

# ЭЛЕКТРОННОЕ УЧЕБНОЕ ПОСОБИЕ ПО ПРОГРАММИРОВАНИЮ АНИМИРОВАННЫХ ИЗОБРАЖЕНИЙ

*Ахметшина Л.Н.  
Волкова Т.И., к.п.н., доцент  
г. Бирск, ФГБОУ ВО Бирский филиал БашГУ*

В настоящее время происходит активное развитие информационных технологий, в том числе компьютерной графики и анимации. Для пользователей сети Интернет уже давно стали привычными появляющиеся почти на каждой Web-странице «живые» мультипликационные изображения. Также, ни для кого не секрет, что видео игры, использующие всевозможные техники компьютерной анимации, прочно заняли свою позицию в современной индустрии развлечений. Компьютерная анимация получила широкое применение не только в сферах производства и развлечений, но также и в научной и деловой сферах. Поэтому, сегодня достаточно актуальна *проблема* обучения и подготовки специалистов в области компьютерной графики и анимации.

В процессе решения данной проблемы было разработано электронное пособие по программированию анимаций на языке С#.

Структура пособия:

1. **Теория**
  - 1.1. Анимация
  - 1.2. Анимация в С#
  - 1.3. Виды анимации
    - 1.3.1. Покадровая анимация
    - 1.3.2. Летающая анимация (анимация перемещения)
2. **Практика**
  - 2.1. Покадровая анимация
  - 2.2. Летающая анимация (анимация перемещения)
3. **Тест** (контроль полученных знаний)

Рассмотрим подробнее каждый раздел:

## а) **Теория**

Данный раздел разбит на подразделы: анимация, анимация в С#, виды анимации: покадровая анимация, летающая анимация (анимация перемещения). В каждом подразделе имеются демонстрации, служащие интерактивными подтверждениями теоретического материала.

Первый подраздел – Анимация – представляет собой общую теорию по анимации [1]. Здесь даны главные определения анимации в целом, а также

определения компьютерной анимации. Как пример анимации в данном подразделе имеется анимация «лошадь в движении», демонстрирующая анимацию с последовательной сменой кадров.

Второй подраздел – это непосредственно теория о возможностях Visual Studio по созданию анимации на ЯПВУ С#. Теория представляет собой описание классов, их методов и свойств, необходимых для создания анимации в С#. Для закрепления теории здесь имеется интерактивная подпрограмма, демонстрирующая следствия изменения свойств и действия методов. В подпрограмме имеется анимация «дракона», которой можно манипулировать (изменять скорость анимации, перемещать анимацию по форме, останавливать и воспроизводить анимацию).

Третий подраздел представлен двумя пунктами – видами анимации. Из различных видов анимации были выбраны два: покадровая анимации и анимация перемещения (летающая анимация). Данные виды анимации позволяют наглядно продемонстрировать возможности Visual Studio по созданию простой анимации в С#.

Первый пункт – покадровая анимация. В этом пункте даются основные определения данного вида анимации [2]. И приводится пример непосредственной реализации покадровой анимации на языке С# в коде. Также в данном пункте имеется интерактивная анимация, наглядно иллюстрирующая реализацию приведенного примера.

Второй пункт – анимация перемещения (или летающая анимация). В данном пункте, также как и в предыдущем, даются определения рассматриваемого вида анимации, представлена интерактивная подпрограмма, демонстрирующая данный вид анимации. Данная подпрограмма моделирует движение объектов в замкнутой системе по закону всемирного тяготения, в ней реализовано движение планет солнечной системы. В программе предусмотрены: изменение скорости движения системы планет, изменение масштаба. Далее в этом пункте рассматривается программный код реализации простого примера анимации на языке С#. Пример представляет собой также реализацию движения планет, но в данном случае используется упрощенная версия реализации (использование численных методов). Здесь также рассмотрен программный код примера.

#### **в) Практика**

Практика – это задание по каждому из видов анимации (покадровой и летающей).

Задания требуют написания фрагмента кода пользователем. Пользователю предоставляется текст условия задачи. Код программы, который необходимо дописать, пользователь вводит в соответствующее поле.

Далее идет обработка кода, введенного пользователем, с последующим анализом результата.

Если в процессе выполнения кода программы возникают ошибки, то пользователь получает по ним сведения. Также предоставлена возможность вводить код сколько угодно раз. Если же обучаемый не может справиться с данным заданием, то он может просмотреть правильный вариант написания кода.

### с) Тест

В тесте реализованы задания разных типов: краткий ответ, на выбор, на соответствие, и на исправление ошибки в представленном коде.

Первый тип – краткий ответ – это непосредственно текст вопроса и поле для ввода ответа, пользователь вводит ответ с клавиатуры. Вопрос определяется рандомным способом. Тексты вопросов и правильные ответы на них находятся в файле.

Второй тип – на выбор – представляет собой текст вопроса и четыре предложенных варианта ответов, пользователь может выбрать один из вариантов. Текст вопроса и ответы на него изменяются случайным образом.

Третий тип – на соответствие – четыре пункта, с которыми нужно соотнести предоставленные четыре элемента. Пользователь должен правильно соотнести данные элементы соответствия.

Четвертый тип – на исправление ошибки в представленном программном коде. Пользователю дается текст кода программы, в котором допущена логическая ошибка, необходимо найти и исправить эту ошибку. Коды программ также выбираются случайным образом из текстового файла.

### ***Интерфейс и структурная схема приложения***

Программа разработана в среде разработки Microsoft Visual Studio. Проект – приложение WindowsForms. Структурная схема приложения представлена на рисунке 1:



Рис. 1. Структурная схема приложения

Интерфейс составляют семь форм: три основные формы: «Главная», «Практика», «Тест» и четыре дополнительные формы: «Приветствие», «Титульная», «О программе», «Руководство управления анимацией «Дракон»». Запуск программы сопровождается анимационным приветствием, за которым следует отображение титульной формы. С титульной формы происходит переход на главную форму обучающей программы.

На форме «Приветствие» реализуются две анимации: «Бабочка» и «Дракон». Интерфейс форм реализован на основе свойств прозрачности самой формы и ее элементов.

На форме «Главная» находятся три надписи-ссылки, реализующие переход на три главных модуля программы: теория, практика, тест. Также на данной форме имеется меню программы, представленное пунктами «Навигация» и «Справка».

Для подразделов теории переход реализован последовательно из раздела к разделу. Информация каждого из подразделов расположена на отдельной панели. Все панели находятся на форме «Главная». Текстовая часть теории каждой из панелей размещена на элементах управления *RichTextBox*. Переход между панелями осуществляется кнопками (*Button*) «Далее» и «Назад».

На панелях «Теория анимации», «Теория покадровой анимации» и «Пример покадровой анимации» после клика по изображениям начинается демонстрация анимации. Для каждого такого изображения предусмотрена «всплывающая подсказка» (элемент управления *ToolTip*).

С панели «Теория анимации в C#» вызывается подпрограмма «Дракон», реализованная в виде картинка, которую можно перемещать по форме с помощью кнопок-стрелок, а также изменять скорость анимации, изменяя значение таймера *Interval* (рис. 2).

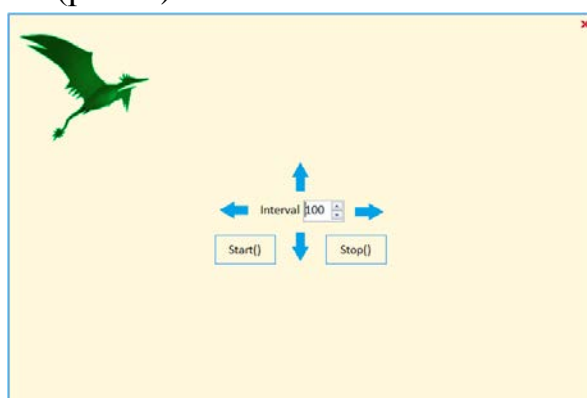


Рис. 2. Подпрограмма «Дракон»

Для вызова данного приложения используется класс *Process*, находящийся в пространстве имен *System.Diagnostics*, который создает и запускает процесс приложения и ждет его завершения.

Код вызова exe-приложения:

```
public static string file_exe =  
Environment.GetFolderPath(Environment.SpecialFolder.InternetCache) + "Dragon.exe";  
private void buttonDemonstationAnimationDragon_Click(object sender, EventArgs e)  
{  
    FileStream fs = new FileStream(file_exe, FileMode.Create);  
    fs.Write(Properties.Resources.DemonstrationDragonAnimation, 0,  
Properties.Resources.DemonstrationDragonAnimation.Length);  
    fs.Close();  
    Process process = new Process();  
    process.StartInfo.FileName = file_exe;  
    process.Start();  
    process.WaitForExit();  
}
```

Таким же способом реализован вызов второго приложения – «Вселенная» (рис.3) на панели «Теория летающей анимации».

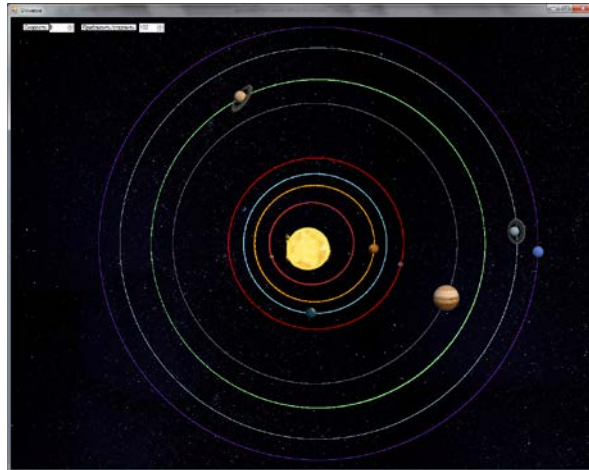


Рис. 3. Подпрограмма «Вселенная»

Раздел практики представлен на отдельной форме ( рис. 4).

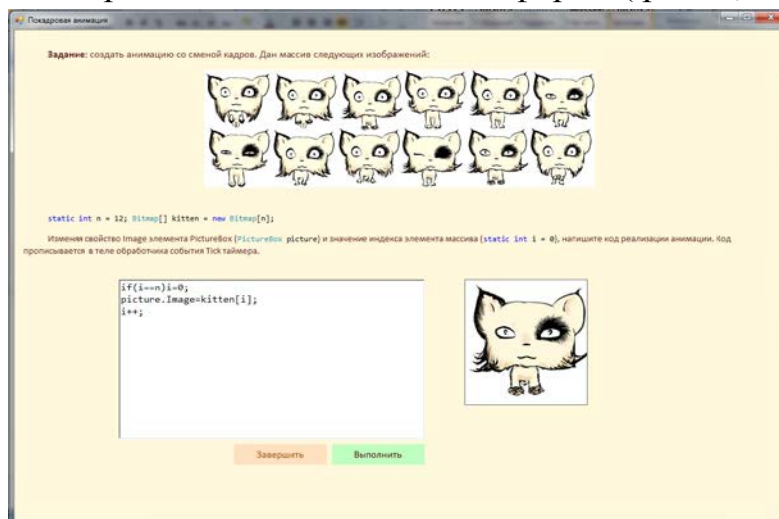


Рис. 4. Форма «Практика», покадровая анимация

Практика интерактивного обучающего приложения реализована в виде задания по каждому виду анимации. Переход на каждое задание осуществляется в форме «Главная», по надписям-ссылкам (*LinkLabel*). На каждый вид анимации дано по одному заданию на создание кода программы, реализующей определенный вид анимации. На форме имеется элемент управления *RichTextBox* для текста условия задачи, текстовое поле для ввода кода, кнопки (*Button*) «Выполнить» и «Завершить» для управления компиляцией.

В программной реализации практического раздела использована динамическая компиляция кода.

Необходимые пространства имен:

```
using System.CodeDom.Compiler;
using Microsoft.CSharp;
using System.Diagnostics;
using System.Reflection;
```

Для создания компилятора используется класс *CSharpCodeProvider*. Для задания параметров компилятора – класс *CompilerParameters*. Результат компиляции – класс *CompilerResults*.

В программе введенный пользователем код, собирается в процессе компиляции в dll-сборку. Статистика ошибок выводится в соответствующее для этого поле. Если компиляция прошла успешно, то в обработчике события *Tick* вызывается метод из скомпилированной сборки:

```
Assembly assembly = result.CompiledAssembly;
type = assembly.GetType("Animation.MotionAnimation");
MethodInfo method = type.GetMethod("StartAnimation");
try
{
    method.Invoke(null, objectSA);
}
catch (Exception exp) // обработка непредвиденных исключений
{
    .....
    richTextBox3.Text = "Исключение : " + exp.Message + '\n' + exp.InnerException + '\n';
}
```

Тестовый раздел реализован на отдельной форме. Каждый тип вопроса находится на отдельной панели (*Panel*). Имеется четыре панели: «Краткий ответ», «На выбор», «На соответствие», «Исправление ошибки в коде». Переход к следующему вопросу происходит по кнопке «Далее». Текст вопроса загружается в элемент управления *Label*. При каждом ответе пользователя и нажатии на кнопку (*Button*) «Далее» происходит запоминание

ответа, сравнение его с правильным ответом, определение и загрузка параметров для следующего вопроса.

1) Для типа вопроса «Краткий ответ» вопросы загружаются из файла, и подбираются случайным образом.

2) «На выбор» представляет собой задание расчетного характера. Значения задания и вариантов ответов создаются случайно в процессе работы программы.

3) Соотнесение вариантов соответствия (вопрос «На соответствие») организовано с помощью создания дополнительного элемента *TextBox* (по клику по выбранному варианту соответствия), в который переносится текст из выбранного варианта, и передвижения его по экрану в соответствии с перемещением компьютерной мыши. После клика по месту расположения соответствия дополнительный *TextBox* удаляется, а текст из него переносится на выбранное место (элемент *TextBox*).

4) Вопрос «Исправление ошибки в коде» организован сравнением исправленного пользователем кода с правильным ответом. Правильные ответы и код для исправления извлекаются из файла и подбираются случайно.

На последнем этапе идет расчет результатов ответов пользователя в процентах от количества вопросов и вывод результата.

#### Литература

1. Компьютерная анимация [Электронный ресурс] // Википедия — URL: [https://ru.wikipedia.org/wiki/Компьютерная\\_анимация](https://ru.wikipedia.org/wiki/Компьютерная_анимация) (дата обращения 16.01.2016).
2. Мультипликация (значения) анимация [Электронный ресурс]// Википедия. - URL:<https://ru.wikipedia.org/wiki/Мультипликация> (дата обращения 18.01.2016). .