Understanding Timer and Common Dialog Controls and their Applications to Mechanics and GDI+

D.A. Adenugba
Department of Physics
The Federal University of Technology, Akure
P.M.B 704, Akure, Ondo State. Nigeria.

ABSTRACT

Colours create beauty; nature is soaked in diverse colours. The primary colours, Red (R), Green (G) and Blue (B) are fundamental colours from which other colours are derived. The painter basic colours of R, Y and B are still derived from primary colours when one realizes that Yellow, Y = R + G. This paper discusses common dialog and timer controls. The unsurpassed significance of common dialog controls was clearly depicted in their applications to Mechanics, Graphics Device Interface Plus (GDI+) and in showing 128 system-defined colours automatically using timer control and manually through colour dialog box. Provision for colour combinations enables over sixteen million colours above the system-defined colours to be displayed as different intensity levels are assigned to the primary colours values between 0-255 via Color.FromArgb method. The robust functionalities of the customized colour control developed with Microsoft Windows Control Library were exposed in a client application, WhatSoft to manipulate various displays of colours. Further applications of timer and colours are made to GDI+. This prompted the development of classes for rectangle and square that are hosted in a namespace, dvShapes. A class library was developed for ten mechanics models and its functionalities were exposed in MechSoft application. Software developers will find the customized colour control useful in their work. Tutors will find WhatSoft and MechSoft packages indispensable in teaching and learning.

General Terms
Colours, MechSoft**,** WhatSoft**,** common dialog box

Keywords
Primary Colour, Mechanical advantage, velocity ratio, timer, GDI+

1. INTRODUCTION

The difference between the developed and developing nations is not more than that between angle and angel, form and from, soft ware and software. A man may possess soft (light) ware (materials) and if the ware (goods) is light (soft) he may wear it during the dry season because of the heat, but he has not developed software, otherwise called application or package. A software package is a catalyst for quality research, teaching and learning. It motivates and inspires researchers to dive deep into areas unsearched to unearth facts and figures promptly and confidently. A software developer is a person who thinks for others; possesses the pipe and dictates the desired tones anytime any day and anywhere.

The first thing created by Almighty God was light (Genesis1:1-2). The primary colours Red (R), Green (G), and Blue(B), which the Physicists study under optics, could be combined in several ways to produce secondary colours.

There are fundamental controls that assist user interfaces to be designed swiftly and to possess the look and feel of a windows package. A common dialog control is invisible at runtime. To show the control for user's selection its ShowDialog method (common to all the dialog boxes), which returns the DialogResult, has to be summoned. They are implemented as modal dialog box; they are not placed on the form but in components tray of the form. As modal dialog box, a common dialog control is required to be attended to before any other action can be taken or task performed.

Colours have been since the inception of the world; it is part of nature. In the primary school mid-60s, objects, numbers, the English and the Yoruba alphabets were taught to be recognized. Drawing book with unpainted diagrams and pictures were to be coloured with crayons. This is the static and rigid efforts to teach pupils to recognize various colours. This trend has not changed. How nice would it has been if the computer in its present status existed then to make the job of the smart, industrious and dedicated teachers easy, flexible and quick to achieve; and for learners to press buttons and self-discover primary and complementary colours.

Colours abound and in right blending proportions they adore the vicinity with beauty. They are the pleasant food for the eyes and mind. They are cherished by all in varying degrees all over the universe. Nature supports colours; indeed, nature is enveloped in colours. The fire depicts red; the plants show green colour everywhere and the sky above our head epitomizes blue colour. The primary colours (RGB) are fundamental to pleasant and meaningful life on earth. Roses are red, violets are blue and grasses are green; they brighten the environment.

Timer is a powerful control that assists the automatic presentation of facts to learners for a specific time, through its Interval property. In this 21st century these imperative tasks and activities of colour and objects display ought to be taught and learnt dynamically with interest, flexibility, promptly, anytime and anywhere in comfort not only of the four walls of classroom but also at home. Unfortunately, this is not what is found in most developing nations.

Therefore, this paper discusses common dialog controls and timer control. The ColourDialog control was customized and applied to display assorted colours manually and automatically. The strength and weaknesses of open and save dialog controls were shown through their application to Mechanics and Graphics Device Interface plus (GDI+). The rich functionalities of a customized control, dvColourCtrl and the intrinsic controls were exposed in a client application WhatSoft developed using Microsoft Visual Studio, 2010.

The target users of WhatSoft package ranges from primary school pupils to undergraduate students; Technologists and Scientists; Painters, and Printers to other Artisans making use of colours. In addition, colours and timer are applied to GDI+. To this end, classes for rectangle and square, hosted in a

namespace, dvShapes were developed. A class library, dvMechCls was developed for ten Mechanics models and its functionalities exposed in MechSoft application to show the flexibility of open and save dialog boxes.

2. COMMON DIALOG CONTROLS

CommonDialog controls easily make common but frequently required tasks available for use promptly. Some of the dialog controls are [1-6]

FontDialog (for font typeface and style to be applied to selected text).
SaveFileDialog (for getting name of file where document will be saved).
ColorDialog (for selecting colour(s) to use).
FolderBrowerDialog (for selecting folder).
PrintDialog(for printing document).
PrintPreview (for previewing document before printing) and
PageSetUpDialog(for setting printing margin, paper etc).

A typical intrinsic OpenFileDialog box is shown in figure 1 for getting input file name. Common dialog controls could be declared and instantiated at run time, besides being added at design time. To add a common dialog control at design time follows these steps: on the form Designer and from the Toolbox locate Dialogs section; select the appropriate dialog box, say ColorDialog box by doubling clicking on it to add it to the components tray below the Form Designer. This could also be achieved by clicking on, and dragging it to the form; instead of staying on the form it is automatically added to the components tray. By setting some properties of the control its functionalities are available for use. Apart from specifying custom colours, ColourDialog allows predefined colours to be selected.

Colours, the food for the eyes and balm to the heart, could be displayed in a picture box, Usercontrol and label control, among other controls. Apart from displaying blending colours, their names could be shown for system-defined colours as figure 2 depicts.

3. CUSTOMIZED COLOUR CONTROL

A Windows Forms Control Library provides a swift means to develop a customized control. The customized colour control, dvColourCtrl developed for this work has the functionalities shown in Table 1 together with the function they perform in the last column. The method, dvGetAnyColour opens the colour Dialog box for colour selection. Any colour selected by user will be returned through the Color property, else the default black colour will be returned. The rest 11 methods are shown in Table 2. The customized colour control in action could be seen in figure 2. The dvInits function initialises some variables and it is called at the constructor, while dvIntiWhat and dvIntiColour initialise some title and colour labels' variables respectively such as the back and fore colours. Through dvChgPicBackColour function the picture box back colour is altered and the second overload positions the box. In addition, the dvLoadColour function loads 128 system defined colours into dvLstColour list(Of Color) object.

Several GDI+ objects utilize colour to render objects to the drawing canvas. Drawing object and filling object methods require pen and brush respectively to be passed as argument to the methods. System defined colours are named colours. Colours could be procured from Argb, Ole, Win32 and HTML colour. The Argb stands for Alpha, Red, Green and Blue, and is otherwise called GDI+ colours or blending value; since the colour could be blended to produce over sixteen million colour combinations. The Alpha is the most significant 8 bits which controls the transparency of the colour when it value is zero and opaque when it is 255. R, G and B (B is least significant 8 bits) are primary colours which determine the intensity of the colour. HTML colour is not a VB.NET colour object consequently it has to be converted using HTML method.HTML colors are defined using a hexadecimal notation (HEX) for the combination of R, G, and B color values. The lowest and highest values are 0 (in HEX: 00) and 255 (in HEX: FF) respectively. HEX values are specified as 3 pairs of two-digit numbers or letters, starting with a # sign; for black, it is #000000 [9-10].The method Argb has four overloads. One accepts 32-bit value, another two arguments of alpha and base colour. The last two has three(rgb) and four (argb) arguments respectively [3, 6].

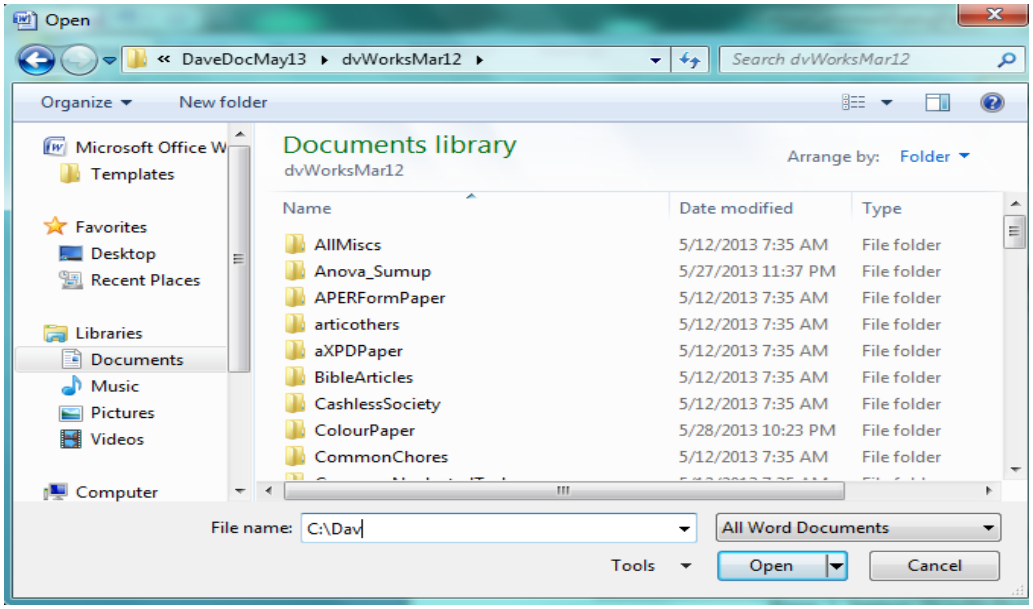

Figure 1: Intrinsic OpenFileDialog Box

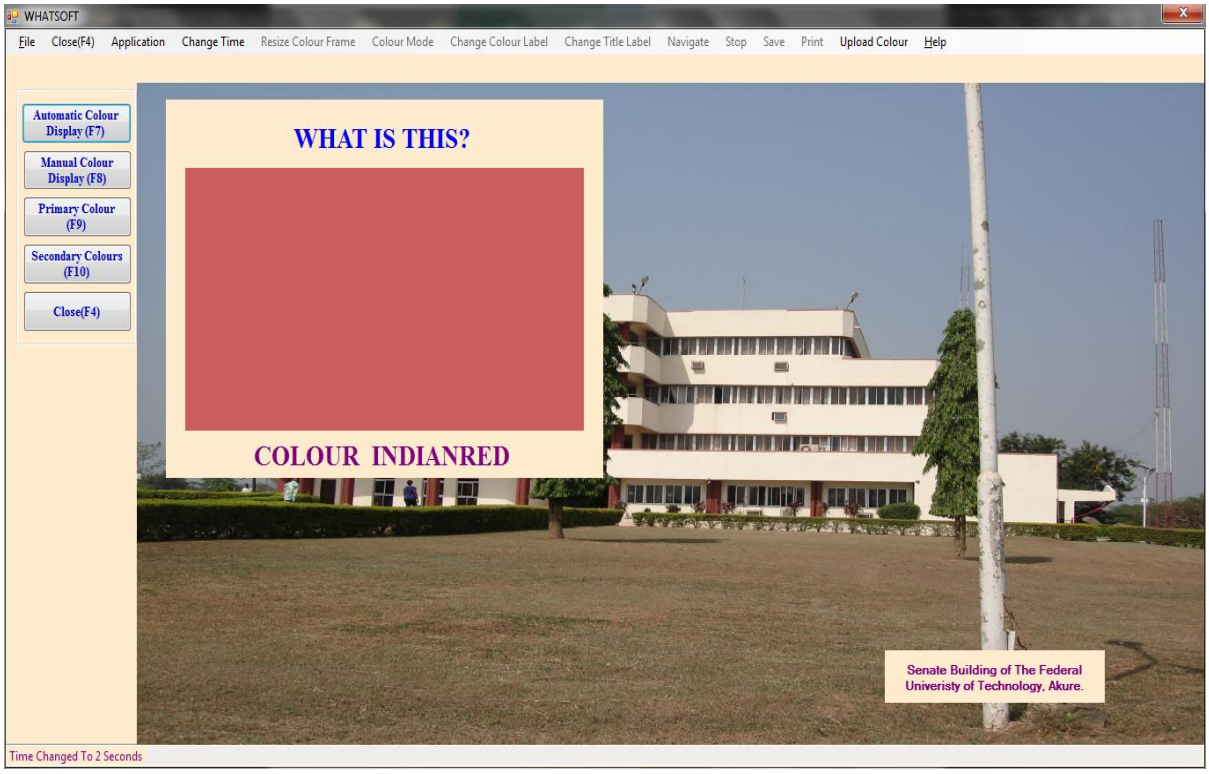

Figure 2: Working Windows for colour display and model evaluation. Background picture is the elegant Senate building of the Federal University of Technology, Akure, Nigeria (FUTA).

Table 1: Properties in dvColourCtrl customized control

S/N	Name	Task(s)
1	dvGetCurrentColour	It returns the colour currently shown.
2	dvGetCurrentColourNam	It returns the name of the currently displayed colour.
3	dvGetInternalColour	It returns internal colour which index is supplied. The index is checked to be within the range of available colours before it is used to retrieve the colour. Error is reported if index is out of range.
4	dvGetSetColourBackColour	It gets or sets colour label back colour.
5	dvGetSetColourForeColour	It gets or sets colour label fore colour.
6	dvGetSetWhatBackColour	It gets or sets the title back colour.
7	dvGetSetWhatForeColour	It gets or sets the title text fore colour.
8	dvGetSetWhatTxt	It gets or sets the title text.
9	dvSetColourAlign	It sets colour label text alignment.
10	dvSetColourBackForeColours	It sets the back and fore colours of colour label
11	dvSetColourBackForeColoursAlign	Besides setting back and fore colours, it sets the colour label text alignment.
12	dvSetColourBorderStyle	It sets the colour label border style.
13	dvSetColourFont	It sets the colour label font.
14	dvSetWhatAlign	It sets the title label alignment.
15	dvSetWhatBackForeColours	It sets both back and force colours of the title label.
16	dvSetWhatBackForeColours	It sets title text in addition to back and fore colours.
17	dvSetWhatBackForeColoursAlign	It sets title text alignment besides back and fore colours.
18	dvSetWhatBorderStyle	It sets the title label border style.
19	dvSetWhatFont	It sets the title label font.
20	dvSetWhatTxtAlign	It sets the title and aligns it.
21	dvShowHideLblColour	It set the colour label border style.
22	dvShowHideLblWhat	It hides the title label when it is false; makes it visible when true.
23	dvTmrEnable	This Boolean property enables the timer when true and disables it when false

Table 2: Methods in dvColourCtrl customized control

S/N	Name	Task(s)
1	dvChgInterval	It changes the timer interval. If the supply value is less than 0, the value is set equal to the default value of 2 seconds (2,000 milliseconds). Also, if it is below half second it is multiplied by 1000 to convert it to milliseconds, the unit of the timer interval.
2	dvGetAnyFont	Using Font Dialog box, it obtains selected font and returns it. If operation is cancelled, the default is returned.
3	dvGetColourParts	It returns the parts (red, green and blue) of the colour supply.
4	dvGetCurrColourAndName	It returns both current colour and colour name.
5	dvResizeLblCtrls	It resizes the user control. The second overload also changes the back colour.
6	dvShowFirstColour	It displays the first colour in dvLstColour object using colour index set to zero. The second overload supplies the control size also.
7	dvShowLastColour	It displays the last colour in dvLstColour; index is less one to account for zero-based dvLstColour object.
8	dvShowNextColour	It displays the next colour in dvLstColour. If index is greater than the maximum, the last colour is shown.
9	dvShowPrevColour	It displays the previous colour in dvLstColour. If index is less than 0, the index is set to zero.
10	dvShowUserColour	It displays the colour supply by user. The second overload specifies new timer interval. Although the first two overloads disable the timer, the third allows the user to specify the enable property of the timer. This was the method called under manual menu of figure 2.
11	dvStart	It starts the timer as its enable property is set true. The second overload sets both interval and enables properties of the timer control.

4. APPLICATIONS

4.1 Colours and Timer

On the working form of figure 2, there are fourteen main menus with different numbers of submenus. When the first button to the left of figure 2 is clicked, the customized control is made visible, and the dvStart method (Table 2, S/N 11) is summoned with the interval set to 3000 milliseconds (3 seconds) and enabled property set true. Different colours are shown with the name below and the title above it.

To alter the timer interval, to suit users like teachers, learners and marketers, the Change Time main menu is pulled down and one of six context menus chosen from. The first five changes the time from 1second to
5seconds; the last submenu obtains the required time from the users, and if found worthy (only integer permitted), it passes the value to dvChgInterval method to alter the time. To change the colour text attributes, there are seven submenus under Change Color label main menu to do this.

Similarly, through the Change Title label the available eight submenus aid the title attributes to be altered to suit user's needs. For instance, when ForeColour submenu is clicked the colour dialog box is displayed; the colour selected is passed to dvGetSetWhatForeColour property (Table 1, S/N 7). To display colour selectively, Navigate main menu has submenus of First, Next, Previous and Last to achieve this purpose. The Upload Colour menu has no submenu, but as soon as it is clicked, OpenFileDialog box, like the one shown in figure 1, is displayed and the user's supply file is opened and the name of the colour to display is retrieved and assigned to dvShowUserColour method. One of the overloads of this method uses the colour name (dvNamColourToShow) to show the colour.

The Stop menu does what its name suggests; it stops the automatic animation of the colour as dvStart method is called with enabled property equals false. When the second button on figure 2 is clicked, dvGetAnyColour method is summoned and the selected colour (not colour name) is assigned to dvShowUserColour method.

The third and fourth buttons respectively display primary and secondary colours. The primary colours could be mixed in varying proportions to give over sixteen million combination of colours (Evangelos, 2010). Table 3 shows that when R (255) is mixed with B (255) fuchsia colour is obtained; but for R = 200, G =128 and B = 100 the name of the colour obtained is given in hexadecimal as column 4 depicts. It was observed that mixed colours with known system defined colours would display the colour name instead of the name in hexadecimal.

By carefully modularizing the customized control future maintenance is guaranteed. Another user interface not shown for lack of space beams out after half a minute if the user does not press C or F2 key to move on. This automatic display is actually the work of a timer on the form, which triggers dvContinue private function.

Table 3: Combined primary colours to produce secondary colours

R	G	B	Hex	Output Colour Name
255	0	0	ffff0000	Red
0	255	0	f00ff00	Lime
0	0	255	ff0000ff	Blue
255	0	255	ffff00ff	Fuchsia
0	255	255	ff00ffff	Cyan
255	255	0	ffffff00	Yellow
255	255	255	ffffffff	White
200	128	100	ffc88064	ffc88064
25	120	60	ff19783c	ff19783c

4.2 Mechanics

The next application is to mechanics models. The mechanical advantage (MA) is expressed by[7-8,12-13]

$$MA = \frac{Load}{Effort}$$

Load, L= Force applied BY the machine; Effort, E =Force applied TO the machine.

$$L = MA^{*} E$$

$$E = \frac{L}{MA}$$

M.A is used to describe the amount of force that is utilized internally by a mechanical device [11]. Ideal or *theoretical* M.A, unlike actual or real M.A, is frictionless, transmits power without adding to or subtracting from it and causes no wear and tear. It is the M.A of an ideal machine and does not exist in the physical world[12-14].

A class library was developed for the mechanics models. The dvCompMAParas method in the server calls five overloaded functions (see S/Ns l-5) in Table 4. It solves Mechanical advantage, MA when L and E and when Efficiency, Eff and velocity ratio, V.R are supplied. V.R is otherwise known as distance ratio[15]. Besides, it calculates all the parameters to which it is related: E, L and Eff. These computations are achieved through dvEnumMechMAType enumeration declared for it. The second method (dvCompVRParas) is for computing V.R.

This method calls five overloaded functions (S/Ns 6-10) in Table 4 base on user's specification through dvEnumMechVRType enumeration. A flexible application MechSoft was developed for exposing the functionalities in dvMechCls class library. Figure 3 is the working form for mechanics. There are two submenus under Compute Mechanics main menu; each has five submenus which in turn has single and multiple results submenus. When the fourth submenu of V.R is clicked and the first submenu for single result is clicked, we obtain this result directly for effort distance:

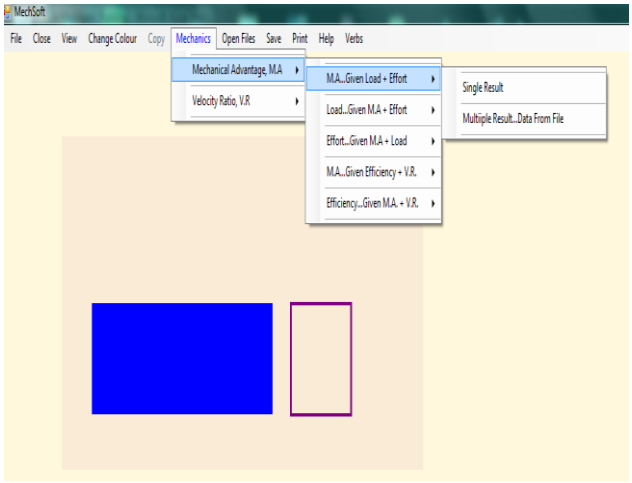

Figure 3: Working Windows for Mechanics

Effort Distance, Ex

Effort Distance=VR*Lx

VR=100, Lx=50

Effort Distance=100*Lx

Effort Distance=100*50

Effort Distance=5000

For all the single result, the workings, as seen here, are provided, which could be utilized for learning and teaching real time. The colour with which to display result could be altered via Change Colour menu. When this main menu is clicked, the colour dialog box is displayed and the selected colour is employed as fore colour to display the result on a label control. When the submenu for multiple result is clicked, open file dialog box like that shown in figure 1 will be displayed for user to upload his/her data file. The file name is capture and passed to the method to compute the multiple results. Typical results for Load and V.R are shown in Tables 5-6. The results could be formatted to the level user needs through dvSetFMT property. Here for Table 5, it is "#.00" (for 2 decimal places) and for Table 6, "#.000 (3 decimal places).

If the user data is not numeric, either for one or both data, NR, for non result, will be returned as second to the last line of Table 6 shows. There is no need to terminate operation when invalid data is encountered here, rather data filtration and action are the required actions as operation continues until the input data is used up. Our package, MechSoft could read and compute one hundred thousand data; but for space only few data and results are shown.

Table 4: Functions in dvMechCls class library

S/N	Function	Task(s)
1	dvCompMA_LE	It calculates MA, given Load, L and Effort, E**.**
2	dvCompMA_EffVR	With given **e**fficiency, Eff and velocity ratio, VR, MA is computed.
3	dvCompEffort_LMA	When L and MA are supplied, E is estimated.
4	dvCompLoad_MAE	Given MA and E, L is evaluated.
5	dvCompEfficiency_MA VR	Outputs Eff, from given MA and VR **.**
6	dvCompVR_EdLd	It computes VR given distance moved by effort, Ed and distance moved by load in the same time, Ld.
7	dvCompVR_MA Eff	It computes VR, given MA and Eff.
8	dvCompVR_NumPulley	It also finds VR, given the number of pulleys. Note that VR is equal to the number of pulleys. It accepts the number**;** say 6 through the UI provided for it and outputs: V.R = NumberPulleys = 6**.**
9	dvCompEffortDist_VR Ld	It calculates distance moved by effort, Ed with supply VR and Ld.
10	dvCompLoadDist_Ed VR	It computes distance moved by load, Ld using Ed and VR supply.

Table 5: Typical result for Load result for velocity ratio

MA	Effort	Load
74.8	20.00	1496.00
61.0	31.98	1950.78
25.0	9.42	235.50
88.0	34.00	2992.00
4.2	0.25	1.05

Table 6: **Typical**

Ed	Ld	VR
24.990	10.77	2.320
60.000	20.00	3.000
0.749	2.50	0.300
Nodat	*100.00*	*NR*
11.460	100.00	0.115

4.3. GRAPHICS DEVICE INTERFACE PLUS

Graphics Device Interface Plus (GDI+) is a powerful drawing tool that could be utilized to draw virtually any object. Aside rectangle and square, GDI+ could be use to draw line, ellipse and polygon, among others.

The namespace, dvShapes hosts two library classes: dvRectCls and dvSqCls for drawing and filling rectangle and square. Square is essentially the same as rectangle when the dimensions are the same; that is, width=height. We separated them because of future expansions to GDI+. Also, there is no need to send in two arguments when drawing or filling a square; the side, which stands for width and height, is just enough.
The dvDrawRect method accepts three arguments, the last being paint event handler(e).

A graphics object, dvgrafic, declared at server's level(Private dvgrafic as Graphics) being used in many place, is created via e (dvgrafic = e.Graphics) in the methods. The bounding Rectangle(dvRect) is created with the user input dimensions of width and height (dvRect = New Rectangle(X,Y, dvWdIn,dvHtIn).
Note that there is no DrawRectangle method that accepts RectangleF, so pass the location(X,Y) and dimension(dvWdIn for width, dvHtIn for height) direct (dvgrafic.DrawRectangle (dvPen, X, Y, dvWdIn, dvHtIn). But for the first this is the simple code: dvgrafic.DrawRectangle(dvPen, dvRect).The X and Y offsets from the top-left coordinates are set internally to 50 for both. The drawing pen colours and widths are server level declared variables; the initial values of blue and 2 for rectangle and purple and 3 for square are set at the constructor.

6. CONCLUSION

An exciting and appealing user interfaces have been designed and developed to display different wide range of colors dynamically using a timer control. Eight methods are written for drawing and filling rectangle and square using GDI+ functionalities. Timer control is a powerful control that enables objects to be displayed without human intervention, aside varying colours. It helps us to vary the colours of the drawn and filled rectangle and square real time.

The client application, WhatSoft developed could be utilized real time to teach and learn manifold colours manually and automatically, which traditional method could not achieve swiftly and efficiently. The color dialog control provides an easy means of accessing and obtaining colours. The server written for mechanics models promises to be a solid and reliable base for an accurate and flexible application development. Researchers and communication experts will find MechSoft package handy in estimating ten Mechanical models. More physics models should be added to further endear the class library, dvMechCls to users all over the world. Additional pens and brushes, as well as shapes and images are expected in the GDI+ classes; their applications to These values are altered every two seconds by the timer on the working windows to give varying and pleasing colours and widths effects. Instead of single data type for dimensions, the second overload accepts integer data type dimensions. These two methods arguments are repeated for the dvFillRect methods which fill drawn rectangle with the brush colour. Solid brush is used all through and the colour is changing through the same timer for the pen objects. For square, the dvDrawSq(for drawing square) and dvFillSq (for filling square) accepts a single dimension (the side) and e as arguments. The shapes on figure 3 are timer in action changing the colours based on the value of R, G and B. The method, varying the integer values of the primary colours, is dvSetRGB that prevents the values from being below 0 and above 255.

5. RESULTS AND DISCUSSION

WhatSoft application has been developed using the famous Microsoft Visual Studio, 2010 to display colours dynamically. It is found to effectively eliminates the use of crayon. It displays a comprehensive range of colours that conventional approach could not offer to user to learn and teach from. Colour emphases a point(s); draw attention to a thing, living or inanimate; satisfies the aesthetic nature of human being. Timer control aids effective and selective teaching and learning through its Interval property, which determines the length of time to which an image or colour is displayed before another one is shown. By circumspectly adjusting the Interval, diverse learners could be cared for. It is quite flexible and useful for applications that will display colours and objects dynamically for a specified time.

The functionalities in dvMechCls class library was exposed and found to be accurate. The open and save dialog boxes are quite useful for opening user's file for uploading data and for getting file name to save results to dynamically as seen in their application to Mechanics. The save menu on figure 3 uses save dialog box to get file name from user and output results to the supply file. Mechanics results obtained are compared with previous ones and found to be accurate[7-8,12-13,15]. The save, open and colour dialog boxes are found to be flexible and highly useful to accomplish frequently and commonly imperative programming tasks.

Physics, Sciences and Engineering drawings will be a welcome development. Also, more areas of colour and timer applications should be explored.

7. REFERENCES

[1] Craig Utley 2001. A Programmer's Guide to Visual Basic. NET. Sams Publishing.

[2] Dave Grundgeiger 2002. Programming Visual Basic.NET. O' Reilly and Associate, Inc.

[3] Evangelos Petroutsos. 2010. Mastering Microsoft Visual Basic 2010. Wiley Publishing Inc.

[4] Jerry Lee Ford, Jr (2009), Microsoft Visual Basic 2008 Express Programming for the Absolute Beginner Course Technology Cengage Learning.

[5] Thearon and Bryan. 2010. Beginning Microsoft Visual Basic 2010. Wiley Publishing Inc.

[6] Tim Patrick. 2008. Programming Visual Basic 2008. O' Reilly and Associate, Inc.

[7] Federick .J Bueche and Eugene Hecht. 2006. Theory and Problems of College Physics Schaum's Outline Series. McGraw-Hill. 10th ed. Pp. 86-92.

[8] Nelkon. M. 1978. Principles of Physics. CSS Bookshops and Hart-Davis Educational. 7th ed. Pp. 87-101.

[9]http://www.w3schools.com/html/html_colors.asp

[10] http://html-color-codes.info/

[11]http://iqa.evergreenps.org/science/phy_science/ma.html

[12]http://library.thinkquest.org/CR0210120/Mechanical%20Advantage.html

[13]http://en.wikipedia.org/wiki/Mechanical_advantage

[14]http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Mechanical_advantage.html

[15]http://www.wisegeek.com/what-is-velocity-ratio.htm