ISSN No. 0976-5697

Volume 2, No. 6, Nov-Dec 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Optimization into UI Application Development through Bubble Sort Algorithm Using Control Array in OOP's

Devendra Gahlot*
Research Scholar,
Dept. of Computer Science,
Suresh Gyan Vihar University,
Jagatpura, Jaipur, Rajasthan, India.
devendragahlot@gmail.com

Prof. (Dr.) S.S.Sarangdevot
Director,
Dept. of Computer Science and Information Technology
J.R.N. Rajasthan Vidyapeeth University
Udaipur, Rajasthan, India
dr.sarangdevot@gmail.com

Abstract: The purpose of this research is to develop an interactive UI for Bubble Sort Algorithm. The Copy of a components (Object) have been avoiding by the developers in Object Oriented Programming since beginning. Because copying of components create array of related components. But using my concern research, we can develop an interactive UI to understand Bubble Sort Algorithm being some Optimization into the same. My Aim is optimization into UI Application using array of component. Which have generally not been used during the development of UI application.

Keywords: visual programming, control array, performance, bubble sort algorithm, Interactive UI development

I. INTRODUCTION

Graphical user interfaces (GUIs) simplify use of computers by presenting information in a manner that allows rapid assimilation and manipulation. The use of visual constructs (widgets) that mimic physical objects such as `switches' and `buttons' can speed learning, by providing an intuitive method to provide input to the computer. A GUI is not always an improvement. As demonstrated by some widely used commercial programs, a poor GUI implementation can obscure functionality. If the GUI is organized in a counterintuitive manner, or if the menu contents are arranged haphazardly, or if commonly performed operations require several unexpected steps to be performed, then a user must typically invest a significant amount of time in learning how to use the program before the program can be used effectively. A good GUI design does not require users to memorize the steps needed to perform an action. This is particularly important for scientific applications, where the goal of the user should be to understand the theory behind a program rather than master the arcane steps needed to perform an action. There is one case where memorization may be unavoidable. It is appropriate that a GUI incorporate shortcuts that simplify multi-step tasks; use of these shortcuts may not always be intuitive and expert users may choose to commit them to memory to speed their work. There should also be an obvious way to perform the same tasks, albeit less efficiently, without use of the shortcut. [1]

The user interface is one of the most important parts of any program because it determines how easily you can make the program do what you want. A powerful program with a poorly designed user interface has little value. Graphical user interfaces (GUIs) that use windows, icons, and pop-up menus have become standard on personal computers.

In current vogue, the computer hag been a part of over life and behind the basic region for the same is an effective and interactive UI Application (user Interface) with computer.

To develop efficient and interactive UI Application, the developer adopts a particular programming language. As far as my research is concern, I am going to adopt Object Oriented Programming language for the same. Array, Polymorphism and Inheritance can play major role to develop an effective, interactive and efficient UI Application. Polymorphism and Inheritance are the main features of Object Oriented Programming.

A. *What is Object Oriented Programming:*

Not all programming languages can be ''object oriented''. Yet claims have been made to the effect that APL, Ada, Clu, C++, LOOPS, and Smalltalk are objectoriented programming languages. I have heard discussions of objectoriented design in C, Pascal, Modula2, and CHILL. Could there somewhere be proponents of objectoriented Fortran and Cobol programming? I think there must be. ''Objectoriented'' has in many circles become a high-tech synonym for ''good'', and when you examine discussions in the trade press, you can find arguments that appear to boil down to syllogisms like:

Ada is good Object oriented is good --------------------------------- Ada is object oriented

''object oriented'' ought to mean in the context of a general purpose programming language.[2]

B. *Array:*

Array programming has two key characteristics:

a. Operations can be directly applied to entire arrays of values.
b. A set of special functions and operators provides powerful means of data manipulation and allows complex data manipulation processes to be expressed concisely.[3]

C. Contol Array:

A control array can be created only at design time, and at the very minimum at least one control must belong to it. You create a control array following one of these three methods:

You create a control and then assign a numeric, non-negative value to its Index property; you have thus created a control array with just one element.

You create two controls of the same class and assign them an identical Name property. Visual Basic shows a dialog box warning you that there's already a control with that name and asks whether you want to create a control array. Click on the Yes button

You select a control on the form, press Ctrl+C to copy it to the clipboard, and then press Ctrl+V to paste a new instance of the control, which has the same Name property as the original one. Visual Basic shows the warning mentioned in the previous bullet.

Control arrays are one of the most interesting features of the Visual Basic environment, and they add a lot of flexibility to your programs:

a. Controls that belong to the same control array share the same set of event procedures; this often dramatically reduces the amount of code you have to write to respond to a user's actions.
b. You can dynamically add new elements to a control array at run time; in other words, you can effectively create new controls that didn't exist at design time.
c. Elements of control arrays consume fewer resources than regular controls and tend to produce smaller executables. Besides, Visual Basic forms can host up to 256 different control names, but a control array counts as one against this number. In other words, control arrays let you effectively overcome this limit.

The importance of using control arrays as a means of dynamically creating new controls at run time is somewhat reduced in Visual Basic 6, which has introduced a new and more powerful capability.[4]

D. Array Programming:

Array programming was introduced in the early sixties to ease the description of mathematical processes manipulating arrays [6]. It was initially thought as a simplified way for manipulating array data structures in the language and many implementations are sequential. However, it was also considered as a mean to take advantage of mainframes vectorial processors and exploit data parallelism. The principle is simple: arrays are considered as first-class entities within the language and traditional arithmetic operators (such as addition, etc) are defined natively to operate on arrays or combination of scalar values and arrays (e.g. if X and Y denote arrays of numerical values, X + Y and 2 _X are valid expressions). Array operations are seen as a convenience to avoid writing explicit loops for simple repetitive operations. They reduce the need for control structures use inside the language.[7]

E. Bubble Sort Algorithm:

Bubble sort algorithm used in the experiments below was described by C++ language as:

```
template<class Type>
void BubbleSort(Type *R,int n)
{
  int i,j; Type temp;
  for(i=0;i<n-1;i++)
    for(j=0;j<n-i-1;j++)
     if(R[j]>R[j+1])
{temp=R[j];R[j]=R[j+1];R[j+1]=temp;}
}
```

In the best case, the input sequence is positive, the algorithm needs *n-1* comparisons, and its time complexity is *O(n)*. In the worst case, the input sequence is negative, the algorithm needs

$$\sum_{i=n}^{2}(i-1)=n(n-1)/2$$

comparisons, and its time complexity is $O(n^2)$. Whatever, the space complexity of bubble sort is *O(1)*.[8]

F. Optimization:

Optimization is the art and science of making your program faster, smaller, simpler, less resource hungry, etc. Of course faster often conflicts with simpler and smaller so optimization is a balancing act.[9]

II. AIMS AND OBJECTIVES OF REASEARCH WORK

The Object Oriented Programming is very powerful and reliable language to develop a UI Application. If the tools of Object Oriented Programming are optimally used then an interactive and efficient UI Application can be built.

The minimum amount of communication in a sorting algorithm involves moving elements from the locations they start out to where they eventually belong (in the sorted order). An optimal sorting algorithm will communicate every element from its adjacent location to a location in remote memory at most once. Our algorithm is optimal in this sense. For instance, if the input is already sorted, no data movement occurs. However, if the input is in reverse sorted order, almost all elements may need to be communicated to their destinations.

A. AIM:

a. As far as my research is concern, my Aim is optimization into UI Application using array of component. Which have generally not been used during the development of UI application?
b. With some optimization into UI application using array of object. I want to purpose to develop a UI. Which will show the method of sorting with moving components (Textbox) towards their real position according to their consisting data? And show also related algorithm for related sorting (Bubble sort, Selection sort, Radix sort etc.) with tracing view of problem.
c. The user of UI application can control all these things regarding sorting. Like speed of sorting, order of sorting, data to sort etc.
d. Nuts and shell, my UI application will be an integrated group of different view of problem solving at same running interface. Like process view of problem solving, tracing view of algorithm of related problem.

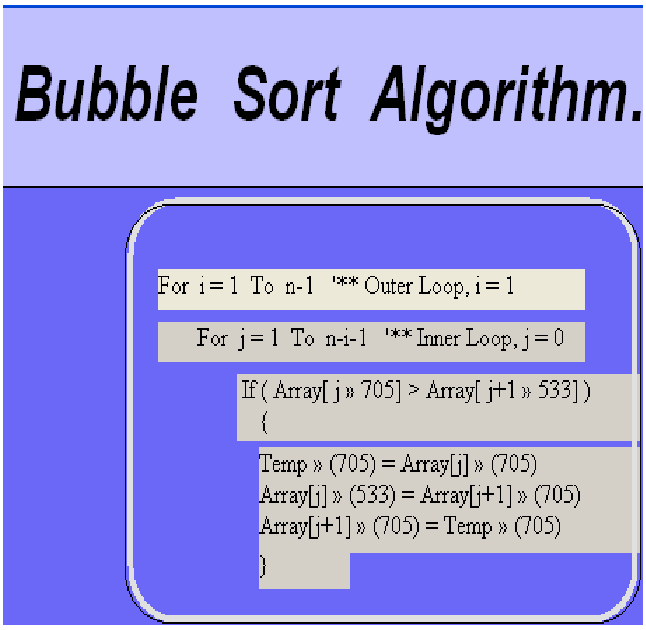
B. Objective:

a. My first objective is, to optimum use of array of object, which is generally not used by the developer.
b. My second objective is optimization into UI application, as Graphical representation to solving the problem of sorting. Because, Graphical User interface has been very popular, interactive and efficient to make easy to interact and understanding the problems.
c. To provide a tool, which can play big role to make easy to explain related problem by the mentor and understand by viewer.

III. METHODOLOGY/LABORATORY WORK

a. As far as, my research is concern, it may only be academic oriented so that for the same, I will get information about algorithm of different types of sorting.
b. I will adopt a particular Object Oriented Programming language among C++, VB, and Java. Or VB.net, PERL etc.
c. Drawing the flow diagram to solve related problem. (Diagrammatical Method)
d. Create the array of component (Like Textbox or label) using copy & paste method at same container (Form or frame). Which does the developer generally not use?

A. Algorithm of Bubble Sort Using Control Array:



*'***** Coding *********

Private Sub cmd_Asc_Click()

'* Checking the Elements to Load *****

```
If Val(Text4) < 1 Then
MsgBox "Please Specify Elements to Load   First.", vbCritical, "Error"
Text4.SetFocus
Exit Sub
End If
```

' Checking the Elements if any Alpha exists ****

```
If Check(Val(Text4) - 1) = False Then
MsgBox "Anomalies."
Text1(tindex).SetFocus
Exit Sub
End If
'*****************************************
For i = 1 To Text4 - 1
If Text1(i) = " " Then
MsgBox "ERROR", vbOKOnly + vbCritical, "specify the values for array elements"
Text1(i).SetFocus
End If
Next
'*****************************************
For i = 1 To Text4 - 1
'** Color Change **
Label3(0).Caption = "For  i = 1  To  n-1   '** Outer Loop, i = " & i
Label3(0).ForeColor = vbRed
If Check1.Value = 1 Then BigDelay (Val(txtd1))
'** Loop Delay **
Label3(0).ForeColor = vbBlack
For j = 0 To Text4 - i - 1
If flag = True Then
Exit Sub
End If
'** Color Change **
Label3(1).Caption = "      For  j = 1  To  n-i-1   '** Inner Loop, j = " & j
Label3(1).ForeColor = &HFF00FF
If Check2.Value = 1 Then BigDelay (Val(txtd2)) '** Loop Delay **
Label3(1).ForeColor = vbBlack
Text2.Left = Text1(j).Left
Text3.Left = Text1(j + 1).Left
Text2.Top = Text1(j).Top
Text3.Top = Text1(j + 1).Top
Text2 = Text1(j)
Text3 = Text1(j + 1)
If Val(Text1(j)) > Val(Text1(j + 1)) Then
```

'* Algorithm *****

```
Label3(2).ForeColor = &H8000&
Label3(4).ForeColor = &H8000&
Label3(2).Caption = " If ( Array[ j » " & Text1(j) & "] > Array[ j+1 » " & Text1(j + 1) & "] )" & vbCrLf & " {"
Label3(4).Caption = "}"
Text2.Visible = True
Text3.Visible = True
Text1(j).Visible = False
Text1(j + 1).Visible = False
temp1 = Text1(j)
'*** Algorithm ***
Label3(3).ForeColor = &H4080&
Label3(3).Caption = "Temp » (" & temp1 & ") = Array[j] » (" & Text1(j) & ")" & vbCrLf
Text1(j) = Text1(j + 1)
Text1(j + 1) = temp1
```

'* Algorithm *****

```
Label3(3).Caption = Label3(3).Caption & "Array[j] » (" & Text1(j) & ") = Array[j+1] » (" & Text1(j + 1) & ")" & vbCrLf
```

'* Algorithm *****

```
Label3(3).Caption = Label3(3).Caption & "Array[j+1] » (" & Text1(j + 1) & ") = Temp » (" & temp1 & ")"
```



```
While Text2.Left > 750
DoEvents
Text2.Left = Text2.Left - 1
Text3.Left = Text3.Left + 1
Delay (HScroll1.Value)
Wend
 'Label3(4).ForeColor = vbBlack
me.Refresh
While Text2.Top < Text1(j + 1).Top
DoEvents
Text2.Top = Text2.Top + 1
Text3.Top = Text3.Top - 1
Delay (HScroll1.Value)
Wend
me.Refresh
While Text2.Left < Text1(j + 1).Left
DoEvents
Text2.Left = Text2.Left + 1
Text3.Left = Text3.Left - 1
Delay (HScroll1.Value)
Wend
Me.Refresh
End If
Text1(j).Visible = True
Text1(j + 1).Visible = True
'*** Change Algorithm Color ***
Label3(2).ForeColor = vbBlack
Label3(3).ForeColor = vbBlack
Label3(4).ForeColor = vbBlack
Next
Next
Label3(5).Caption = "Result - Array is Sorted (Ascending
Order)."
                    '**** End Procedure ****
End Sub
```

IV. CONCLUSION

a. My interactive and effective User Interface will simplify complicated task into easy task.
b. Effective way to explain sorting and their algorithm for mentor or trainers.
c. Effective way to understand sorting and their algorithm for student.
d. The real implementation of array of Object in Object Oriented Programming.
e. The problems can be solved using other problems being their proper optimization (Like Copy of Textbox1 at same form).

V. REFERENCES

[1]. Brian H. Toby, EXPGUI, a graphical user interface for GSAS, NIST Center for Neutron Research, National Institute of Standards and Technology, Gaithersburg, Maryland 20899-8562, USA. J. Appl. Cryst. (2011). pp. 210-213, Feb-2001.

[2]. Bjarne Stroustrup AT&T Bell Laboratories Murray Hill, New Jersey 07974, What is ‘‘ObjectOriented Programming’’? pp. 1 to 21, Software, IEEE, 1988- ieeexplore.ieee.org.

[3]. Philippe Mougin, St´ephane Ducasse, Software Composition Group University Of Bern Bern, Switzerland, Integrating Array Programming in ObjectOriented Programming, PP 1 to 13, 2003, http://www.fscript.org/download /OOPAL.pdf

[4]. http://visualbasic.freetutes.com/learn-vb6/lesson11.html

[5]. Michael Schidlowsky. "Java and K". pp. 1 to 3, Retrieved 2008-01-23. http://en.wikipedia.org/wiki/Array_programming

[6]. H. Hellerman. Experimental personalized array translator system. Communications of the ACM (CACM), 7(7):433{438, 1964.

[7]. Johan Montagnat CNRS / Univ. of Nice I3S laboratory, Benjamin Isnard INRIA ENS Lyon, LIP “A data-driven workflow language for grids based on array programming principles” pp 4 to 10. http://rainbow.polytech.unice.fr/publis/montagnat-isnard-etal:2009 .pdf

[8]. You Yang, Ping Yu, Yan Gan, School of Computer and Information Science Chongqing Normal University Chongqing, China., “Experimental Study on the Five Sort Algorithms” pp 1 to 4, 978-1-4244-9439-2/11/$26.00 2011 IEEE

[9]. WIKIBOOKS(Open books for an open world), PP1, http://en.wikibooks.org/wiki/Visual_Basic/Optimizing_Visual_Basic,