ABUSE.DOC

Written by Duong Dai Nguyen, level designer for Abuse
Edited by Mitch Burton.

TUTORIAL NUMBER 1

Here at Crack dot Com we strive to create fun games. Many elements
go into making a fun game, one of which is level design. Supplied with
your official Abuse package is a level editor. This editor allows
you to produce your own Abuse levels, using prefabricated pieces which
we call OBJECTS. The interface is a little daunting at first, but with some
persistence you can be blasting your way down long, dark alleys pursuing (or
being pursued by!) aliens, robots, and flyers of your own creation. Let's
begin!

Table of contents

(Do a key-word search for the number/letter combination
to jump to the section you want to read)

[1]

Key Listings and Extended Options

[1.a] Key Listings
----

a   : Toggle toolbar window.
A   : Disable auto lighting (when computer is too slow, lighting shuts off)
b   : Toggle background tile picker window.
B   : Make background window wider.
c   : Center on player.
C   : Clone object under cursor.
d   : Delete object under cursor.
D   : Toggle delays.
E   : Toggle enlarged render.
f   : Toggle foreground tile picker window.
F   : Make foreground tile picker window wider.
g   :
h   : Help!
i   : Invert color black (for foreground tile window, for discerning
          transparent and non-transparent tiles)
j   : Jump player to mouse position.
k   : Kill first link of object under cursor.
K   : Kill last link of object under cursor.
l   : Toggle light picker window.
L   : Toggle layer show window.
m   : Toggle map.
n   : Next player focus (not working/useful).
o   : Toggle object picker window.
p   : Toggle palette picker window.
P   : Toggle profiler window.
Q   : Quit (does not ask permission).
r   : Raise/lower tile under cursor.
s   : Search for object. Not working yet.
S   : Save level.
t   : Flood fill the current tile. If "grabbing" a palette window, flood
          fills with the palette pattern.
U   : Ultra smooth drawing. Not working yet.
v   : Volume.
w   : Where. Tells the x & y position of the mouse in the game.
x   : Flip object under cursor on the X axis.
z   : Clear weapons.
Z   : Toggle god mode. Cursor must be inside of view to toggle god-mode.
space : Pick up object, so that you can move it around (click to drop).
ctrl : When moving an object : Hold down to align on a tile grid.
          When not moving an objects : Hold down on one object and let go on
                          top of another to make a link.
alt   : Hold down while moving to align to center of tile grid.

`   : "Pick up" the foreground tile the mouse is current on.
F1   : Help!
F2   : Save a single screen shot to scrnshot.pcx.
F3   : Save a single screen shot, but name shot0001.pcx, shot0002.pcx...
F5   : Save a series of screen shots, one every 5 frames.
F10 : Reduce screen to what it would for 320x200. Useful for hi-res editing.
+   : Increase view size.
SHIFT+: Increase view size a lot.
+   : Decrease view size.
SHIFT-: Decrease view size a lot.
arrows: Scroll view.
TAB : Toggle edit/play mode.
/   : Toggle console window.

[1.b] Extended Options

-nosound
-sfx_volume [0..127]
-vmode [G640x480x256 | // SVGA for linux
      G800x600x768 |
      G1024x768x256]
-size xres yres   // for X11 and DOS, for DOS -size ? will give you
                 // a list of available modes
-2   // doubles each pixel during rendering
-demo   // quits game after an idle timeout and goes to
                 // demo
-lisp   // starts up with an interactive lisp command line
-lock_size   // X11, does not allow window to be resized
-disp/-display   // specify an alternative X server than $DISPLAY
-no_delay   // starts with no delays, plays as fast as computer
                 // goes
-cprint   // print all output to stderr instead of console
-nojoy   // disables joystick use (there isn't any, anyway!)
-lsf [filename]   // use a different lisp startup file than abuse.lsp
-edit   // Initially start in EDIT_MODE, loads "edit.lsp"
                 // as well
-t [filename]   // auto-number and load tiles from SPE file
-f [filename]   // load filenames as "level1.spe"
                 // don't shoot yourself in the foot!

[1.c] X11 (X windows for UNIX) options :

-noshm   // don't use Shared memory for X11
-grab_pointer   // causes abuse to keep pointer inside of window

While playing, press <TAB> to go into development mode, if you ran abuse with
the -edit parameter. Type '/' to bring up the console window, into which you
can type LISP commands.

[2]

Object Listings and Descriptions

[2.a] Links

Before you can learn about the objects you need to learn about
LINKING. Links are the grey lines drawn from one object to another. They
act as visual cues, indicating that those objects are referencing each
other for particular purposes that will be discussed later on in this
document. The exact function of a link depends upon the objects being
linked and the order in which the links are created. Creating a link is
actually quite simple.

First, position your mouse cursor over the object you wish to
start the link from. A grey bounding box will surround the object. Now,
you have two options. The first, and slightly faster, way is to hold down
the <CTRL> key, move the cursor to the object you want to create the link
to, and let go. The second way is to click on the first object. An extended
options window will open. Now, click on the second object, and you've
made your link. Note that the extended options window now belongs to the
second object, not the first. The advantage of this method is that it can
be used to link objects to lights, whereas the first method cannot.

Some objects can have multiple links. To make multiple links,
simply repeat either method. However, if you are using the second
method, you must close the extended options window before you can make
another link from the first object. It is also important to note that links
are sensitive to which object they originate from. When you create a link
from Object A to Object B, Object B is modified only in that its range is
extended when the player is within range of Object A.

There are two ways to kill a link. The first is the most
flexible and efficient, and involves only the shortcut keys. First, move
the cursor over the object that owns the link. Since objects can have
multiple links, you can now either break the links first to last, by
pressing <k>, or last to first, by pressing <SHIFT>-<k>. To see whether an
object is making a link or being linked, move the cursor over an object with
a link line originating from it. If the line turns yellow, then that object
owns the link.

The second method uses the extended options window. First, open said window by clicking on the object. There will be two buttons with a red X on them. The button on the left will break links with objects from first to last. The button on the right will break links with objects from last to first. Note that links to lights must be broken using this method. It is suggested that you close the extended options window when you are finished, to prevent any accidental linking or delinking. Simply click on the check mark to close the window.

Most links that objects make act as triggers. That is, an object usually makes a link to another object, such as a sensor or switch, which can only have two states (on or off). When this sensor or switch changes its state, the object is triggered and then proceeds to do whatever it does best.

[2.b] AI

There are two ways to access an object's AI. The first method is to click on the object, thus bringing up the extended options window. Then, click on the AI button, which will bring up the AI window. The second method is to simply position the cursor over the object, and then press <SHIFT>-<a>.

The AI window will have a bunch of labeled fields, which are different for each object. For the most part, these should be self-explanatory. To modify them, move your mouse cursor over the box for the parameter you wish to modify, use the <BACKSPACE> key to delete the current value, and, finally, type in the desired value. Click on the check to close the AI window and make the changes complete.

[2.c] Range

Another important facet of objects is the range concept. Levels can be quite large, and have hundreds of objects. Obviously, if all of them were given processing time, the game would run very slowly. Thus, only objects within a certain distance of the player are processed. However, not all objects have the same range. For example, distant aliens can walk through closed doors, because the aliens have a longer range. There are two ways to fix this. The first is to search through the LISP code, find the portion that defines the AI of the objects, and change the range there. The second way is to link another object, usually a sensor, to the short-ranged object. When the closer or longer ranged object is active, it will extend the range of the linked object, allowing it to perform its special functions.

[2.d] Object List

Some objects which have on/off states will change their color to reflect
these states. Green, of course, means on, while red means off. The
following objects do this :
  Sensor
  Death_respawner
  Death_sensor
  All of the logic gates
  Indicator
  Switch

Now let's take a look at the objects.

AMBIENT_SOUND
      AMBIENT_SOUNDs are played continuously with a user definable delay
      and volume. They have a range of 500 pixels. Volume is highest at the
      point of origin, and diminishes with distance. You can use the
      number keys to change the sound type quickly. Use SHIFT-number key
      to get more of the sound effects.

      It can be linked to a sensor or switch and will be
      activated when the sensor or switch is in an on state.

ANT_CRACK
      An ANT_CRACK is an irregular object from which any number and type
of
      aliens can come bursting forth. The user can set the number and type
      of aliens by accessing the extended AI options. The alien types are
      as follows :

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

      You can also press a number key while the cursor is on the ant to
      quickly change the type of weapon it fires.

      If it has no links it will NOT be active.
      It can be linked to a sensor or switch and will be
      activated when the sensor or switch is in an on state.

ANT_ROOF

An ANT_ROOF is an alien whose normal state is a invulnerable coccon.
Usually placed on the roof, hence its name. It will activate when
the player comes within range when UNDERNEATH it. It will not activate
if the player is above it, even when in range. An ANT_ROOF can be of
any type of alien. To change its type, place the mouse cursor over the
alien of your choice, watching for a grey bounding box to be drawn.
Once you are over the alien press any of the number keys from 1-7 to
change the alien type, refer above to alien type and number.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

BIG_BOMB

A really big bomb which is triggered by a sensor or switch.
It has a user definable delay which can be set in its AI.
It can kill the player if they are in range of its blast, which is
not very large. They can be chained together by linking them
sequentially. If placed with no links, the bomb will explode on its
own.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

BLOCK

A BLOCK is just an object which can be hurt, but otherwise
only blocks the player. The player can walk on BLOCKs. BLOCKs have
a
health of 30.

It ignores all links.

BOULDER

A BOULDER is a huge, spiked ball. If dropped from a height, it
will bounce on the floor doing damage to anything beneath it,
including aliens, HIDDEN tiles, players, and anything else
capable of being hurt. Its x and y velocity can be modified in
its AI. It can be killed.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

BOMB

This is a smaller version of the BIG_BOMB and does the same thing,
but has a weaker and smaller explosion. See BIG_BOMB above.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

CONC

This is a mine. When touched by the player, it explodes, causing
damage. Otherwise it will just sit there waiting for the player.
Typically placed on the floor.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

CONC_AIR

This is a air mine. Essentially the same as the CONC mine, except
it looks different. Typically placed in the air.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

DEATH_RESPAWNER

The DEATH_RESPAWNER will spawn a new object at the location of
another
object, when that object dies. For example, if linked to an alien, a
user chosen object will appear where that alien died. This can be
useful for creating types of puzzles where the player must kill a
certain alien. This object can only be used on the alien objects.

The linking order is as follows:

First, link the DEATH_RESPAWNER to the object you want to
be respawned.

Next, link the DEATH_RESPAWNER to any number of aliens
(ANTs).

DEATH_SENSOR

This object, like all other sensors, has two states : on and off.
Whenever the object this sensor is linked to dies, or is deleted
(which is the case with ammo and health), the sensor will switch to an
on state. It can have multiple links. If it does, the sensor will
switch to the on state only when all of the objects it has been linked
to are dead or deleted.

DFRIS_ICON10

This is the large ammo unit for the Death Frizbee. It gives the player
10 additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

DFRIS_ICON4

This is the small ammo unit for the Death Frizbee. It gives the player
4 additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

FBOMB_ICON1

This is the small ammo unit for the Fire Bomb. It gives the player 1
additional unit of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

FBOMB_ICON5

This is the large ammo unit for the Fire Bomb. It gives the player 5
additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

FLYER

This is a flying robot, which can fire any of the 7 weapon types.
To change the weapon type, use the same method used for the
ANT_ROOF.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

FORCE_FIELD

This object is a wall of energy which blocks the character from
crossing. This wall can be up to one screen in height. Position the
object at the top of where you want the wall.

If it has no links it will be active (on or off depending).
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

GATE_AND
Logic gates modify their state depending upon the states of their
links. A GATE_AND will turn on only when all of its links are in
the on state. This logic gate can have multiple links.

GATE_DELAY
Logic gates modify their state depending upon the states of their
links. A GATE_DELAY reflects the state of its link, however, it has
a user definable delay before it switches states. This logic gate
can have only one link.

GATE_NOT
Logic gates modify their state depending upon the states of their
links. The state of a GATE_NOT will always be the opposite of the
state of its link. A GATE_NOT only supports one link.

GATE_OR
Logic gates modify their state depending upon the states of their
links. A GATE_OR will be in the on state whenever any one of its
links are on, and is off only when all of its links are off. This
logic gate supports multiple links.

GATE_PULSE
Logic gates modify their state depending upon the states of their
links. A GATE_PULSE will, when switched on by its link, pulsate
between on and off states, with a user definable delay between
off states. This gate only supports one link.

GATE_XOR
Logic gates modify their state depending upon the states of their
links. A GATE_XOR switches on if and only if an odd number of its
links are on. That is, if there are two links, and one is on while
the other is off, the GATE_XOR will be on. If they were both off or
both on, then the GATE_XOR would be off. This gate supports multiple
links.

GREEN_FLYER
This is a green flying robot, which can fire any of the 7 weapon
types. See ANT_ROOF for a description of how to change the weapon
type.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

GRENADE_ICON10

This is the large ammo unit for the Grenade. It gives the player 10 additional units of ammo. Like all ammo, if 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.
It can be linked to a sensor or switch and will be activated when the sensor or switch is in an on state.

GRENADE_ICON2

This is the small ammo unit for the Grenade. It gives the player 5 additional units of ammo. Like all ammo, if 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.
It can be linked to a sensor or switch and will be activated when the sensor or switch is in an on state.

HEALTH

This is a heart icon. It increases your health by 5-20 points, depending on difficulty level.

HIDDEN_ANT

This is an ANT whose normal state is invisible. It acts like the ANT_ROOF, but is invisible.(see ANT_ROOF)

If it has no links it will be active.
It can be linked to a sensor or switch and will be activated when the sensor or switch is in an on state.

HIDDEN_WALL1

Hidden objects are used to hide secrets. They can be destroyed using any of your weapons. Hidden walls can be made indestructible by linking them up to sensors or switchs. When the switch or sensor turns on, the hidden wall will self destruct, causing a lot of damage to any aliens, and/or other hidden walls surrounding it, but no damage to the player. This hidden wall looks like a floor tile.

If it has no links it will be active.
It can be linked to a sensor or switch and will be activated when the sensor or switch is in an on state.

HIDDEN_WALL2

This hidden wall looks like a roof tile.

If it has no links it will be active.

It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

HIDDEN_WALL3

This hidden wall looks like a wall tile.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

HIDDEN_WALL4

This hidden wall looks like a blocking tile.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

HIDDEN_WALL5

This hidden wall looks like a floor tile.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

HIDDEN_WALL_2x2

This hidden wall looks like a 2x2 blocking tile.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

HIDDEN_WALL_3FLR

This hidden wall looks like a 3 tile wide floor tile.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

HIDDEN_WALL_3TOP

This hidden wall looks like a 3 tile wide roof tile.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

HIDDEN_WALL_3WALL

This hidden wall looks like a 3 tile wide wall tile.

If it has no links it will be active.
It can be linked to a sensor or switch and will be

activated when the sensor or switch is in an on state.

HIDDEN_WALL_AFLR
This hidden wall looks like an alien floor tile.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

INDICATOR
This object switches its color between red and green when it turns off
and on. It's useful for showing the player info about sensor states
or switch states.

It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

JUGGER
This is a walking or stationary robot which tosses grenades.
Its AI modifies whether is walks or is stationary, and the angle
at which it throws the grenades.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

LADDER
LADDERs allow the player to climb up and down. The LADDER must be
linked to another object to work. The LADDER marks the upper left
corner of the box the player can climb up in, and the other object
marks the lower right corner. When a player enters this box, they
can climb the ladder by pressing up. Note that a player is
considered to be in this box if his feet are.

This object requies one and only 1 link to another object
(usually a marker) to function.

LAVA
LAVA constantly does damage to the player, for as long as they are
touching.

LIGHTHOLD
A LIGHTHOLD allows other objects to have lights attached to them.
Simply link the LIGHTHOLD to the light and to the object you wish to
hold the light. The order in which these links are created does not
make a difference. The light will now follow the other object wherever
it goes. Remember to use the second method of linking for lights.

LIGHTIN
This object hurts the player when the lighting flashes. The speed is

user definable.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

LSABER_ICON10
This is the small ammo unit for the Laser Saber. It gives
the player 10 additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

LSABER_ICON50
This is the large ammo unit for the Laser Saber. It gives
the player 50 additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

MARKER
An object which does nothing, but can hold links for objects
which need to be linked to other objects such as the LADDER.

MBULLET_ICON10
This is the large ammo unit for the Laser Rifle. It gives
the player 10 additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

MBULLET_ICON5
This is the small ammo unit for the Laser Rifle. It gives
the player 5 additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

NEXT_LEVEL
When the player activates the NEXT_LEVEL object (by pressing the down
arrow while touching it), it will take the player to a level listed as
a number in its AI. To use the NEXT_LEVEL object it is necessary for
the level to be in the levels directory and use the naming convention
"levelXX.spe", where XX is any number from 0 to 99. Note that the first
0 must be included if the number is less than 10.

It is recommended that you double link the NEXT_LEVEL to its top,
NEXT_LEVEL_TOP. To double link something means to link one object to
another and that second object back to the first.

NEXT_LEVEL_TOP
The top part to the NEXT_LEVEL. Should be linked to a NEXT_LEVEL
object.

OBJ_HOLDER
This object is similar to the LIGHTHOLD, but instead of attaching
a light to an object, it glues together two objects. This object is
sensitive to the ordering of the links. The first link is the object
to be held. The second link is the object which is holding the first.
There is an optional third link which, if it exists, will act as a
trigger for the OBJ_HOLDER. The object will be held only if this link
is on.
Note that if the first or second link is broken (which will happen if
the 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 the level designer create paths for objects to
move on. It accepts two links. The first link marks the endpoint
for the path, and the second is the object which is to be moved.
By linking OBJ_MOVERs in a circle, the object can be kept moving
indefinitely. When the OBJ_MOVER moves the object to its
destination, the link is then transfered to the end object, which
in a chain of OBJ_MOVER 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.
You can modify the speed at which it will move in its AI.

PLASMA_ICON10
This is the small ammo unit for the Plasma Beam. It gives

the player 10 additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be activated when the sensor or switch is in an on state.

PLASMA_ICON50

This is the large ammo unit for the Plasma Beam. It gives the player 50 additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be activated when the sensor or switch is in an on state.

POINTER

An object which does nothing, but can be used to draw the players attention to certain points in the level. It looks like a golden arrow.

POWER_FAST

Power icons, when collected, augment the player's abilities. The player can only have one power icon at a time. All power icons are activated by pressing the right mouse button.
This power icon will allow the player to move twice as fast as normal and jump higher.

POWER_FLY

Power icons, when collected, augment the player's abilities. The player can only have one power icon at a time. All power icons are activated by pressing the right mouse button.
This power icon will allow the player to fly.

POWER_HEALTH

Power icons, when collected, augment the player's abilities. The player can only have one power icon at a time. All power icons are activated by pressing the right mouse button.
This power icon will allow the player to have 200 health points.

POWER_SNEAKY

Power icons, when collected, augment the player's abilities. The player can only have one power icon at a time. All power icons are activated by pressing the right mouse button.
This power icon will allow the player to become paritally invisible.

PUSHER

This object will push the player in the direction of the arrows.
The level designer can change the strength at which it pushes the
player in its AI.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

RESPAWNER

(This object is not avaible until net abuse is available)
This object, like the DEATH_RESPAWNER, spawns new objects. It has
but
one link, the object it will respawn. There is a user definable delay
between respawns in its AI. An object is respawned only if it is
deleted once spawned. The object will be spawned at the location of
the RESPAWNER. One use of this object is to have infinite sources of
ammo and health in network games.

RESTART_POSITION

This object functions as a user activated save position in the game.
When the levels tend to be very large, it is necessary to put in
RESTART_POSITIONs, so that if the player dies, they do not have to
start from the very beginning again. In general, the more
RESTART_POSITIONs a level has the easier it is.

ROB1

A large robot which can chase you down halls. It has large spinning
blades which are situated in front of it, that do a decent amount of
damage. It has user definable speed and health settings in its AI.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

ROCKET_ICON2

This is the small ammo unit for the Rocket Launcher. It gives
the player 2 additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

ROCKET_ICON5

This is the large ammo unit for the Rocket Launcher. It gives
the player 5 additional units of ammo. Like all ammo, if it 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.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

SENSOR

The sensor is the basic triggering device. In addition to responding
to the player's position, a sensor can also be used to extend the
range of other objects, since sensors have the longest range. A sensor
has on and off area boxes. When the player enters the on area (green
box) the sensor will turn on and stay on until the player leaves the
off area (red box).

Note that a player is only in the box if his feet are.

The off area typically encloses the on area, but this does not have
to be the case. Each of the areas is user definable in the sensor's
AI.

Sensors do not processs their links.

SMART_PLAT_BIG

The smart platforms are objects which require two links, typically
(but not always) a pair of sensors. If a link is on, it will travel to
that linked object's location. A third optional link will, if it
exists, direct the platform to be active or inactive, depending on the
link's state (on will activate, and off will deactivate). The speed of
the platform can be changed in its AI.
This platform is a big techno looking platform.

SMART_PLAT_RED

This platform is a big alien looking platform.

SMART_PLAT_SMALL

This platform is a small techno looking platform.

SPRAY_GUN

This is a gun object which, when active, will shoot out sprays of
fire. It is capable of shooting any of the weapon types. You can use
the ANT_ROOF method for altering the weapon type, or change it in the
AI. The arc range and firing speed are also user configurable.
    You can use the number keys to change the weapon type quickly.

If it has no links it will NOT be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

SPRING

This object will allow the player to jump higher than normal.
The longer the player is in contact with the SPRING object
the further it will push them. The strength of the push is
user definable.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

START
This object sets the starting point for the player when the game
is begun anew. If the player is set behind objects when the game
starts, then is nessecary to raise this object and save the level
again with the raised START object.
To raise objects, position the mouse cursor over the object, and
press <>> (<SHIFT>-<.>).

It is possible to have multiple starting points in a level. The
game will start where the last one was placed.

STEP
A simple step that the player can walk on.

SWITCH
This is a user activated object. It has two states, on and off.
Other objects reference this object often, as it is a versatile tool.

SWITCH_BALL
This is a user activated object. It has two states, on and off.
It is similar to the regular switch, but is activated when it is
shot. It cannot be deactivated.

SWITCH_DELAY
This switch, once activated, will revert back to its default state
after a user definable time delay. Its default state is off.

SWITCH_DIMMER
This is the only object that can modify lights. 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, you can link them in any
order. The amount of decrease in light can be set in the AI.
If you flip a SWITCH_DIMMER around (press the <x> key with your
cursor over the SWITCH_DIMMER), the light will grow instead of
shrink. The SWITCH_DIMMER will only activate when the object it is
linked to is in the on state.

If it has no links it will NOT be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

SWITCH_DOOR

A SWITCH_DOOR requires one link. When this link is off, the
SWITCH_DOOR will be in a blocking state, but when on, the
SWITCH_DOOR will be in a non-blocking state.

It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

SWITCH_MOVER

This object requires two links, the first to the object which is
to be moved and the second to any object which can change states (on
or off). When the second object's state is on, the switch mover will
move the first object to the SWITCH_MOVER's current position. The
engine will then delete the SWITCH_MOVER.

A SWITCH_MOVER with no links will be deleted by the engine.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

SWITCH_ONCE

This is a user activated object. It has two states, on and off.
Other objects reference this object often as it is a versatile
tool. It can only be switched ONCE.

TELE2

This is an object which requires one link. When activated, it will
teleport the player to the postion of the object it is linked to.
TELE2s can be linked to each other, thus creating connections within
levels which have no direct connections.

TP_DOOR

A TP_DOOR requires one link. This link should be to another
TP_DOOR.
When the TP_DOOR is activated, it will teleport the player to the
TP_DOOR it is linked to.

TRACK_GUN

This object, when active, will track and shoot the player. Rate of
fire and weapon type are modifiable via its AI. You can use the
number keys to change the weapon type quickly.

If it has no links it will be active.
It can be linked to a sensor or switch and will be
activated when the sensor or switch is in an on state.

TRAIN_MSG

When a player is near this object, a preset message will be printed.
These messages are defined externally.

TRAP_DOOR2
        A door similar to the SWITCH_DOOR, but this is used on the floor.
        (see SWITCH DOOR)

        It can be linked to a sensor or switch and will be
        activated when the sensor or switch is in an on state.

TRAP_DOOR3
        A door similar to the SWITCH_DOOR, but this is used on the floor.
        (see SWITCH DOOR)

        It can be linked to a sensor or switch and will be
        activated when the sensor or switch is in an on state.

WHO
        Another type of flying robot, with the same attributes as the FLYER.
        (see FLYER)

        If it has no links it will be active.
        It can be linked to a sensor or switch and will be
        activated when the sensor or switch is in an on state.

[3]

                Preliminary Level Designing, Windows and Menus
                ----

[3.a] Preliminary Level Designing

        When creating a new level, it is generally a good idea to think
about a general theme. Will it be all action? Will there be puzzles, and
if so, have difficult will they be? What is the mood of the level? What
is the skill level of the target player? Keep these questions in mind. You
may also find it helpful to sketch out the general design of a level on
paper before you begin. Levels can be more or less as large as you want
them
to be, but the default size of 100 by 100 tiles should be large enough for
most levels.

        However, before you can begin putting down tiles and objects, you
must first learn to tame the windows and menus that populate the world
we call ABUSE.

[3.b] Windows and Menus

      First, you will need to start the game in edit mode. This will allow you to access the built-in editor. To do this, go to the current ABUSE directory and type in this line :

      abuse -edit

      The "-edit" is an Extended Option. See section 1.b for a complete listing of these.

      The game will now start. It will load level 00, the training level by default. This will suit our goal of learning the windows and menus just fine. Notice that the game is not running yet. If you have already played the game, you will notice there now is a menu bar at the bottom of the screen that is not present when you normally play, and that your cursor is now an arrow instead of the usual target. Let's examine the menus and their functions.

      Under the farthest left hand menu called FILE, we find a listing of file utility commands and some advanced options.

Under FILE-->
        Load Level
        Save Level (S)
        Save level as
        Save game
        New level
        Resize map
        Suspend non-players
        Play mode toggle (TAB)
        Save Palettes
        Start cache profile
        End cache profile

The first 5 options from the top are basic file utility commands found in almost all programs.

     Load level
        This command loads in levels. There are two methods you
        can use to do this. The first is the command line -- simply
        point the cursor at the text box at the top of the window,
        use the <BACKSPACE> key to delete the current contents, and
        type in the path and name of the level you wish to load.
        The second is the file selection system that uses the mouse.
        Under the text box, you will notice a list of files in the
        current directory. By double clicking on a directory, you
        can move around your directory structure. Double clicking on
        a file will have the editor try to load it as a level.
           -------NOTE----------

The file selection system does not work for loading up files outside the directory from which ABUSE was ran. Use the command line to load up levels from other directories outside the current one.

Save level (S)

This command saves levels. A shortcut key is <SHIFT>-<s>. This command will save the level with the name and path of the currently loaded level.

Save level as

This command saves a level using a different name from the currently loaded level. Its methods are the same as the Load Level command, and it has the same limitaions.

Save game

The save game command saves the level with your current position and weapon status. When the level is loaded up, the player will begin at the position from which the game was saved. This differs from the other save level commands in those do not save your current position, and weapon status.

New level

This command creates a new level.

Resize map

This command resizes the map (level). The default size is 100 x 100 tiles. A tile is 30 x 15 pixels, so this translates into a 3000 x 1500 pixel world, or roughly 9.5 x 7.5 screens at the default resolution of 320x200. To change the size of the world type in the new parameters and press <OK>.

Suspend non-players

This command stops all objects from being processed, aside from the main character. You will notice that when you execute this command, the main characters upper body is frozen. This is because the main character is composed of two objects, and only the lower half is processed. This mode is useful in level designing when you don't want to disturb the world as you roam about.

Play mode toggle (TAB)

This command starts the game from edit mode, allowing you to play the game as you normally would. Once you select this command you can only use the short-cut key (TAB) to return to edit mode. It is suggested you use this key and forgo the menu command, as the shortcut is much faster and not as tedious, and you will find yourself using it often.

Save palettes

> This command saves palettes to a file called edit.lsp, which
> can be edited using a conventional text editor. Palettes will
> be discussed in detail later in this document.

Start cache profile

> This is a level optimization feature. It lets the engine know
> in advance which graphics and sounds will be used in a level,
> so that it can store them in an efficient manner, resulting
> in reduced disk access when playing the level.
> -----------------------NOTE--------------------------
> This command should only be used for COMPLETED levels.
> To use this command properly these steps must be followed :
>
> 1) Start a new level from the proper position and state.
> 2) Select this command from the menu system.
> 3) Enter play mode.
> 4) Play through the level as you normally would. Using
>   save positions is ok. Each time graphics, sound,
>   etc. is used, the game makes note of it.
> 5) Continue until you reach the end of the level. Do NOT
>   go to the next level.
> 6) Press tab to return to edit mode.
> 7) Use the load command to reload the level.
> 8) Save the level from the menu, or by using <SHIFT>-<s>
> 9) Choose End Cache Profile from the file menu.

End/save cache profile

> This command will save and end the cache optimization
> procedure. It saves the data to a file under the name
> and directory of the original level with an extensiton of
> .cpf.

From the FILE menu you will use the Load Level command most often.
The rest are occasionally used, and the Save Level command has a
keyboard
short-cut so it will not be necessary to use the menu for that command. Now
let's take a look at the EDIT menus.

Under EDIT ---->
> Toggle light
> Set scroll rate
> Center on player
> Add palette
> Toggle Delays
>
> God mode
> Clear weapons (z)
> Mouse scroll
> Lock palette windows

Raise all foreground
Toggle object names

Toggle map   (m)
Shrink to 320x200 (F10)
Disable view shifts
Ultra Smooth draw (U)
Disable Autolight (A)
Show FPS/Obj count

Record demo
Play demo

Toggle light
This command turns on and off the lighting. Lighting is a
unique feature of the Abuse engine, allowing the level
designer to create new and interesting scenes simply
by moving around the light sources. However, it can be
distracting while editing a level, so this command is
provided.

Set scroll rate
This command modifies the rate at which the foreground
layer and background layer scroll relative to each other.
The Abuse engine has the capability for 2 layers of
parallaxing. The top layer is called the foreground layer
and the bottom is called the background layer. The parameters
for the scroll rate are:

rate for background scroll in x axis:
rate for foreground scroll in x axis:

rate for background scroll in y axis:
rate for foreground scroll in y axis:

For example if the parameters were:

X mul   1
X div   2

Y mul   1
Y div   2

Then background would scroll twice as fast as the
foreground in the x and y axes.

Center on player
This command will center the screen on the main character.
It's very useful since the Abuse engine allows the level

designer to move the center of the screen away from the character. The short-cut key is <c>.

Add palette

This command allows the level designer to make custom palettes. When you choose this command, a dialog box will open with several fields. Type in an appropriate name, as well as width and height values. Since the palette can be resized with the arrow keys, you do not need to worry about getting the size perfect. You will now have a blank palette window. You can add foreground tiles to this window by selecting a tile from the foreground tile window (see below), and then clicking in the palette window with the RIGHT mouse button. Palettes are discussed in detail in the section on the Palettes menu option, below.

Toggle Delays

This command will allow the game to run at the fastest speed your machine is capable of. There should be a message telling you whether delays are being turned on or off after you choose this command. The keyboard short-cut is <SHIFT>-<d>.

God mode

In this mode you have all the weapons, unlimited ammo, and cannot be killed. The SHAREWARE version has only four weapons as opposed to the COMMERCIAL versions full eight weapon types. The keyboard short-cut is <SHIFT>-<z>.

Clear weapons (z)

This command resets your weapon status, leaving you with only the laser rifle. All ammo will be set to 0. This is useful when designing and debugging levels, since the Abuse engine does not clear your weapons status when you reset the level. The keyboard shortcut is <z>.

Mouse scroll

An optional feature is the ability to use the mouse as a scrolling device. If this command is active (as indicated by an asterisk), you can move around the level by pushing the cursor against the boundaries of the screen. You can also use the arrow keys to scroll around, however, you must take care that you do not accidentally resize palette windows when using them.

Lock palette windows

This command prevents any accidental alteration to palettes. When this command is active (as indicated by an asterisk) none of the palette windows can be resized nor can the tiles in them be changed. To deactivate this command,

simply select it again.

Raise all foreground
This command will, when active as indicated by the asterisk, make all the foreground tiles laid down in the level be raised. Raised foreground tiles are drawn in front of the character. When in edit mode all raised foregrounds will have an 'X' drawn over them. You can deactivate this command by selecting it from he menu again. To raise and lower individual tiles, position the cursor over the tile you wish to raise or lower and press the <r> key.

Toggle object names
When this command is active, each object will have its name displayed below it in green.

Toggle map   (m)
This command will display a map of the level. The map is a global representation of the level, with area boxes shown. An area box is an area of the level that has its own ambient light and screen displacement values. The map can be scrolled around with the arrow keys, or with the mouse if mouse scroll is activated.

Ambient light is the global level of lighting. A player can only have one ambient light value at a time. However with the help of area boxes, you can have multiple ambient light values. Once a player leaves an area box, the screen displacement and ambient light do not return to their original values. The values of the screen displacement and ambient light change only when the player enters another area box. Screen displacement shifts the center of the screen away from the player. You can access these parameters and others via the toolbar window, which can be opened from the WINDOWS menu. See the section on the toolbar window, below, for a description of the tools and what they do.

The keyboard shortcut for map mode is <m>. To get out of map mode, simply press <m> again.

Shrink to 320x200 (F10)
This command shrinks the view window to 320x200 resolution. When working in hi-res mode, this can give you an idea of what the player will see in low-res mode.

Disable view shifts
If this command is activated (it will have an asterisk next to it in the menu selection) view shifts will not take place when the player enters a new area box. When developing a level, this command can prevent accidental and annoying view shifts. To enable view shifts, simply reselect this command.

Ultra Smooth draw (U)
>   This command will draw the characters as smoothly as possible.
>   This command currently is under development and should
>   not be used.

Disable Autolight (A)
>   This command disables the autolight feature. The autolight
>   feature will turn off the lighting if the frame rate drops
>   below 9 frames per second. Disable this if you're working on
>   a slow machine and want to see the lighting no matter what
>   the speed. To enable autolighting again, reselect this from
>   the menu. The keyboard short cut is <SHIFT>-<a>.

Show FPS/Obj count
>   This command prints the number of frames per second and the
>   number of objects currently being processed in the upper
>   left hand corner of the screen.

Record demo
>   This command will begin recording a demo of the game. These
>   demos can be viewed with the play demo command.

Play demo
>   this will play the demo which you have recorded.

The EDIT menu is very important to designing levels. Most of the
important functions have keyboard shortcuts and can greatly speed up
level designing. Now let's take a look at the WINDOWS menu.

WINDOWS->
>   Foreground (f)
>   Background (b)
>   Draw layers (L)
>   Lighting   (l)
>   Palettes   (p)
>   Objects   (o)
>   Console   (/)
>   Tool Bar   (a)
>   Search   (s)
>   Profile   (P)
>   Save positions

Foreground (f)
>   This command will bring up the foreground tile window.
>   From this window you can select foreground tiles to place
>   in the game world.
>   To expand the foreground window, place your mouse cursor
>   inside the foreground window and press <SHIFT>-<f>.
>   Multiple presses will expand the window even further,

though it will eventually return to a single tile window window. If the foreground tiles seem small, place the cursor in the window and press the <+> key to make them full size. Similarly, if they are taking up too much space, press the <-> key to shrink them. Note that the 0 color black is drawn as beige in the foreground tiles, for identification purposes. To select a tile from this window, simply click on it. You can then place the tile by moving to where you want to put it and left clicking. The keyboard shortcut is the <f> key.

Background (b)

This selection will bring up the background tile window. These tiles are drawn behind the foreground tiles, and will show through only where there are no foreground tiles, or where the color black (0 color) is present in foreground tiles. You can select tiles from this window using the same method as for the foreground window. However, to place tiles, you must right click, as a left click will place a foreground tile. The keyboard shortcut is <b>.

Draw layers (L)

This selection opens the layers window. The buttons in this window toggle the display of various layers. That is, you can toggle the drawing of, among other things, lighting effects and foreground tiles. The latter would be useful when laying down background tiles.

    FORE : Toggles the foreground layer.
    BACK : Toggles the background layer.
    BOUND : Toggles the bounding boxes on foreground tiles.
        Bounding boxes block the player, so that one can
        walk on floors and be stopped by walls.
    LIGHT : Toggles the lighting.
    LINKS : Toggles the display of links between objects.
        In areas with lots of links, they can sometimes
        obscure the foreground. Note that when off, the
        light icons will not be drawn.

The keyboard shortcut is <SHIFT>-<l>.

Lighting   (l)

This will bring up the lighting window. This window has buttons for a variety of light source shapes, as well as an ambient button. Lights are respresented by a white, circular or semi-circular icon, depending upon its shape. You select and place a light in the same way you do a foreground tile.

Sometimes you will want to modify the size or shape of the light. There are several ways to do this. The first method uses the keyboard almost entirely :

1) Position the cursor over the light whose shape
   you wish to change. Or, you can skip to step 3,
   if you have selected a light but have not dropped it.
2) Press the <SPACEBAR> to pick it up.
3) Use the arrow keys to reshape the light, or + and - to
   resize it.

The second method involves the use of the extended options
window :

1) Click on the light.
2) Modify the W and H fields for the light. These will
   change the width and height, respectively. Note
   that these fields expect small numbers. The L2 field
   will change the size of the light. This field can
   accept very large numbers.

The keyboard shortcut is <l>. Lighting is an excellent
tool for creating mood in the levels. Learn how to use the
lighting well.

Palettes   (p)
     This command will open a window containing a list of palettes.
     Choose one and double click on it to open a palette window.
     Sometimes the palette windows will be shrunken so as to take
     up less space. Palette windows can be shrunk and expanded,
     just as the foreground and background windows can. While in
     this window, hold shift and type a letter to jump to the
     palettes starting with that letter.

     Palettes have a variety of uses. Not only can they be used as
     custom tile libraries, they can be used to fill areas, or to
     stamp down one set of tiles at a time. They are quite useful
     for creating complex patterns of tiles with great speed. To
     stamp down the palette, click on the palette's window bar with
     the left mouse button. While keeping the mouse button held
     down, drag the upper left corner of the window to the position
     where you wish to stamp, then press the <SPACEBAR> key. If you
     wish to fill, press the <t> key instead. The result of a
     palette stamp is perhaps best shown rather than described.
     Assume that the palette is being stamped at the upper left
     corner of the grid. A-d are foreground tiles.

```
     Palette      Grid before stamp      Grid after stamp
      +-+-+           +-+-+                  +-+-+
      |a|b|           | | |                  |a|b|
      +-+-+           +-+-+                  +-+-+
      |c|d|           | | |                  |c|d|
      +-+-+           +-+-+                  +-+-+
```

Palettes can also be used to fill or stamp one tile
at a time. Simply left click on a tile in the palette window,
then place it as you normally would. Do NOT right click, as
you might accidentally alter your palette. To help prevent
accidental alterations to the palette, read about the Lock
Palette Windows command, above.

To expand or shrink a palette, simply position the
cursor in the palette window and use the arrow keys to resize
it. BE CAREFUL with this -- if you shrink a palette, any tiles
that were in area you just shrunk will no longer be there,
even if you expand the palette back to where it was. This is
annoying, at best.

Also, make sure to use the Save Palettes command in
the FILE menu to save your palettes so that you may use them
in the future. Saving the level does NOT save the palettes.
Palettes are very powerful tools to a level designer. They
will allow you to do complex fills and create repetitive
structures easily, as well as being custom libraries for
tiles. Palettes are your friends.

Objects   (o)
Objects are the core of the game. They are all the characters
which you interact with. Each one has its own, unique AI
settings. This window gives you a listing of all the objects
in the game. While in this window, hold down shift and type a
letter to jump to the objects beginning with that letter. See
Object Listings and Descriptions for a complete overview of
all the objects in the game. Some basic object utility
commands are as follows :

    <SPACEBAR>   Picks up object under cursor.
    <x>   Flips object under cursor in x axis.
    <d>   Deletes object under cursor.
    <SHIFT>-<c> Clone object under cursor.
    <CTRL>   Position object to a grid.
       (object must be picked up)
    <ALT>   Position object in the middle of a grid.
       (object must be picked up)
    <SHIFT>-<k> Kills last link of object under cursor.
    <k>   Kills first link of object under cursor.

The keyboard shortcut key for this window is <o>

Console   (/)
This will bring up the console window. The console window
can accept command line inputs. It can also be used to
modify the game and receive information about the game's

state.
The keyboard shortcut is </>.

Tool Bar   (a)
This will bring up the toolbar window. It is used in
conjunction with the map view to add and modify area boxes
and place tiles in the map view.

The toolbar menu has 3 icons : a pencil, an arrow, and a box
made of broken lines. The pencil lets you lay down tiles
that you have chosen from either the foreground window,
palette window, or the current level itself. To lay down a
tile in the game, simply press the right mouse button over
an area in the map. The pencil mode is also useful for using
the fill option. To fill an area in the map with one
foreground tile, follow these steps :

   1) Select a tile from any of the three sources.
   2) Make sure the area you wish to fill is completely
      surrounded by tiles. If not, then lay down some tiles
      to fill the gaps.
   3) Move the mouse pointer into the area.
   4) Press the <t> key.

      To select a tile which has already been laid down,
position your mouse cursor over the tile and press
the <`> key. If your foreground window is open at the
same time, it will jump to the tile you just selected.
This selection mechanism will work in map mode as well.

      The box tool allows you to create a new area
box. To do this, follow these simple steps :

   1) Select the box tool.
   2) Enter map mode, if you haven't already.
   3) Hold down the left mouse button, then drag to
      create the box.
   4) Release the mouse button.

      The arrow icon lets you modify the settings of an
area box. To do this, follow this procedure :

   1) Select the arrow tool.
   2) Go to one of the yellow endpoints of the area box.
      Each area box has two endpoints; one at the upper left
      corner, and another at the lower right corner.
   3) Double click on the endpoint. You will know when you
      have selected an area box, as it will have turned into
      a yellow box. This will bring up a window with several
      parameters :

Ambient
   The global lighting level. It ranges from 0 to 63.
   A value of -1 means no change.
Ambient Speed
   This is the increment at which the ambient light
   level changes when the player enters the area box.
   Obviously if the ambient light level is the same,
   there will be no change.
View x off
   This is the number of pixels the player should be
   offset from the center of the screen, in the X axis.
   A negative value will shift the screen left, and a
   positive value will shift the screen right.
View y off
   This is the number of pixels the player should be
   offset from the center of the screen, in the Y axis.
   If the value is negative, it will shift the screen
   upwards. A positive value will shift the screen
   downwards.
View x speed
   The rate the view shifts, in pixels.
View y speed
   The rate the view shifts, in pixels.

   Click on the <OK> (check) button to close the area
box's extended option window to save the changes, or the
<DELETE> (big red X) button to delete the area box.

Search   (s)
   This is currently under development, it will search for
   objects within a level. Do not use.

Profile   (P)
   The profile window will display the current objects and
   rank their CPU time. The higher up on the list, the more
   CPU time taken up by that object or group of objects.

Save positions
   This command will save the positions of all opened windows.
   Thus, when reopened, they will appear where you left them.

[4]
   Level Designing Hints and Practicalities

[4.a] Level Designing Hints and Practicalities

If anyone has read this far, they either have to be crazy, or very determined. You make the call.

If you don't want to draw your own tiles, you need to familiarize yourself with the tile set. A good way to do this is to take a look at the current levels, like LEVEL00.SPE, the training level. Using the <`> key, with the foreground window open, select the tiles which you have an interest in. Learn what they look like, and what they do. Also look at how the various tiles fit together. Use the layers window to turn on the bounding layer and see the bounding boxes for the various tiles. Use palettes so that you don't have to search through the foreground window for your favorite tiles. Play with the objects, see how they work with the player and with each other. And save your level often. Since you often mess up your level while testing it and killing monsters and so on, make use of the restore key (<SHIFT>-<r>), which will reload your level. Note that if you saved the game at a RESTART_POSITION, your savegame and not the original level will be reloaded.
Make sure not to save a level after you've gone through and killed everything.

Making a level is an incremental process, building block by block as you go. Keep in mind your global goals for the level. Keep the action steady, but gradually build up to the "big finish". But, more than anything, make levels and get more experience. This is the best way to learn to make great levels.

With all that said, level designing is a rewarding process filled with little twists and turns that will keep you enthralled and entertained for hours on end.

Please direct UK order's to us. You'll get the game faster and cheaper !

To order any of our products by mail, please complete the following form
and return it to:

Parsec Computer Systems Ltd.
Gaming Central,
Baldrine,
Isle of Man.
IM4 6EG
British Isles

To place your order phone: 01624 861233 or FAX: 01624 862190

YES! I would like to purchase the following

Name: _____

Company: _____

Address: _____

_____

State/County: _____ Zip/Post Code: _____

Country: _____

Telephone: _____

I obtained your Catalog from: _____

ITEM   QUANTITY   DESCRIPTION & FORMAT   PRICE   TOTAL

_____

_____

_____

_____

_____

Delivery to UK addresses:

1st Class œ1.50 per game
Next day delivery œ3.50   Postage and Packing: œ_____

Total: œ_____

* Please make payment in U.K. Sterling *
Checks from foreign countries must be drawn on a U.K. Bank.
Please make cheques payable to: Parsec Computer Systems Ltd,