И. А. Буяковская

ОРГАНИЗАЦИЯ ЗАНЯТИЙ ПО ТЕМЕ «ГРАФИКА И ДВИЖЕНИЕ» В СРЕДЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ GAMBAS В ДИСЦИПЛИНЕ «ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ»

Дисциплина «Информатика и программирование» входит в базовую часть цикла программы бакалавриата по направлению «Прикладная информатика» профиль «Прикладная информатика в образовании» с кодом (Б2.Б) и изучается на 1 курсе в 1,2 семестрах. В раздел «Программирование» данной дисциплины включены следующие темы:

- Объектно-ориентированное программирование. Понятие класса. Инкапсуляция, наследование, полиморфизм. Свойства, методы, события. Объектно-событийное и объектно-ориентированное программирование.
- Алфавит и синтаксис языка программирования Gambas. Реализация основных алгоритмов средствами языка Gambas.
- Изучение внутренней структуры модулей, создание модуля и подключение его к проекту. Многооконные приложения. Создание модальных и немодальных окон в среде Gambas.
- Графические возможности Gambas. Функции и процедуры пользователя. Программирование рекурсивных алгоритмов.
- Массивы и их обработка.
- Строки. Файлы. Типы файлов. Различия в доступе. Чтение и запись данных.
- Организация динамических структур данных (абстрактных типов данных).

В рамках темы «Графические возможности Gambas» изучаются приемы создания графических примитивов с применением циклов и подпрограмм, построение графиков функций, применение рекурсии на графике и создание движущихся объектов.

Рассмотрим на примере темы «Графика и движение» какими заданиями представлена фронтальная и индивидуальная работа студентов.

Перед решением типовой задачи необходимо актуализировать знания по данной теме, полученные в предшествующих темах дисциплины и связанных с опорными понятиями: свойства, методы, события, ветвление, подпрограмма в языке Gambas.

Для создания пользователем собственных анимационных эффектов применяется принцип смены кадров (изображений), как это выполняется в мультипликации. Программа, имитирующая движение, должна реализовывать следующие этапы:

- Создание изображения в области рисования;
- Реализация временной паузы для того, чтобы глаз зафиксировал изображение;
- Проведение коррекции изображения.

В учебном пособии Быкова В.Л. [1] описываются основные способы создания анимационных эффектов. При этом можно использовать различные способы и возможности языка программирования Gambas:

- применение элемента управления Animation и его методов;
- пересчет координат объекта и использование свойств Top и Left объекта для переопределения координат объекта;
- использование организации пауз и применение элементов управления Image и PictureBox;
- использование буфера обмена;
- прямое присвоение значений свойств одного графического объекта другому.
- Общий алгоритм работы программы при создании анимации следующий:
- воспроизвести изображение в начальной точке;
- сделать паузу на некоторое время, достаточное для фиксации изображения. Длительность паузы, когда изображение на экране неподвижно определяет и скорость перемещения объекта;
- стереть изображение;
- пересчитать координаты объекта и воспроизвести его в новой позиции.
- Таким образом, важной составляющей в организации процесса движения является замедление объекта. Это можно выполнить двумя способами: уменьшением шага переноса и организацией пауз.

Для организации пауз можно использовать различные приемы: использование пустых циклов, использование системных часов, использование таймера.

Представим пример фронтальной задачи на создание анимации в среде Gambas с использованием специального объекта - таймер.

Основные свойства таймера: **Intrval** и **Enabled**. Основное событие - **Timer**. **Delay** - позволяет установить интервал выдачи сигнала. Enabled - позволяет запускать и останавливать таймер. Если Enabled равно True, то таймер запускается. При установке значения свойства Enabled в False таймер останавливается.

Событие **Timer** используется для размещения текста программы, которая должна выполняться через заданный интервал времени.

Задание 1. Требуется создать простейшую анимацию, создающую эффект перемещения по форме изображения на основе четырех кадров. Для этого проекта нам потребуется объект PictureBox, он находится на закладке (Form). Свойство Picture этого объекта, служит для помещения в него изображения. Кадры для анимации с расширением *.bmp добавлены в папке проекта. Итак, размещаем на форме 4 объекта PictureBox (одинаковых размеров). Предварительно загружаем кадры анимации в свойство объекта Picture.

На форме также разместим две кнопки Button: одна будет запускать анимацию кадров, а вторая необходима для организации выхода из проекта. Так как наши изображения загружены предварительно, то воспользуемся процедурой, которая выполняется при запуске проекта на исполнение (см. рис. 1). В данной процедуре объектам PictureBox зададим свойство Hide, позволяющее скрыть данные объекты при запуске приложения.

```
PUBLIC i AS Integer

PUBLIC SUB _new()
  PictureBox1.hide
  PictureBox2.hide
  PictureBox3.hide
  PictureBox4.hide
END
PUBLIC SUB Button2_Click()

  ME.Close

END
```

Рисунок 1. Листинг процедуры выхода и процедуры _new

Как показано на рисунке 1 зададим глобальную переменную i в которой будет храниться номер текущего кадра.

На форме разместим таймер, у которого изменим свойство Delay, отвечающее за задержку времени между кадрами, чем выше данное значение, тем медленнее воспроизводятся кадры анимации. Установим в данном свойстве значение 400. Изменим также свойство Enabled на False, отвечающее за включение и выключение таймера, так как таймер должен включаться по нажатию на кнопку.

В процедуре Button1_Click() задаем исходное значение счетчика, равное нулю и включаем таймер (см. рисунок 2). В процедуре включения таймера увеличиваем счетчик на 1. Далее используя, условный оператор последовательно показываем текущий кадр, используя соответствующее свойство объекта и скрывая остальные изображения. При достижении счетчика кадра значения 4 выключаем таймер.

```
PUBLIC SUB Timer1_Timer()
  i = i + 1
  IF i = 1 THEN
  PictureBox1.Show
  PictureBox2.hide
  PictureBox3.hide
  PictureBox4.hide
  ENDIF
  IF i = 2 THEN
  PictureBox1.hide
  PictureBox2.Show
  PictureBox3.hide
  PictureBox4.hide
  ENDIF
  IF i = 3 THEN
  PictureBox1.hide
  PictureBox2.hide
  PictureBox3.Show
  PictureBox4.hide
  ENDIF
  IF i = 4 THEN
  PictureBox1.hide
  PictureBox2.hide
  PictureBox3.hide
  PictureBox4.Show
  ENDIF
 IF i = 4 THEN Timer1.Enabled = FALSE
END

PUBLIC SUB Button1_Click()
  i = 0

  Timer1.Enabled = TRUE
END
```

Рисунок 2. Листинг процедуры таймера и процедуры для кнопки «Запуск»

Другой способ анимации состоит в присваивании значения свойства Picture одного графического элемента управления другому. Рассмотрим данный вариант реализации на примере следующей задачи.

Задание 2. Зададим анимацию с перемещением объекта на основе четырех кадров.

Вначале располагаем на форме 5 объектов PictureBox, в которые последовательно загружаем 4 кадра в PictureBox1 - первый кадр, в PictureBox2 - 2 кадр (свойство Proportional позволит автоматически подгонять размеры картинки под размеры объекта на форме), в PictureBox3 - 3 кадр, PictureBox4 - 4 кадр. Объект PictureBox5 остается пустым. В него мы будем последовательно загружать наши кадры анимации. Далее расположите объекты один над другим. Добавьте две кнопки: «Анимация» и «Выход», а также таймер (см. рисунок 3).

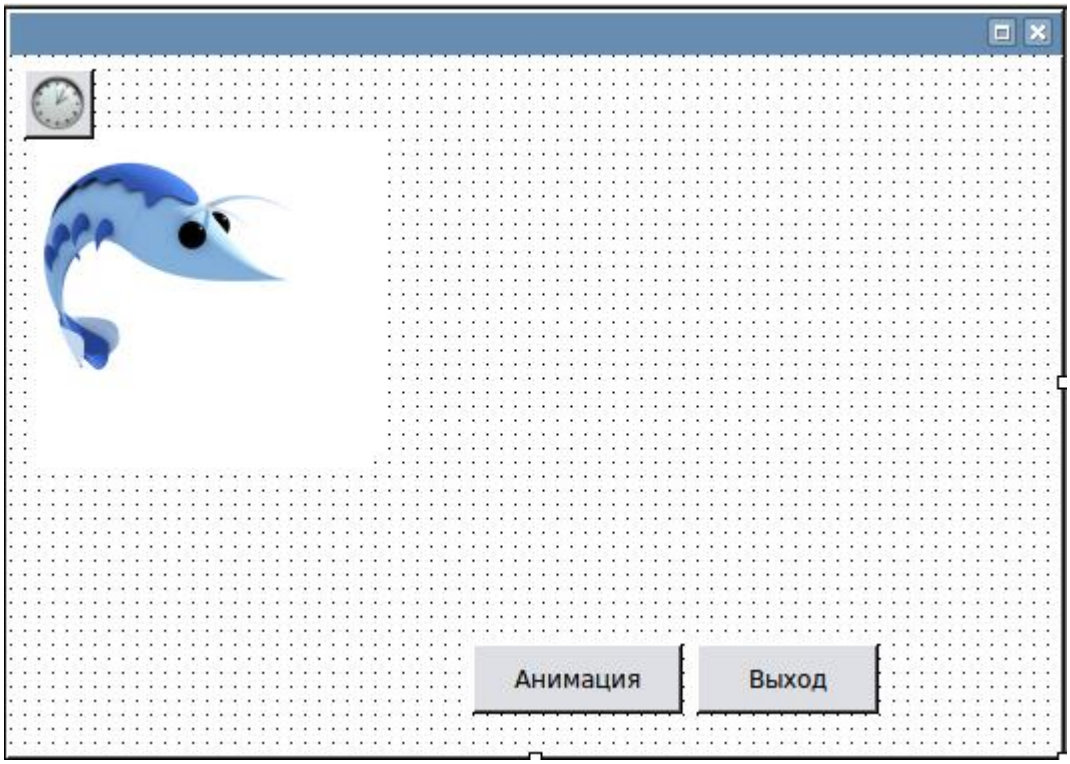

Рисунок 3. Форма проекта

Объект PictureBox5 с течением времени должен перемещаться по форме и в него последовательно будут загружаться картинки из объекта PictureBox1, потом из PictureBox2, PictureBox3 и PictureBox4, затем снова из PictureBox1 и т.д. Загружать картинки из одного объекта в другой можно так: PictureBox5.Picture:=PictureBox1.Picture;

Объявим глобальную переменную i, которая будет отвечать за номер кадра. Остановка анимации произойдет после того как мы дойдем до края формы. Данную проверку мы осуществим в условном операторе, посчитав сумму измерения левого края кадра и его ширины. Полученное значение не должно превосходить ширину формы. Чтобы задать эффект перемещения необходимо после отображения кадра увеличить атрибут Left на 10. Для того, чтобы исходные кадры не отображались при запуске проекта, в процедуре _new, зададим свойству Visible (видимый) значение False.

Ниже представлен листинг программного кода к данной задаче:

```
PUBLIC i AS Integer

PUBLIC SUB _new()
PictureBox1.Visible = FALSE
PictureBox2.Visible = FALSE
PictureBox3.Visible = FALSE
PictureBox4.Visible = FALSE
END

PUBLIC SUB Timer1_Timer()
i = i + 1
IF PictureBox5.Left + PictureBox5.Width <= FMain.Width THEN
 IF i MOD 4 = 3 THEN PictureBox5.Picture = PictureBox1.Picture
 IF i MOD 4 = 2 THEN PictureBox5.Picture = PictureBox2.Picture
 IF i MOD 4 = 1 THEN PictureBox5.Picture = PictureBox3.Picture
 IF i MOD 4 = 0 THEN PictureBox5.Picture = PictureBox4.Picture
 PictureBox5.Left = PictureBox5.Left + 10
ELSE
Timer1.Enabled = FALSE
ENDIF
END

PUBLIC SUB Button1_Click()
  i = 0
  Timer1.Enabled = TRUE

END

PUBLIC SUB Button2_Click()

  ME.Close

END
```

Рисунок 4. Листинг программного кода

Также в рамках фронтальной лабораторной работы, рассматриваются способы применения элемента управления Animation, задание анимации посредством переноса изображений через буфер обмена с использованием методов SetData, GetData( ), GetForm, Clear). Рассматривается задача, основанная на перерисовке объекта с изменением координат. Например: изобразите на экране точку, движущуюся по окружности.

После решения фронтальных задач, студентам предлагаются индивидуальные задания, примеры которых приведены в учебном пособии «Введение в структурное программирование» [2] приведем несколько примеров:

1. В рисованных мультфильмах иллюзия движения создается последовательной сменой кадров, каждый из которых фиксирует очередное положение движущегося объекта. Используя этот принцип, получить на экране мультфильм, показывающий идущего человечка.

2. Изобразить на экране правильный треугольник, вращающийся в плоскости экрана вокруг своего центра.

Список литературы

1. Быков В. Л. Основы программирования на языке Visual Basic 6.0: Учебное пособие - Брест: БГТУ, 2002. - 229 с.

2. Можаров М.С., Бойченко Г.Н. Введение в структурное программирование: (Учебное пособие. Рекомендовано Учебно-методическим объединением по образованию в области прикладной информатики в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальности 080801 «Прикладная информатика (в юриспруденции)» и другим экономическим специальностям)//Новокузнецк: КузГПА, 2009. -Ч.1 - 2. - 238 с.