButton3.Text = tempArr[2];

}

else if (RandomNumb == 1)

{

radioButton1.Text = tempArr[1];

radioButton2.Text = tempArr[0];

radioButton3.Text = tempArr[2];

}

else if (RandomNumb == 2)

{

radioButton1.Text = tempArr[2];

radioButton2.Text = tempArr[1];

radioButton3.Text = tempArr[0];

}

}

private void checkAnswer(int num)

{

Artist myArtist = new Artist();

myArtist = myArtistData[num];

bool bscore = true;//used to see correct answer

if (radioButton1.Checked && !myArtist.Name.Contains(radioButton1.Text))

{

bscore = false;

}

if (radioButton2.Checked && !myArtist.Name.Contains(radioButton2.Text))

{

bscore = false;

}

if (radioButton3.Checked && !myArtist.Name.Contains(radioButton3.Text))

{

bscore = false;

}

if (!radioButton1.Checked && !radioButton3.Checked && !radioButton2.Checked)

bscore = false;

radioButton1.Checked = false;

radioButton2.Checked = false;

radioButton3.Checked = false;

if (bscore)

GlobalVariables.activePlayer.IncScore();

}

public static void GameEnd(Form form)//for timer thread

{

frmIntermission newIntermission = new frmIntermission();

newIntermission.Show();

form.Hide();

}

private void game1BackBtn\_Click\_1(object sender, EventArgs e)

{

frmPreGame newPreGameForm = new frmPreGame();

newPreGameForm.Show();

this.Close();

}

private void game2song1btn\_Click(object sender, EventArgs e)

{

checkAnswer(count);

lblScore.Text = "Score: " + GlobalVariables.activePlayer.Score.ToString();

count++;

if (count <= 9)

{

textControl(count);

}

else if (count == 10)

{

frmIntermission newIntermission = new frmIntermission();

newIntermission.Show();

this.Close();

}

}

private void Game2\_Load\_1(object sender, EventArgs e)

{

count = 0;

textControl(count);

game2TimerLable.Text = "Time: 30";

TimerClass timer = new TimerClass(game2TimerLable, this);

timer.timerThread = new Thread(timer.timerMethod);

timer.timerThread.Start();

lblScore.Text = "Score: " + GlobalVariables.activePlayer.Score.ToString();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Project

{

class Game2Controller

{

// Artist Information

private List<Artist> game1RawInfo = new List<Artist>();

private int score = 0;

private string[] allArtists = new string[] { "A.K.A Forbes", "Alex Baker", "Anderson .Paak", "Beyonce", "Blac Youngsta", "Black Coffee", "Calvin Harris", "Childish Gambino", "Ctrlgang", "Paul Rudder", "Buzzcocks", "The Cramps", "The Jam" };

private List<string[]> questionOptions = new List<string[]>();

public List<Artist> Game1RawInfo { get => game1RawInfo; set => game1RawInfo = value; }

public int Score { get => score; set => score = value; }

public string[] AllArtists { get => allArtists; } //Read Only

public List<string[]> QuestionOptions { get => questionOptions; set => questionOptions = value; } // hold the 3 options for catagories in game1

public Game2Controller()

{

}

public void GenerateSongInfo()

{

// Gets question information on application start --- could use threading

Game2GenerateQuestions myList = new Game2GenerateQuestions();

Game1RawInfo = myList.RandomList;

}

public void GenerateQuestionOptions()

{

var RNG = new Random();

var RNG\_AllArtists = new Random();

var RNG\_CurrentSongArtists = new Random();

Artist tempSong = new Artist();

foreach (Artist item in Game1RawInfo)

{

tempSong = item;

int lenght = tempSong.Song.Count;

string[] tempStringArr = new string[3];

tempStringArr[0] = tempSong.Name + " " + tempSong.Surname;

tempStringArr[1] = AllArtists[RNG\_AllArtists.Next(9)];

//Ensures no duplicates

while (tempStringArr[0] == tempStringArr[1])

{

tempStringArr[1] = AllArtists[RNG\_AllArtists.Next(9)];

}

tempStringArr[2] = AllArtists[RNG\_AllArtists.Next(9)]; // -- replace with actual genre

while ((tempStringArr[0] == tempStringArr[2]) || (tempStringArr[1] == tempStringArr[2]))

{

tempStringArr[2] = AllArtists[RNG\_AllArtists.Next(9)];

}

tempStringArr.OrderBy(x => RNG.Next(3)); // Randomisez order

QuestionOptions.Add(tempStringArr);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Project

{

class Game2GenerateQuestions

{

private List<Artist> randomList = new List<Artist>();

private int listSize = 10;

public List<Artist> RandomList { get => randomList; set => randomList = value; }

public int ListSize { get => listSize; }

//Default Constructor

public Game2GenerateQuestions()

{

ArtistData myArtistData = new ArtistData();

List<Artist> tempList = myArtistData.AllArtistData;

//Sets size

setListSize(tempList, ListSize);

//Randomize methode

RandomList = randomizeList(tempList);

}

//Randomizes inputed List

public List<Artist> randomizeList(List<Artist> myList)

{

var RNG = new Random();

int size = myList.Count;

var randList = myList.OrderBy(Artist => RNG.Next(size));

myList = new List<Artist>();

foreach (var item in randList)

{

myList.Add(item);

}

return myList;

}

public void setListSize(List<Artist> myList, int ListSize)

{

foreach (Artist newArtist in myList)

{

int count = 0;

if (count < ListSize)

{

count++;

RandomList.Add(newArtist);

}

}

}

}

}