

Basic Game Programming with FreeBASIC

The following sample codes are used to demonstrate only the very basics of Game Programming, which are:

1. [Initializing Graphics library and creating a Screen](#)
2. [Loading and Displaying an Image](#)
3. [Making a simple Animation](#)
4. [Putting it all together](#)
5. [Cleaning up the code](#)

Before you start reading, please bear in mind that I am not claiming that these are the best approach in doing the tasks listed above. Just like any other programming, each problem can be done with different algorithms. My approach is not a standard design pattern for game programming.

I. Initializing Graphics library and creating a Screen

Above anything else, we need to have a screen to display our game on. The screen is also known as **window/form** in applications development.

```
#include "fbgfx.bi"
Using FB

ScreenRes 320, 240, 32

Do
    Sleep 10
Loop Until MultiKey(SC_ESCAPE)
```

II. Loading and Displaying an Image

Having a screen, we can now load and display an image.

```
#include "fbgfx.bi"
Using FB

Dim stand As Any Ptr

Const NULL As Any Ptr = 0

Dim x as Integer
Dim y as Integer
```

```

ScreenRes 320, 240, 32

x = 100
y = 150

stand = ImageCreate(47, 88)

BLoad "char1/stand/r/1.bmp", stand

If stand = NULL _
    Then
        Print "Image creation failed!"
        Sleep
    End
End If

Do
    Put (x,y), stand

    Sleep 10
Loop Until MultiKey(SC_ESCAPE)

ImageDestroy stand

```

II.b. Making the object transparent

We remove the unwanted color, *magenta*, by setting it in **ImageCreate** and toggling **Trans** onto **Put** command.

```

#include "fbgfx.bi"
Using FB

Dim stand As Any Ptr

Const NULL As Any Ptr = 0

Dim x as Integer
Dim y as Integer

ScreenRes 320, 240, 32

x = 100
y = 150

stand = ImageCreate(47, 88, RGB(255,0,255))

BLoad "char1/stand/r/1.bmp", stand

If stand = NULL _
    Then
        Print "Image creation failed!"
        Sleep
    End
End If

Do

```

```

    Put (x,y), stand,Trans

    Sleep 10
Loop Until MultiKey(SC_ESCAPE)

ImageDestroy stand

```

III. Making a simple Animation

To make a simple animation, just load another image. Simply declare a variable that will be the pointer to which image is to be displayed.

```

#include "fbgfx.bi"
Using FB

Dim stand1 As Any Ptr
Dim stand2 As Any Ptr
    Dim standCtr As Integer

Const NULL As Any Ptr = 0

Dim x as Integer
Dim y as Integer

ScreenRes 320, 240, 32

x = 100
y = 150

stand1 = ImageCreate(47, 88, RGB(255,0,255))
stand2 = ImageCreate(47, 88, RGB(255,0,255))

BLoad "char1/stand/r/1.bmp", stand1
BLoad "char1/stand/r/2.bmp", stand2

If stand1 = NULL Or stand2 = NULL _
    Then
    Print "Image creation failed!"
    Sleep
    End
End If

Do
    Cls
    standCtr = standCtr + 1
    If standCtr >= 60 Then standCtr = 0

    If standCtr < 30 Then
        Put (x,y), stand1, Trans
    ElseIf standCtr < 60 Then
        Put (x,y), stand2, Trans
    End If

    Sleep 10

```

```
Loop Until MultiKey(SC_ESCAPE)
```

```
ImageDestroy stand1  
ImageDestroy stand2
```

III.b Animation with Basic Interaction

To know which animation-set is to be displayed, we would need *flags* – commonly known as *Boolean*.

```
#include "fbgfx.bi"  
Using FB  
  
Dim standR1 As Any Ptr  
Dim standR2 As Any Ptr  
    Dim standL1 As Any Ptr  
    Dim standL2 As Any Ptr  
        Dim standCtr As Integer  
  
    Dim face As Byte  
  
Const NULL As Any Ptr = 0  
Const FACE_RIGHT As Integer = 0  
Const FACE_LEFT As Integer = 1  
  
Dim x as Integer  
Dim y as Integer  
  
ScreenRes 320, 240, 32  
  
x = 100  
y = 150  
  
standR1 = ImageCreate(47, 88, RGB(255,0,255))  
standR2 = ImageCreate(47, 88, RGB(255,0,255))  
    standL1 = ImageCreate(47, 88, RGB(255,0,255))  
    standL2 = ImageCreate(47, 88, RGB(255,0,255))  
  
BLoad "char1/stand/r/1.bmp", standR1  
BLoad "char1/stand/r/2.bmp", standR2  
    BLoad "char1/stand/l/1.bmp", standL1  
    BLoad "char1/stand/l/2.bmp", standL2  
  
If standR1 = NULL Or standR2 = NULL _  
    Or standL1 = NULL Or standL2 = NULL _  
    Then  
    Print "Image creation failed!"  
    Sleep  
    End  
End If  
  
Do  
    Cls  
    standCtr = standCtr + 1  
    If standCtr >= 60 Then standCtr = 0  
  
    If standCtr < 30 Then
```

```

        If face = FACE_RIGHT Then
            Put (x,y), standR1, Trans
        Else
            Put (x,y), standL1, Trans
        End If
    ElseIf standCtr < 60 Then
        If face = FACE_RIGHT Then
            Put (x,y), standR2, Trans
        Else
            Put (x,y), standL2, Trans
        End If
    End If

    standCtr = standCtr + 1

    If MultiKey(SC_RIGHT) Then
        face = FACE_RIGHT
    ElseIf MultiKey(SC_LEFT) Then
        face = FACE_LEFT
    End If

    Sleep 10
Loop Until MultiKey(SC_ESCAPE)

ImageDestroy standR1
ImageDestroy standR2
    ImageDestroy standL1
    ImageDestroy standL2

```

III.c Animated-movement with Basic Interaction

Flagging can have multiple states (or values) which is used in movement animation.

```

#include "fbgfx.bi"
Using FB

Dim standR1 As Any Ptr
Dim standR2 As Any Ptr
    Dim standL1 As Any Ptr
    Dim standL2 As Any Ptr
    Dim standCtr As Integer

Dim runR1 As Any Ptr
Dim runR2 As Any Ptr
Dim runR3 As Any Ptr
Dim runR4 As Any Ptr
Dim runR5 As Any Ptr
Dim runR6 As Any Ptr
    Dim runL1 As Any Ptr
    Dim runL2 As Any Ptr
    Dim runL3 As Any Ptr
    Dim runL4 As Any Ptr
    Dim runL5 As Any Ptr
    Dim runL6 As Any Ptr
    Dim runCtr As Integer

```

```

Dim face As Byte
Dim state As Byte

Const NULL As Any Ptr = 0
Const FACE_RIGHT As Integer = 0
Const FACE_LEFT As Integer = 1

Const STATE_STAND As Integer = 0
Const STATE_RUN As Integer = 1

Dim x as Integer
Dim y as Integer

ScreenRes 320, 240, 32

x = 100
y = 150

standR1 = ImageCreate(47, 88, RGB(255,0,255))
standR2 = ImageCreate(47, 88, RGB(255,0,255))
    standL1 = ImageCreate(47, 88, RGB(255,0,255))
    standL2 = ImageCreate(47, 88, RGB(255,0,255))

runR1 = ImageCreate(72, 77, RGB(255,0,255))
runR2 = ImageCreate(72, 77, RGB(255,0,255))
runR3 = ImageCreate(72, 77, RGB(255,0,255))
runR4 = ImageCreate(72, 77, RGB(255,0,255))
runR5 = ImageCreate(72, 77, RGB(255,0,255))
runR6 = ImageCreate(72, 77, RGB(255,0,255))
    runL1 = ImageCreate(72, 77, RGB(255,0,255))
    runL2 = ImageCreate(72, 77, RGB(255,0,255))
    runL3 = ImageCreate(72, 77, RGB(255,0,255))
    runL4 = ImageCreate(72, 77, RGB(255,0,255))
    runL5 = ImageCreate(72, 77, RGB(255,0,255))
    runL6 = ImageCreate(72, 77, RGB(255,0,255))

BLoad "char1/stand/r/1.bmp", standR1
BLoad "char1/stand/r/2.bmp", standR2
    BLoad "char1/stand/l/1.bmp", standL1
    BLoad "char1/stand/l/2.bmp", standL2

BLoad "char1/run/r/1.bmp", runR1
BLoad "char1/run/r/2.bmp", runR2
BLoad "char1/run/r/3.bmp", runR3
BLoad "char1/run/r/4.bmp", runR4
BLoad "char1/run/r/5.bmp", runR5
BLoad "char1/run/r/6.bmp", runR6
    BLoad "char1/run/l/1.bmp", runL1
    BLoad "char1/run/l/2.bmp", runL2
    BLoad "char1/run/l/3.bmp", runL3
    BLoad "char1/run/l/4.bmp", runL4
    BLoad "char1/run/l/5.bmp", runL5
    BLoad "char1/run/l/6.bmp", runL6

If standR1 = NULL Or standR2 = NULL _
    Or standL1 = NULL Or standL2 = NULL _
    Or runR1 = NULL Or runR2 = NULL Or runR3 = NULL Or runR4 = NULL Or runR5 =

```

```

NULL Or runR6 = NULL
    Or runL1 = NULL Or runL2 = NULL Or runL3 = NULL Or runL4 = NULL Or
runL5 = NULL Or runL6 = NULL
    Then
    Print "Image creation failed!"
    Sleep
    End
End If

Do
    Cls
    If state = STATE_STAND Then
        standCtr = standCtr + 1
        If standCtr >= 60 Then standCtr = 0

        If standCtr < 30 Then
            If face = FACE_RIGHT Then
                Put (x,y), standR1, Trans
            Else
                Put (x,y), standL1, Trans
            End If
        ElseIf standCtr < 60 Then
            If face = FACE_RIGHT Then
                Put (x,y), standR2, Trans
            Else
                Put (x,y), standL2, Trans
            End If
        End If
    ElseIf state = STATE_RUN Then
        runCtr = runCtr + 1
        If runCtr >= 60 Then runCtr = 0

        If runCtr < 10 Then
            If face = FACE_RIGHT Then
                Put (x,y), runR1, Trans
            Else
                Put (x,y), runL1, Trans
            End If
        ElseIf runCtr < 20 Then
            If face = FACE_RIGHT Then
                Put (x,y), runR2, Trans
            Else
                Put (x,y), runL2, Trans
            End If
        ElseIf runCtr < 30 Then
            If face = FACE_RIGHT Then
                Put (x,y), runR3, Trans
            Else
                Put (x,y), runL3, Trans
            End If
        ElseIf runCtr < 40 Then
            If face = FACE_RIGHT Then
                Put (x,y), runR4, Trans
            Else
                Put (x,y), runL4, Trans
            End If
        ElseIf runCtr < 50 Then

```

```

        If face = FACE_RIGHT Then
            Put (x,y), runR5, Trans
        Else
            Put (x,y), runL5, Trans
        End If
    ElseIf runCtr < 60 Then
        If face = FACE_RIGHT Then
            Put (x,y), runR6, Trans
        Else
            Put (x,y), runL6, Trans
        End If
    End If
End If

If MultiKey(SC_RIGHT) Then
    state = STATE_RUN
    x = x + 2
    face = FACE_RIGHT
ElseIf MultiKey(SC_LEFT) Then
    state = STATE_RUN
    x = x - 2
    face = FACE_LEFT
Else
    state = STATE_STAND
End If

Sleep 10
Loop Until MultiKey(SC_ESCAPE)

ImageDestroy standR1
ImageDestroy standR2
    ImageDestroy standL1
    ImageDestroy standL2

ImageDestroy runR1
ImageDestroy runR2
ImageDestroy runR3
ImageDestroy runR4
ImageDestroy runR5
ImageDestroy runR6
    ImageDestroy runL1
    ImageDestroy runL2
    ImageDestroy runL3
    ImageDestroy runL4
    ImageDestroy runL5
    ImageDestroy runL6

```

IV. Putting it all together

Let us load then animate all remaining images and have something to toy with.

```

#include "fbgfx.bi"
Using FB

```

```

Dim standR1 As Any Ptr
Dim standR2 As Any Ptr
    Dim standL1 As Any Ptr
    Dim standL2 As Any Ptr
        Dim standCtr As Integer

Dim runR1 As Any Ptr
Dim runR2 As Any Ptr
Dim runR3 As Any Ptr
Dim runR4 As Any Ptr
Dim runR5 As Any Ptr
Dim runR6 As Any Ptr
    Dim runL1 As Any Ptr
    Dim runL2 As Any Ptr
    Dim runL3 As Any Ptr
    Dim runL4 As Any Ptr
    Dim runL5 As Any Ptr
    Dim runL6 As Any Ptr
        Dim runCtr As Integer

Dim kickR1 As Any Ptr
Dim kickR2 As Any Ptr
Dim kickR3 As Any Ptr
Dim kickR4 As Any Ptr
    Dim kickL1 As Any Ptr
    Dim kickL2 As Any Ptr
    Dim kickL3 As Any Ptr
    Dim kickL4 As Any Ptr
        Dim kickCtr As Integer

Dim kickhiR1 As Any Ptr
Dim kickhiR2 As Any Ptr
Dim kickhiR3 As Any Ptr
Dim kickhiR4 As Any Ptr
    Dim kickhiL1 As Any Ptr
    Dim kickhiL2 As Any Ptr
    Dim kickhiL3 As Any Ptr
    Dim kickhiL4 As Any Ptr

Dim punchR1 As Any Ptr
Dim punchR2 As Any Ptr
Dim punchR3 As Any Ptr
Dim punchR4 As Any Ptr
    Dim punchL1 As Any Ptr
    Dim punchL2 As Any Ptr
    Dim punchL3 As Any Ptr
    Dim punchL4 As Any Ptr
        Dim punchCtr As Integer

Dim jumpR1 As Any Ptr
Dim jumpR2 As Any Ptr
    Dim jumpL1 As Any Ptr
    Dim jumpL2 As Any Ptr
        Dim jumpCtr As Integer

Dim face As Byte
Dim state As Byte

```

```

Dim jump As Byte

Const NULL As Any Ptr = 0
Const FACE_RIGHT As Integer = 0
Const FACE_LEFT As Integer = 1

Const STATE_STAND As Integer = 0
Const STATE_RUN As Integer = 1
Const STATE_KICK As Integer = 2
Const STATE_KICK_HI As Integer = 3
Const STATE_PUNCH As Integer = 4
Const STATE_JUMP As Integer = 5

Dim x as Integer
Dim y as Integer

ScreenRes 320, 240, 32

x = 100
y = 150

standR1 = ImageCreate(47, 88, RGB(255,0,255))
standR2 = ImageCreate(47, 88, RGB(255,0,255))
    standL1 = ImageCreate(47, 88, RGB(255,0,255))
    standL2 = ImageCreate(47, 88, RGB(255,0,255))

runR1 = ImageCreate(72, 77, RGB(255,0,255))
runR2 = ImageCreate(72, 77, RGB(255,0,255))
runR3 = ImageCreate(72, 77, RGB(255,0,255))
runR4 = ImageCreate(72, 77, RGB(255,0,255))
runR5 = ImageCreate(72, 77, RGB(255,0,255))
runR6 = ImageCreate(72, 77, RGB(255,0,255))
    runL1 = ImageCreate(72, 77, RGB(255,0,255))
    runL2 = ImageCreate(72, 77, RGB(255,0,255))
    runL3 = ImageCreate(72, 77, RGB(255,0,255))
    runL4 = ImageCreate(72, 77, RGB(255,0,255))
    runL5 = ImageCreate(72, 77, RGB(255,0,255))
    runL6 = ImageCreate(72, 77, RGB(255,0,255))

kickR1 = ImageCreate(84, 82, RGB(255,0,255))
kickR2 = ImageCreate(84, 82, RGB(255,0,255))
kickR3 = ImageCreate(84, 82, RGB(255,0,255))
kickR4 = ImageCreate(84, 82, RGB(255,0,255))
    kickL1 = ImageCreate(84, 82, RGB(255,0,255))
    kickL2 = ImageCreate(84, 82, RGB(255,0,255))
    kickL3 = ImageCreate(84, 82, RGB(255,0,255))
    kickL4 = ImageCreate(84, 82, RGB(255,0,255))

kickhiR1 = ImageCreate(58, 97, RGB(255,0,255))
kickhiR2 = ImageCreate(58, 97, RGB(255,0,255))
kickhiR3 = ImageCreate(58, 97, RGB(255,0,255))
kickhiR4 = ImageCreate(58, 97, RGB(255,0,255))
    kickhiL1 = ImageCreate(58, 97, RGB(255,0,255))
    kickhiL2 = ImageCreate(58, 97, RGB(255,0,255))
    kickhiL3 = ImageCreate(58, 97, RGB(255,0,255))
    kickhiL4 = ImageCreate(58, 97, RGB(255,0,255))

```

```

punchR1 = ImageCreate (70, 85, RGB (255,0,255))
punchR2 = ImageCreate (70, 85, RGB (255,0,255))
punchR3 = ImageCreate (70, 85, RGB (255,0,255))
punchR4 = ImageCreate (70, 85, RGB (255,0,255))
    punchL1 = ImageCreate (70, 85, RGB (255,0,255))
    punchL2 = ImageCreate (70, 85, RGB (255,0,255))
    punchL3 = ImageCreate (70, 85, RGB (255,0,255))
    punchL4 = ImageCreate (70, 85, RGB (255,0,255))

jumpR1 = ImageCreate (53, 83, RGB (255,0,255))
jumpR2 = ImageCreate (53, 83, RGB (255,0,255))
    jumpL1 = ImageCreate (53, 83, RGB (255,0,255))
    jumpL2 = ImageCreate (53, 83, RGB (255,0,255))

BLoad "char1/stand/r/1.bmp", standR1
BLoad "char1/stand/r/2.bmp", standR2
    BLoad "char1/stand/l/1.bmp", standL1
    BLoad "char1/stand/l/2.bmp", standL2

BLoad "char1/run/r/1.bmp", runR1
BLoad "char1/run/r/2.bmp", runR2
BLoad "char1/run/r/3.bmp", runR3
BLoad "char1/run/r/4.bmp", runR4
BLoad "char1/run/r/5.bmp", runR5
BLoad "char1/run/r/6.bmp", runR6
    BLoad "char1/run/l/1.bmp", runL1
    BLoad "char1/run/l/2.bmp", runL2
    BLoad "char1/run/l/3.bmp", runL3
    BLoad "char1/run/l/4.bmp", runL4
    BLoad "char1/run/l/5.bmp", runL5
    BLoad "char1/run/l/6.bmp", runL6

BLoad "char1/kick/r/1.bmp", kickR1
BLoad "char1/kick/r/2.bmp", kickR2
BLoad "char1/kick/r/3.bmp", kickR3
BLoad "char1/kick/r/4.bmp", kickR4
    BLoad "char1/kick/l/1.bmp", kickL1
    BLoad "char1/kick/l/2.bmp", kickL2
    BLoad "char1/kick/l/3.bmp", kickL3
    BLoad "char1/kick/l/4.bmp", kickL4

BLoad "char1/kickhi/r/1.bmp", kickhiR1
BLoad "char1/kickhi/r/2.bmp", kickhiR2
BLoad "char1/kickhi/r/3.bmp", kickhiR3
BLoad "char1/kickhi/r/4.bmp", kickhiR4
    BLoad "char1/kickhi/l/1.bmp", kickhiL1
    BLoad "char1/kickhi/l/2.bmp", kickhiL2
    BLoad "char1/kickhi/l/3.bmp", kickhiL3
    BLoad "char1/kickhi/l/4.bmp", kickhiL4

BLoad "char1/punch/r/1.bmp", punchR1
BLoad "char1/punch/r/2.bmp", punchR2
BLoad "char1/punch/r/3.bmp", punchR3
BLoad "char1/punch/r/4.bmp", punchR4
    BLoad "char1/punch/l/1.bmp", punchL1
    BLoad "char1/punch/l/2.bmp", punchL2
    BLoad "char1/punch/l/3.bmp", punchL3

```

```

    BLoad "char1/punch/l/4.bmp", punchL4
BLoad "char1/jump/r/1.bmp", jumpR1
BLoad "char1/jump/r/2.bmp", jumpR2
    BLoad "char1/jump/l/1.bmp", jumpL1
    BLoad "char1/jump/l/2.bmp", jumpL2

If standR1 = NULL Or standR2 = NULL _
    Or standL1 = NULL Or standL2 = NULL _
    Or jumpR1 = NULL Or jumpR2 = NULL _
    Or jumpL1 = NULL Or jumpL2 = NULL _
    Or runR1 = NULL Or runR2 = NULL Or runR3 = NULL Or runR4 = NULL Or runR5 =
NULL Or runR6 = NULL _
    Or runL1 = NULL Or runL2 = NULL Or runL3 = NULL Or runL4 = NULL Or
runL5 = NULL Or runL6 = NULL _
    Or kickR1 = NULL Or kickR2 = NULL Or kickR3 = NULL Or kickR4 = NULL _
    Or kickL1 = NULL Or kickL2 = NULL Or kickL3 = NULL Or kickL4 = NULL _
    Or kickhiR1 = NULL Or kickhiR2 = NULL Or kickhiR3 = NULL Or kickhiR4 = NULL
_
    Or kickhiL1 = NULL Or kickhiL2 = NULL Or kickhiL3 = NULL Or kickhiL4 =
NULL _
    Or punchR1 = NULL Or punchR2 = NULL Or punchR3 = NULL Or punchR4 = NULL _
    Or punchL1 = NULL Or punchL2 = NULL Or punchL3 = NULL Or punchL4 =
NULL _
    Then
    Print "Image creation failed!"
    Sleep
    End
End If

Do
    Cls
    If state = STATE_STAND Then
        standCtr = standCtr + 1
        If standCtr >= 60 Then standCtr = 0

        If standCtr < 30 Then
            If face = FACE_RIGHT Then
                Put (x,y), standR1, Trans
            Else
                Put (x,y), standL1, Trans
            End If
        ElseIf standCtr < 60 Then
            If face = FACE_RIGHT Then
                Put (x,y), standR2, Trans
            Else
                Put (x,y), standL2, Trans
            End If
        End If
    ElseIf state = STATE_RUN Then
        runCtr = runCtr + 1
        If runCtr >= 60 Then runCtr = 0

        If runCtr < 10 Then
            If face = FACE_RIGHT Then
                Put (x,y), runR1, Trans
            Else

```

```

        Put (x,y), runL1, Trans
    End If
ElseIf runCtr < 20 Then
    If face = FACE_RIGHT Then
        Put (x,y), runR2, Trans
    Else
        Put (x,y), runL2, Trans
    End If
ElseIf runCtr < 30 Then
    If face = FACE_RIGHT Then
        Put (x,y), runR3, Trans
    Else
        Put (x,y), runL3, Trans
    End If
ElseIf runCtr < 40 Then
    If face = FACE_RIGHT Then
        Put (x,y), runR4, Trans
    Else
        Put (x,y), runL4, Trans
    End If
ElseIf runCtr < 50 Then
    If face = FACE_RIGHT Then
        Put (x,y), runR5, Trans
    Else
        Put (x,y), runL5, Trans
    End If
ElseIf runCtr < 60 Then
    If face = FACE_RIGHT Then
        Put (x,y), runR6, Trans
    Else
        Put (x,y), runL6, Trans
    End If
End If
ElseIf state = STATE_KICK Then
    If kickCtr < 15 Then
        If face = FACE_RIGHT Then
            Put (x,y+5), kickR1, Trans
        Else
            Put (x,y+5), kickL1, Trans
        End If
    ElseIf kickCtr < 30 Then
        If face = FACE_RIGHT Then
            Put (x,y+5), kickR2, Trans
        Else
            Put (x,y+5), kickL2, Trans
        End If
    ElseIf kickCtr < 45 Then
        If face = FACE_RIGHT Then
            Put (x,y+5), kickR3, Trans
        Else
            Put (x,y+5), kickL3, Trans
        End If
    ElseIf kickCtr < 90 Then
        If face = FACE_RIGHT Then
            Put (x,y+5), kickR4, Trans
        Else
            Put (x,y+5), kickL4, Trans

```

```

        End If
    Else
        kickCtr = 0
        state = STATE_STAND
    End If
    kickCtr = kickCtr + 1
ElseIf state = STATE_KICK_HI Then
    If kickCtr < 10 Then
        If face = FACE_RIGHT Then
            Put (x,y-10), kickhiR1, Trans
        Else
            Put (x,y-10), kickhiL1, Trans
        End If
    ElseIf kickCtr < 20 Then
        If face = FACE_RIGHT Then
            Put (x,y-10), kickhiR2, Trans
        Else
            Put (x,y-10), kickhiL2, Trans
        End If
    ElseIf kickCtr < 30 Then
        If face = FACE_RIGHT Then
            Put (x,y-10), kickhiR3, Trans
        Else
            Put (x,y-10), kickhiL3, Trans
        End If
    ElseIf kickCtr < 80 Then
        If face = FACE_RIGHT Then
            Put (x,y-10), kickhiR4, Trans
        Else
            Put (x,y-10), kickhiL4, Trans
        End If
    Else
        kickCtr = 0
        state = STATE_STAND
    End If
    kickCtr = kickCtr + 1
ElseIf state = STATE_PUNCH Then
    If punchCtr < 10 Then
        If face = FACE_RIGHT Then
            Put (x,y+5), punchR1, Trans
        Else
            Put (x,y+5), punchL1, Trans
        End If
    ElseIf punchCtr < 20 Then
        If face = FACE_RIGHT Then
            Put (x,y+5), punchR2, Trans
        Else
            Put (x,y+5), punchL2, Trans
        End If
    ElseIf punchCtr < 30 Then
        If face = FACE_RIGHT Then
            Put (x,y+5), punchR3, Trans
        Else
            Put (x,y+5), punchL3, Trans
        End If
    ElseIf punchCtr < 70 Then
        If face = FACE_RIGHT Then

```

```

        Put (x,y+5), punchR4, Trans
    Else
        Put (x,y+5), punchL4, Trans
    End If
Else
    punchCtr = 0
    state = STATE_STAND
End If
punchCtr = punchCtr + 1
ElseIf state = STATE_JUMP Then
    If jumpCtr <= 30 Then
        y = y - 3
        If face = FACE_RIGHT Then
            Put (x,y), jumpR1, Trans
        Else
            Put (x,y), jumpL1, Trans
        End If
    ElseIf jumpCtr < 60 Then
        y = y + 3
        If face = FACE_RIGHT Then
            Put (x,y), jumpR2, Trans
        Else
            Put (x,y), jumpL2, Trans
        End If
    Else
        jumpCtr = 0
        y = 150
        state = STATE_STAND
    End If
    jumpCtr = jumpCtr + 1
End If

If MultiKey(SC_UP) Then
    If state <> STATE_JUMP Then
        state = STATE_JUMP
    End If
End If

If MultiKey(SC_RIGHT) Then
    If state <> STATE_JUMP Then
        state = STATE_RUN
    End If
    x = x + 2
    face = FACE_RIGHT
    kickCtr = 0
    punchCtr = 0
ElseIf MultiKey(SC_LEFT) Then
    If state <> STATE_JUMP Then
        state = STATE_RUN
    End If
    x = x - 2
    face = FACE_LEFT
    kickCtr = 0
    punchCtr = 0
ElseIf MultiKey(SC_S) Then
    state = STATE_KICK
ElseIf MultiKey(SC_A) Then

```

```

        state = STATE_KICK_HI
    ElseIf MultiKey(SC_D) Then
        state = STATE_PUNCH
    Else
        If (state = STATE_KICK Or state = STATE_KICK_HI) And kickCtr <> 0 Then
            'DO NOTHING
        ElseIf state = STATE_PUNCH And punchCtr <> 0 Then
            'DO NOTHING
        ElseIf state = STATE_JUMP And jumpCtr <> 0 Then
            'DO NOTHING
        Else
            state = STATE_STAND
        End If
    End If

    End If

    Sleep 10
Loop Until MultiKey(SC_ESCAPE)

ImageDestroy standR1
ImageDestroy standR2
    ImageDestroy standL1
    ImageDestroy standL2

ImageDestroy runR1
ImageDestroy runR2
ImageDestroy runR3
ImageDestroy runR4
ImageDestroy runR5
ImageDestroy runR6
    ImageDestroy runL1
    ImageDestroy runL2
    ImageDestroy runL3
    ImageDestroy runL4
    ImageDestroy runL5
    ImageDestroy runL6

ImageDestroy kickR1
ImageDestroy kickR2
ImageDestroy kickR3
ImageDestroy kickR4
    ImageDestroy kickL1
    ImageDestroy kickL2
    ImageDestroy kickL3
    ImageDestroy kickL4

ImageDestroy kickhiR1
ImageDestroy kickhiR2
ImageDestroy kickhiR3
ImageDestroy kickhiR4
    ImageDestroy kickhiL1
    ImageDestroy kickhiL2
    ImageDestroy kickhiL3
    ImageDestroy kickhiL4

ImageDestroy punchR1
ImageDestroy punchR2
ImageDestroy punchR3

```

```
ImageDestroy punchR4
    ImageDestroy punchL1
    ImageDestroy punchL2
    ImageDestroy punchL3
    ImageDestroy punchL4
```

```
ImageDestroy jumpR1
ImageDestroy jumpR2
    ImageDestroy jumpL1
    ImageDestroy jumpL2
```

V. Cleaning up the code

Having a working code is one thing. Having an optimized and extendable code is another. With this remaining task at hand, a game programmers' best friend are arrays and loops – not to mention, writing reusable functions/procedures.

For languages that supports OOP, using them will also come in handy.

```
#include "fbgfx.bi"
Using FB

Dim Shared tmp As Any Ptr

Function Image_Load(p_width As Integer, p_height As Integer, p_file As String) As Any Ptr
    Const NULL As Any Ptr = 0
    tmp = ImageCreate(p_width, p_height, RGB(255,0,255))
    If tmp = NULL Then
        Print "Image creation failed!"
        Sleep
    End If
    BLoad p_file, tmp
    Return tmp
End Function

Dim standR(2) As Any Ptr
    Dim standL(2) As Any Ptr
        Dim standCtr As Integer
        Dim standIdx As Integer

Dim runR(6) As Any Ptr
    Dim runL(6) As Any Ptr
        Dim runCtr As Integer
        Dim runIdx As Integer

Dim kickR(4) As Any Ptr
    Dim kickL(4) As Any Ptr
        Dim kickCtr As Integer
        Dim kickIdx As Integer
```

```

Dim kickhiR(4) As Any Ptr
    Dim kickhiL(4) As Any Ptr

Dim punchR(4) As Any Ptr
    Dim punchL(4) As Any Ptr
        Dim punchCtr As Integer
        Dim punchIdx As Integer

Dim jumpR(2) As Any Ptr
    Dim jumpL(2) As Any Ptr
        Dim jumpCtr As Integer
        Dim jumpIdx As Integer

    Dim face As Byte
    Dim state As Byte
    Dim jump As Byte

Const FACE_RIGHT As Integer = 0
Const FACE_LEFT As Integer = 1

Const STATE_STAND As Integer = 0
Const STATE_RUN As Integer = 1
Const STATE_KICK As Integer = 2
Const STATE_KICK_HI As Integer = 3
Const STATE_PUNCH As Integer = 4
Const STATE_JUMP As Integer = 5

Dim x as Integer
Dim y as Integer

ScreenRes 320, 240, 32

x = 100
y = 150

Dim i As Integer

For i = 0 To 2 - 1
    standR(i) = Image_Load(47,88,"char1/stand/r/" & (i+1) & ".bmp")
    standL(i) = Image_Load(47,88,"char1/stand/l/" & (i+1) & ".bmp")

    jumpR(i) = Image_Load(53,83,"char1/jump/r/" & (i+1) & ".bmp")
    jumpL(i) = Image_Load(53,83,"char1/jump/l/" & (i+1) & ".bmp")
Next

For i = 0 To 6 - 1
    runR(i) = Image_Load(72,77,"char1/run/r/" & (i+1) & ".bmp")
    runL(i) = Image_Load(72,77,"char1/run/l/" & (i+1) & ".bmp")
Next

For i = 0 To 4 - 1
    kickR(i) = Image_Load(84,82,"char1/kick/r/" & (i+1) & ".bmp")
    kickL(i) = Image_Load(84,82,"char1/kick/l/" & (i+1) & ".bmp")

    kickhiR(i) = Image_Load(58, 97,"char1/kickhi/r/" & (i+1) & ".bmp")
    kickhiL(i) = Image_Load(58, 97,"char1/kickhi/l/" & (i+1) & ".bmp")

```

```

punchR(i) = Image_Load(70, 85,"char1/punch/r/" & (i+1) & ".bmp")
punchL(i) = Image_Load(70, 85,"char1/punch/l/" & (i+1) & ".bmp")
Next
Do
  Cls
  If state = STATE_STAND Then
    standCtr = standCtr + 1
    If standCtr >= 60 Then standCtr = 0

    If standCtr < 30 Then
      runIdx = 0
    ElseIf standCtr < 60 Then
      runIdx = 1
    End If
    If face = FACE_RIGHT Then
      Put (x,y), standR(runIdx), Trans
    Else
      Put (x,y), standL(runIdx), Trans
    End If
  ElseIf state = STATE_RUN Then
    runCtr = runCtr + 1
    If runCtr >= 60 Then runCtr = 0

    If runCtr < 10 Then
      runIdx = 0
    ElseIf runCtr < 20 Then
      runIdx = 1
    ElseIf runCtr < 30 Then
      runIdx = 2
    ElseIf runCtr < 40 Then
      runIdx = 3
    ElseIf runCtr < 50 Then
      runIdx = 4
    ElseIf runCtr < 60 Then
      runIdx = 5
    End If
    If face = FACE_RIGHT Then
      Put (x,y), runR(runIdx), Trans
    Else
      Put (x,y), runL(runIdx), Trans
    End If
  ElseIf state = STATE_KICK Then
    If kickCtr < 15 Then
      kickIdx = 0
    ElseIf kickCtr < 30 Then
      kickIdx = 1
    ElseIf kickCtr < 45 Then
      kickIdx = 2
    ElseIf kickCtr < 90 Then
      kickIdx = 3
    Else
      kickCtr = 0
      state = STATE_STAND
    End If
    If face = FACE_RIGHT Then

```

```

        Put (x,y+5), kickR(kickIdx), Trans
    Else
        Put (x,y+5), kickL(kickIdx), Trans
    End If
    kickCtr = kickCtr + 1
    ElseIf state = STATE_KICK_HI Then
        If kickCtr < 10 Then
            kickIdx = 0
        ElseIf kickCtr < 20 Then
            kickIdx = 1
        ElseIf kickCtr < 30 Then
            kickIdx = 2
        ElseIf kickCtr < 80 Then
            kickIdx = 3
        Else
            kickCtr = 0
            state = STATE_STAND
        End If
        If face = FACE_RIGHT Then
            Put (x,y-10), kickhiR(kickIdx), Trans
        Else
            Put (x,y-10), kickhiL(kickIdx), Trans
        End If
        kickCtr = kickCtr + 1
    ElseIf state = STATE_PUNCH Then
        If punchCtr < 10 Then
            punchIdx = 0
        ElseIf punchCtr < 20 Then
            punchIdx = 1
        ElseIf punchCtr < 30 Then
            punchIdx = 2
        ElseIf punchCtr < 70 Then
            punchIdx = 3
        Else
            punchCtr = 0
            state = STATE_STAND
        End If
        If face = FACE_RIGHT Then
            Put (x,y+5), punchR(punchIdx), Trans
        Else
            Put (x,y+5), punchL(punchIdx), Trans
        End If
        punchCtr = punchCtr + 1
    ElseIf state = STATE_JUMP Then
        If jumpCtr <= 30 Then
            y = y - 3
            jumpIdx = 0
        ElseIf jumpCtr < 60 Then
            y = y + 3
            jumpIdx = 1
        Else
            jumpCtr = 0
            y = 150
            state = STATE_STAND
        End If
        If face = FACE_RIGHT Then
            Put (x,y), jumpR(jumpIdx), Trans

```

```

Else
    Put (x,y), jumpL(jumpIdx), Trans
End If
jumpCtr = jumpCtr + 1
End If

If MultiKey(SC_UP) Then
    If state <> STATE_JUMP Then
        state = STATE_JUMP
    End If
End If

If MultiKey(SC_RIGHT) Then
    If state <> STATE_JUMP Then
        state = STATE_RUN
    End If
    x = x + 2
    face = FACE_RIGHT
    kickCtr = 0
    punchCtr = 0
ElseIf MultiKey(SC_LEFT) Then
    If state <> STATE_JUMP Then
        state = STATE_RUN
    End If
    x = x - 2
    face = FACE_LEFT
    kickCtr = 0
    punchCtr = 0
ElseIf MultiKey(SC_S) Then
    state = STATE_KICK
ElseIf MultiKey(SC_A) Then
    state = STATE_KICK_HI
ElseIf MultiKey(SC_D) Then
    state = STATE_PUNCH
Else
    If (state = STATE_KICK Or state = STATE_KICK_HI) And kickCtr <> 0 Then
        'DO NOTHING
    ElseIf state = STATE_PUNCH And punchCtr <> 0 Then
        'DO NOTHING
    ElseIf state = STATE_JUMP And jumpCtr <> 0 Then
        'DO NOTHING
    Else
        state = STATE_STAND
    End If
End If

Sleep 10
Loop Until MultiKey(SC_ESCAPE)

For i = 0 To 2 - 1
    ImageDestroy standR(i)
    ImageDestroy standL(i)
    ImageDestroy jumpR(i)
    ImageDestroy jumpL(i)
Next

```

```
For i = 0 To 6 - 1
    ImageDestroy runR(i)
    ImageDestroy runL(i)
Next
For i = 0 To 4 - 1
    ImageDestroy kickR(i)
    ImageDestroy kickL(i)
    ImageDestroy kickhiR(i)
    ImageDestroy kickhiL(i)
    ImageDestroy punchR(i)
    ImageDestroy punchL(i)
Next
Sleep
```

Where to go from here?

There are lots more to learn in game programming. Depending on which specific platform you are going, you can learn 3D programming, Physics programming, Network programming, Database programming, Script Engine programming, Augmented Reality programming, and many more. All of which could lead into developing a Game Engine.

But you don't need to learn all as you need a *team* to make games.

Resources

You can get a PDF copy of this article and the images used plus the source code and compiled binary from the site indicated below.

<http://konsolscript.org/~mjmandoza/freebasic/basic-game-programming-with-freebasic/>

Related Article

Interested in 3D Programming and/or Flash ActionScript? Check out:

<http://konsolscript.org/~mjmandoza/papervision/>