Part 1: Moving PictureBox

Objective: Control Image using WASD keys

Following video: <https://www.youtube.com/watch?v=K0g54G3YxUs>

In visual studio – create new project – Visual Basic – Windows Forms App (.NET Framework)

Stretch form area to larger size.

Add PictureBox from Toolbox column.

Click on arrow of PictureBox to select picture.

Local resource – navigate to image file and select.

Set to stretch image and resize.

Click outside of PictureBox so Form1 for instance.

In Properties column, click Lightning bolt.

Scroll down to Key – KeyDown, select and hit enter.

Opens new area for code and creates following code:

Public Class Form1

Private Sub Form1\_KeyDown(sender As Object, e As KeyEventArgs) Handles MyBase.KeyDown

End Sub

End Class

Add if statements for when W, A, S, or D are pressed.

Code to go within Private Sub:

Public Class Form1

Private Sub Form1\_KeyDown(sender As Object, e As KeyEventArgs) Handles MyBase.KeyDown

If e.KeyCode = Keys.W Then

PictureBox1.Top = PictureBox1.Top = -1

End If

If e.KeyCode = Keys.A Then

PictureBox1.Left = PictureBox1.Left = -1

End If

If e.KeyCode = Keys.S Then

PictureBox1.Top = PictureBox1.Top = 1

End If

If e.KeyCode = Keys.D Then

PictureBox1.Left = PictureBox1.Left = 1

End If

End Sub

End Class

Press Ctrl+Shift+S to save all.

Top = -1 will decrease space between PictureBox1 and Form1’s top edge by 1 pixel

Left = -1 will decrease space between PictureBox1 and Form1’s left edge by 1 pixel

Top = 1 will increase space between PictureBox1 and Form1’s top edge by 1 pixel

Left = 1 will increase space between PictureBox1 and Form1’s left edge by 1 pixel

Click Build – Build Solution.

Click Start to run program.

TEST: When program runs press W, A, S and D keys to see if they function as expected.

RESULT: Does not run as expected.

What happens?

On keypress W: PictureBox1 moves to top edge of Form1.

On keypress A: PictureBox1 moves to left edge of Form1.

On keypress S: PictureBox1 moves to top edge of Form1.

On keypress D: PictureBox1 moves to left edge of Form1.

Why does this happen?

Are there build errors? No, there are no warnings or red underlines indicating errors and no errors found when I pressed Build Solution.

What else could be the cause if no errors are detected? Answer, human error.

It happens because the code I’ve entered is wrong.

What I want the code to do is increment PictureBox1’s position by 1 pixel in the direction the key represents. What I have mistakenly done is re-assign the position of PictureBox1 along it’s x or y axis. It’s doing this because I have entered an = sign before the amount.

So:

Top = -1 Moves PictureBox1 to Form1’s top edge (x, 0)

(The form starts at 0, 0 so won’t position PictureBox1 in negative range)

Left = -1 Moves PictureBox1 to Form1’s top edge (0, y)

Top = 1 Moves PictureBox1 to Form1’s top edge + 1 pixel (x, 1)

Left = 1 Moves PictureBox1 to Form1’s left edge + 1 pixel (1, y)

If I remove the = sign in each If statement it should behave how I want it.

So, the code now looks like:

If e.KeyCode = Keys.W Then

PictureBox1.Top = PictureBox1.Top -1

End If

If e.KeyCode = Keys.A Then

PictureBox1.Left = PictureBox1.Left -1

End If

If e.KeyCode = Keys.S Then

PictureBox1.Top = PictureBox1.Top 1

End If

If e.KeyCode = Keys.D Then

PictureBox1.Left = PictureBox1.Left 1

End If

Before running this the error list is finding fault with the integer “1” for the S and D lines of code. It’s because I haven’t declared a + or - operator before it. I assumed that a number without a + before it would be interpreted as positive anyway but this isn’t the case.

When, declaring an object’s position you don’t have to declare the + operator if the object is within a positive number range, so PictureBox1.Left = 1 is valid.

If you want to increment an objects position then you do have to declare a + operator if you want to increment positively. E.g. to increment PictureBox1.Left by 1 pixel, becomes PictureBox1.Left + 1.

So, to fix the code I will change it to the following:

Public Class Form1

Private Sub Form1\_KeyDown(sender As Object, e As KeyEventArgs) Handles MyBase.KeyDown

If e.KeyCode = Keys.W Then

PictureBox1.Top = PictureBox1.Top - 1

End If

If e.KeyCode = Keys.A Then

PictureBox1.Left = PictureBox1.Left - 1

End If

If e.KeyCode = Keys.S Then

PictureBox1.Top = PictureBox1.Top + 1

End If

If e.KeyCode = Keys.D Then

PictureBox1.Left = PictureBox1.Left + 1

End If

End Sub

End Class

Press Ctrl+Shift+S to save all.

Click Build – Rebuild Solution.

Click Start to run program.

TEST: When program runs press W, A, S and D keys to see if they function as expected.

RESULT: Pass.

What happens?

On keypress W: PictureBox1 moves up by 1 pixel.

On keypress A: PictureBox1 moves left by 1 pixel.

On keypress S: PictureBox1 moves down by 1 pixel.

On keypress D: PictureBox1 moves right by 1 pixel.

Objective Met

Future Consideration: Controlling the image with arrow keys or 2,4,8,6 on number pad would be a nice addition in functionality. For single player only though. If a two player mode was added arrow keys would be needed to control player 2’s character.