CSC470 Computer Graphics
Table of Contents

I Course Organization
II Assignments
III Midterm Project
IV Final Project
V Visual BASIC tutorial: A paint program
VI Trispectives: A full featured 3D modeling tool.

CSC470 Computer Graphics
Course Outline

Instructor: Dr. Walter Johnson, Jr.
Office; Graphics Building #3
Phone: 546-8383
e-mail: wjohnson@shawu.edu

Class location: 107 TUP
Time: 2:15-3:45

Course Objectives:
- To study the theory of fundamental methods used in computer graphics
- To explore some advanced methods and current research problems in computer graphics
- To gain practical in developing event-driven programs, designing graphical user interfaces, and developing useful graphics programs.

Ethical Responsibilities: Students are expected to demonstrate ethical policies and practices in the course. Cheating will not be tolerated (zero Tolerance) If caught, you will receive a zero.

Course Organization:

The course will be organized by two themes: theory and practice. The theory component will be presented in a traditional lecture-based classroom format. In the first half of the course we will cover fundamentals as well as propose some possible research topics. We will follow this outline:

1. Basic methods of raster graphics
In this course we will consider some advanced topics in rendering, modeling, scientific visualization, and computer animation. Some of the material presented in the course will be dictated by your interests, as determined by the projects you choose to work on.

The purpose of this course’s practical component will be to introduce you to two program development environment with relative shallow learning curves. It will be possible for you to develop a sophisticated project using either of these tools without prior experience, in the relatively short time frame of this course. If you are already familiar with a more complex development environment such as C++ you are welcome to work with such a system.

The two development environments we will look into are Visual BASIC and Trispective. Visual BASIC and Trisective both run on PC’s. Both tools support event-driven programming and specification of graphical user interfaces through a visual programming environment. Moreover, both provide relatively powerful programming languages: Visual BASIC’s programming language is a much-enhanced version of BASIC. For this class we will enjoy a hands-on tutorial of Visual BASIC and a demonstration of Trispective a 3D application.

Projects
Two programming projects will be required for the course. The first will involve the implementation of a simple paint program. This first project will also provide you with a chance to work with a programming environment such as Visual BASIC.

The second programming project will be less confining. You choose from a number of topics. Here are some examples:
- A drawing program
- A ray tracer
- An interesting game
- Several hidden surface removal methods
- A modeling system (eg. CSG, or a Bezier surface editor)

A detailed handout describing each of these projects is included in this booklet.

Course Requirements
Class participation (20% of total grade)
Midterm Programming project (40%)
Final Programming project (40%)

A Word of Caution
In a course like this, which mixes theory and programming, students often have a tendency to spend too much time programming and too little time on theory. The theory is an important element of the course. Please keep up with the material covered in lectures.

Introduction to Visual BASIC

- What is Visual BASIC
- Elements of the opening screen
- Event-driven programming

Developing a Paint Program in Visual BASIC

- Creating a Canvas
- Controls for Drawing Lines and Circles
- Adding a Message Bar
- A Control for Clearing the Picture
- A Control for Changing the Pen Width
- Rubberbanding Lines
- Rubberbanding Circles
- A Control for Updating Stroke Color
- A Control for Updating Fill Color
- Producing a Stand-Alone Application

What is Visual BASIC

Visual BASIC is a visual programming environment designed to run in the Windows environment; the applications it produces also run in Windows. It was first released by Microsoft in mid-1991, and subsequent releases have appeared about a year apart ever since. Visual BASIC supports the following:

- An interactive application development process
- An easy approach to Windows programming, without the need to master the Windows application programming interface (API)
- Event-driven programming, in which objects respond to messages they are sent.
- The use of a visual programming environment for the development of Graphical User Interfaces (GUI)
- An extensive library of useful controls such as buttons, list boxes, editable text fields and dialogue boxes.
- The use of the Visual BASIC textual programming language, a simple yet powerful descendent of Basic.

Elements of the Opening Screen

- Title bar- the horizontal bar across the top of the screen containing the name of the application and the current mod.
Menu bar - The horizontal bar containing eight or so, drop-down menus.

For window - The large window in the center of the screen. In design mod, the form window is where you place controls such as text boxes and buttons. In run mode or when the compiled application is run, the form is the window that the user sees.

Toolbar - The horizontal row, of icons near the top of the screen. The icons give easy access to frequently used commands.

Toolbox - The vertical column of icons along the left of the screen. The toolbox provides a set of controls that you place onto your application’s forms.

Project window - This window lists the files, forms and modules that comprise the current application.

Properties window - This window shows the properties (or attributes) associated with the selected form or control. You can use this window to ascertain or modify the value of any design mode property.

Modes

Visual BASIC provides three modes which together support an interactive development process.

- Design mode - While in design mode you develop your application: place controls on forms, write procedures, set property values, and so forth.
- Run mode - You enter run mode in order to try your application out.
- Break mode - You enter break mode by suspending your application while it is running in run mode. While in break mode you can use various debugging tools.

You can switch between modes by either using the RUN menu on the menu bar, or by using the Run mode, and Stop buttons of the toolbar.

Visual BASIC serves as an interpreted, highly interactive environment for application development. You will generally switch back and forth between modes frequently during the application development process. You make a change or add an enhancement to the application in Design mode, then enter Run mode to test the procedure calls, or set breakpoints.

Event-Driven Programming

Visual BASIC is an event-driven language. This means that as you develop your application, you define a number of objects. In Visual BASIC, the primary objects are forms and controls.

An object combines state and behavior:

- The state of an object consists of the current values of all its properties.
- The behavior of an object is determined by the event procedures defined for the objects. The method consists of a sequence of program statements, that is, Visual BASIC instructions.
- An object performs an action (i.e. Behaves) in response to the message it receives. Whenever an object receives a message, it responds to the message by invoking the appropriate event procedure.
- For example, one object might be a button with the name Button1. Part of Button1’s state is described by the following property values:

<table>
<thead>
<tr>
<th>Name</th>
<th>Button1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>“Push Me”</td>
</tr>
</tbody>
</table>

When you click on Button1, the event procedure Button1_Click is automatically invoked. Thus the button behaves and executes the code stored in this event procedure.

Paint Programs and Draw Programs

A paint program is an application for creating pictures on the screen and editing at the level of individual pixels. Pictures are stored only in the frame buffer (the screen’s memory). Once an object is drawn, the paint program “forgets” the object it does not maintain an internal data structure of objects. For instance, once a line is drawn, the line loses its integrity as an object: it cannot be selected, moved, resized, and so forth.

Paint programs support such operations as painting various filled and unfilled graphics primitives into the frame buffer, copying rectangular or irregular regions of pixels, and flood-fill operations.

A draw program is an application for creating pictures on the screen and editing them at the level of graphics objects and groups of objects. Objects are stored in an internal data structure that the draw program maintains.
Once an object is defined and drawn, the line remains a line: it can be selected, moved, resized, and duplicated; its endpoints can be repositioned.

Draw programs support such operations as selecting, duplicating, deleting, transforming, and grouping and ungrouping primitives, as well as modifying the attributes and/or geometry of a primitive after it has been drawn.

Creating a Canvas

1. Let us develop the paint program outlined at the beginning of the talk. First set the property value for the form:
   - Name: frm1
   - Caption: Canvas

2. In the code window, encode the following event procedure:
   ```vba
   Sub Form_Load()
     DrawingLine = False
   End Sub
   ```
   Procedure Form_Load is automatically invoked when the application is launched and the window is opened.

3. In the general declarations section of the code window, add this declaration:
   ```vba
   Dim DrawingLine As Single
   Dim FirstX As Single
   Dim FirstY As Single
   ```

4. The canvas, in which all drawing takes place, will be implemented by a picture box.
   Place a picture box on the form and set some of its property values:
   - Name: Picture1
   - ScaleMode: Pixel
   - AutoRedraw: True
   - MousePointer: 2-Cross

The Picture Box’s Coordinate System

A graphic container is a control that contains graphic objects. The primary containers for graphics are forms and picture boxes. Inside a container, locations are specified by a pair (x,y) consisting of an x-coordinate and y-coordinate. The upper-left hand corner of the container is at location (0,0); x-coordinate increase to the right and y-coordinate increase toward the bottom. The coordinate system for a graphic container is depicted on the next page. The scale is determined by the value of the container’s ScaleMode property.
1. In the code window add this event procedure:

```vba
Picture1_MouseDown(…)
    If Not DrawingLine Then
        FirstX = x
        FirstY = y
        DrawingLine = True
    Else
        Picture1.Line (FirstX, FirstY) – (x,y)
        DrawingLine = False
    End If
End Sub
```

2. Now we can draw lines in the picture box! We fix the first endpoint of a line by clicking in the picture box; we fix the second endpoint of the line by clicking in the picture box once again. The value of variable DrawingLine indicates the significance of each click: DrawingLine=True if and only if one endpoint has been fixed but the other has not yet been fixed.

3. The procedure call gc.Line (x1,y1) - (x2,y2) draw a line from (x1,y1) to (x2,y2) in the graphic container gc. Here points are given in the graphic container’s coordinate system.

**USE OF THE SHAW UNIVERSITY LIBRARY**

It is imperative that you familiarize yourself with the instructional materials that are available to you in the James E. Cheek library on campus. Not only are there numerous books, periodicals, magazine articles, encyclopedias, and newspapers on hand for your perusal in our Library. There are course textbooks and related instructional materials that your professors have placed **On Reserve** for you as well. In addition, there you will have access to state-of-the-art computers and laptops, which help to place the world virtually at your fingertips. Don't wait another day. Become a regular patron at the James E. Cheek Library here on the campus of Shaw University. You’ll become a far better scholar, a more capable and well-rounded intellectual, and a sharper and more competitive individual. It’s your library. Use it!