DS Game Maker

DS Game Maker
Coding at a Glance

Introduction

DS Game Maker (DSGM) supports the C Programing language as well as its own language known as DBAS (Dynamic BASic). The two languages can be used at the same time in any script or action however it is recommended to generally stick to one language to avoid syntax errors in DSGM. As such, this guide overviews how to write code in DBAS so that it can be used if desired. Lastly, the information from this guide was sourced from the original DSGM Website.

- DigitalDesignDude

Commenting

For code to be maintainable in the future and possibly by people other than yourself, it needs to be annotated. You can easily comment in DBAS using the ' symbol. A commented line is ignored when the code runs. It has only the purpose of assisting you in your development. Note in DBAS you may only comment out one line. Use C code if you wish to comment out multiple lines at once.

Examples

```
'This line does nothing
'The line below sets the Ball X to 500
BallX = 500
```

Functions

Calling a DBAS function follows the following syntax:

[Function Name]([Argument 1], [Argument 2], [Argument 3] ...)

Example

```
Set_Camera_XY(Top_Screen, 32, 32)
```

Declarations

To declare a variable in DBAS, use the following form: [Scope] [Name] As [Type] = [Value]

[Scope] is either:

- Dim, declaring the variable only in the context of the current Script
- Public, declaring the variable so that all scripts may access it

[Name] is the name of the variable:

Alpha-numeric characters only plus underscores; cannot start with a number

[Type] is one of the following:

- Integer - 16-bit number (default type)
- String - Variable for holding text or charaters, such as "James"
- Boolean - Holds a true or false value
- Signed Byte - An Integer with the range -127 to +127
- Unsigned Byte - An Integer with the range 0 to +128
- Float - A number which can store a decimal value (these take up lots of memory)

[Value] is the default value that the variable should have.

You may provide a constant or a function.

Examples

```
Dim MyName As String = "James"
Dim MyAge As Integer = 17
Public SomethingGlobal as Boolean = false
```

DBAS is truly dynamic. You can skip parameters of the declaration line, such that the following are possible:

```
'No default value:
Dim MyName As String
'No Type so 'Integer' presumed:
Dim MyAge = 15
```

Conditions

To check whether a condition is true, use the If statement in DBAS:

```
If [Expression] = [Value]
  'Executing statements
End If
```

[Expression] can be a variable like X or Y, or it may also be another DBAS function. There is never a need for 2 equal signs.

Examples

```
'Example 1
If X = 5
 'X is 5
End If

'Example 2
If Get_X(0) = 32
 'Instance ID 0 has X position of 32
End If

'Example 3
If Test_Chance(10)
 '1 in 10...
End If
```

Notice that in the last example no equals sign or value was provided. You do not have to provide them, and if you do not you are providing the shorthand for 'equals true'. The above should read 'If Test_Chance(10) = true'. An If then ends with 'End If' which signifies the end of the DBAS functions or statements that occur when the 'If' condition is met.

To detect if a condition is false, there are 2 ways to do it:

```
If Is_Divisible(X, 5) = false
 'X is not divisible by 5 - this works
End If

If Not Is_Divisible(X, 5)
 'X is not divisible by 5 - this works
End If
```

They are both exactly the same, so use which one makes most sense to you.

Looping

To Loop in DBAS, use the 'For' statement, accompanied by a 'Next' (just like for If ... End If). You must declare the variable to loop beforehand or the script will not work. A Loop has the following syntax:

```
For [Variable] = [Start] To [End]
 'Use i as the iteration
Next
```

Example

```
'Draw a vertical line of A characters
Dim i as Integer
For i = 0 to 10
 Draw_Text(Top_Screen, 1, i, "A")
Next
```

Looping is really as easy as that, and there is not much more to say.

DBAS Tips

- Do not use double equals signs in Ifs - that only works in C Scripts and C programming
- Make sure you declare a variable before you use it in a For loop or DBAS function
- String arguments in DBAS functions should be enclosed in double-quote marks eg. Set_String(FirstName, "James")