Maps and Tiles

Insert scenery for your games, with these commands that allow you to create and manage maps and tiles.

CreateMaps WIDTH, HEIGHT
Create a maps set width a specific size
Parameters:
WIDTH: Number of tiles in width
HEIGHT: Number of tiles in height

AddMap
Add a new map to the current maps set.

RemoveMap MAPINDEX
Remove a map from the current maps set.
Parameters:
MAPINDEX: Index of map to remove.

InsertMap MAPINDEX
Insert a new map int the current maps set at the position given
Parameters:
MAPINDEX: Position of the new map in the current maps set

LoadMaps MAPNAME$
Load a maps file. File must be one of these formats : aozmap or tmx (Tiled)
Parameters:
MAPNAME$: Filename to load.

GetMapIndex VIEWID$
Return the index of map associated to a map view.
Parameters:
VIEWID$: Name of the map view.
Value returned:
integer: Index of the map

SetMapIndex VIEWID$, MAPINDEX
Link a map to a map view
Parameters:
VIEWID$: Name of the map view.
MAPINDEX: Index of the map.

GetMapWidth VIEWID$
Return the number of tiles in width of a map view
Parameters:
VIEWID$: Name of the map view

GetMapHeight VIEWID$
Return the number of tiles in height of a map view

Parameters:
VIEWID$: Name of the map view

GetTileValue VIEWID$, TILEINDEX
Return the value of a tile of a Map View
Parameters:
VIEWID$: Name of the map view
TILEINDEX: Index of tile
Value returned:
string: The value of tile

GetTileValue VIEWID$, X, Y
Return the value of a tile of a Map View for a position XY
Parameters:
VIEWID$: Name of the map view
X: Horizontal position of the tile
Y: Vertical position of the tile
Value returned:
string: The value of tile

SetTileValue VIEWID$, TILEINDEX, VALUE$
Set the value of a tile of a Map View
Parameters:
VIEWID$: Name of the map view
TILEINDEX: Index of tile
VALUE$: Value of the tile

SetTileValue VIEWID$, X, Y, VALUE$
Set the value of a tile of a Map View
Parameters:
VIEWID$: Name of the map view
X: Horizontal position of the tile
Y: Vertical position of the tile
VALUE$: Value of the tile

ResetMaps
Reset the maps set and clean all the maps datas

MapCount
Return the number of maps in the current maps set
Value returned:
integer: The number of maps

MapView Copy VIEWID$
Store the actual state of a map view
Parameters:
VIEWID$: ID of the map view

MapView Restore VIEWID$
Restore the state of a map view stored by "MapView Copy" commands
Parameters:
VIEWID$: ID of the map view

MapView Open VIEWID$, X, Y, WIDTH, HEIGHT
Open a view to display a map on the current screen
Parameters:
VIEWID$: ID of the map view
X: Position X in pixel of the view on the current Screen of the map view
Y: Position Y in pixel of the view on the current Screen of the map view
WIDTH: Width of the view in pixel of the map view
HEIGHT: Height of the view in pixel of the map view

MapView Close VIEWID$
Close a map view opened with the "MapView Open" commands.
Parameters:
VIEWID$: ID of the map view

MapView Offset VIEWID$, X, Y, Z
Scrolling of the map into the map view
Parameters:
VIEWID$: ID of the map view
X: Position X in pixel of the scrolling of the map view
Y: Position Y in pixel of the scrolling of the map view
Z: Position Z in pixel of the scrolling of the map view (not used)

MapView Display VIEWID$, X, Y
Set the position of the map view on the current screen
Parameters:
VIEWID$: ID of the map view
X: Position X in pixel of the view on the current Screen of the map view
Y: Position Y in pixel of the view on the current Screen of the map view

MapView Display VIEWID$, X, Y, WIDTH, HEIGHT
Set the position and the size of the map view on the current screen
Parameters:
VIEWID$: ID of the map view
X: Position X in pixel of the view on the current Screen of the map view
Y: Position Y in pixel of the view on the current Screen of the map view
WIDTH: Width of the view in pixel of the map view
HEIGHT: Height of the view in pixel of the map view

MapView Offset X VIEWID$
Parameters:
VIEWID$: ID of the map view
Value returned:
integer: Position in pixel.

MapView Offset Y VIEWID$
Parameters:

VIEWID$: ID of the map view
Value returned:
integer: Position in pixel.

MapView Offset Z VIEWID$
Parameters:
VIEWID$: ID of the map view
Value returned:
integer: Position in pixel.

MapView Redraw VIEWID$
Redraw the map view on the screen
Parameters:
VIEWID$: ID of the map view

MapView Redraw VIEWID$, BACKCOLOR
Redraw the map view on the screen with a background color
Parameters:
VIEWID$: ID of the map view
BACKCOLOR: Color RGB of the background

MapTile Ref VALUE$, NUMIMAGE
Link a image of the images bank to a tile value
Parameters:
VALUE$: Tile value to associated
NUMIMAGE: Number of image of the images bank.

MapTile Ref VIEWID$, VALUE$, NUMIMAGE
Link a image of the images bank to a tile value for a specific map view
Parameters:
VIEWID$: ID of the map view
VALUE$: Tile value to associated
NUMIMAGE: Number of image of the images bank.

MapTile Size WIDTH, HEIGHT
Set the width and height of the map tiles in pixel
Parameters:
WIDTH: Width of the map tiles in pixel
HEIGHT: Height of the map tiles in pixel

MapTile Size VIEWID$, WIDTH, HEIGHT
Set the width and height of the map tiles in pixel for a specific map view
Parameters:
VIEWID$: ID of the map view
WIDTH: Width of the map tiles in pixel
HEIGHT: Height of the map tiles in pixel

MapTile Width VIEWID$
Return the width of the tiles map for a specific map view
Parameters:
VIEWID$: ID of the map view

Value returned:
integer: Width of the map tiles

MapTile Height VIEWID$
Return the wheight of the tiles map for a specific map view
Parameters:
VIEWID$: ID of the map view
Value returned:
integer: Height of the map tiles

MapTile Count VIEWID$, TILEVALUE$
Return the number of a specifis tile value in map view
Parameters:
VIEWID$: ID of the map view
TILEVALUE$: Value of the searched tile
Value returned:
integer: Height of the map tiles

MapTile Find VIEWID$, TILEVALUE$
Return the first number of tile on the map view where is tile value asked
Parameters:
VIEWID$: ID of the map view
TILEVALUE$: Value of the searched tile
Value returned:
integer: The first index of tile found. If -1 is returned then no tile found.

MapTile Next
Return the next number of tile on the map view after to had called the "MapTile Find" command.
Value returned:
integer: The next index of tile found. if -1 is returned then no tile found.

MapTile X VIEWID$, TILEINDEX
Return the horizontal position in pixel of a tile in the map view.
Parameters:
VIEWID$: ID of the map view
TILEINDEX: Index of tile
Value returned:
integer: The horizontal position in pixel.

MapTile Y VIEWID$, TILEINDEX
Return the vertical position in pixel of a tile in the map view.
Parameters:
VIEWID$: ID of the map view
TILEINDEX: Index of tile
Value returned:
integer: The vertical position in pixel.

MapTile Test VIEWID$, X, X
Return the index of tile at the position x and y on a map view
Parameters:
VIEWID$: ID of the map view

X: Position X in pixel to test.

X: Position Y in pixel to test.

Value returned:

string: Index of tile found.