

УДК 004.93.1

СОЗДАНИЕ МУЗЫКАЛЬНЫХ ДОРОЖЕК ИЗ ГИТАРНЫХ АККОРДОВ

Охремук Д.А.,

Научный руководитель – Разоренов Н.А., к.т.н., доцент

Для получения приложения способного обучать на гитаре необходимо создать комплекс уроков, таблицу аккордов, электронный дневник в котором будут находиться песни для обучения, а также создание музыкальных дорожек.

Для создания музыкальных дорожек сначала нужно подключить необходимые библиотеки, которые понадобятся для правильной работы. Сначала нам понадобится библиотека NAudio[1] которая будет воспроизводить кусок файла с определенной позиции определенной длины далее мы подключим библиотеку WMPLib[2] которая будет воспроизводить все собранные файлы вместе и выводить звук. Необходимые библиотеки приведены на рисунке 1.

```
using WMPLib;  
using NAudio.Wave;
```

Рисунок 1 – Необходимые библиотеки для работы

Сначала нам необходимо определиться как будет выглядеть интерфейс нашего приложения для этого я создам макет где будет показан будущий интерфейс. Макет представлен на рисунке 2.

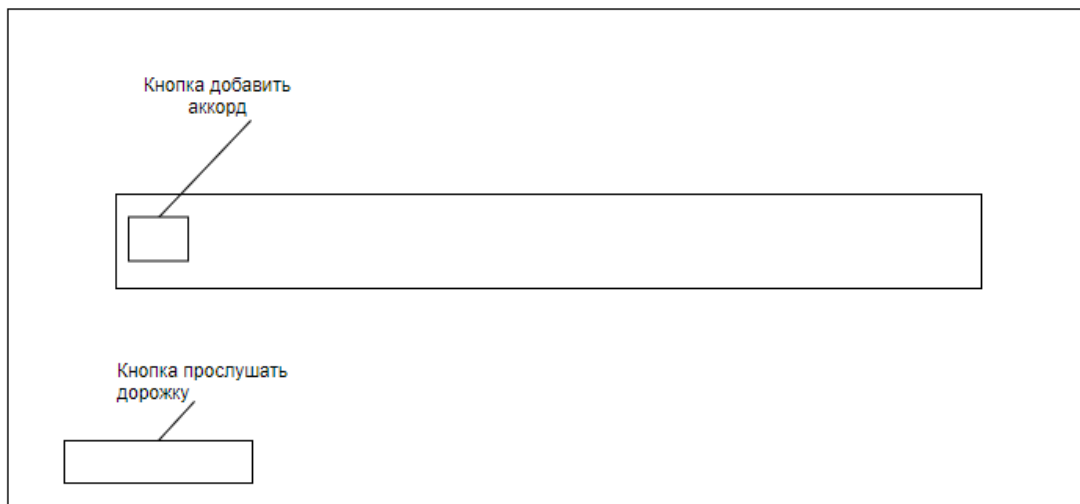


Рисунок 2 – Макет будущего интерфейса

Теперь можно приступить к разработке основного функционала приложения.

На форме имеется кнопка для добавления аккорда, ноты в звуковую дорожку. Событие нажатия на кнопку «Добавить (+)», на главной форме представлена на рисунке 3.

```
private void bAdd_Click(object sender, EventArgs e)
{
    Selected_path = string.Empty;
    Selected_name = string.Empty;
    fChords f = new fChords(this);
    f.ShowDialog();
    if (Selected_name == "" || Selected_path == "") return;
```

Рисунок 3 – Событие нажатия на кнопку «Добавить (+)»

Сначала происходит обнуление переменных Selected_path и Selected_Name, в которые заносятся путь к мелодии и наименование аккорда. Затем объявляется и инициализируется экземпляр класса Chord. Объектом класса Chord является форма с аккордами. В конструктор данного класса передается текущая форма. Это необходимо для заполнения переменных Selected_path и Selected_name.

После этого вызывается метод ShowDialog, который открывает форму с аккордами в диалоговом режиме. Это значит, что нельзя будет работать с какой-либо другой формой, не закрыв форму f. Если после закрытия формы строковые переменные (не было выбрано аккорда), назначения которых написано выше не были заполнены, тогда выполнение данного события прекращается ключевым словом return. Но, а если пользователь выбрал аккорд, тогда выполняется следующий код, показанный на рисунке 4.

```
mas[mas.Length - 1] = new Button();
mas[mas.Length - 1].Parent = panel1;
mas[mas.Length - 1].Text = Selected_name;
mas[mas.Length - 1].Width = bAdd.Width;
mas[mas.Length - 1].Height = bAdd.Height;
mas[mas.Length - 1].MouseDown += MouseClicks;
mas[mas.Length - 1].Tag = Selected_path;
SetLocation();
Array.Resize<Button>(ref mas, mas.Length + 1);
```

Рисунок 4 – Пользователь выбрал аккорд

При выборе аккорда инициализируется новая кнопка. После инициализации меняются ее свойства: указывается контейнер в котором она лежит, надпись, размеры. После этого данная кнопка привязывается к событию MouseDown[3], которое происходит при клике мышкой по ней. Так же ее свойству Tag присваивается путь к мелодии.

После установки всех необходимых свойств, выполняется функция `SetLocation`, которая задает расположения всего массива кнопок `mas`. После того, как кнопка добавлена и помещена на панель возникает необходимость увеличить размер массива. Это необходимо для добавления еще одной кнопки. Делается это статическим методом `Resize`, который находится в классе `Array`.

Описание кода из функции `MouseClicks[3]`, которая подключена к событию `MouseDown`.

А теперь рассмотрим, что происходит при правом или левом клике мышкой по созданной кнопке с аккордом. Код из этой функции можно условно разделить на две части.

Первая часть выполняется, когда кнопка нажата левой кнопкой мыши продемонстрировано на рисунке 5:

```
private void MouseClicks(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButton.Left)
    {
        wmp.URL = (string)(sender as Button).Tag;
        wmp.controls.play();
    }
}
```

Рисунок 5 – Нажата левая кнопка мыши

В данном отрывке кода в блоке `if` проверяется какой кнопкой мыши была нажата кнопка и если она нажата ЛКМ, то происходит следующее: задается путь к песни для объекта `wmp`, а после этого запускается проигрывание мелодии с помощью библиотеки `WMPLib[2]` которая подключалась выше.

Вторая часть кода рассматриваемой нами функции выполняется, когда нажата ПКМ можно увидеть на рисунке 6:

```
if (e.Button == MouseButton.Right)
{
    if (MessageBox.Show("Вы действительно хотите удалить этот аккорд?", "Внимание!"))
    {
        wmp.controls.stop();
        wmp.URL = "";
        (sender as Button).Tag = "del";
        (sender as Button).Visible = false;
        SetLocation();
    }
}
```

Рисунок 6 – Нажата правая кнопка мыши

После перехода в данный блок `if` пользователю показывается сообщение с вопросом об удалении выбранного аккорда, если пользователь нажмет да, то выполниться следующее: воспроизведение мелодии остановится, кнопка будет помечена, как удаленная, затем она скроется с экрана. А после этого произойдет вызов функции `SetLocation`, которая заново поставит кнопки по необходимым координатам.

Литература

1. NAudio [Электронный ресурс]. Режим доступа: <https://github.com/naudio/NAudio>, свободный, - Загл. с экрана. – Язык русский, английский.
2. WMPlib [Электронный ресурс]. Режим доступа: [https://msdn.microsoft.com/en-us/library/windows/desktop/dd564585\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd564585(v=vs.85).aspx), свободный, - Загл. с экрана. – Язык русский.
3. Control.MouseClick [Электронный ресурс]. Режим доступа: [https://msdn.microsoft.com/ru/library/system.windows.forms.control.mouseclick\(v=vs.110\).aspx](https://msdn.microsoft.com/ru/library/system.windows.forms.control.mouseclick(v=vs.110).aspx), свободный, - Загл. с экрана. – Язык русский.