ABUSE™

FOR MACINTOSH

Abuse is a trademark of Crack dot Com.

Thank You

We would like to thank you for purchasing Abuse for the Macintosh. Bungie is committed to providing the highest quality software products and would appreciate any feedback you might have to make our games even better.

Customer Support

As our customer, you are our most important asset. If you have any questions about how to use or install this software, please get in touch with us through one of the channels listed below. We are more than happy to help you out.

Bungie Software Products Corporation
Customer Support
PO Box 7877
Chicago, IL 60680-7877
http://www.bungie.com

Telephone: (312) 255-9600
Facsimile: (312) 397-0502

If you want to buy our cool Bungie Stuff, call 1-800-295-0060 or e-mail sales@bungie.com.

Email:
Technical Support: macsupport@bungie.com
General Info: info@bungie.com
Sales: sales@bungie.com

Introduction

You are Nick Vrenna. It is the year 2009. You have been wrongly incarcerated inside a high security underground prison where illegal genetic experiments are taking place. Alan Blake, the head research scientist, has isolated the specific gene which causes violence and aggression in humans. This genetic sequence, called “Abuse” is highly infectious, causing horrific transformations and grotesque side-effects.

You are the only person to show immunity to it. A prison riot erupts, and in the confusion all the cell doors are opened. Soon everyone, guards and convicts alike, become infected and transform into a horde of mutants which take over the building. Your only chance for escape is to don battle armor and reach the Control Room situated in the structure’s deepest level. You must first stop the prison’s Abuse-infected water supply from contaminating the outside world.

Freedom and the fate of the world now depend on you.

System Requirements

Abuse requires a Macintosh Computer or compatible with a 68040 or better processor, 13” 256 color monitor, System 7.1 or higher, and 6 megabytes of available RAM.

Abuse has an option to fill the screen if you have a monitor larger than 640 X 480. If you have a Macintosh with a system installed prior to System 7.5.3 and a monitor larger than 640 X 480, you must install Display Enabler (found in the “For System Folder” folder) to get the full screen option.

On some systems prior to System 7.5, you will not get any game sounds if you do not have Sound Manager 3.2.1 or higher installed. The most recent version of Sound Manager comes with Quicktime which is available at www.quicktime.com.

Installation

Copy the Abouse folder from the Sack CD to your hard drive.

Game Options

After you launch the game, you'll be presented with the Main Menu. Use the mouse to select the following options.

Load Game

This option is only available after you have saved your position during a game. To load a saved game, click on the Load Game icon. On the left of the Load Game screen are up to five save areas. Click on one of the save areas to restore your game to that location.

Start New Game

Choose this option when you want to begin a single-player game from the start. You will still be able to access previously saved game positions.

Difficulty Levels

There are four levels of difficulty in Abuse: Wimp, Easy, Normal, and Extreme. The harder the level, the more difficult it is to kill your enemies, and the easier it is for them to kill you. Change the difficulty level by clicking on this box.

Gamma Correction

Abuse is meant to be a dark game. To play the game as it is designed, select the darkest shade of gray that you can disern. It should be one step lighter than black. You may, of course, adjust the gamma correction any way you prefer. Save your gamma changes by clicking the red "check" button.

Volume Control

You can adjust the sound effects by clicking on the arrow buttons. Up arrows increase the volume; down arrows decrease it. Save your volume changes by clicking the check button or press Escape to cancel any changes.

Networking

To play Abuse over the network you must have at least two Macintosh computers physically connected via LocalTalk or Ethernet on an AppleTalk network. The networking screen allows you to create new network games or join in if one is currently available.

Start A New Net Game. Click this button to start a new net game, then fill out the network options to begin a new game. Press the Escape key to return to the Main menu screen.

Join Existing Game. If there is a game available or in progress, you can join into that game by clicking on it. Then fill out your name and hit the check mark to join into the game.

Exit Net Game. Exits you from your current game so you can join another one. If you're not currently in a game, this option will not appear.

Screen Options

The screen options in this menu allow you to set the various screen modes. Rounded and Smoothed Pixel modes offer the best graphics but require a Power Macintosh or better for best performance. Double, single, and scan skip mode are intended for slower Macintoshes. Click on the screen until you find a screen mode to your liking.

Key Settings

Key settings allows you to customize your keyboard settings. Use the mouse to aim and fire with the mouse button.

The default keyboard settings are:

i = move up

j = move left

l = move right

k = use/enter/down

Command = use special power

u = toggle weapons counter-clockwise

o = toggle weapons clockwise

Keys 1 through 7 activate the corresponding weapon slot.

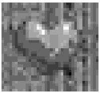
Credits

Find out who made Abuse!

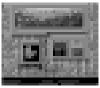
Quit

Press this button to exit the game.

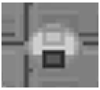
Environment



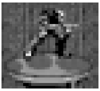
Health. These restore hit points.



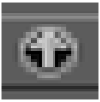
Stations. Stations are places where you can save games. Stand in front of the station and press the down arrow to activate it. To save your position, click on any of the save screens on the left.



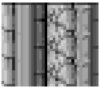
Switches & Doors. To activate a switch, press the down arrow. Sometimes you'll need to use more than one switch to remove an obstacle.



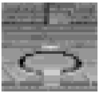
Moving Platforms. Press the down arrow to activate a moving platform. If you use either side-step keys, you will step off the platform.



Jump Enhancers. Jump toward a jump enhancer; when you hit it for the first time, your momentum temporarily increases. An enhancer needs to recharge before it will work again.

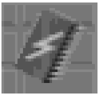


Destroyable Walls. Some walls will crumble and reveal secret rooms. Usually weak walls show cracks or signs of damage, but not always. When destroyable walls are hit, the weapon's fire will terminate with a red glow. When a solid wall is hit, fire terminates with a white glow.

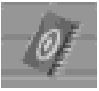


Teleporters. There are two types of teleporters: Local and Level. You use both by pressing the down arrow. Local teleporters send you to a different area of the same level while Level exit teleporters send you to a new level.

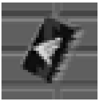
Compass. Only available in network play. When you've acquired the Compass, the map view shows the locations of all non-cloaked players.



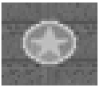
Flash Speed. This power-up increases your speed. Use the Command Key to use this power-up.



Cloak. Only available in network play. It makes you almost (but not quite) invisible to other players, and it shields you from appearing on large-scale maps. Use the Command key to use this power-up.

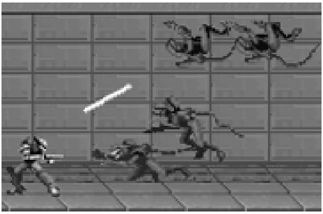


Anti Grav Boots. This power-up gives you the ability to fly. “Hey, Johnny, look at me, I can fly!”

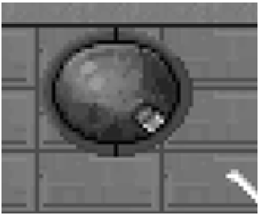


Ultra Health. With this power-up you can accumulate up to 200 hit points, instead of the usual 100 points.

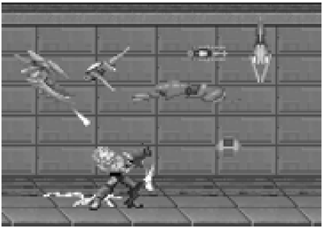
Enemies & Obstacles



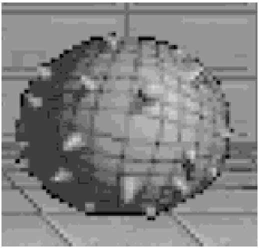
Mutants



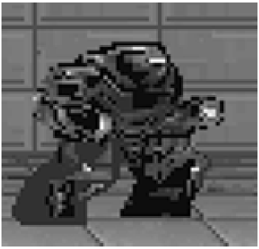
Turret



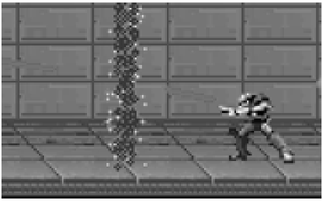
Fliers



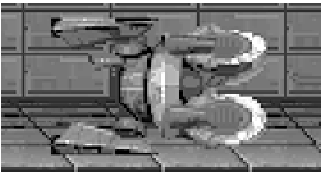
Boulder



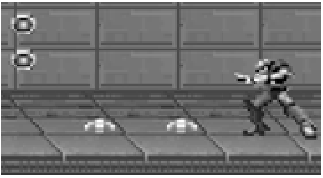
Jugger



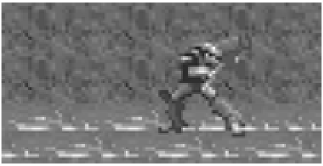
Force Field



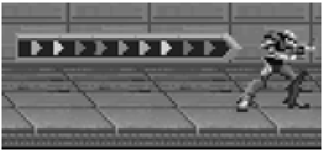
Drillbot



Land & Air Mines



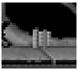
Lava



Pusher Arrow

Weapons & Ammo

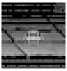
Laser



5 Ammo

20 Ammo

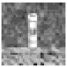
Incendiary Grenade Launcher



2 Ammo

20 Ammo

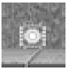
Heat Seeking Rocket Launcher



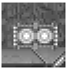
2 Ammo

5 Ammo

Napalm

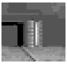


1 Ammo

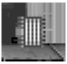


5 Ammo

Energy Rifle



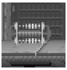
20 Ammo



50 Ammo

Nova Spheres

4 Ammo



10 Ammo

Death Saber

50 Ammo

100 Ammo

ABUSE™

FOR MACINTOSH

Level Editor

Abuse Editor Product Support: Due to the complexity of the level editor, Bungie will not be able to answer questions about the Editor's use or level creation. Some of the features of this editor require a two-button mouse which is not natively supported on the Macintosh. Currently, there are no workarounds that will allow these features to work. We appologize for this inconvenience.

Introduction

One of the unique features of Abuse is its Level Editor. The Editor allows you to actually build new levels and even entire new games of Abuse, in both single and multi-player mode.

This manual will teach you the basics of level creation — how to create a map using foreground and background tiles, how to place objects on the map, and how to link objects so that they can affect one another.

Using the Editor, you have access to all the tricks, traps, menaces and puzzles that exist in the game itself, and the freedom to configure them into something totally new. Have fun.

What Do I Need to Know?

Do you need programming experience to use the Editor? Absolutely not. While programmers will find the level Editor simple and intuitive, and many of the terms defined in this manual self-explanatory, even casual gamers can use the Editor to build new levels and games.

Do not be afraid to experiment. There is very little that you can do with the Editor that might interfere with your game or system. Many concepts that seem strange and difficult the first time you read about them will turn out to be extremely easy and intuitive when you actually try to do them.

What Can I Do With My Creations?

So now that you've created this great new level, what can you do with it? You can play it yourself all you want, but of course you really want to share it with the world.

Abuse, the Abuse engine and the Abuse Editor are all the property of Crack dot Com. This means that any levels you create with the Abuse Editor are also the property of Crack dot Com. However, it is Crack dot Com's policy to allow you to publicly distribute any new level you design. The only condition attached to this is that you must distribute the level free of charge. Therefore, you can take your level and make copies for friends, put it on a network, and even upload it to a BBS, online service or the Internet — just as long as you don't take money or any other kind of payment for it.

This permission applies only to levels you yourself have created using the Editor. Obviously, you cannot distribute the game engine itself or any of the levels that came with it.

Before You Begin

When starting a new level, you should first come up with a general idea of what you want the level to be. Will it be all action, or will it have difficult puzzles? Or simple puzzles? What is the skill level of the player? Levels, in general, can be as large as you want them to be, but the default value of 100 x 100 tiles is large enough for most purposes. Before you can begin putting down tiles and objects, however, you first must learn how to navigate the windows and menus.

Starting the Editor

1. Go to Screen Options in the Main Menu and click on "Edit."

2. Start a new game. You'll be dropped into edit mode.

The map for Level 0, the training level, will appear as the default. The game will not be running.

Notice that a menu bar has appeared at the bottom of the screen. Also, your cursor is now an arrow instead of the usual target.

Selecting a menu item once will activate it. Selecting it again will deactivate it. Note that in the menus, there will be an asterisk (*) next to a toggle command when that command is active.

File Menu

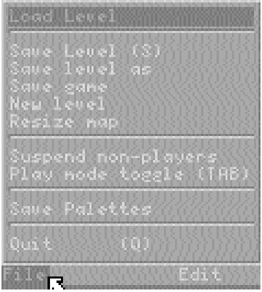
Under file, the leftmost menu, you'll find a listing of file utility commands and some advanced options.

Load Level

Start the Editor as described at the beginning of this chapter. From the menu bar open the file menu, and click on load level. Click on the subdirectory that contains your file, then click on the filename of the level you want. Once you've found the file you want, click on ok to load it. Note that this file selection system will not load files outside the Abuse directory.

Save Level [SHIFT S]

This command saves the level. It uses the same name and directory structure as the currently loaded level. If you have not titled your level before you save this way, it names the level "untitled.spe".



Save Level As

This command saves the currently loaded level under a new name. The first time you save a newly designed game or level, you'll need to use the Save Level As command to establish a filename and directory.

Save Game

This saves the level, including Nick's (the main character's) current position and weapon status. When the level is loaded, the player begins at the position from which the game was saved. The other save commands do not save Nick's current position and weapon status.

New Level

This brings up a blank level, ready to be designed.

Resize Map

This command resizes the map. The default is 100 x 100 tiles. A tile is 30 x 15 pixels, so in pixel terms this translates into a 3000 x 1500 pixel world. To change the size of the world, type in the new parameters (in tiles) and click on ok.

Suspend Non-Players

This stops all objects, except Nick, from carrying out instructions or becoming active. Notice that when you execute this command Nick's upper body is frozen. This is because he is composed of two objects and only the lower half remains active. This mode is useful in level designing, when you don't want to disturb the level as you roam about.

Play Mode Toggle [TAB]

This allows you to play the game normally without leaving edit mode. Once you select this command the only Editor command you can use is the short-cut key t, which will return you to edit mode and re-enable all the other Editor commands.

Save Palettes

Palettes are windows in which you can place an assortment of foreground tiles in any arrangement. Once a palette is created, it can be used to quickly fill entire regions of the map. A full explanation of palettes and foreground tiles is discussed later

Edit Menu

The options in the Edit Menu are very important to designing levels. Most of the important functions have keyboard shortcuts which can greatly speed up level design. Mastering these commands and their shortcuts will allow you to create levels much more quickly and efficiently.

Toggle Light

This command toggles the light level from the set values to maximum. It helps you see what you're doing while editing dimly-lit areas in the level. (See p. 20 for more on changing light values.)

Set Scroll Rate

The scroll rate modifies the rate at which the foreground layer and background layer scroll relative to each other. The Abuse engine has the capability for two layers of parallaxing, which gives the screen perspective and the illusion of depth. The top layer is called the foreground layer and the bottom called the background layer.

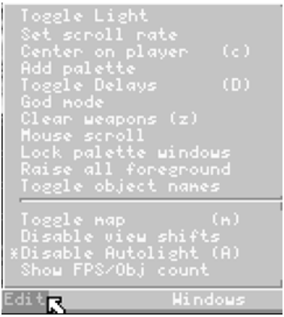
Scroll Rate parameters:

rate for foreground scroll in x axis = X nul #
rate for background scroll in x axis = X div #
rate for foreground scroll in y axis = Y nul #
rate for background scroll in y axis = Y div #

For example, if the parameters were:

X nul 2
X div 1
Y nul 2
Y div 1

the foreground would scroll twice as fast as the background in both the x and y axes. Normally, setting these values so the foreground scrolls twice as fast as the background creates a very realistic sense of relative motion.



Center on Player [C]

This command will center the screen on Nick. Since the Editor allows you (the designer) to move around the level at will, this command is useful to find Nick again when you need him.

Add Palette

You can make your own custom "palettes." Palettes are very powerful tools to a level designer. They allow you to do complex fills and make repetitive structures easily, and serve as custom libraries for tiles. Palettes are your friends.

Palettes are windows in which you can place an assortment of foreground tiles in any arrangement and then use that arrangement to stamp one copy or fill entire regions. To make your palette, put your cursor over a tile you want to copy into the palette and press [~]. Then right-click on the area in your palette box where you want the tile to go.

NOTE: This feature obviously requires a two-button mouse which is not natively supported on the Macintosh. Currently, there are no workarounds that will allow these features to work. We appologize for this inconvenience.

Once you've created the pattern you want, you can hit [SPACEBAR] to place it on the level map. Make sure to save the pattern immediately, or the Editor will forget it the next time you shrink or enlarge the palette box.

Accessing a List of Available Palettes

1. Press [P]. You will need to have the cursor on the level itself, not on another child window. A window pops up listing all current palettes by filename.

2. Double-click on the palette you would like to see.

Unshrinking Palettes

Sometimes the palette windows are shrunk for manageability. To unshrink one, press [+] while the cursor is inside the appropriate window.

Reshrinking Palettes

Reshrink a palette by pressing [-].

Using a Palette

Palettes can be used to fill an area or stamp down a single copy.

1. Open the appropriate palette window.

2. Left-click-and-hold on the palette's window bar (the thick part of the window at the top).

- Use [T] to flood-fill an area. This will fill the tile you're clicking on, plus all identical tiles directly adjacent to the first. It will continue to fill as long as there are identical tiles directly adjacent to the tiles being filled.

- Use [SAPACEBAR] to just stamp out a single copy.
  Position the stamp or fill you have selected so the upper left-hand corner of the stamp is lined up with the upper left-hand corner of the area to be filled. If you click anywhere within the corner tile, the stamp will fill to the upper left corner of that tile, so don't worry about being pixel-perfect when stamping or filling.

Creating a Custom Palette

1. Select the add palette command from the menu.

2. Name your palette something representative of the palette. For your first time something like "test" will do. Notice the parameters for the X and Y size — you can adjust these later, if you choose.

3. Choose the ok button. A palette window pops up in the upper left-hand corner of the screen with the chosen name.

Adding Foreground Tiles

1. Press [F] to bring up the foreground window.

2. Place your cursor inside the foreground window and press [SHIFT F] to expand the foreground window. Multiple presses of [SHIFT F] continue to expand the foreground window until it reaches maximum size; the next press returns it to a single-tile-wide window.

- Use [SHIFT +], with the cursor inside the foreground window, to expand the foreground tiles.

- Use [-], with the cursor inside the foreground window, to shrink foreground tiles. Shrinking palettes is more time-efficient than organizing them by moving them around the screen.

- Press the upper left-hand button on a window to close it.

3. Click on a tile in the foreground window to choose it.

4. Move your cursor over any palette window and right-click.

- If you cannot lay down a tile in a palette window, you might have a protection command activated. Open LOCK PALETTE WINDOWS in the EDIT Menu. If this command selection has an asterisk next to it, then your palette is locked and cannot be modified. Click on this menu item to deactivate it.

Changing a Palette Size

1. Place your cursor inside the palette window.

2. Press the arrow keys.

- [Right Arrow] expands the window's width, [Left Arrow] shrinks the window's width.

- [Down Arrow] expands the window's height, [Up Arrow] shrinks the window's height.

Picking a Tile

1. Move your cursor over a tile.

2. Pick up the tile by left-clicking.

If you right-click without having LOCK PALETTE WINDOWS (below) active, you might alter the palette. If you shrink a palette window, any tiles you've added to the palette since your last save will be lost. LOCK PALETTE WINDOWS protects against accidental alteration of palettes.

You should save each time you add a new palette you want to keep in the game, by using the SAVE PALETTES command from the FILE Menu. Saving a level does not save palette information.

Toggle Delays [SHIFT D]

This command turns off all delays and runs the game at the fastest speed capable for your computer. It's useful if you're designing a level on a machine that's a bit too slow to run it normally. There should be a message telling you whether delays are being turned on or off after you choose this command.

God Mode

In this mode you have all the weapons, unlimited ammo and cannot be killed. In EDIT mode, click on god mode and then hit [Command] and [Tab]. Repeat the process to exit God Mode.

Clear Weapons [SHIFT Z]

When level designing and debugging, it is sometimes necessary to manually reset your weapons, since the Abuse engine does not clear Nick's weapon status when you switch from one level to another. This command will reset weapon status, leaving Nick with only the laser rifle and no extra ammo.

Mouse Scroll

If you wish, you can use the mouse as a scrolling device. When MOUSE SCROLL is active, there is an asterisk beside its name in the EDIT Menu. You can scroll the screen by using the mouse to push the cursor against the four boundaries of the screen.

You can also use [UP ARROW], [DOWN ARROW], [LEFT ARROW] and [RIGHT ARROW] to scroll around the level, but be careful of accidentally altering any unlocked palette windows when using the arrow keys.

Lock Palette Windows

Prevents any accidental alteration to a palette. When this command is active none of the palette windows can be resized nor can the tiles in them be changed.

Raise All Foreground

When you activate this command, all the foreground that is laid down from that moment until you deactivate the command are "raised." Raised foregrounds are drawn in front of Nick, obscuring him.

When in EDIT mode all raised foregrounds have an "X" drawn over them.
To deactivate this command select this command again. To "lower" a raised foreground tile, place the cursor over the tile and press [R]. If this is successful the "X" over the foreground tile should disappear. Pressing [R] again will raise the tile once more.

Toggle Object Names

Turning this on displays the name of all objects on screen in green underneath the object.

Toggle Map [M]

This displays a global representation of the level, with area boxes indicated by yellow boxes, connected by gray lines.

Area boxes. These are locations where Nick's entrance triggers a preset ambient (see below) and screen displacement (below).

Ambient. Ambient refers to the global lighting. An area can only have one ambient at a time. However, with the use of area boxes, multiple ambiances throughout the level can be achieved. Once Nick leaves an area box the screen displacement and ambient do not return to their previous values. The values of the screen displacement and ambient remain constant until Nick enters another area box.

Screen displacement. This shifts the center of the screen relative to Nick's position. These parameters and others can be accessed using the Tool Bar Window. The Tool Bar Window can be activated either using the menu command under the WINDOWS Menu or the short-cut [A].

Disable View Shifts

While this command is activated the view does not shift when Nick is in an area box. This is most useful when you find that entering an area box accidentally causes annoying and unwanted view shifts. To deactivate this, just reselect it under the EDIT Menu.

Disable Autolight [SHIFT A]

Autolight turns off the lighting whenever the frame rate drops below nine frames per second, allowing the game to run faster. Select DISABLE AUTOLIGHT if you're working on a slow machine or want to see the lighting regardless of speed.

Show FPS/Obj Count

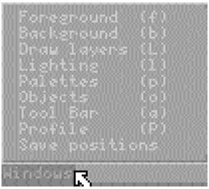
This command shows both the number of frames per second and the number of objects being processed (currently) in the upper left-hand corner of the screen.

Windows Menu

The windows Menu bar takes you to all the important windows. When creating a level, you will be using these windows more than anything else.

Foreground [F]

This selection brings up the Foreground Window where you can select the tiles which Nick interacts with (walks on, is blocked by, etc.). Note that not all foreground tiles can interact with Nick.



The shortcut is [F]; it will toggle the Foreground Window. The color black is drawn as a beige color in the foreground tiles, for identification purposes.

Expanding the Foreground Window

Place your cursor inside it and press [SHIFT F].
Continuing to press [SHIFT F] expands the Foreground Window further, until maximum size is reached; pressing it again returns the window to a single tile-width.

Expanding the Foreground Tiles

[SHIFT +] expands foreground tiles.
[-] shrinks foreground tiles.

Background

This selection brings up the Background Window.

The Abuse engine is capable of two layers of parallaxing. The top layer is called the foreground layer and the bottom called the background layer.

Background tiles, as the name implies, are laid on the background layer. This layer is drawn behind the foreground layer, and will show through only in areas where the foreground tiles are black.

Expanding the Background Tiles

[SHIFT +] expands background tiles.
[-] shrinks background tiles.

Laying Background Tile [B]

1. Select a background tile from the background window by moving your cursor over the background tile and pressing either mouse button.

Note that, unlike foreground tiles, you cannot use [~]to select already-laid-down background tiles.

2. Move the cursor over the location where you want to place the background tile.

3. Right-click to lay down a background tile. Only the right mouse button will work. This feature is not native to the Macintosh. Again, we apologize for this incomptability.

Because of the grid method employed by the Abuse engine, you can only lay down background tiles in a 60x30 pixel grid. It is extremely difficult to lay down background tiles correctly when they are covered by foreground tiles.

Draw Layers [SHIFT L]

You can turn off selective layers, lighting, bounding areas and links by using the Draw Layers Window.

The Draw Layers Window is useful in clearing up the clutter accumulated by developing levels.

For example, when laying down background tiles it is sometimes necessary to turn off the foreground layer. This is done by pressing the button labeled FORE. Pressing FORE again turns the foreground layers back on.

Layer Buttons

FORE	Toggles the foreground layer on/off.
BACK	Toggles the background layer on/off.
BOUND	Toggles the bounding boxes of the foreground tiles on/off.

Bounding boxes block Nick, so he can walk on floors (not fall through them) and is stopped by walls, etc. This mode is useful for determining whether the junctions of the foreground tiles' bounding boxes line up correctly.

LIGHT Toggles the lighting on/off. Lighting is a unique feature of the Abuse engine, allowing you to create new and interesting scenes simply by moving around the light sources. Turning off the lighting is useful when you want speed, as the lighting slows the machine down. It is also useful because developing levels with the lighting on can be distracting. It's usually best to disable the lighting until the very end of the level design process.

LINKS Links are the gray lines drawn from one object to another. They act as visual cues, indicating that those objects are connected to each other for particular purposes (see chapter on LINKS). If the links are so dense as to obscure the foreground, then it is a good idea to turn them off. When the links are toggled off, the light icons do not appear on screen.

Lighting [L]

Lighting is an integral tool in creating mood in the levels. Learn how to use the lighting well, but don't overuse it — lighting uses a good deal of memory. Also remember Abuse is designed to be a dark, moody game.

This selection will bring up the Lighting Window, which shows the various light sources that can be placed in a level. There is also an ambient button (see "Toggle Map" in the "Edit Menu" chapter). Lights are represented by a circular, semi-circular or wedge-shaped icon, depending on the direction of the light produced.

Dropping Lights

1. Move your cursor over one of the light icons in the Lighting Window.

2. Click on it, using either mouse button.

3. Move your cursor over onto the screen. Notice how the light icon is dragged with your cursor.

4. Position the light icon where you want it.

5. Click to drop it in place.

Modifying a Light's Default Size and Shape

1. Position the cursor over the light to be modified.

2. Press [SPACEBAR] to pick up the light. (Obviously, if you've just selected the light and it is already picked up, you don't have to pick it up again.)

3. Use the arrow keys to extend the light's effects in the direction desired, and [SHIFT +] and [-] to increase and decrease the size of the light. You can also increase or decrease the size of the light by clicking on the light icon to open an extended option window, then modifying the R2 field to change the size of the light.

Palettes [P]

This window gives you a listing of your palettes. See the ADD PALETTE command (in the "Edit Menu" chapter) for a more complete explanation of palettes.

Objects [O]

Objects are the core of the game. Everything that interacts with you is an object. Each one is unique, as defined by its artificial intelligence. This window lists all the objects in the game. See the OBJECT LISTINGS AND DESCRIPTIONS chapter for a complete overview of all the objects in the game.

Tool Bar [A]

The Tool Bar is used in conjunction with the map view, to add and modify area boxes and place tiles in the map view. When the Tool Bar is active, you see three icons.

PENCIL ICON When this icon is selected (click on it), you can select any tile from the Foreground Window, Background Window, or even the currently loaded level (left-click) and copy it to anywhere else in the level with another mouse-click.

ARROW ICON This icon allows you to modify an area box's parameters, but only after the box has been created (see "Laying Down and Area Box below).

AREA BOX ICON This icon allows you to create an area box (see "Laying Down and Area Box below).

Laying Down a Tile

- Right-click over an area in the map. (NOTE: This feature is not natively supported on the Macintosh.) This mode is useful for using the fill option.

Filling an Area with One Type of Foreground Tile

1. Select a tile from any of three sources: current level, Foreground Window or Palette Window.

2. Position your cursor over the tile.

3. Press [~].

If your foreground window is open at the same time, it will jump to the tile which you just selected.

4. Find the desired location (which must be composed of the same tile in continuous array).

5. Move the cursor over this area.

6. Press [T] to flood fill the area.

Modifying an Area Box's Parameters

1. Select the arrow icon by double-clicking on one of the area box's yellow endpoints (there are two endpoints per area box: one at the upper left and one at the lower right). When you have selected an area box it will flash yellow. You then get the Area Box extended option window, with these parameters:

AMBIENT The global lighting level. It ranges from 0 - 63.

AMBIENT SPEED The rate at which the ambient changes from one ambient to the preset, if they differ from one another. If an ambient speed of negative one (-1) is specified, the engine uses whatever the current ambient is as the preset.

VIEW X OFF How many pixels Nick's center is offset from the screen's center on the x axis. If the value is negative it will shift the screen left; if the value is positive it will shift the screen right.

VIEW Y OFF How many pixels Nick's center is offset from the screen's center on the y axis. A negative value shifts the screen upwards; a positive value shifts the screen downwards.

VIEW X SPEED The speed at which the screen shifts along the x axis in terms of pixels per second.

VIEW Y SPEED The speed at which the screen shifts along the y axis in terms of pixels per second.

2. Change the parameters to your satisfaction.

3. Click on the check button to close the Area Box extended option window, or the red “X” button to delete the Area Box.

Laying Down an Area Box
1. Select the Area Box icon by positioning your cursor over it and left-clicking.

2. Go to the map.

3. Left-click, hold and drag the mouse. The distance you drag the mouse (down and to the right) defines the area of the box.

Profile [SHIFT P]

The Profile Window displays the current objects and ranks their CPU time. The higher up on the list, the greater the CPU time taken up by that object or group of objects. To use Profile Window you first open the window, then (with your cursor placed outside the Profile Window) press t to start the game running. The Profile Window will remain up as the game runs, listing objects as they become active.

Save Positions

This command will save all open window positions to their current location. When reopened, each will reappear at its current location.

LINKS

Before you can learn about the objects you need to learn about linking. Links are indicated in the Editor by gray lines drawn from one object to another. The lines act as visual cues, indicating that those objects have some sort of interaction. The object at the beginning of the link (the trigger) controls the object at the end of the link (the reactor). The first object is not modified by the link in any way.

An object like a sensor or a switch, for example, has two states — on or off. When the sensor or switch changes its state, the object that is linked to it will activate its AI code and modify its state. Normally when the sensor or switch is on, the object it is linked to is active, and when the sensor or switch is off the linked object is inactive.

Making a Quick Link

1. Put your cursor over the trigger.

2. Press-and-hold [Control].

3. Drag the cursor to the reactor. A gray bounding box should appear.

4. Release [Control]. The link is indicated by a gray line between the two objects.

Note that you cannot link objects to lights with a quick link; you must use a window link to link objects to lights.

Making a Window Link

1. Click on the trigger. An extended option window opens up.

2. Click on the reactor. A gray line appears between the two objects.

Note that the extended window now belongs to the reactor — not the trigger. Thus, if you modify the AI in the window, you are modifying the reactor’s AI.

Making Multiple Links
Some objects can have multiple links. These are noted in their descriptions in the OBJECT LIST.

- To make multiple quick links, just repeat the procedure described above, starting with the trigger each time.

- To make multiple window links, you must first close the reactor's extended options windows before you can make another link from the trigger.

Checking Links
To see whether an object is part of a link, you can use the cursor.

- Move the cursor over the object.

If a link line turns yellow, then you know that the object is linked to another object. If no line changes color then the object has no links.

Unlinking by Using Shortcut Keys
1. Move your cursor over the object making the link.

2a. To break the first link on the object, press [K].

2b. To break the last link on the object, use [SHIFT K].

2c. To break all the links coming from that object, press [K] or [SHIFT K] repeatedly. If links remain, they are not linked to that object.
Note that links from objects to lights can only be broken by using the window method.

Unlinking by Using a Window
1. Open the extended options window by clicking on the object.

- The button on the left, with the red "X", breaks links with objects in order from first to last. There is no way to break links with objects from last to first.

- The button on the right, also with a red "X", breaks links with lights from first to last. As with breaking links with objects, you cannot break links with lights from last to first.

2. Close the extended options window when you are through by clicking on the check button, or click the X-button in the upper left-hand corner to delete the object. Leaving the window open can cause accidental unlinking or linking.

MODIFYING AN OBJECT'S AI

All objects have an Artificial Intelligence. The AI tells the object what to do. Modifying the AI of an object changes what the object does.

Windows
1. Click on the object, thus bringing up the extended options window.

2. Click on the AI button. The AI window, with all modifiable parameters for that window, appears.

3. Use [Backspace] to delete the current parameter's value.

4. Type in the desired value.

5. Click on the check button to close the AI window and make the changes complete.

Shortcut Keys

1. Move the cursor over the object.

2. Press [SHIFT A] to bring up the AI window with all modifiable parameters for that window.

3. Move your cursor over the parameters box you want to change.

4. Click or press [ENTER].

5. Use [BACKSPACE] to delete the current parameter's value.

6. Type in the desired value.

7. Click on the check button to close the AI window and make the changes complete.

AI Parameters

The vast majority of the items in the Object List have at least one or two modifiable parameters in their AI. Some of the more complex objects can have half a dozen or more. In general, you don't have to worry about what every possible field in an object's AI box does (some of them, literally, don't do anything — they might be something the programmers tried that didn't work out or proved unnecessary, and never got removed). Other parameters are set to a certain default, and you'll find yourself with no need to change them. If an object has a parameter that does something absolutely unique, or that must be modified, that is noted in the object's description below.

Here are the AI parameters and normal settings for Nick Vrenna:

		Gravity	**0**
Xvel	**0**	**Health**	**100**
Yvel	**0**	**Morph Pr**	**0**
Xacel	**0**	**AI Type**	**0**
Yacel	**0**	**AI State**	**1**
ST time	**0**	**Fade 0-15**	**0**

Of these, **Xvel, Yvel, Xacel, Yacel** and **Gravity** cannot be modified for Nick. They will always revert to the defaults.

ST Time and **Morph PR** don't actually apply to Nick in any way. You can enter any number you want and it won't effect how he reacts.

Health can be modified from the AI. This is the amount of damage Nick can take before dying.

AI Type doesn't effect Nick in any way, but for mutants and certain other objects it is extremely important. When there are multiple kinds of the same object with different properties (most often this applies to mutants, robots and turrets with different weapon types) this field defines which kind any given object actually is.

AI State is the one parameter possessed by virtually every object in the game. If an object's AI state is 0 the object is inactive, and if it's 1 it's active.

Fade 0-15 controls how completely Nick blends into the background. When using his special Stealth power-up his Fade is at 15, but normally he's at 0 — fully visible. This can be changed manually, but doing so is of little practical use, since it only fades the bottom half of Nick's body — the top half remains fully visible.

Don't be afraid to experiment with different AI settings — you may discover an interesting ability or special effect. On the other hand, don't be disappointed if a parameter won't change, or if changing it doesn't seem to do anything significant.

Object Listings and Descriptions

On/Off Objects

Objects which can be toggled on or off change their color to reflect these states: green means on and red means off. On/Off objects include:

- Sensor
- Death_respawner
- Death_sensor
- All switches
- All logic gates (see Gates, below)

Enemy Weapons

Mutants and wall turrets can be any one of eight types. Types vary according to the kind of weapon they use, and are assigned by a numerical system.

0 = yellow-white fast Laser shot
1 = golden-orange slow Laser shot
2 = Grenade
3 = Rocket
4 = Plasma
5 = Fire Bomb
6 = Death Frisbee
7 = Laser Saber

Weapons

Note that there are no weapon objects in the game, just ammo objects. When Nick first acquires any new type of ammo, the weapon associated with that ammo automatically appears in his inventory, and remains there for the rest of the game.

Codes

Four code letters are used to indicate common classes of object:

R. The object is a reactor. It can be linked to a sensor or switch and will be activated when the sensor or switch is on. Unless otherwise noted, if a reactor is not linked, it will be active.

T. The object is a trigger. It causes other objects (reactors) to go from inactive to active (and vice-versa). Triggers are sensors, switches or logic gates.

A. The object is ammunition for one of the seven weapon types. If ammo is connected to a sensor or switch which is off, the ammo will be invisible, untouchable, and unaffected by gravity, until the sensor or switch turns on. If it has no links it will be active.

W. The object is capable of mounting various weapons. The weapon type may be selected from the "Enemy Weapon" list above, and the correct number entered in the Object's AI under the AI Type parameter.

Ambient_sound R

Ambient sounds play continuously with a user-definable repeat and random delay and volume. They have a range of 500 pixels, and are played at their highest defined volume at their point of origin, decreasing with distance.

Ant_crack R W

This is an irregularly shaped object — like a crack in a wall — from which any number and type of mutant can come bursting forth.

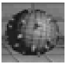
Ant_roof R W

This is a mutant whose initial state is an invulnerable cocoon. The cocoon is usually placed on the roof, hence its name. If not linked to a sensor or switch, it will activate when Nick walks underneath it; it will not activate if Nick is above it, no matter how close. It can be any type of mutant.

Big_bomb R

This is a really big bomb which is detonated by a sensor or switch. It has a user-definable delay which can be set in its AI. It can kill Nick if he is in range of its blast (which is not very large). Several can be chained together by linking them sequentially. If placed with no links, the bomb will be active and will explode after the pre-set delay.

Block
A block is just an object which can be hurt, but otherwise does nothing else but block Nick. Nick can walk on blocks. Blocks ignore all links.

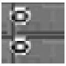


Boulder R
A boulder is an object which (when active) will bounce on the floor doing damage to anything underneath it, including mutants, hidden tiles, Nick, and anything else capable of taking damage. It must be dropped from a height to bounce. Its x and y velocity can be modified. It can be destroyed — damage breaks it up into a number of small boulders which explode when they hit the ground.

Bomb R
This is a smaller version of the big_bomb and has the same attributes, except its explosion is weaker and it's smaller. See big_bomb above.

Compass
This is used only in multi-player games. It lets the player see where everyone else in the game is located when he uses the map [M].

Conc R
This is a mine. If Nick touches this object, it explodes damaging him. It is most effective when placed on the floor.



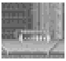
Conc_air R
This is an airborne mine. If Nick touches this object, it explodes. Otherwise it will just float there waiting for him

Death_respawner
This object allows you to spawn an object onto another object's location when it dies. If linked to a mutant, then upon the mutant's death an object you choose will appear where the mutant died. This can be useful for creating certain types of puzzles that require that Nick kill a certain mutant. This object can only be used on the ant (mutant) characters. The linking order is as follows:

First link the death_respawner to any object you want to be respawned. (Note: the first link is always the object that will be respawned. It is also interesting to note that if for some reason the first link's reactor dies, the second link becomes the first link.) Now link the death_respawner to any number of mutants (only mutants) which you want the chosen object to be respawned from.

Death_sensor T
This object, like all other sensors, has two states — on or off. When the object the sensor is linked to dies or is acquired (as with ammo and health) the sensor will switch on. It can have multiple links. If that is the case, it will only turn on when all its triggers are dead or deleted. This object allows you to link other objects (only doors, trap doors, lifts, etc.) to it as reactors, which are activated when its triggers die.

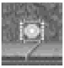
Dfris_icon10 R A
The large (10 round) ammo unit for Nova Spheres.



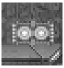
Dfris_icon4 R A
This is the small (4 round) ammo unit for Nova Spheres.

End_game R
This runs the endgame.

End_of_s
This writes End Of Shareware on the screen.

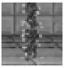


Fbomb_icon1   R   A
This is the small (1 round) ammo unit for napalm.



Fbomb_icon5   R   A
This is the large (5 round) ammo unit for napalm.

Flyer   R   W
This is a flying robot, which can fire any of the weapon types. The flyer has many modifiable AI parameters.



Force_field   R
When activated by a sensor or switch, this object will make a wall of energy which blocks Nick from crossing. It must be placed on the ceiling in order for it to work. It can be any height, as long as both the top and bottom are visible on screen together.

Gates: For the purposes at hand, a logic gate is an object that allows sensors and switches to interact with one another, and with the objects they control, in various complex ways. Normally, switches and sensors cannot be linked directly to one another, nor can a single object be linked directly to more than one sensor or switch. Logic gates make such combinations possible, and also allow the sensors and switches to act in unexpected ways.

Gate_and   T
A gate_and will only turn on when all the sensors or switches it's linked to are in the on state. If, however, one of the links turns off, the gate_and will revert to an off state. This logic gate can be linked between a sensor or switch and more than one object.

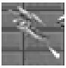
Gate_delay   T
If the gate_delay is linked to a sensor which is on, it will eventually turn on as well. However, the special property of the gate_delay is that it has a user-definable delay (in frames) in between switching states, whether from on to off or vice versa. It can be linked between a sensor and switch and only one object.

Gate_not   T
A gate_not will be in the opposite state of the sensor or switch it is linked to. If a gate_not is linked to a sensor in an on state, it will be in the off state. Gate_not can be linked between a sensor or switch and only one object.

Gate_or   T
A gate_or will be in the on state whenever any one of the sensors or switches it is linked to are in an on state, and is off only when all the sensors or switches it is linked to are in an off state. This logic gate can be linked between a sensor or switch and more than one object.

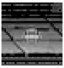
Gate_pulse   T
A gate_pulse will, when switched on by an on sensor or switch, alternate between on and off states with a user-definable delay (in frames). Gate_pulse can be linked between a sensor or switch and only one object.

Gate_xor   T
This logic gate can be linked between and object and two and only two sensors or switches. A gate_xor will switch on if and only if its sensors/switches are not in the same state. If the two links are in different states, the gate_xor will be in the on state. However if one of the links switches states, making them both in the same state, the gate_xor will be in the off state.

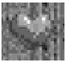


Green_flyer   R   W
This is a green flying robot, which can fire any of the seven weapon types. This object has many modifiable AI parameters.

Grenade_icon10 R A
This is the large (10 round) ammo unit for the Grenade Launcher.



Grenade_icon2 R A
This is the small (2 round) ammo unit for the Grenade Launcher.



Health
This is a heart icon. It increases your health. It restores 4 to 20 health points, depending upon the difficulty level the game is set to.

Hidden Ramps and Walls: Hidden objects are used to hide secrets. Normally, they can be destroyed using any of your weapons. However, hidden ramps and walls can be made temporarily indestructible by linking them to sensors or switches. When the switch or sensor comes on, the hidden ramp or wall will self-destruct, causing a lot of damage to any mutants, and/or other hidden ramps or walls surrounding it, but no damage to Nick. If is has no links it can be destroyed normally. Hidden ramps and walls with a right or left orientation show up in the editor as right-hand tiles, but can be flipped using [X].

Hidden_ramp1	**R**	This hidden ramp is a ceiling part.
Hidden_ramp2	**R**	This hidden ramp is a floor part.
Hidden_wall1	**R**	This hidden wall is a wall tile.
Hidden_wall2	**R**	This hidden wall is a ceiling tile.
Hidden_wall3	**R**	This hidden wall is an interior tile.
Hidden_wall4	**R**	This hidden wall is a floor tile.
Hidden_wall5	**R**	This hidden wall is a corner floor-to-wall tile.
Hidden_wall_2x2	**R**	This hidden wall is four interior tiles in a square.
Hidden_wall_3flr	**R**	This hidden wall is three floor tiles.
Hidden_wall_3top	**R**	This hidden wall is three ceiling tiles.
Hidden_wall_3wall	**R**	This hidden wall is three wall tiles.
Hidden_wall_aflr	**R**	This hidden wall is a red floor tile.

Indicator R
This object switches its color from red to green when it turns off and on. It's useful when showing Nick info about sensor states or switch states. It's basically just a very large and obvious way to show whether a given switch or sensor is on or off.

Jugger R
This is a walking or stationary robot which tosses grenades. Its AI can modify whether it walks or is stationary, and the angle and speed at which it throws the grenades.

Ladder
This object allows Nick to climb within the area bounded by the ladder object and another object the ladder is linked to. This means the ladder needs to be linked to another object to work. Don't confuse the ladder object (which is invisible in play and which allows Nick to climb) with the ladder art in the *Foreground Tile* window (which does nothing unless the ladder object is used).

The ladder object should be in the upper left hand corner of the ladder and the linked object in the lower right hand corner. If Nick cannot climb down the ladder, then the ladder object in the upper left hand corner is too low and needs to be raised above Nick's feet. When he enters the area, he can climb up the ladder when the player presses up. (Note: Nick is considered to be in the box when his feet are in the box.) This object requires one and only one link to another object (usually a marker) to function.

Lava
Lava is an object which does constant damage to Nick as long as he is in contact with it. Lava can be placed only on the floor, and is laid down in tiles. The damage for a lava area can be increased by stacking two or more layers of tiles.

Lighthold
This object allows objects to have light icons, and thus lights, associated with them. A light can be 'glued' onto another object by linking the lighthold to an object that you want to hold the light, and then linking the lighthold again to the light you want that object to hold. (Note: you must use a window link for lights — see p. 30.)

Lightin R
This object is an electrical floor that flashes lightning and damages Nick if it comes in contact with him. The time delay between flashes (in frames) is user-definable.

Lsaber_icon10 R A
This is the small (10 round) ammo unit for the Laser Saber.

Lsaber_icon50 R A
This is the large (50 round) ammo unit for the Laser Saber.

Marker
This object which does nothing but mark location and hold links for objects which need to be linked to other objects. For a specific description of this object in use, see Ladder, above.

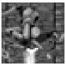
Mbullet_icon10 R A
This is the large (10 round) ammo unit for the Laser Rifle.

Mbullet_icon5 R A
This is the small (5 round) ammo unit for the Laser Rifle.

Next_level
This object will take you to another level when activated (Nick must be touching it while the player presses the down arrow). The next level it takes Nick to is user defined, listed as a number in its AI.

To use the next_level object it is necessary for the next level to be in the levels directory under the current ABUSE directory and use the naming convention, level##.spe where ## = any number from 00 - 99. (Note: the first 0 must be included if the number is less than 10).

It is advisable to double-link the next_level to its top, the next_level_top, for convention and future expansion purposes. To double link something means to link the objects mutually — link one object to another and the second object back to the first.



Next_level_top
The top part to the next_level. Should be double-linked (see Next_level, above) to the next_level object.

Obj_holder R
This object is similar to the lighthold, but instead of "gluing" a light to an object, it glues together two objects.

This object is sensitive to the ordering of the links. The linking order is as follows:

First link the object_holder to the object to be held.

Next, link the object_holder to the object that is holding the first.

There is an optional third link to a switch or sensor which, if it exists, will act as a trigger for the obj_holder, when the trigger switches on. (Note: if the first or second link is broken, as will happen if the linked object dies or is deleted, the links will reorder themselves.)

The obj_holder has modifiable AI for the displacement of the first object with respect to the second object. An obj_holder with no links or one link will be deleted by the engine.

Obj_mover
This object allows you to create paths for objects to move on. It accepts two links. The linking order is as follows:

The first link marks the endpoint for the path. The obj_mover can be linked to a marker or another obj_mover.

The second is the target object which is to be moved.

By linking obj_movers in a cyclic pattern the target object can be kept moving indefinitely. When the obj_mover moves the target object to its destination, the link is then transferred to the end object, which in a chain of obj_movers will then move the target object to the next link in the chain. A chain composed of multiple obj_movers can hold multiple target objects, one for each pair of obj_movers.

The user can define the speed at which the object_mover moves the target object in the AI.

Plasma_icon10 R A
This is the small (10 round) ammo unit for the Plasma Beam.

Plasma_icon50 R A
This is the large (50 round) ammo unit for the Plasma Beam.

Pointer
An object which does nothing, but is used to draw the player's attention to certain points in the level. It looks like a golden arrow.

Power-ups: These are collected by Nick to change his abilities. Nick can only possess one at a time. Special powers are activated using the command key (except for power_health, which doesn't need to be activated).

Power_fast
This power-up allows Nick to move twice as fast as normal and jump higher.

Power_fly
This power-up allows Nick to fly around.

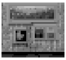
Power_health
This power-up allows Nick to have up to 200 health points. This power-up is activated automatically — the player doesn't need to use the right mouse button.

Power_sneaky
This power-up allows Nick to be partially invisible to other players in multi-player games (note that the power is useless in single-player games).

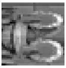
Pusher R
This object will push Nick in the direction of the arrows. It has a modifiable AI allowing you to change the strength at which it pushes Nick.

Respawner
This object respawns another object. It has only one link, the object which it will respawn, and in its AI is a user-definable delay time between the respawnings. An object can only be respawned if it is deleted (killed or picked up) once it has been spawned. The object will be spawned at the same location of the respawner.

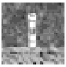
This object is very useful in network games, since ammo and health tend to get depleted very quickly and need to be respawned often. (Note: The respawn icon must be placed sitting on the floor for ammo rounds. Power-ups and health can be placed in the air. If ammo clips are showing up foggy, then the respawn icon is either too high or too low.)



Restart_position
This object functions as a user-activated save position in the game. When the levels are very large, it is necessary to put in restart_positions, so if Nick dies, he does not have to start from the beginning of the level again. The more restart_positions in a level, in general, the easier the level.



Rob1 R
This is a large robot that can chase Nick down halls. It has large spinning blades that are situated in front of it, which can do damage. It has user-definable speed and health settings in its AI.



Rocket_icon2 R A
This is the small (2 round) ammo unit for the Rocket Launcher.

Rocket_icon5 R A
This is the large (5 round) ammo unit for the Rocket Launcher.

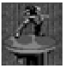
Sensor T
This is a basic triggering device that turns on and off depending on how close Nick approaches.

It has on and off area boxes that must be set by the designer. When Nick enters the on area (green box) the sensor will turn on and stay on until Nick leaves the off area (red box). (Note: the feet of Nick must be inside the area boxes for him to be considered inside the area box, therefore sensor boxes should extend well into the floor tiles, to maximize their effectiveness.) Usually the off area encloses the on area, but this is not required. Each of the areas are user-definable in the sensor's AI.

Sensors act as triggers, responding to Nick's position. Many different objects can be linked to a sensor. Sensors are also used to extend the range of objects by linking them to other objects. You cannot link a sensor to a switch, or to another sensor.

Smart Platforms. The smart platforms are objects which require two links, usually two sensors (but not always). If the sensor (or whatever) for one link is on, the platform will travel to that object's location. A third optional link to another sensor or switch will direct the platform to be active or inactive, by the trigger's state (on = active, off = inactive). They all have modifiable AIs for their traveling speed.

Smart_plat_big R
This platform is a big lift.



Smart_plat_red R
This platform is a red lift.

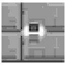
Smart_plat_small R
This platform is a small lift.

Spray_gun R W
This is a gun which, when active, will sweep an area with a spray of shots. It is capable of shooting any of the weapon types. The arc, range and firing speed are user-configurable.

Spring R
This object allows Nick to jump higher than normal. The longer Nick is in contact with the spring the farther it will push him. The strength of the push is user-definable.

Start
This object sets the starting point for Nick when a new level is started. It is possible to place multiple starting points in a level (and this is, in fact, necessary for multi-player levels). However, a single-player game will always start at the last start placed.

Step
This object just allows Nick to walk on it. It is useful as a stepping stone.



Switch T
This is a player-activated object. Along with the sensor, it is one of the basic triggering devices in the game. It has two states, on and off.

Switch_ball T
This is a variant on the switch principle. It can only be activated when Nick shoots it. It cannot be deactivated.

Switch_delay T
This switch, once activated, will revert back into its default state after a certain user-defined time delay. Its default state is off.

Switch_dimmer T
This is the only object that can modify lights during play. It requires two links, one to a switch or any object which can change states (on or off), and the other to a light which it modifies. Since light and object links are different, it does not matter in which order you link them to the switch_dimmer. The switch_dimmer can increase or decrease the size of the light, as set by its AI.

The switch_dimmer is one of the few objects which have different properties depending upon their orientation. If you flip a switch_dimmer around, by using X when your cursor is on it, it will function in reverse. That is, it will make the light grow, instead of shrink.

The switch_dimmer will only be active when the object it is linked to is on.

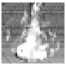
Switch_door R
This is a door that can be opened or closed. It requires one link to a sensor or switch, which when on will set the switch_door to open, and when off will set the switch_door to closed.

Switch_mover R T

This object requires two links, the first to the object that can change states (on or off) and the second to the object which is to be moved. When the first object's state is on, the switch_mover will move the second object to the switch_mover's position and then the engine will delete the switch_mover. A switch_mover with no link will be deleted by the engine.

Switch_once T

This is a variant on the basic switch object. It can only be switched once, from off to on.



Tele2

This is an object which requires one link. When activated, it will teleport Nick to the position of the object it is linked to. Tele2 can be double-linked to each other (see next_level, above), thus creating connections between locations which have no direct connections.

Tele_beam

This is just art. It is to be placed with the next_level and next_level_top objects.

Tp_door

This object is a door that requires one link, to the door it leads to. A second, optional link can go to a sensor or switch which will allow the door to be used when the switch is on. When activated, the tp_door will teleport Nick to another tp_door. It is convention that a tp_door be linked to another tp_door, only.

Track_gun R W

When active, this object will track and shoot Nick with a rate of fire and weapon type defined by its AI.

Train_msg

This object places the training messages found on Level 0 in the game. To select which pre-established message to place, go to the train_msg's AI and enter a number from 0 on up.

Trap_door2 R

(Note: there is no trap_door1 in the Editor.) This is a trap door that can be opened and closed. It requires one link to a sensor or switch, which when on will set the trap_door2 to open, and when off will set the trap_door2 to closed. This object goes on the floor.

Trap_door3 R

This is a trap door that can be opened or closed. It requires one link to a sensor or switch, which when on will set the trap_door3 to open, and when off will set the trap_door3 to closed. This object goes on the floor.

Who R W

This is a flying robot, which can fire any of the weapon types. It has many modifiable AI parameters.

Jonathan’s Rules of Thumb for Level Design

By Jonathan Clark, Crack dot Com

1. Fill in the level nicely. Take your time and pick out textures that fit/look well together. Use lighting to enhance these. Don’t just hit [T] for flood fill, and don’t just use the default light setting. You must combine lighting, textures and ambient sounds to create the atmosphere needed to keep the player immersed.

2. Avoid sharp edges in textures and lighting. Do not use half-lights in the middle of a room.

3. Avoid using teleporters unless you have to. The player can easily get lost with doors and teleporters jumping all over the place.

4. Keep rooms separated. Don’t let the player see another part of the level he cannot get to yet. If you don’t give enough space between areas the player will feel like a mouse in a maze, rather than getting immersed in the game.

5. Let the player kill everything. People want to kill. Avoid puzzles where the only solution is to run away.

6. Pace the action. Don’t keep the player running forever. Give him resting points. On the other hand, don’t let him run down long hallways with nothing to do.

7. Surprise the player. They dig that.

8. Don’t make it too easy, nor too hard. The level should be barely passable by experts on extreme and passable by good players on hard. If a player makes it through the whole level the first time, without reloading, then it is too easy.

9. Don’t be repetitive. Each problem the player encounters should be different.

10. Lead the player, but don’t let him know you are leading him. Make the level mostly linear, so the player does not have much chance of getting lost (and frustrated), but use tricks to make it seem less linear – like back-tracking over the same area with different puzzles appearing the second time.

11. Avoid using a lot of ladders. They are slow and drag down the action.

12. When using doors, do not let the new area the door takes you to loop back. When you go through a door, you are going into a new area, and it’s disorienting when you come across the door you just went through.

13. Stock the player with ammo according to the task. Don’t be too stingy, and don’t give them so much the level becomes too easy.

14. Avoid over-using the ANT_CRACK. Sure it’s easy to put one down and have tons of aliens pour out of a wall, but if used too often it loses its excitement.

15. Avoid jumping puzzles. Jumping from one platform to another platform gets old real fast.

Object List Quick Reference

*SEE LEGEND IN THE “MODIFYING AN OBJECT’S AT” CHAPTER FOR ABBREVIATIONS.

Ambient_sound	**R**	Background noise for an area.
Ant_crack	**R W**	An irregularly shaped object from which any number or type of mutant can emerge.
Ant_roof	**R W**	An invulnerable cocoon from which any type of mutant can emerge.
Big_bomb	**R**	A bomb detonated by a sensor or switch.
Block		Just a large mass that blocks Nick. It can be damaged or walked on.
Boulder	**R**	A large, bouncing object capable of doing damage.
Bomb	**R**	A somewhat weaker version of the big_bomb.
Compass		A device that allows a player to see where all the other players are located in multi-player games.
Conc	**R**	A mine that explodes when touched.
Conc_air	**R**	An airborne mine that explodes when touched.
Death_respawner		When an object the death_respawner is linked to dies, it causes a second object it is linked to to appear in the first object’s place.
Death_sensor	**T**	A sensor that turns on only when a linked object dies or is acquired.
Dfris_icon10	**R A**	The large (10 round) ammo unit for Nova Spheres.
Dfris_icon4	**R A**	The small (4 round) ammo unit for Nova Spheres.
End_game	**R**	Runs the endgame.
End_of_s		Writes End Of Shareware on the screen.
Fbomb_icon1	**R A**	The small (1 round) ammo unit for napalm.
Fbomb_icon5	**R A**	The large (5 round) ammo unit for napalm.
Flyer	**R W**	A flying robot that can fire any weapon types.
Force_field	**R**	An uncrossable wall of energy.
Gate_and	**T**	Turns on when all sensors or switches it is linked to are in an on state.
Gate_delay	**T**	Delays the reactor becoming active after the sensor or switch goes on.
Gate_not	**T**	This gate is off when the sensor or switch it is linked to is on.
Gate_or	**T**	This gate is on whenever any one of the sensor or switches it is linked to are on.
Gate_pulse	**T**	Alternates between on and off with a user definable delay.
Gate_xor	**T**	Turns on only if the two sensors it’s linked to are not in the same state.
Green_flyer	**R W**	A flying green robot.

Grenade_icon10	**R A**	The large (10 round) ammo unit for the Grenade Launcher.
Grenade_icon2	**R A**	The small (2 round) ammo unit for the Grenade Launcher.
Health		The heart icon that restores lost health.
Hidden_ramp1	**R**	A ceiling part.
Hidden_ramp2	**R**	A floor part.
Hidden_wall1	**R**	A wall tile.
Hidden_wall2	**R**	A ceiling tile.
Hidden_wall3	**R**	An interior tile.
Hidden_wall4	**R**	A floor tile.
Hidden_wall5	**R**	A corner floor-to-wall tile.
Hidden_wall_2x2	**R**	Four interior tiles in a square.
Hidden_wall_3flr	**R**	Three floor tiles.
Hidden_wall_3top	**R**	Three ceiling tiles.
Hidden_wall_3wall	**R**	Three wall tiles.
Hidden_wall_aflr	**R**	A red floor tile.
Indicator	**R**	A large colored panel that's green when it's on and red when it's off.
Jugger	**R**	A large robot that throws grenades.
Ladder		Allows Nick to climb.
Lava		Damages Nick when he crosses it.
Lighthold		Allows a light to be linked to an object.
Lightin	**R**	Electric floor that flashes and does damage.
Lsaber_icon10	**R A**	The small (10 round) ammo unit for the Laser Saber.
Lsaber_icon50	**R A**	The large (50 round) ammo unit for the Laser Saber Marker. An object that serves as a link for objects that need to be linked to another location.
Mbullet_icon10	**R A**	The large (10 round) ammo unit for the Laser Rifle.
Mbullet_icon5	**R A**	The small (5 round) ammo unit for the Laser Rifle.
Next_level		Takes Nick to the next level when activated.
Next_level_top		Top art for the next_level teleporter.
Obj_holder	**R**	"Glues" two objects together.
Obj_mover		Establishes a path for objects to move on.
Plasma_icon10	**R A**	The small (10 round) ammo unit for the Plasma Beam.
Plasma_icon50	**R A**	The large (50 round) ammo unit for the Plasma Beam.
Pointer		An arrow used to draw the player's attention to certain points in the level.

Power_fast		Allows Nick to move twice as fast as normal and jump higher.
Power_fly		Allows Nick to fly around.
Power_health		Allows Nick to have up to 200 health points.
Power_sneaky		Allows Nick to be partially invisible to other player in multi-player games.
Pusher	**R**	Pushes Nick in the direction indicated.
Respawner		Causes ammo and health to be respawned a certain interval after they're taken.
Restart_position		A user-activated save position.
Rob1	**R**	A large robot with spinning blades.
Rocket_icon2	**R A**	The small (2 round) ammo unit for the Rocket Launcher.
Rocket_icon5	**R A**	The large (5 round) ammo unit for the Rocket Launcher.
Sensor	**T**	A basic triggering device that turns on and off depending on Nick's position.
Smart_plat_big	**R**	A big lift.
Smart_plat_red	**R**	A red lift.
Smart_plat_small	**R**	A small lift.
Spray_gun	**R W**	A gun that sweeps an area with a spray of shots.
Spring	**R**	Allows Nick to jump higher than normal.
Start		Sets the starting point for a new level.
Step		A stepping stone.
Switch	**T**	A player-activated trigger.
Switch_ball	**T**	A switch that's activated when it's shot.
Switch_delay	**T**	A switch that turns itself off after a certain time.
Switch_dimmer	**T**	Dims or brightens lights.
Switch_door	**R T**	A door that can be opened or closed.
Switch_mover	**R**	Moves an object when a switch is thrown.
Switch_once	**T**	A switch that only works once.
Tele2		A teleport platform.
Tele_beam		The third piece of art for the end of level teleporter.
Tp_door		A door that teleports to another location.
Track_gun	**R W**	A gun that tracks and shoots Nick.
Train_msg		Leaves one of the pre-defined training messages.
Trap_door2	**R**	A trap door that can be opened or closed.
Trap_door3	**R**	Another trap door that can be opened or closed.
Who	**R W**	A flying robot.

KEY LISTINGS

General

Edit/play modes (toggle):	[Tab]
Help:	[H] or [F1]
Map Toggle:	[M]
Save level:	[Shift S]
Save a single screenshot as scrnshot.pcx:	[F2]
Save a single screenshot:	[F3] *(its filename will be shot0001.pcx, shot0002.pcx, etc.)*
Save a series of screenshots, one every 5 frames:	[F5]
Scroll view in indicated direction:	[Arrow Keys]

Object Manipulation

Create a link between two objects:
Put the cursor on one object, press-and-hold [Control],
then move the cursor over the second object and release [Control]

Delete object:	[D]
Flip object's direction:	[X]
Kill first link:	[K]
Kill last link:	[Shift K]
Pick up and move object:	[Spacebar] (left-click to release)

Tile Manipulation

Align on a tile grid:	Hold [Control L] and move mouse
Align to the center of a tile grid:	Hold [Alt] and move mouse
"Pick up" foreground tile under mouse:	[~]
Raise/lower tile under mouse:	[R]
Flood-fill an area of identical tiles:	[T]

Windows

Background tile selection window:	[B]
Console window:	[/]
Foreground tile selection window:	[F]
Layer show window:	[Shift L]
Light-source selection window:	[L]
Object selection window: (hold down s<first letter of object name> to jump to an object)	[O]
Palette selection window: (hold down s<first letter of window name> to jump to a palette)	[P]
Profiler window:	[Shift P]
Toolbar window:	[A]
Widen background window:	[Shift B]
Widen foreground window:	[Shift F]

Debugging

Auto lighting disable:	[Shift A]
Center on Nick (the main character):	[C]
Clear weapons:	[Shift Z]
Clone object:	[Shift C]
Delays (toggle):	[Shift D]
Jump Nick to current cursor location:	[J]
Tell where the cursor is (x & y coordinates):	[W]

Credits

Crack dot Com

Programming

Oliver Yu
Jonathan Clark

Art

Murray McMillian
Kevin Tyler
Oliver Yu

Design and Art

Duong Nguyen

Sound

Bobby Prince

Production

Dave Taylor

Based on Abuse screenplay by

James Robinson
Nelson Woss

Abuse PC Version Documentation

Catherine Cantieri
Chris McCubbin
Melissa Tyler
David Ladyman
Incan Monkey God Productions
Design Island

Special Thanks to:

Gnu Foundation
Steven Donaldson
Carly Staehlin for voices
Apple Game Evangelists/Engineers
James Osborne for networking
Mark Gavini
Steve Bollinger
Chris DeSalvo
Michael Evans
Cary Farrier

Bungie Software

Production Manager

Tuncer Deniz

Development Manager

Eric Klein

Packaging

Randy Nelsen

Damage & Spin

Alexander Seropian
Jay Barry
Alex Rosenberg
Doug Zartman
Jonas Eneroth
Matt Soell

Special Thanks to:

Al Carnley at Origin Systems
Abuse Beta Testers