

**FIT
100**

Congratulations!

- ❖ The Binary Search project is done!!
- ❖ REFLECT: This is a significant accomplishment
 - ❑ Understand a fundamental algorithm -- Binary Search
 - ❑ Understand how to translate an algorithm into a program
 - ❑ Have encoded the solution in VB6, showing that you know:
 - + Declaring variables and their types
 - + Assignment and expressions
 - + Conditionals (If-Then-Else)
 - + Event Handlers
 - ❑ Getting it working also shows skill in trouble-shooting and debugging

© Copyright, Larry Snyder, 1999

**FIT
100**

Why Has It Been So Challenging?

- ❖ Algorithm design, programming, application development, etc. are intellectually tough ... why?
 - ❑ There is no "cookbook solution" ... each case has its own logic and requires its own reasoning
 - ❑ The solution must be *exactly right* in every detail
 - ❑ The language used to express the solution (VB6) is new to you, strange and unforgiving
 - ❑ The context -- Windows operating system, the VB6.0 development environment, the UW computing facilities -- is also new and complicated

Learn by example and analogy

© Copyright, Larry Snyder, 1999

Graphics, Randomization and the Beginnings of Animation



Having introduced the main programming ideas for FIT100, it is time to put it all together. The task will be to draw something visually interesting on a form and in the process get experience writing procedures

© Copyright, Larry Snyder, 1999



What You Already Know...

- ❖ How to:
 - ❑ Write procedures
 - ❑ Write procedures that use parameters
 - ❑ Write a procedure that calls another procedure
 - ❑ Use iteration (Do-While Loop)
 - + Which can call a procedure in the body of the loop
 - ❑ Use conditionals (If-Then-Else)

© Copyright, Larry Snyder, 1999

**FIT
100**

What You Need to Learn

- ❖ How to:
 - ❑ Color
 - ❑ Make shapes (lines, boxes, rectangles, circles)
 - ❑ Color in shapes
 - ❑ Use a random number
 - ❑ Convey a sense of motion ...

© Copyright, Larry Snyder, 1999

**FIT
100**

Let's Get Started: Drawing On A Form

- ❖ VB6: The form is logically divided into a grid, and a position is designated by how many grid points it is from the Left and the Top
 - ❑ The upper left corner is position (0,0)
 - ❑ The position (x, y) is x units from Left, and y units from Top
 - ❑ Increasing the x value moves to the right
 - ❑ Unlike graphing, though, increasing the y value moves down
 - ❑ The lower right corner is position (ScaleWidth, ScaleHeight)
 - ❑ To resize the form, change ScaleWidth and ScaleHeight
 - ❑ To make the window the size of the display set WindowState = 2

The Unit is a twip

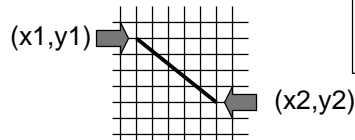
Picture	(None)
RightToLeft	False
ScaleHeight	3195
ScaleLeft	0
ScaleMode	1 - Twip
ScaleTop	0
ScaleWidth	4680
ShowInTaskbar	True
StartPosition	3 - Windows Default
Tag	
Top	0
Visible	True

© Copyright, Larry Snyder, 1999

**FIT
100**

Drawing A Line

- ❖ To draw a line on Form1, call the procedure
Form1.Line (x1, y1) - (x2, y2)



Notice the "minus"
between the two
coordinates





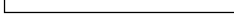
- ❖ If there is only one form, the form name can be *elided* (not be used) (e.g. Line (x1, y1) - (x2, y2))
- ❖ To get a color, follow the positioning information with the specification of the color
Form1.Line (x1, y1) - (x2, y2), RGB(255,255,255)

Draw a white line beginning x1 units from the Left
and y1 units from Top, and extending to a point
x2 units from Left and y2 units down from Top

© Copyright, Larry Snyder, 1999

**FIT
100**

Red, Green and Blue

- ❖ Recall that colors are created on the screen with a combination of three colors of light -- red, green, blue
- ❖ When drawing, one can specify the exact color by calling a procedure, RGB(r, g, b) whose three parameters are the contribution of the three colors in the range 0 -- 255
 - ❑ RGB(0, 0, 0) 
 - ❑ RGB(255, 0, 0) 
 - ❑ RGB(0, 255, 0) 
 - ❑ RGB(0, 0, 255) 
 - ❑ RGB(255,255,255) 

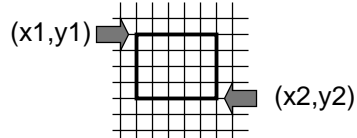
© Copyright, Larry Snyder, 1999

**FIT
100**

Drawing A Box

- ❖ Drawing a rectangle is like drawing a line except that there is a final parameter "B" which stands for "box"

Line (x1, y1) - (x2, y2), RGB(r, g, b), B



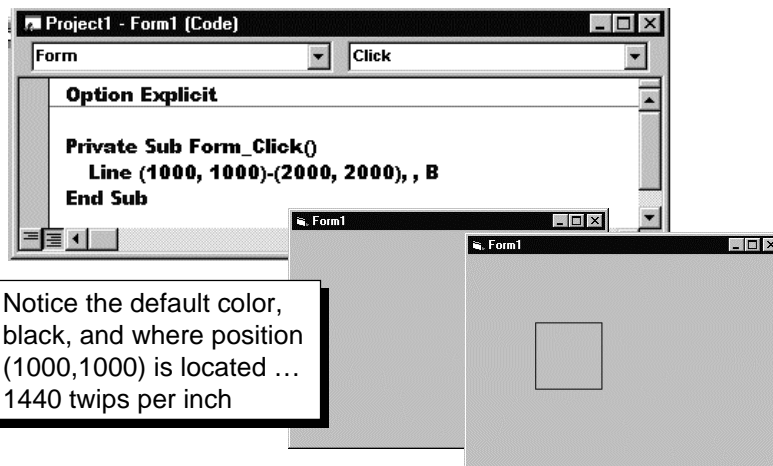
- ❖ A specific fill color can be achieved by having two properties set
 - ❑ FillColor = RGB(r, g, b)
 - ❑ FillStyle = 0 ← Indicates opaque

© Copyright, Larry Snyder, 1999

**FIT
100**

Programming A Rectangle

- ❖ To begin, draw a box in the Form_Click event handler

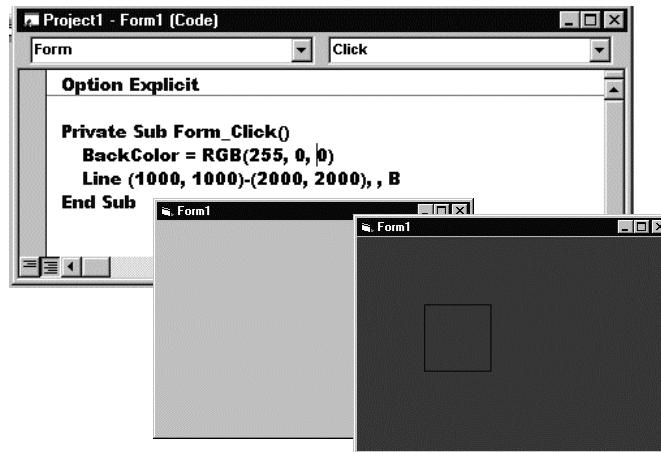


© Copyright, Larry Snyder, 1999

**FIT
100**

Color

- ❖ A black rectangle on a gray form is a little dull ...
- ❖ Set the background color of the form to red

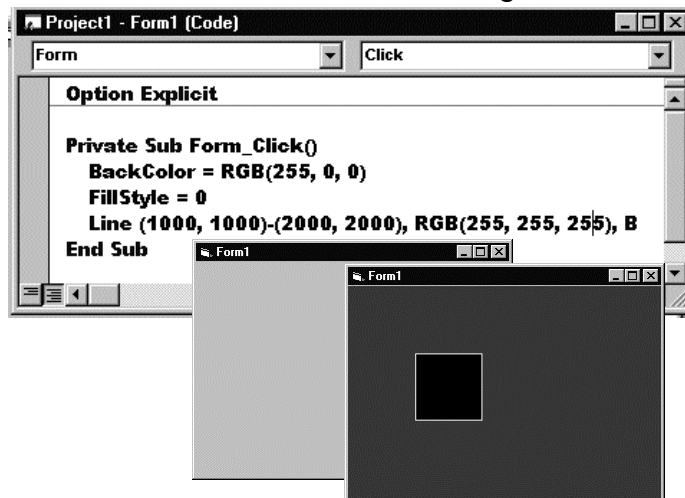


© Copyright, Larry Snyder, 1999

**FIT
100**

Primp Up The Form

- ❖ Make box fill with black and change line to white line

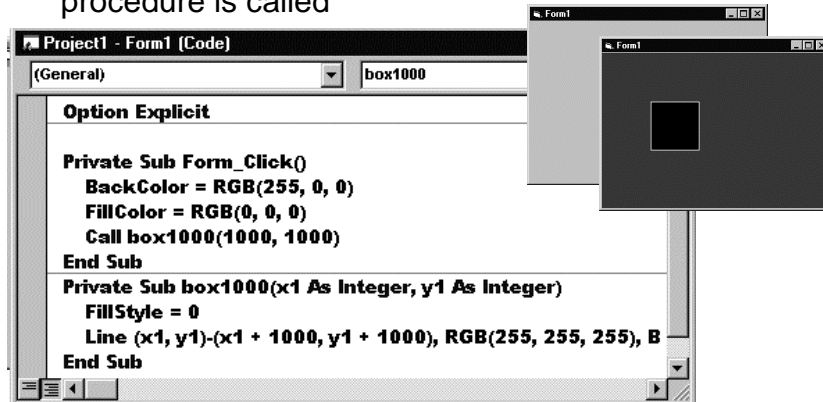


© Copyright, Larry Snyder, 1999

**FIT
100**

Make A Procedure For Box Drawing

- ❖ Draw a 1K×1K box with black fill and a white line
- ❖ The fill color will be whatever color is set when the procedure is called

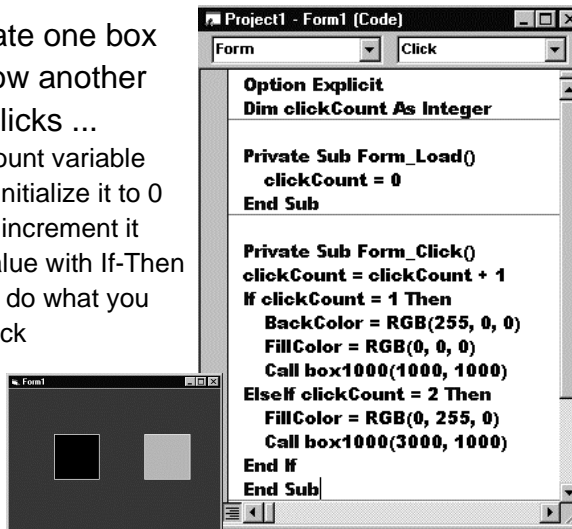


© Copyright, Larry Snyder, 1999

**FIT
100**

More Action, Please

- ❖ Click once, create one box
- ❖ Click again, show another
- ❖ Steps for multiclicks ...
 - Declare clickCount variable
 - In Form_Load initialize it to 0
 - In Form_Click, increment it
 - Then test its value with If-Then
 - For each value do what you want on that click
- ❖ 1st: black box
- ❖ 2nd: green box

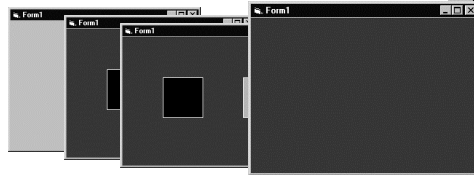


© Copyright, Larry Snyder, 1999

**FIT
100**

Add Another Option

- ❖ Increase the form size to cover whole screen
- ❖ Add another “click” case
 - ❑ WindowState has 3 values
 - ❑ Setting 2 maximizes form
 - ❑ Drawing box from (0,0) to (ScaleWidth,ScaleHeight) covers the entire form ... make it red!



```
Project1 - Form1 (Code)
Form Click
Private Sub Form_Load()
    clickCount = 0
End Sub

Private Sub Form_Click()
    clickCount = clickCount + 1
    If clickCount = 1 Then
        BackColor = RGB(255, 0, 0)
        FillColor = RGB(0, 0, 0)
        Call box1000(1000, 1000)
    ElseIf clickCount = 2 Then
        FillColor = RGB(0, 255, 0)
        Call box1000(3000, 1000)
    ElseIf clickCount = 3 Then
        WindowState = 2
        FillColor = RGB(255, 0, 0)
        Line (0, 0)-(ScaleWidth, ScaleHeight),R
    End If
End Sub
```

© Copyright, Larry Snyder, 1999

**FIT
100**

To Give Motion, Draw On Timer Tick

- ❖ Adding a timer allows changes to be made a regular intervals ... place timer anywhere on form

```
Project1 - Form1 (Code)
tmrClock Timer
Option Explicit
Dim clickCount As Integer
Dim timerCount As Integer

Private Sub Form_Load()
    clickCount = 0
    timerCount = 0
End Sub

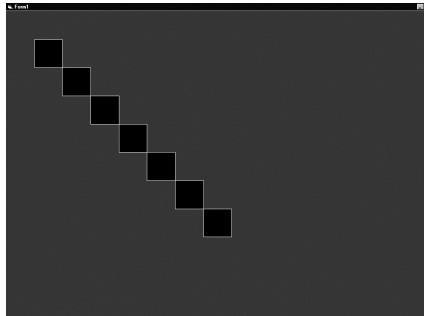
Private Sub tmrClock_Timer()
    timerCount = timerCount + 1
    Call box1000(timerCount * 1000, timerCount * 1000)
End Sub
```

© Copyright, Larry Snyder, 1999

**FIT
100**

Turn Timer On/Off With Click

- ❖ The 4th click starts box draw and the 5th click stops it



```
Project1 - Form1 (Code)
Form Click
Private Sub Form_Click()
clickCount = clickCount + 1
If clickCount = 1 Then
BackColor = RGB(255, 0, 0)
FillColor = RGB(0, 0, 0)
Call box1000(1000, 1000)
ElseIf clickCount = 2 Then
FillColor = RGB(0, 255, 0)
Call box1000(3000, 1000)
ElseIf clickCount = 3 Then
WindowState = 2
FillColor = RGB(255, 0, 0)
Line (0, 0)-(ScaleWidth, ScaleH
FillColor = RGB(0, 0, 0)
ElseIf clickCount = 4 Then
tmrClock.Interval = 200
Else
tmrClock.Interval = 0
End If
```

© Copyright, Larry Snyder, 1999

**FIT
100**

Randomize!

- ❖ Diagonal boxes are boring ... randomize
- ❖ To do this, we'll need to tell the computer to use random numbers...

© Copyright, Larry Snyder, 1999

**FIT
100**

Random Numbers

- ❖ **CONCEPT:** Random numbers are numbers that are independent or unrelated to each other
 - ❑ Coin flipping can produce random bits ... heads (0), tails (1)
 - ❑ Rolling a die can produce random digits ... 1 through 6
 - ❑ Drawing cards from a shuffled deck can produce
 - + Random bits ... red or black
 - + Random digits ... 1 through 4 (Suit)
 - + Random digits ... 1 through 13 (Value)

- ❖ Rnd is VB6's random number generator

© Copyright, Larry Snyder, 1999

**FIT
100**

Using Rnd

1. *Initialize* the Random Number Generator: First, initialize VB6's random number generator Randomize (typically in the Form_Load() event handler)

2. *Generate* a Specific Random Number: Rnd(1) gives a specific Double (large number) between 0 and 1, e.g., 0.410592664
To chose randomly among an integral number of things, say x, multiply Rnd(1) by x and truncate ... the result is a random integer between 0 and x-1
 - ❑ Pick among 6 things:
 - + $\text{Int}(\text{Rnd}(1)*6) \rightarrow \text{Rnd}(1) * 6 \rightarrow 0.410592664 * 6 \rightarrow 2.463555984 \rightarrow 2$
 - + $\text{Int}(\text{Rnd}(1)*6) \rightarrow \text{Rnd}(1) * 6 \rightarrow 0.092154388 * 6 \rightarrow 0.552926328 \rightarrow 0$

To truncate a number
x in VB6 write Int(x)

© Copyright, Larry Snyder, 1999



Randomness? Really?

- ❖ What is randomness?
- ❖ Computers are deterministic – they exactly follow instructions and do exactly what is asked ... how can they do something random?

© Copyright, Larry Snyder, 1999



Randomness? Really?

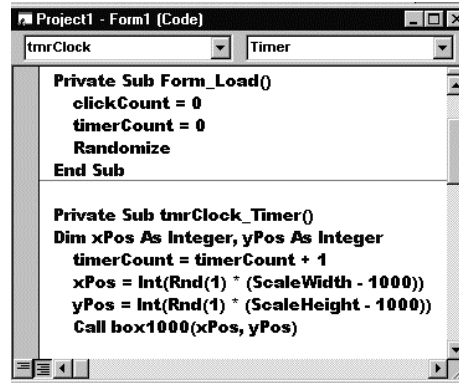
- ❖ What is randomness?
- ❖ Computers are deterministic – they exactly follow instructions and do exactly what is asked ... how can they do something random?
- ❖ CONCEPT: Rnd is “Pseudo-random,” a deterministic computation that produces numbers that appear to be random and pass standard tests for randomness

© Copyright, Larry Snyder, 1999

**FIT
100**

Randomize!

- ❖ Diagonal boxes are boring ... randomize
- ❖ To place boxes randomly,
 - ❑ Initialize Randomize in Form_Load
 - ❑ Declare xPos, yPos in tmrClock
 - ❑ Pick a random number in (0,1) range with a Rnd(1) procedure call
 - ❑ Multiply by the largest size to scale & make Int



```
Project1 - Form1 (Code)
tmrClock Timer

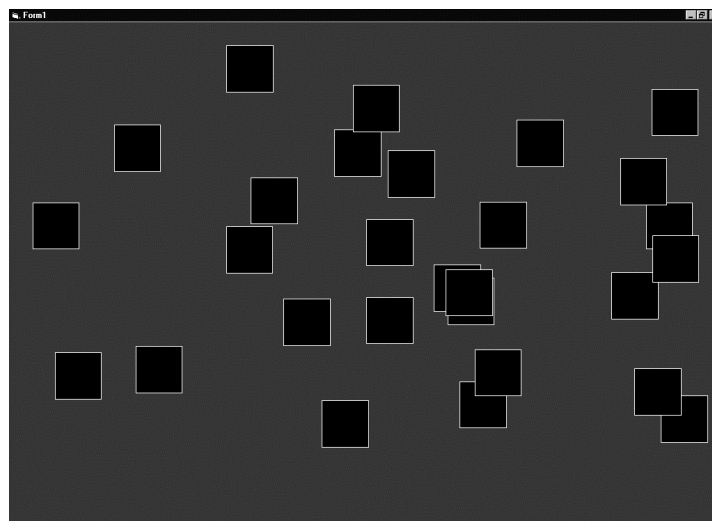
Private Sub Form_Load()
    clickCount = 0
    timerCount = 0
    Randomize
End Sub

Private Sub tmrClock_Timer()
    Dim xPos As Integer, yPos As Integer
    timerCount = timerCount + 1
    xPos = Int(Rnd(1) * (ScaleWidth - 1000))
    yPos = Int(Rnd(1) * (ScaleHeight - 1000))
    Call box1000(xPos, yPos)
End Sub
```

© Copyright, Larry Snyder, 1999

**FIT
100**

One result ...



© Copyright, Larry Snyder, 1999

**FIT
100**

Modify this program to...

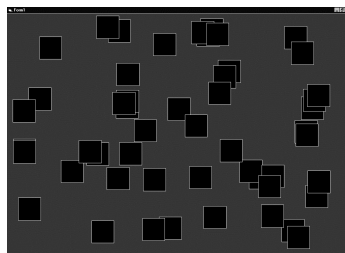
- ❖ Fill the boxes with random shades of red
- ❖ Make the boxes appear more quickly on the screen
- ❖ Make the program draw larger boxes
- ❖ Make the program draw circles on the screen

© Copyright, Larry Snyder, 1999

**FIT
100**

Summary

- ❖ The goal of Project 3 is to explore the idea of creativity and computation
- ❖ Create a (cool!) program with visually interesting output
- ❖ You now know everything you need to get started...



© Copyright, Larry Snyder, 1999