

Introduction

This is a list of all of the functions exclusively built for DS Game Maker. You can use these anywhere in actions, C Scripts and DBAS Scripts. To get even more functions, refer to the PALib functions documentation which overviews functions from the PALib Code Library that DS Game Maker also uses. Lastly, credit for this documentation goes toward the author/s of the original [DSGM Website](#).

- *DigitalDesignDude*

Objects

Get_Single_ID(ObjectName)

If you have just one instance of an object, this function returns its ID which you can use in other functions. If there is more than one instance, the first ID will be returned.

Example Usage

```
Dim ID As Integer = Get_Single_ID(ball)
Delete_Instance(ID)
```

NextInstance()

Returns the next available instance ID that you can use to create an object. When you create an object, you have to provide an instance ID for it (where you want it to go in the instances array). If you create and delete many objects, you can lose track of what positions are free. In these cases use this function. If you don't care about what instance number an object gets, use this function for the instance ID every time you create an object.

Example Usage

```
Create_Object(SofaObject, NextInstance(), Top_Screen, 32, 32)
Create_Object(DoorObject, NextInstance(), Top_Screen, 64, 64)
```

Sprite_Collision(InstanceID1, InstanceID2)

Detects the collision of 2 objects, specified by their instance numbers. This is the function used for the 'If Collision' event. You cannot provide the names of 2 objects like this:

Is_Sprite_Collision(ball, paddle), you must use their instance IDs. See the function Get_Single_ID.

Example Usage

```
Dim BulletID As Unsigned Byte = Get_Single_ID(Bullet_Object)
Dim PlayerID As Unsigned Byte = Get_Single_ID(Player_Object)

If Is_Sprite_Collision(BulletID, PlayerID)
Delete_Instance(BulletID)
End If
```

Sprite_Collision_Mid(InstanceID1, InstanceID2)

Very similar to the function Is_Sprite_Collision but a collision is only detected if the mid-point of an instance intersects the boundary of another. This is more useful for non-square objects.

Object_Under_Point(Screen, X, Y, ObjectName)

Returns whether there is an instance of ObjectName under a specified point.

Object_Under_H_Line(Screen, X1, X2, Y, ObjectName)

Returns whether there is an instance of ObjectName under the line specified with the co-ordinates (horizontal line; so Y applies to both X points).

Object_Under_V_Line(Screen, Y1, Y2, X, ObjectName)

Returns whether there is an instance of ObjectName under the line specified with the co-ordinates (vertical line; so X applies to both Y points).

Move_Object_Stylus(InstanceID)

Allows the instance to be dragged about with the Stylus. This function must be called every frame; therefore it must be called in a Step event.

Animate(InstanceID, FirstFrame, LastFrame, FPS)

Animates an instance (based upon the frames in its Sprite). The first argument is the instance number to animate. The second is the start frame. The fourth argument is 'frames per second' and it means how many frames should cycle visually within 1 second.

Set_V_Flip(InstanceID, Flip)

Determines whether an instance appears flipped vertically or not. The last argument should be true or false.

Set_H_Flip(InstanceID, Flip)

Determines whether an instance appears flipped horizontally or not. The last argument should be true or false.

Create_Object(ObjectName, InstanceID, Screen, X, Y)

Creates an instance of an Object. The first argument is the object name. For Screen, use Top_Screen or Bottom_Screen. InstanceID can be NextInstance() if the next free ID is not known.

Set_Sprite(InstanceID, SpriteName, DeleteOldSprite)

Sets the Sprite of the provided instance. The Sprite Name should be enclosed in quotes. DeleteOldSprite is a boolean value that is used mostly when DSGM calls the function.

Delete_Instance(InstanceID)

Deletes the selected instance.

Get_X(InstanceID)

Returns the X Position of the instance.

Example Usage

```
Set_X(Get_Single_ID(Ball2), 64)
'Whatever object is in instance slot 0, set X position to 64:
Set_X(0, 64)
```

Get_Y(InstanceID)

Returns the Y position of the instance.

Set_X(InstanceID, X)

Sets the X Position of an instance.

Set_Y(InstanceID, X)

Sets the Y Position of an instance.

Set_XY(InstanceID, X, Y)

Sets the X and Y Positions of an instance.

Set_Frame(InstanceID)

Sets the animation Frame of the instance. The first frame is number 0.

Get_Frame(InstanceID, Frame)

Returns the animation Frame of the instance.

Count_Instances(ObjectName)

Returns the number of instances of the specified object.

Example Usage

```
'Breakout type game
Dim Count As Unsigned Byte = Count_Instances(BrickObject)
If Count = 0
    Goto_Next_Room()
End If
```

Graphics

Text functions require the X and Y Positions to be provided in Tiles.

Draw_Health(Screen, XTiles, YTiles)

Draws the 'health' variable on the selected screen at the provided X and Y tile co-ordinates.

Draw_Lives(Screen, XTiles, YTiles)

Draws the 'lives' variable on the selected screen at the provided X and Y tile co-ordinates.

Draw_Score(Screen, XTiles, YTiles)

Draws the 'score' variable on the selected screen at the provided X and Y tile co-ordinates.

Draw_String(Screen, XTiles, YTiles, StringVariable)

Draws a Sting variable at the specified X and Y tile co-ordinates on the specified screen. The variable must be declared in the Script or Global Variable/Arrays/Structures managers as a 'String' type.

Example Usage

```
Dim MyName As String
MyName = "James"
Draw_String(Top_Screen, 1, 1, MyName)
```

Draw_Variable(Screen, XTiles, YTiles, VariableName, IsFloat)

Draws the value of a Variable. The last argument, IsFloat, takes either true or false. Use 'true' for a floating (decimal number). For types such as Integer or Unsigned Byte, use false.

Set_BG_Scroll(Screen, Layer, X, Y)

Scrolls the loaded background to the specified X and Y co-ordinates. The layer argument allows you to scroll any of the 4 background layers (0-3). If you scroll past the constraints of the background (greater X or Y than the width or height), the background may repeat or show corrupt data. If the function does not do anything, check the Layer and that in the room 'Scroll BG with Camera' is not checked because this will overwrite the work of this function. The default Layer number is 2, for backgrounds placed on the screens within DS Game Maker.

Draw_Text(Screen, XTiles, YTiles, "Text")

Draws text at the specified X and Y tile co-ordinates on the specified screen. The text must be enclosed in quotes and you cannot use a variable. To do that see Draw_String.

Set_Camera_XY(Screen, X, Y)

Sets the position of the camera in the room. Moving the camera to the right, for example, shifts plotted objects to the left. Objects still retain their X and Y co-ordinates after camera movement; the offset is only visual. You can compute the locations of objects relative to the physical screen borders by using the variables RoomData.TopX, (..) TopY, BottomX, BottomY.

Example Usage

```
Dim XPos as Integer = 0
Dim YPos as Integer = 0
If Pad.Newpress.Left
    XPos -= 1
End If
If Pad.Newpress.Right
    XPos += 1
End If
Set_Camera_XY(Bottom_Screen, XPos, YPos)
```

Get_Camera_X(Screen) or Get_Camera_Y(Screen)

Returns the X or Y position of the Camera.

Example Usage

```
'Move the camera to the right
Set_Camera_XY(Top_Screen, Get_Camera_X(Top_Screen) + 5,
Get_Camera_Y(Top_Screen))
```

Logic

Test_Chance(Sides)

You specify the number of sides of a dice which is then thrown. Use this function to test the probability of the event happening with the supplied sides. If the dice is thrown and lands on 1, the function returns true, otherwise false. For example, when the ball hits a brick you can provide a particular 'chance' of a bonus score.

Example Usage

```
If Test_Chance(10)
  '1 in 10 times (random), these functions will run...'
End If
```

Is_Divisible(Number, Divisor)

Returns whether the first number is divisible by the second (with the answer being integral). Arguments of 10, 5 return true but 19, 6 return false.

Is_Position_Free(Screen, X, Y)

Returns if a position on a collision map background is free (is magenta), at the specified X and Y co-ordinates. Use in conjunction with the action 'Load Collision Map'.

Is_Position_Occupied(Screen, X, Y)

Returns if a position on a collision map background is occupied (is not magenta), at the specified X and Y co-ordinates. Use in conjunction with the action 'Load Collision Map'.

Random(Minimum, Maximum)

Returns a random number between the provided constraints. Numbers can be negatively signed. For better random numbers, use a physical Nintendo DS.

Set_String(Variable, Text)

Sets a String variable to the provided value. This is a rare function to have to use as DBAS automatically does this when you do an = assignment for a String.

Example Usage

```
Set_String(FirstName, "James")
Set_String(LastName, "Garner")
```

Strings_Same(String1, String2)

Returns true if the strings are equal, and false if they are not.

Example Usage

```
Dim MyName as string = "James"
Dim HisName as string = "James"
If Strings_Same(MyName, HisName)
    Draw_Text(Top_Screen, 1, 1, "Names are the same.")
Else
    Draw_Text(Top_Screen, 1, 1, "They are different.")
End If
```

Pause_Alarm(AlarmID, Pause)

Pauses the specified Alarm (0 - 31). Pause takes true or false; true to pause the alarm and false to unpause it.

Set_Alarm(AlarmID, Time)

Sets the time of the specified Alarm (0 - 31) and activates it.

Goto_Next_Room()

Goes to the next room from the current room, if another room exists.

Sound

`Stream_BG_Sound("SoundName")`

Streams the provided Background sound. Use in a 'Step' Event.

`Stop_BG_Sound()`

Stops the Background (MP3) sound from playing.

`Pause_BG_Sound()`

Pauses the current playing Background sound. Due to the nature of the streaming, use with care if you plan to create and delete objects that stream the sound.

`Unpause_BG_Sound()`

Unpauses the current playing Background sound; does the opposite of the above.

`Unpause_BG_Sound(Loop)`

Sets the looping of the Background sound. Provide a boolean value such that 'true' makes the Background sound play again at the end, and 'false' does not.

`Stop_Sound_Effects`

Stops all sound effects from playing.

`Set_Volume(u8 Volume)`

Sets the master volume for all sounds.

`Set_BG_Panning(Panning)`

Sets the horizontal panning of all sounds. The Panning parameter ranges from -64 to 64. -64 is far-left panning and 64 is far right; 0 being the mid-point as default.

PRO Functions

The following functions were originally only available in the Pro Edition but now are useable in DSGM 5.20.

Read_File(Content, Path)

Reads the content from the file at 'Path' into the previously declared 'Content' variable.

```
Dim FileContent As String
Read_File(FileContent, "story.txt")
Draw_Text(Top_Screen, 1, 1, FileContent)
'Draws the contents of story.txt on the top screen at position 1, 1
```

Write_File(Content, Path)

Writes the 'Content' to the file at the 'Path'.

```
Dim Content As String = "Hello, World"
Write_File(Content, "new_born.txt")
'Writes 'Hello World' to 'new_born.txt'
```

List_Files(Screen, X, Y, Path)

Lists files in the selected file system (see Switch_To_NitroFS() and Switch_To_FAT()). Default file system is NitroFS. Add files in 'Game Settings'.

Switch_To_NitroFS()

Switches the file system commands to work with NitroFS (files contained within the NDS - See 'Game Settings' window).

Switch_To_FAT()

Switches the file system commands to work with FAT (files stored alongside the NDS on the MicroSD card).

Enable_Rotation(InstanceID)

Enables the rotation of the specified instance. Most commonly, call it once in the 'Creation' event for the object. You can then set its angle with 'Set_Angle'.

Disable_Rotation(InstanceID)

Disables the rotation of the specified instance caused by Enable_Rotation or Set_Angle.

Set_Angle(InstanceID, Angle)

Sets the rotation angle for the specified instance. The angle is in Radians, and be sure to use Enable_Rotation first. To convert to and from Radians, see the below functions.

Angle_To_Radians(Angle)

Converts a "human" angle (0-360) to Radians; commonly for use with Set_Angle.

Radians_To_Angle(Radians)

Converts a Radian angle back to "human form" (0-360).

Angle_Between_Points(X1, Y1, X2, Y2)

Returns the angle between 2 points in Radians. You can use this with [X], [Y], Stylus.X, Stylus.Y to make objects face the stylus.

Distance_Between_Points(X1, Y1, X2, Y2)

Returns the distance between 2 points using Pythagoras. This is slow so use with care.