I still get requests from time-to-time for the Blitz Basic and PureBasic books I wrote some years back, so here they are for download! They both have all the source code that I included in the original too. Hopefully it's everything as it's all I've got! ;)

Please note I am no longer supporting this book and I'm sure that PureBasic has moved on in language structure since this was created. So this is really just up here for nostalgia.

Click here for the *Learn to Program 2D Games In Blitz Basic book*

Click here for the *Programming 2D Scrolling Games for PureBasic book*

BLITZ BASIC Book Snafus

Here is where you can find any book problems that have been found and how to resolve them.

Page 68: Type Identifiers Snafu

Turns out that there's something in the Types section of the book that's totally wrong. On page 68, I put the following:

```
So, why do we need an identifier anyway?  Well, we may have different types of information that
weâ€™re tracking that require the same values.  Letâ€™s use an example of space ships.  You have your
good guy ship and all of your bad guy ships, plus youâ€™ve got freighters and destroyers.  They all
have a name, laser power, shield power, armor, missile compliment, speed, etc.  So why would you
want to create a different TYPE for each one? Just use one and use different identifiers.

Type Ships
   Field Name$             ; name of this ship
   Field LaserPower%       ; 1-20 points per hit
   Field Armor%            ; 100-150 points
   Field ShieldPower%      ; 150-300 points-adds to Armor
   Field Missiles%         ; 5-50 depending on ship type
   Field TopSpeed#         ; 2.00â€“3.50 based on ship type
End Type

Now, instead of having to create three (or many more depending on the number of ship types you have)
complete TYPE constructs, you can use three separate identifiers.

PlayerFighter.Ships = New Ships
BadguyFighter.Ships = New Ships
Freighter.Ships = New Ships
Destroyer.Ships = New Ships

Each one can now be accessed directly and assigned values that will only be applicable to the
identifier that they are associated with.  That saves some major headache, believe me!
```

Well, the headache is back. Turns out that while you can certainly do what I'm showing above, you *can't* use the identifiers to snag them directly. I thought I'd tested this, but apparently I missed it.

If, for example, you filled up those Types with 100 various ships and you wanted to print the names of ONLY the Destroyers...you can't without somehow flagging them.

So the following:

```
For Destroyers.Ships = Each Ships
    Print Destroyers\Name$
Next
```

Would print ALL 100 ships, not just the Destroyers.

So, here's ONE way to handle it using flagging. I know there are other ways to flag, but this way seems easy enough for most to grasp:

```
Graphics 640,480
SetBuffer BackBuffer()

;;;;;;;;;;;;;;;;;;;;;;;;;;;;; setup consts for ship flags;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Const PlayerFighter = 0
Const BadGuyFighter = 1
Const Freighter = 2
Const Destroyer = 3

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; setup a currentship flag ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Global CurrentShip = PlayerFighter

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; type defination ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Type Ships
     Field ShipType     ; What kind of ship is it?
     Field Name$        ; name of this ship
End Type

;;;;;;;;;;;;;;;;;;;;;;;;;;;;; initialize a bunch of ships;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
For i = 0 To 24
    Ship.Ships = New Ships
    Ship\ShipType = PlayerFighter
    Ship\Name = "PlayerFighter" + i
Next

For i = 0 To 24
    Ship.Ships = New Ships
    Ship\ShipType = BadGuyFighter
    Ship\Name = "BadGuyFighter" + i
Next

For i = 0 To 24
    Ship.Ships = New Ships
    Ship\ShipType = Freighter
    Ship\Name = "Freighter" + i
Next

For i = 0 To 24
    Ship.Ships = New Ships
    Ship\ShipType = Destroyer
    Ship\Name = "Destroyer" + i
Next

;;;;;;;;;;;;;;;;;;;;;;;;;;;;; main program;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
While Not KeyHit(1)
      Cls
      If KeyHit(2)
         CurrentShip = PlayerFighter
      EndIf

      If KeyHit(3)
         CurrentShip = BadGuyFighter
      EndIf

      If KeyHit(4)
         CurrentShip = Freighter
      EndIf

      If KeyHit(5)
         CurrentShip = Destroyer
      EndIf

      Locate(0,0)
      Print "1=PlayerShips, 2=BadGuyShips, 3=Freighters, 4=Destroyers, ESC = Quit"
      Print
      For Ship.Ships = Each Ships
          If Ship\ShipType = CurrentShip
             Print Ship\Name
          EndIf
      Next

      Flip
Wend

End
```

Page 94: Float Size is NOT 8 Bytes

When I was writing this text I'd talked to Mark about the size of the FLOAT and I thought he'd said 8 bytes. I know that FLOATs are usually 4 bytes, but since Blitz didn't have a DOUBLE I thought it best to verify that FLOATs weren't acting as DOUBLEs.

Well, I must have been a little nutty that day cause I could swear he said they were 8 bytes. I even recall testing things out in the DataBank area and confirming that myself, go figure.

Anyway, bottom-line is that FLOATs are 4 bytes wide, not 8. Thus, you should use (see Page 95):

```
PokeFloat MyBank,0,100.175
PokeFloat MyBank,4,200.25
```

...instead of using "PokeFloat MyBank,8,200.25" on the second line. And, also, make sure to use 4-byte offsets when using PeekFloat.

Chapter 20, starting on page 205: Edge Independent Scrolling

I've received a few emails over the last year critiquing the technique that I was using to handle that stuff. The critics were right. The technique that I used was meant to be a catch-all, covering the various types of scrolling and such. Unfortunately, one often finds that writing a catchall doesn't always turn out the way you'd expect. In this case there were little annoyances that I thought wouldn't be such a big deal, but they kinda are. The screen will scroll to a point that is beyond the edge on occassion, and it even seems that the sprite itself will act erratically under certain circumstances. Now, I would love to chalk this up as being a little fun A.I. that I included in, but alas I'm not bright enough to pull that off.

So please refer to the following link for full details on the new method I'm using, and snag the code too:

Handling Sprites on Scrolling Tiled Backgrounds with Collisions

Hope that helps!