Version

0.1

USER MANUAL (BETA)

By Tim Partlett

CryEngine® Sandbox™ Far Cry Edition

CRYENGINE® SANDBOX™ – FAR CRY EDITION

User Manual

CryEngine® Sandbox™ - Far Cry Edition
Crytek GmbH
Coburg • Germany

Table of Contents

CHAPTER 1 – MAP

CHAPTER 2 - OBJECTS

CHAPTER 3 - ARTIFICIAL INTELLIGENCE

CHAPTER 4 - EVENTS

CHAPTER 5 - INTERNAL AREAS

CHAPTER 6 - MULTIPLAYER MAPS

CHAPTER 7 - SINGLE PLAYER MISSIONS

Introduction

The User Manual released with this CryEngine Sandbox editor is a beta document, and will therefore not be complete and may possible contain errors or omissions. The full version will be released either shortly before the game reaches the store shelves, or shortly afterwards, depending on circumstances, so be sure to check the news at www.farcry-thegame.com for information and updates. This version is perfectly good for creating most levels, and is only missing the significant sections of sound placement and cut-scene editing. The final release of the document should contain these sections, along with numerous other additions, including short-cut key set up and walkthroughs. The finished document will also be fully reviewed for error and omissions, in both content and in comprehension. Updates will also be made to various sections, especially the appendices, like the events and object properties list. Additional appendices, covering aspects outside the remit of this document, will be added as they are acquired.

The remit of this User Manual is to explain the workings of the CryEngine Sandbox editor, in so far as it affects the client end of the service, i.e. the person who buys the game. The documentation is designed to enable the user to create multiplayer maps, levels and combinations of levels that are as complex and detailed as any of the levels or multiplayer maps released with the game. That means that the user should, with the skills learned from using the manual, be able to produce multiplayer maps and levels of the quality of anything provided with the game, but only if that user has the requisite level design skills. What the manual does not provide, is details on how to modify the game engine itself, so that it performs differently to that which is provided. Therefore there is no information provided in the main documentation that relates to such aspects of modifying games as scripting, importing models, etc. These aspects may be commented on in the documentation, but only in so far as it relates to the creation of levels within the remit already specified. This does not preclude the inclusion of appendices to explain such aspects of the game's design.

Also included in this documentation is the Demo level, created by Alex Werner, which exemplifies many of the aspects of the editor outlined in this user manual. This example level should be used to see how many common elements from the game are actually implemented, and can be found in the Demo folder of the Levels directory on the CD.

Acknowledgements

While this User Manual is the written and editorial work of the named author, it would not have been possible without the help of many of the skilled people working on the Far Cry project. This acknowledgements page recognises the great contribution of these people in the making of this document. The key contributors will be listed here, but there will be many more whose incidental assistance was invaluable.

In alphabetical order:

Ben Bauer; Internal Levels section, Elevator operation, and Events and Lighting appendix.

Sebastien Couture; Multiplayer Maps section, Artificial Intelligence section, and Objects appendix.

Sten Hübler; Objects and Events appendix.

Robert Peterson; original documentation.

Alex Werner; Demo level and Single Player Missions section.

Also thanks to Marco Corbetta, Timur Davidenko, Chris Natsuume, Richard Tsao, and many others for checking the documentation, offering assistance, corrections and updates.

Chapter

1

Map Creation

This section describes the process of creating the terrain, from the first grey scale height map to the painting of vegetation brushes and applying light.

Creating a basic map for any level consists of several processes, terrain, textures, lighting and vegetation. While this section can teach you the tools you need to create a map, the real hard work for this will come from how you use the tools to implement the map which is in your own imagination.

Terrain

There are three ways that you can generate terrain. You can generate the terrain automatically, with the Generate Terrain option in the Terrain window, you can import a grey scale height map, or you can alter the terrain directly using brush tools contained within the editor. Using the Generate Terrain option is the easiest, and quickest, means of creating believable looking terrain, but it doesn't allow for much in the way of control. Height maps allow for much more control, but takes longer than automatic generation and doesn't provide a great deal of fine detail. Using the brush tools in the editor takes a lot of skill and time to create the perfect map with, but it allows for the greatest amount of control, and is the preferred means of terrain generation with experienced level designers.

Automatic Terrain Generation

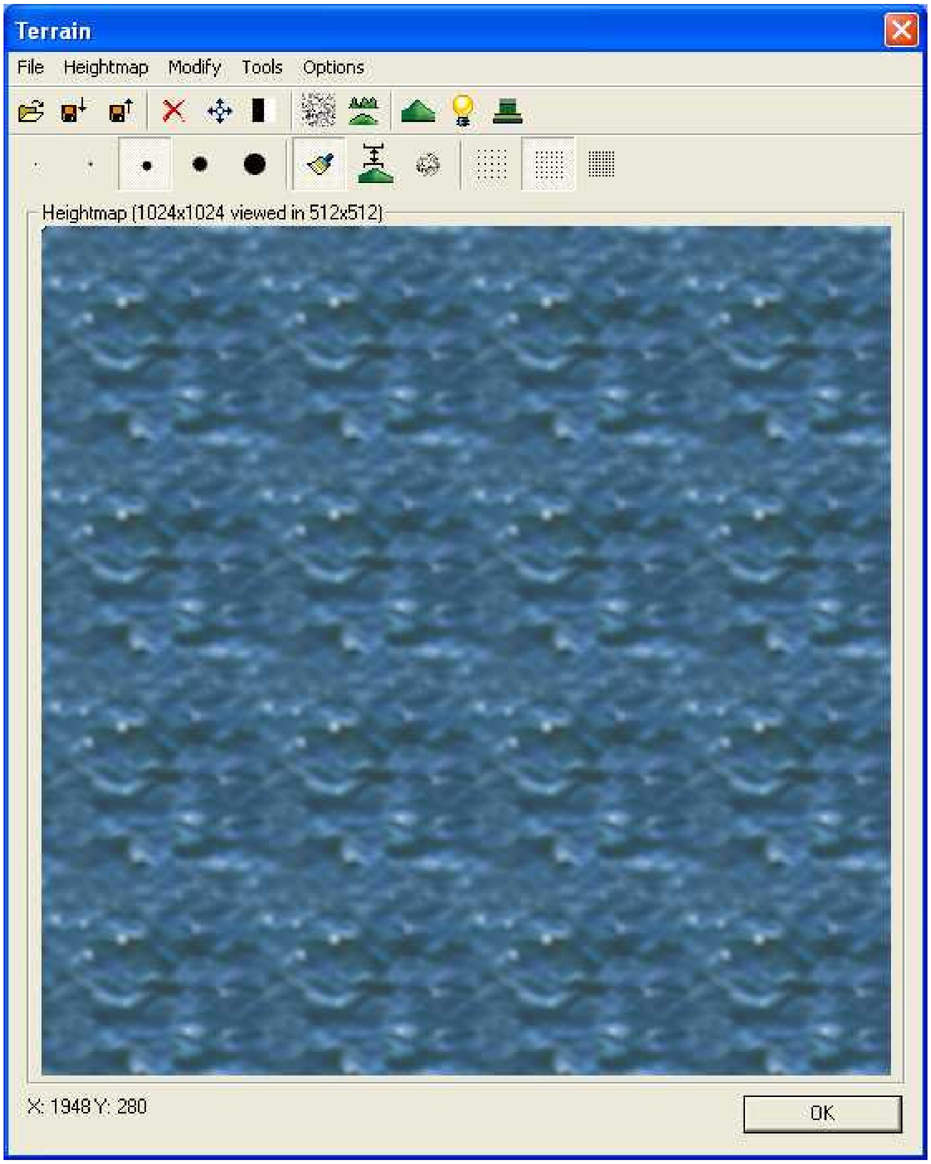
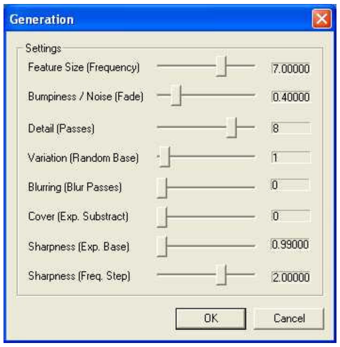

Terrain
File Heightmap Modify Tools Options
Heightmap (1024x1024 viewed in 512x512)
X: 1948 Y: 280
OK

Figure 1.1 Use the Terrain Window to create height maps for your terrain.

If you click the Terrain icon, you will be presented with a window with a map containing nothing but sea. From the menu at the top, you can select the Generate Terrain option to quickly create a landscape to work on. The option will present you with a number of parameters, which you can use to influence the way in which the function generates the terrain.

- Feature Size Frequency - This determines the amount of "noise grains" to be applied to the map by the noise function.
- Bumpiness/Noise - Affects the bumpiness of the terrain.
- Detail - This determines the number of times the noise function will be applied.

- Variation - Random seed.
- Blurring - Sets the number of times the smoothing filter is applied to the noise function.
- Cover - Not Used
- Sharpness - Not Used
- Sharpness - Not Used



Generation

Settings

Feature Size (Frequency) 7.00000

Bumpiness / Noise (Fade) 0.40000

Detail (Passes) 8

Variation (Random Base) 1

Blurring (Blur Passes) 0

Cover (Exp. Substract) 0

Sharpness (Exp. Base) 0.99000

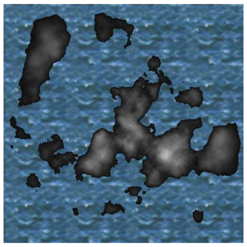
Sharpness (Freq. Step) 2.00000

The key parameters are Feature Size Frequency, Bumpiness and Blurring. The Feature Size Frequency affects the amount of land that is created, while Bumping and Blurring affects how bumpy or smooth the land that is generated will be. The Variation parameter is also important, and provides the random seed for the function. Maps generated with the same random seed will tend to follow similar patterns. Changing this value can dramatically alter the way the map looks.

Note

The default settings will create an archipelago of islands with jagged mountain tops. To create a more solid continent based map with smoother edges, increase the Feature Size, Variation and Blurring parameters.

Height Maps



The terrain is based on a gray scale, with pure white being the highest point on the map, and pure black being the lowest. Shades of grey in-between give differing heights. Thus a spectrum of grey shades from pure white to pure black will give a shallow slope, and pure black next to pure white will give a long sheer cliff face. This means that you can paint the map that you want, using a grey scale template. You can create this template with the height map editor provided with the CryEngine Sandbox editor, or import it from one of your favourite paint packages, like Paint Shop Pro.

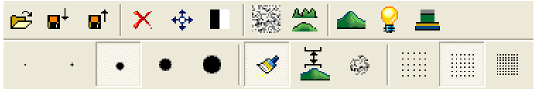


Figure 1.2 Terrain Window Tool Bar

The height map editor allows you to paint the terrain with a brush, the size of which you can choose from the different sized circles on the tool bar. Next to this on the tool bar you can select your brush, either a using a plain brush, the height brush, or the noise brush. The plain brush paints a smooth surface terrain, the height brush paints a grey scale of the exact height you choose, and the noise brush makes the terrain more random and spiky. You can also alter the pressure by which the brush is applied, by altering the opaqueness of the brush. The opaqueness is set by choosing from the three grid shapes on the left of the tool bar. The more compact the grid pattern, the less opaque the brush, and the harder the brush is applied.

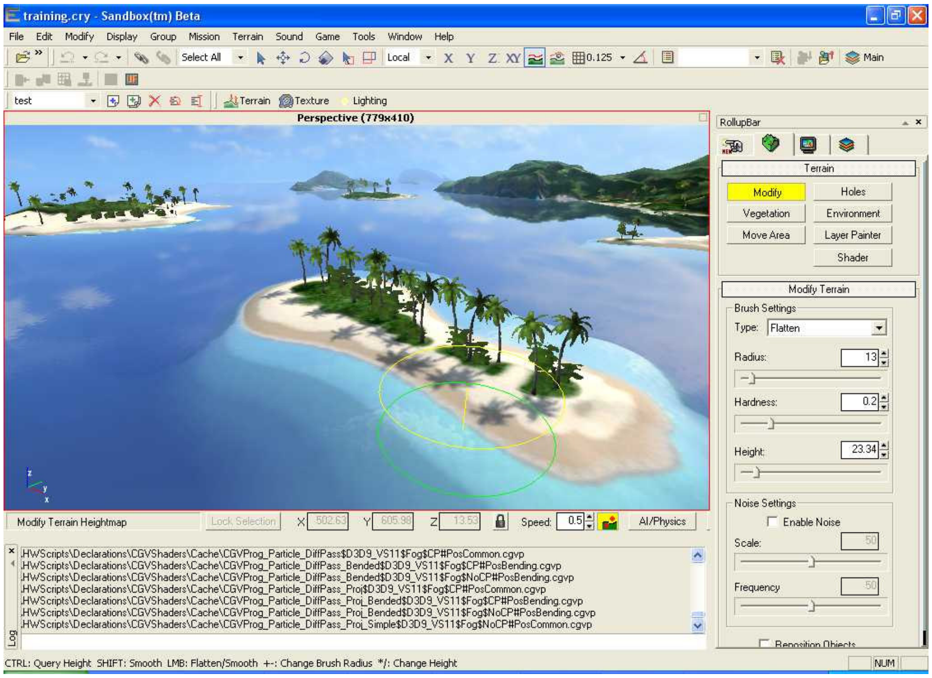
Note

Imported image files must be in windows bitmap (.bmp) format.

Direct Terrain Editing

Tip: use the wire-frame view to give you a clearer view of the terrain heights when editing in perspective view.

The most powerful method of terrain editing is to create the terrain yourself, in the map and perspective view, using the editor tools for raising, lowering and smoothing the terrain. This method gives you the greatest control over the landscapes created, as well as being able to see much more clearly what you are producing, but it also takes the greatest amount of time. While this is the preferred method of terrain editing by those who are more experienced with the CryEngine Sandbox editor and level editing in general, it may be better for less experienced editors to use either automatic terrain generation or a height map, and to only use the terrain editing tools to fine tune the end product.



Brush Settings
Type: Flatten
Radius: 13
Hardness: 0.2
Height: 23.34

Noise Settings
☐ Enable Noise
Scale: 50
Frequency: 50

☐ Reposition Objects

X 502.63 Y 605.98 Z 13.53 Speed: 0.5

```
.HWScripts\Declarations\CGVShaders\Cache\CGVProg_Particle_DiffPass$D3D9_VS11$Fog$CP#PosCommon.cgvp
.HWScripts\Declarations\CGVShaders\Cache\CGVProg_Particle_DiffPass_Bended$D3D9_VS11$Fog$CP#PosBending.cgvp
.HWScripts\Declarations\CGVShaders\Cache\CGVProg_Particle_DiffPass_Bended$D3D9_VS11$Fog$NoCP#PosBending.cgvp
.HWScripts\Declarations\CGVShaders\Cache\CGVProg_Particle_DiffPass_Proj$D3D9_VS11$Fog$CP#PosCommon.cgvp
.HWScripts\Declarations\CGVShaders\Cache\CGVProg_Particle_DiffPass_Proj_Bended$D3D9_VS11$Fog$CP#PosBending.cgvp
.HWScripts\Declarations\CGVShaders\Cache\CGVProg_Particle_DiffPass_Proj_Bended$D3D9_VS11$Fog$NoCP#PosBending.cgvp
.HWScripts\Declarations\CGVShaders\Cache\CGVProg_Particle_DiffPass_Proj_Simple$D3D9_VS11$Fog$NoCP#PosCommon.cgvp
```

CTRL: Query Height SHIFT: Smooth LMB: Flatten/Smooth +-: Change Brush Radius */: Change Height

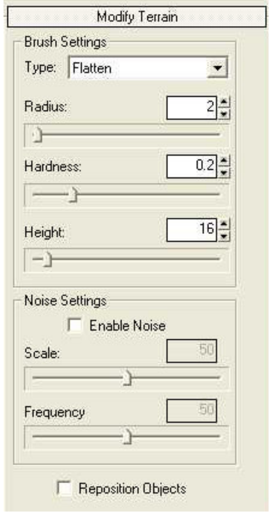
Figure 1.3 The Perspective View is excellent for navigating through your map.

Tip: you can alter the sea level by clicking the Terrain icon, and selecting Set Water Level from the Modify menu.

To edit the terrain you can switch to either perspective or map view, although perspective view is usually preferable. Once you have an area of the map you wish to edit, click on the terrain tab of the RollupBar and click the Modify button. In the Modify Terrain window that appears you will be able to select between Flatten and Smooth terrain. Flatten raises or lowers the terrain under the brush to bring it towards the value set in the Height parameter. The Height parameter can be set to any value between 0 and 255, although values over 150 are not all that common in most maps. One thing to consider when setting the height is the water level, which defaults to a height of 16. Anything above the height set for the water level will obviously be above sea level, and anything else will be below.

Tip: you can automatically reposition objects on the map as you edit the height of the terrain by checking the Reposition Objects check box.

For both the Flatten and Smooth brush you can alter the size of the brush, and the pressure at which it is applied. The Radius parameter defines the size of the brush's radius in Game Units, and the Hardness parameter determines how quickly the terrain will rise or fall to meet the new Height value. Higher values for Hardness will result in cliffs being produced, while lower versions will make for shallower slopes. The Flatten brush can also include Noise, with the Enable Noise check box ticked. With noise added, you can alter the Scale and Frequency parameters of the noise function, to alter the how bumpy the painted terrain looks. If you switch to the Smooth brush, it performs the exact opposite function, removing the bumpiness from the created terrain. For the Smooth brush, the Noise function is obviously not available.



Modify Terrain

Brush Settings

Type: Flatten

Radius: 2

Hardness: 0.2

Height: 16

Noise Settings

□ Enable Noise

Scale: 50

Frequency: 50

□ Reposition Objects

When working on heights there are a number of very useful hotkeys you can use to speed up your editing. You can increase and decrease the size of the brush radius with the + and - keys. You can also increase and decrease the height parameter with the * and / keys. Even better than this, however, is to alter the height parameter by sampling a height from the current terrain. If you want to make change the height of the surrounding terrain to the same height as you are working on, you can hold the Control key and click the area of the map where you want the height sampled.

Surface Texture

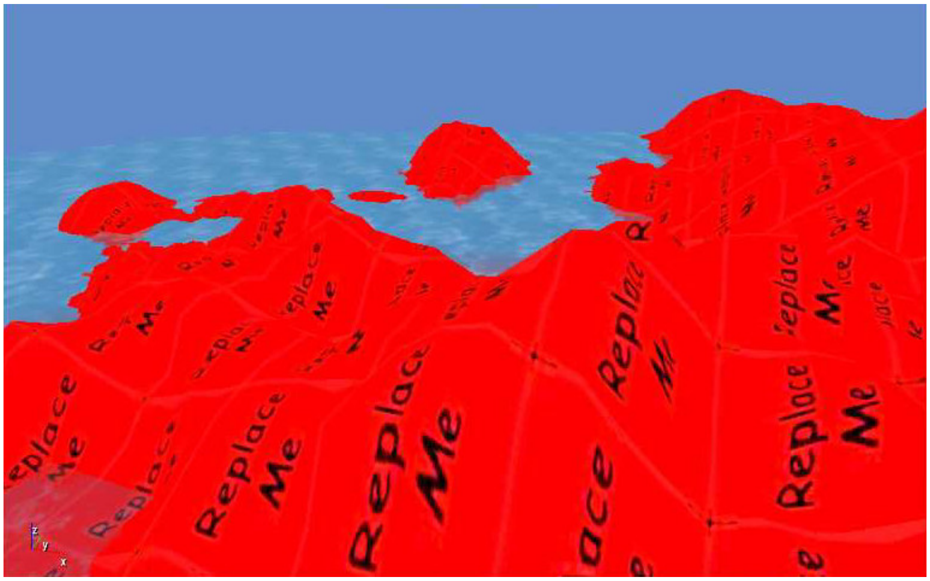
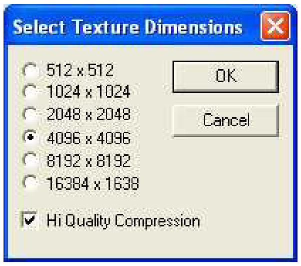


Figure 1.4 Untextured surfaces will be covered in red Replace Me tiles.



○ 512 x 512
○ 1024 x 1024
○ 2048 x 2048
● 4096 x 4096
○ 8192 x 8192
○ 16384 x 1638

☑ Hi Quality Compression

When using the Generate Surface Texture function, you will need to select the texture dimensions. The greater the resolution, the better the image quality, but the greater the CPU requirements.

You will notice that your landscape is covered in red Replace Me squares. These are your surface textures, and you will need to, as advised, replaced these with something of your own design. The surface textures are the basic elements that cover the map, like grass, sand and rock. You can include vegetation when you are painting these surface textures, but most of your vegetation will be added later as objects. When you have finished adding your textures, you must remember to select the Generate Surface Texture option from the File menu. It is worth waiting until you have a significant amount of texture to generate before selecting this function, as it can take several minutes to generate.

Terrain Layers

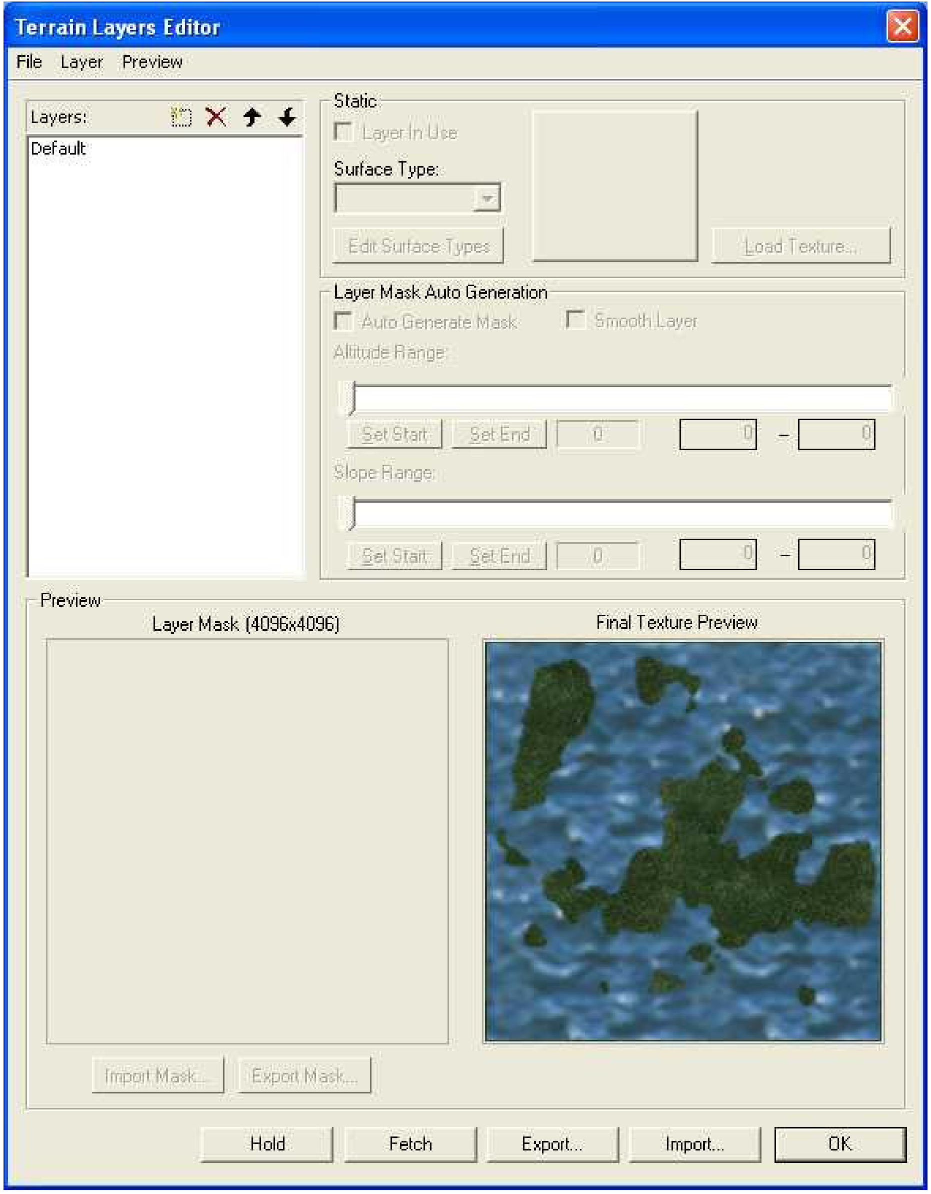


Figure 1.5 Use the Terrain Layers Editor to create the beaches, grass plains, etc. for your map.

The terrain surfaces in the Far cry editor are generated as layers, one on top of the other. To edit these layers click on the Texture icon on the tool bar, or select Texture from the Terrain menu. This will bring up the Terrain Layers Editor window, and from here you can create all the terrain textures you need for the map. There are four windows in the Terrain Layers Editor, Layers, Static, Layer Mask and Preview. The Layers window lists the names of all the layers in the map. The Static window displays the actual texture, and allows you to edit its finer details, such as the noise it makes when walked upon. The Layer Mask Auto Generation window sets the altitude and slope range that the layer will be applied to, and the

Tip: you can create layers that won't be used to auto-generate texture, but specifically for painting with the Layer Painter tool. The Layer Painter tool will help you perfect your terrain.

Preview window shows you what the map will look like based on the settings currently chosen.

The Layers window displays the layers you are using in the order in which they will be placed on the map when you run the Generate Surface Texture function. That means that the first texture layer will form the base, and each subsequent layer will be placed on top of it. For example, if you want a sand layer to appear on top of your default rock layer, then you must make sure that the sand layer comes after the default layer in the list. The icons at the top of the list of layers allow you to add new layers, delete existing ones and move them up and down in the order. If you want to rename the layer, simply double-click it.

Surface Texture

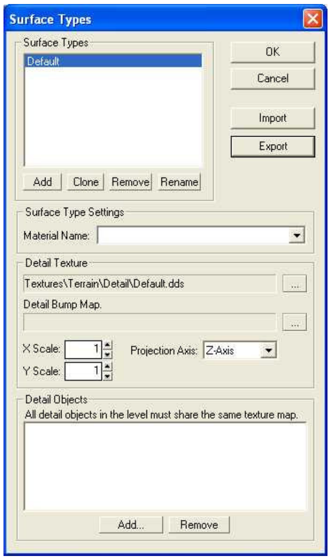
Next to the Layers window is the Static window. This allows you to select both the surface type, and the surface texture itself. In the preview box you will see the currently selected surface texture for the layer. To change this, click on the Load Texture button, and select a suitable texture from the terrain directory. If you want a different surface type to the Default, which isn't of great use, then you will want to create a new one. To do this click on the Edit Surface Types button, and a new window will pop up.

Note

You are only allowed a maximum of seven surface types, so use them wisely.

The Surface Types window allows you to define the surface type of the texture, for example how it sounds when you walk on it, what it looks like close up in detail, and what vegetation that will appear on the texture when it is applied. To create a new surface type, simply click the Add button under the Surface Types list, and give it a name. Then you will need to give the surface type a material, which you can choose from a drop down list of settings, for example you might want to choose the mat_sanddry material for a beach. The material defines how the surface responds to being walked on, shot at, etc.

You will also want to define a detail texture. This texture is for when you zoom in close to a texture layer. Without the detail, when you zoom in close to a texture layer you will see a nasty red ReplaceMe signs. You can set the detail texture by clicking the "..." icon and choosing a suitable detail from the detail directory. The detail texture doesn't necessarily have to be the same as your surface texture, as long as it works in combination. For example you can have a snow texture, with rock as the detail.



Surface Types

Surface Types: Default

Material Name:

Detail Texture: Textures\Terrain\Detail\Default.dds

Detail Bump Map.

X Scale: 1

Y Scale: 1

Projection Axis: Z-Axis

Detail Objects

All detail objects in the level must share the same texture map.

Also in the Detail Texture setting is the projection settings. When a texture is applied, it is applied from a certain direction, the default being from above, i.e. the Z axis. This can result in tiles getting stretched across surfaces that have long sheer faces, like cliffs. For certain textures, like those applied to steep slopes, you may wish to change the default projection axis from Z, to X or Y. The size of the texture's tiles is determined by the X and Y Scale settings. Decreasing the value of the X and Y Scale increases the size of the tile on the X and Y axis. If you experience tiling then you may wish to decrease this value to make the tiles bigger, but if they look stretched, then you may wish to increase the values.

Finally you can add objects to your textures, so that certain objects will be painted onto every occurrence of the texture on your map when generated. For example, you may wish to have flowers appear wherever you have a default green surface texture. To add textures, simply click the add icon, and pick an object detail texture from the objects/natural/details folder. You can add as many objects as you like to the list, and every time the texture generating function creates an instance of the layer, it will add the selected objects too. It should be noted that this is a somewhat rough means of applying objects to the map, and placing them manually is usually preferred by experienced editors.

Layer Masks

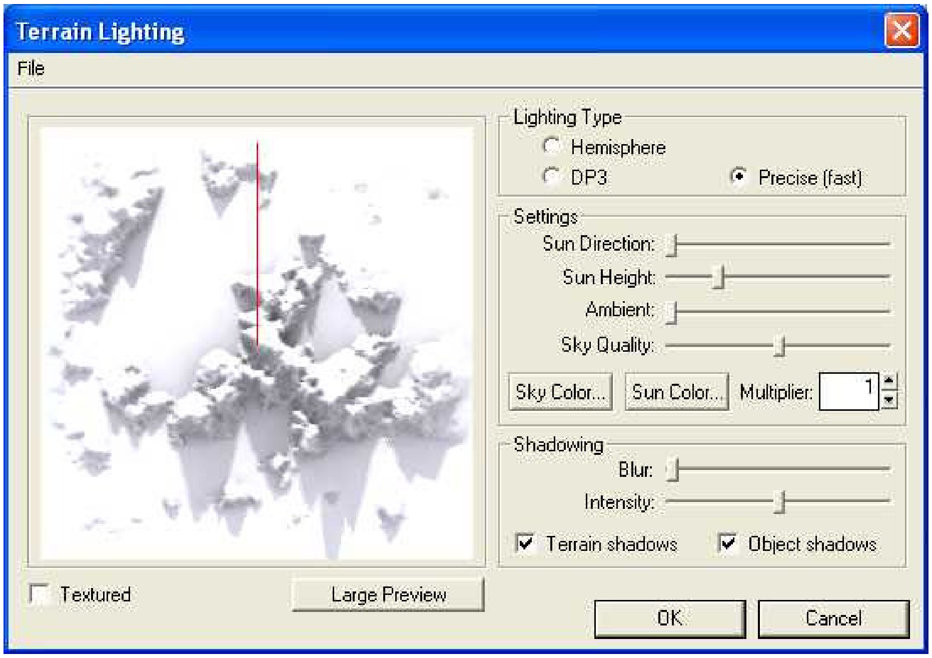
Underneath the Static window is the Layer Mask Auto Generation window, where you can select where on the map the new texture layer will be applied. To do this you need to create a mask, which you can do automatically using the altitude and slope range sliders, or manually through the use of an imported mask. To create a mask manually, you need to uncheck the Auto Generate Mask check box, and import a mask image from file. Mask images work in the same way as height map images, using a grey scale to determine where the texture is applied. The texture will be applied to anywhere that is pure white, but nowhere that is pure black. Any grey scales in-between will be blended accordingly.

For auto generated masks, the altitude and slope range sliders do all the work for you. The altitude range slider sets the range of altitudes across which the layer will be placed. For example, a sand layer can be placed across a layer that is just above and below the altitude at which you have set the water, say 10 to 20. To set the start range for the altitude, move the slider and click Set Start. To set the end range for the altitude, move the slider and click Set End. You can also set the start and end altitude values by entering these into the two edit boxes under the slide bar. The slope range slider works in the same way, only it allows you to define the range of slopes that the layer can be applied to. For example a rock layer can be applied to a range of slopes that are close to vertical, say from 200-255.

Note

Slopes are not measured in degrees, but as a ratio of 90 degrees, where 0 represents 0 degrees from the horizontal, and 255 represents 90 degrees.

Lighting and Environment



Lighting Type
○ Hemisphere
○ DP3
● Precise (fast)

Settings
Sun Direction:
Sun Height:
Ambient:
Sky Quality:
Multiplier: 1

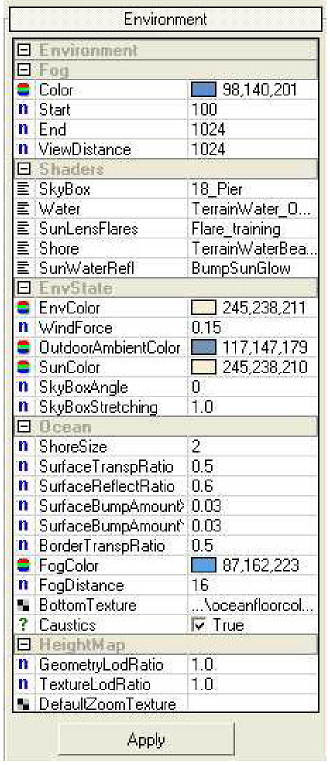
Shadowing
Blur:
Intensity:
☑ Terrain shadows
☑ Object shadows

□ Textured

Figure 1.6 The Terrain Lighting window allows you to alter basic lighting configuration for the map.

The first place you will want to go in order to change the lighting effects in your map is the Terrain Lighting window, which can be accessed via the Lighting icon on the tool bar. The key changes you can make here are to where the sunlight shines from. You can alter the direction the light shines from, and also from what height. For height the lowest setting is dawn/dusk and the greatest setting is midday. You can also set the colour of the sun and sky, which again affects the lighting in the map, but also the colour tints of the terrain surfaces and objects. Remember that if you want to see the effects of your changes to the sun and sky's colour, you need to regenerate the surface textures. However, the effects of changing the direction and height of the sun can be seen immediately.

Shadows are also an important factor in lighting your map. From the Terrain Lighting window you can change the Blur and Intensity of the shadows that affect your surfaces. The intensity of the shadows determines how dark they will appear on the map. You may also turn Terrain and Object shadows on or off. Terrain shadows are cast by mountains and other landscape features, while Object shadows are cast by objects placed on your map like vegetation and vehicles. Having these turned on significantly increases the amount of time it takes to generate textures, so you may want to turn them off until you have completed the design of your map terrain.



Environment	
⊟ Environment	
⊟ Fog	
Color	98,140,201
Start	100
End	1024
ViewDistance	1024
⊟ Shaders	
SkyBox	18_Pier
Water	TerrainWater_O...
SunLensFlares	Flare_training
Shore	TerrainWaterBea...
SunWaterRefl	BumpSunGlow
⊟ EnvState	
EnvColor	245,238,211
WindForce	0.15
OutdoorAmbientColor	117,147,179
SunColor	245,238,210
SkyBoxAngle	0
SkyBoxStretching	1.0
⊟ Ocean	
ShoreSize	2
SurfaceTranspRatio	0.5
SurfaceReflectRatio	0.6
SurfaceBumpAmount‹	0.03
SurfaceBumpAmount'	0.03
BorderTranspRatio	0.5
FogColor	87,162,223
FogDistance	16
BottomTexture	...\oceanfloorcol...
Caustics	☑ True
⊟ HeightMap	
GeometryLodRatio	1.0
TextureLodRatio	1.0
DefaultZoomTexture	

You can add some finishing touches to your terrain and the environment in the Environment window of the RollupBar, under the Terrain tab. Here you can alter the sun and sky colour, just the same as from the Terrain Lighting window, but you get a number of other options too. In addition the Environment settings allow you to alter the environment colour (EnvColor) and the ambient colour (OutdoorAmbientColour). Both of these settings affect the colours on the entire map, and setting either of them to a very dark colour will block out most of the colours on your terrain. They are best set to light colours, unless you want a dark map, say for a night mission.

The fog settings also affect the colour of the map, and how much of the map players can see. Fog is the environment effect which obscures objects beyond the view distance set for the map. You can set the colour of the fog here, as well as how far away from the player it starts and ends. The ViewDistance parameter defines the point at which objects start to become invisible to the player. There are fog effects for underwater too, and the fog colour and fog distance can be set for the "Ocean" in the same way as it can be set for on the land. This allows you to add a gloom to the underwater sections of the map, to make it look more realistic.

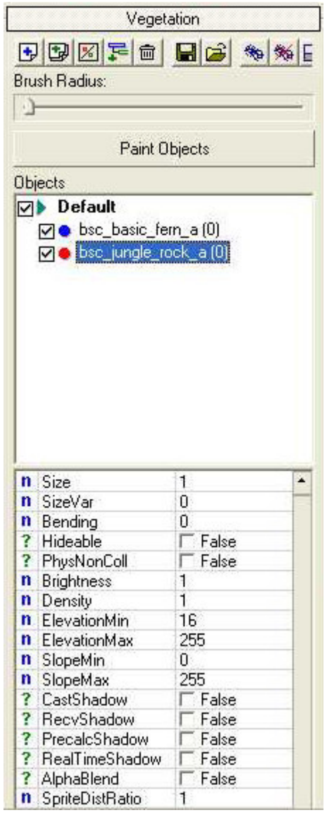
In addition to the colours and lighting effects upon the environment, there are a number of other useful settings in the Environment window. Probably the most dramatic effect you can change here is the SkyBox parameter. From here you can select what the sky looks like, from the sunniest of clear blue days with fluffy white clouds, to the most foreboding of stormy blue twilights, with dark brooding thunder clouds on the horizon. Pre-storm skies will likely require a fair bit of wind blowing through the trees, and so you can alter the WindForce settings to reflect this. The more WindForce the greater the degree to which the vegetation bends, and works in conjunction with the "Bending" parameter that you can set for your vegetation when painting it onto the map.

Note

See Appendix ****** for a details on the parameters of the Environment window.

Vegetation

Once you have a convincing looking landscape, you will want to add a layer of vegetation over the top to make it look like a living world, rather than a dead planet. You can place individual vegetation objects onto the map, and this will be explained in the next chapter on objects, but the first layer of vegetation you will want to put down will be with the Environment Brush. The Brush can be located under the Terrain tab on the RollupBar, by clicking on Vegetation. This will provide you with a list of available vegetation objects to be painted, along with the tools to paint them onto the landscape.

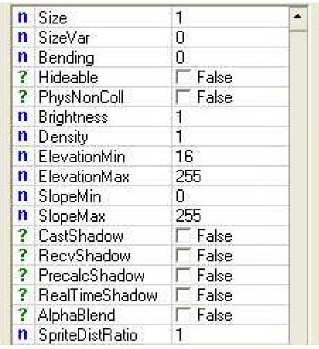


☑ Default
☑ ● bsc_basic_fern_a (0)
☑ ● bsc_jungle_rock_a (0)

n	Size	1
n	SizeVar	0
n	Bending	0
?	Hideable	□ False
?	PhysNonColl	□ False
n	Brightness	1
n	Density	1
n	ElevationMin	16
n	ElevationMax	255
n	SlopeMin	0
n	SlopeMax	255
?	CastShadow	□ False
?	RecvShadow	□ False
?	PrecalcShadow	□ False
?	RealTimeShadow	□ False
?	AlphaBlend	□ False
n	SpriteDistRatio	1

The first thing you will want to do is to add vegetation to the list of available objects to be painted. You will likely be starting from scratch, and so you will want to add vegetation to your brush list. You can do this by clicking on the left most icon in the Vegetation toolbar, the blue plus symbol. Once added you can click on the object in the list, click the Paint Objects bar at the top, and proceed to apply the vegetation brush to your map. If you add another vegetation type to the list, you can select them both by holding down the control key, this allows you to paint both objects at the same time. You can do this for as many objects as you like; all objects to be painted will have the red circle by their name.

The tool bar has a number of functions that give you a lot of power over your vegetation creations. Apart from adding and cloning objects, you can create new categories to work in. This allows you to select whole groups of vegetation types to paint with, or to delete completely. When it comes to deleting your vegetation, you can either remove it completely, which removes it from both the map and the object list, or you can replace it. Replacing a vegetation object swaps the selected vegetation object in the object window list with another chosen from file. This results in all instances of that previously chosen object being replaced on the map. For example, if you painted one type of grass onto a patch of terrain, and didn't like the look of it, you can swap all of the grass that you have painted to another kind by using this option.



n	Size	1
n	SizeVar	0
n	Bending	0
?	Hideable	□ False
?	PhysNonColl	□ False
n	Brightness	1
n	Density	1
n	ElevationMin	16
n	ElevationMax	255
n	SlopeMin	0
n	SlopeMax	255
?	CastShadow	□ False
?	RecvShadow	□ False
?	PrecalcShadow	□ False
?	RealTimeShadow	□ False
?	AlphaBlend	□ False
n	SpriteDistRatio	1

All of the vegetation objects have a number of settings. These settings are applied universally to all instances of the object that you paint with this selected. Therefore, if you use the instance of the object in the object list to paint a beach shrub on every beach on the map, then when you alter the particular parameters for that object instance, all the beach shrubs on all the beaches of your map will be affected. If you don't want to alter all the settings for all instances of a particular vegetation object on your map, then you will need to create two or more instances of it in the object list, so that you can paint them as individual brushes.

TIP: you can quickly change the settings for an entire group of vegetation, by selecting the category rather than the individual objects. For example, if you have a category set up as "trees", and you want to make all the trees sway at the maximum rate, you can select the "trees" category and alter the value of "Bending" universally for that category.

The settings have a number of key parameters that greatly affect the way the vegetation looks on the map. Size, clearly, alters the size of the object that is painted. Altering the value of Size scales the object so that a value of 1 is the default size and a value of 2 is double. The SizeVar parameter allows you to set the variance of the object's size as it is painted on the map. By default this value is zero, and means that all objects will be the same size. To give the map a more natural look it is necessary to alter the variance of the vegetation sizes on the map. With SizeVar set to a value other than zero, the objects will be painted onto the map with a size which varies from the norm by a percentage defined this variable. For example, if SizeVar is set to 2, then vegetation objects will be painted with a size ranging from the norm to 200% larger than normal. For other values, 0.5 gives 50%, 1 gives 100%, 3 gives 300%, etc.

Note

Shadows make for more realism, but can lower FPS rates.

You can also affect the way the vegetation appears on the map by changing its density. The value represents the number of game units, thus a value of 10 will set a density of one object per 10 game units, and therefore the lower the number, the greater the density. There are a number of lighting parameters too, allowing you to set whether the object receives shadows from other objects, or casts them itself. It should be noted that shadow settings can bleed the CPU of resources, and so are best applied sparingly to keep the FPS rate high. In addition to the lighting, you can also change the brightness of the vegetation object directly in the settings.

One final set of parameters that won't affect the way your vegetation looks, but will help you in painting the objects only in the locations that you want them, are the elevation and slope delimiters. The elevation delimiters are useful in preventing you from painting trees under the sea, and coral on mountain tops. The slope delimiters ensure that you only paint objects onto slopes within a range of your choosing. This can help if you don't want to paint trees onto the sides of sheer cliffs, or if you don't want your wall climbing vines stretched across flat beaches.

Note that like when you were creating layer masks, the slope angles are not defined in terms of degrees, but as a ratio of 90 degrees, thus a value of 255 is 90 degrees.

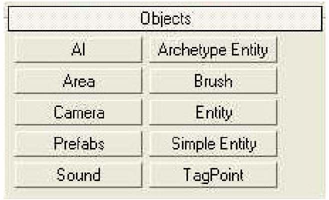
Chapter 2

Objects

This section discusses some of the most important objects in the editor; and examines how they are placed, manipulated, and organized.

Objects are the living part of your world, and consist of all the vegetation, buildings, people, animals, vehicles, boxes, etc. that you will want to place around your map. There are two categories that objects can be divided into dynamic and static. Dynamic objects, referred to by the editor as "entities", can be changed during run-time, e.g. they explode, move, etc. Static objects, however, are, like rocks, immovable. Most objects are of the standard solid variety, like trees and buildings, but there are a number of non-standard objects, such as shapes like the Forbidden Area that delimits where an AI can travel. There are also a number of special objects that act as triggers for events, waypoints for AIs, and other tasks. There are so many, and so many different, objects in the game that it is impossible to discuss them all in this manual. However, you can review a list of all available objects and their properties at the time of this revision, in Appendix *****************.

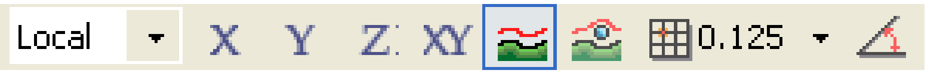
Object Placement



There are ten object groups, and you can place them onto the map either by drag-and-drop, or by point-and-click, depending on the object group. Although that may seem nonsensical, you can tell which type of placement method is required by the type of list you are presented with. If you see a file management type menu, then you can use drag-and-drop to move objects onto the map. If you see a simple list on a gray background, then you need only click on the object, and then move the object around on the map to place.

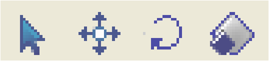
Tip: you can quickly set the locks and snaps by using the keys 1 through 5. The 1, 2 and 3 keys lock the X, Y and Z plane respectively, and the 5 key locks the XY plane. Pressing 4 sets the placement to snap to the terrain and toggles between snapping to objects or not.

Placing Standard Objects

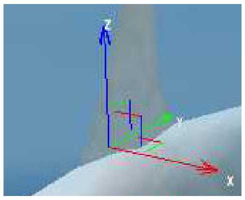


Another way you can make placement easier is by using the "snap to" options. You can snap the placement of objects locations to a grid, the spacing of which you can define yourself. In the image above the icon second from the left is a toggle, but also a pull down menu. You can toggle the grid by clicking the icon, and change the grid spacing by clicking on the arrow on the right of the icon. From there you can either select from a range of values, or set up the grid/snap values yourself. Setting up the grip/snap values yourself also allows you to alter the number of degrees the angles will snap to. The default value is 5 degrees, but you can change that to anything you want. The icon on the far right of the locks image above toggles whether angle snap is on or off.

Object Movement and Manipulation



Once you have placed the object on your map, there are a number of ways you can fine tune your placement. First you must select the object using the arrow tool. Simply clicking on the object, or dragging a box around all or part of the object, will select it. Once selected you can select the second icon from the left to enable you to move the object around the map. You can use the same movement lock icons as described above to make this assist placement. In addition to moving the object, you also have two more placement options for fine tuning. First you may rotate the object along any plane, and you may also scale the object to any size you desire, big or small.



The Gizmo is used to move and manipulate objects.

On every object you select there will be what is referred to as a "gizmo", as seen in the image above. You can use this device to move your object around, but also to scale and rotate. On the gizmo, the blue line indicates the Z axis, the red line the X axis and the green line the Y axis. Clicking on any of these lines turns it yellow, and means that all movement, rotation or scaling of the object will be along that axis. At the center of the gizmo there are three planes for each of the axes. Again, clicking on any of these planes selects it, and you can move, rotate and scale along them. If you want to move, rotate or scale while following the terrain, then click on either of the toggles for terrain lock with the object selected.

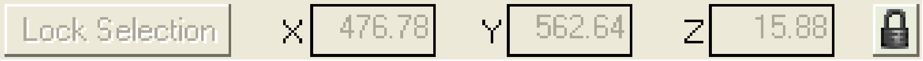
On every object you select there will be what is referred to as a "gizmo", as seen in the image above. You can use this device to move your object around, but also to scale and rotate. On the gizmo, the blue line indicates the Z axis, the red line the X axis and the green line the Y axis. Clicking on any of these lines turns it yellow, and means that all movement, rotation or scaling of the object will be along that axis. At the center of the gizmo there are three planes for each of the axes. Again, clicking on any of these planes selects it, and you can move, rotate and scale along

them. If you want to move, rotate or scale while following the terrain, then click on either of the toggles for terrain lock with the object selected.

Note

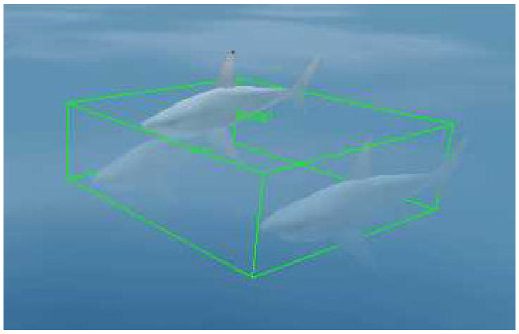
You can alter the relative projection of the gizmo on the screen by changing the relative view for the gizmo from the drop down menu to the left of the object selector icon.

Tip: if you want to concentrate on one selection, and not accidently start working on nearby objects, you can "lock" the selection by using the toggle next to the axes fields below the perspective window.

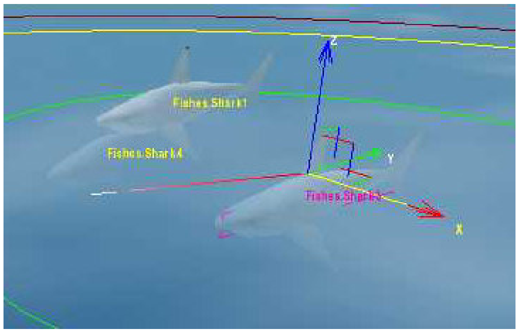

Lock Selection X 476.78 Y 562.64 Z 15.88

If you really want to fine tune your object placement to the last micron, you can use the axis fields beneath the perspective window. You can use this for moving, rotating and scaling, although when scaling it won't matter which of the X, Y or Z axes you change, as the change is applied to all three. For moving and rotating functions, changing any of the values in the X, Y and Z axes results in exact changes in their position and rotation on the map itself. For scaling the X, Y and Z axes are replaced by a single figure representing the scale of the object, set at 1. Increasing this value increases the size of the object, in relation to the objects original size, so a value of 2 will double its size.

Grouping and Linking Objects



You can easily work on multiple objects at the same time, say a set of barrels, by grouping the objects together. This can be achieved by dragging a selection box around every object you want in your group. Once grouped, you can move, rotate and scale the objects as one. You can add and remove objects from the group by holding down Ctrl and clicking the object; this toggles their inclusion. The group can be made permanent by selecting Group | Group from the top menu, and giving the group a name. Similarly the group can be disbanded by selecting Group | Ungroup from the same top menu.



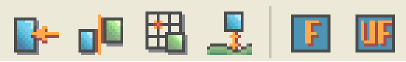
You can also link and unlink objects, using the link and unlink icons on the tool bar, to the right of the undo and redo icons. Linking objects is similar to grouping them, only the first object is linked to the second object in a parent-child relationship, where any action taken on the parent, such as rotate, will affect the child, but any action taken on the child will not affect the parent.

Tip: you can quickly generate multiple objects at the same time by pressing Ctrl and C while having an object selected to clone it.

Linking works by clicking the link icon, and dragging a red link line between the child object and the object you want to be its parent. To unlink objects, click on the object you want unlinked and click the unlink icon to remove all links to it.

You can add objects from these permanent groups by selecting the group you want to add to, selecting the object you want added, and then selecting Group | Attach from the top menu. Detaching objects from permanent groups is a little more difficult. First you must open the group, by selecting the group and clicking Group | Open. Then you must select the object to be detached from the group, and finally click Group | Detach. You can then close the group again by selecting Group | Close. This prevents you from clicking on individual objects within the group by mistake.

Object Management Toolbar



TIP: snapping to an object is very useful when you are trying to place objects upon other objects, such as a ladder on the side of a tower, or a weapon pick-up on the top of a table.

You can manage objects in the Select Objects window, which can be opened by clicking the list icon to the right of the lock toggles on the tool bar. This window lists all the objects in the map. If you want to jump to a particular object, select it on the list by double-clicking it. The window will disappear, and you can click the Go to Selected Object icon, on the left of the Object Management Toolbar. From the list, or by using the two icons on the right of the Object Management Toolbar, you can freeze and unfreeze objects or groups of objects. Freezing objects means that you can no longer do any work on them at all, you may not even select them. The Select Objects windows allow you a number of ways of displaying and filtering your objects, making them easier to manage.

In addition to the icons mentioned already, the Object Management Toolbar allows you to align selections to objects, align them to the grid, or set the object height. Aligning the selection to the grid will force the object to snap to the grid layout that you have selected. Aligning the selection to object, will cause it to become aligned to the next object you select. Setting the object height gives the object a height above its placement of the number of game units that you enter as a value. This value can be positive or negative, and moves the object in relation to its set position on the map, and is not an absolute value.

Placing Area Objects

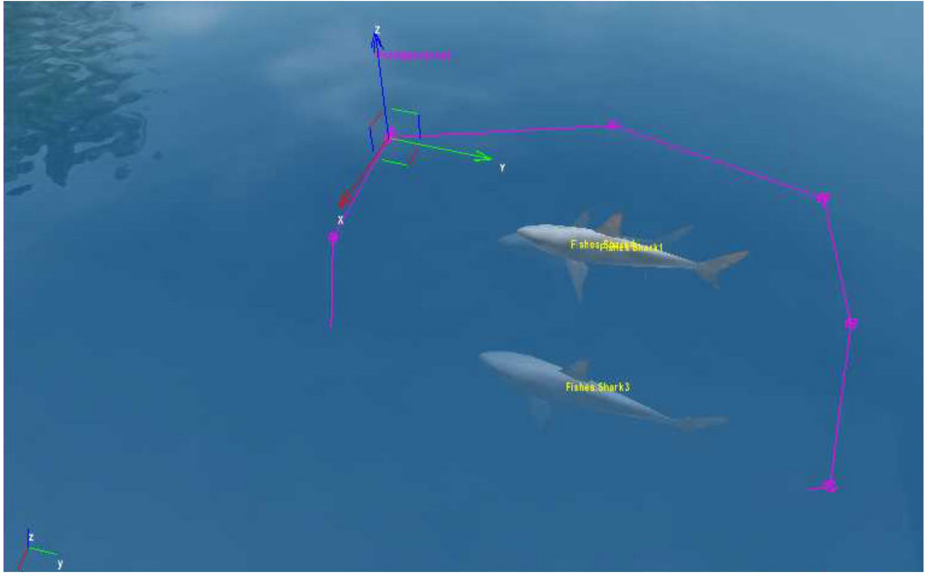
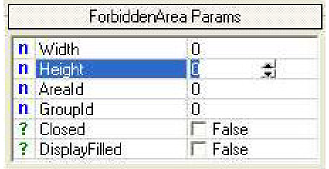


Figure 2.1 Create area objects by clicking each vertex point on the map.

Most objects are graphical entities of some kind, or at least a simple shape, that can be placed on the map without any consideration for boundaries. Some objects, however, are area based, and follow different placement rules to simpler objects. If you place an area object on the map, such as the objects in the Area list of the Objects window, or the Forbidden Area of the AI list, then you will need to define the objects boundaries. You do this by placing a start point on the map, and then indicating each vertex of the shape by clicking where you want it on the map. When you have finished defining the area of your shape, either double-click the last point to automatically close the shape, or click the starting vertex. Each point on the area shape can be placed like any object, and can use the same locks. It is usually best to lock the placement of vertices to the terrain.

All area objects have a height value, but it is not usually necessary to give them a particular height, as a height of 0 means it's only calculated as a two dimensional shape. Assign a shape a height only if you want to limit it in the third dimension, which will be not necessary for most outdoor shapes. A two dimensional shape is preferred to save speed. To give the area a height, simply enter a value in the Height parameter of the object in the RollupBar. Once placed, you can also edit the area object in the same way as other objects. You can move, rotate and scale just like any other object placed on the map. In addition you can edit any of the vertices of the area shape. To do so click the Edit Shape button in the Shape Parameters tab of the Objects window. Note that the vertices are at the base of the area object, and so if they are under the terrain you will need to raise them up before editing, or turn collision detection off.

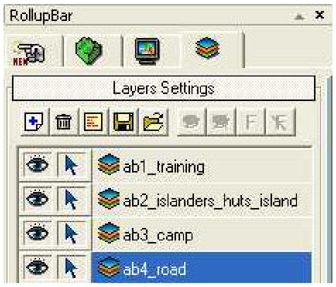

ForbiddenArea Params

n	Width	0
n	Height	0
n	AreaId	0
n	GroupId	0
?	Closed	□ False
?	DisplayFilled	□ False

Organising Objects

You will find as your levels get bigger that you will create large numbers of objects all over the map, and keeping on top of them can become difficult. To help you prevent you becoming lost in a sea of objects the editor offers several ways of organising your objects, as well as the way you view them. It is possible to organise your objects into multiple layers, to view, sort and select them in list format, and to hide objects on the map by their type. With careful planning, and appropriate use of these methods of organisation, it is possible to keep even the most complex of levels in a manageable state.

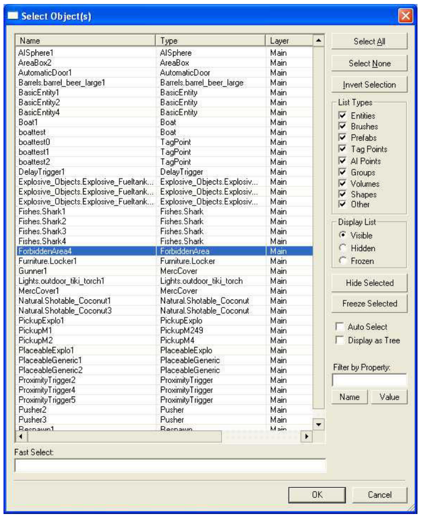
Layers



- ab1_training
- ab2_islanders_huts_island
- ab3_camp
- ab4_road

Layers can be organised in any way you feel suits your map, for example you can have a layer for each section of the map, like opening beach, hilltop battle, etc. You can create new layers in the Layer Settings tab of the RollupBar by clicking the plus icon and entering the layer's name. Once you have created a new layer, you can then lock the layer, so you can't edit it in any way, or even hide it from the map completely. Hiding objects you aren't currently working on can increase the speed of the editor significantly, and is a real boon when you have object heavy maps. To add objects to your newly created layer, you can select the object, or group of objects, and in the Objects tab of the RollupBar click the multi-coloured three tiered layer icon and choose the layer of your choice. For example, if you want to add all the icons on a beach to a Beach layer, drag a box around all the items, click the layer icon, and select Beach from the list.

Select Object Window



Name	Type	Layer
AISphere1	AISphere	Main
AreaBox2	AreaBox	Main
AutomaticDoor1	AutomaticDoor	Main
Barrels.barrel_beer_large1	Barrels.barrel_beer_large	Main
BasicEntity1	BasicEntity	Main
BasicEntity2	BasicEntity	Main
BasicEntity4	BasicEntity	Main
Boat1	Boat	Main
boattest	Boat	Main
boattest0	TagPoint	Main
boattest1	TagPoint	Main
boattest2	TagPoint	Main
DelayTrigger1	DelayTrigger	Main
Explosive_Objects.Explosive_Fueltank...	Explosive_Objects.Explosiv...	Main
Explosive_Objects.Explosive_Fueltank...	Explosive_Objects.Explosiv...	Main
Explosive_Objects.Explosive_Fueltank...	Explosive_Objects.Explosiv...	Main
Fishes.Shark1	Fishes.Shark	Main
Fishes.Shark2	Fishes.Shark	Main
Fishes.Shark3	Fishes.Shark	Main
Fishes.Shark4	Fishes.Shark	Main
ForbiddenArea4	ForbiddenArea	Main
Furniture.Locker1	Furniture.Locker	Main
Gunner1	MercCover	Main
Lights.outdoor_tiki_torch1	Lights.outdoor_tiki_torch	Main
MercCover1	MercCover	Main
Natural.Shotable_Coconut1	Natural.Shotable_Coconut	Main
Natural.Shotable_Coconut3	Natural.Shotable_Coconut	Main
PickupExplo1	PickupExplo	Main
PickupM1	PickupM249	Main
PickupM2	PickupM4	Main
PlaceableExplo1	PlaceableExplo	Main
PlaceableGeneric1	PlaceableGeneric	Main
PlaceableGeneric2	PlaceableGeneric	Main
ProximityTrigger2	ProximityTrigger	Main
ProximityTrigger4	ProximityTrigger	Main
ProximityTrigger5	ProximityTrigger	Main
Pusher2	Pusher	Main
Pusher3	Pusher	Main
Respawn1	Respawn	Main

List Types
- ☑ Entities
- ☑ Brushes
- ☑ Prefabs
- ☑ Tag Points
- ☑ AI Points
- ☑ Groups
- ☑ Volumes
- ☑ Shapes
- ☑ Other

Display List
- ● Visible
- ○ Hidden
- ○ Frozen

- □ Auto Select
- □ Display as Tree

Filter by Property:

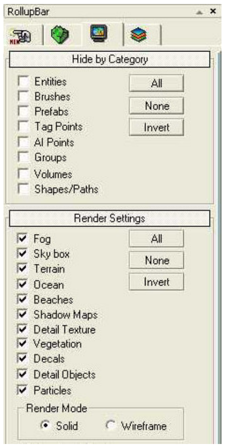
Fast Select:

The object list window is also a powerful way of organising, and maneuvering through, your growing array of objects. You can access this object list by either clicking the three lined icon to the right of the snap to grid and angle icons on the tool bar, or by selecting Select Objects from the Edit menu. The Select Objects window allows you to sort your objects by name, type and layer. You can select multiple layers, and hide them or freeze them, or bring them back to normal editing status. You can also filter out certain object names, and quickly search for them using the Fast Select option. Possibly the greatest function of the Select Objects window is locating particular objects in huge nests of objects on large levels. If you click on the object you want to locate on the map in the Select Objects window, click ok, and then click the Go to Selected Object icon on the left of the toolbar.

Note

If you jump to the select object window with an object or group of objects selected, they will automatically be highlighted in the object list.

Hide by Category



☐ Entities
☐ Brushes
☐ Prefabs
☐ Tag Points
☐ AI Points
☐ Groups
☐ Volumes
☐ Shapes/Paths

☑ Fog
☑ Sky box
☑ Terrain
☑ Ocean
☑ Beaches
☑ Shadow Maps
☑ Detail Texture
☑ Vegetation
☑ Decals
☑ Detail Objects
☑ Particles

● Solid ○ Wireframe

One final way of keeping the map from getting clogged up with objects that you are not working on, or are not interested in dealing with, are the Hide by Category options in the Display Tag. These allow you to hide objects by classification on your map, such as Entities, AI Points and Volumes. So, if you wanted to work on the placement of scenery objects on a section of your map, without the complicated links and tag points of the AI, you could switch all of these off. This window also allows you the opportunity to switch off rendering features on the map. This can give you a clearer view, but can also significantly increase the performance of the editor when the level becomes object heavy.

The Entity Library

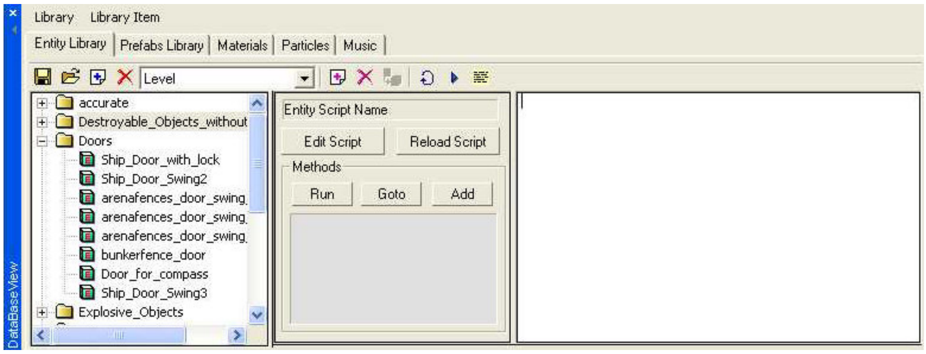


Figure 2.2 The Entity Library is an excellent source of ready to go objects.

Many of the commonly used objects are stored, with appropriate parameters, in the Entity Library, which can be accessed by clicking Show Database View in the Window menu. The database will show up as a window on your work screen, and can be shut down again once you have finished with it. The database is very useful for quickly applying objects to your map that can be difficult to get to work because they require particular parameters. The database provides ready-made "cookie-cutter" objects of almost everything that you might want in a Far cry level, including particle effects, destroyable objects, doors and switches. To access these objects, click on the folder icon in the database, and open any of the files in the EntityLibrary folder. These will then automatically be listed in the Archetype Entity list on the Object tab of the RollupBar for easy access. Once you have opened all the libraries you want, close the database to create more room for your work.

Destroyable and Physicalized Objects

Part of the dynamic object set are the destroyable and physicalized objects. The destroyable objects consist of plain breakable objects, computer screens, and explosive objects. All three types can be simply placed on the map, and will work as is. Consideration only need be given the explosive objects, as you don't want them killing or destroying objects and entities that you need for your mission to be successfully completed. Physicalized objects have physical properties that allow them to move about the map realistically. This means that you can push them, shoot them, drop them in water, and they will react like they were in the real world. Empty barrels float, cans dance when shot, and cobwebs flutter when you brush past them. There are also animated objects, like flames and fans, and even a coconut that you can shoot.

Doors and Switches

The doors and switches folder provides you with two quite different objects. The doors can be just dropped onto the map, and they will work as expected, automatically, with a push, or only with a keycard. You will obviously want to make sure the door opens up onto something, and that there is a building for the door, one that also fits its shape and design. The switches are objects that the player, or even the AI, can activate in order to set off an event, for example to set off an alarm. These switches are no use on their own, and need to be linked to an event, such as the alarm switch triggering an alarm sound, a flashing red light, and the sending of reinforcement troops to the area. Events will be described in a later chapter. One thing to consider when placing switches, or doors that require key cards, is to trigger a message when the player goes near, advising them of how to operate the switch or door, for example "press f to turn on the alarm".

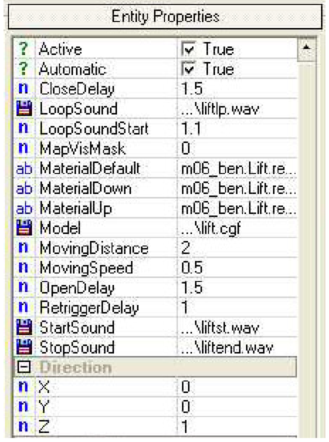
Particle Effects

Some of the finer touches to your level design can be added by the use of particle effects. The CryEngine Sandbox provides you with a number of ready-made effects for your map, including fire, smoke, spark, steam and water effects. These will add dynamism to your maps, but will usually need to be used in conjunction with other static objects. For example, you may want to use the steam with a broken pipe, or the sparks near some broken computer terminals. Some of the effects, like the cigarette smoke, are particular to one kind of action, and so will probably require a trigger to activate them, in order to work properly. The cigarette smoke won't look any good unless there is a soldier standing by it pretending to smoke one. See the chapters on AI and Events to learn how to put these two things together.

Elevators and Flying Foxes

The elevators in Far cry are complex, but also very flexible, beasts. They take a lot of event programming in order to work properly, but you can have a very simple one up and running in a couple of minutes. First place an AutomaticElevator object on the map, from the Elevator directory of the Entity objects list in the Object tab of the RollupBar. This Elevator object will be invisible, so you will need to change it from the default "lift" model. To do this click on the Model parameter, and then click the folder that appears to select a different lift model. You then need to make sure that the correct material is used, by changing the MaterialDefault, MaterialUp and MaterialDown parameters to reflect the new model. To do this you need to change the file names in these parameters to include the new name. For example, if the lift is lift_3x3x3m, then you need to change the Material file from lift or research_lift to lift_3x3x3xm.

If you now test the lift, it should start moving up the moment you enter it, but there is a problem. With the current release the lift will revert back to its original starting point the moment you enter the vehicle before moving upwards. That starting point defaults to zero, unless you reload the script, so click the Reload Script button now before you do anything else. The speed at which the lift moves, and the distance it moves can be altered by setting the MovementDistance and MovementSpeed parameters. If you want to change the direction the lift moves, then change the Z value in the directions category of parameters to -1, for down, or 1, for up. You can even make the lift go sideways, or diagonally, by changing the values of the X and Y axes from zero in the same category.



Entity Properties

?	Active	☑ True
?	Automatic	☑ True
n	CloseDelay	1.5
💾	LoopSound	...\liftlp.wav
n	LoopSoundStart	1.1
n	MapVisMask	0
ab	MaterialDefault	m06_ben.Lift.re...
ab	MaterialDown	m06_ben.Lift.re...
ab	MaterialUp	m06_ben.Lift.re...
💾	Model	...\lift.cgf
n	MovingDistance	2
n	MovingSpeed	0.5
n	OpenDelay	1.5
n	RetriggerDelay	1
💾	StartSound	...\liftst.wav
💾	StopSound	...\liftend.wav
⊟	Direction	
n	X	0
n	Y	0
n	Z	1

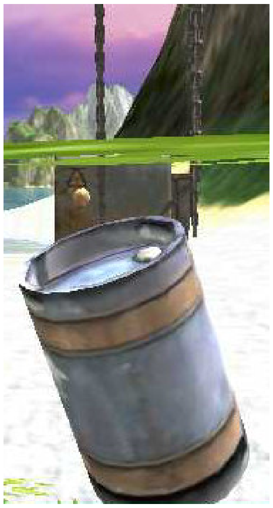
Note

There are far more parameters than this, and it requires a lot of programming to get the lift to work with AIs and doors, but for now this will be enough to give you a working lift. Later on, once you have learned about AI and events, you will feel happier about investigating the deeper aspects of lift design.

Flying Foxes are another way of changing altitude in Far cry, at least in the downward direction, and are a lot simpler to set up than elevators. To create a Flying Fox slide, you need a Flying Fox object from the same directory as the AutomaticElevator above, and a tag point. Drag and drop the Flying Fox on top of the place you want the player to be able to get down from, and place the tag point at the bottom, where he will alight the Flying Fox. Give the tag point a name, and copy that name into the destination parameter of the Flying Fox; this will tell it where to fly to. If you now test this, your player will be able to activate the Flying Fox and it will descend to the tag point destination. It will look strange, because it will have no cable or wire, so you will need to select one from one of the Brushes, and align it so that the Flying Fox careens down it from start to finishing tag point.

Other Objects

There is one final set of interesting objects on the Objects tab, and that can be found in the Others directory of the Entity file list. These include a variety of fascinating object types that fulfil a number of different functions. Probably the most interesting of them all is the Chain object, a physicalised object which can be used to hang other objects from. There are also DIY constructs for destroyable objects, breakable objects, etc., that can be used as a base for your own creations, as well as proximity damage objects, shoot targets and a number of other interesting things.



The Chain can be implemented fairly easily, for simple objects, but gets a little more complicated when you want to start attaching the chains to complex objects, like dead bodies, where you need to connect to individual bones. Chains can be attached to pretty much any object, as long as it has physical properties, like weight. What they can't be attached to are living objects, like animals and soldiers, and the chain will just fall to the ground if you try this. To attach an object to a chain, copy the name of the object you are attaching and place it in the AttachTo parameter of the chain. It is best to move the attached object close to the chain, and have the chain actually hanging from something, rather than thin air. The chain will attach to the nearest part of the attached object, and you can attach as many chains to the object as you like.

Note

You may find it easier to group chains and their attached objects together in order to move them around the map, but you will find that if you leave them grouped together, they will no longer work properly. You can however link objects together to move them, and they will still work as expected.

DIY objects, like breakable ones, can be customised to your own designs. For example, you can take the DestroyableObject entity from the Other directory, and change its Model to, say, an ammo_crate. Then you can change the ModelDestroyed box to ammo_crate_open to create a whole new destroyable object. You can also modify almost every aspect of its behaviour, including the amount of damage it causes the player, its explosion radius, and the sound that the explosion makes. Other objects, like Proximity damage objects can be simply placed on the map. The ProximityDamagae object can deal out damage to the player as he comes near to it. It can be used for fires, or even schools of piranha fish.

Lighting

There are two kinds of lighting that can be applied in the editor, dynamic and static. Dynamic lighting is calculated in real-time, while the game is being played, and is therefore highly expensive in terms of the computer's resources. Static lighting is pre-calculated, and while it doesn't look as realistic, it reduces the processing overhead considerably. Because of the ease at which dynamic lighting can be placed on the map, and the dramatic effects it can produce, it can be very tempting to throw them down everywhere. However, carefully planning and a good balance between static and dynamic lighting is necessary in order to ensure the level runs on machines that aren't powered by super-cooled parallel processors.

Dynamic Lighting

The Dynamic Light object has a number of parameters that can affect its properties, the most important of which are those that affect how it projects light. After all, what is the point in using dynamic lighting if you aren't going to show of its greatest strengths by having it plaster flourescent pulsating light on every corner of the room, while swinging violently around its axis. The simplest parameters you can change are the dynamic light's colour and style. The default colour and style is boring white and a plain direct beam. You can change the RGB value of the light in the Specular parameter, for example pink. In the LightStyle parameter you can change the light style from a steady beam to various kinds of pulsating, flashing or strobing lights.

Note

See the Objects Properties listing in Appendix ***** for a more detailed look at the Dynamic Light objects parameters.

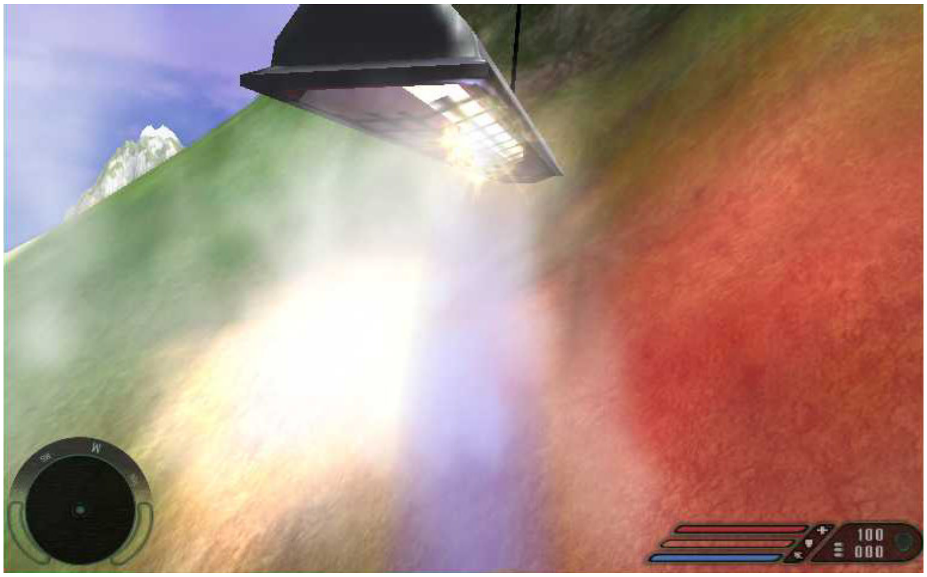


Figure 2.3 Dynamic lights add amazing realism to your maps.

In addition to strobes and pink flashes, you can also make the light swing about its axis, or even swing wildly in orbits around its axis. To do this you need to set the light to shake. This can be achieved by setting the shakeRefreshTime parameter to anything but zero, which is the default. The light will then automatically shake every time the parameter is reset, thus the larger the value you enter for shakeRefreshTime the less often the light will shake. The degree by which the light shakes is determined by shakeAmount, which defaults to a rather large 100. You can reduce the shaking by increasing the damping parameter, or increase the shaking by increasing the max_step_time parameter. By default the light is set to the first model type, which is usually an invisible box. You can alter the model type by either pointing to a different model type, 1, 2, or 3, with the lighttype parameter, or by changing the model types themselves.

Note

You need to reload the script after changing the light-type, by clicking the Reload Script button. If you don't, then the light may not look as it should, and may rotate around a very eccentric axis.

One final parameter of dynamic lighting is the lightshader parameter, which can dramatically affect the appearance of the light. You can choose a shader from the Select Shader window - light shaders usually start with the word Light, for example LightFlicker_flare2. There are many different shaders, but they mostly fall into these groups: beam, flicker, pules, and sway. Beam directs a strong direct beam of light from the source, flicker makes the light source flash on and off, pules creates a pulsing light source, and sway moves the swings the light source slowly from side to side, like a hanging lamp that has just been pushed. There are many of these effects, some of which are not detailed here and some that are combinations of effects. You will need to experiment to discover the properties of them all.

Static Lights

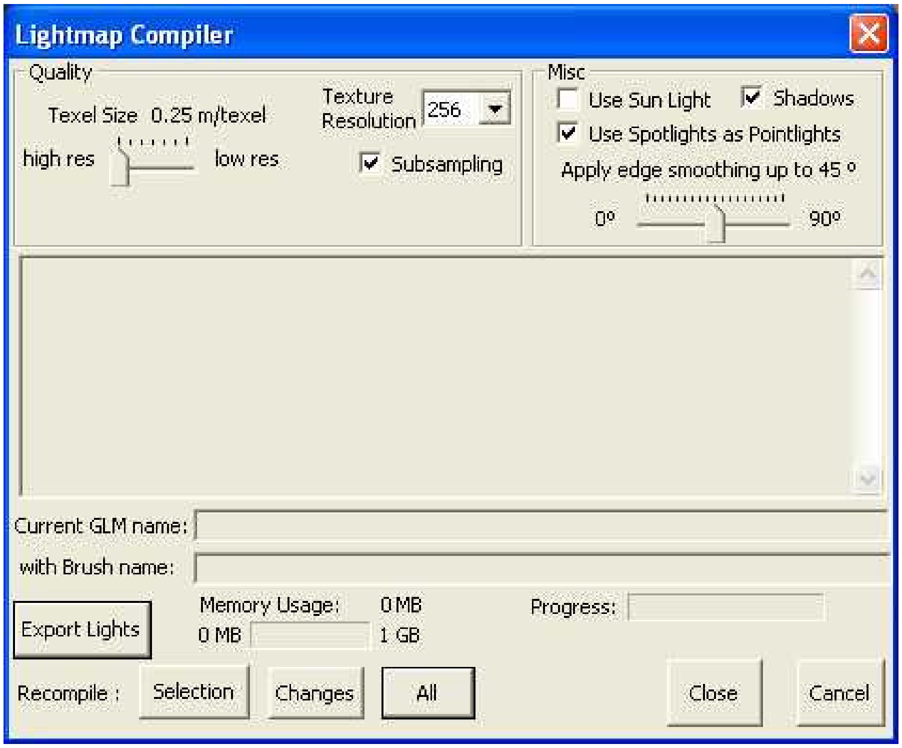
Static lights and objects can have their light and shadows "mapped" before the level has even been started, thus saving valuable processing power. The light map itself is a static texture, and should only be applied to static objects. You can apply it to dynamic objects, but if they should move during the game, the static lightmap texture will remain, making the scene look unrealistic, with the light source acting as if the object was still there. To produce the lightmap texture, all lighting must be pre-calculated using the Generate Lightmaps function in the Game menu.

When creating lightmaps the generator uses the following parameters of each object to calculate the correct texture:

- CastLightMaps - Sets whether an object casts lightmap.
- RecvLightMaps - Sets whether an object receives light from light casting objects (ignored for lights).

Usually both paramaters are set to true.

When you are ready to generate your lightmaps, call the compiler with the Generate Lightmaps function.



Lightmap Compiler

Quality

Texel Size 0.25 m/texel

high res – low res

Texture Resolution 256

☑ Subsampling

Misc

☐ Use Sun Light ☑ Shadows

☑ Use Spotlights as Pointlights

Apply edge smoothing up to 45 °

0° – 90°

Current GLM name:

with Brush name:

Memory Usage: 0 MB

0 MB – 1 GB

Progress:

Figure 2.4 The Lightmap Compiler generates all the non-dynamic lightig for your map.

There are a number of settings to consider before running the compiler, the first of which being the quality of the lightmap. In the quality box is a Texel Size slider and an Subsampling check-box. A texel is a ratio of the size of lightmap to the world map. The smaller the size of the texel the greater the quality of the texture, but the greater the demand on the computer's resources. Larger sizes result in poorer quality graphics, but greater speed. Subsampling determines whether shadow anti-aliasing is applied, which again improves image quality, but drains processing power. As subsampling can be turned off by the user in the options, you can turn it off yourself for testing purposes, and then turn it back on for the final product, to keep the testing environment from becoming too sluggish.

In the Misc box you can set the Texture Resolution. The resolution you will want depends on both the size of the objects that are being light mapped, and the size you have set your texels. If the texture size is set too low, you may find errors occurring during compiling. If it is too big, you will waste texture memory. You will need to strike a balance which will create the best texture for the least resources. In addition to the Texture Resolution, you can set whether the compiler uses sunlight and shadows in its calculations, by setting the appropriate check boxes.

When you have finally configured your pre-compiler settings, you will want to run the compiler itself. There are three options for compilation, Selection, Changes and All. Selection will only calculate light maps for the objects you have selected. To get the correct shadow casting with this option, you may have to select more objects that you need to calculate for. The Changes compilation option reviews the map for changes in light source properties, and recalculates the light map for only the objects that need it. Finally the All option recalculates the light map for all objects. This creates a completely new light map file, unlike the Selection and Change options, and can take several hours to complete. This compilation time means that is best reserved for only finished levels where it can significantly increase rendering speed.

Chapter 3

Artificial Intelligence

A guide to placing and controlling Artificial Intelligence entities in your levels.

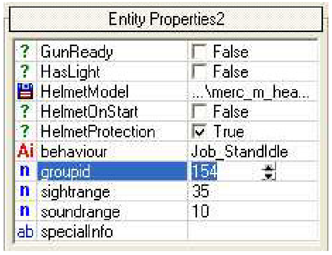
Artificial Intelligence entities include ordinary soldier units, like mercenaries, mutants, vehicles, like attack boats and gunships, and animals, like the pig. While they do have a large degree of autonomy programmed into them, you need to tell them what to do, in order for them to do it. If not they will tend to stand around and wait for something to happen, before filling it with a ton of lead.

Placing AI Objects

There are two ways of placing AI objects on the map, the easy way and the hard way. The easy way is simply a matter of dragging and dropping pre-designed AI objects that are all set up to run and gun. The hard way is to drop an unfinished AI object, and set up all the parameters yourself. The former is obviously quicker, but the latter offers you more control over the behaviour and properties of the objects you place. To quickly place AI objects on the map, simply select the pre-designed AI object from the Archetype Entity list of the Objects window in the RollupBar. For example, a MercLeader_Defensive_M4_ will provide you with a Mercenary Leader who will protect a defined point, and will carry an M4 gun. To place an undefined AI object on the map, select one from the Entity list, and set its properties. Use the object property listing in the Reference Manual to assist you.

Tip: in order for your AIs to respond correctly when under fire, they need to know which objects they can hide behind for cover. For every object on your map which an AI can hide behind effectively, make sure that the Hideable property is ticked and set to True.

When placing AI entities close to each other, you may want to consider grouping them together. You can do this simply by ensuring that each AI you want in the same group has its groupid set to the same value as every other member of the group. When AI entities are grouped together they behave differently to when they are on their own, especially if you include a Leader type AI. Unless specified otherwise, the AI group will not react until it spots an enemy, at which point the leader takes over, and the individual members follow their respective roles. For example, a defensive leader will move to defend a pre-defined spot on the map, while the Merc_Cover entities will provide cover for the AI entities that try to flush the enemy out. It is important to consider what members



Entity Properties2

?	GunReady	□ False
?	HasLight	□ False
💾	HelmetModel	...\merc_m_hea...
?	HelmetOnStart	□ False
?	HelmetProtection	☑ True
Ai	behaviour	Job_StandIdle
n	groupid	154
n	sightrange	35
n	soundrange	10
ab	specialInfo	

you include in your group, as a well balanced combination can result in effective AI, but badly combined groups can result in bumping and confusion.

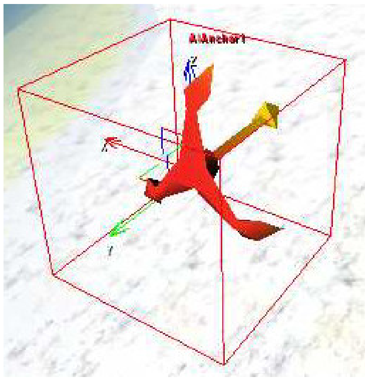
Tip: you can quickly change the group number, or any shared parameters of a group of objects. Selecting multiple objects shows you the parameters they share with each other, in the case of AI entities you can change their group number when selected together, and every entity selected will be given the same group number.

Note

To create a point on the map for a defensive leader to protect, then place an AI Anchor on the map, and set its property to AIANCHOR_PROTECT_THIS_POINT. Make sure that the Anchor point is not in a place that is inaccessible to the leader, such as inside a Forbidden Area, or behind walls and objects that are impassable or too complicated for the AI to calculate a path through.

Controlling AI Actions

Anchor Points



If you don't tell the AI entities otherwise, they will stand around doing nothing until they spot and enemy. To give the AI a more realistic look, you will want to program them to do something. The simplest way of creating actions is to place AI Anchors on the map near to your AI entities, and set the Anchor's property to some form of action or other. For example, you can place an AI Anchor down, and in the Actions property of the object, select AIANCHOR_SMOKE. When you test the AI, remembering to Generate AI Triangulation first, nearby AI entities will walk towards the anchor and start smoking a cigarette. You will probably want to place a cigarette smoking particle effect near to the soldier to make it look more realistic.

Note

The AI Anchor can be found in the Objects window of the RollupBar, in the AI listing. To access its Action property, click the parameter value, and then click the three dots that appear next to it to select an Action property from a list provided.

AI Paths

A more complex means of getting AI entities to move around your map is to make your entities follow paths. The simplest path you can create for your AI is to set down a number of AI Anchors, as in the previous section, and to set the AI Anchor property to INVESTIGATE_HERE. Then set the AI Behaviour property of the AI entity that you want to control to Job_Investigate. After generating the AI triangulation and testing the AI, the AI entity should wander between the INVESTIGATE_HERE AI Anchor points at random, until interupted by the spotting of an enemy.

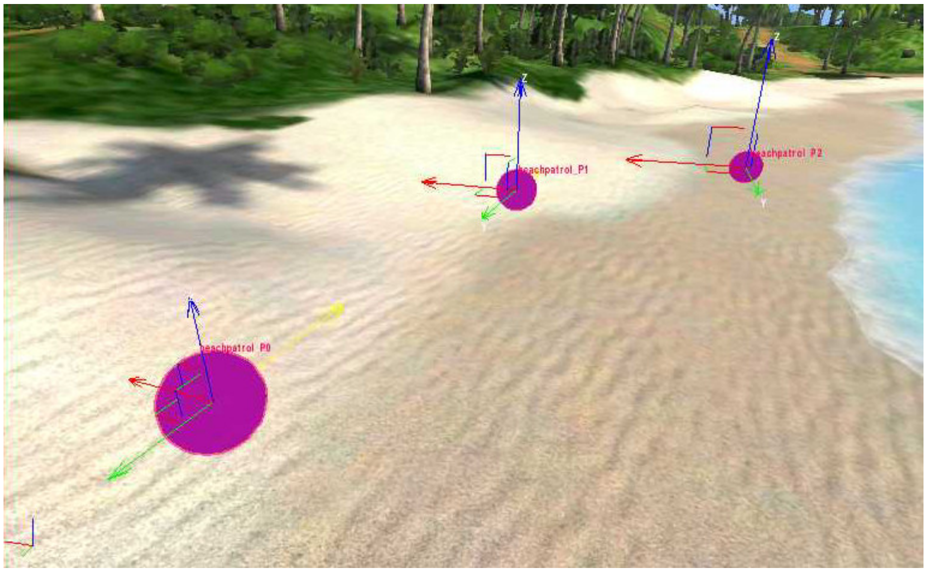
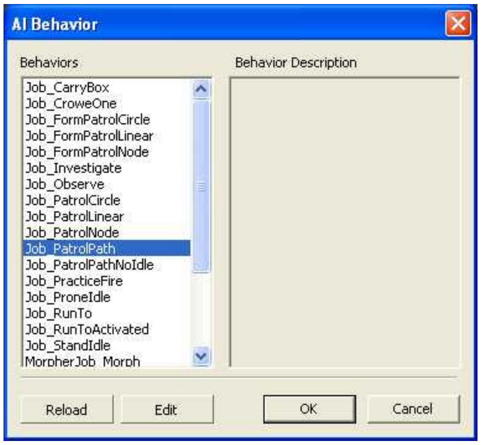


Figure 3.1 Create paths for the AI to patrol using only tag points.

Instead of using Anchor points, you can gain even more control by using Tag Points from the TagPoint list in the Objects window. In order to get the AI to follow your tag points, you must name the tag points and the AI correctly. Each tag point that is to be followed must follow the naming convention name_Pn, for example you may have three tag points called beachpatrol_P0, beachpatrol_P1 and beachpatrol_P2. Then the AI entity that needs to follow these tag points must share the same name of the tag points. In the example we are using here, the AI entity following the path must be called beachpatrol.

Simply giving the AI entity the same name as the tag points isn't enough on its own, as it also needs to know how to negotiate the tag points. Once the tag points and AI entity have been named, set the AI Behaviour property of the AI entity to Job_PatrolCircle, Job_PatrolLinear or Job_PatrolNode. All of these patrol behaviour results in the AI entities walking randomly from one tag point to another, however the style influences the randomness. A circle patrol results in a circular style of patrol, linear will produce a linear patrol route, and node will simply result in the AI entity wandering at random from one tag point to another, with no particular route at all.



Tip: be careful about the constituents of your group, as they will influence the way in which the group forms and moves from tag point to tag point. For example, a Merc_Rear will take up the rear of the group, and if you place him initially in front of the group, he will get in everyone's way as he makes his way to the back of the group.

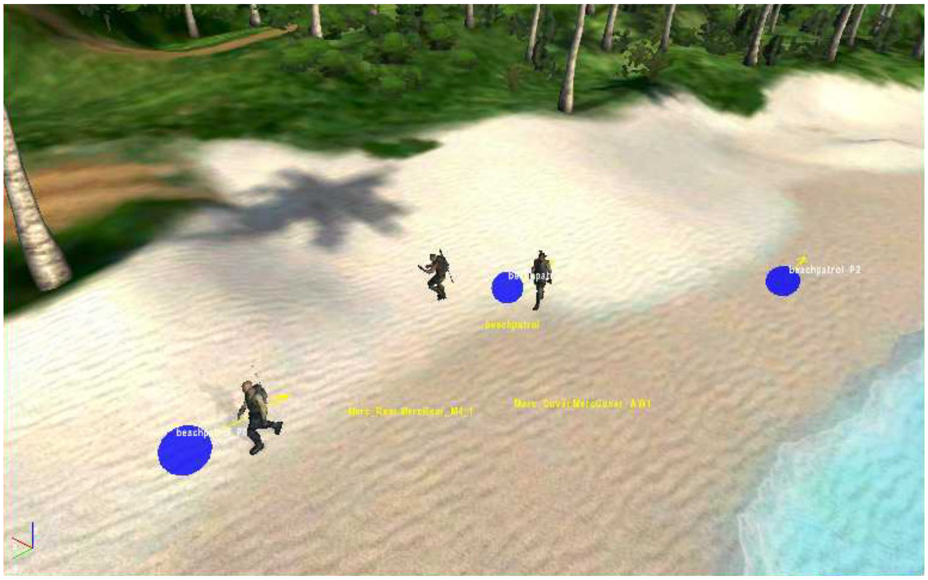


Figure 3.2 Once grouped, AI entities will play follow the leader, taking up their positions as accorded by their roles.

It is possible to use the tag points to create a patrol group, where a leader forms a group that wanders from one tag point to another. To create a patrol group, follow the instructions already given, but instead of giving the AI entity an AI Behaviour property of Job_PatrolCircle, etc., choose one of Job_FormPatrolCircle, Job_FormPatrolLinear, or Job_FormPatrolNode. These are exactly the same as for single entities, but they cause the entity to be followed by other entities in the same group. Make sure the entity with the form patrol group behaviour is a leader, and that the entities that you want to follow have the same groupid as the leader, and have the AI Behaviour properties set to Job_Observe.

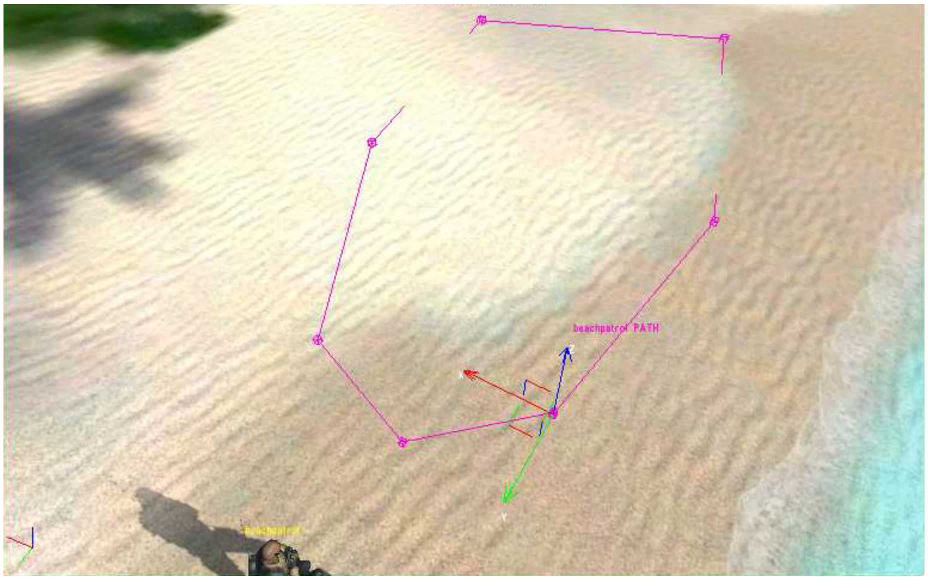


Figure 3.3 AI entities can also be programmed to follow strict paths.

When using tag points, the movement between the points is not exact, and the AI will wander somewhat from the route. If you want the AI entity to follow an exact path, you can create an AI Path. To do this select AIPath from the AI list in the Objects window. Then click on the map where you want the path to start, and follow that by clicking on all the points on the map you want your AI to go to. When you are finished, double click the last point, and the loop will automatically be closed. Then, similarly to the tag point patrol, name your AI Path by the following convention, name_PATH, for example beachpatrol_PATH. Then ensure that the AI entity that needs to follow this path has the same name, e.g. beachpatrol. The AI entity also needs the correct AI Behaviour, and you can choose between Job_PatrolPath and Job_PatrolPathNoIdle. Both options result in the same patrol route, but with NoIdle the AI will not pause at each point in the path.

Restricting AI Movement

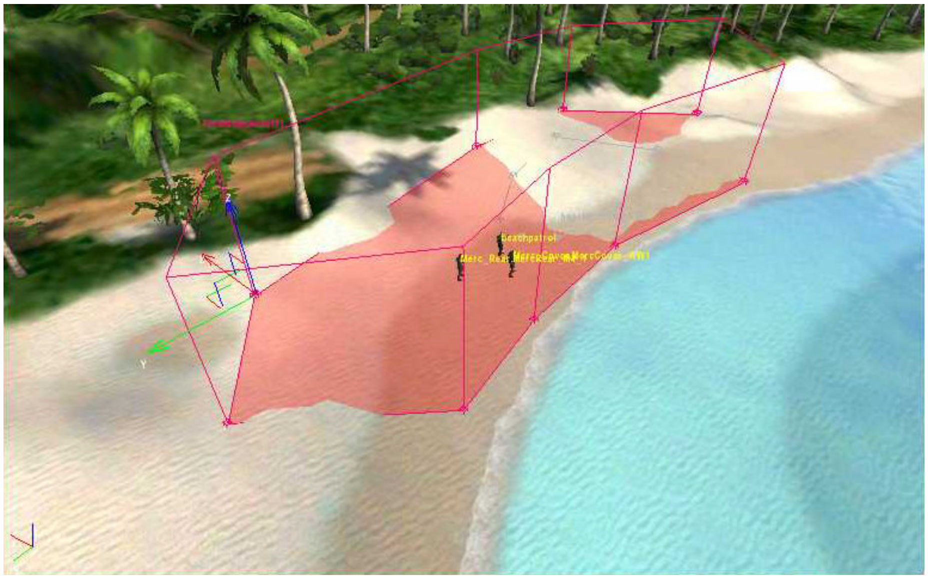


Figure 3.4 Keep AI entities caged and controlled with Forbidden Areas.

Tip: make sure that forbidden areas do not overlap each other, otherwise the newest forbidden zone will be cancelled out by the zone already created.

When wandering randomly between tag and anchor points, or when responding to enemy movement, the AI entities can stray from their positions and go places where you don't want them to, such as wading into the sea, getting stuck on rocky outcrops, or falling off cliffs. In order to prevent them from doing this you need to create Forbidden Areas. Forbidden Areas can either prevent an AI entity from entering, or from leaving, depending on whether the area is created with the AI inside or outside. To create the forbidden area, simply click on the ForbiddenArea option from the AI list in the Objects window. Then, on your map, click points to outline the the area that you want to make into a forbidden area, before double clicking the last point to automatically enclose the area. Because the forbidden area is likely to cover uneven ground on different levels, you will want to give the area a height in order to ensure that it covers everything necessary. You may also want to move the entire area up or down, to give complete coverage.

Movement in Non-Standard Areas

In certain areas of the map, such as on objects like bridges, and inside buildings, the AI can get confused, or even refuse to enter. To solve this problem we need to treat such areas differently to others. We also need to create Forbidden Areas to prevent the AI walking into these zones by accident, and risking getting in a mess, and also to create means by which the AI can move into the forbidden area correctly. To do this we need to create an AI Navigation Modifier zone, to isolate the non-standard area, as well as creating exits and entrances for the AI to move into the area safely. These entrances and exits need to be linked to AI Waypoints, to enable the AI to navigate its way through the non-standard area. The Waypoints can also be augmented by Hide Points, to give the AI somewhere to seek cover when under fire.

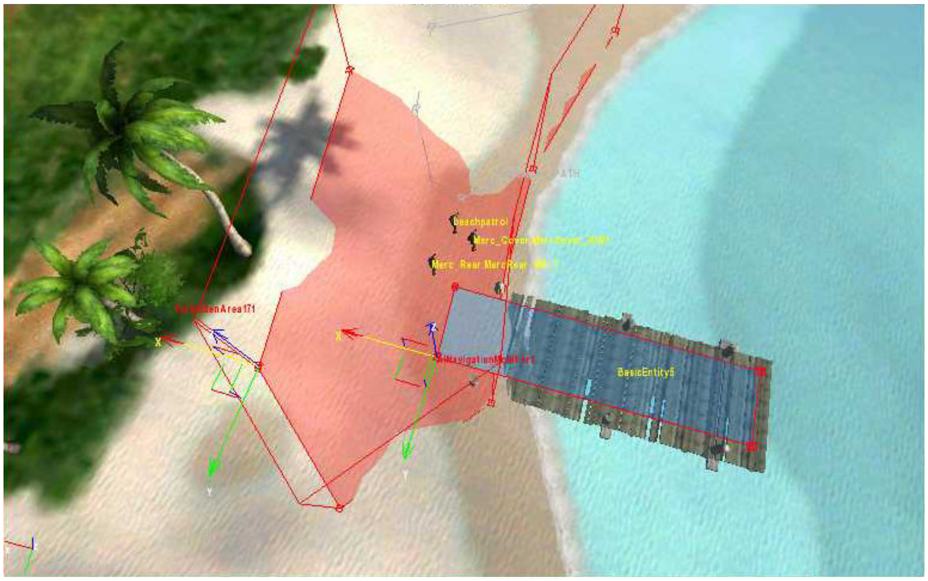


Figure 3.5 Allow AI entities to walk onto bridges, and into buildings, etc., using AI Navigation Modifers.

In order to facilitate movement in non-standard areas, you must first create an area called an AI Navigation Modifier. To do this select the AINavigationModifier option from the AI list in the Objects window. Then click on the map to outline the area you want covered, just as with the forbidden area, double-clicking the last point to automatically enclose it. When creating the AI Navigation Modifier it is vital that the area overlaps the area where the AI is allowed to walk, such as a forbidden zone enclosing the AI. It needs to overlap these areas where the AI can both enter and exit. Once you have created the AI Navigation Modifier, you must give it a height value, 10 is usually sufficient, in order that it covers everything sufficiently. You may wish to move the area up or down to give better coverage.

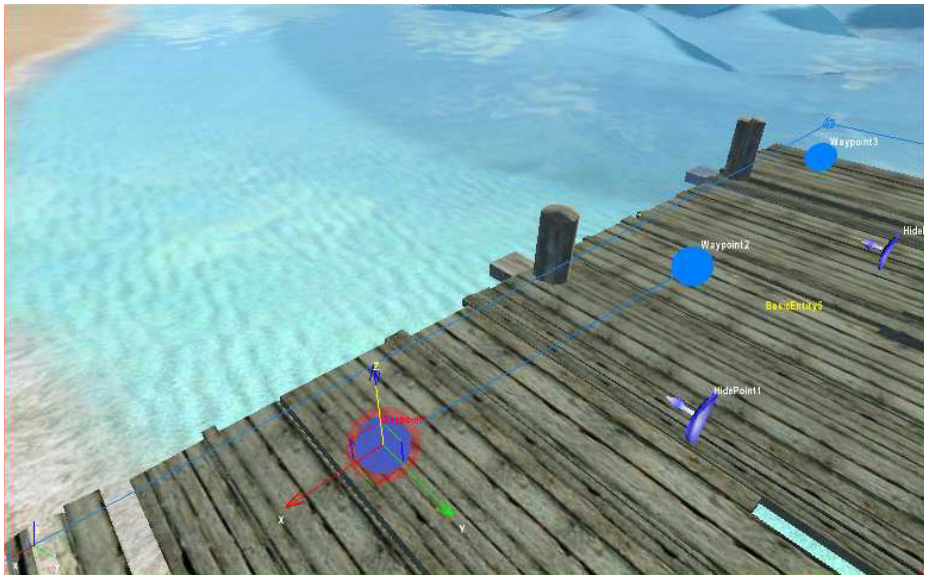


Figure 3.6 Guide the AI around the Navigation Modifier, by placing way points in a path from entry to exit.

Once the AI Navigation Modifier has been created, place an AI Point on the both overlaps where you want to place an exit and entry point. Set the Type property on these AI Points to Entry Point and Exit Point. It is important where you place these as the AI will enter and exit an Entry Point, but only leave from an Exit Point. With these points set, you want to place AI Points in all the places inside the non-standard area that you want the AI to be able to move to. For each of these points that you set, you want to set the type to Waypoint. Once placed you must then link all of these points together. You may link them in any way you choose, but there must be a continuous and uninterrupted link of waypoints from the Exit Point to the Exit Point; if there is a gap in-between, the AI won't enter the area.

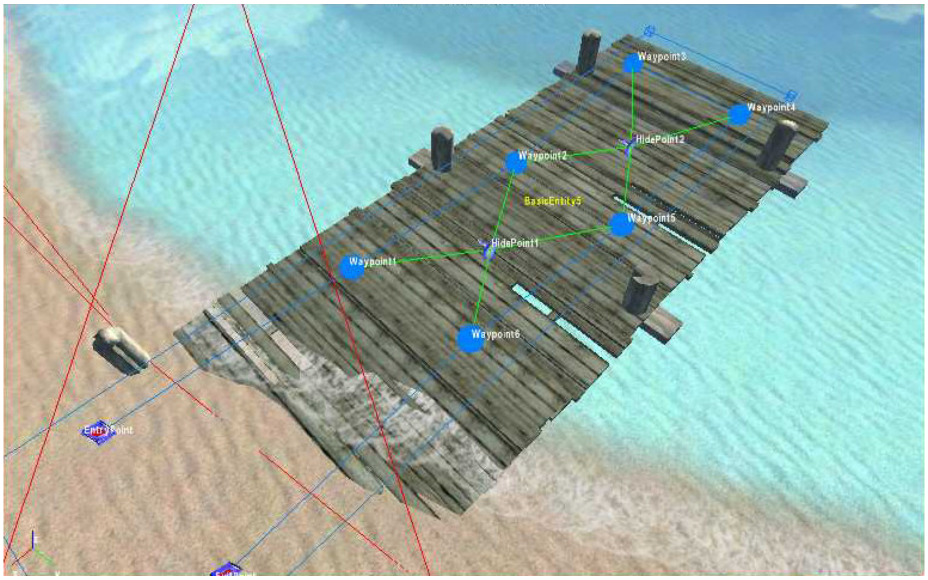


Figure 3.7 Way points must link in an uninterrupted path, from entry to exit.

The points can be linked by clicking Pick, underneath the Linked Waypoints list on each Waypoint, Exit Point, Entry Point and Hide Point, and then clicking on the next point you want to link. You can link as many points as you like, and all points that you link will automatically update their own Linked Waypoints list, so there is no need to back link objects. In addition to the link types already explained you can also augment the AI's behaviour with Hide Points. You can place and link these in the same way as the other points detailed, and they will allow your AI somewhere to hide when needing to find cover from enemy fire. These Hide Points must not replace way points in the continuous link between Entry and Exit points, or the AI will refuse to enter.

AI Vehicles

Land and Sea Vehicles

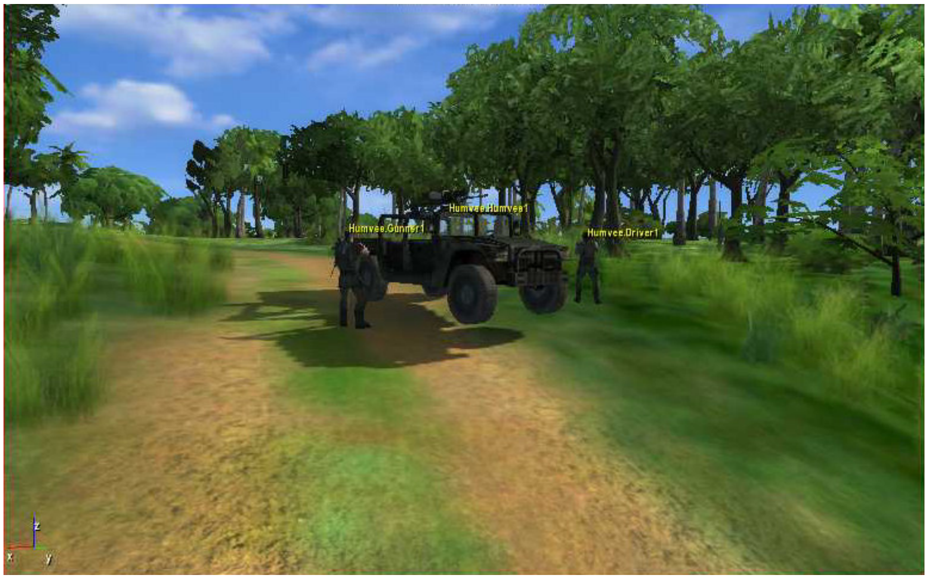


Figure 3.8 Place driver and gunner near to the vehicle they will pilot.

There are a number of land vehicles in the game, and they all follow similar rules when setting their behavior. Like AI entities, vehicles can be given paths to follow or attack instructions, but they are obviously inanimate objects by themselves. That means you need to give them drivers before they can do anything. To give the vehicle a driver, place an AI entity by the vehicle, like a Merc_Cover_M4, and give the two objects the same group number. That way the soldier will enter the vehicle when it is activated. As well as drivers you can add gunners and passengers in the same way. For as many places there are on the vehicle, the number of AI entities you place in the same group nearby will enter the vehicle in the various available positions. So, for example, if you place six soldiers near a Humvee and place them all in the same group as the Humvee, they will all enter the vehicle as driver, gunner and passengers.

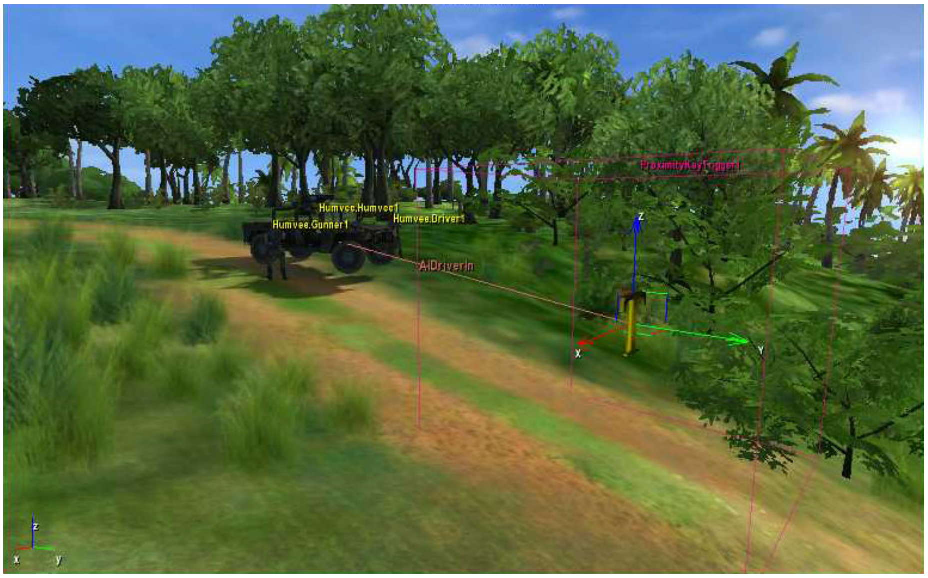


Figure 3.9 Use Pick to link the trigger to the vehicle.

To activate the vehicle, you need to set up a trigger which will send an instruction to the vehicle to start a specific behavior. There are a number of triggers, with two commonly used ones being the Proximity and Area triggers. The proximity trigger can activate the target object whenever a player passes nearby. To program the trigger to do this, first place a Proximity Trigger on the map where you want the trigger to occur. You will find the Proximity Trigger in the Triggers directory of the Entity list in the Objects window. Once the object has been placed, click the On Enter event in the Proximity Trigger tab of the Object window. When you do this, you will see the Pick New button become available. Click this button, and then click the vehicle that you want to trigger in the map work window. Underneath On Enter you will now see the name of the vehicle, followed by the default trigger in brackets. You will likely want a different trigger, so right click on this new entry, and choose the correct trigger, such as GoPath.

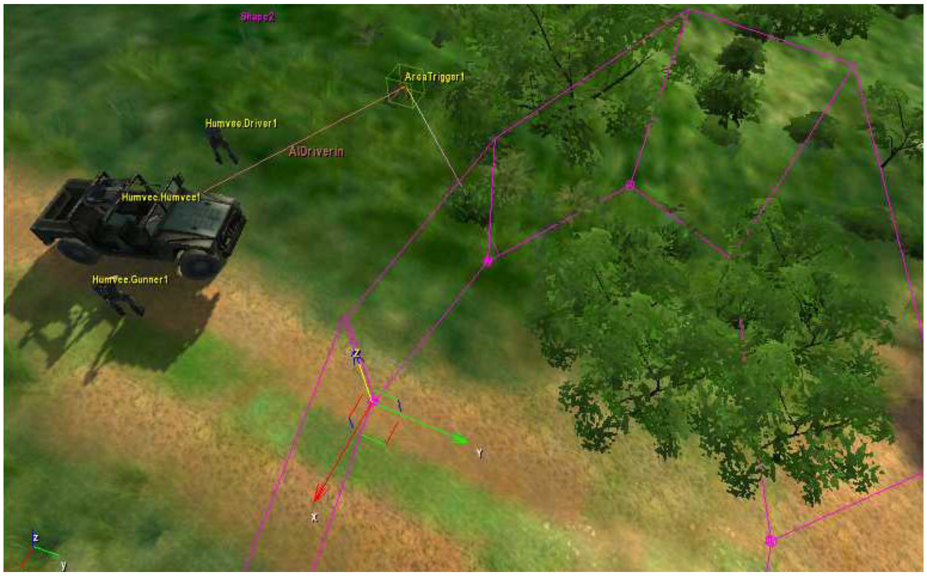
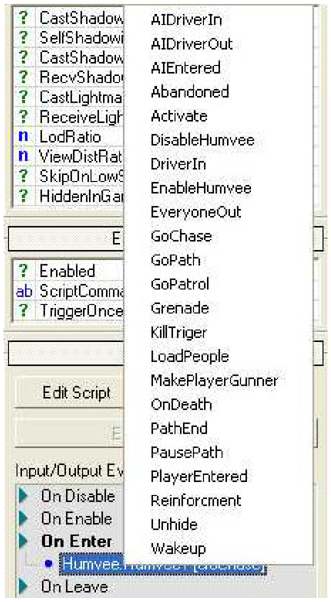


Figure 3.10 Area linked to area trigger linked to vehicle.

Tip: you may want to put the Area Trigger out of the way of the area that triggers it, to make the map clearer and avoid confusion.

The Proximity Trigger has an area the shape of a square or rectangle, which can be set by altering the DimX, DimY and DimZ values in the object's parameters. Note that changing the size of the area with the scaling tool doesn't alter the area which is triggered through proximity, and so you must use the X, Y and Z values. Due to the limitations of this shape, you may sometimes want to use a self-defined area to trigger the response. Instead of a proximity to the trigger activating the vehicle, you can instead have the player's entry into an area of the map activate the trigger using an Area Trigger, and the trigger in turn activating the vehicle. To do this, create an area on the map with the Shape object in the Area list of the Objects window (see Object Placement chapter). Give the Shape object a height, and move it so that it covers the area you want triggered, without a gap. Finally click the Pick button under the list of Target Entities, and click on the Area Trigger that you want to activate. Program the Area trigger in the same way as you did the Proximity Trigger.



One of the simplest triggers you can send to the vehicle is GoChase, which will result in the vehicle giving chase to the player as soon as it enters its view. Make sure to set the vehicle's sight range to a value that will cause it to respond to the player when he passes. The vehicle will give chase until either it is destroyed or the gunner is dead. To get the vehicle to follow a path, you need to give it a GoPath or GoPatrol trigger. These two triggers are very similar, and can be set up in the same way, but differ in the way the vehicle reacts to sighting the player. With GoPath the vehicle will continue to follow the path until the gunner is killed, but with GoPatrol, the occupants of the vehicle will exit the moment they spot the player.

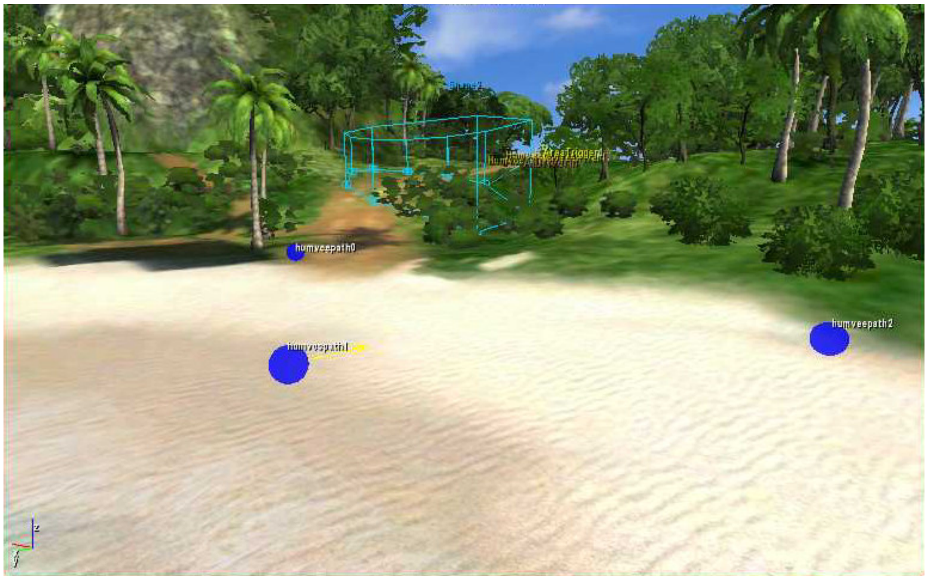


Figure 3.11 You can set up paths for buggies in a similar way to mercenaries.

To set up a path, place tag points on the map like you would with a walking soldier. For vehicles there are no naming conventions to follow, except for making sure there is a number at the end of every tag point name, and that each tag point has the same name. For example, you can have three points on a path called buggypath0, buggypath1 and buggypath2. To get the vehicle to follow this path you need to edit the pathname property in the vehicles property. Change the name to whatever you named the tag points, minus any numbering, for example buggypath. You will also need to tell it which number to start at, and how many steps there are in the path. In the example given the path start would need to be set to 0, and the number of steps set to 3.

One thing you need to be wary of when setting up paths for land vehicles is the sight range of the vehicles and their occupants. As the occupants of the vehicle will exit when the player enters the vehicle's sight range, the sight range of the vehicle and the occupants needs to be the same, otherwise you will get a situation where the occupants will exit the vehicle when the player is far away, and then stand around idle, because they can no longer see him. If the sight range of the vehicle is too far, it can even result in the drivers getting into the vehicle upon being triggered, only to exit again immediately as they "spot" the player in the distance. Usually it is best to reduce the sight range of the vehicle to the same as the soldiers driving, to avoid bizarre behavior.

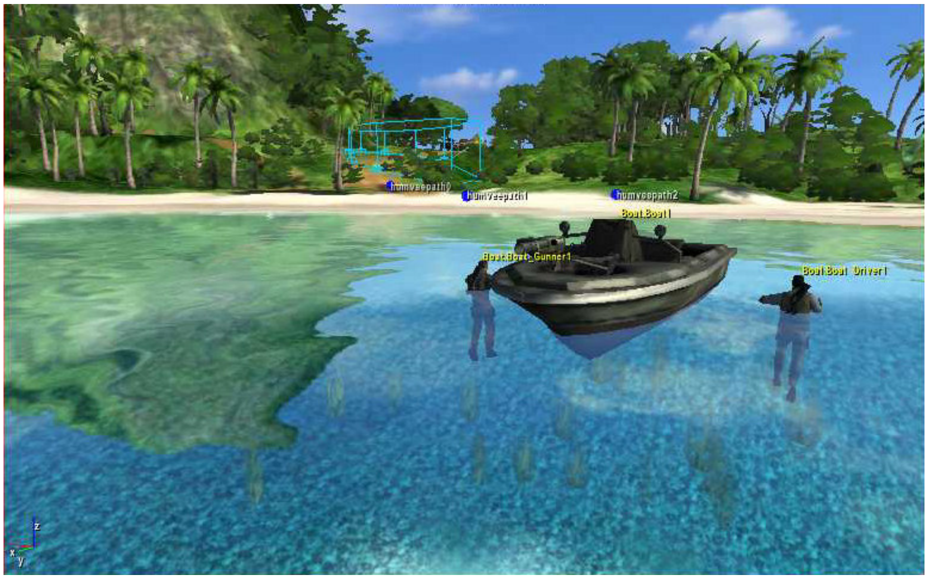


Figure 3.12 You can leave the pilot and gunner for a boat swimming in the water.

Sea vehicles, like boats and zodiacs, can be activated in the same way as land vehicles, with a few considerations for the environment. Firstly the soldiers you group with the boat will often have to be placed in the water, so you will want to make sure that you don't put more soldiers down than can fit in the boat, otherwise they will be left swimming uselessly. You also don't want the swimming soldiers to be discovered by the player without first being activated, else they will likewise be completely helpless. The only major difference to the methods described in the previous section is the way that boats are programmed to attack and give chase. Boats don't have a GoChase trigger, instead they have a GoAttack trigger, but apart from that they are practically identical.

Aircraft

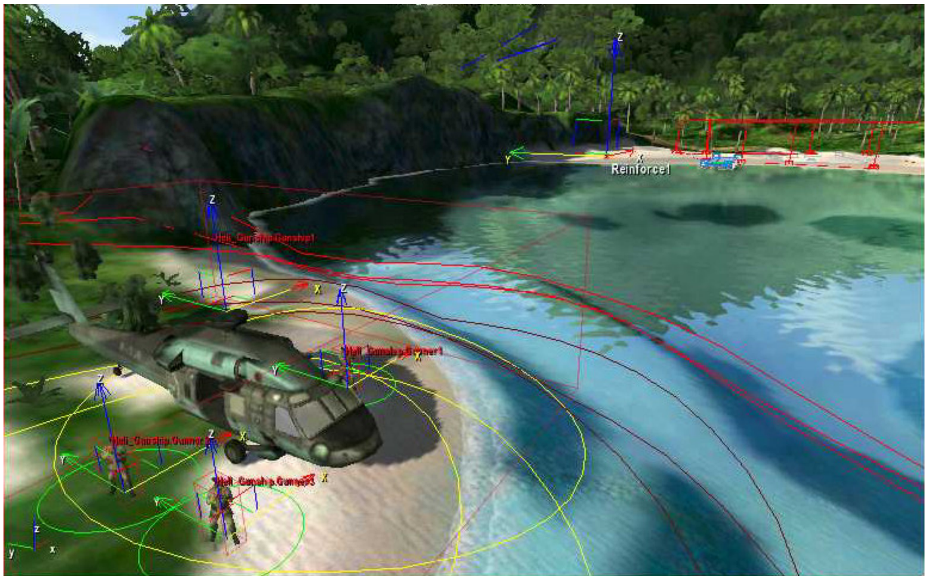
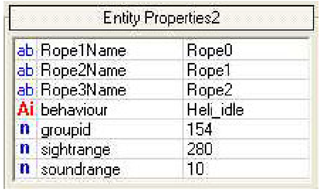


Figure 3.13 Place passengers and gunner, just like you would for a land vehicle.

There is one other AI vehicle type to consider in Far cry, and that is aircraft. There are two key aircraft, and these are the Gunship and the V22. Both work almost exactly the same way, and both can be programmed to work in the same way as both land and sea vehicles, with a few minor behavioral differences. The key role for the Gunship and V22 is in re-inforcement. You can use the Reinforcement event for any passenger carrying vehicle, but the aircraft perform this role the best, as they do not have to worry about difficult terrain, and can reinforce from almost anywhere on the map. The V22 in particular is set up to reinforce in a special way, unlike any other vehicle in the game.



Entity Properties2

ab	Rope1Name	Rope0
ab	Rope2Name	Rope1
ab	Rope3Name	Rope2
Ai	behaviour	Heli_idle
n	groupid	154
n	sightrange	280
n	soundrange	10

In order to get the Gunship to re-inforce, simply set up a trigger, as with other events, using the Reinforce event. Place a Tag Point where you want the re-inforcement to take place, and make sure in the Gunship properties the pointReinforce value is changed to match the name of your re-inforcement tag point. The V22 works in exactly the same way, except that instead of landing troops, it drops them via ropes. These ropes need to be placed on your map, or soldiers will not drop from the vehicle. You can find the ropes in the Entities list of the Objects window in the RollupBar. It doesn't matter where you place the ropes on the map, but you must place three of them, and name them exactly as they appear in the three RopeName properties in the entity's parameters. These default to Rope0, Rope1 and Rope2.

Concluding Words

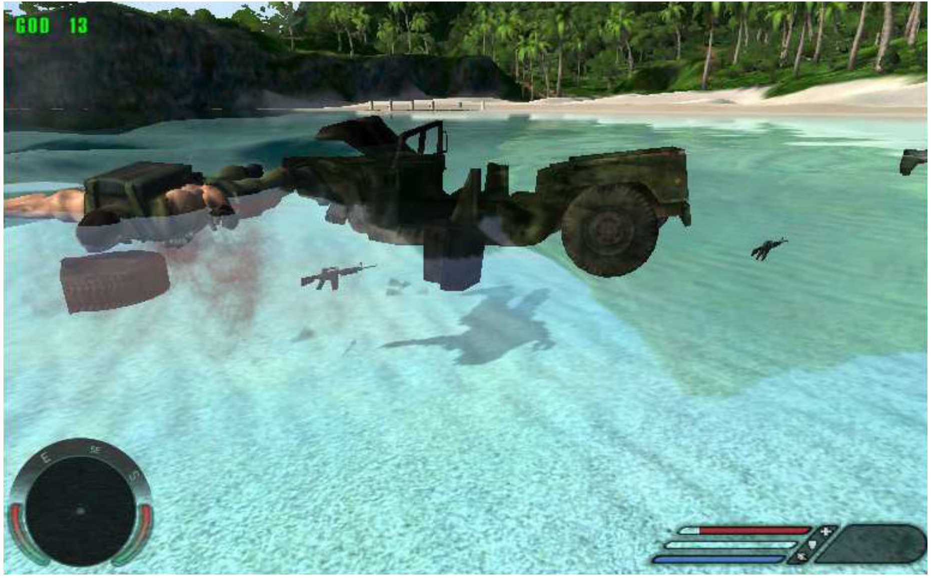


Figure 3.14 Be careful where you tell your vehicles to go.

One final consideration when placing AI vehicles on the map and getting them to behave as you wish, and that is the terrain. All the vehicles operate with different physics, for example the Boat has a small turning circle, and so you can set up path points very close to each other. The Humvee has a much wider turning circle, and so placing tight path points can result in the Humvee missing its target. If you place path points over shallow water, the boat can get stuck, and if you place path points for the Humvee near water it can crash into the sea. Air vehicles have less problems with terrain, but even they can get stuck, so it is necessary to consider the route between them and their target when deciding where to place them on the map.

Animals

In order to populate your world with more realistic creatures than just grunts and mutants, you will want a variety of believable fauna, like pigs and birds. There are animals for land, sea and air, and there are animals that can be placed on the map and work as they are, like pigs, and others that must be triggered, like birds flying from the undergrowth. There are even sharks. All AI wildlife can be shot and killed, and all can be programmed to follow paths and other behaviour that both human and vehicle AI can do.

Pigs

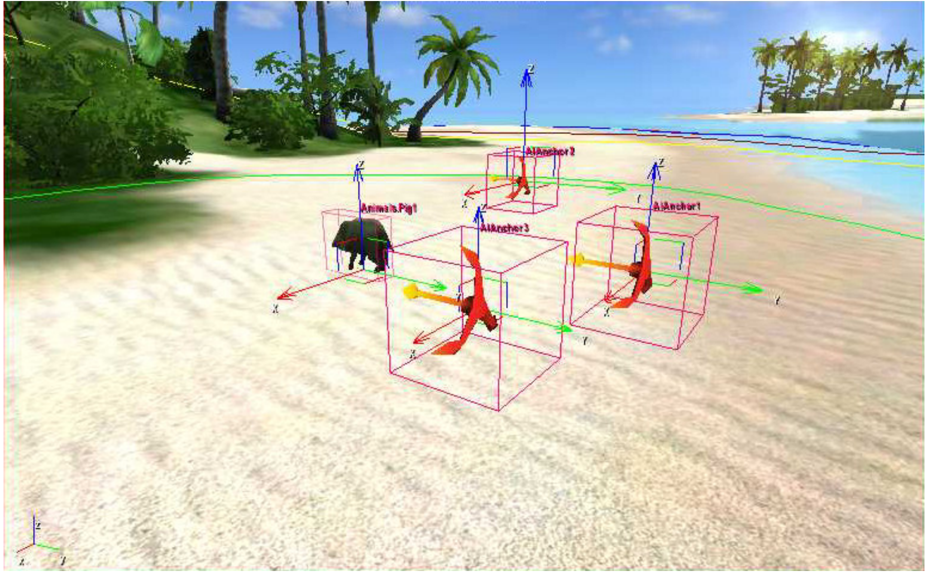


Figure 3.15 Gets pigs sniffing around by giving them investigation points to go to.

The pig is one object that can just be placed on the map, and it will automatically start doing something. That default behaviour is to head towards the player. It will keep coming at the player even if it shoots it. You can change this behaviour to anything you like, although if it hasn't got the animation it might not actually do what you tell it. For example, you can lay down a number of AI Anchor points and set them to INVESTIGATE_HERE, and the pig will walk between the AIs investigating them. You can also give it path and patrol jobs to follow.

BOIDS

TO FOLLOW

Chapter 4

Events

This section describes how to control the actions triggered in many of the objects in Far cry.

Events are the nervous system of a Far cry level. They control everything, and can be strung together to form complex relationships that work like a program or script to pull everything together and bring the map to life. We touched upon events in the previous chapters, but there's far more to events than simply waking up AI entities or opening doors, for they can be used to power an interactive storyline to its conclusion. Events glue the objects together into an interactive whole, rather than as an experience made up of interacting with individually activated components. Events can turn a walk through a tunnel, into an Indiana Jones adventure, ducking booby traps, placing bombs, shooting down chains, and escaping thundering boulders.

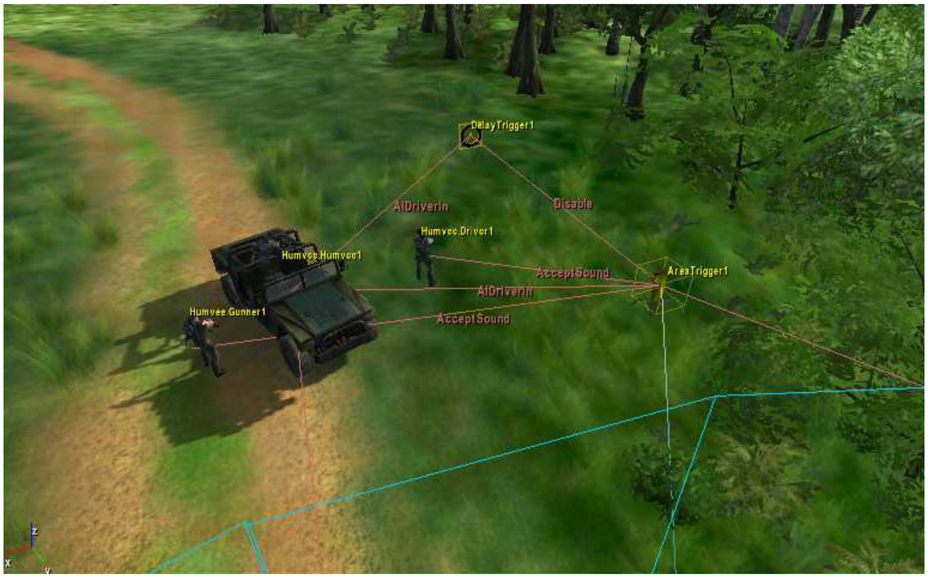
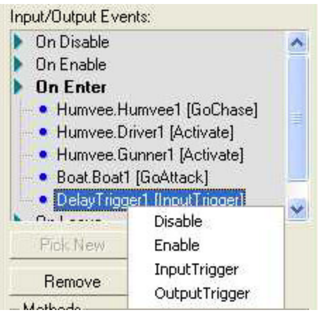


Figure 4.1 Objects can trigger objects that can trigger triggers that can trigger objects....

Events can apply to any kind of entity: archetype entity, AI entity, simple entity, etc., but not to brushes, tag points, areas, etc. Entities can both receive and send event signals, and the signals that they can receive are the same ones as they can send. Therefore, if you need to know what events a particular object can send, you



can look at the object itself and see what kinds of events it can act upon to find the answer. As you will have probably learned from earlier chapters events can be activated by selecting the object that you want to use to trigger the event, and picking the object that you want an event triggered in, using the Pick tool underneath the Input/Output events in the RollupBar. You'll want to select an event signal you want the picked object to act upon, and the event you want triggered. For example, you could select the On Enter event signal from the Proximity Trigger and pick an explodable gas canister to send the Explode event signal to.

Note

Many events can only receive or send signals, while others can both send and receive. For example Die can receive an event that results in a mercenary dying, but On Die will not function to send a signal upon the death of the merc. You will need to use OnDeath for this.

Simple Events

There are a number of simple events common to many objects, that can be readily understood and used. First of all there are the simple enable and disable events, which enable and disable entities until you want them to function. For example, if you have a proximity trigger that explodes a building only upon the player having safely exited it, you would want to have the trigger disabled as the player enters the building, and then enable it with an event signal once the he is inside, so that when he exits the building it will work, but not as he enters it. You may also want to enable and disable objects that aren't in view, because they are a heavy drain on the processor, like dynamic lights. The explode event itself is very simple, and can be used to propagate another event, for example once you have exploded the object in the building, you may want to play a mission hint upon the On Explode event being triggered for the object.

Some other simple events are Hide and Unhide. These events do exactly as they describe, and hide and unhide objects from the player's view. This can be particularly useful when using AI entities, like mercenaries. You may, for example, want to have a scene where the player is exiting from the building that has just blown up, and he is intercepted by some mercenaries. You don't want these mercenaries being there, or being seen by the player, until after he has been in the building, so you may want to have them hidden, and then send the unhide event signal once the explosion in the building has been set off. Mercenaries themselves have a number of simple events, in addition to those mentioned in the AI chapter. You can kill an AI, by issuing a Die event, or act upon that death by sending an event signal to another object when the mercenaries On Die event is triggered. For example, you may want to send an event signal to trigger a mercenary's buddy to say "what the...?" when he sees the mercenary die.

Triggers

The more complex mechanics behind events that can allow you to run them more like a script or programming language, than a series of interconnected actions, are the triggers. Triggers operate independently of objects, and can fire an event without any object being affected in any way. Objects can be triggered by the fact that a player enters a particular area, because a player is exiting the trigger, or simply because another trigger has triggered the trigger. Triggers can also trigger themselves, or have another entity trigger them, such as an mercenary walking into an AI Trigger, or an explosion sending an Input Trigger to a Delay Trigger.

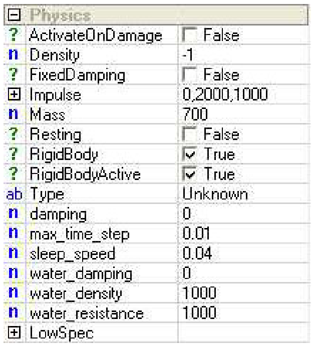
The Multiple trigger allows you to delay the triggering of an event until the trigger has received the event as many times as defined in its parameters. For example, you may have several tasks you want the player to complete, each one sending an event to the same Multiple trigger, and when each completed event has been triggered, the multiple trigger can then open the door to the next level. Like the Delay Trigger, the Multiple Trigger is set off with the Input Trigger. However, you must send the signal for the next event On InputTrigger, as no additional signal is generated. The number of times the Multiple Trigger needs to events sent to it is defined by the NumInputs value. Once the Multiple Trigger has counted that many Input Triggers, it will shut down and no longer be active.

More Complex Events

Some of the events triggered in the game take a little more setting up than the ones discussed earlier in this section, like AddImpulse. Some objects have events that need to be chained together in order to get the whole system to work properly, like the Elevator. Some objects, like the elevator, have events that aren't immediately apparent in their use. We will discuss a few of the more difficult, and commonly used, events like these here, but for the rest you will have to work them out for yourself, by looking at the already coded game levels for inspiration, or through trial and error. With the events themselves being so flexible, you will likely invent your own ways of doing things anyway.

Adding Impulse

You can easily add impulse to many objects, simply by sending the AddImpulse event to those objects that will accept it. In most cases, however, this will simply result in the object wobbling slightly, or not even moving at all. That is because the default impulse for objects is 1, 2, 3 in the X, Y and Z axes. In order to get the AddImpulse to work more flexibly, and to get objects to catapult themselves over great distances, you will need to use specific objects and set them up in a specific way. To create your own impulse parameters, you won't be able to use pre-made entities like Archetypes, you will have to choose a simple entity and code the physics yourself.



⊟	Physics	
?	ActivateOnDamage	□ False
n	Density	-1
?	FixedDamping	□ False
⊞	Impulse	0,2000,1000
n	Mass	700
?	Resting	□ False
?	RigidBody	☑ True
?	RigidBodyActive	☑ True
ab	Type	Unknown
n	damping	0
n	max_time_step	0.01
n	sleep_speed	0.04
n	water_damping	0
n	water_density	1000
n	water_resistance	1000
⊞	LowSpec	

First place the simple entity on the map that you want to launch, say a locker from the indoor/furniture/cabinets folder. Then change the physics properties to turn it into an object that can be pushed around the map. In the object's parameters, first check the RigidBody box, to give the object a physical presence, then give the object a mass that reflects what it is. The default weight is 700, which is far too much for something like a coconut, which is better off with a mass of about 2. Then, in order to get it to move somewhere you you add impulse, you will need to change the X, Y and Z axes of the Impulse parameter. Adding impulse on the Z axis will push the object up in the air, and the X and Y axis will give it forward momentum. All you need to do then is connect it up to something that can send it an AddImpulse event, like a Proximity Trigger, and you can test launch the object.

Note

There is a specialised trigger for launching boats, called the BoatTrampolineTrigger.

Tip: if you move an elevator, you will need to reload the script, or it will think it is still in the last area it was located. You will also need to do this when first placing it, as the last area a new elevator object was located was 0, 0, 0.

Elevators

PLACEHOLDER DIAGRAM: elevator movement (open, opened, etc.). (4.2)

The elevators events need a little explaining, as they don't seem altogether clear at first viewing. The most confusing events are Close, Closed, Open, and Opened, as these appear to be pretty much the same thing. To understand these events better, take a look at the diagram below. The default position for the elevator is stationary, at its starting point. When the player sets the elevator in motion, the Open event is triggered, and hence to get the elevator moving you must trigger it with an Open signal. Once the elevator reaches its secondary point, which is above or below the starting point depending on how you set the lift up, it sends the Opened signal. Once it has reached its destination, it will eventually return to its starting point. When it starts this return journey, it will send the Close signal. Again, you can get the elevator to start this journey by sending the Close signal to it. Once in the elevator finally returns to its starting point it will send the Closed event signal.

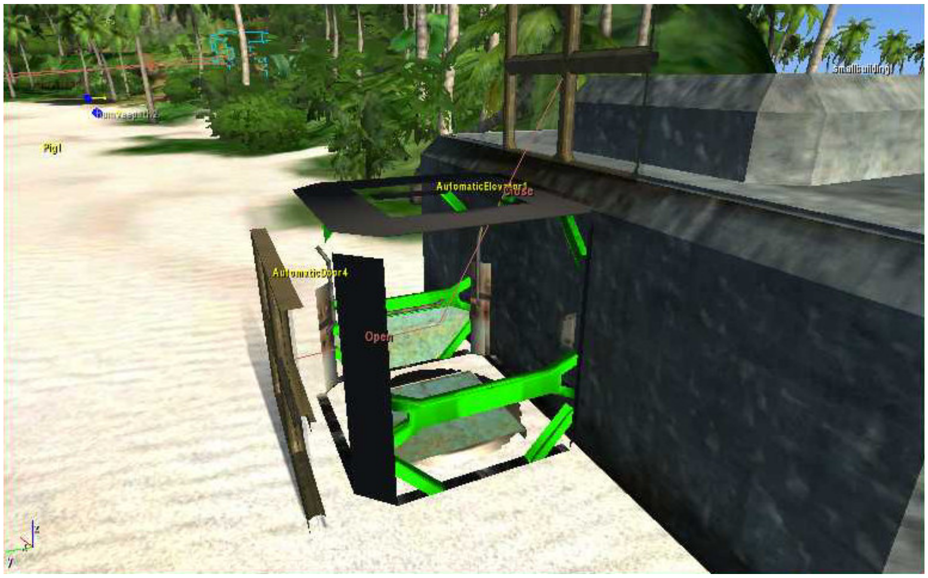


Figure 4.2 Link doors to elevators to cause them to move up or down.

The Open and Close signals are enough to get a basic elevator moving backwards and forwards, but a real elevator is going to need more than that. Real elevators have buttons that you need to press in order to get them started. Real elevators won't start moving until the door has closed completely. You will need events to make an elevator work like a real one. For example, you may want to set the doors so they send Open and Close events to the elevator when they close, so that the elevator will only move in response to a door closing, rather than having it respond automatically. You can also put call buttons on each floor that will send the same events to the elevator, to cause it to move to the floor the player is on. Even more can be done for the elevator, including changing its materials depending on its direction, changing the light, and allowing the elevator to be accessed with a keycard.

Note

The AutomaticElevator is set to automatically move to its next point by default, the moment a player enters. You will want to turn off the automatic option in the parameters if you want to control it with events.

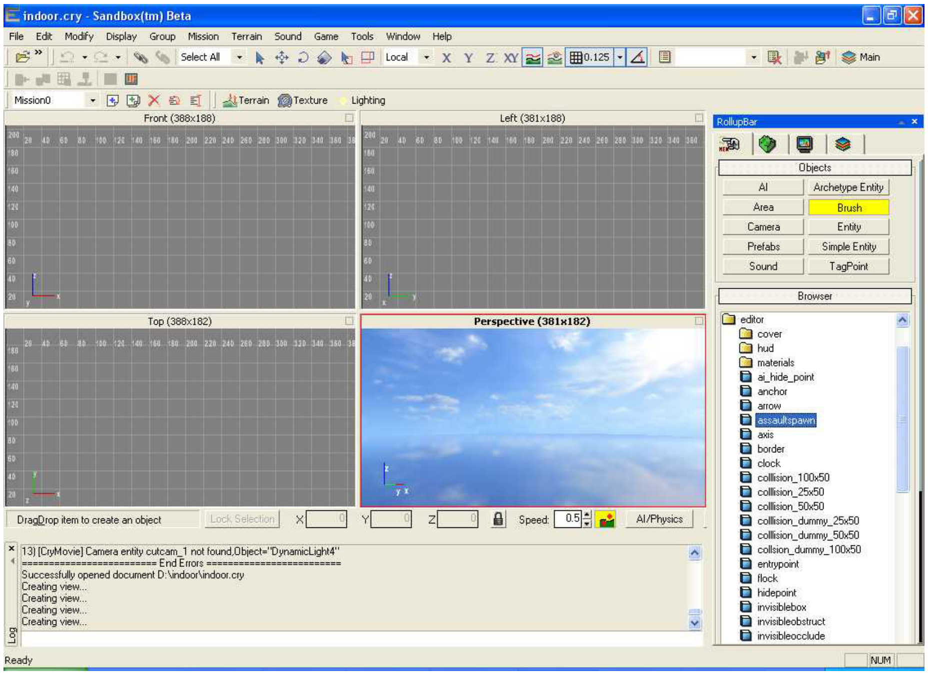
Chapter

5

Internal Areas

Not all levels are set out in the open, so you will need to learn how to create levels inside buildings and underground.

Although Far cry has been designed to create impressive outdoor environments, no game would be complete without the ability to create indoor areas. The actual creation of the indoor scenes is as simple as building the rooms with wall, floor, ceiling and door objects. However, there are some technical aspects involved in making sure that a player, as well as AIs, can see inside these areas, as well as special considerations for lighting, particularly in terms of how it affects the enclosed areas and the portals that allow you to see into these enclosed spaces. Some consider interior level design to be almost completely separate from outdoor environments.



```
13) [CryMovie] Camera entity cutcam_1 not found,Object="DynamicLight4"
======================== End Errors ========================
Successfully opened document D:\indoor\indoor.cry
Creating view...
Creating view...
Creating view...
Creating view...
```

Figure 5.1 A four window layout will make it easier to see what you are placing.

When laying the foundations for your indoor level, you will be best served by changing from the usual viewing layout that you use for outdoor environments. To better see what you are placing, switch to the four square layout from the

Layout Configuration option in the Display menu. This will allow you to view the actual design, as it will appear in the game, along with top down, left side and frontal views, to help you better see the positioning of floors/ceilings, walls and doors/portals respectively. Either work with all four windows at the same time, or click on one to concentrate on that window, and give yourself more space to work.

Floors and Ceilings

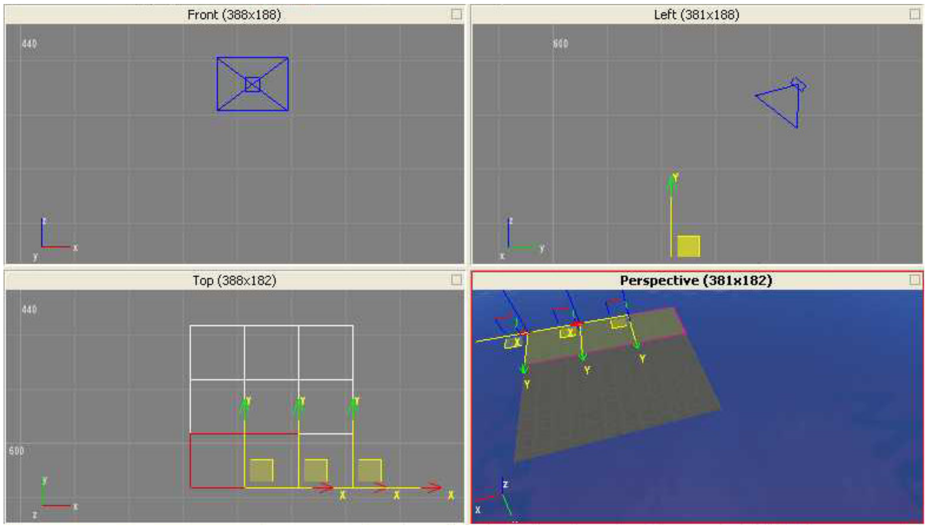


Figure 5.2 Clone objects to place them more quickly.

Tip: change the grid to the largest size possible when laying floor objects, for ease of placement.

To place your floors, grab a floor object from the Brush object list. You will find a lot of useful floor objects, as well as doors, etc. in the glm directory, for example the floor4x4y from the ww2_indust_set1 folder. Place the floor objects to your liking, making use of the Clone object function (Control-C) to build floor areas quickly. Once you have completed the basic floor layout, you may want to freeze the floor objects, so that you don't accidentally move them about as you build the rest of the level. You may also find it easier, with the large number of objects needed for indoor levels, to turn off the names displayed for each object. You can do this by going to Preferences from the Tools menu, and disabling Text Labels in the Viewports folder. Ceilings can be placed in the exact same way, but you may want to leave these until last.

VisAreas and Portals

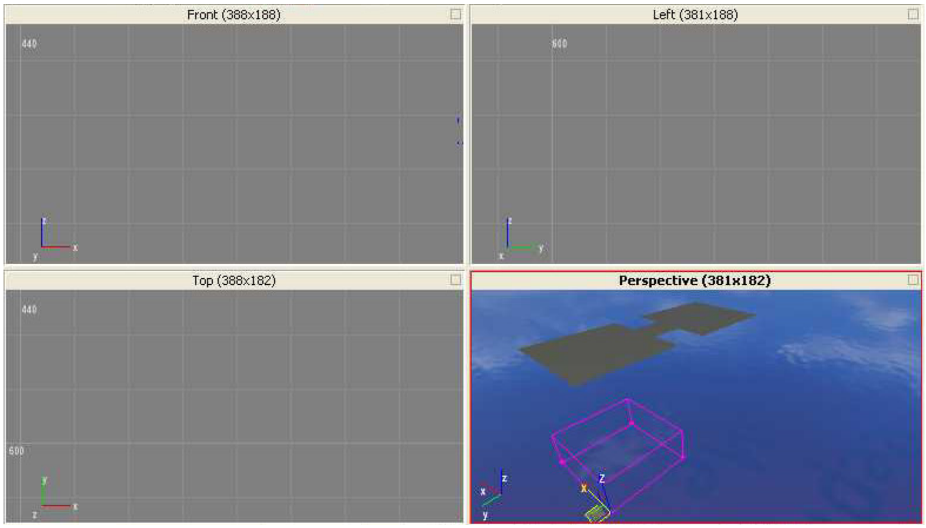


Figure 5.3 Check Portals and VisAreas after placement to make sure they are at the correct height.

Tip: while VisAreas should be placed with the largest grid, Portals are best placed with the smallest.

VisAreas define areas of visibility, but also occlude areas from being viewed from outside. Portals allow those outside VisAreas to view inside, and those inside the VisAreas to see out of the one VisArea and into another. You can work without VisAreas and Portals, but you risk problems with light seepage, and other serious issues, if you neglect them. They are considered best practice, and avoid problems later on in your interior design. To create your VisArea, place the VisArea as a shape object, clicking each corner until they join up. It is vital that you push the VisArea to the very edges of the floor, so use a zoomed in top view to ensure this. Once you have created the VisArea, you will need to check its position and height. Make sure that it is placed just underneath the floor and covering the maximum height of your building. Remember that any object, like floor tile or wall block, will not show if its centre is outside the VisArea.

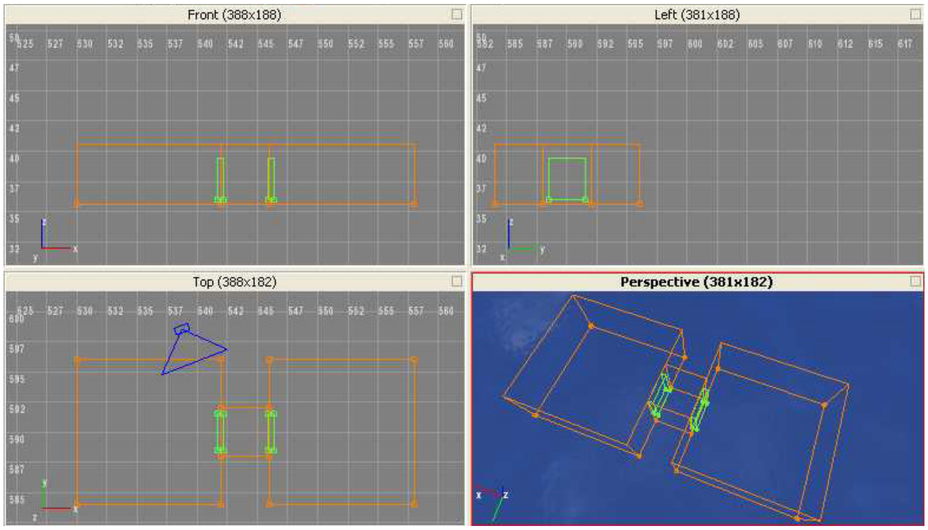


Figure 5.4 Vis Areas, in orange, with connecting Portals, in green.

Portal placement needs to be very exact, and even after placing the object, you will probably have to alter its size and height a few times to ensure that it doesn't occlude any object that you place nearby. Place the points of the Portal between the two VisAreas, like you placed the points of the VisArea itself. Keep the Portal as narrow as possible, and push its edges as close to the VisArea as possible. While the Portal must traverse both VisAreas, you musn't have the edges of the Portal touching the edges of the VisArea. Use the provided image as a guide. Once you have placed the base of the Portal, using the top view and smallest grid, you will need to check the height to make sure that it covers as much of the gap between the rooms as possible, without touching any edges. Move the Portal if necessary.

PLACEHOLDER DIAGRAM: VisArea occluding object when portal outside player FOV (5.1)

The interaction of objects and portals can be problematic. If you have an object that traverses two VisAreas, through a portal, the player may not be able to see it. If the centre of the portal is inside the VisArea that the portal links the player too, as in Diagram 5.1 above, then the player will not be able to see the object, even if it is inside his field of view, if the portal moves outside his field of view. To fix this, any object that traverses two VisAreas, must have the centre of its bounding box inside the Portal. This means that great care must be taken when placing objects around portals, especially objects like walls that are placed close up to the edges of VisAreas. Try to ensure that wall objects finish at the edge of one VisArea and that the next one starts in the next VisArea. This will help obviate significant problems later on in your design.

Walls

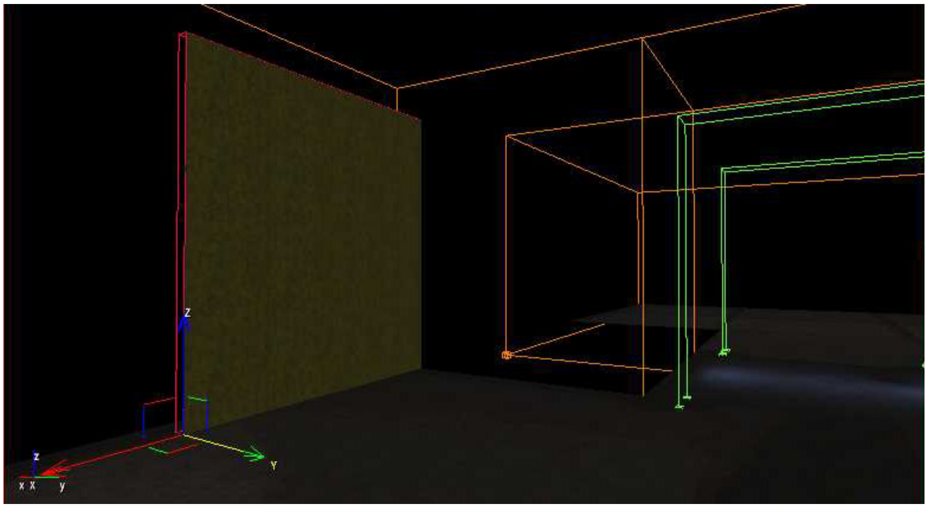


Figure 5.5 Use the VisArea and portals to guide your wall placement.

It makes life easier when placing walls to choose ones of the same size as the floor objects you placed earlier, i.e. 4x4 floor objects and 300x400 wall objects. Walls need to be comfortably within your VisArea, and so you will want to place them about half a meter in from the edge of your floor. Like with the placement of floors, you will find it easier to clone blocks of walls, rather than place lots of individual pieces. If you work within the VisAreas, you will want to switch the Brush Selector so that it selects only Brushes, otherwise you risk moving the VisAreas and Portals about. If you find it too dark inside the building to see what you are placing, then increase gamma by entering the console command r_gamma 2 into the log command line.

Doors

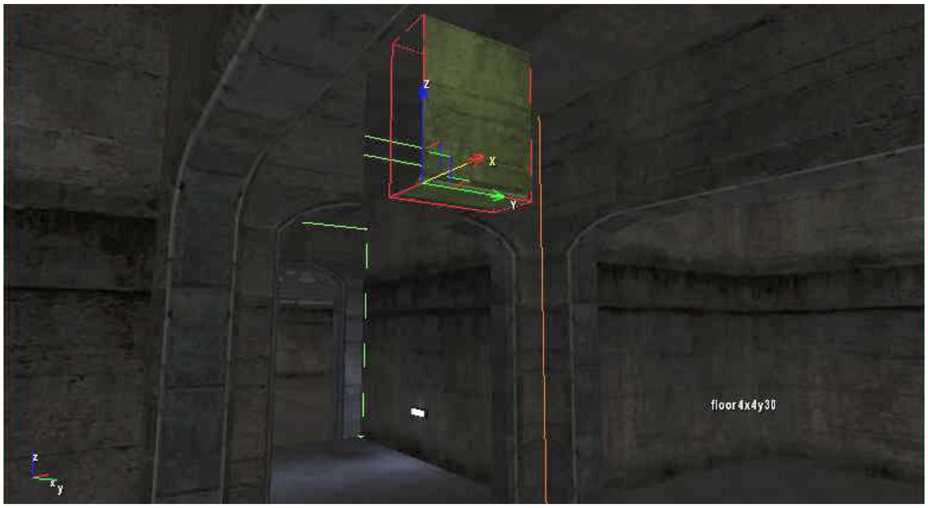


Figure 5.6 Use door frame pieces to cover the gaps around doors.

Doors are probably the trickiest element to place, as you are unlikely to find a door that perfectly fits the gap you have left between the walls. Instead you will have to fit pieces together from the individual pieces that you can find in the directories. You will find the industrial ones are easy to work with, and you will find pieces named _sidestraight, _topstraight, _corner, etc. in the same folder as the walls and floors. If you find you have any gaps between the door and the walls, you should have enough play from the gap you left between the wall and the outside to manoeuver the objects into a perfect fit.

Once you have fitted the door, or jigsaw door pieces, into the gap, you will need to check that the portal is the correct size to accommodate the new doorway. If the portal is too big, then it will waste processor resources, and if it is too small, you will find that parts of the door will disappear. Again, like the VisArea, you must make sure that the centre of any object is within the portal for it to be seen. You can test whether the door object will be seen or not by de-selecting the portal and viewing the door objects that you have placed. Change the height, or even the position, of the portal until everything shows correctly.

Note

If the middle point of a door bounding box is inside the portal it will actually disable the portal while it is closed.

Lighting

Tip: it is a good idea to place the basic object layout in your room before testing the lights, so you get an idea of how it looks when finished.

For indoor areas lighting is vitally important, and there are a number of light types that you will want to consider when designing your indoor levels. You can see a selection of settings for common types of these lights in Appendix ************.

1. Dynamic Lights
2. Radiosity Lights
3. Dynamic Fake Lights
4. Non-Dynamic Fake Lights

Dynamic Lights

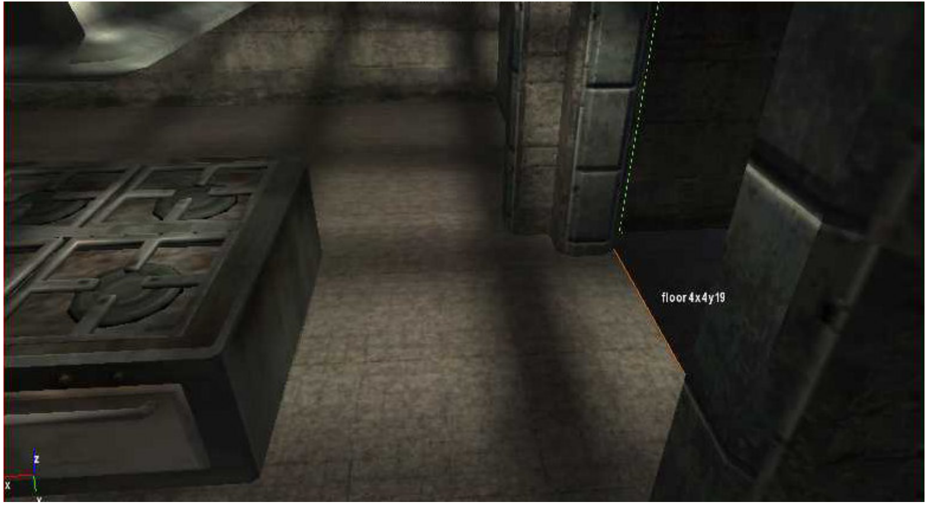


Figure 5.7 Note where the light ends. Dynamic Light won't normally pass through portals.

One key aspect of Dynamic Lights to consider when designing interiors is that the light from these objects won't normally pass through portals, which can lead to very unrealistic looking rooms. You can set the Affect This Area Only parameter to false, but this is bad practice, as it is costly in terms of processor overhead, and can result in light seeping into the wrong areas. Better than this is to be careful in your placement of dynamic lights, making sure that they either don't shine into portals, or only a small amount that can be blurred by floor textures or light maps.

Radiosity Lights

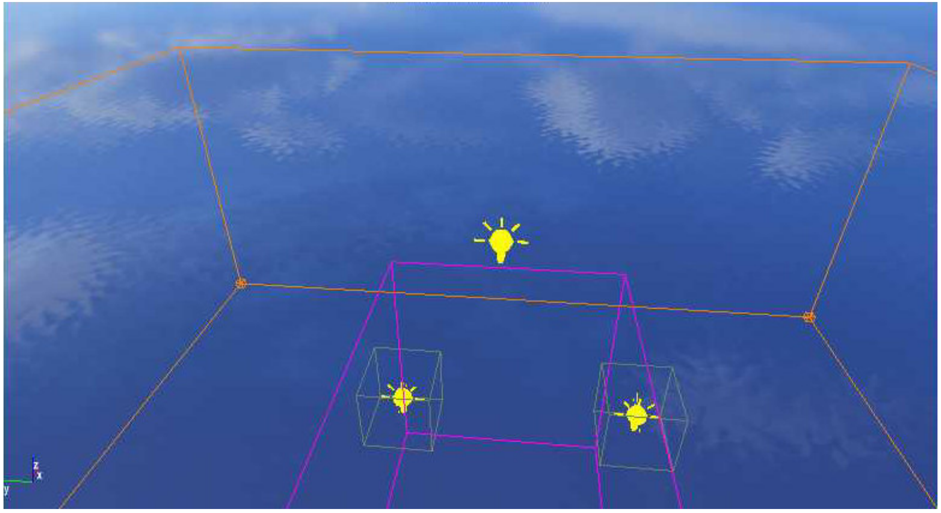


Figure 5.8 Place radiosity lights on the ground to fill the light volume gaps left by dynamic lights.

You will find that dynamic lights can fail to light up certain areas, due to the way in which they work, and so you will need radiosity lights to cast light maps onto the areas that dynamic lights cannot reach. Usually you will place the dynamic lights on the ceiling, and the radiosity lights on the floor, but it will depend on the room itself. You will often need several radiosity lights to cast light around the side of large objects and into corridors. See the example in the picture above, for an idea as to how these lights should be placed.

Radiosity Settings for Dynamic Light Object:

- Fake Radiosity: True
- Light Type: 1
- Use in Real Time: False
- Low Diffuse Multiplier : 0.1 (or any similarly low number)
- Cast Light Map: True

To see the effect of any radiosity lights you place on your map, you will need to generate light maps, by using the Generate Lightmaps function in the Game menu. See the Map Creation section for more details on this function.

Fake Lights

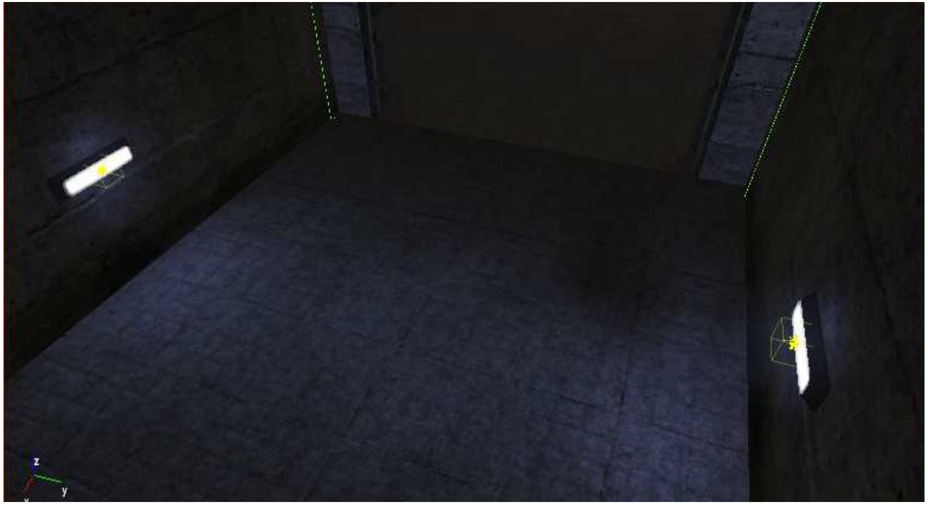


Figure 5.9 Place fake lights in corridors to help blend light between VisAreas.

Tip: decrease the size of the Helper parameter for the object, when dealing with small objects that get obscured by the objects editor frame.

You can also create "fake lights" with the Dynamic Light object. Fake lights cast a shader, such as a beam of light, but don't create light which is used in the more complex calculations which result in the rooms ambient and refractive light. Because of this they use less processing power than full dynamic lights, but can create a similar effect.

Note

You can read more information on lights, and see some example settings for their creation, in appendix *********.

Creating Holes in the Terrain

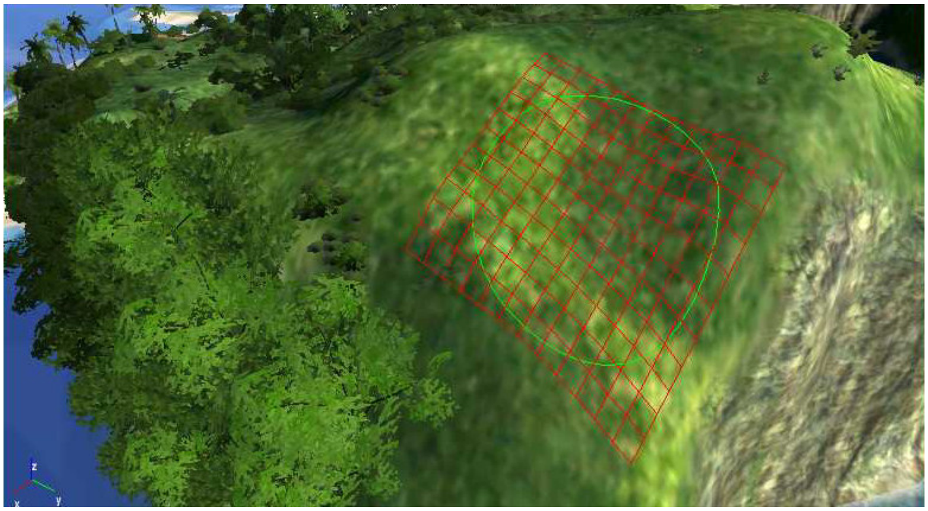


Figure 5.10 With the Hole Brush, the red square denotes the area that will be cut from the terrain.

In order for you to be able to interface the interior designs with the outside world, you will need to place your internal objects inside the terrain and link them with the external objects through a terrain hole. To create a hole you will need to use the Hole Brush from the Terrain tab on the RollupBar. From there you will see that you have a brush, like when you are editing the terrain, only this time there are four red squares under the circle. This shows you the size and shape of the hole you will cut, because the hole is square, and not round like the brush. You can change the size of your brush, the same way as you change the size of your terrain brush.

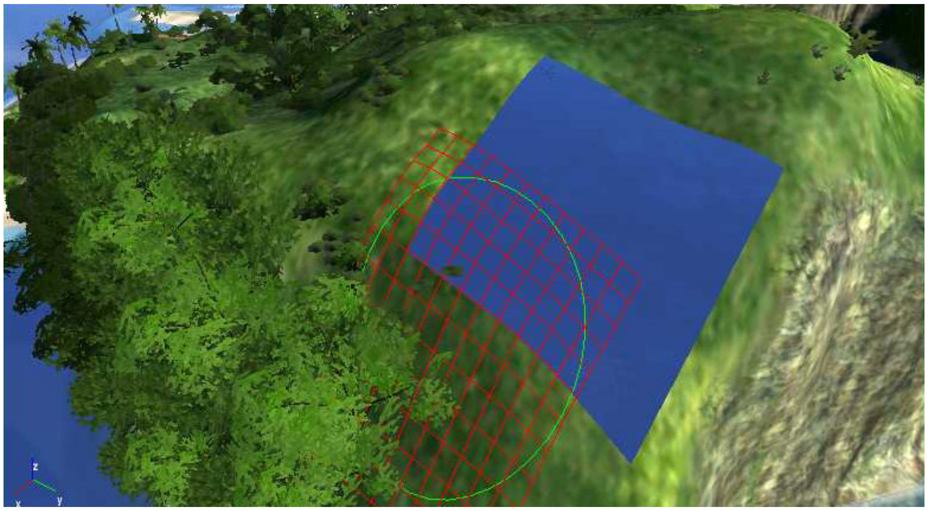


Figure 5.11 Cut holes can be replaced with the Remove Hole brush.

To cut a hole, simply make sure that the brush selected is Make Hole, and click anywhere on the terrain. You can similarly patch holes by selecting the Remove Hole brush, and clicking in the same place. You can switch quickly from one mode to the other by holding the Control key down. Cutting a hole allows the AI and players to pass through, but the polygons that created the cut terrain are still there, they just aren't drawn, or calculated during collision detection. To interface the cut hole with your interior level, you must place the entrance to the internal building next to the hole, and use a portal to allow the players and AI to see into the VisArea of the interior.

Chapter

6

Multiplayer Maps

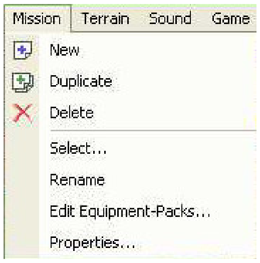
With all the knowledge you have gained up to this point, you will no doubt be itching to create your first level. The easiest place to start is with a multiplayer map.

Tip: if you want to fully test the multiplayer aspects of your game, like the Assault checkpoints, then you will have to export the level to the game and set up a multiplayer game to run it in the actual game itself.

Multiplayer maps are a lot easier to set up than single player, as they don't need AI, save points, cut-scenes, and are in general a whole lot simpler to implement. Of course, the fact that multiplayer maps don't need the likes of AI mercenaries doesn't restrict you from including these in your creations if you want. Multiplayer maps are standalone levels that run over a network, and come in three flavours: Free For All (Deathmatch), Team Deathmatch and Assault. Free For All and Team Deathmatch are essentially the same from a design point of view, and can happily run on the same maps, but Assault is a brand new style of multiplayer gaming unique to Far cry, and requires a little extra complexity in the design. Both, however, are fairly straight-forward, and if you have a map already designed, it should only take a few minutes to implement multiplayer, although it will obviously take you a lot longer to perfect it.

Free For All and Team Deathmatch

The first thing you will need to do when creating any multiplayer map is to create a multiplayer "mission". You can do this by creating a new mission, and calling it FFA or TDM. Being as you can play these two game types on the same map, you may as well create a mission for both. The best way to do this is to create your FFA map, and when you have completed it, save it, and then duplicate it with the Mission | Duplicate function, giving it the name TDM. If you forget to save before duplicating, you will end up with a copy of what the map looked like when you last loaded it, which will likely not be what you want. In that case you will need to delete it, using Mission | Delete, and try again.



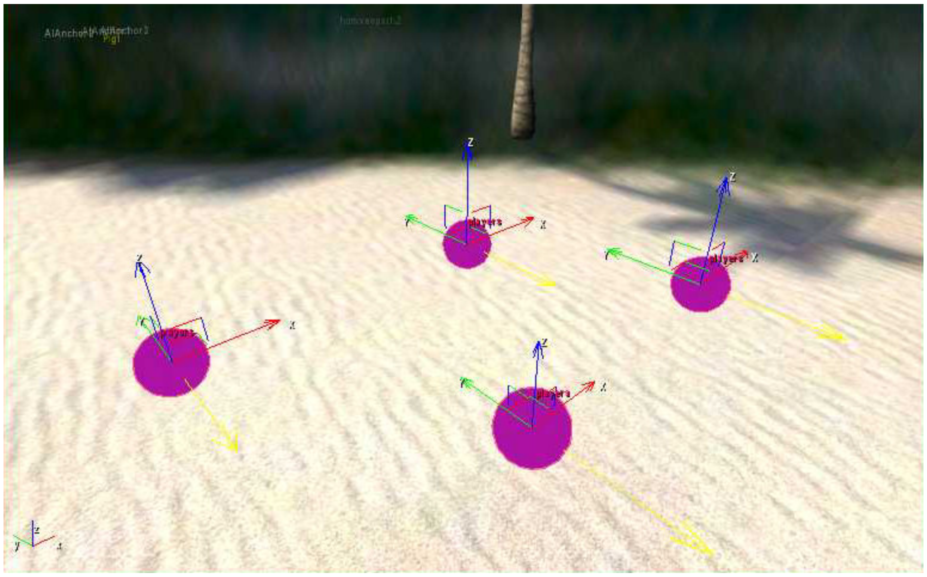


Figure 6.1 The yellow arrow on spawn points shows the direction the spawning player will be facing.

To give your players a way of joining the map, you will need to create a number of spawn points. You will need as many spawn points equivalent to the maximum number of players that can join your map, otherwise you might get players spawning on top of each other. To create a spawn point on the map, take a ReSpawn object from the TagPoint list in the Objects tab of the RollupBar. You will need to name each one of them "players" - no number is necessary. In placing them you will want to consider a few things, such as height and location. You won't want the spawn point too far off the floor or the spawned player will die, you also don't want to lump all the spawn points together, otherwise players will be spawning into a bloodbath. Well, that is unless you want that for some reason.

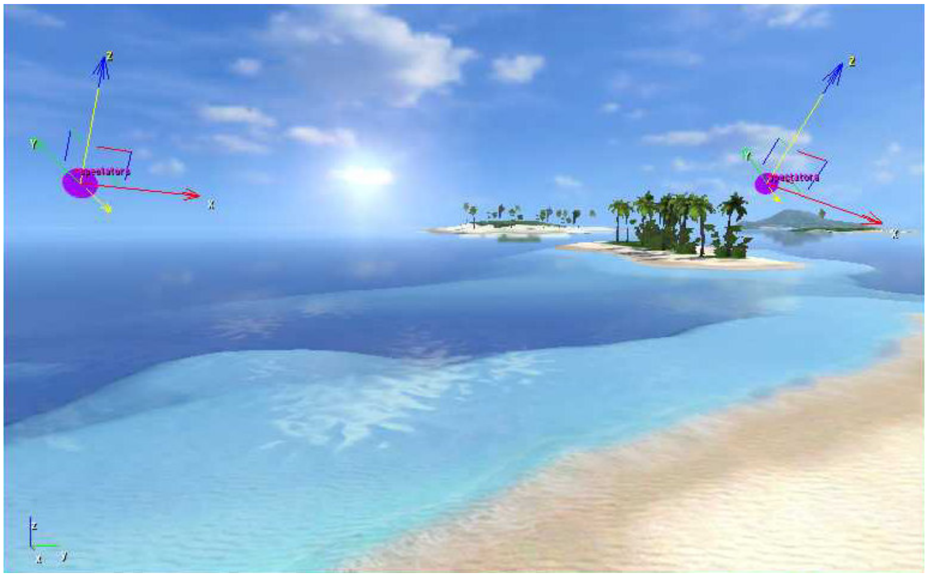
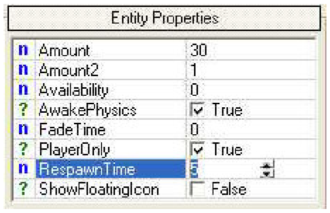


Figure 6.2 Spectator will join at your spawn points at random.

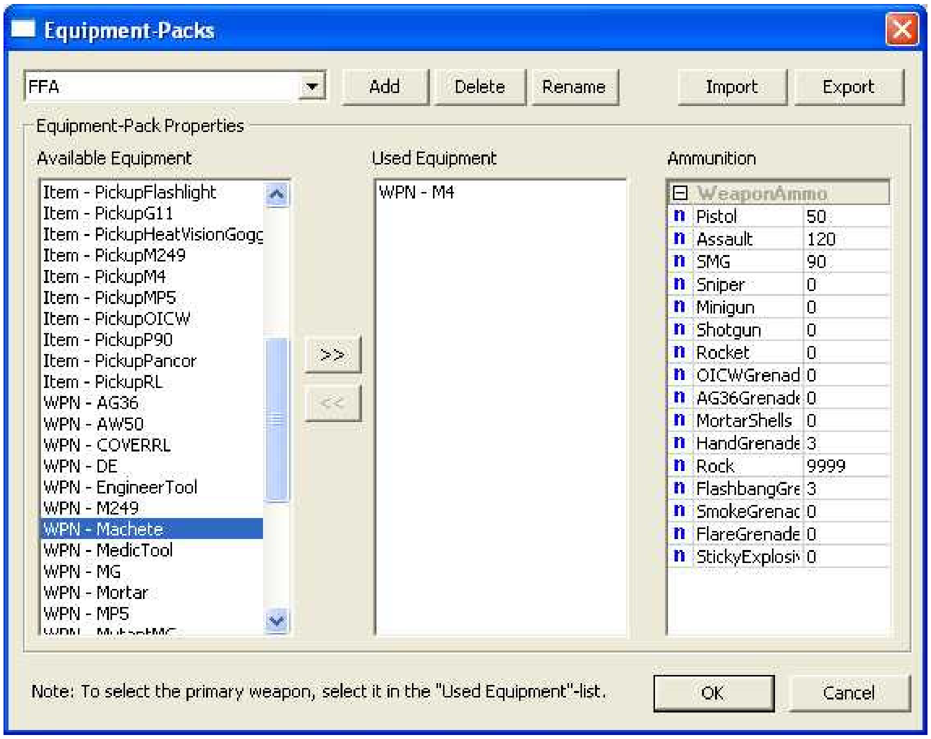
In addition to spawn points for the players, you will also want to add spectator points so that players can view the action. You can place spectator points in the same way as spawn points, using the same object. Name the spectator spawn point "spectators". Players will spawn at a spectator point at random; the same as ordinary spawn points. You can control the direction that players face when they spawn, at either type of spawn point, by altering the direction of the yellow arrow on the gizmo. To change the direction a player will face on spawning select the rotate pointer, and rotate the object around its X, Y and Z axes until a suitable direction is found.



Entity Properties

n	Amount	30
n	Amount2	1
n	Availability	0
?	AwakePhysics	☑ True
n	FadeTime	0
?	PlayerOnly	☑ True
n	RespawnTime	5
?	ShowFloatingIcon	☐ False

The map will be no fun with just spawn points, and so you will need to add some weapons so that players can kill and maim each other. There are two ways of giving players weapons, either as pick-ups on the map, or in their starting inventory. Pick-ups are placed just like any other object, and should be placed liberally around the map, usually near spawn points, as you don't generally want the player running around for ever searching for a decent weapon. You will likely want to place pick-ups for weapons, ammo, health and armour around the map, as well as special items like binoculars or cryvision goggles. Consider carefully the items you place, for example you might not want to place cryvision goggles on sunny maps and sniper rifles where there is nowhere to snipe from. Finally, make sure that all your pick-ups can respawn by setting the RespawnTime parameter to a value other than 0.



Equipment-Packs

FFA

Add | Delete | Rename | Import | Export

Equipment-Pack Properties

Available Equipment

Item - PickupFlashlight
Item - PickupG11
Item - PickupHeatVisionGogg
Item - PickupM249
Item - PickupM4
Item - PickupMP5
Item - PickupOICW
Item - PickupP90
Item - PickupPancor
Item - PickupRL
WPN - AG36
WPN - AW50
WPN - COVERRL
WPN - DE
WPN - EngineerTool
WPN - M249
WPN - Machete
WPN - MedicTool
WPN - MG
WPN - Mortar
WPN - MP5

Used Equipment

WPN - M4

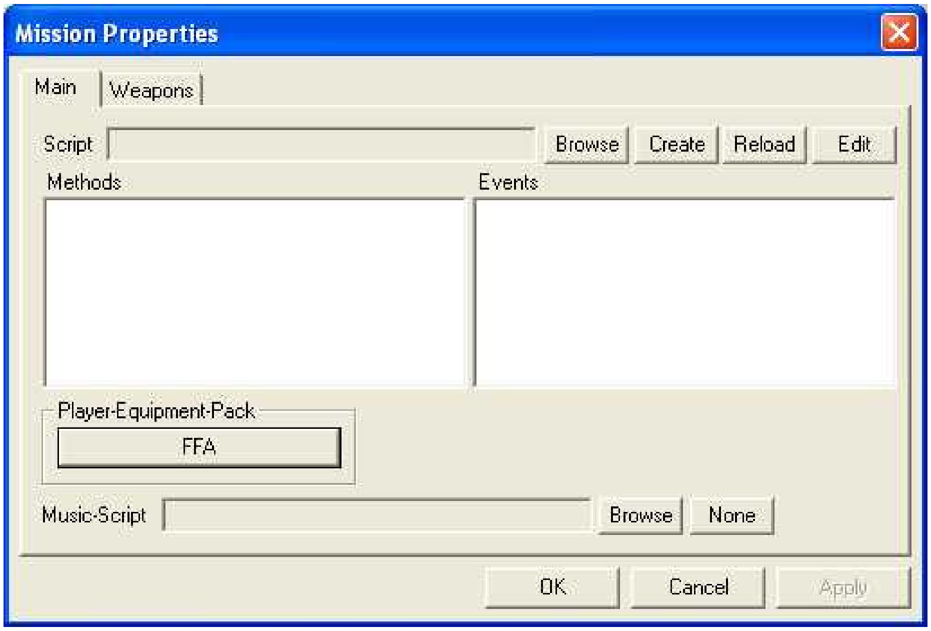
Ammunition

WeaponAmmo		
n	Pistol	50
n	Assault	120
n	SMG	90
n	Sniper	0
n	Minigun	0
n	Shotgun	0
n	Rocket	0
n	OICWGrenad	0
n	AG36Grenad	0
n	MortarShells	0
n	HandGrenade	3
n	Rock	9999
n	FlashbangGre	3
n	SmokeGrenac	0
n	FlareGrenade	0
n	StickyExplosi	0

Note: To select the primary weapon, select it in the "Used Equipment"-list.

Figure 6.3 Set up equipment packs to give your players weapons to start the game with.

You may also want your players to start with a weapon, possibly something basic like a machete. In order to do this you will need to edit the Equipment Packs for the mission by selecting Mission | Edit Equipment-Packs. From this window click Add, and give your pack a name, like FFA, before adding whatever weapons you want the players to start with. When finished click OK, and then select Mission Properties from the same menu. Click on Player-Equipment pack and then select the pack you just created, say FFA, from the pull down list at the top and press OK again. Your players should now start with all the weapons you included in the equipment-pack.



Player-Equipment-Pack
FFA

Figure 6.4 Pick the right equipment pack for your map in the Mission Properties window.

To add even more fun and dynamism to your maps, you may want vehicles on your map, in addition to weapons and other pick-ups. Weapons, like pick-ups, can simply be drag-and-dropped onto your map. Like weapons, vehicles usually need to respawn, after they have been destroyed. Vehicles, however, have a slightly more complicated respawn procedure to weapons. In order to make a vehicle respawn you will need to make use of the Phoenix object that can be found in the Multiplayer directory in the Entity objects list. Place both the vehicle and the Phoenix object on the map, and then create an event link from the vehicle to the Phoenix object, so that the Phoenix object is reset when the vehicle's OnDeath event is triggered. Do this by clicking on the vehicle, scrolling down to the Input/Output Events list in the object's properties and selecting On OnDeath. Then click pick, and select the Phoenix object. Make sure to set a RespawnTime in the Phoenix object's parameters, so that it will respawn at regular intervals after it has been destroyed.

Assault Maps

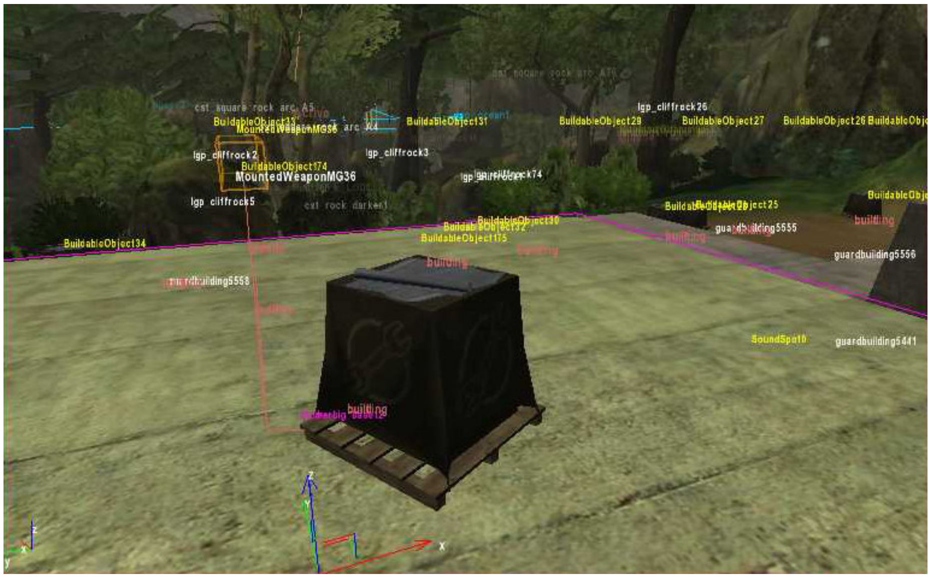
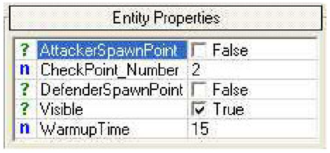


Figure 6.5 Airfield is typical of Assault type multiplayer maps.

Assault maps work by having three points which teams of players must capture and defend. The attacking team has a set amount of time to capture all three points, in order. When they capture one, the action moves to the next point, until the game is over. While you can set up Assault maps on Deathmatch maps, it is likely you will want to create separate maps especially for these, as there are many differences in strategies and objects required. To create a new Assault map, all you need to get started is to create a new mission. Select Mission | New, and name the mission ASSAULT, which will tell the game that this map can be played as an ASSAULT varient. If you have already created the map, you can just rename it, as a new mission will remove all the objects you have already placed.



Entity Properties

?	AttackerSpawnPoint	□ False
n	CheckPoint_Number	2
?	DefenderSpawnPoint	□ False
?	Visible	☑ True
n	WarmupTime	15

Spawn points are required in assault maps too, but they differ in their type and placement. Instead of using ordinary ReSpawn objects you will need to take one of the special ASSAULTCheckPoints from the Multiplayer folder of the Entity objects list. So that the game will know that this is a spawn point, and not a check point, you will need to change the Visible setting to False. This will make it not show up on the map. In addition to this, you will need to uncheck either AttackerSpawnPoint or DefenderSpawnPoint, depending on whether it is a defender's spawn point or an attacker's. Remember that this is a team map, and you don't want your players spawning in the enemy's camp, or they won't be very happy.

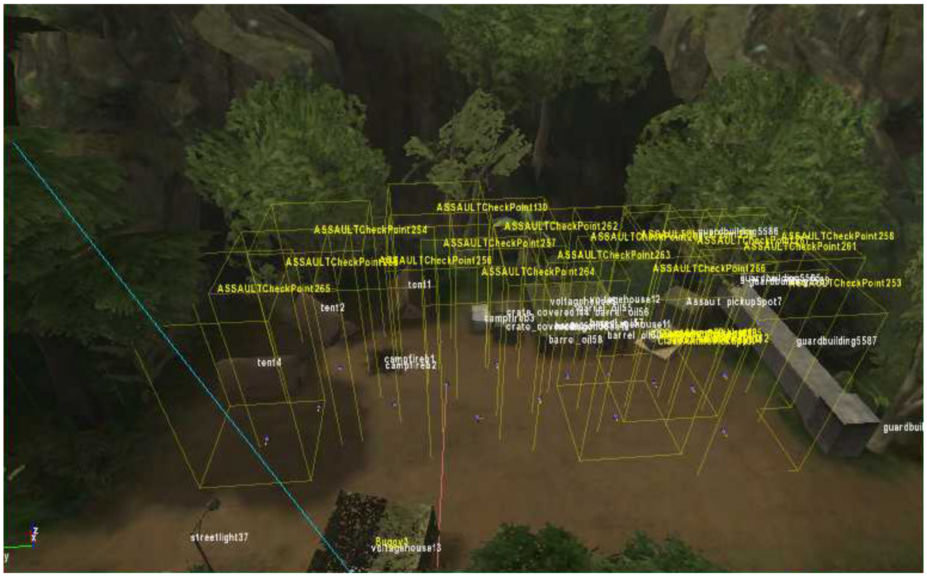


Figure 6.6 It is best to group spawn points for defenders and attackers separately, close to their allies and far from their enemies.

You can name your player spawn points whatever you like, and place them where you like, but you should arrange them so that they are grouped both by their team members, and near to the objective they are attacking or defending. You will need three groups of defender and attacker spawn points, one for each objective, and to make sure that players spawn at the correct point for the objective, you will need to set the CheckPoint_Number to 1 for the first objective, 2 for the second and 3 for the last. You will want as many spawn points at the first objective as there are players in the game, as they could all spawn at the same time. Later objectives can make do with far less, maybe half as many.

Note

Spawn points for spectators can be set up in the exact same way as in deathmatch maps.

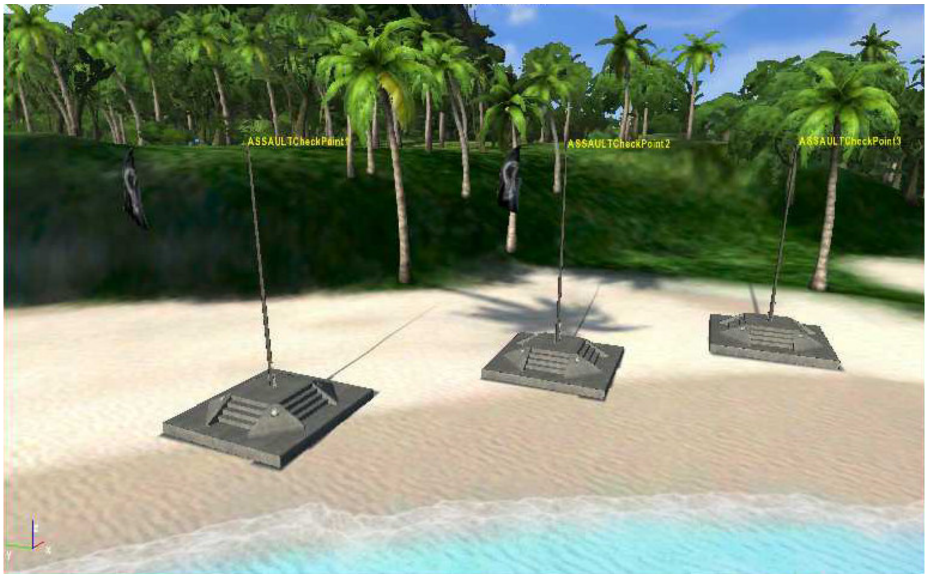


Figure 6.7 Players need to complete three check-point objectives on Assault maps.

Once you have the spawn points set up, you will need the objectives themselves. Objectives utilise the same object as the spawn points, only these are visible. You will also need to make sure that the AttackerSpawnPoint and DefenderSpawnPoint parameters are set to false, as these are not spawn points at all. Place the ASSAULTCheckPoint objects where you want the objectives to be, and name them whatever you like. For the first objective, set the CheckPoint_Number to 2, the second to 3 and the last to 4. One final consideration is the warm up time, which defines how long it will take before the objective can be captured by the enemy. The longer the warm up time, the more time the defenders will have to prepare.

In order to let your players know what is going on, you will need to advise them when an objective has been captured, and what they need to do next. To do this you need CurrentMission objects, which can be linked to the objectives, and triggered when they are captured, to present the player with an acknowledgement and a message. You will need four CurrentMission objects, the first will be linked to an attacker spawn point for the first objective, and the remaining three will be linked to each of the objectives, in order, with the final object giving the victory message. You must name these CurrentMission0 through CurrentMission4.

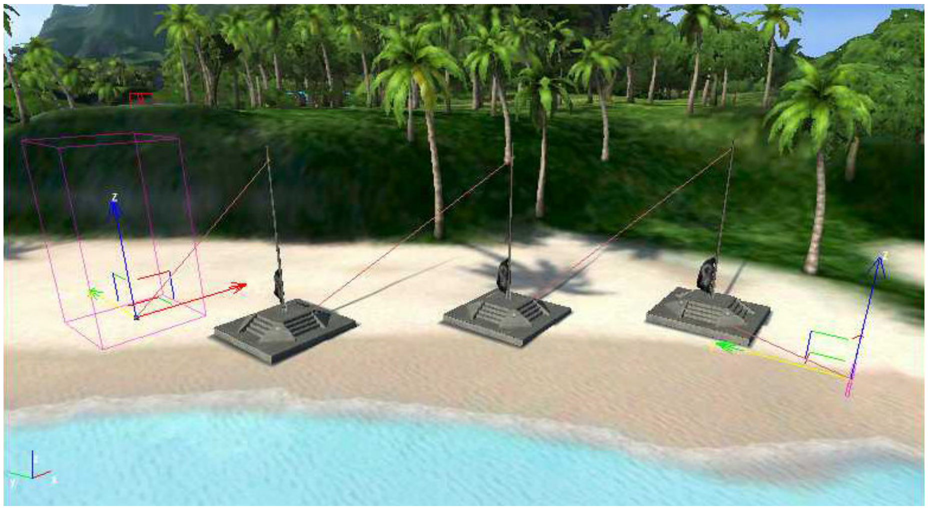


Figure 6.8 Place CurrentMission objects on the objectives and link them from attacker spawn point to closing CurrentMission object.

Place the first of these objects on the map, preferably near the first objective if you want the radar beacon to work properly, and name it CurrentMission0. You can link it to any of the attacker spawn points - it doesn't matter which as long as they are spawn points for the first objective. Link the two by selecting the attacker spawn point, and selecting the On Spawn event from the Input/Output Events list. Then click the Pick icon, and click on the CurrentMission0 object. Next, give the attackers and defenders a message, by editing the MissionTextAttacker and MissionTextDefender parameters, for example tell the defenders to "defend the tower". You can also add a sound for both teams, to indicate the game is underway. You will want to place CurrentMission1 near the next objective, and link that to the first objective. Link the two by selecting the objective and selecting the On Spawn event from the Input/Output Events list, and then picking the CurrentMission1 object in the same way as before.

Note

On Spawn works the same for both Objectives and Spawn Points when activating the CurrentMission objects, as the Objectives can't spawn anything.

You might want to link up the objectives to events, in addition to the CurrentMission objects. Many of Assault maps link a pulsing particle effect to the objective, so when it is captured by the attackers it gives a visible alert, along with the message and sound. You can achieve this by selecting the ParticleEffect object from the Particles directory of the Entity objects list, and placing it at the top of the objective's mast. Then make an event link between the objective and the particle effect, by causing a Pulse event in the particle effect, when the objective Spawns (On Spawn -> Pulse). You can also set it to alert the players in a similar way, for when it is under threat of capture, by triggering a signal on the objective's On Capturing event.

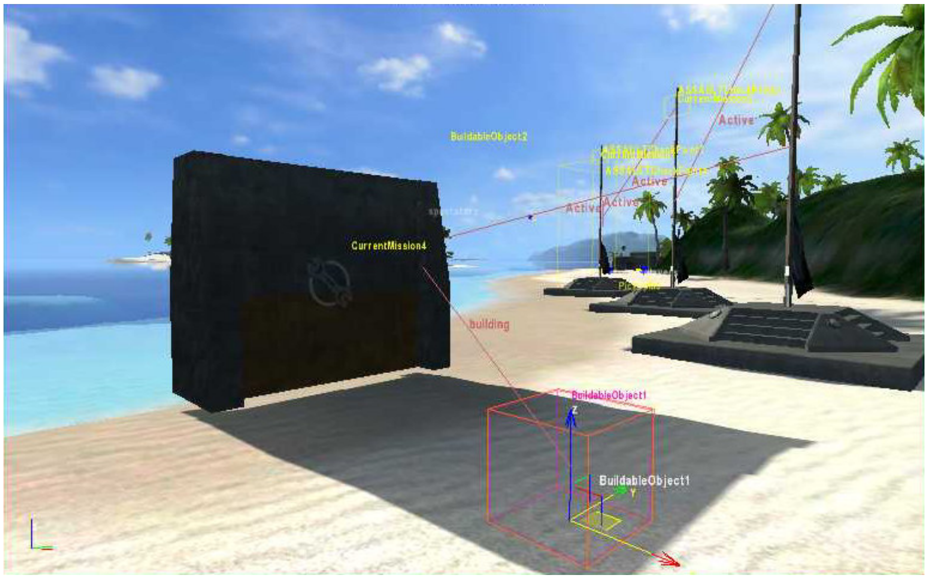
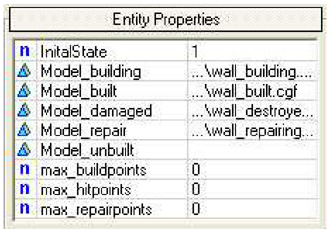


Figure 6.9 Buildable object base and selected unbuilt object.

One more vital component of an assault map are the buildable objects, such as walls and turrets, that the engineer can build and repair. To create a buildable object on your map, first drag-and-drop a BuildableObject from the Others directory of the Entity objects list. This will be the base that the engineer will build the actual object from with his wrench. For this first object, give it the model_building and model_unbuilt of whatever it is you are intending to built on that spot, for example if you want a wall building, give it a model such as unbuilt_wall.cgf. You won't want any models for the other build statuses, as you want the first object to be invisble at those points. Once that has been placed, you will need the actual building that will be constructed, and for that you will need another BuildableObject, the same as before.

Note

wall models can be found in the multiplayer/buildables/ folder of the objects directory.



Entity Properties

n	InitalState	1
▲	Model_building	...\wall_building....
▲	Model_built	...\wall_built.cgf
▲	Model_damaged	...\wall_destroye...
▲	Model_repair	...\wall_repairing...
▲	Model_unbuilt	
n	max_buildpoints	0
n	max_hitpoints	0
n	max_repairpoints	0

For this one you will need to give it a different model, for every model parameter except model_unbuilt, as you don't want anything displayed on the map before an engineer starts constructing it. You will want to choose models from the same family as you did for the first BuildableObject, for example wall_building, wall_built, wall_destroyed and wall_repairing, for the building, built, damaged and repair model parameters respectively. Once you have set this BuildingObject up, give it an amount of health points that you think reasonable, and link the first BuildableObject to the second via the On Building and On Built event triggers. From these send a building and built signal to the second BuildableObject (OnBuilt -> Built).

Finally, you will want weapons and other objects on your map for the players to use. Unlike in the deathmatch maps, you won't want to be placing weapons around the map, as the players can only use the weapons that are associated with the particular class they chose at the start of the game, for example an engineer can carry medical kits and revolver. Because of this you will need to place a special weapons pick-up object, called the ClassAmmoPickup, from the same Pickups directory as you found the others. You can still place other objects on the map like the cryvision goggles, as these are not class based.

Chapter

7

Single Player Missions

You can use the CryEngine Sandbox editor to create entire levels, and then link those levels together to create an entire game.

Your single player missions are likely to be more complicated than any multiplayer map that you create. You will need to tie a series of events together, from mission start to mission end, and update the player on what he has to do, etc., even linking one mission to another to create an entire game or sub-game. The missions are also a lot more freeform than multiplayer, and you will have to decide for yourself how things will proceed. That means that this section will cover as much as is necessary to create a mission, but you will have to decide what you will need for yourself.

Setting Up

The first thing you will need on any map is a respawn point. You need only one respawn point, but placing additional ones at each savespot helps debugging the mission. You can place a spawn point simply by placing a Respawn object from the TagPoint list in the Objects tab. To make it work you must name it Respawn and give it a number, for example Respawn1. You need to number each subsequent Respawn point sequentially, but the numbers don't need to follow one after another, i.e. you can have a sequence of spawn points numbered 1, 2, 4, 5, 8, and they will be triggered in that order. The lowest number will always be the first one.

You will also need to give the player some weapons, and tell the game what weapons will be allowed for that mission. To tell the game what weapons the player will have at the start, you need to create an Equipment Pack, the same way as described in the Multiplayer Map section. After picking the equipment pack in the Mission Properties window, you will then want to tell the game what weapons can be collected on that level. To do this click on the weapons tab, and add each weapon that will be used in the mission. If you don't add the weapon, it is possible that when the player picks it up in the game, it won't show in his inventory. When you change the equipment pack the player is starting with, the weapon that is active will be the one the player is starting with in his hands.

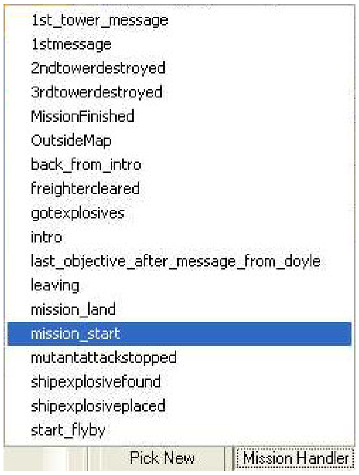
Note

The active weapon in the Equipment Pack is the one that is highlighted when you accept the weapons you have chosen.

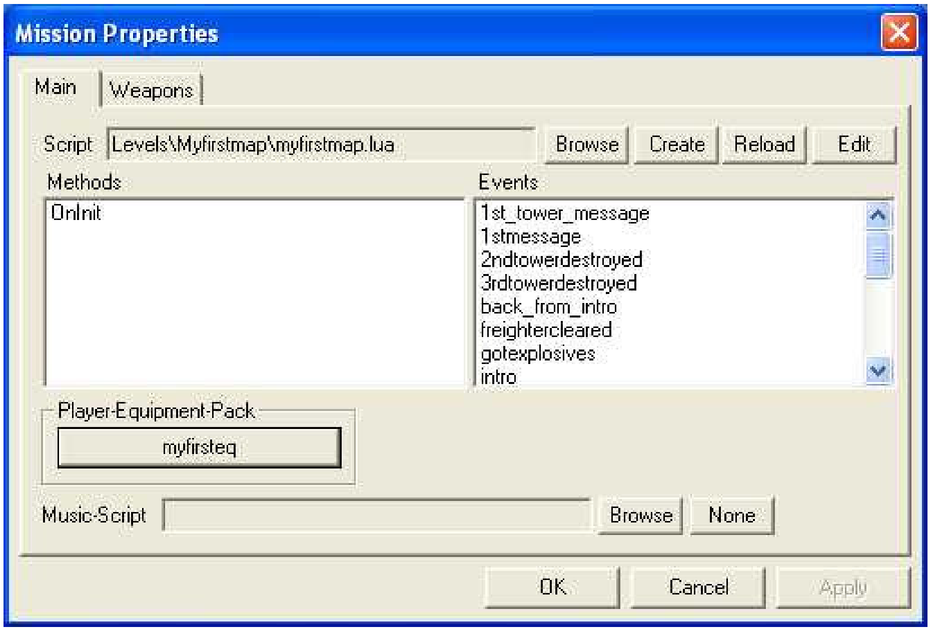
The last thing you will want to do in setting up the level is to trigger the first objective. To understand how to do this, you must first understand how the game processes the mission script.

Mission Scripts

The mission script is a LUA script which lists all the functions associated with the Mission Handler in the game. Each function in the Mission Handler can be actioned in the same way as an event triggered in another object. Instead of picking another object, and selecting and event to trigger in that object, you can instead invoke the Mission Handler, and select from a set of mission events instead. For example, you could set up a Proximity Trigger to set off a mission event which clears an objective from your objective list that instructed the player to reach the point the Proximity Trigger is covering. Every one of these mission events is defined within the mission script.



1st_tower_message
1stmessage
2ndtowerdestroyed
3rdtowerdestroyed
MissionFinished
OutsideMap
back_from_intro
freightercleared
gotexplosives
intro
last_objective_after_message_from_doyle
leaving
mission_land
mission_start
mutantattackstopped
shipexplosivefound
shipexplosiveplaced
start_flyby



Script: Levels\Myfirstmap\myfirstmap.lua

Methods
OnInit

Events
1st_tower_message
1stmessage
2ndtowerdestroyed
3rdtowerdestroyed
back_from_intro
freightercleared
gotexplosives
intro

Player-Equipment-Pack: myfirsteq

Music-Script:

Figure 7.1 Use the Mission Properties Window to create your mission script.

To create a new mission script, select the Main tab in the Mission Properties window, and click the Create button next on the top tool bar. Mission Scripts are usually stored in the root directory of each level's directory, and given the level name, with the ".lua" extension. For example, the Fort level is stored in the Fort folder of the Levels directory, and called, simply, fort.lua. Once you have created this script, it should be listed in the script file name box. You can then click the Edit button, and you will be presented with the default script for any level, consisting of mission opening lines, and three functions, OnInit, OnUpdate and Finish.

The mission script itself is simply a list of function that create mission events for the Mission Handler. There are four types of key script instructions that you will need to create a basic mission script, and these are:

1. HUD instructions; information presented to the player on screen.

2. Game instructions; game related events like starting a new level.

3. Movie instructions; playing cut scenes.

4. Console instructions; actions console commands in the script.

There are also a number of other instruction types, some that relate specifically to the game, and some that are purely LUA related. Anything of particular use will be detailed in this section.

All mission event functions can be defined in the same way, and can include as many commands as required. First you must start the function with the following header:

function Mission:Event_eventname()

All you need to do is replace eventname with the name you wish to call the event; the event name will appear in the Mission Handler events list exactly as stated here. After this function header, you can list as many commands as you want, delimited by the "end" statement. The header, each command statement, and the end delimiter should all be on separate lines. You can also comment the code by using a double dash "--". Everything after the double dash will be ignored by the script compiler.

Note

After changing the script you have to reload the script so the changes are registered.

HUD instructions

Hud instructions relate to the information presented to the player via the screen, but the instructions aren't limited to simply passing messages to the player. They can also include setting and resetting radar points to direct a player to a new position, setting, completing, and clearing objectives. All Hud instructions are prefixed by "Hud:". The following are a list of common Hud instructions, and explanations for their usage.

Hud:AddMessage("text", seconds);

This instruction will pass a message, text, that will display on the players screen for as many seconds as are specified in the seconds parameter.

Hud:PushObjective({}, "text");

PushObjective adds the message, text, to the player's objectives list, the one that is accessed by pressing the Tab key. The first parameter is empty and can be ignored, although you must include the empty brackets.

Hud:CompleteObjective("text");

This completes the objective by greying it out on the player's objective screen. The text must match the text of the objective that has been completed, for example if you pushed an object with the text "get here!" then you must complete it with the text "get here!".

Hud:FlashObjectives({}, "");

This simply clears all objectives from the objectives list.

Hud:SetRadarObjective("tagpointname");

It isn't necessary to have a radar point for every object, but if you want the player to know where he is meant to go, you must give him a radar point to move to. You can tell the game where the radar point is by placing a tag point object on the map, and giving it a name. There are no naming conventions for this type of object, but you will probably find life easier if you give it a name like Radar1, Radar2, etc. With the radar tag point on the map, just enter that tag point name in the instruction above to set it for the player. Setting a radar objective removes any previous radar objective set.

Hud:SetRadarObjective("nil");

Sometimes you won't want to give the player any particular objective on the radar. In that case you can clear the object, rather than changing it, by passing the "nil" parameter in to SetRadarObjective.

Game Instructions

There are two major game instructions that you will want to use, and both relate to starting a new level. Both are actioned by sending a particular message to the game, rather than to the player, and will both be used at the end of your levels. They are both detailed below.

Game:SendMessage("StartLevelFade levelname");

This instruction starts the next level, named by levelname, and fades into it gracefully.

Game:SendMessage("StartLevel levelname");

Essentially this is the same as the above instruction, only it starts the next level abruptly, without any fading.

Movie Instructions

You can play movie sequences, i.e. cut scenes, with the following instruction:

Movie:PlaySequence("moviename");

Here moviename defines the name of the cut scene to be played.

Console Commands

All console commands can be used in the mission script, for example "fov=90" sets the player's field of view to 90. These console commands can be particularly useful for initialising a level.

Miscellaneous

There are much more to scripting a mission than can be explained here, and for a more detailed list see appendix ***********. There is one useful command that doesn't fit under the previous headings, and that is:

_localplayer.cntSavePlayerElements();

This command, when used at the end of a mission, will save all the equipment that a player has collected in the just completed level, and allow him to use this in the next. If you have set up an equipment pack in the next mission, this instruction will override that.

Save Points

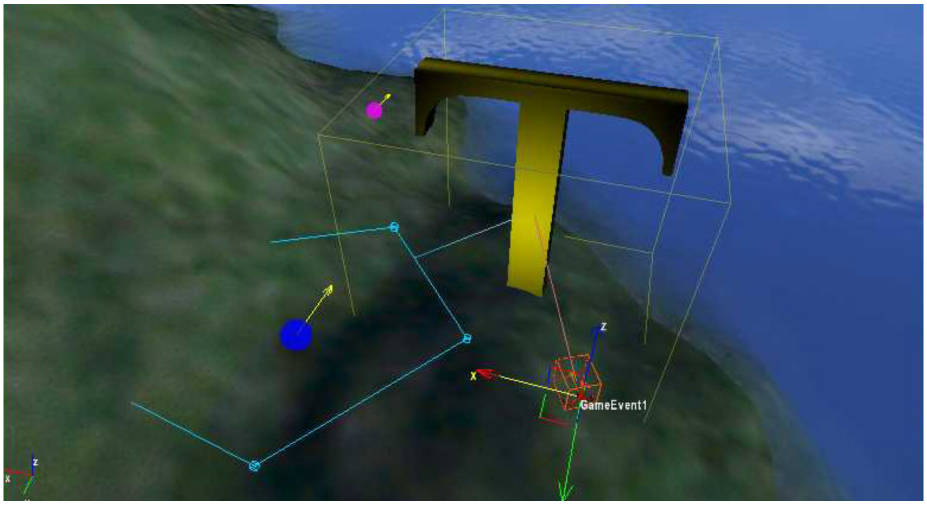
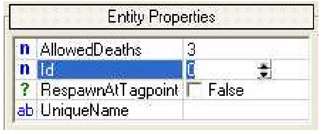


Figure 7.2 Connect GameEvents to triggers to create save game points.

You will want to have a number of save points in your game, and you will usually connect these up to completed objectives, as it can confuse a player to have to re-complete objectives. To save the game, you need a GameEvent object, and you need to trigger this with an event. You can use any kind of object to trigger this, even the death of an mercenary, but you will usually want to trigger the GameEvent from an area trigger that covers the point where you have placed your objective radar tag point.



Entity Properties

n	AllowedDeaths	3
n	Id	0
?	RespawnAtTagpoint	□ False
ab	UniqueName	

To place your save point on the map, simply drop a GameEvent object somewhere, it doesn't matter where, but you will likely want it close to your objective. Then place the object that will trigger it, such as an area trigger, and then link the trigger to the GameEvent by picking the Save event. To prevent the trigger from re-saving the game repeatedly, for example if the player walks back into the trigger area, you will want to make sure that the trigger's TriggerOnce parameter is set to true. In the parameters for the GameEvent object you will need to give the object a unique ID. The ID must be unique for the level, as it is used in defining the save game name.

Note

The GameEvent object is used as a spawn point for when the player dies after the game has been saved. Like all spawn points you can alter the direction the player is facing by rotating the arrow, this time an anchor arrow, to point in the direction you want the player to be facing.

Testing your Mission

You will want to test your completed mission in the full game. You will want to do this even before it is finished, as some aspects of the game will simply not work in the editor, and only function properly in the full game. To test the game this way you must first export it, using the Export to engine function in the File menu. You will then need to load the game, and from the main menu call up the console by pressing the ` key. From the console type the following console command:

```
\map mapname
```

replacing mapname with the name of your map.

Appendix A

Object Properties List

A detailed list of all the known properties of objects at time of documentation.

BUTTON AI

AIPath: defines a path for AI entities.
Naming convention: n_PATH

Parameter	Explanation	Range
Width	Unknown for this tool.	
Height	Sets height of path; must be zero for path to work.	
AreaID	Unknown for this tool.	
GroupID	Unknown for this tool.	
Closed	Determines if area is closed, i.e. no gap.	T/F
DisplayFilled	Unknown for this tool.	T/F

ForbiddenArea: defines the area where an AI can or can't walk.

Parameter	Explanation	Range
Width	Unknown for this tool.	
Height	Always 0	
AreaID	Unknown for this tool.	
GroupID	Unknown for this tool.	
Closed	Determines if area is closed, i.e. no gap.	T/F
DisplayFilled	Display closed area filled.	T/F

AINavigationModifier: defines an area within Forbidden Area to allow AI entity to traverse it.

Parameter	Explanation	Range
Width	Unknown for this tool.	
Height	Sets height of Area. If 0 and closed it will generate a space where the AI will work as if indoors, with waypoints, etc. If it has a value it will generate this space at this height. Everything above or below will be considered outdoor. Everything inside will be considered indoor.	
AreaID	Unknown for this tool.	
GroupID	Unknown for this tool.	
Closed	Determines if area is closed, i.e. no gap.	T/F
DisplayFilled	Display closed area filled.	T/F

AIHorizontalPlane: defines a plane for the ocean and water that the AI will see as "soft-cover".
That means a plane will block the AI's view, like dense foliage.

Parameter	Explanation	Range
Width	Unknown for this tool.	
Height	Unknown for this tool..	
AreaID	Unknown for this tool.	
GroupID	Unknown for this tool.	
Closed	Determines if area is closed, i.e. no gap.	T/F
DisplayFilled	Display closed area filled.	T/F

AIPoint: point for AI navigation within an area, usually NavigationModifier area.

Parameter	Explanation	Range
Waypoint	Defines object as being used as a point for the AI entity to move to.	Radio
Hide Point	Defines object as being used as a point for the AI entity to hide at.	Radio
Entry Point	Defines object as being used to allow the AI entity to enter or exit a NavigationModifier area.	Radio
Exit Point	Defines object as being used to allow the AI entity to enter or exit a NavigationModifier area.	Radio

AIAnchor: action modifier for nearby AI entities.

Parameter	Explanation	Range
Action	Determines he action carried out by AI entity.	List

BUTTON Area

Shape: defines an area to be used in conjunction with another object, e.g. area trigger.

Parameter	Explanation	Range
Width	Unknown for this tool.	
Height	Sets height of shape.	
AreaID	Unknown for this tool.	
GroupID	Unknown for this tool.	
Closed	Determines if area is closed, i.e. no gap.	T/F
DisplayFilled	Unknown for this tool.	T/F

AreaBox: same as shape, but with a fade property and a set rectangular shape.

Parameter	Explanation	Range
AreaID	Unknown for this tool.	
FadeInZone	Determines if there is a fade area inside the box.	
Width	Sets width of the zone.	
Length	Sets length of zone.	
Height	Sets height of the zone.	
GroupID	Unknown for this tool.	

AreaSphere: same as AreaBox, but a set sphere shape.

Parameter	Explanation	Range
AreaID	Unknown for this tool.	
FadeInZone	Determines if there is a fade area inside the box.	
Radius	Defines radius of sphere.	
GroupID	Unknown for this tool.	

WaterVolume: defines a volume of water.

Note: must be used in conjunction with Database material editor for water specifications.

Parameter	Explanation	Range
Width	Unknown for this tool.	
Height	Sets height of water; must be negative as volume goes below surface.	
AreaID	Unknown for this tool.	
GroupID	Unknown for this tool.	
Closed	Determines if area is closed, i.e. no gap.	T/F
DisplayFilled	Unknown for this tool.	T/F
Shader	Defines shader used for surface texture.	File
Speed	Defines how fast the water will flow.	
TriMinSize	Unknown for this tool.	
TriMaxSize	Unknown for this tool.	
AffectToVolFog	Unknown for this tool.	

VisArea: defines for the engine what are internal areas.

Parameter	Explanation	Range
Width	Unknown for this tool.	
Height	Sets height of area; must not be zero.	
AreaID	Unknown for this tool.	
GroupID	Unknown for this tool.	
Closed	Determines if area is closed, i.e. no gap.	T/F
DisplayFilled	Unknown for this tool.	T/F
AmbientColor	Defines ambient colour within the VisArea	RGB
DynAmbientColor	Defines the dynamic ambient colour wiith the VisArea.	RGB
AffectedBySun	Determines if area is affected by the sun.	T/F
ViewDistRatio	Defines the view distance within the area.	
SkyOnly	If a player is inside the VisArea,, determines if only the sky will be rendered. Used for some kind of windows at ceilings.	T/F

Portal: defines a view area for internal/external areas.

Note: must overlap a VisArea.

Parameter	Explanation	Range
Width	Unknown for this tool.	
Height	Sets height of water; must not be zero.	
AreaID	Unknown for this tool.	
GroupID	Unknown for this tool.	

Closed	Determines if area is closed, i.e. no gap.	T/F
DisplayFilled	Unknown for this tool.	T/F
AmbientColor	Defines ambient colour within the VisArea	RGB
DynAmbientColor	Defines the dynamic ambient colour wiith the VisArea.	RGB
AffectedBySun	Determines if area is affected by the sun.	T/F
ViewDistRatio	Defines the view distance within the area.	
SkyOnly	Unknown for this tool.	T/F
UseDeepness	Unknown for this tool.	T/F
DoubleSide	Enables or disables viewing from both sides of portal.	T/F

OccluderArea: defines an area that will not be drawn until player is in actually in area.

Parameter	Explanation	Range
Width	Unknown for this tool.	
Height	Sets height of water; must not be zero.	
AreaID	Unknown for this tool.	
GroupID	Unknown for this tool.	
Closed	Determines if area is closed, i.e. no gap.	T/F
DisplayFilled	Unknown for this tool.	T/F
AmbientColor	Defines ambient colour within the VisArea	RGB
DynAmbientColor	Defines the dynamic ambient colour wiith the VisArea.	RGB
AffectedBySun	Determines if area is affected by the sun.	T/F
ViewDistRatio	Defines the view distance within the area.	
SkyOnly	Unknown for this tool.	T/F
UseDeepness	Unknown for this tool.	T/F
DoubleSide	Enables viewing from both sides of portal.	T/F

FogVolume: UNKNOWN TOOL

Parameter	Explanation	Range
Width	Unknown for this tool.	
Length	Unknown for this tool.	
Height	Unknown for this tool.	
ViewDistance	Unknown for this tool.	
Shader	Unknown for this tool.	File
Color	Unknown for this tool.	RGB

BUTTON Brush

Generic Parameters: all objects have the following properties by default and can be resized

Parameter	Explanation	Range
Prefab	Defines 3D model of object.	File
OutdoorOnly	Determines if the object is only seen outdoors. If the object belongs outdoors, it will be forced outside even if physically inside.	T/F
CastShadowVolume	Determines if object casts a volumetric shadow.	T/F
SelfShadow	Determines if the object can cast a shadow on itself.	T/F
CastShadowMap	Determines if there will be a shadow texture of the object cast on the environment.	T/F
RecvShadowMap	Determines if the object can receive shadow textures from other objects.	T/F
CastLightMap	Determines if the object can cast a light map upon other objects.	T/F
ReceiveLightMap	Determines if the object can receive a light map cast by other objects.	T/F
Hideable	Determines if the object can be used by AI entities to hide behind.	T/F
LodRatio	Defines the distance at which the object reverts to low detail.	
ViewDistanceRatio	Defines the distance at which the object will no longer be visible to the player.	
NotTriangulate	Determines if object will not be included in AI triangulation. If set the AI will disregard the object.	T/F
LightMapQuality	Determines the quality of the light map cast.	

BUTTON Entity

Generic Parameters: all objects have the following properties by default and cannot be resized

Parameter	Explanation	Range
CastShadowVolume	Determines if object casts a volumetric shadow.	T/F
SelfShadow	Determines if the object can cast a shadow on itself.	T/F
CastShadowMap	Determines if there will be a shadow texture of the object cast on the environment.	T/F
RecvShadowMap	Determines if the object can receive shadow textures from other objects.	T/F
CastLightMap	Determines if the object can cast a light map upon other objects.	T/F
ReceiveLightMap	Determines if the object can receive a light map cast by other objects.	T/F
LodRatio	Defines the distance at which the object reverts to low detail.	
ViewDistanceRatio	Defines the distance at which the object will no longer be visible to the player.	
SkipOnLowSpec	Determines if the object will appear on low spec computers.	T/F
HiddenInGame	Determines if the object will be invisible in the game.	T/F

AI Folder

AISphere: UNKNOWN TOOL

Parameter	Explanation	Range
InnerRadius	Determines radius of the AI Sphere	

CreatureGenerator: spawns a number of AI entities.

Parameter	Explanation	Range
CreatureType	Name of creature to be generated.	Text
MaxCreatures	Defines the maximum number of creatures to be created.	

Grunt: basic soldier.

Parameter	Explanation	Range
GunReady	Determines if the weapon starts in the grunt's hand.	T/F
HelmetModel	Defines the 3D model used for this object's helmet.	File
HelmetOnStart	Determines if the grunt starts with a helmet or not.	T/F
HelmetProtection	Determines if the grunt receives protection from a headshot.	T/F
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the grunt's group.	
Sightrange	Defines the maximum sight range of the grunt.	
Soundrange	Defines the maximum hearing range of the grunt.	
AffectSOM	Determines if the AI will affect the radar.	T/F
AnimPack	Names the animation pack the AI will use.	Text
DropPack	Names the equipment the AI will drop upon death.	Equip
Equipment	Names the equipment the AI will use.	Equip
GroupHostility	Unknown for this tool.	
HasArmor	Determines if the AI has armour.	T/F
KEYFRAME_TABLE	Unknown for this tool.	Text
Model	Points to the 3D model of the AI.	File
Persistence	Unknown for this tool.	
ReinforcePoint	Names the tag point used by AI for reinforcement event.	Text
SOUND_TABLE	Unknown for this tool.	
SleepOnSpawn	Determines if the AI is activated once the map starts.	T/F
SoundPack	Names the sound pack the AI uses.	Text
SpeciesHostility	Unknown for this tool.	
TakeProximityDamage	Determines if the AI takes proximity damage.	T/F
Trackable	Unknown for this tool.	T/F
Accuracy	Sets the shooting accuracy of the AI.	
Aggression	Sets the aggression of the AI.	
Attackrange	Sets the range within which the target must come before the AI will start shooting.	
Back_speed	Defines the speed at which the AI can walk backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
Eye_height	Unknown for this tool.	
Forward_speed	Defines the speed at which the AI can walk forwards.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	Sets the number of the first path node.	
Pathsteps	Defines the number of steps involved in the path.	
Responsiveness	Unknown for this tool.	
Special	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	

AniRefSpeeds	Determines the movement speed of the AI.	
SpeedScales	Determines the affect upon speed of AI stances.	

Gunship: Helicopter attack ship and troop carrier.

Parameter	Explanation	Range
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the vehicle's group.	
Sightrange	Defines the maximum sight range of the vehicle.	
Soundrange	Defines the maximum hearing range of the vehicle.	
AttackAltitude	The height from which the helicopter AI will attack from.	
BendForce	Unknown for this tool.	
DropPack	Unknown for this tool.	
DmgScaleBullet	Damage modifier for the helicopter's gun fire.	
DmgScaleExplosion	Damage modifier for the explosions from this helicopter.	
FadeEngineSound	Sets whether the engine sounds fades in and out with the distance of the AI from the player.	T/F
FlightAltitude	Sets the maximum altitude for the helicopter.	
FlightAltitudeMin	Sets the minimum altitude for the helicopter.	
GroupHostility	Unknown for this tool.	
GunModel	Defines the 3D model of the gun mounted on the helicopter.	File
IgnoreCollisions	Determines if the helicopter can ignore collisions.	T/F
IsKiller	Unknown for this tool.	T/F
KillDist	Unknown for this tool.	
Model	Defines the 3D model for the helicopter	File
Pathloop	Determines if the helicopter will loop at the end of its path.	T/F
Persistence	Unknown for this tool.	
SoundOutdoorOnly	Determines if the you will only hear the sound of the object outside of buildings.	T/F
SpeciesHostility	Unknown for this tool.	
StartDelay	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Attackrange	Sets how close the AI must be to target before it will start firing.	
Back_speed	The speed at which the AI can move backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
DropAltitude	Unknown for this tool.	
Eye_height	Unknown for this tool.	
Forward_speed	The speed at which the AI can move forwards.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
PointAttack	Defines the name of the tag point that the helicopter will attack once ordered.	Text
PointBackOff	Defines the name of the tag point that the helicopter will return to after finishing its job, losing its pilot, or taking damage.	Text
PointReinforce	Defines the name of the tag point that the helicopter will reinforce once ordered.	Text
Responsiveness	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
Vertical_fov	Sets the vertical field of view for the helicopter.	0-180
ExplosionParams	Settings for the explosion caused by the helicopter's destruction.	
Damage	Damage caused when the helicopter is destroyed.	
ImpulsivePressure	Unknown for this tool.	
Radius	Unknown for this tool.	
RadiusMax	Defines maximum radius of explosion.	
RadiusMin	Defines minimum radius of explosion	
GunnerParams	Settings for the helicopter's gunner.	
AttackRange	Sets how close the gunner must be to target before it will start firing.	
Horizontal_fov	Defines the horizontal field of view for the gunner.	0-360
Responsiveness	Responsiveness of the turret to enemy movement.	
Sightrange	Defines the sight range of the gunner.	

MercCover: mercenary soldier that provides cover for other soldiers when grouped.

Parameter	Explanation	Range
GunReady	Determines if the weapon starts in the grunt's hand.	T/F
HasLight	Determines if the Mercenary holds a light or not.	T/F
HelmetModel	Defines the 3D model used for this object's helmet.	File
HelmetOnStart	Determines if the grunt starts with a helmet or not.	T/F
HelmetProtection	Determines if the grunt receives protection from a headshot.	T/F
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the grunt's group.	
Sightrange	Defines the maximum sight range of the grunt.	
Soundrange	Defines the maximum hearing range of the grunt.	
SpecialInfo	Unknown for this tool.	
AffectSOM	Unknown for this tool.	T/F

AnimPack	The name of the animation pack the AI will use.	Text
AwareOfPlayerTarget	Unknown for this tool.	T/F
DamageMultiplier	Unknown for this tool.	
DropPack	The equipment the AI will drop upon death.	Equip
DumbRocket	Determines whether the merc can fire a rocket.	T/F
Equipment	The equipment the AI will use.	Equip
GroupHostility	Unknown for this tool.	
HasArmor	Determines if the AI has armour.	T/F
HasShield	Determines if the AI has a shield.	T/F
Invulnerable	Determines if the AI can be killed.	T/F
KEYFRAME_TABLE	Unknown for this tool.	Text
MeleeDistance	Defines the melee range of the merc.	
Model	The 3D model of the AI.	File
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
RocketDamageOverride	Unknown for this tool.	
RocketSpeed	Unknown for this tool.	
RushPercentage	Unknown for this tool.	
SOUND_TABLE	Unknown for this tool.	
ShootSmartRocketForward	Determines if the merc can fire a seeker missile forward of its position.	T/F
SleepOnSpawn	Determines if the AI is activated once the map starts.	T/F
SoundPack	The name of the sound pack the AI uses.	Text
SpeciesHostility	Unknown for this tool.	
TakeProximityDamage	Determines if the AI takes proximity damage.	T/F
Trackable	Unknown for this tool.	T/F
Accuracy	Sets the shooting accuracy of the AI.	
Aggression	Sets the aggression of the AI.	
Attackrange	Sets the range within which the target must come before the AI will start shooting.	
Back_speed	The speed at which the AI can walk backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
CustomParticle	Unknown for this tool.	Text
Eye_height	Unknown for this tool.	
Forward_speed	The speed at which the AI can walk forwards.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
Responsiveness	Unknown for this tool.	
Special	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
SuppressedThrhld	Unknown for this tool.	
AniRefSpeeds	Determines the movement speed of the AI.	
SpeedScales	Determines the affect upon speed of AI stances.	

MercRear: mercenary soldier that defends the rear of a group.

Parameter	Explanation	Range
GunReady	Determines if the weapon starts in the grunt's hand.	T/F
HelmetModel	Defines the 3D model used for this object's helmet.	File
HelmetOnStart	Determines if the grunt starts with a helmet or not.	T/F
HelmetProtection	Determines if the grunt receives protection from a headshot.	T/F
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the grunt's group.	
Sightrange	Defines the maximum sight range of the grunt.	
Soundrange	Defines the maximum hearing range of the grunt.	
AffectSOM	Unknown for this tool.	T/F
AnimPack	The name of the animation pack the AI will use.	Text
DropPack	The equipment the AI will drop upon death.	Equip
Equipment	The equipment the AI will use.	Equip
GroupHostility	Unknown for this tool.	
HasArmor	Determines if the AI has armour.	T/F
KEYFRAME_TABLE	Unknown for this tool.	Text
Model	The 3D model of the AI.	File
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SOUND_TABLE	Unknown for this tool.	
SleepOnSpawn	Determines if the AI is activated once the map starts.	T/F
SoundPack	The name of the sound pack the AI uses.	Text
SpeciesHostility	Unknown for this tool.	
TakeProximityDamage	Determines if the AI takes proximity damage.	T/F
Trackable	Unknown for this tool.	T/F
Accuracy	Sets the shooting accuracy of the AI.	
Aggression	Sets the aggression of the AI.	

Attackrange	Sets the range within which the target must come before the AI will start shooting.	
Back_speed	The speed at which the AI can walk backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
Eye_height	Unknown for this tool.	
Forward_speed	The speed at which the AI can walk forwards.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
Responsiveness	Unknown for this tool.	
Special	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
SuppressedThrhld	Unknown for this tool.	
AniRefSpeeds	Determines the movement speed of the AI.	
SpeedScales	Determines the affect upon speed of AI stances.	

MercScout: mercernary soldier that plays the role of scout for a group.

See MercRear

MercSniper: mercenary soldier that plays the role of sniper for a group.

See MercRear

MutantBezerker: not known to work

Parameter	Explanation	Range
HelmetProtection	Determines if the grunt receives protection from a headshot.	T/F
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the grunt's group.	
Sightrange	Defines the maximum sight range of the grunt.	
Soundrange	Defines the maximum hearing range of the grunt.	
AffectSOM	Unknown for this tool.	T/F
AnimPack	The name of the animation pack the AI will use.	Text
DropPack	The equipment the AI will drop upon death.	Equip
Equipment	The equipment the AI will use.	Equip
GroupHostility	Unknown for this tool.	
HasArmor	Determines if the AI has armour.	T/F
HelmetModel	Defines the 3D model used for this object's helmet.	File
HelmetOnStart	Determines if the grunt starts with a helmet or not.	T/F
KEYFRAME_TABLE	Unknown for this tool.	Text
MeleeDamage	Defines the amount of melee damage caused by monkey.	
MeleeDistance	Defines the range of the melee attack.	
Model	The 3D model of the AI.	File
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SOUND_TABLE	Unknown for this tool.	
SleepOnSpawn	Determines if the AI is activated once the map starts.	T/F
SoundPack	The name of the sound pack the AI uses.	Text
SpeciesHostility	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Accuracy	Sets the shooting accuracy of the AI.	
Aggression	Sets the aggression of the AI.	
Attackrange	Sets the range within which the target must come before the AI will start shooting.	
Back_speed	The speed at which the AI can walk backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
Eye_height	Unknown for this tool.	
Forward_speed	The speed at which the AI can walk forwards.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
Responsiveness	Unknown for this tool.	
Special	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
SuppressedThrhld	Unknown for this tool.	
AniRefSpeeds	Determines the movement speed of the AI.	
SpeedScales	Determines the affect upon speed of AI stances.	

MutantCover: not known to work

Parameter	Explanation	Range
HelmetProtection	Determines if the grunt receives protection from a headshot.	T/F
Scale	Unknown for this tool.	
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the grunt's group.	
Sightrange	Defines the maximum sight range of the grunt.	
Soundrange	Defines the maximum hearing range of the grunt.	
AffectSOM	Unknown for this tool.	T/F
AnimPack	The name of the animation pack the AI will use.	Text
DropPack	The equipment the AI will drop upon death.	Equip
DumbRockets	Determines whether the mutant can fire a rocket.	T/F
Equipment	The equipment the AI will use.	Equip
GroupHostility	Unknown for this tool.	
HasArmor	Determines if the AI has armour.	T/F
HelmetModel	Defines the 3D model used for this object's helmet.	File
HelmetOnStart	Determines if the grunt starts with a helmet or not.	T/F
KEYFRAME_TABLE	Unknown for this tool.	Text
MeleeDamage	Defines the amount of melee damage caused by monkey.	
MeleeDistance	Defines the range of the melee attack.	
Model	The 3D model of the AI.	File
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SOUND_TABLE	Unknown for this tool.	
ShootSmartRockets	Determines if the mutant can fire a seeker missile.	T/F
SingleMeleeKillAI	Unknown for this tool.	T/F
SleepOnSpawn	Determines if the AI is activated once the map starts.	T/F
SoundPack	The name of the sound pack the AI uses.	Text
SpeciesHostility	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Accuracy	Sets the shooting accuracy of the AI.	
Aggression	Sets the aggression of the AI.	
Attackrange	Sets the range within which the target must come before the AI will start shooting.	
Back_speed	The speed at which the AI can walk backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
Eye_height	Unknown for this tool.	
Forward_speed	The speed at which the AI can walk forwards.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
Responsiveness	Unknown for this tool.	
Special	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
SuppressedThrhld	Unknown for this tool.	
AniRefSpeeds	Determines the movement speed of the AI.	
SpeedScales	Determines the affect upon speed of AI stances.	

MutantMonkey: not known to work

Parameter	Explanation	Range
Helmet Protection	Determines if the grunt receives protection from a headshot.	T/F
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the grunt's group.	
Sightrange	Defines the maximum sight range of the grunt.	
Soundrange	Defines the maximum hearing range of the grunt.	
AffectSOM	Unknown for this tool.	T/F
AnimPack	The name of the animation pack the AI will use.	Text
DropPack	The equipment the AI will drop upon death.	Equip
Equipment	The equipment the AI will use.	Equip
GroupHostility	Unknown for this tool.	
HasArmor	Determines if the AI has armour.	T/F
HelmetModel	Defines the 3D model used for this object's helmet.	File
HelmetOnStart	Determines if the grunt starts with a helmet or not.	T/F
JUMP_TABLE	Unknown for this tool.	Text
JumpAngle	Unknown for this tool.	Text
KEYFRAME_TABLE	Unknown for this tool.	Text
MeleeDamage	Defines the amount of melee damage caused by monkey.	
MeleeDistance	Defines the range of the melee attack.	
Model	The 3D model of the AI.	File
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SOUND_TABLE	Unknown for this tool.	
SleepOnSpawn	Determines if the AI is activated once the map starts.	T/F

SoundPack	The name of the sound pack the AI uses.	Text
SpeciesHostility	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Accuracy	Sets the shooting accuracy of the AI.	
Aggression	Sets the aggression of the AI.	
Attackrange	Sets the range within which the target must come before the AI will start shooting.	
Back_speed	The speed at which the AI can walk backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
Eye_height	Unknown for this tool.	
Forward_speed	The speed at which the AI can walk forwards.	
GravityMultiplier	Unknown for this tool.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
Responsiveness	Unknown for this tool.	
Special	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
SuppressedThrhld	Unknown for this tool.	
AniRefSpeeds	Determines the movement speed of the AI.	
SpeedScales	Determines the affect upon speed of AI stances.	

MutantRear: not known to work

Parameter	Explanation	Range
HelmetProtection	Determines if the grunt receives protection from a headshot.	T/F
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the grunt's group.	
Sightrange	Defines the maximum sight range of the grunt.	
Soundrange	Defines the maximum hearing range of the grunt.	
AffectSOM	Unknown for this tool.	T/F
AnimPack	The name of the animation pack the AI will use.	Text
AwareOfPlayerTarget	Unknown for this tool.	T/F
DropPack	The equipment the AI will drop upon death.	Equip
Equipment	The equipment the AI will use.	Equip
GroupHostility	Unknown for this tool.	
HasArmor	Determines if the AI has armour.	T/F
HelmetModel	Defines the 3D model used for this object's helmet.	File
HelmetOnStart	Determines if the grunt starts with a helmet or not.	T/F
KEYFRAME_TABLE	Unknown for this tool.	Text
MeleeDamage	Defines the amount of melee damage caused by monkey.	
MeleeDistance	Defines the range of the melee attack.	
Model	The 3D model of the AI.	File
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SOUND_TABLE	Unknown for this tool.	
SleepOnSpawn	Determines if the AI is activated once the map starts.	T/F
SoundPack	The name of the sound pack the AI uses.	Text
SpeciesHostility	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Accuracy	Sets the shooting accuracy of the AI.	
Aggression	Sets the aggression of the AI.	
Attackrange	Sets the range within which the target must come before the AI will start shooting.	
Back_speed	The speed at which the AI can walk backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
CustomParticle	Unknown for this tool.	
Eye_height	Unknown for this tool.	
Forward_speed	The speed at which the AI can walk forwards.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
Responsiveness	Unknown for this tool.	
Special	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
SuppressedThrhld	Unknown for this tool.	
AniRefSpeeds	Determines the movement speed of the AI.	
SpeedScales	Determines the affect upon speed of AI stances.	

MutantScout: not known to work

Parameter	Explanation	Range
HelmetProtection	Determines if the grunt receives protection from a headshot.	T/F
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the grunt's group.	
Sightrange	Defines the maximum sight range of the grunt.	
Soundrange	Defines the maximum hearing range of the grunt.	
AffectSOM	Unknown for this tool.	T/F
AnimPack	The name of the animation pack the AI will use.	Text
AwareOfPlayerTarget	Unknown for this tool.	T/F
DropPack	The equipment the AI will drop upon death.	Equip
Equipment	The equipment the AI will use.	Equip
GroupHostility	Unknown for this tool.	
HasArmor	Determines if the AI has armour.	T/F
HelmetModel	Defines the 3D model used for this object's helmet.	File
HelmetOnStart	Determines if the grunt starts with a helmet or not.	T/F
JUMP_TABLE	Unknown for this tool.	Text
JumpAngle	Unknown for this tool.	
KEYFRAME_TABLE	Unknown for this tool.	Text
MeleeDamage	Defines the amount of melee damage caused by monkey.	
Model	The 3D model of the AI.	File
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SOUND_TABLE	Unknown for this tool.	
SleepOnSpawn	Determines if the AI is activated once the map starts.	T/F
SoundPack	The name of the sound pack the AI uses.	Text
SpeciesHostility	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Accuracy	Sets the shooting accuracy of the AI.	
Aggression	Sets the aggression of the AI.	
Attackrange	Sets the range within which the target must come before the AI will start shooting.	
Back_speed	The speed at which the AI can walk backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
Eye_height	Unknown for this tool.	
Forward_speed	The speed at which the AI can walk forwards.	
Gravity_multiplier	Unknown for this tool.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
Responsiveness	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
SuppressedThrhld	Unknown for this tool.	
AniRefSpeeds	Determines the movement speed of the AI.	
SpeedScales	Determines the affect upon speed of AI stances.	

NPC: non player character.

Parameter	Explanation	Range
HelmetModel	Defines the 3D model used for this object's helmet.	File
HelmetOnStart	Determines if the grunt starts with a helmet or not.	T/F
HelmetProtection	Determines if the grunt receives protection from a headshot.	T/F
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the grunt's group.	
Sightrange	Defines the maximum sight range of the grunt.	
Soundrange	Defines the maximum hearing range of the grunt.	
AffectSOM	Unknown for this tool.	T/F
AnimPack	The name of the animation pack the AI will use.	Text
DropPack	The equipment the AI will drop upon death.	Equip
Equipment	The equipment the AI will use.	Equip
GroupHostility	Unknown for this tool.	
HasArmor	Determines if the AI has armour.	T/F
KEYFRAME_TABLE	Unknown for this tool.	Text
Model	The 3D model of the AI.	File
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SOUND_TABLE	Unknown for this tool.	
SleepOnSpawn	Determines if the AI is activated once the map starts.	T/F
SoundPack	The name of the sound pack the AI uses.	Text
SpeciesHostility	Unknown for this tool.	
TakeProximityDamage	Determines if the AI takes proximity damage.	T/F
Trackable	Unknown for this tool.	T/F
Accuracy	Sets the shooting accuracy of the AI.	
Aggression	Sets the aggression of the AI.	

Attackrange	Sets the range within which the target must come before the AI will start shooting.	
Back_speed	The speed at which the AI can walk backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
Eye_height	Unknown for this tool.	
Forward_speed	The speed at which the AI can walk forwards.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
Responsiveness	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
SuppressedThrhld	Unknown for this tool.	
AniRefSpeeds	Determines the movement speed of the AI.	
SpeedScales	Determines the affect upon speed of AI stances.	

Pig: AI animal.

See NPC

Shark: AI animal.

See NPC

SoundSupressor: suppresses sound in a certain range.

Parameter	Explanation	Range
Radius	Defines the range of the sound suppression.	

V22: VTOL troop carrier.

Parameter	Explanation	Range
Rope1Name	Defines the name of the first rope used by descending troops.	Text
Rope2Name	Defines the name of the second rope used by descending troops.	Text
Rope3Name	Defines the name of the third rope used by descending troops.	Text
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the vehicle's group.	
Sightrange	Defines the maximum sight range of the vehicle.	
Soundrange	Defines the maximum hearing range of the vehicle.	
AttackAltitude	The height from which the helicopter AI will attack from.	
BendForce	Unknown for this tool.	
DropPack	Unknown for this tool.	
DmgScaleBullet	Damage modifier for the helicopter's gun fire.	
DmgScaleExplosion	Damage modifier for the explosions from this helicopter.	
FadeEngineSound	Sets whether the engine sounds fades in and out with the distance of the AI from the player.	T/F
FlightAltitude	Sets the maximum altitude for the helicopter.	
FlightAltitudeMin	Sets the minimum altitude for the helicopter.	
GroupHostility	Unknown for this tool.	
GunModel	Defines the 3D model of the gun mounted on the helicopter.	File
IgnoreCollisions	Determines if the helicopter can ignore collisions.	T/F
IsKiller	Unknown for this tool.	T/F
KillDist	Unknown for this tool.	
Model	Defines the 3D model for the helicopter	File
Pathloop	Determines if the helicopter will loop at the end of its path.	T/F
Persistence	Unknown for this tool.	
SoundOutdoorOnly	Determines if the you will only hear the sound of the object outside of buildings.	T/F
SpeciesHostility	Unknown for this tool.	
StartDelay	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Attackrange	Sets how close the AI must be to target before it will start firing.	
Back_speed	The speed at which the AI can move backwards.	
Character	Sets the job of the AI.	List
Commrange	Defines the communication of the AI.	
DropAltitude	Unknown for this tool.	
Eye_height	Unknown for this tool.	
Forward_speed	The speed at which the AI can move forwards.	
Horizontal_fov	Defines the horizontal field of view of the AI in degrees.	0-360
Max_health	Sets the starting health of the AI.	
Path_name	Defines the name of the path that the AI will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
PointAttack	Defines the name of the tag point that the helicopter will attack once ordered.	Text
PointBackOff	Defines the name of the tag point that the helicopter will return to after finishing its job, losing its pilot, or taking damage.	Text

PointReinforce	Defines the name of the tag point that the helicopter will reinforce once ordered.	Text
Responsiveness	Unknown for this tool.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
Vertical_fov	Sets the vertical field of view for the helicopter.	0-180
ExplosionParams	Settings for the explosion caused by the helicopter's destruction.	
Damage	Damage caused when the helicopter is destroyed.	
ImpulsivePressure	Unknown for this tool.	
Radius	Unknown for this tool.	
RadiusMax	Defines maximum radius of explosion.	
RadiusMin	Defines minimum radius of explosion	
GunnerParams	Settings for the helicopter's gunner.	
AttackRange	Sets how close the gunner must be to target before it will start firing.	
Horizontal_fov	Defines the horizontal field of view for the gunner.	0-360
Responsiveness	Unknown for this tool.	
Sightrange	Defines the sight range of the gunner.	

Worm: AI animal.

Model no longer in editor.

Boids Folder

Birds: animated birds that fly across the map.

Parameter	Explanation	Range
ActivateOnStart	Determines if the birds will fly as soon as the map loads.	T/F
BoidFOV	Field of view of the birds.	0-360
BoidSize	Size of the Boid.	
FactorAlign	Unknown for this tool.	
FactorAvoidLand	Unknown for this tool.	
FactorCohesion	Unknown for this tool.	
FactorOrigin	Unknown for this tool.	
FactorSeparation	Unknown for this tool.	
FollowPlayer	Determines if the birds will follow player.	T/F
InnerRadius	Unknown for this tool.	
MaxAnimSpeed	Unknown for this tool.	
MaxAttractDist	Unknown for this tool.	
MinHeight	Unknown for this tool.	
MaxSpeed	Unknown for this tool.	
Model	Defines 3D model used by birds.	File
NoLanding	Determines if the birds can land on the ground or not.	T/F
NumBirds	Defines number of birds in the sphere.	
ObstacleAvoidance	Unknown for this tool.	T/F
OuterRadius	Unknown for this tool.	
VisibilityDistance	Defines the distances at which bugs will be displayed.	
Boid_mass	Unknown for this tool.	
Boid_radius	Unknown for this tool.	
Gravity_at_death	Unknown for this tool.	

Bugs: animated insects.

Parameter	Explanation	Range
ActivateOnStart	Determines if the birds will fly as soon as the map loads.	T/F
Animation	Unknown for this tool.	
AnimationSpeed	Unknown for this tool.	
Behaviour	Unknown for this tool.	
Character	Unknown for this tool.	
FactorOrigin	Unknown for this tool.	
FollowPlayer	Unknown for this tool.	
HeightMax	Unknown for this tool.	
HeightMin	Unknown for this tool.	
Model1	Defines first 3D model for bugs.	File
Model2	Defines second 3D model for bugs.	File
Model3	Defines third 3D model for bugs.	File
Model4	Defines fourth 3D model for bugs.	File
Model5	Defines fifth 3D model for bugs.	File
NoLanding	Determines if the bugs can land on the ground or not.	T/F
NumBugs	Defines the number of bugs used.	
Radius	Unknown for this tool.	
RandomMovement	Unknown for this tool.	
Scale	Unknown for this tool.	
SpeedMax	Defines the maximum movement speed.	
SpeedMin	Defines the minimum movement speed.	
VisibilityDist	Defines the distances at which bugs will be displayed.	

Fish: animated fish for underwater.

Parameter	Explanation	Range
ActivateOnStart	Determines if the fish will be animated as soon as the map loads.	T/F
BoidFOV	Field of view of the fish.	0-360
BoidSize	Size of the Boid.	
FactorAlign	Unknown for this tool.	
FactorAvoidLand	Unknown for this tool.	
FactorCohesion	Unknown for this tool.	
FactorOrigin	Unknown for this tool.	
FactorSeparation	Unknown for this tool.	
FollowPlayer	Determines if the birds will follow player.	T/F
InnerRadius	Unknown for this tool.	
MaxAnimSpeed	Unknown for this tool.	
MaxAttractDist	Unknown for this tool.	
MinHeight	Unknown for this tool.	
MaxSpeed	Unknown for this tool.	
Model	Defines 3D model used by fish.	File
NoLanding	Unknown for this tool	T/F
NumFish	Defines number of fish in the sphere.	
ObstacleAvoidance	Unknown for this tool.	T/F
VisibilityDistance	Defines the distances at which bugs will be displayed.	

Doors Folder

AutomaticDoor: door which opens automatically when player approaches.

Parameter	Explanation	Range
AnimatedModel	Unknown for this tool.	File
Automatic	Determines if door opens by itself, or if the player needs to press the use key.	T/F
CloseDelay	Defines the time before the door closes after player passes through.	
CloseSound	Defines the sound played upon closing.	File
CloseTimer	Sets whether the door closes or not after a period of time.	T/F
Enabled	Sets whether the door can be opened or not.	T/F
Model_Left	Defines the left hand door model.	File
Model_Right	Defines the right hand door model	File
MovingDistance	Defines how far the door will move into the wall.	
MovingSpeed	Defines the speed at which the door will open.	
NeededKey	Defines the number of the key needed to open door; -1 = none.	
OpenSound	Defines the sound made when door is opened.	File
PlayerBounce	Unknown for this tool.	
PlayerOnly	Determines if only the player can open the door.	T/F
TextInstruction	Defines the text to be displayed on the HUD when the player is in the doors bounding box.	Text
UnlockSound	Defines the doors unlocking sound.	File
UseAnimatedModel	Unknown for this tool.	T/F
UsePortal	Unknown for this tool.	T/F
AI Sound Event	Unknown for this tool.	
Enabled	Unknown for this tool.	T/F
Radius	Unknown for this tool.	
BBOX_Size	Defines the size of the bounding box.	
X	Size of bounding box on the X axis.	
Y	Size of bounding box on the Y axis.	
Z	Size of bounding box on the Z axis.	
Direction	Defines the direction the door will open in.	
X	Direction the door will move on the X axis.	
Y	Direction the door will move on the X axis.	
Z	Direction the door will move on the X axis.	

AutomaticDoor1Piece: UNKNOWN TOOL

Parameter	Explanation	Range
Automatic	Determines if door opens by itself, or if the player needs to press the use key.	T/F
CloseDelay	Defines the time before the door closes after player passes through.	
CloseSound	Defines the sound played upon closing.	File
CloseTimer	Sets whether the door closes or not after a period of time.	T/F
Enabled	Sets whether the door can be opened or not.	T/F
Model	Defines the 3D door model.	File
MovingDistance	Defines how far the door will move into the wall.	
MovingSpeed	Defines the speed at which the door will open.	
NeededKey	Defines the number of the key needed to open door; -1 = none.	
OpenSound	Defines the sound made when door is opened.	File
PlayerBounce	Unknown for this tool.	

PlayerOnly	Determines if only the player can open the door.	T/F
BBOX_Size	Defines the size of the bounding box.	
X	Size of bounding box on the X axis.	
Y	Size of bounding box on the Y axis.	
Z	Size of bounding box on the Z axis.	
Direction	Defines the direction the door will open in.	
X	Direction the door will move on the X axis.	
Y	Direction the door will move on the X axis.	
Z	Direction the door will move on the X axis.	

Door: non-automatic door.

Parameter	Explanation	Range
AnimatedModel	Unknown for this tool.	File
AnimationClose	Unknown for this tool.	Text
AnimationOpen	Unknown for this tool.	Text
AnimationOpenBack	Unknown for this tool.	Text
AnimationSpeed	Defines the speed at which the door opens.	
Automatic	Determines whether the player requires the use key in order to open door.	T/F
Bounds	Defines the bounding box parameters for the door.	XYZ
CloseDelay	Defines the delay before the door closes.	
Enabled	Sets whether the door can be used or not.	T/F
NeededKey	Defines the number of the key needed to open door; -1 = none.	
PlayerOnly	Determines if only the player can open the door.	T/F
TextInstruction	Defines the text to be displayed on the HUD when the player is in the doors bounding box.	Text
UsePortal	Unknown for this tool.	T/F
Sound	Defines sound properties for object.	
Close	Closing sound.	File
InnerRadius	Radius at which sound can be heard at full volume.	
Open	Opening sound.	File
OuterRadius	Radius at which sound can be heard at fading volume.	
Unlock	Unlocking sound.	File
Volume	Volume of sound.	

Elevator Folder

AutomaticElevator: elevator which operates automatically.

Parameter	Explanation	Range
Active	Sets elevator to active or inactive.	T/F
Automatic	Determines whether elevator operates automatically or not.	T/F
CloseDelay	Sets the delay before the elevator will automatically "close", i.e. move to its second location.	
LoopSound	Defines sound played when elevator moves; must be a loop.	File
MapVisMask	Unknown for this tool.	
MaterialDefault	Material used when elevator is stationary.	Text
MaterialDown	Material used when elevator is moving down.	Text
MaterialUp	Material used when elevator is moving up.	Text
Model	Defines 3D model used by elevator.	File
MovingDistance	Defines the distance the elevator will move upon activation by player.	
MovingSpeed	Defines the speed at which the elevator will move.	
OpenDelay	Sets the delay before the elevator will automatically "open", i.e. move to its first location.	
RetriggerDelay	Defines the amount of time before the elevator can be used again.	
StartSound	Defines the sound when the elevator starts.	File
EndSound	Defines the sound when the elevator stops.	File
Direction	Defines the direction the elevator travels in.	
X	Movement on the X axis.	
Y	Movement on the Y axis.	
Z	Movement on the Z axis; -1 = down.	
WarnLight	Defines parameters for the warning light.	
AffectsThisAreaOnly	Affects only the inside of elevator.	T/F
FakeLight	Use non-dynamic light.	T/F
HasWarnLight	Elevator has warning light.	T/F
HeatSource	Light is considered a heat source.	T/F
LightDiffuse	The diffuse colour of the light.	RGB
LightRadius	Radius of the light.	
LightRotSpeed.	Speed at which light rotates.	
LightSpecular	The specular colour of the light.	RGB
ProjectInAllDirections	Light will be projected in all directions.	T/F
ProjectorFOV	The angle at which light will be projected.	
ProjectedTexture	The texture the light will project.	File
UsedInRealTime	Unknown for this tool.	T/F
LightShader	Shader used for light.	List
LightAngles	Angle of the light source.	XYZ

FlyingFox: slide used for carrying play down rapidly.

Parameter	Explanation	Range
Enabled	Determines if the flying fox can be used or not.	T/F
LimitedRAngle	Sets the limits of the player's field of view when on the flying fox.	
Model	Defines the 3D model used for the flying fox.	File
Acceleration	Defines the acceleration speed of the flying fox.	
Destination	Defines the name of the tag point that the flying fox will move to.	Text
Message	Sets the text message to be used when player is in proximity to the flying fox.	Text
Velocity	Defines the maximum speed of the flying fox.	

Ladder: object used for manual movement up and down.

Parameter	Explanation	Range
AngleOffset	Unknown for this tool.	
HangleLimit	Unknown for this tool.	
LadderCGF	Defines 3D model for ladder.	File
LockDist	Defines the distance from the ladder at which the player is considered to be on it.	
Physicalize	Determines if the ladder is a physicalized object or not.	T/F

Lights Folder

DynamicLight: generic dynamic light object for use in all maps.

Parameter	Explanation	Range
Active	Sets the light on or off.	T/F
AffectsThisAreaOnly	Determines if light affects only the area in which it is placed.	T/F
AnimName	Name of animation applied to light.	Text
AnimationSpeed	Defines the speed at which the light is animated.	
CoronaScale	Unknown for this tool.	
Diffuse	Defines the diffuse colour of light.	RGB
DiffuseMultiplier	Defines brightness of light.	
Dot3Type	Determines if light will be a Dot3 type.	T/F
FakeLight	Determines if the light will use a non-dynamic source.	T/F
FakeRadiosity	Determines if fake radiosity will be pre-calculated for light.	T/F
HeatSource	Sets whether the light is treated as a heat source or not.	T/F
IgnoreTerrain	Determines if the light will affect the terrain or not.	T/F
LightDir	Defines the direction of the light.	XYZ
LightStyle	Sets the style of the light, e.g. flashing, pulsating, etc.	
Model01	Defines the model used for types 0 and 1.	File
Model2	Defines the model used for type 2.	File
Model3	Defines the model used for type 3.	File
OuterRadius	Unknown for this tool.	
ProjectInAllDirections	Determines if the light will shine in all directions.	T/F
ProjectorFOV	Defines the angle at which the light will be projected.	
ProjectorTexture	Defines the texture to be used by the light.	File
RndPosFreq	Unknown for this tool.	
Specular	Defines the colour of the specular light.	RGB
SpecularMultiplier	Defines the brightness of the specular light.	
UseAnimation	Unknown for this tool.	
UsedInRealTime	Unknown for this tool.	
Damping	Defines how much the impulse effects on the light will be reduced (dampened).	
LightShader	Defines the shader for the light, e.g. beam.	List
LightType	Selects which of the 3D models to use.	0-3
Max_time_step	Unknown for this tool.	
ShakeAmount	Defines how much impulse the light is given each time it is shaken.	
ShakeRefreshTime	Defines how often the light is shaken; 0 is the default and means the light will not be shaken.	
SleepSpeed	Unknown for this tool.	
Weight	Unknown for this tool.	
Optimization	Optimizations for low-spec machines.	
OnlyForHighSpec	Determines if the light is used only on high-spec machines.	T/F
SpecularOnlyForHighSpec	Determines if specular light is used only on high-spec machines	T/F

Mines Directory

AreaMine: mine which explodes when player enters an area shape linked to it.

Parameter	Explanation	Range
Enabled	Determines if mine is active or not.	T/F

FrogMine

Parameter	Explanation	Range
Delay	Defines the amount of time before the mine explodes.	
Enabled	Determines if the mine is active or not.	T/F
OnlyPlayer	Determines if only the player can activate the mine.	T/F
Radius	Defines the radius size that will set off the mine.	

ProximityMine: mine that explodes when player is near.

Parameter	Explanation	Range
ActivationSound	Defines sound to be played when mine is triggered.	File
Delay	Defines the amount of time before the mine explodes.	
Enabled	Determines if the mine is active or not.	T/F
ExplosionDamage	Defines how much damage the explosion will cause.	
Model	Defines the 3D model for the mine.	File
OnlyPlayer	Determines if only the player can activate the mine.	T/F
Radius	Defines the radius size that will set off the mine.	

Multiplayer Folder

ASSAULTCheckPoint: spawn point and check point on assault mode.

Parameter	Explanation	Range
AttackerSpawnPoint	Sets the spawn point to be attacker.	T/F
CheckPoint_Number	Sets the check point number.	
DefenderSpawnPoint.	Sets the spawn point to be defender.	T/F
Visible	Determines if the spawn point is visible or not.	T/F
WarmUpTime	Defines how much time there is before each game starts.	

BuildPoint: SELECTION OF THIS OBJECT CRASHES EDITOR

CAHFlag: UNKNOWN TOOL

Parameter	Explanation	Range
TimeDelay	Unknown for this tool.	
BlendType	Unknown for this tool.	
Bouncyness	Unknown for this tool.	
Count	Unknown for this tool.	
Draw_last	Unknown for this tool.	
Fadeintime	Unknown for this tool.	
Focus	Unknown for this tool.	
Frames	Unknown for this tool.	
Lifetime	Unknown for this tool.	
Particle_time	Unknown for this tool.	
Physics	Unknown for this tool.	
Size	Unknown for this tool.	
Size_speed	Unknown for this tool.	
Speed	Unknown for this tool.	
Tail_length	Unknown for this tool.	
Turbulence_size	Unknown for this tool.	
Turbulence_speed	Unknown for this tool.	
Gravity	Unknown for this tool.	XYZ
Rotation	Unknown for this tool.	XYZ

CTFFlag: flag for CTF games.

Parameter	Explanation	Range
Team	Defines colour of the team the flag belongs to, i.e. red or blue.	Text

CurrentMission: used in ASSAULT games to indicate to player what their next objective is.

Parameter	Explanation	Range
MissionTextAttacker	Sets the text the attacker will see in their HUD for the mission (flag).	Text
MissionTextDefender	Sets the text the defender will see in their HUD for the mission (flag).	Text
RadarBeacon	Determines if there is a beacon objective on the radar.	T/F
StartSoundAttacker	Defines the sound the attacker will hear upon activating the objective.	File
StartSoundDefender	Defines the sound the defender will hear upon activating the objective.	File

HealingPoint: a point on the map that heals players incrementally.

Parameter	Explanation	Range

Amount	Defines the amount of healing to be given each tick.	
Amount2	Unknown for this tool.	
AwakePhysics	Unknown for this tool.	
FadeTime	Unknown for this tool.	
PlayerOnly	Determines if the healing is for the player only.	T/F
RespawnTime	Defines the respawn time of the player attached to the point.	
ShowFloatingIcon	Unknown for this tool.	T/F
Team	Defines colour of the team the point belongs to, i.e. red or blue.	Text

Phoenix: a tool for making an entity respawn.

Parameter	Explanation	Range
RespawnTime	Defines the time before the entity attached can respawn.	
WithRespawnCycle	Determines if the entity will respawn at the same time as player.	T/F

UnitHighLight: UNKNOWN TOOL

Others Folder

AICrate: a crate that the AI can move around.

Parameter	Explanation	Range
Density	Defines density of the object.	
Mass	Defines the crate's mass.	
Model	Defines the 3D model used by the crate.	File

AnimObject: UNKNOWN TOOL

Parameter	Explanation	Range
AlwaysUpdate	Unknown for this tool.	T/F
Animation	Unknown for this tool.	Text
Model	Unknown for this tool.	File
Physicalize	Unknown for this tool.	T/F
Playing	Unknown for this tool.	T/F
Attachmet1	Unknown for this tool.	
Object	Unknown for this tool.	File
BoneName	Unknown for this tool.	Text
Attachmet2	Unknown for this tool.	
Object	Unknown for this tool.	File
BoneName	Unknown for this tool.	Text
Attachmet3	Unknown for this tool.	
Object	Unknown for this tool.	File
BoneName	Unknown for this tool.	Text
Attachmet4	Unknown for this tool.	
Object	Unknown for this tool.	File
BoneName	Unknown for this tool.	Text

BasicEntity: used in making an object physicalized.

Parameter	Explanation	Range
AIAction	Defines the job that the AI will perform.	List
AnchorRadius	Defines the radius which will act upon the AI.	
AnimStart	Unknown for this tool.	File
AnimEnd	Unknown for this tool.	File
Model	Defines the 3D model for the object.	File
DamagePlayer	Unknown for this tool.	
Animation	Defines the animation parameters for the object.	
Animation	Name of the animation.	Text
Loop	Whether the animation loops or not.	T/F
Playing	Is the animation playing.	T/F
Speed	The speed at which the animation plays.	
Physics	Defines the physical parameters of the object.	
ActivateOnDamage	Whether the physics will be activated on damage caused to object.	T/F
Density	Density of object.	
FixedDamping	Unknown for this tool.	
Impulse	Defines the direction and amount of impulse given to object upon AddImpulse event signal being received.	XYZ
Mass	The mass of the object.	
Resting	Unknown for this tool.	T/F
RigidBody	Whether the object has a rigid body.	T/F
RigidBodyActive	Unknown for this tool.	T/F
Type	Unknown for this tool.	
Damping	The damping to be applied to the impulse of the object's movement.	

Max_time_step	Unknown for this tool.	
Sleep_speed	Unknown for this tool.	
Water_damping	Unknown for this tool.	
Water_density	Unknown for this tool.	
Water_resistance	Unknown for this tool.	
LowSpec	Defines the parameters applied to the object on low-spec machines.	
Density	Unknown for this tool.	
KeepMassAndWater	Unknown for this tool.	T/F
KeepRigidBody	Unknown for this tool.	T/F
Mass	Unknown for this tool.	
RigidBody	Unknown for this tool.	T/F
Max_time_step	Unknown for this tool.	
Sleep_speed	Unknown for this tool.	
Water_density	Unknown for this tool.	
Water_resistance	Unknown for this tool.	

BreakableObject: can be destroyed and will leave particles on the map for a time.

Parameter	Explanation	Range
BreakImpuls	Unknown for this tool.	
Damage	Defines how much damage is caused by the explosion.	
Explosion	Determines if the object can be exploded or not.	T/F
ExplosionRadius	Defines the range of the explosion.	
Model	Defines the 3D model for the explosion.	File
TriggeredOnlyByExplosion	Determines whether the object will only be triggered by an explosion.	T/F
ImpulsivePressure	Unknown for this tool.	
Rmax	Unknown for this tool.	
Rmin	Unknown for this tool.	
DyingSound	Defines the sound of the object being destroyed.	
Filename	Points to the filename of the sound.	File
InnerRadius	Inner radius of the sound; full volume.	
OuterRadius	Maximum range of sound.	
Volume	Volume of sound.	
Parts	Unknown for this tool.	
Density	Unknown for this tool.	
LifeTime	Unknown for this tool.	
RigidBody	Unknown for this tool.	T/F
Animation	Defines the animation parameters for the object.	
Animation	Name of the animation.	Text
Loop	Whether the animation loops or not.	T/F
Playing	Is the animation playing.	T/F
Speed	The speed at which the animation plays.	
Physics	Defines the physical parameters of the object.	
ActivateOnDamage	Whether the physics will be activated on damage caused to object.	T/F
Density	Density of object.	
FixedDamping	Unknown for this tool.	
Impulse	Defines the direction and amount of impulse given to object upon AddImpulse event signal being received.	XYZ
Mass	The mass of the object.	
Resting	Unknown for this tool.	T/F
RigidBody	Whether the object has a rigid body.	T/F
RigidBodyActive	Unknown for this tool.	T/F
Type	Unknown for this tool.	
Damping	The damping to be applied to the impulse of the object's movement.	
Max_time_step	Unknown for this tool.	
Sleep_speed	Unknown for this tool.	
Water_damping	Unknown for this tool.	
Water_density	Unknown for this tool.	
Water_resistance	Unknown for this tool.	

BuildableObject: can be constructed.
For use on assault mode maps and single players.

Parameter	Explanation	Range
InitialState	Defines the initial state of the object.	
Model_building	Defines the 3D model for the object being built.	File
Model_built	Defines the 3D model of the completed building.	File
Model_damaged	Defines the 3D model of the building when damaged.	File
Model_repair	Defines the 3D model of the building while under repair.	File
Model_unbuilt	Defines the 3D model of the building when not built.	File
Max_builtpoints	Defines how long it takes to build the object.	
Max_hitpoints	Defines the number of hit points the building has before being destroyed.	
Max_repairpoints	Defines how long it takes to repair the object.	

CameraSource: UNKNOWN TOOL

CameraTargetPoint: UNKNOWN TOOL

Capture: SELECTION OF THIS OBJECT CRASHES EDITOR

ChainSwing: physicalized chains that can be attached to other objects.

Parameter	Explanation	Range
AttachTo	Defines name of object to attach chain to.	Text
AttachToPart	Unknown for this tool.	
AttachToPartUp	Unknown for this tool.	
AttachToUp	Unknown for this tool.	Tex t
Awake	Unknown for this tool.	T/F
CheckCollision	Determines if chain checks for collisions with player.	T/F
CheckTerrainCollision	Determines if chain checks for collisions with terrain.	T/F
DetachOnDamage	Determines if chain detaches from object upon being damaged.	T/F
Model	Defines the 3D model for chain.	File
Shootable	Determines if shooting chain will break it.	T/F
Coll_dist	Defines the size of the collision detection box around the chain.	
Damping	Defines how much the chain's impulse is affected by damping.	
Friction	Unknown for this tool.	
Mass	Defines the mass of the chain.	
Material	Unknown for this tool.	Text
Max_time_step	Unknown for this tool.	
Num_ropes	Unknown for this tool.	
Rope_name	Unknown for this tool.	Text
Sleep_speed	Unknown for this tool.	
LowSpec	Defines parameters for low-spec machines	
KeepCollision	Keep collision for low-spec machines.	T/F
Max_time_step	Unknown for this tool.	
Sleep_speed	Unknown for this tool.	
Gravity	Defines the strength of gravity through each axis.	
X	Gravity on the X axis.	
Y	Gravity on the Y axis.	
Z	Gravity on the Z axis.	

DamageArea: an area that will damage player.

Parameter	Explanation	Range
Enabled	Determines if damage is enabled.	T/F
DamageRate	Defines how much damage is dealt to the player.	

DeadBody: an object for dead bodies with physics.

Parameter	Explanation	Range
CollidesWithPlayers	Determines if the object can collide with players.	T/F
Mass	Defines the mass of the dead body.	
Model	Defines the 3D model used by the dead body.	File
PushableByPlayer	Determines if the player can push the dead body around.	T/F
Resting	Unknown for this tool.	T/F
Lying_damping	Unknown for this tool.	
Lying_gravity	Unknown for this tool.	

DestroyableObject: leaves wreckage behind after destruction.

Parameter	Explanation	Range
AllowMeleeDamage	Sets whether the player can inflict damage to object with melee weapons.	T/F
Damage	Defines the damage caused by the object exploding.	
Explosion	Determines whether the object makes an explosion upon destruction.	T/F
ExplosionRadius	Defines the range of explosion.	
ExplosionTable	Unknown for this tool.	Text
Hidden	Determines whether the object is hidden in the game or not.	T/F
Model	Defines the 3D model for the object.	File
ModelDestroyed	Defines the 3D model for the object when destroyed.	File
PlayerDamage	Unknown for this tool.	
PlayerDamageRadius	Unknown for this tool.	
PlayerOnly	Determines if the explosion will damage only the player.	T/F
TriggeredOnlyByExplosion	Determines if the destruction can only be caused by an explosion.	T/F
ImpulsivePressure	Defines the strength of the explosion.	
Rmax	Unknown for this tool.	
Rmin	Unknown for this tool.	
Timer	Unknown for this tool.	

AliveSoundLoop	Defines the parameters of the object sound when not destroyed.	
FileName	Points to the filename of the sound.	File
InnerRadius	Inner radius of the sound; full volume.	
OuterRadius	Maximum range of sound.	
Volume	Volume of sound.	
DeadSoundLoop	Defines the parameters of the object sound when destroyed.	
FileName	Points to the filename of the sound.	File
InnerRadius	Inner radius of the sound; full volume.	
OuterRadius	Maximum range of sound.	
Volume	Volume of sound.	
Animation	Defines the parameters of the object sound when being destroyed.	
FileName	Points to the filename of the sound.	File
InnerRadius	Inner radius of the sound; full volume.	
OuterRadius	Maximum range of sound.	
Volume	Volume of sound.	
Physics	Defines the physical parameters of the object.	
ActivateOnDamage	Whether the physics will be activated on damage caused to object.	T/F
Density	Density of object.	
FixedDamping	Unknown for this tool.	
Impulse	Defines the direction and amount of impulse given to object upon AddImpulse event signal being received.	XYZ
Mass	The mass of the object.	
Resting	Unknown for this tool.	T/F
RigidBody	Whether the object has a rigid body.	T/F
RigidBodyActive	Unknown for this tool.	T/F
Type	Unknown for this tool.	
Damping	The damping to be applied to the impulse of the object's movement.	
Max_time_step	Unknown for this tool.	
Sleep_speed	Unknown for this tool.	
Water_damping	Unknown for this tool.	
Water_density	Unknown for this tool.	
Water_resistance	Unknown for this tool.	

FlagEntity: for use as an animated flag.

Parameter	Explanation	Range
Model_supporter	Defines the 3D model of the flagpole.	File

GameEvent: checkpoint for saving game.

Parameter	Explanation	Range
AllowedDeath	Defines the number of deaths before difficulty change when "auto difficulty" is being used.	
Id	Defines the Game Event's identification number.	
RespawnAtTagpoint	Determines whether player will respawn at his tag point.	T/F
UniqueName	Defines the unique name of the save point.	Text

HeliStatic: an animated gunship that cannot be moved.

Parameter	Explanation	Range
DmgScaleBullet	Unknown for this tool.	
DmgScaleExplosion	Unknown for this tool.	
Mass	Defines the mass of the helicopter.	
RigidBodyActive	Unknown for this tool.	T/F
Trackable	Unknown for this tool.	T/F
Engine_file	Points to the sound file for the engine.	File
Max_health	Defines the amount of health for the helicopter.	
ExplosionParams	Defines the parameters for the helicopter's explosion	
Damage	The amount of damage caused by the explosion.	
ImpulsivePressure	The amount of impulse generated by the explosion.	
Radius	Unknown for this tool.	
RadiusMax	Unknown for this tool.	
RadiusMin	Unknown for this tool.	

Piece: UNKNOWN TOOL

Parameter	Explanation	Range
Behaviour	Unknown for this tool.	List
GroupID	Unknown for this tool.	
SightRange	Unknown for this tool.	
SoundRange	Unknown for this tool.	
GroupHostility	Unknown for this tool.	
SpeciesHostility	Unknown for this tool.	T/F
Trackable	Unknown for this tool.	
Aggression	Unknown for this tool.	

Attackrange	Unknown for this tool.	
Character	Unknown for this tool.	List
Cohesion	Unknown for this tool.	
Commrange	Unknown for this tool.	
Eye_height	Unknown for this tool.	
Forward_speed	Unknown for this tool.	
Horizontal_fov	Unknown for this tool.	
Responsiveness	Unknown for this tool.	
Species	Unknown for this tool.	

ProximityDamage: causes a degree of damage to a player in proximity to the object.

Parameter	Explanation	Range
DamageRate	Defines the damage caused per second.	
DamageType	Unknown for this tool.	
Enabled	Sets proximity damage on or off.	T/F
Height	Defines the height of the damage object.	
Radius	Defines the radius of the damage object.	
ShakeOnly	Sets the object to only shake the player's view.	T/F
ShakeType	Defines the type of shake applied to the view.	
SkipAI	Determines whether AI entities are affected by damage object.	T/F
SkipPlayer	Determines whether players are affected by damage object.	T/F
Trigger	Sets the damage object to trigger once or repeatedly.	T/F

Pusher: gives a push to a physicalized object.

Parameter	Explanation	Range
Enabled	Turns the object on or off.	T/F
Impulse	Defines the amount of "push" given to the object.	
Once	Sets the pusher to work once or repeatedly.	T/F

Radio: used by AI to call for reinforcements.

Parameter	Explanation	Range
Damage	Defines damage caused in explosion area.	
DimX	Unknown for this tool.	
DimY	Unknown for this tool.	
DimZ	Unknown for this tool.	
Explosion	Determines if there is an explosion or not.	T/F
ExplosionEffect	Defines the name of the explosion effect.	Text
ExplosionRadius	Defines the radius of the explosion.	
ExplosionScale	Defines the size of the explosion.	
Model	Points to the file name of the 3D model for the radio.	File
ModelDestroyed	Points to the file name of the 3D model for the radio when destroyed.	File
OnlyAICanTrigger	Determines if only AI entities can trigger the radio alarm.	T/F
Trackable	Unknown for this tool.	T/F
TriggeredOnlyByExplosion	Determines whether the alarm can be triggered only by explosions.	T/F
ImpulsivePressure	Defines the strength of the explosion.	
Rmax	Defines the maximum range of the pressure.	
Rmin	Defines the minimum range of the pressure.	
AliveSoundLoop	Defines the sound of the radio loop when not destroyed.	
FileName	Points to the filename of the sound.	File
InnerRadius	Inner radius of the sound; full volume.	
OuterRadius	Maximum range of sound.	
Volume	Volume of sound.	
DeadParticles	Defines the parameters of the particles used when the radio is destroyed.	
Active	Sets the particles on or off.	T/F
AdditiveBlend	The particle light will blend additively with the environment lighting.	T/F
Bounciness	How much "bounce" the particles will have.	
ChildSpawnPeriod	How long the particles will subside on the screen.	
ColorEnd	Colour at the end of the particle spray.	RGB
ColorStart	Colour at the start of the particle spray.	RGB
Count	Number of particles	
DrawOrder	Unknown for this tool.	
FadeInTime	The fade out time for the particles.	
Focus	Unknown for this tool.	
Frames	The speed that the particles go up.	
LifeTime	How long the particles last.	
LinearSizeSpeed	The speed of the particle will be constant or not.	T/F
Physics	The particles are affected by physics, e.g. wind.	T/F
ShaderName	The name of the particle shader.	Text
Size	Size of the particles.	
SizeSpeed	The speed at which the size will change.	
Speed	The speed at which particles will be born.	

Tail	Unknown for this tool.	
TimeDelay	Time delay between each particle creation.	
Type	Type of particle used.	
Turbulence_size	How much turbulence will affect the particles.	
Turbulence_speed	How quickly turbulence will affect the particles.	
ChildProcess		
Gravity	The direction the particles will travel.	XYZ
Objects	Unknown for this tool.	File
Rotation	Angle and axis of particle rotation, if any.	XYZ
Textures	The different textures of the smoke effects.	File
DeadSoundLoop	Defines the sound of the radio when it is dead.	
FileName	Points to the filename of the sound.	File
InnerRadius	Inner radius of the sound; full volume.	
OuterRadius	Maximum range of sound.	
Volume	Volume of sound.	
DyingSound	Defines the sound of the radio as it is dying	
FileName	Points to the filename of the sound.	File
InnerRadius	Inner radius of the sound; full volume.	
OuterRadius	Maximum range of sound.	
Volume	Volume of sound.	

RaisingWater: for creating an area of rising water.

Parameter	Explanation	Range
Speed	Defines the speed at which the water will rise.	
UpdateTime	Unknown for this tool.	
WaterVolume	Unknown for this tool.	
Height_end	Defines the final height of the water.	
Height_start	Defines the starting height of the water.	

RigidBody: used to create any kind of physicalized rigid body.

Parameter	Explanation	Range
ActivateOnRocketDamage	Sets whether the physics will activate when damaged after hit by a rocket.	T/F
Density	Defines the density of the rigid body.	
Mass	Defines the mass of the rigid body.	
Model	Points to the 3D model of the rigid body.	File
Resting	Unknown for this tool.	T/F
Visible	Determines whether the object is visible in the game or not.	T/F
Damping	Defines the amount of damping put on the object's inertia.	
Max_time_step	Unknown for this tool.	
Sleep_speed	Unknown for this tool.	
Water_damping	Unknown for this tool.	
Water_resistance	Unknown for this tool.	
Impulse	Unknown for this tool.	

Rope: used by AI entities to climb down from buildings and helicopters.

Parameter	Explanation	Range
RetrieveRope	Unknown for this tool.	T/F
DropName	Defines the name of the AI entity that will use the rope.	Text

Rotator: SELECTION OF THIS OBJECT CRASHES EDITOR

ShootTarget: UNKNOWN TOOL

Parameter	Explanation	Range
Density	Unknown for this tool.	
Mass	Unknown for this tool.	
Model	Unknown for this tool.	File
Name	Unknown for this tool.	Text
Physicalized	Unknown for this tool.	T/F

SwingingObject: used to create objects that swing.

Parameter	Explanation	Range
Model	Pointer to 3D model used by the swinging object.	File
Resting	Determines whether the object is swinging on initialisation, or at rest.	T/F
Shootable	Determines whether the object can be shot or not.	T/F
ActivationImpulse	Defines the strength of the first swing upon activation.	
Coll_dist	Defines the size of the collision box around the object.	

Damage_player	Defines the amount of damage caused to a player on contact.	
Damage_scale	Defines the size of the damage caused to the player on contact.	
Damping	Defines how much damping affects the inertia of the swinging object.	
Mass	Sets the mass of the swinging object.	
Material	Names the material used by the swinging object.	Text
Max_time_step	Unknown for this tool.	
Sleep_speed	Unknown for this tool.	
Gravity	Sets the direction the swinging object will be attracted to.	XYZ

SwivilChair: UNKNOWN TOOL

Parameter	Explanation	Range
Model	Points to the 3D model used for the object.	File

TestCloth: soft body object.

Parameter	Explanation	Range
Model	Points to the 3D model for the object.	File
Accuracy	Unknown for this tool.	
Air_resistance	Defines the resistance of air against this object.	
Collision_impulsion	Defines how much impulse is given to the object upon collision.	
Damping	Defines the effect of damping upon the objects inertia.	
Damping_ratio	Unknown for this tool.	
Density	Sets the density of the object.	
Explosion_scale	Unknown for this tool.	
Friction	Defines the friction level of the object.	
Impulse_scale	Unknown for this tool.	
Mass	Sets the mass of the object.	
Max_iters	Unknown for this tool.	
Max_safe_step	Unknown for this tool.	
Max_time_step	Unknown for this tool.	
Sleep_speed	Unknown for this tool.	
Stiffness	Defines the stiffness of the cloth.	
Thickness	Defines the thickness of the cloth.	
Water_resistance	Defines the level of water resistance against the cloth.	
Gravity	Sets the direction of gravity for the object.	XYZ
Wind	Sets the wind direction for the object.	XYZ

TV: an in-game TV.

Can only be destroyed by explosion – must add collision manually.

Parameter	Explanation	Range
Damage	Defines the amount of health the TV has.	
Model	Points to the 3D model of the TV.	File
ModelDestroyed	Points to the 3D model of the TV after destruction.	File
SndRadius	Defines the radius at which the TV can be heard.	
SndVolume	Sets the volume for the TV.	
AliveSoundLoop	Defines the parameters of the object sound when not destroyed.	
FileName	Points to the filename of the sound.	File
InnerRadius	Inner radius of the sound; full volume.	
OuterRadius	Maximum range of sound.	
Volume	Volume of sound.	
DeadSoundLoop	Defines the parameters of the object sound when destroyed.	
FileName	Points to the filename of the sound.	File
InnerRadius	Inner radius of the sound; full volume.	
OuterRadius	Maximum range of sound.	
Volume	Volume of sound.	
Animation	Defines the parameters of the object sound when being destroyed.	
FileName	Points to the filename of the sound.	File
InnerRadius	Inner radius of the sound; full volume.	
OuterRadius	Maximum range of sound.	
Volume	Volume of sound.	

Particle Folder

ParticleEffect: explosions, clouds, etc.

Parameter	Explanation	Range
Active	Determines whether the particle effect is active or not.	T/F
ParticleEffect	Names the shader effect used with the object.	Text
Scale	Sets the size of the effect.	
SpawnPeriod	Defines how long the effect will last.	

UpdateRadius	Defines the expansion rate of the effect.	

ParticleSpray: flames, smoke, waterfalls, etc.

Parameter	Explanation	Range
Active	Sets the particles on or off.	T/F
AdditiveBlend	The particle light will blend additively with the environment lighting.	T/F
Bounciness	How much “bounce” the particles will have.	
ChildSpawnPeriod	How long the particles will subside on the screen.	
ColorEnd	Colour at the end of the particle spray.	RGB
ColorStart	Colour at the start of the particle spray.	RGB
Count	Number of particles	
DrawOrder	Unknown for this tool.	
FadeInTime	The fade out time for the particles.	
Focus	Unknown for this tool.	
Frames	The speed that the particles go up.	
LifeTime	How long the particles last.	
LinearSizeSpeed	The speed of the particle will be constant or not.	T/F
Physics	The particles are affected by physics, e.g. wind.	T/F
ShaderName	The name of the particle shader.	Text
Size	Size of the particles.	
SizeSpeed	The speed at which the size will change.	
Speed	The speed at which particles will be born.	
Tail	Unknown for this tool.	
TimeDelay	Time delay between each particle creation.	
Type	Type of particle used.	
UpdateRadius	Unknown for this tool.	
Turbulence_size	How much turbulence will affect the particles.	
Turbulence_speed	How quickly turbulence will affect the particles.	
ChildProcess	Defines the parameters for the child process.	
Active	Sets the particles on or off.	T/F
AdditiveBlend	The particle light will blend additively with the environment lighting.	T/F
Bounciness	How much “bounce” the particles will have.	
ChildSpawnPeriod	How long the particles will subside on the screen.	
ColorEnd	Colour at the end of the particle spray.	RGB
ColorStart	Colour at the start of the particle spray.	RGB
Count	Number of particles	
DrawOrder	Unknown for this tool.	
FadeInTime	The fade out time for the particles.	
Focus	Unknown for this tool.	
Frames	The speed that the particles go up.	
LifeTime	How long the particles last.	
LinearSizeSpeed	The speed of the particle will be constant or not.	T/F
Physics	The particles are affected by physics, e.g. wind.	T/F
ShaderName	The name of the particle shader.	Text
Size	Size of the particles.	
SizeSpeed	The speed at which the size will change.	
Speed	The speed at which particles will be born.	
Tail	Unknown for this tool.	
TimeDelay	Time delay between each particle creation.	
Type	Type of particle used.	
Gravity	The direction the particles will travel.	XYZ
Rotation	Angle and axis of particle rotation, if any.	XYZ
Gravity	The direction the particles will travel.	XYZ
Objects	Unknown for this tool.	File
Rotation	Angle and axis of particle rotation, if any.	XYZ
SpaceLoopBoxSize	Unknown for this tool.	XYZ
Texture	The different textures of the smoke effects.	File

Pickups Folder

Ammoweaponname : all ammo pick-ups for weaponname weapon.

Parameter	Explanation	Range
Amount1	Defines the amount of ammo given in a pick-up.	
Amount2	Not used for this tool.	
Availability	Unknown for this tool.	
AwakePhysics	Unknown for this tool.	T/F
FadeTime	Unknown for this tool.	
PlayerOnly	Determines if this pick-up is for player only.	T/F
RespawnTime	Defines the amount of time before the pick-up respawns.	
ShowFloatingIcon	Unknown for this tool.	T/F

Armor: armour pick-up.

Parameter	Explanation	Range
Amount1	Defines the amount of armour given in a pick-up.	
Amount2	Not used for this tool.	
Availability	Unknown for this tool.	
AwakePhysics	Unknown for this tool.	T/F
FadeTime	Unknown for this tool.	
PlayerOnly	Determines if this pick-up is for player only.	T/F
RespawnTime	Defines the amount of time before the pick-up respawns.	
ShowFloatingIcon	Unknown for this tool.	T/F

Checkpoint: UNKNOWN TOOL

Parameter	Explanation	Range
Amount2	Unknown for this tool.	
Availability	Unknown for this tool.	
AwakePhysics	Unknown for this tool.	T/F
FadeTime	Unknown for this tool.	
Id	Unknown for this tool.	
PlayerOnly	Determines if this pick-up is for player only.	T/F
RespawnTime	Defines the amount of time before the pick-up respawns.	
ShowFloatingIcon	Unknown for this tool.	T/F

ClassAmmoPickup: used for ASSAULT class ammunition.

Parameter	Explanation	Range
Amount2	Defines the amount of ammo given in a pick-up.	
Availability	Unknown for this tool.	
AwakePhysics	Unknown for this tool.	T/F
FadeTime	Unknown for this tool.	
Model	Points to the 3D model for the pick-up.	File
PlayerOnly	Determines if this pick-up is for player only.	T/F
RespawnTime	Defines the amount of time before the pick-up respawns.	
ShowFloatingIcon	Unknown for this tool.	T/F

Health: health pick-up.

Parameter	Explanation	Range
Amount1	Defines the amount of health given in a pick-up.	
Amount2	Not used for this tool.	
Availability	Unknown for this tool.	
AwakePhysics	Unknown for this tool.	T/F
FadeTime	Unknown for this tool.	
PlayerOnly	Determines if this pick-up is for player only.	T/F
RespawnTime	Defines the amount of time before the pick-up respawns.	
ShowFloatingIcon	Unknown for this tool.	T/F

KeyCardn: pickups for keycard n.

Parameter	Explanation	Range
Amount2	Unknown for this tool.	
Availability	Unknown for this tool.	
AwakePhysics	Unknown for this tool.	T/F
FadeTime	Unknown for this tool.	
KeyNumber	Sets the ID of the key.	
Model	Points to the 3D model for the key card pick-up.	File
PlayerOnly	Determines if this pick-up is for player only.	T/F
RespawnTime	Defines the amount of time before the pick-up respawns.	
ShowFloatingIcon	Unknown for this tool.	T/F
Sound	Points to the sound file for the pick-up.	File

Pickupweaponname: pick-up for weaponname.

Parameter	Explanation	Range
Amount1	Defines the amount of ammo given for this weapon’s primary attack.	
Amount2	Defines the amount of ammo given for this weapon’s secondary attack.	
Availability	Unknown for this tool.	
AwakePhysics	Unknown for this tool.	T/F
FadeTime	Unknown for this tool.	
PlayerOnly	Determines if this pick-up is for player only.	T/F
RespawnTime	Defines the amount of time before the pick-up respawns.	
ShowFloatingIcon	Unknown for this tool.	T/F

PickupGeneric: object for creating other pick-ups.

Parameter	Explanation	Range
Amount2	Defines the amount of ammo given for this weapon's secondary attack.	
Availability	Unknown for this tool.	
AwakePhysics	Unknown for this tool.	T/F
FadeTime	Unknown for this tool.	
Message	Details the message to be displayed when player picks up object.	Text
Model	Points to the 3D model of the object.	File
Objects	Details the kind of object to be picked up.	Text
PlayerOnly	Determines if this pick-up is for player only.	T/F
RespawnTime	Defines the amount of time before the pick-up respawns.	
ShowFloatingIcon	Unknown for this tool.	T/F
Sound	Points to the sound file to be played upon pick-up.	

Player Folder

Player: copy of the player entity.

Parameter	Explanation	Range
HasArmor	Determines if the player entity has armour or not.	T/F
HelmetOnStart	Sets whether the player entity has a helmet on start-up.	T/F
Trackable	Unknown for this tool.	T/F
Eye_sight	Unknown for this tool.	
GroupID	The group number of the player entity.	
Max_health	Defines the health points of the player.	
Species	Defines the species number of the entity for the player entity, for when determining enemy status for the AI.	

Spectator: spectator point.

Rename spectator, without capitals or additional text of any kind.

Render Folder

Bfly: BlackFly object.

Parameter	Explanation	Range
BflyNumber	Defines the number of blackfly.	

EnvColor: environment colour for an area.

Parameter	Explanation	Range
Color	Defines the new environment colour.	

Fog: add fog to an area.

Parameter	Explanation	Range
Color	Defines the colour of the fog.	
EndDist	Defines the point at which the fog will end.	
StartDist	Defines the distance from the player at which the fog will start.	
XSkyEnd	Unknown for this tool.	
XSkyStart	Unknown for this tool.	

Grasshopper: grasshopper object.

Parameter	Explanation	Range
CGF1	Points to the 3D model for grasshopper.	
CGF2	Points to the 3D model for grasshopper.	
CGF3	Points to the 3D model for grasshopper.	
CGF4	Points to the 3D model for grasshopper.	
GrasshopperNumber	Defines the number of grasshoppers for the object.	

Storm: adds rain to an area.

Parameter	Explanation	Range
DistanceFromTerrain	Defines the altitude that the rain starts to appear.	
RainAmount	Defines the amount of rain.	
RandomFrequency	Sets the random seed for the rain.	
SoundDistortionTime	Unknown for this tool.	

VWindDir	Has no changeable value.	

ViewDist: changes the view distance for an area.

Parameter	Explanation	Range
MaxViewDistance	Sets the new maximum view distance for an area.	

Sound Folder *NO DESCRIPTIONS FOR ANY OF THESE OBJECTS*

EAXArea

Parameter	Explanation	Range
EAXEnvironment	Unknown for this tool.	
EaxReverbProperties	EAX reverb parameters.	
AirAbsorptionHF	Unknown for this tool.	
DecayHFRatio	Unknown for this tool.	
DecayLFRatio	Unknown for this tool.	
DecayTime	Unknown for this tool.	
Density	Unknown for this tool.	
Diffusion	Unknown for this tool.	
EchoDepth	Unknown for this tool.	
EchoTime	Unknown for this tool.	
EnvDiffusion	Unknown for this tool.	
EnvSize	Unknown for this tool.	
Environment	Unknown for this tool.	
Flags	Unknown for this tool.	
HFReference	Unknown for this tool.	
LFReference	Unknown for this tool.	
ModulationDepth	Unknown for this tool.	
ModulationTime	Unknown for this tool.	
Reflections	Unknown for this tool.	
ReflectionsDelay	Unknown for this tool.	
Reverb	Unknown for this tool.	
ReverbDelay	Unknown for this tool.	
Room	Unknown for this tool.	
RoomHF	Unknown for this tool.	
RoomLF	Unknown for this tool.	
RoomRolloffFactor	Unknown for this tool.	
fReflectionsPan	Unknown for this tool.	XYZ
fReverbPan	Unknown for this tool.	XYZ

EAXPresetArea2

Parameter	Explanation	Range
EAXPreset	Unknown for this tool.	List
OffWhenLeaving	Unknown for this tool.	T/F

MissionHint2

Parameter	Explanation	Range
AllowedToSkip	Unknown for this tool.	
Enabled	Unknown for this tool.	T/F
Loop	Unknown for this tool.	T/F
Once	Unknown for this tool.	T/F
SkipAcknowledge	Unknown for this tool.	File
Volume	Unknown for this tool.	
Hints	Hint soundfiles.	Files

MusicMoodSelector2

Parameter	Explanation	Range
Mood	Unknown for this tool.	

MusicThemeSelector2

Parameter	Explanation	Range
DefaultMood	Unknown for this tool.	
IndoorOnly	Unknown for this tool.	T/F
Mood	Unknown for this tool.	
OutdoorOnly	Unknown for this tool.	T/F
Theme	Unknown for this tool.	

RandomAmbientSound1

Parameter	Explanation	Range
EAXEnvironment	Unknown for this tool.	
LIndoorOnly	Unknown for this tool.	T/F
LoutdoorOnly	Unknown for this tool.	T/F
Scale	Unknown for this tool.	
EaxReverbProperties	EAX reverb parameters.	
AirAbsorptionHF	Unknown for this tool.	
DecayHFRatio	Unknown for this tool.	
DecayLFRatio	Unknown for this tool.	
DecayTime	Unknown for this tool.	
Density	Unknown for this tool.	
Diffusion	Unknown for this tool.	
EchoDepth	Unknown for this tool.	
EchoTime	Unknown for this tool.	
EnvDiffusion	Unknown for this tool.	
EnvSize	Unknown for this tool.	
Environment	Unknown for this tool.	
Flags	Unknown for this tool.	
HFReference	Unknown for this tool.	
LFReference	Unknown for this tool.	
ModulationDepth	Unknown for this tool.	
ModulationTime	Unknown for this tool.	
Reflections	Unknown for this tool.	
ReflectionsDelay	Unknown for this tool.	
Reverb	Unknown for this tool.	
ReverbDelay	Unknown for this tool.	
Room	Unknown for this tool.	
RoomHF	Unknown for this tool.	
RoomLF	Unknown for this tool.	
RoomRolloffFactor	Unknown for this tool.	
fReflectionsPan	Unknown for this tool.	XYZ
fReverbPan	Unknown for this tool.	XYZ
Soundn	Defines parameters for sound n.	
Centered	Unknown for this tool.	T/F
ChanceOfOccurring	Unknown for this tool.	
DoNotOverlap	Unknown for this tool.	T/F
Sound	Unknown for this tool.	File
Volume	Unknown for this tool.	

RandomAmbientSoundPreset

Parameter	Explanation	Range
LIndoorOnly	Unknown for this tool.	T/F
LOutdoorOnly	Unknown for this tool.	T/F
Once	Unknown for this tool.	T/F
PlayFromCenter	Unknown for this tool.	T/F
Scale	Unknown for this tool.	
SoundPreset	Unknown for this tool.	List

SoundExclusive1

Parameter	Explanation	Range
Soundn	Defines parameters for sound n.	
AreaID	Unknown for this tool.	
Centered	Unknown for this tool.	T/F
ChanceOfOccuring	Unknown for this tool.	
DoNotOverlap	Unknown for this tool.	T/F
Sound	Unknown for this tool.	File
Volume	Unknown for this tool.	

SoundSpot

Parameter	Explanation	Range
Enabled	Unknown for this tool.	T/F
FadeValue	Unknown for this tool.	
InnerRadius	Unknown for this tool.	
Loop	Unknown for this tool.	T/F
Once	Unknown for this tool.	T/F
OuterRadius	Unknown for this tool.	
Play	Unknown for this tool.	T/F

Source	Unknown for this tool.	File
Volume	Unknown for this tool.	

Triggers Folder

AITrigger: triggers a change in AI behaviour when entered by an AI entity, similar to an anchor.

Parameter	Explanation	Range
AIAction	Defines the action the AI will perform when entering trigger.	
AILadder	Allows the AI to climb a ladder.	T/F
AnchorRadius	Radius of effect for this trigger.	
DimX	Sets the size of the X axis for trigger.	
DimY	Set the size of the Y axis for trigger.	
DimZ	Sets the size of the Z axis for trigger.	
Enabled	Sets the trigger active or not.	T/F
ExitDelay	Defines how long it takes for the AI entity to leave the trigger after activating it.	
Model	Points to the 3D model used by the trigger.	File
SkipSpecialAI	Unknown for this tool.	T/F
ToggleStance	Unknown for this tool.	T/F
TriggerOnce	Sets the trigger to be activated once or repeatedly.	T/F
Signal	Defines the parameters for signalling another AI entity.	
Readibility	Unknown for this tool.	T/F
SendSignal	Determines whether a signal will be sent or not.	T/F
SignalRadius	Defines the radius within which all other AI entities will receive the signal.	
SignalText	Details the system text sent as a signal.	Text

AreaTrigger: triggers an event when a player enters the attached area shape.

Parameter	Explanation	Range
Enabled	Sets the trigger on or off.	T/F
ScriptCommand	Details any special script to be run on being triggered.	Text
TriggerOnce	Sets the trigger to be activated once or repeatedly.	T/F

BoatTrampolineTrigger: trigger for making boat jump that works similarly to pusher object.

Parameter	Explanation	Range
DimX	Sets the size of the X axis for trigger.	
DimY	Set the size of the Y axis for trigger.	
DimZ	Sets the size of the Z axis for trigger.	
Enabled	Sets the trigger active or not.	T/F
ImpulseDuration	Defines how long the push will last.	
ImpulseFadeInTime	Defines how much the push will degrade over time.	
ImpulseStrength	Sets the initial impulse strength of the push.	
KillOnTrigger	Unknown for this tool.	T/F
MaxAngleOfImpact	Sets the maximum angle of the push.	
MinSpeed	Sets the minimum speed the boat must be travelling in order to activate the trigger.	
PlaySequencce	Names a sequence to play upon activation.	Text
ScriptCommand	Details a special LUA script to add.	Text
TriggerOnce	Sets the trigger to be activated once or repeatedly.	T/F

DelayTrigger: trigger with a time-delay before activation.

Parameter	Explanation	Range
Delay	Sets the amount of time that the trigger is delayed by.	
Enabled	Sets the trigger to be activated once or repeatedly.	T/F
PlaySequence	Names a sequence to play upon activation.	Text
ScriptCommand	Details a special LUA script to add.	Text
TriggerOnce	Sets the trigger to be activated once or repeatedly.	T/F

Impulse Trigger: gives a push to an object by adding impulse.

Parameter	Explanation	Range
DimX	Sets the size of the X axis for trigger.	
DimY	Set the size of the Y axis for trigger.	
DimZ	Sets the size of the Z axis for trigger.	
Enabled	Sets the trigger active or not.	T/F
ImpulseDuration	Defines how long the push will last.	
ImpulseFadeInTime	Defines how much the push will degrade over time.	
ImpulseStrength	Sets the initial impulse strength of the push.	
KillOnTrigger	Unknown for this tool.	T/F
OnlyAI	Determines if the trigger only affects AI.	T/F
OnlyMyPlayer	Determines if the trigger only affects the player currently under control; for mp/sp.	T/F

OnlyPlayer	Determines if the trigger affects all players; for mp/sp.	T/F
TriggerOnce	Sets the trigger to be activated once or repeatedly.	T/F

MultipleTrigger: trigger that will work only a certain number of times before deactivating.

Parameter	Explanation	Range
Enabled	Sets the trigger active or not.	T/F
NumInputs	Defines the number of times the trigger will work before deactivation.	
PlaySequence	Names a sequence to play upon activation.	Text
ScriptCommand	Details a special LUA script to add.	Text

PlaceableExplo: a trigger for placeable explosives.

Parameter	Explanation	Range
Active	Determines if the explosion will trigger or not.	T/F
AutomaticPlaceable	Sets the explosion to be placed automatically.	T/F
DimX	Sets the size of the X axis for trigger.	
DimY	Set the size of the Y axis for trigger.	
DimZ	Sets the size of the Z axis for trigger.	
ExplosionEffect	Names the explosion effect to be used.	Text
ExplosionScale	Defines the size of the explosion .	
InitiallyVisible	Determines if the bomb placeholder will be visible.	T/F
Model	Points to a 3D model of the bomb before destruction.	File
ModelDestroyed	Points to a 3D model of the bomb after destruction.	File
PlaySequence	Names a sequence to play upon activation.	Text
ScriptCommand	Details a special LUA script to add.	Text
TextInstruction	Details the text instruction to the player when near the bomb placeholder.	Text
Countdown	Sets the countdown for after the bomb is placed.	
DummyModel	Points to the 3D model of the bomb placeholder.	Text
ExplDamage	Defines the amount of damage caused by the explosion.	
ExplImpulsive_Pressure	Defines the impulsive impact of the explosion.	
ExplRadius	Defines the radius of the explosion.	
ExplRmax	Defines the maximum range of the impact.	
ExplRmin	Defines the minimum range of the impact.	

PlaceableGeneric: template trigger for other placeable objects.

Parameter	Explanation	Range
Active	Determines if the explosion will trigger or not.	T/F
AutomaticPlaceable	Sets the explosion to be placed automatically.	T/F
DimX	Sets the size of the X axis for trigger.	
DimY	Set the size of the Y axis for trigger.	
DimZ	Sets the size of the Z axis for trigger.	
InitiallyVisible	Determines if the bomb placeholder will be visible.	T/F
Model	Points to a 3D model of the bomb before destruction.	File
ModelDestroyed	Points to a 3D model of the bomb after destruction.	File
PlaceableObject	Names the object that will be placed	Text
PlaySequence	Names a sequence to play upon activation.	Text
ScriptCommand	Details a special LUA script to add.	Text
DummyModel	Points to the 3D model of the bomb placeholder.	Text

ProximityTrigger: activated by proximity to the trigger object.

Parameter	Explanation	Range
AIAction	Defines the action the AI will perform when entering trigger.	
ActivateWithUseButton	Determines whether the trigger will be activated by a use button.	T/F
AnchorRadius	Radius of effect for this trigger.	
DimX	Sets the size of the X axis for trigger.	
DimY	Set the size of the Y axis for trigger.	
DimZ	Sets the size of the Z axis for trigger.	
Enabled	Sets the trigger active or not.	T/F
EnterDelay	Defines how long it takes after the player/entity has entered the trigger before it is activated.	
ExitDelay	Defines how long it takes for the AI to leave the trigger after activating it.	
InVehicleOnly	Determines if trigger will only work if entity or player is in a vehicle.	T/F
KillOnTrigger	Unknown for this tool.	T/F
OnlyAI	Determines if the trigger only affects AI.	T/F
OnlyMyPlayer	Determines if the trigger only affects the player currently under control; for mp/sp.	T/F
OnlyPlayer	Determines if the trigger affects all players; for mp/sp.	T/F
OnlySpecialAI	Unknown for this tool.	T/F
ScriptCommand	Details a special LUA script to add.	Text
TextInstruction	Details the text instruction to the player when near the bomb placeholder.	Text
TriggerOnce	Sets the trigger to be activated once or repeatedly.	T/F

VisibilityTrigger: will trigger upon being seen by the player.

Parameter	Explanation	Range
DimX	Sets the size of the X axis for trigger.	
DimY	Set the size of the Y axis for trigger.	
DimZ	Sets the size of the Z axis for trigger.	
Distance	Defines how far away player before he can see the trigger.	
Enabled	Sets the trigger active or not.	T/F
EnterDelay	Defines how long it takes after the player/entity has entered the trigger before it is activated.	
ExitDelay	Defines how long it takes for the AI to leave the trigger after activating it.	
PlaySequence	Names a sequence to play upon activation.	Text
ScriptCommand	Details a special LUA script to add.	Text
TextInstruction	Details the text instruction to the player when near the bomb placeholder.	Text
TriggerOnce	Sets the trigger to be activated once or repeatedly.	T/F
UseKey	Unknown for this tool.	T/F

Vehicles Folder

Bigtrack: large utility vehicle.

Parameter	Explanation	Range
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the vehicle's group.	
Sightrange	Defines the maximum sight range of the vehicle.	
Soundrange	Defines the maximum hearing range of the vehicle.	
AbandonedTime	Sets the amount of time before the vehicle is considered abandoned.	
Active	Determines whether the vehicle is considered active or not.	T/F
ApproachPlayer	Determines whether the vehicle will approach player or not.	T/F
DmgScaleAIBullet	Damage modifier for the AI's bullet.	
DmgScaleAIExplosion	Damage modifier for the AI's gun fire explosions.	
DmgScaleBullet	Damage modifier for the vehicle's gun fire.	
DmgScaleExplosion	Damage modifier for the explosions from this vehicle.	
DrawDriver	Unknown for this tool.	T/F
GroupHostility	Defines the hostility towards the group the vehicle belongs to.	
LightsOn	Sets the vehicle's lights on or off.	T/F
LimitLRAngle	Defines limit of the player's left/right mouse look angle when inside vehicle.	
LimitUDMaxAngle	Defines limit of the maximum up/down mouse look angle when inside vehicle.	
LimitUDMinAngle	Defines limit of the minimum up/down mouse look angle when inside vehicle.	
Model	Points to 3D model of the vehicle.	File
Pathloop	Determines if the helicopter will loop at the end of its path.	T/F
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SpeciesHostility	Unknown for this tool.	
StartDelay	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Usable	Determines whether the vehicle can be used by the player or not.	T/F
UsePathFind	Sets whether the vehicle will use pathfinding AI.	T/F
Aggression	Defines aggression level of the AI.	
Attackrange	Sets the range the player must be within before the vehicle will give chase.	
Bodypos	Unknown for this tool.	
Character	Names the AI script the vehicle will use.	List
Cohesion	Unknown for this tool.	
Commrange	Defines the communication range of the vehicle.	
Damage_player	Defines how much damage is inflicted upon player when rammed.	
Eye_height	Unknown for this tool.	
Forward_speed	Sets the forward speed of the vehicle.	
Hit_upward_velocity	Unknown for this tool.	
Horizontal_fov	Sets the horizontal field of view for the vehicle.	
Max_health	Sets the maximum hit points for the vehicle.	
Pathname	Names the path the vehicle will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
PointBackOff	Names the tag point the vehicle will retreat to when called.	Text
PointReinforce	Names the tag point the vehicle will move to when requested to reinforce.	Text
Responsiveness	Defines the ease of handling of the vehicle.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
Vertical_fov	Defines the vertical field of view for the vehicle.	
AICarDef	Parameters for AI vehicle control.	
AI_use	Unknown for this tool.	T/F
Damping_vehicle	Unknown for this tool.	
Dyn_friction_ratio	Unknown for this tool.	
Handbraking_value	Unknown for this tool.	
Max_braking_friction	Unknown for this tool.	
Max_steer_v0	Unknown for this tool.	

Steer_relaxation_v0	Unknown for this tool.	
Steer_speed	Unknown for this tool.	
Steer_speed_min	Unknown for this tool.	
ExplosionParams	Defines the parameters for the vehicle's explosion when destroyed.	
Damage	Damage caused by explosion.	
ImpulsivePressure	Impulse of explosion.	
Radius	Radius of explosion.	
RadiusMax	Maximum range of impact.	
RadiusMin	Minimum range of impact.	

Boat: gun boat.

Parameter	Explanation	Range
Usable	Determines if the vehicle can be used by the player or not.	T/F
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the vehicle's group.	
Sightrange	Defines the maximum sight range of the vehicle.	
Soundrange	Defines the maximum hearing range of the vehicle.	
AISoundRadius	Unknown for this tool.	
AbandonedTime	Sets the amount of time before the vehicle is considered abandoned.	
Active	Determines whether the vehicle is considered active or not.	T/F
ApproachPlayer	Determines whether the vehicle will approach player or not.	T/F
DmgScaleAIBullet	Damage modifier for the AI's bullet.	
DmgScaleAIExplosion	Damage modifier for the AI's gun fire explosions.	
DmgScaleBullet	Damage modifier for the vehicle's gun fire.	
DmgScaleExplosion	Damage modifier for the explosions from this vehicle.	
DrawDriver	Unknown for this tool.	T/F
DriverName	Names the AI entity that will drive the boat.	Text
GroupHostility	Defines the hostility towards the group the vehicle belongs to.	
LightsOn	Sets the vehicle's lights on or off.	T/F
LimitLRAngle	Defines limit of the player's left/right mouse look angle when inside vehicle.	
LimitUDMaxAngle	Defines limit of the maximum up/down mouse look angle when inside vehicle.	
LimitUDMinAngle	Defines limit of the minimum up/down mouse look angle when inside vehicle.	
Name	Points to 3D model of the vehicle.	File
Persistence	Unknown for this tool.	
SameGroupID	Unknown for this tool.	T/F
SetInvestigate	Sets the vehicle to go investigate when activated.	T/F
SpeciesHostility	Unknown for this tool.	
StartDelay	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
UsePathFind	Sets whether the vehicle will use pathfinding AI.	T/F
UseRL	Unknown for this tool.	T/F
UseRLGuided	Unknown for this tool.	T/F
Accuracy	Sets the shooting accuracy of the AI.	
Aggression	Sets the aggression of the AI.	
Attackrange	Sets the range within which the target must come before the AI will start shooting.	
Bodypos	Unknown for this tool.	
Character	Names the AI script the vehicle will use.	List
Cohesion	Unknown for this tool.	
Commrange	Defines the communication range of the vehicle.	
Damage_player	Defines how much damage is inflicted upon player when rammed.	
Eye_height	Unknown for this tool.	
Forward_speed	Sets the forward speed of the vehicle.	
Horizontal_fov	Sets the horizontal field of view for the vehicle.	
Max_health	Sets the maximum hit points for the vehicle.	
Pathname	Names the path the vehicle will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
PointBackOff	Names the tag point the vehicle will retreat to when called.	Text
PointReinforce	Names the tag point the vehicle will move to when requested to reinforce.	Text
Responsiveness	Defines the ease of handling of the vehicle.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
Vertical_fov	Defines the vertical field of view for the vehicle.	
WaterDamping	Unknown for this tool.	
Water_resistance	Defines the level of water resistance against the boat.	
Water_sleep_speed	Unknown for this tool.	
AttackParams	Defines the attack parameters for the boat.	
Horizontal_fov	Horizontal field of view.	
Sightrange	Sight range.	
ExplosionParams	Defines the parameters for the vehicle's explosion when destroyed.	
Damage	Damage caused by explosion.	
ImpulsivePressure	Impulse of explosion.	
Radius	Radius of explosion.	
RadiusMax	Maximum range of impact.	
RadiusMin	Minimum range of impact.	

GunnerParams	Defines parameters for the boat's gunner.	
AttackRange	How close the gunner must be to target before it will start firing.	
Horizontal_fov	Horizontal field of view for the gunner.	0-360
Responsiveness	Responsiveness of the turret to enemy movement.	
Sightrange	Sight range of the gunner.	

BoatPatrol

Parameter	Explanation	Range
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the vehicle's group.	
Sightrange	Defines the maximum sight range of the vehicle.	
Soundrange	Defines the maximum hearing range of the vehicle.	
AbandonedTime	Sets the amount of time before the vehicle is considered abandoned.	
ApproachPlayer	Determines whether the vehicle will approach player or not.	T/F
DmgScaleBullet	Damage modifier for the vehicle's gun fire.	
DmgScaleExplosion	Damage modifier for the explosions from this vehicle.	
DrawDriver	Unknown for this tool.	T/F
GroupHostility	Defines the hostility towards the group the vehicle belongs to.	
LimitLRAngle	Defines limit of the player's left/right mouse look angle when inside vehicle.	
LimitUDMaxAngle	Defines limit of the maximum up/down mouse look angle when inside vehicle.	
LimitUDMinAngle	Defines limit of the minimum up/down mouse look angle when inside vehicle.	
Name	Points to 3D model of the vehicle.	File
Pathloop	Determines if the helicopter will loop at the end of its path.	T/F
Persistence	Unknown for this tool.	
SpeciesHostility	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Usable	Determines whether the vehicle can be used by the player or not.	T/F
UsePathFind	Sets whether the vehicle will use pathfinding AI.	T/F
Aggression	Defines aggression level of the AI.	
Attackrange	Sets the range the player must be within before the vehicle will give chase.	
Bodypos	Unknown for this tool.	
Character	Names the AI script the vehicle will use.	List
Cohesion	Unknown for this tool.	
Commrange	Defines the communication range of the vehicle.	
Damping	Unknown for this tool.	
Eye_height	Unknown for this tool.	
Horizontal_fov	Sets the horizontal field of view for the vehicle.	
Max_health	Sets the maximum hit points for the vehicle.	
Pathname	Names the path the vehicle will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
PointBackOff	Names the tag point the vehicle will retreat to when called.	Text
PointReinforce	Names the tag point the vehicle will move to when requested to reinforce.	Text
Responsiveness	Defines the ease of handling of the vehicle.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
Vertical_fov	Defines the vertical field of view for the vehicle.	
WaterDamping	Unknown for this tool.	
Water_resistance	Defines the level of water resistance against the boat.	
Water_sleep_speed	Unknown for this tool.	
ExplosionParams	Defines the parameters for the vehicle's explosion when destroyed.	
Damage	Damage caused by explosion.	
ImpulsivePressure	Impulse of explosion.	
Radius	Radius of explosion.	
RadiusMax	Maximum range of impact.	
RadiusMin	Minimum range of impact.	

Buggy: light four wheeled vehicle.

Parameter	Explanation	Range
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the vehicle's group.	
Sightrange	Defines the maximum sight range of the vehicle.	
Soundrange	Defines the maximum hearing range of the vehicle.	
AbandonedTime	Sets the amount of time before the vehicle is considered abandoned.	
Active	Determines whether the vehicle is considered active or not.	T/F
ApproachDist	Unknown for this tool.	
ApproachPlayer	Determines whether the vehicle will approach player or not.	T/F
DrawDriver	Unknown for this tool.	T/F
GroupHostility	Defines the hostility towards the group the vehicle belongs to.	
LightsOn	Sets the vehicle's lights on or off.	T/F
LimitLRAngle	Defines limit of the player's left/right mouse look angle when inside vehicle.	
LimitUDMaxAngle	Defines limit of the maximum up/down mouse look angle when inside vehicle.	
LimitUDMinAngle	Defines limit of the minimum up/down mouse look angle when inside vehicle.	
Pathloop	Determines if the helicopter will loop at the end of its path.	T/F

Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SameGroupID	Unknown for this tool.	T/F
SetInvestigate	Sets the vehicle to go investigate when activated.	T/F
SpeciesHostility	Unknown for this tool.	
StartDelay	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Usable	Determines whether the vehicle can be used by the player or not.	T/F
UsePathFind	Sets whether the vehicle will use pathfinding AI.	T/F
Aggression	Defines aggression level of the AI.	
Attackrange	Sets the range the player must be within before the vehicle will give chase.	
Bodypos	Unknown for this tool.	
Character	Names the AI script the vehicle will use.	List
Cohesion	Unknown for this tool.	
Commrange	Defines the communication range of the vehicle.	
Damage_player	Defines how much damage is inflicted upon player when rammed.	
Eye_height	Unknown for this tool.	
Forward_speed	Sets the forward speed of the vehicle.	
Hit_upward_velocity	Unknown for this tool.	
Horizontal_fov	Sets the horizontal field of view for the vehicle.	
Max_health	Sets the maximum hit points for the vehicle.	
Pathname	Names the path the vehicle will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
PointBackOff	Names the tag point the vehicle will retreat to when called.	Text
PointReinforce	Names the tag point the vehicle will move to when requested to reinforce.	Text
Responsiveness	Defines the ease of handling of the vehicle.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
Vertical_fov	Defines the vertical field of view for the vehicle.	
AICarDef	Parameters for AI vehicle control.	
AI_use	Unknown for this tool.	T/F
Damping_vehicle	Unknown for this tool.	
Dyn_friction_ratio	Unknown for this tool.	
Handbraking_value	Unknown for this tool.	
Max_braking_friction	Unknown for this tool.	
Max_steer_v0	Unknown for this tool.	
Steer_relaxation_v0	Unknown for this tool.	
Steer_speed	Unknown for this tool.	
Steer_speed_min	Unknown for this tool.	
ExplosionParams	Defines the parameters for the vehicle's explosion when destroyed.	
Damage	Damage caused by explosion.	
ImpulsivePressure	Impulse of explosion.	
Radius	Radius of explosion.	
RadiusMax	Maximum range of impact.	
RadiusMin	Minimum range of impact.	

Forklift

Parameter	Explanation	Range
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the vehicle's group.	
Sightrange	Defines the maximum sight range of the vehicle.	
Soundrange	Defines the maximum hearing range of the vehicle.	
AbandonedTime	Sets the amount of time before the vehicle is considered abandoned.	
Active	Determines whether the vehicle is considered active or not.	T/F
ApproachDist	Unknown for this tool.	T/F
DmgScaleAIBullet	Damage modifier for the AI's bullet.	
DmgScaleAIExplosion	Damage modifier for the AI's gun fire explosions.	
DmgScaleBullet	Damage modifier for the vehicle's gun fire.	
DmgScaleExplosion	Damage modifier for the explosions from this vehicle.	
DrawDriver	Unknown for this tool.	T/F
GroupHostility	Defines the hostility towards the group the vehicle belongs to.	
LightsOn	Sets the vehicle's lights on or off.	T/F
LimitLRAngle	Defines limit of the player's left/right mouse look angle when inside vehicle.	
LimitUDMaxAngle	Defines limit of the maximum up/down mouse look angle when inside vehicle.	
LimitUDMinAngle	Defines limit of the minimum up/down mouse look angle when inside vehicle.	
Model	Points to 3D model of the vehicle.	File
Pathloop	Determines if the helicopter will loop at the end of its path.	T/F
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SpeciesHostility	Unknown for this tool.	
StartDelay	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Usable	Determines whether the vehicle can be used by the player or not.	T/F
UsePathFind	Sets whether the vehicle will use pathfinding AI.	T/F

Aggression	Defines aggression level of the AI.	
Attackrange	Sets the range the player must be within before the vehicle will give chase.	
Bodypos	Unknown for this tool.	
Character	Names the AI script the vehicle will use.	List
Cohesion	Unknown for this tool.	
Commrange	Defines the communication range of the vehicle.	
Damage_player	Defines how much damage is inflicted upon player when rammed.	
Damage_scale	Sets the damage modifier.	
Eye_height	Unknown for this tool.	
Forward_speed	Sets the forward speed of the vehicle.	
Hit_upward_velocity	Unknown for this tool.	
Horizontal_fov	Sets the horizontal field of view for the vehicle.	
Max_health	Sets the maximum hit points for the vehicle.	
Pathname	Names the path the vehicle will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
PointBackOff	Names the tag point the vehicle will retreat to when called.	Text
PointReinforce	Names the tag point the vehicle will move to when requested to reinforce.	Text
Responsiveness	Defines the ease of handling of the vehicle.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
Vertical_fov	Defines the vertical field of view for the vehicle.	
AICarDef	Parameters for AI vehicle control.	
AI_use	Unknown for this tool.	T/F
Damping_vehicle	Unknown for this tool.	
Dyn_friction_ratio	Unknown for this tool.	
Handbraking_value	Unknown for this tool.	
Max_braking_friction	Unknown for this tool.	
Max_steer_v0	Unknown for this tool.	
Steer_relaxation_v0	Unknown for this tool.	
Steer_speed	Unknown for this tool.	
Steer_speed_min	Unknown for this tool.	
ExplosionParams	Defines the parameters for the vehicle's explosion when destroyed.	
Damage	Damage caused by explosion.	
ImpulsivePressure	Impulse of explosion.	
Radius	Radius of explosion.	
RadiusMax	Maximum range of impact.	
RadiusMin	Minimum range of impact.	

Humvee

Parameter	Explanation	Range
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the vehicle's group.	
Sightrange	Defines the maximum sight range of the vehicle.	
Soundrange	Defines the maximum hearing range of the vehicle.	
AbandonedTime	Sets the amount of time before the vehicle is considered abandoned.	
Active	Determines whether the vehicle is considered active or not.	T/F
ApproachDist	Unknown for this tool.	
ApproachPlayer	Determines whether the vehicle will approach player or not.	T/F
AttackStickDist	Unknown for this tool.	
DisabledMessage	Sets text message to be displayed when vehicle is disabled.	
DrawDriver	Unknown for this tool.	T/F
GroupHostility	Defines the hostility towards the group the vehicle belongs to.	
LightsOn	Sets the vehicle's lights on or off.	T/F
LimitLRAngle	Defines limit of the player's left/right mouse look angle when inside vehicle.	
LimitUDMaxAngle	Defines limit of the maximum up/down mouse look angle when inside vehicle.	
LimitUDMinAngle	Defines limit of the minimum up/down mouse look angle when inside vehicle.	
LockUser		
Pathloop	Determines if the helicopter will loop at the end of its path.	T/F
Persistence	Unknown for this tool.	
ReinforcePoint	Name of tag point used by AI for reinforcement event.	Text
SameGroupID	Unknown for this tool.	T/F
SetInvestigate	Sets the vehicle to go investigate when activated.	T/F
Sleeping	Unknown for this tool.	T/F
SpeciesHostility	Unknown for this tool.	
StartDelay	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
Usable	Determines whether the vehicle can be used by the player or not.	T/F
UsePathFind	Sets whether the vehicle will use pathfinding AI.	T/F
Aggression	Defines aggression level of the AI.	
Attackrange	Sets the range the player must be within before the vehicle will give chase.	
Character	Names the AI script the vehicle will use.	List
Cohesion	Unknown for this tool.	
Commrange	Defines the communication range of the vehicle.	
Damage_player	Defines how much damage is inflicted upon player when rammed.	

Eye_height	Unknown for this tool.	
Forward_speed	Sets the forward speed of the vehicle.	
Hit_upward_velocity	Unknown for this tool.	
Horizontal_fov	Sets the horizontal field of view for the vehicle.	
Max_health	Sets the maximum hit points for the vehicle.	
Pathname	Names the path the vehicle will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
PointBackOff	Names the tag point the vehicle will retreat to when called.	Text
PointReinforce	Names the tag point the vehicle will move to when requested to reinforce.	Text
Responsiveness	Defines the ease of handling of the vehicle.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
AICarDef	Parameters for AI vehicle control.	
AI_use	Unknown for this tool.	T/F
Damping_vehicle	Unknown for this tool.	
Dyn_friction_ratio	Unknown for this tool.	
Handbraking_value	Unknown for this tool.	
Max_braking_friction	Unknown for this tool.	
Max_steer_v0	Unknown for this tool.	
Steer_relaxation_v0	Unknown for this tool.	
Steer_speed	Unknown for this tool.	
Steer_speed_min	Unknown for this tool.	
ExplosionParams	Defines the parameters for the vehicle's explosion when destroyed.	
Damage	Damage caused by explosion.	
ImpulsivePressure	Impulse of explosion.	
Radius	Radius of explosion.	
RadiusMax	Maximum range of impact.	
RadiusMin	Minimum range of impact.	
GunnerParams	Defines parameters for the boat's gunner.	
AttackRange	How close the gunner must be to target before it will start firing.	
Horizontal_fov	Horizontal field of view for the gunner.	0-360
Responsiveness	Responsiveness of the turret to enemy movement.	
Sightrange	Sight range of the gunner.	

Paraglider: unpowered vehicle.

Parameter	Explanation	Range
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the vehicle's group.	
Sightrange	Defines the maximum sight range of the vehicle.	
Soundrange	Defines the maximum hearing range of the vehicle.	
AbandonedTime	Sets the amount of time before the vehicle is considered abandoned.	
DmgScaleAIBullet	Modifier to damage caused by AI bullet.	
DmgScaleAIExplosion	Modifier to damage caused by AI weapon explosions.	
DmgScaleBullet	Modifier to damage caused by player bullet.	
DmgScaleExplosion	Modifier to damage caused by player weapon explosions.	
LimitLRAngle	Defines limit of the player's left/right mouse look angle when inside vehicle.	
LimitUDMaxAngle	Defines limit of the maximum up/down mouse look angle when inside vehicle.	
LimitUDMinAngle	Defines limit of the minimum up/down mouse look angle when inside vehicle.	
Name	Points to 3D model of vehicle.	File
Trackable	Unknown for this tool.	T/F
UserPassenger	Unknown for this tool.	T/F
Damping	Unknown for this tool.	
Water_damping	Unknown for this tool.	
Water_resistance	Unknown for this tool.	

Zodiac: small inflatable boat.

Parameter	Explanation	Range
Behaviour	Sets the behaviour for the unit.	List
Groupid	Sets the ID number of the vehicle's group.	
Sightrange	Defines the maximum sight range of the vehicle.	
Soundrange	Defines the maximum hearing range of the vehicle.	
AISoundRadius		
AbandonedTime	Sets the amount of time before the vehicle is considered abandoned.	
Active	Determines whether the boat can be used by a player/AI or not.	T/F
DmgScaleAIBullet	Modifier to damage caused by AI bullet.	
DmgScaleAIExplosion	Modifier to damage caused by AI weapon explosions.	
DmgScaleBullet	Modifier to damage caused by player bullet.	
DmgScaleExplosion	Modifier to damage caused by player weapon explosions.	
DrawDriver	Unknown for this tool.	T/F
GroupHostility	Defines the hostility towards the group the vehicle belongs to.	
LimitLRAngle	Defines limit of the player's left/right mouse look angle when inside vehicle.	
LimitUDMaxAngle	Defines limit of the maximum up/down mouse look angle when inside vehicle.	
LimitUDMinAngle	Defines limit of the minimum up/down mouse look angle when inside vehicle.	

Name	Points to 3D model of the vehicle.	File
Persistence	Unknown for this tool.	
ReinforcePoint	Unknown for this tool.	
SpeciesHostility	Unknown for this tool.	
Trackable	Unknown for this tool.	T/F
UserPassenger		T/F
Accuracy	Defines the accuracy of the AI.	
Aggression	Defines aggression level of the AI.	
Attackrange	Sets the range the player must be within before the vehicle will give chase.	
Bodypos	Unknown for this tool.	
Character	Names the AI script the vehicle will use.	List
Cohesion	Unknown for this tool.	
Commrange	Defines the communication range of the vehicle.	
Damping	Unknown for this tool.	
Eye_height	Unknown for this tool.	
Forward_speed	Defines the forward movement speed of the vehicle.	
Horizontal_fov	Sets the horizontal field of view for the vehicle.	
Max_health	Sets the maximum hit points for the vehicle.	
Pathname	Names the path the vehicle will follow.	Text
Pathstart	The number of the first path node.	
Pathsteps	The number of steps involved in the path.	
PointBackOff	Names the tag point the vehicle will retreat to when called.	Text
PointReinforce	Names the tag point the vehicle will move to when requested to reinforce.	Text
Responsiveness	Defines the ease of handling of the vehicle.	
Species	Sets the species number for this AI, for use when calculating the hostility of other AIs.	
Vertical_fov	Defines the vertical field of view for the vehicle.	
WaterDamping	Unknown for this tool.	
Water_resistance	Defines the level of water resistance against the boat.	
ExplosionParams	Defines the parameters for the vehicle's explosion when destroyed.	
Damage	Damage caused by explosion.	
ImpulsivePressure	Impulse of explosion.	
Radius	Radius of explosion.	
RadiusMax	Maximum range of impact.	
RadiusMin	Minimum range of impact.	

Weapons Folder

Special effects for special projectile weapons – no additional properties detailed.

BUTTON TagPoint

Respawn: used as a spawn point for the player.
Numbered object, where number = target savepoint.

TagPoint: generic target point for various game functions.

Comment: inactive object used for leaving text messages in the map.

Parameter	Explanation	Range
Comment	Details the text of the comment.	Text
Fixed	Unknown for this tool.	T/F

Appendix B

Events List

A more detailed examination of all the events in the editor at time of documentation.

Input/Output Events

Generic events: common to many objects.

Event	Explanation
Activate	Activates an entity.
AddImpulse	Adds a physical impulse as a vector (x, y, z) to the object.
Awake	Used to activate dead bodies. They get updated by the physics and will start moving.
Deactivate	Deactivates an entity.
Enable	Enables an Entity (input event).
Enter	Object's bounding box entered.
Explode	Object explodes.
Hide	Object becomes hidden.
IsDead	Object is dead (cannot signal from On Die, so must use this).
Picked	Pickup got picked up
Reset	Reset to default state.
Spawn	Player has spawned or objective has been captured.
Trigger	
Use	
Unhide	Object becomes unhidden.

AI Folder

Soldier/Mutant/NPC/Animal

Event	Explanation
AcceptSound	Entity will accept a sound spot to be redirected to the AI. AI will have Lipsync if lipsync files are generated and the sound will come from this spot.
Die	Entity is ordered to die.
DisablePhysics	Disable the physics for the Entity – no bounding box anymore. This event was requested for "Valerie" to perform a special animation. AI cannot move with disabled physics.
EnablePhysics	Enable physics - bounding box is back and player will bounce.
Follow	Special behaviour that makes AI follow the player. Will only work if AI has species "0", i.e. the same as the player.
GoDumb	AI will behave as if there is no enemy is around. Used to make AI make certain actions, for example to approach an anchor and perform a certain animation.
HalfHealthLeft	Entity is reduced to half health.
HoldSpot	Entity is ordered to hold position. AI will try to get to "SPECIAL_HOLD_SPOT" anchor.
LastGroupMemberDied	Every AI from this group got killed. There is no AI in this map anymore with the same group ID. Used to trigger other events. Note: every single AI in the group must trigger that event.
Lead	AI will lead, special case for Valerie to lead the player to special points.
MakeVulnerable	AI becomes vulnerable.
OnDeath	Triggered after the AI died, can be used to trigger other events
Relocate	AI can be relocated to TagPoints. Must use tag point naming convention: [name]_RELOCATE .
Ressurect	AI will be resurrected.
SPECIAL_ANIM_START	A "special" animation has started. Only tested with Valerie and "PLANT_BOMB_HERE" anchor.
StopSpecial	AI will stop any behaviour.

Aircraft

Event	Explanation
Fly	Aircraft is ordered to fly.
GoAttack	Aircraft is ordered to go into attack mode. Will try to get to the tag point defined in the pointAttack parameter or attack any target (player or different species).
GoPath	Aircraft is ordered to follow path.
GoPathULTIMATE	
IsDead	Event after the entity got killed.
Kill	Entity is ordered to die. Will explode.
Land	Aircraft is ordered to land. Will land at once and turn of engine.
LoadPeople	AI will get inside the aircraft. AI needs to have same group ID as aircraft.
LowHelath	
Reinforcement	Aircraft is ordered to reinforce point. All AI in the same group will try to enter aircraft.

Doors Folder

Doors

Event	Explanation
Close	Door is in the process of closing.
Closed	Door is fully closed.
ForceClose	Door will close even something is blocking it, like dead bodies.
Open	Door is in the process of opening.
Opened	Door is fully open.
Unlocked	If a door is locked it can be forced to unlock, by a trigger or event. Will stay unlocked.

Elevator Folder

Elevator

Event	Explanation
Close	Elevator is in the process of returning to original position.
Closed	Elevator has returned to original position.
ForceClose	
Open	Elevator is in the process of moving to next position.
Opened	Elevator has reached next position.
RestartAnimation	
StartAnimation	

Lights Folder

DynamicLight

Event	Explanation
Shake	Light is shaken - triggers the shake event of the phys light model.
SwitchMaterial1	This can be used to change materials for the light model, i.e. the geometry of the light. It will try to switch to the material with the original material name + 1.
SwitchMaterial2	This can be used to change materials for the light model, i.e. the geometry of the light. It will try to switch to the material with the original material name + 1.
SwitchToMaterialOriginal	This can be used to change materials for the light model, i.e. the geometry of the light. It will try to switch to the material with the original material.

Multiplayer Folder

ASSAULTCheckPoint

Event	Explanation
AttackerTouch	
Averted	
Blocked	
Capturing	Check point is being captured.
Spawn	Check point has been captured.
Touched	
Untouched	
Warmup	

CAHFlag Unused Multiplayer Mode

Event	Explanation
Blue	
Neutral	

Red	

Others Folder

AnimObject

Event	Explanation
HideAttached	
ShowAttached	
StartAnimation	Start the animation
StopAnimation	Stop the animation

BasicEntity

Event	Explanation
Activate	Activates an entity.
AddImpulse	Adds an physical impulse as a vector (x,y,z) to the object.
Hide	Hide object.
ResetAnimation	
StartAnimation	
StopAnimation	
SwitchToMaterial1	Used to change material. Will switch to material with number 1.
SwitchToMaterial2	Used to change material. Will switch to material with number 2.
SwitchToMaterialOriginal	Used to change material. Will switch to original material.
Unhide	Unhide object.

BuildableObject

Event	Explanation
building	Object is building.
built	Object is completed.
damaged	Object has been damaged.
hidden	Object is hidden.
repair	Object is being repaired.
unbuilt	Object is not built.

ChainSwing

Event	Explanation
ChainBroken	Chain's attachment to object is broken.

DestroyableObject

Event	Explanation
Explode	Object explodes.
OnDamage	
SwitchToMaterial1	
SwitchToMaterial2	
SwitchToMaterialOriginal	

GameEvent

Event	Explanation
Save	Trigger to save game.

Pusher

Event	Explanation
Push	Object is given impulse.

RaisingWater

Event	Explanation
RaiseWater	Start to raise or lower the water volume.
WaterStopped	Output trigger if the water volume has stopped rising, or falling.

RigidBody

Event	Explanation
OnTouch	

Rope

Event	Explanation
DoRope	

TV

Event	Explanation
Off	
On	
OnDamage	

Particle Folder

ParticleEffect

Event	Explanation
Pulse	Pulses the particle effect.

Pickups Folder

Pickups

Event	Explanation
Picked	Pickup is collected by player.

Keycard

Event	Explanation
KeyCardPickup	Keycard is collected by the player.
Picked	

Sound Folder

MissionHint

Event	Explanation
Play	Will play the sound.
Stop	

MusicMoodSelector

Event	Explanation
ResetDefaultMood	
SetDefaultMood	
SetMood	

MusicThemeSelector

Event	Explanation
SetTheme	

SoundSpot

Event	Explanation
Play	Will play the sound.
Redirect	Redirect the sound to an AI. AI needs to "Accept" the sound to work.
Stop	

Triggers Folder

All Triggers

Event	Explanation
Enter	Trigger’s bounding box is entered.
Leave	Trigger’s bounding box is exited.

AITrigger

Event	Explanation
Signal	

DelayTrigger/MultipleTrigger

Event	Explanation
InputTrigger	Trigger this event to activate the delay or trigger something on this event.
OutputTrigger	Triggered after the delay finished.

PlaceableExplo

Event	Explanation
DeActivateAndHide	
ExplosivePlaced	Triggered after explosive got placed.

PlaceableGeneric

Event	Explanation
ExplosivePlaced	Triggered after explosive is placed.

VisibilityTrigger

Event	Explanation
Invisible	Trigger is invisible to player.
Visible	Trigger is visible to player.

Vehicles Folder

Land Vehicles

Event	Explanation
Abandonded	Vehicle is abandoned.
AIDriverIn	AI driver ordered to enter vehicle.
AIDriverOut	AI drives ordered to exit vehicle.
AIEntered	Triggered whenever AI enters the vehicle.
DisableHumvee	Disable Humvee. Player can not use it anymore.
DriverIn	
EnableHumvee	Enable Humvee.
EveryoneOut	
GoChase	Vehicle is ordered to enter chase player stance.
GoPath	Vehicle is ordered to follow path.
GoPatrol	Vehicle is ordered to patrol path.
Grenade	
KillTriger	Destroy vehicle. Vehicle will explode.
MakePlayerGunner	
OnDeath	Triggered after vehicle is destroyed.
PathEnd	Vehicle reached destination.
PausePath	Vehicle will pause on the path. You need to cause event trigger GoPath again.
PlayerEntered	Triggered whenever player entered the vehicle.
Reinforcement	Vehicle is ordered to reinforce point.
Wakeup	

Sea Vehicles

Event	Explanation
Abandonded	Vehicle is abandoned.
AddPlayer	
DriverIn	
GoAttack	Vehicle is ordered to enter attack mode.
GoPath	Vehicle is ordered to follow path.
GoPatrol	Vehicle is ordered to patrol path.
KilTriger	Destroy vehicle. Will explode.
Load	Order AI with same group AI into the vehicle.
OnDeath	Triggered after vehicle is destroyed.

Reinforcement	Vehicle is ordered to reinforce point.
Release	
StartAniPath	
StopAttack	
TArgetOnLand	

Paraglider

Event	Explanation
Abandonded	Vehicle is abandoned.
DriverIn	
OnDeath	

Index