Breed

COLLABORATORS			
	TITLE : Breed		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 25, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Breed

1.1 The Breed Guide.

```
             ____
   The      /    |                             /
           /    /                             /
          /    /          ____   ____   _____/
         /----    /----  /      /      /    /
        /    |   /      /___   /___   /    /
       /    /   /      /      /      /    /
      /____/   /      /____  /____  /____/  o

            ____
           /                          /
          /                 o        /
         /                     _____/  ____
        / ___    /    /   /   /    /  /
       /    /   /    /   /   /    /  /___
      /    /   /    /   /   /    /  /
     /____/   /____/   /   /____/  /____  Ver. 2.0
```

Guide conversion and additional text by Michael Bromery.

New sky and backdrop info written by Nick McKenzie.

All original text instructions by Andy Clitheroe.

Click to Continue.

1.2 Index of Related Editor Topics.

Welcome to the Breed guide. This guide file contains
step by step instructions, on how to use the editors included.

Click Retrace from a certain topic, to get back to this Index.

Topics Include:

Starting Up:

Floors/Ceilings:

Walls:

Textures:

Note: Textures, in this context, are used for Vector objects only. Make note. Includes Vector Aliens, etc.

Objects:

Note: The link just below talks about the Anim select, which is done from the gamelinker program.

Lights:

Change Floor/Ceiling Lighting in Levels

Aliens:

Obvious Note: Of course, Aliens don’t have to BE aliens. The term is given to any robot or live creature, in the game.

Defining Aliens for the game

Defining Alien Animations

Adding Aliens to your Levels

Placing Control Points for your Aliens tracking paths

Player:

Defining the Player(s)’ Appearance

Guns/Bullets:

Defining Bullets or fast flying objects (Blood anyone?)

Define Player Guns

Doors/Lifts:

Adding Doors and Lifts to Levels

Teleports:

Adding Teleports to the levels

Sound/Audio:

Incorporating Sound into the levels

Levels:

Starting a New Level

Saving the Level

Making Good and Challenging (And devious) Levels

NEW Skydrops/Backdrops:

Creating your own backdrops for outside scenes

Palette:

Note: At the time of this writing. This feature is/was not available for edit. Andy states that you may look for future Breed patch(es). One of them, perhaps, to set this feature active.

Change the Global Palette

Remarks:

Other Remarks

1.3 Additional Remarks.

Breed.guide Remarks.

A special note on textures. Before you start saying, Oh no, I'm limited on textures for my game, certain things should be clarified. The most limiting for textures in floors snd ceilings. Since you have only 16 of them, you should think of 16 main textures you can use for them. It's not so bad, since you normally wont use much differing textures for Floors and Ceilings anyway. Most architectures I know, don't exactly go crazy with those. You can have loads of wall textures, however. Wall Textures have their own set, and since you can choose (to some extent) what part of each graphic to map to a wall, you can align multiple wall textures together, side by side, in a single wall graphic.

The gamelinker also allows for you to set up a number of different wall files to be used with the game.

Textures, for objects. So, you have 32 of them. Well, one thing is, that you can put multiple tiles for a single texture, if your size is not too demanding. Also, you, under many cases, can benefit by thinking of textures as materials. Instead of making a few wood textures, you can make only 1 or two wood textures ever. This is because, not only can you select exactly what part of the Texture to use as an object polygon surface, you can use the same function of the editor to do realtime warping or bending of a texture, onto the surface. So if you have a straight-lined wood texture, you can bend it to a more circular or curvy texture for use on the object. The editor, and game support realtime texture warping. A Very nice feature, which greatly expands the amount of textures you really have.

Well, I hope you can make good use of this guide file. As usual, with an initial release, things may have been left out. Others are planned for a future release. One of the things, is to make the guide file adhere to the 79 column max standard. Also, more than the use of Retrace to get back to the Index. I've been toying with the possibility of adding links to little audio digitized sound bites, voice, and perhaps some graphical picture links. Well, the last stuff isn't of priority, right now. Perhaps during a time I have more time on my hands.

Any special content suggestions can be emailed to me, at the address below. Also, if you have found some neat special abilities, of the editors, not mentioned

```
    here, let me know; I'd love to include them.

    --- Mike Bromery.
    Email: mbrome1@gl.umbc.edu
```

1.4 Setting it all up.

```
Thankyou for having the patience to install this bumper selection of
fixes and additional files. Listed below are the problems encountered
by you, the long-suffering public, and how they have been addressed, but
first, here's how to install the new files:

  Click on the install icon!

  This process will take a while, and some of it is interactive,
  so please be patient!

  NOTES:

  When prompted for a texturemap filename, enter
  "ab3:includes/newtexturemaps"

  This will allow the game to function without modification of the
  game link file.

*=====================================================================*


    Common problems encountered:
    ----------------------------


1. LEVED DOESN'T WORK!

  Leved needs floor graphics in ab3:graphics/floors. These files
  are now present. It also needs ab3:includes/betpts, which has
  also been added.

2. THE GAMELINKER WON'T LOAD THE SUPPLIED TEST.LNK FILE!

  Sorry about this one. When loading the file, go to the directory
  and TYPE IN 'test' as the file to load, NOT 'test.lnk'. The
  '.lnk' is appended automatically by the program. THIS ALSO
  APPLIES WHEN SAVING, so save it as 'blah:blahblah/TKG-EDITORS/includes/test'.

3. THERE'RE NO EXAMPLE GRAPHICS TO GET ME STARTED!

  Due to space problems with the game, it was not feasible to
  include the (extremely large) data files for aliens and so
  on with the editors. Supplied is an example alien in
  LIGHTSOURCED format, and a future tutorial will deal with
  creating more aliens in this format.

4. THERE'RE NO GUNS SUPPLIED!
```

There is now! The plasma gun is supplied converted and included in the game link file, with two other weapons available in the 'vectobj/' drawer for loading into the 3D editors and converting yourself.

5. DO YOU SERIOUSLY EXPECT ME TO FIGURE OUT HOW TO USE ALL THIS?!?

   Yes.

6. THE INSTALLER DOESN'T MAKE AN ICON FOR THE DRAWER!

   Well, it would if you had installed Workbench properly as it copies your default icon from ENV: when you install.

Before you can leap headlong into using this professional-quality suite of editing software (hem hem), there are a few other things you have to do...

1. Run 'LEVED', load level A, calculate the clip file and save the level again.

   Note: calculating the clip file is done by (in this case with level A) clicking on the icon, in the second row of Leved, which has a smaller sized picture representation of this:

   ```
   ---------
   | 1 | 2 |
   ---------
   ```
   Then Enter A (For level A). This will generate the Clip file for Level A.

2. Run the 'linkup' program and type 'a' for the level.

3. Type 'pm a' and 'pgc a' into the shell prompt, which packs all the level data for loading into the game.

   P.S. -- In order for this to work properly, as it is, you will want to make sure everything in the ab3:s and ab3:c dir is in your path. You may copy the files to C: or something or add the path to those directories. Also, the script files, in the ab3:s dir should be made executable by this command:
   ```
    protect <file> s add
   eg. protect pm s add
   ```
   This will ensure that you can operate these instructions as written.

4. To run the game on the demo level, type 'tkg' whilst in the ab3: directory. The game will run, prompt for copy protection, then you can play the level as normal.

   Note: Almost as normal. Problem is that the ab3: directory isn't the only assign needed anymore. Make sure you assign tkg1: tkg2: and sfx: to the ab3: directory. Included, hopefully, with the lha release of this Breed.guide is a sample AB3DII-TKG script file, which assumes you made the ab3: assign already (perhaps in your s:user-startup).

5. Read the tutorial in the magazine to create your own levels!

Note: Another good note about running the game (especially a full game), is that it is quite Chip Ram heavy. I would give at least 1.64 or 1.65 megs of Chip ram open, to properly run the game, and Load/Save game positions. As noted by Team17 the game may or may not have problems with your current version of MCX, if it is running. (It was noted, as MCX is so popular, to avoid shouts from loads of MultiCX running Amiga users.)

VERY VERY IMPORTANT NOTICE
==========================

Although it is stated in the documentation that it is possible to modify the palette used in the game, this is currently not the case. All graphics you draw will be converted automatically into this palette (with greater or lesser degrees of success depending upon what colours they use. A future 'fix' will deal with this problem and allow you to modify the palette to suit yourself.

AND FINALLY:
============

You have my most sincere apologies for the hassle you have experienced in getting these editors to work. The missing files were entirely my responsibility and I hope the frustration at having to wait for a fix has not dampened your enthusiasm for designing and playing levels for AB3DII.

Oh, and one last thing:

The difficulties many of you have experienced after installing the 4-meg version are due to the last-minute implementation of the new menu system which on certain speeds of accellerator and when run in certain ways INEXPLICABLY FAILS TO WORK! If you have any difficulty, run the game from shell, or in extreme cases from boot-no-startup. Remember to type

```
  execute AB3DII-TKG
```

from within the AB3DII-TKG directory.

** STOP PRESS **

I've just heard of a bizarre possible 'fix' for the above problem:

Copy (not move) the 'shell' icon from sys:system/ to the AB3DII_TKG drawer.
Select this new icon and select 'information' from the 'icons' menu of workbench.
Change the 'from=s:shell-startup' to 'from=BLAH:BLAHBLAH/AB3DII-TKG/AB3DII-TKG'
then double click the icon and it should (allegedly) work. Hmmm.

Bye for now!

Andy Clitheroe.

1.5 Change Level Lighting

```
[HOW TO:

        Change Level Lighting
        =====================

  One of the most important aspects of AB3D as far as atmosphere is concerned is  ↩
     how the levels
  are lit.

  In AB3D, the walls and floors are smoothly shaded between brightness's defined  ↩
     at their corners.

  Each zone contains four 'layers' of brightnesses; a lower floor layer, a lower  ↩
     ceiling layer,
  an upper floor layer and an upper ceiling layer. Brightnesses are defined at the ↩
      corners of
  zones, i.e. at the points which you selected to create the zone in the first  ↩
     place.

  NB: In the diagram below, where the two zones share two points, remember that  ↩
     you must define
      brightnesses for the two shared points in BOTH zones individually. This may  ↩
         sound odd and
      long-winded, but it allows you to have sharp edged bright and dark areas.


              SHARED POINTS
                     |
                                         V

            *--------------*--------------*
            |              #              |
            |              #              |
            |              #              |
            |              #              |
            |              #              |
            |              #              |
            |              #              |
            |              #              |
            *--------------*--------------*

  The procedure for defining a layer of brightnesses is as follows:

  1. Select the appropriate icon. They are arranged in a line of four in the
     middle row, fourth from the left end. Listed left to right, they are:

     UPPER ROOF BRIGHTNESS
     UPPER FLOOR BRIGHTNESS
     LOWER ROOF BRIGHTNESS
     LOWER FLOOR BRIGHTNESS

  2. Notice that as you click, three sliding bars appear, labelled BRIGHT, ANIM  ↩
     and DIST. These
     values are clipboard values which will be pasted onto points.
     BRIGHT:
    The default brightness value, ranging from -20 to +20. NEGATIVE numbers represent the
    same on-screen brightness as positive numbers, BUT they contribute nothing to the
    illumination of aliens, vector objects (e.g. the player's gun) and so on. Also,
    remember that numbers close to 0 are BRIGHT, whereas numbers towards -20 or +20 are
    DIM.
    The reason for the positive/negative values is simple: the light-sourceing of objects
    doesn't work well without it! For example, a brightly lit room with glowing panels
    on the ceiling means that the light should be coming from above, and so aliens and
    guns should similarly be lit from above. In this case, the floor brightnesses should
    be set to -1, and the roof brightnesses to +1, so the room appears uniformly lit BUT
    aliens are lightsourced from above.

     ANIM:
      There are several pre-defined lighting animations available:

      0   = no animation
      1-5   = 5 out of phase steady pulsing anims
      6-7 = Flickering effects.
      8-15    = Undefined at this time.

     DIST:
      This is used to describe how far towards the animation the brightness of the point is
      'pulled' from its default brightness. 0 means it doesn't animate at all, 15 means it
      animates completely, independent of the brightness it is given.

     Set these values as desired.

  3. Select a zone in which you want to change some brightnesses with the RIGHT mouse button, and
     click on the point(s) you wish to set to the new values with the LEFT mouse button.
     If you wish to grab brightness settings from a point, move the cursor near it and press 'g'.

OTHER TOOLS:
------------

  There are several pre-defined settings which can be pasted into entire zones with one keypress.
  Simply select the zone with the right mouse button and press one of the following keys:

  1 Zone bright, lit from roof
  2 Zone bright, lit from floor
```

```
3 Bright floor, dark roof
4 Bright roof, dark floor
5 Zone dark, lit from roof (i.e. what little light there is is coming from above )
6 Zone dark, lit from floor (i.e. what little light there is is coming from below)
7 Zone roof glowing (each roof point glowing in one of the five pulsing anims)
8 Zone floor glowing
9 Zone Glowing in sync (i.e. corresponding floor and ceiling points are glowing with
       the same animation)
0 Zone Glowing out of sync (all floor and ceiling points pulse randomly and independently)
```

1.6 Add Aliens To Levels.

```
HOW TO:

        Add Aliens To Levels
        --------------------

  Before reading this, make sure you understand the process of placing objects as laid out
  in 'HOW2-Place_Objects_In_Levels'.

  Adding aliens rather than objects is very similar. Just select the smiley face as usual and
  click on the 'Object Type' line until it shows 'Alien'.

  You can select what type of alien to paste in the same way as selecting which object to paste,
  and you can specify doors and lifts to be held until the alien is killed in the usual way also.

  Instead of 'start frame' which is not applicable to aliens, you have a 'team' option. A value
  of '-1' represents 'no team', i.e. the alien works alone. Entering a different value allows you
  to place several aliens on the same 'team'. Team members share information about  your whereabouts;
  the team member who saw you most recently will tell all the other team members (activating them
  if they are out of range of you) where it saw you. You can set up lots of nice things like
  ambushes in this way (See HOW2-Create_Challenging_Levels for tricks and tips on using things
  like teamwork to cause the player grief)

  Also, by clicking in the edit area with the RIGHT mouse button, you can select the control point
  which the alien will head towards as soon as the game starts. In this way you can set up initial
  alien movements to your satisfaction (e.g. making a squad of aliens run towards where the player
  starts, killing him unless he hides before they get there).
```

1.7 Add Doors and Lifts To Levels

```
HOW TO:

        Add Doors and Lifts To Levels
        =============================

  No level would be complete without some doors (locked or otherwise) and lifts ( ↩
      stuck or
  otherwise) for the player to negotiate.

DEFINING A DOOR:
----------------

  Select the 'Define door' icon. This is fifth from the left on the bottom row,  ↩
      and depicts
  a red door with an arrow pointing upwards beneath it to show it opening.
  You will be presented with a bewildering array of options, including:

  DOOR NUM: X
    You are currently editing door X, which is a letter from A to P (16 doors).  ↩
        Pressing
    '<' and '>' with the cursor in the edit area scrolls through these values.
    NB: Holding down RSHIFT when pressing '<' or '>' not only moves to the  ↩
        previous or
    next door, but also copies all the settings from the current door to it. This  ↩
        is useful
    when most of the doors in your level are of the same type.

  Stays Open For (50=1 sec)
    This value represents the amount of time the door will remain open for after  ↩
        it reaches
    its full open height. By experimenting with this value you can create  ↩
        combinations of
    doors and switches which require the player to run across the map before the  ↩
        door
    closes again! Click on the value to change it.

  Opening Speed
    Opening and closing speeds can be more or less anything you like, but for  ↩
        example,
    an opening speed of 16 is fast, whilst an opening speed of 1 is very, very  ↩
        slow (useful
    for very big, heavy doors) Click on the value to change it.

  Closing Speed
    Same as above, but in the opposite direction.

  Opening SFX:
    This is the sound effect triggered by the door being opened. Click and type  ↩
        the number
    of a sound effect (0=no sound) which you have loaded into the game linker.
  Closing SFX:
    Same as above but triggered by the door starting to close.

  Open SFX:
    Similar to above, but triggered by the door REACHING its fully open state.

  Closed SFX:
    Ditto, but triggered by the door reaching its completely closed state.

  RAISE CONDITIONS:
    Plr touch+Space
      The player must come into contact with the door and press the space bar to  ↩
          trigger
      it into opening.
    Plr Touch
      The player only needs to touch the door to trigger it into opening.
    Bullet Touch
      NOT IMPLEMENTED IN CURRENT VERSION; DO NOT USE.
    Alien Touch
      NOT IMPLEMENTED IN CURRENT VERSION; DO NOT USE
    On Timeout
      As soon as the door becomes unlocked, or finishes closing, it will begin to
      open again.
    Never
      Fairly obvious; the door will never open. Not a lot of use, but included for
      completeness.

  LOWER CONDITIONS
    On Timeout
      Once the timer has expired (see above) the door will begin to close once  ↩
          more.
    Never
      Once open the door will never begin to close again.


PLACING A DOOR ON THE MAP
-------------------------

  Note: Place doors between open zones, to save time. Because, the idea of  ↩
      defining
         walls to raise and lower with the door (in adjacent zone(s)) is  ↩
             automatically
               placed. That is, you don’t need o have a wall there, already. If you ↩
                   do, that
               first click will erase the wall, then the second click will put a  ↩
                   wall to lift
               with door, there. The editor makes this simple to see. White are  ↩
                   just walls.
         Grey, means a separation of Zones, but the zones are connected with no  ↩
             walls
               inbetween. (I.E. Connected Zones) Blue means that it is a wall that  ↩
                   will lift with
               the corresponding Door. This also goes for Lifts.

  1. Click on the ’define door’ icon as above.
```

2. Select the zone which you want to act as the door (i.e. its ceiling will rise and fall) with the RIGHT mouse button.

3. Press 'r'. The letter of the door will appear roughly in the middle of the zone.

4. This bit requires a bit of thought. Remember that the walls of zones face INWARDS, not outwards, so you will now have to specify which walls in the surrounding zones you want to rise and fall with the ceiling of the door zone:

```
     1
  ----------------*-------*-----------------
       #       #
       #       #
       #       #
       #       #
    ZONE X   #       #  ZONE Y
       #  DA   #
       #       #
       #       #
       #       #
       #       #
            ----------------*-------*-----------------
                   2
```

   In this case, where a corridor is interrupted by a door, zones X and Y each have ONE wall which must appear to rise and fall with the ceiling marked 'DA'.

   To select these walls, click in, for example, zone X with the RIGHT mouse button. Then click with the LEFT mouse button on the point at the LEFT END of the wall you wish to change.

   NB: The 'LEFT END' refers to the end of the wall which would be on the left when you looked at it in the game from within zone X; i.e. point 1 in the diagram.

   Notice that the wall selected becomes blue. You can deselect a wall just by clicking on it again.

   Repeat this process for all other zones bordering the door zone, (e.g. zone Y and point 2).

5. You have done it! The door will now rise and fall between the defined floor height and the defined roof height of the zone in which it was placed. You can now place objects which hold the door shut until collected, or put in aliens which hold it shut until killed, or anything!

NB: The list of settings for the door in the bottom right of the screen is NOT a clipboard; these

```
      are the ACTUAL settings for the door. You can change the settings at any time before or after
      placing the door, and the changes will take effect automatically.

DELETING A DOOR:
----------------

  Sometimes it may be necessary to delete a door from the map. The ONLY safe way to do this, boys
  and girls, is to follow this procedure:

  1. Click on the 'DEFINE DOOR' icon, and use < and > to reach the door you want to delete.

  2. Use the technique described above to toggle OFF all the blue walls in zones around the door.

  3. Select the door zone itself with the RIGHT mouse button and press 'r'. The door letter should
     vanish.

  NB: It is not possible to delete a zone which is either a door, or contains a wall bordering a door.
      I made this so because every time I deleted a zone containing or bordering a door whilst
      programming AB3D, the level corrupted irretrievably. I never did find out why, so I put in a
      safety feature which prevents zones being deleted in such a situation. You MUST delete the
      door before deleting neighbouring zones or the door zone itself.

DEFINING A LIFT:
----------------

  This is very, very similar to the above process, with the following changes:

  Additional Note: One of these changes, not in the original text is that, to place the Lift in
                       a zone, you must type 'f' and NOT 'r', to make a zone into a lift.

  LOWER CONDITIONS:
    Plr touch+Space
      The player must come into contact with the lift and press the space bar to trigger
      it into coming down.
    Plr Touch
      The player only needs to touch the lift to trigger it into descending.
    On Timeout
      As soon as the lift becomes unlocked, or finishes rising, it will begin to
      lower again.
    Never
      The lift, once raised, will never descend.

  There is an additional setting, defining where the lift begins (at the top or the bottom of its
  movement) which can be set by pressing 't' or 'b' with the cursor in the edit  ↩
      area.

  Once a lift zone has been defined, in the same way as above, there is one extra  ↩
      piece of
  information you must give: the maximum height to which the lift rises. Define  ↩
      this by selecting the
  lift zone with the right mouse button....

        Additional Note: The rest was cut off from the original text. But here's  ↩
            the addition.
                         After you selected the lift zone, press 'h' and you will  ↩
                             be requested for the
                         height. Type in a number, for where you want the maximum  ↩
                             height of the lift.
                         REMEMBER: Your lifts go up, so you must subtract. Don't  ↩
                             make a lift, where the
                         floor is 0 and the lift max height is 144. That is wrong. ↩
                              It goes UP to -144.
                         Don't forget your negative Heights. If you test your  ↩
                             level, and you see a wierd
                         crazy hole, where your lift should be, with strange  ↩
                             choppy graphics inside,
                         that means, you mad ethe lift where the maxiumum height  ↩
                             was BELOW the floor.
                         A mistake, on my part, when I was first designing a level ↩
                             . However, you have
                         been forewarned.
```

1.8 Add Objects To Levels.

```
HOW TO:

          Add Objects To Levels
          =====================

  NB: This document ONLY covers instructions for placing or changing OBJECTS. See  ↩
      the file
  HOW2-Add_Aliens_to_levels for details on placing aliens.

  To add an object, click on the small smiley face in the bottom left corner of  ↩
      the icon strip.
  You will be presented with an extra set of options representing the object ' ↩
      clipboard' from which
  you can paste multiple copies of a single object into the level. Click on the ' ↩
      Object Type'
  line of the options until it reads 'Object'.

  Adding objects to levels is quite simple, but there are a few things to bear in  ↩
      mind:

  1. Any object can act as a 'key' for lifts and doors. To avoid difficulties, try ↩
       to think
     of keys as things that 'hold doors shut' and 'hold lifts still' until they  ↩
         are removed
```

from the map by collecting or destroying them, or until they are activated (e.g. a switch). The doors and lifts held by an object are represented by two lists of letters, A-P. Click on each letter to toggle it from dark to highlighted. Highlighted letters represent doors or lifts which will be held by that object. Any number of doors and lifts can be held by a single object, and similarly any number of objects can hold any one door or lift. The player must remove or activate them ALL before the door/lift will operate.

2. Any object can trigger a text message when it is activated, collected or destroyed. This is represented by the 'Text:' number. Click on this number and type the number of the text string you wish it to trigger (See 'HOW2-Define-In-Game-Text' for details).

3. Clicking on the 'Type' line with the left and right buttons of the mouse scrolls forward and backwards through the 30 possible objects you can define in the game linker. Objects MUST be defined in the game linker before you can use them in levels.

4. The 'START FRAME' of an object is a very, very useful function indeed. If you read the document 'HOW2-Define_New_Objects' you will know that it is possible to define loops of animation within the animation data of each object. Using that, along with the START FRAME, allows a single object to have many different graphical appearances in the game without needing any more than ONE object defined in the game linker! For example, the keys in AB3DII were defined as a single object with 13 frames of animation. Each frame of animation was edited to display a DIFFERENT colour/number of key (e.g. red or green or blue or yellow etc.) and also to point at ITSELF as the 'next frame' to be used. There are therefore 13 independent animations within the animation definition for 'key'. The 'START FRAME' value was set before pasting a key into the level. This value represents which frame of animation the object will first be drawn using. Since in this case all frames pointed to themselves as 'NEXT' frames, this also meant that the object would remain in that frame throughout its life! In effect there were 13 different LOOKING keys in the game, but only one 'KEY' object was ever defined!

5. In the game, the computer keeps track of all objects and aliens which you can see or are near to, and only moves and animates those. So that aliens can chase you, there is also a short 'timeout' so that an object or alien continues to be active for a few seconds after you move out

of range. The 'PERMANENT CALCULATION' toggle disables this feature for the object you paste, and forces the computer to animate and control it constantly.

NB: The computer will never ignore the 'holding shut' of doors, no matter how far away objects become, so don't worry about doors springing open as soon as you move out of range of the key; it won't happen!

6. Because of the availability of gantries and the like, it is important to specify whether the object is to be placed in the upper or lower part of a zone. The 'Start in Up/Lo rooms:' toggle allows you to specify where to place the object.

7. Finally, to place an object somewhere, just click in a zone with the LEFT mouse button. If you specified in the gamelinker that the object should be locked to the nearest wall, the object will appear, stuck to the wall in the zone nearest where you clicked.

OTHER TOOLS:

If, when you have the 'place object' icon selected, you move the cursor over an existing object and press 'g', you will grab the settings of that object into the clipboard. You can then modify them if necessary and paste them back into the same, or indeed any other object, by moving the cursor over it and pressing 'p'.

The icon immediately to the right of 'place object' is 'move object'. If you wish to move an object, select this icon then click on the object to move with the RIGHT MOUSE BUTTON, then on the position you wish it to move to with the LEFT mouse button.

The icon to the right of the 'move object' icon is 'delete object'. Clicking in the edit area with this icon selected will delete the nearest object to your cursor.

1.9 Change the Global Palette

HOW TO...

Change the Global Palette
=========================

AB3DII runs on a 256 colour screen, which requires a palette. There are three files supplied containing the original AB3DII palette information in three different formats:

EXAMPLES/INCLUDES/IFFPAL

This is in the form of an IFF palette and can therefore be loaded into most commercial art packages.

EXAMPLES/INCLUDES/256PAL

This file contains the same information in a format specifically for use by the editors and conversion programs supplied with AB3DII

EXAMPLES/INCLUDES/256PALETTE

This file is the one loaded by the game itself when run.

Say you are drawing a wall graphic. You can do this in one of two ways:

Method ONE:

Using an art package with a 256 colour screen, load in the IFFPAL file as a palette. Draw your graphics WITHOUT making any changes to the palette, and save it to disk. Then convert the picture down to 32 colours (most art packages will allow you to do this automatically) and save it again under a different filename. You will notice that the colours you have used from the 256 palette will be present in the smaller 32 colour palette. You can now convert the graphics as normal (see HOW2-Convert_Wall_Graphics).

This method usually gives best results because the colours you use are actually present in the palette the game will use.

Method TWO:

Draw your wall on a 32 colour screen using whatever palette you wish, and save it to disk. The conversion program will match each colour you use to the best one available in the global palette stored in 256PAL.

This method may not work very well if you try to draw a wall using colours not present or very different from those in the global palette.

The palette supplied with AB3DII was used in the game but may not be to everyone's taste. If you want to change the palette, this is the procedure you must follow:

RECOMMENDED METHOD:

```
  Use your favourite art package to edit the IFFPAL file (make sure you are  ↩
      editing a copy
  in your own directory, not the original file), then use the PALETTECREATOR  ↩
      program to
  produce equivalent 256PAL and 256PALETTE files.

ALTERNATIVE METHOD:
  Use the PALETTECREATOR program to edit the palette and save out the 256PAL and  ↩
      256PALETTE
  files. It is not possible to save out IFFPAL files from the PALETTECREATOR  ↩
      program.

HINT:

  When designing your own palette, make sure there are plenty of shades of each  ↩
      colour, from
  the brightest one you ever use down to very dark or black. Remember, the game  ↩
      only has YOUR
  palette to work from; if you only have a bright red in the palette it's going to ↩
       look very
  wrong when it tries to draw something dark red (for example a red object in a  ↩
      dark room).

NB:
  If you convert some graphics and then change the 256PAL file so that some  ↩
      colours you have used
  are in different places or gone altogether, you MUST reconvert all the graphics  ↩
      you have
  previously converted. The reason for this is that the old converted graphics won ↩
      't otherwise
  know that the file has changed and if, for example, you have changed a red  ↩
      colour to a green,
  anything previously converted which used the red colour will suddenly start  ↩
      drawing it in
  green instead! For this reason it is a VERY good idea to decide upon a palette  ↩
      before beginning
  to draw and stick to it.

          SPECIAL EFFECTS:
          ================

It is possible, with only a little effort, to enhance the appearance of your game  ↩
    by modifying the palette
file the game itself loads, without changing the IFF palette file or the 256pal  ↩
    file used by the converters.
How does this work? Easy. The converters will load your graphics and make  ↩
    decisions about which colours should
be used from the global palette not only for the graphics as you drew them, but  ↩
    also at all the
darker shades in which they may be visible. Hence if you have a grey wall, the  ↩
    conversion program will find
the nearest greys available when the wall is viewed in full brightness down to  ↩
    full darkness.  It is very likely
that you will have a grey-scale in your palette somewhere (and I suggest that you  ↩
    set aside
AT LEAST 32 colours for this purpose, because grey is the most difficult colour to ↩
     shade smoothly). Assuming
```

that you do, use the palette creator to change the high and low extremes of the grey scale, just slightly.
For example, you could make it shade from a dark, slightly red-tinted grey to a very bright, slightly yellow-tinted white, rather than simply dark-white. Save out the modified 256palette file BUT NOT the 256pal file.
What happens when you run the game? Well, since the conversion programs are none the wiser about your tampering, they will still match up dark grey to what is now dark, reddish grey. In the game, this means that everything which is grey shades off to a slight red tinge in the distance, which works wonders for the atmosphere of the game. AB3DII used this technique, changing the grey scale to be slightly green at the dark end, which accented the alien surroundings.

1.10 Change Floor and Ceiling Heights.

HOW TO:

```
    Change Floor and Ceiling Heights
    ================================
```

Since levels are a little dull without things like stairs, cliffs, low tunnels etc., you will no doubt want to alter the default floor and ceiling heights of your level. Here is how you do it:

```
1. Overview
-----------
```

The floor and ceiling height changing routines work on a 'clipboard' principle. The height of a floor or ceiling is read into the clipboard by clicking with the RIGHT mouse button on a zone. The height in the clipboard can then be changed and pasted back into the zone, or as many zones as you wish, by clicking within them with the LEFT mouse button. This allows you to speedily change the floor or ceiling heights of many zones.

There are four heights which can be changed in each zone. These are:

Upper Roof Height
Upper Floor Height
Lower Roof Height
Lower Floor Height

The icons to click on to edit each of the above are arranged on the top row of the icon bar in the level editor, starting on the fourth icon from the left. Remember, the icon bar can be scrolled left and right using '(' and ')' on the keypad.

Normally, only the Lower Roof and Lower Floor heights are defined. The other two are initialised to 5000, meaning 'DO NOT USE'. The Upper Floor and Roof heights are used in the definition of gantries or two-storey rooms (see below).

NB: VERY VERY VERY IMPORTANT! In AB3DII, heights are measured DOWNWARDS! This means that a height of -16 is HIGHER UP than a height of 0, which is HIGHER UP than a height of 16.

NB: VERY IMPORTANT: Floor and ceiling heights are measured as absolute values. The floor is not measured by its distance from the ceiling, and neither is the ceiling measured by its distance from the floor. If the floor height in a room is 128, and the ceiling height is 64, then the height of the room is 128-64=64.

Rooms default to being 128 high, with the floor at 0 and the roof at -128.

2. General Usage

Click on the icon corresponding to the value you wish to change. If you wish to grab the corresponding height of a zone, click in that zone with the RIGHT mouse button.

Use the + and - keys to alter the height in the clipboard. Holding RSHIFT down allows you to change them in increments of 8 rather than increments of 1. Note that due to a bug, either in AMOS or the keyboard hardware itself, you cannot hold down RSHIFT and + on the keypad to make the key auto-repeat (e.g. if you want to change a height by a great deal). If you wish to autorepeat, press RSHIFT and the + key on the main keyboard.

When the desired height has been achieved, click with the LEFT mouse button in any zones whose height you wish to set to the new value.

Heights can be positive or negative, but REMEMBER that heights are measured DOWNWARDS!

3. Creating... Stairs

Usually, stairs go down in steps of 16. There's no reason you can't make them steeper or shallower than this, but 16 is a good value.

4. Creating... Gantries

This is a feature unique to AB3D on the Amiga. It is possible to define overhead walkways or

```
corridors for the player to use:


UPPER roof height ->  ---------------- -------- <-- Roof height of neighbouring  ↩
   Zone
              .
              .
              .
              .
UPPER floor height -> ________________.
                              |
                      SOLID       |
                      WALKWAY       |
       Roof Height -> ________________|
                          .
                          .
                          .
                          .
      Floor Height -> ---------------- -------- <-- Floor height of neighbouring ↩
          zone


This is a simple procedure for creating a gantry.

1. Define the area in which the gantry is to be created. REMEMBER: Gantries can  ↩
   only be
   built from ENTIRE ZONES. You CANNOT HAVE a big room made of a single zone,  ↩
      with a gantry
   across it. You must have something like this:

   *--------*----*-------------------*
   |        #    #                   |
   |        #    #                   |
   |   A    # B  #        C          |
   |        #    #                   |
   |        #    #                   |
   *--------*----*-------------------*

   If you want to be able to see both under and over the gantry in zone B,  ↩
      ensure the floor
   and ceiling heights of zones A and C are far enough apart (about 256 apart is ↩
       adequate).
   Now, select 'DEFINE ROOF HEIGHT'. Click with the RIGHT mouse button on zone A ↩
      , to grab the
   height.
   Now select 'DEFINE UPPER ROOF HEIGHT'. Click ONLY in zone B with the LEFT  ↩
      mouse button. The
   Roof height from zone A will be copied into the Upper roof height of zone B.
   Repeat the above, but with the Floor Height of zone A and the Floor height ( ↩
      not the UPPER
   floor height) of Zone B.

   You have now defined the upper ceiling and lower floor heights of B. It  ↩
      remains to define the
   in-between values to describe the walkway.
```

You should still have the Floor height of A in the clipboard. Click on ' DEFINE UPPER FLOOR HEIGHT' and then adjust the value upwards using 'RSHIFT -' until it is about 128 above the floor height. Click with the LEFT mouse button in zone B. This defines the upper surface of the walkway.

Select 'DEFINE UPPER ROOF HEIGHT' and click in zone B with the RIGHT mouse button. Select 'DEFINE ROOF HEIGHT' and adjust the value to something slightly further down than the upper floor value you just defined. A difference of 8 is a very thin walkway; greater differences mean greater thicknesses of gantry.

There is no reason why the floor height and upper roof height of B should be the same as the floor and roof heights of A and C, so feel free to change any of them to create any setup you wish. For example, the following yields a pair of corridors which cross but are invisible from one another:

```
      ___________________
      |                 |
      |                 |
      |                 |
      |     UPPER B     |
      |                 |
      |                 |
      |_________________|
    .                 .
    .                 .
______________________________________________________
    .                 .
    .                 .
    .                 .
  A .     LOWER B     .         C
    .                 .
    .                 .
    .                 .
______________________________________________________
```

You can put gantries in as many zones as you wish, even ones next to each other, which allows for all sorts of nice effects, eg:

Floating platforms
Two storey rooms with a connecting hole or lift or flight of stairs.
Bridges over water
Multi-level staircases (tricky but possible; check out level 11 of the first AB3D)

1.11 Change the Level Names

HOW TO:

Change Level Names

Levels are described briefly by their names on the option screen when playing the game,
and also saved game positions are referred to by the name of the level you are about to
start when you load the position.

To change these names, load the gamelinker program and load the link file
in the usual manner.

Select

LEVELS -> CHANGE LEVEL NAMES

then click on level names and type to change them. Click on 'DONE' when you have finished
and save the link file again.

1.12 Convert Floor Ceiling Graphics.

HOW TO:

Convert Floor/Ceiling Graphics

For this to work, you must place all your floor and ceiling tiles in a single directory.
There must be exactly 16 of them, and they must be called

```
FLOOR.1
 . . .
 . . .
FLOOR.16
```

If you don't have enough different tiles, it's a good idea to have one 'default tile'
which you make 16 copies of and over-write when you create new ones. This way you can
test out the ones you HAVE drawn.

Run the 'FLOORCONVERT' program.

Use the requester to select the DIRECTORY in which the floor tiles are located. DO NOT
select an individual floor tile, but make sure their names are visible in the requester.

```
  The program is fully automatic, loading and converting the graphics, then  ↩
     matching colours to
  the ones in

  AB3:INCLUDES/256PAL

  for this reason, make sure you have set up the assign for AB3: to the directory  ↩
     you are building
  your game in.

  When the program is finished, just select the save name under which you wish to  ↩
     save the
  floor tile file, and the program will self-terminate after saving is complete.

  See document 'HOW2-Incorporate_Floor_Ceiling_Graphics' for instructions on  ↩
     incorporating them
  into your game.
```

1.13 Convert Texture Graphics

```
HOW TO:

      Convert Texture Graphics
      ========================

  For this to work, you must place all your Texture tiles in a single directory.
  There must be exactly 32 of them, and they must be called

  Texture.1
   . . .
   . . .
  Texture.32

  Textures you want to be used as Glare textures, must be called

  Glare.1
         . . .
         . . .
  Glare.32

  Note:
   Now you can only have 32 of any of these Object using Textures. That is, the  ↩
      index
        of a Texture and Glare can not collide. I.E. If you have a Glare.2, then  ↩
           there CAN
        NOT be a Texture.2.

  If you don't have enough different tiles, it's a good idea to have one 'default  ↩
     tile'
  which you make 32 copies of and overwrite when you create new ones. This way you ↩
      can
  test out the ones you HAVE drawn.
```

You will then run the 256texture command. to Convert the Textures. It will load these texture and then produce a filerequestor. The file requester should point at the ab3:Includes directory. The filename should be newtexturemaps (which is the default).
This is what you want to save it as.

You will then want to use the 256brighten command, in the ab3: dir. This will prompt you to put in the texture file. It IS the ab3:includes/newtexturemaps file , which was generated from the 256texture command. This may take a little time. Just a few minutes on my Blizzard IV 030/50 (noted by Mike B.).

1.14 Convert Wall Graphics.

HOW TO...

```
    Convert Wall Graphics.
    ======================
```

Before graphics can be used in AB3DII they must be converted into a special, compressed format.

NOTE: Wall graphics MUST obey the following rules:

1. They MUST be in 32 colours, IFF format. The palette you use is up to you,, but they must be drawn on a 32 colour screen.

2. They MUST be a power of 2 in height i.e. 16 (min),32,64,128 (max) pixels high. The height of the screen they are drawn on is unimportant, but the graphics themselves must comply to this rule.

It is a good idea to keep all the wall graphics you are going to use in a separate directory, for example:

```
MYDIR
  WALLGFX
```

and to keep all the converted files in another directory e.g.

```
MYDIR
  WALLINC
```

If you have a picture which you want to convert, e.g.

```
MYDIR
  WALLGFX
```

```
    examplewall

then you should run the "WALLCONVERT" program provided. This program will prompt  ↩
    you for
a filename to load. Select your file, and follow the on screen instructions. You  ↩
    will be
asked for various parameters including height in pixels and width in pixels. Once  ↩
    conversion
is complete you will be prompted for a filename to save, e.g.:

MYDIR
  WALLINC
    examplewall

This file is in the correct format to be used in AB3DII. The program will  ↩
    automatically append
'.256wad' to the filename you supply, so if you click on an existing file to over- ↩
    write,
make sure you delete the '.256wad' from the end before selecting OK or pressing  ↩
    RETURN.
```

1.15 Create Challenging Levels

```
HOW TO:

          Create Challenging Levels
          =========================

  There are an awful lot of features in TKG, and how to use them all together may  ↩
      not seem obvious
  at first. This document tries to show how to use features to create levels which ↩
       are challenging
  for the player by being well-designed, not simply by having lots of aliens and  ↩
      no health packs!

1. Layout of the level
----------------------

  Before putting mouse to screen, draw out your level on paper. The worst way to  ↩
      do this is to
  just draw a room, then a corridor, then another room with a couple of corridors, ↩
       then another
  room and so on. Levels drawn like this tend to be dull and samey. Instead, try  ↩
      one or more of
  these techniques:

a) Designing to a purpose.

  Think of a function the level might serve. It could be a hospital, or a training ↩
       camp, or a sewage
  system, or perhaps two areas linked together. Once you have an idea in mind,  ↩
      layout ideas will
  usually follow. For example, the sewage system would have a lot of tunnels, with ↩
       steps or lifts
  leading up into various buildings, perhaps.
```

```
b) Designing to fit a space

  Draw an outline on you paper, a 'perimeter wall' which your level must fit into. ↩
      Make it excitingly
  chunky and purposeful. I find this helps a lot; having to fit your level into  ↩
     the shape prevents
  it sprawling like a very old bag of potatoes.

c) Designing to suit an alien

  When you design your aliens you are probably going to have one or more big,  ↩
     tough critters that
  are going to severely impede the player's progress. Such aliens should have  ↩
     levels custom-designed
  to suit their abilities.

2. Useful level building blocks
-------------------------------

a) Did I just step on something...?

  A very useful thing to define is an 'invisible trigger'. This can be used to set ↩
      off messages and
  trigger doors and lifts, preferably with lots of aliens behind/on them. Simply  ↩
     place the trigger
  somewhere the player has to go and arrange for it to dispense aliens at the most ↩
      inconvenient
  moment possible.

b) They're appearing out of thin air! AAARGGHH!

  This is a good one. When the player picks up a gun, shoots a particular alien or ↩
      whatever, this
  can trigger several aliens to teleport from their 'homes' outside the level,  ↩
     into the room the
  unlucky player is standing in. How to do it? Well, first of all, define several  ↩
     alien 'houses'.
  An alien 'house' is a pair of zones not connected in any way to the rest of the  ↩
     level:

  *-------*
  |       |
  |       |
  |       |
  *=======*
  |       |
  |       |
  |       |
  *-------*

  Define one zone as a lift and the other as a teleporter. Put your alien(s) on  ↩
     the lift, making
  sure that their 'permanent calculation' flag is ON. Make sure the lift starts at ↩
      the TOP of its
  movement, and that at that height it is ABOVE the ceiling height of the  ↩
     teleporter (this prevents
  flying aliens making their way into the teleporter before the lift is triggered).
  Arrange the alien/gun/health pack/key or whatever so that it triggers the lift into falling to the
  same height as the teleporter. Place a control point in each zone and link them together. This will
  ensure the aliens will wander onto the teleporter as soon as they can.

c) Smithers, did you just press that big, red button?

  Even more annoying to the player than tripping invisible triggers is when he voluntarily pulls a
  large, inviting lever only to discover that it removes the only barrier between him and an alien
  horde. Don’t over-use this one.

d) We’re going to need BIGGER guns.

  The ’immensely powerful-looking gun’ trick is so ludicrously annoying that it should only ever be
  used once, if at all. Define a really really impressive looking weapon and place it somewhere
  obvious but extremely difficult to reach. Only when the player, after hours of trying, finally
  manages to pick it up, will he discover that it does, in fact, fire ping-pong balls. For real
  humiliation make it trigger a horde of aliens which the player will confidently attempt to dispatch,
  failing miserably and probably dying in the process.

e) Neighbourhood Watch.

  You’ve all got one; a weird neighbour who ’keeps himself to himself’ and watches people through
  his bedroom window. These slightly disconcerting people can now be put to good use in TKG. Stick
  one up in an alcove, preferably near a door the player will come through. Make him part of a team,
  the rest of whom are waiting in a room down a side corridor. As soon as the watcher sees the player,
  it will trigger the others. If the watcher can shoot, too, it will usefully distract the player
  while the bulk of the squad come running. However, if the player is good (or has played the level
  before), he will pop out and shoot the watcher before he is seen, then run down and lob a grenade
  or seven into the room with all his mates in.

f) I’m gonna get my big brother on you *sniff*.

  Fill a maze of corridors with small, useless aliens and one big, dangerous one, making them all part
  of the same team. If the player is seen, all the little aliens will converge on him, reporting
  his position to the big guy.
```

1.16 Define Bitmap Alien Animations.

HOW TO:

Define Bitmap Alien Animations
==============================

This document assumes that you have read and understood the 'HOW2-Define_Aliens' document and that you have got as far as selecting the

[DEFINE ANIMATION FRAMES]

option from the alien statistics screen.

You will see a list of all the animations which an alien will perform.

NB: The diagonal views (Front-left, Front-right, Back-left and Back-right) are NOT IMPLEMENTED in the current version, so don't bother drawing or defining animations of aliens in those directions.

All the animations have the same format, but their usage varies as follows:

The 'movement' animations refer to the various directions it is possible to see the alien from as it walks/flies around. These animations are played looped, ie when the last frame is reached, animation continues from the first frame.

NB: It is VERY important that all the direction animations have the same number of frames. This is because, to improve continuity, the program DOES NOT reset the animation when your view of an alien changes. In other words, if it reaches frame 5 whilst walking away from you, then turns left, it will start from frame 5 as it walks left.

The other, 'action' animations are one-shot anims played when the specified action occurs.

DEFINING AN ANIMATION SEQUENCE

Click on the animation sequence you wish to define.

You will see a complicated looking line of values at the top, and buttons for adding a new frame to the end of the list and deleting the last in the list. As for objects and bullets, when a new frame is added, it takes on the values of its predecessor, so set up invariants, for example which graphic file the frames are from, before adding frames.

The values, from left to right, are:

```
0-19  Frame number

GF: Graphic File. This is the number of the file the frame is to come from.  ↩
    Click on the
  value and then on the name of the file you have previously loaded.

FN: Frame number. This is the actual frame within the graphic file which will be ↩
     drawn.
  an 'R' in front of the number indicates the frame will be flipped horizontally ↩
       before
  drawing on-screen. This is handy as you can have one set of frames for left  ↩
      AND right
  movement, with the computer flipping them as needed.

SW: Scaled Width. This is the width to which the frame will be scaled (Not  ↩
    including the
  effects of distance). Must be 255 or less. Click to change.

SH: Scaled Height. This is the height to which the frame will be scaled (Not  ↩
    including the
  effects of distance). Must be 255 or less. Click to change

VO: Vertical offset. Normally the frame is drawn centred on the alien's positon, ↩
     however with
  this value you can offset each frame vertically by a set amount, perhaps to  ↩
      make it
  appear to bob up and down. Click to change.

FX: This is the number of the sound effect which will be played whenever the  ↩
    alien reaches
  this frame of animation. Click on the number to get a list of currently loaded ↩
       sound
  effects, and click on an effect OR at the top of the screen for no effect.  ↩
      With this
  you can easily synchronise footstep sounds with the correct frames of  ↩
      animation on-screen,
  or shooting effects with the precise frame the gun appears to fire!

AC: Action Star. Clicking next to this toggles on and off an 'action star'. If  ↩
    an action
  star is present, the alien will perform an action, such as moving or shooting. ↩
       The precise
  nature of the action is dependant upon which animation sequence you are  ↩
      defining. Using
  the action stars you can make an alien walk slowly by having a sequence like  ↩
      this:

  0 GF: 1 FN: 1 . . . . . . . . . . . . . . . . AC: *
  1 GF: 1 FN: 1 . . . . . . . . . . . . . . . . AC:
  2 GF: 1 FN: 1 . . . . . . . . . . . . . . . . AC:
  3 GF: 1 FN: 1 . . . . . . . . . . . . . . . . AC:
  4 GF: 1 FN: 2 . . . . . . . . . . . . . . . . AC: *
  5 GF: 1 FN: 2 . . . . . . . . . . . . . . . . AC:
  6 GF: 1 FN: 2 . . . . . . . . . . . . . . . . AC:
  7 GF: 1 FN: 2 . . . . . . . . . . . . . . . . AC:
```

Assuming frames 1 and 2 are part of a walking sequence, the above would instruct the computer to display each frame for 4 time units, and only move the alien in the first of the four time units, thus preventing the alien 'moonwalking' between frames.

You can make an alien fire a specific number of times by including several repetitions of the shooting animation, with an Action Star next to the appropriate frame in each repetition.

AI: Animation Instruction. Usually, this contains a downward arrow 'v', indicating that the animation will proceed to the next frame in sequence (or to the first if the last one is reached). However, it is sometimes very useful to be able to repeat a certain sequence a set number of times, or perhaps even a random number of times. There are therefore another 3 Animation Instructions:

```
S X Store number X in counter.
R X Add a random number between 0 and X to the counter
D X Subtract 1 from the counter, and jump to frame X if the result was NOT zero.
```

Hence you could make an alien howl, then take a random number of steps between 10 and 20 before howling again, like this:

```
0 GF: 1 FN: 5 . . . . . . . . . . . FX: 3 AC:   AI S10
1 GF: 1 FN: 5 . . . . . . . . . . . FX: 0 AC:   AI R10
2 GF: 1 FN: 1 . . . . . . . . . . . FX: 0 AC: * AI  v

    * WALKING ANIMATION

9 GF: 1 FN: 4 . . . . . . . . . . . FX: 0 AC: * AI D 2
```

Click and follow the instructions to change the AI value of a frame.

AUX: Auxilliary object frame number. This selects a frame defined in the auxilliary object for that alien to be displayed over the top of the alien. Click on the value and you will be shown the animation sequence you defined for the auxilliary object. Simply click on the appropriate frame OR at the top of the screen for no frame.

XO: Auxilliary object X Offset. The amount by which the centre of the auxilliary object is offset from the centre of the alien graphic. Useful for gun barrel flashes, so you can make gun flashes appear over the top of the gun they appear from, wherever the gun may be!

YO: Auxilliary object Y Offset. As above, but vertically rather than horizontally.

1.17 Define Aliens for your game.

HOW TO:

```
      Define Aliens
      =============
```

This is probably the most arduous task facing the prospective game designer, as it requires the most data to be input.

Select ALIENS -> DEFINE ALIEN STATS.

You will be shown a list of aliens to define. To change their name, click on the existing name and type the new one. You can copy an aliens name and characteristics by clicking on the appropriate 'COPY' button and clicking on the target space.

You can also change the scale of an alien by clicking on the button near the bottom of the screen, then clicking on the alien in question, then typing the percentage by which you wish to scale the alien.

NOTE: The game DOES NOT REMEMBER any previous sizes you have defined for aliens, so if you scale an alien up by 200%, you CANNOT scale it down again by scaling it by 100% later on. You must scale it by 50% to reduce it to the first size you gave it.

Clicking on DEFINE CHARACTERISTICS of an alien brings up the following menu:

1. GRAPHIC TYPE:

This can be one of three main types:

BITMAP
The alien will use standard bitmap graphics converted by the OBJECTCONVERTOR program. The alien will only be lightsourced so far as to be made brighter in bright rooms and dimmer in dark rooms. This option is the one used in all other games of this sort.

VECTOR
If you have designed animation frames for a vector alien, you can specify that you wish to use them by selecting this option.

LIGHTSOURCED PALETTE 1/2/3/4
These options can be selected, but you won't be able to use them because the support programs for producing lightsourced graphics were not included in this

package. Look out for a coverdisk with tutorials and support programs soon! ↩
For
now, you’ll have to be content with the options above.

2. [DEFINE ANIMATION FRAMES]

If you are designing a BITMAP alien, consult the file

HOW2-Define_Alien_Animations

otherwise check out

HOW2-Define_Vector_Alien_Anims

AFTER YOU HAVE READ THE REST OF THIS FILE!!!

3. Default Behaviour

Aliens can either walk, or fly. This can be indicated by either
WALK RANDOMLY or FLY RANDOMLY
being selected (click to toggle)

4. Reaction time

This is the amount of time the alien must see you for (uninterrupted) before it ↩
will notice
you and take appropriate action. If you shoot it and it doesn’t die, it will ↩
immediately
spot you. This value is measured in 50ths of a second. If your aliens are too ↩
easy, try
decreasing this value rather than increasing hit points or whatever.

5. Default Movement Speed

This is the speed which the alien wanders around at when it can’t see you and is ↩
not attacking.
5 is an average sort of speed, 10 is zippy and more than 10 is ludicrous.

6. Response Behaviour

This is what the alien will do when it notices you; either:

```
CHARGE      Run straight at player, biting and munching him when close enough.
CHARGE TO SIDE    Run towards player, keeping out of line of fire, biting when ↩
    close enough.
ATTACK WITH GUN   Execute the ’attacking frames’ animation, with each ’action ↩
    star’
       corresponding to a bullet being fired.
```

There are three more options, identical to the above but for flying aliens.

7. Response Movement Speed

Only used in charging aliens, this governs how quickly the alien will charge.

8. Response Timeout

You can make the alien 'get bored' and stop charging after a certain amount of time. IF you don't
want this to happen, type in a nice big number like 10000, representing over three minutes!
Value is measured in 50ths of a second.

9. Damage Taken Limit -> Retrest

NOT IMPLEMENTED IN THIS VERSION

10. Damage Inflicted Limit -> Followup

NOT IMPLEMENTED IN THIS VERSION

11. Followup Behaviour

Once the alien has executed its attack animation, or has timed out of its response counter,
it will 'followup'. In this mode it can:

PAUSE
  Stop dead for a moment, watching you
Approach
  Move straight towards player (but inflicts no damage when it reaches him).
Approach To Side
  Same as above, but keeping out of line of fire.
Approach flying
  Same as above but flying
Approach to side flying
  Do I really need to explain?

12. Followup Movement Speed

The speed at which the alien will approach you (if at all)

13. Followup Timeout

Assuming the alien can still see you once this timer expires, it will attack again.
This value is measured in 50ths of a second.

14. RETREAT BEHAVIOUR/RETREAT MOVEMENT SPEED/RETREAT TIMEOUT

NOT IMPLEMENTED IN THIS VERSION.

15. NUMBER OF HIT POINTS

The 'toughness' of the alien. Think about how tough you want the alien to be relative to
each of your guns. Do you want it to keel over after a single shotgun blast? Then make its
hit points less than the damage done by your shotgun rounds. Want it to soak up rockets and
laugh chestily? Then make its hit points HUGE, so that the damage of the rocket PLUS its
explosive force (the maximum amount of damage it can do to an alien or yourself) is much

less.

16. Physical Height

How tall is your alien? To make it walk on the ground, a good height to use is TWICE the SH (Scaled Height) of the animation frames you are using. Notice that aliens CANNOT WALK into rooms whose floor-ceiling height is LESS than this value, and CANNOT MOVE in rooms which are too short if you plonk them in there to begin with.

17. Minimum distance to walls

Small or thin aliens can get quite close to walls, so a value of 0 will suffice. But if it’s a fat beasty, a value of 1 or 2 will prevent it squeezing unrealisically into narrow tunnels.

18. Projectiles Ejected Upon Death

Clicking on this will bring up a list of projectiles (bullets) which the alien can splurt out when it dies. Interestingly, you can make it spit out actual bullets (as long as they are VISIBLE ones) when it dies, thus allowing sneaky aliens to hemmorhage grenades as their innards rupture, which makes the player’s situation just a little dicey...

Alternatively, you can choose to click on ’Select Alien to spawn instead’ and pick a type of alien. Upon death, the alien will spawn FOUR of the aliens you selected. Please don’t be silly and make aliens spawn themselves, or make them spawn other aliens which in turn spawn them back. You’ll only get annoyed playing the game as hordes of replicating aliens engulf you mercilessly.

19. Auxilliary Object Type:

Click on this and select an object whose frames have been set up correctly.

By ’correctly’, what do I mean? Well, if you define a DECORATION object, and make it look like an animation of a gun barrel flash, you can use it as an AUXILLIARY OBJECT. At any point in any animation sequence for the alien, you can make the game paste a frame from the auxilliary object somewhere over the top of the alien graphic. This is how the muzzle flashes and glaring lights of the various alien guards were produced. Auxilliary objects may be BITMAP or GLARE/SHADOW, but not VECTOR. If you are not using any auxilliary graphics on the alien, you don’t have to define this value at all.

20. Torch Brightness

Aliens can carry 'torches', illuminating the level in front of them. A value of 30 is a reasonable torch brightness, whilst a value of 0 indicates no torch.

21. Bullet Source X and Y offsets.

These value indicate how far from the centre of the alien its bullets should originate. This is largely a matter of trial and error, but remember that negative values mean the bullet comes from the left or the top, whilst positive values mean the right or the bottom.

1.18 Add Teleporters to Levels.

HOW TO:

Add Teleporters to Levels

Teleporters are a very useful feature of the game. You can make them obvious, with trademark glowing floors and ceilings or particular graphics, or invisible, so players wandering unwarily down a corridor will find themselves chucked into the middle of a bloodbath!

Teleporters are also very easy to implement. Just bear a few things in mind:

1. Teleporters move the player to a different part of the level, but retain his height relative to the ground; ie if he jumps into the teleporter he will appear at the same height above his landing spot as he was above the teleporter, even if the teleporter is higher or lower than the arrival spot. Because of this, you must ensure that a teleporter has NO HIGHER A ROOF relative to the floor than the arrival point, otherwise it might be possible for the player to materialize in the middle of the roof!

2. Teleporters don't teleport bullets, but they DO teleport aliens, so be careful if you don't want that to happen. On the other hand, if you DO want it to happen, you can place a control point in the teleport, and one at the arrival point, and link them in ONE direction only. Aliens will then happily wander into teleports if they know they can get to you by doing so, but won't attempt the return journey.

3. Teleporters won't teleport something into the space occupied by another object. This is handy in some respects, because you can lay traps where if the player collects a gun or medipac

or similar object, aliens can teleport to the location. In other respects it isn't so useful
because if an alien or an object is too near the arrival point, the teleport won't activate when
the player walks into it. Try to stop aliens from wandering around arrival points as it can
be very frustrating for the player when the teleport he needs to proceed further doesn't work!

DEFINING A TELEPORT:

In the level editor, with your level loaded, click on the TELEPORT icon. This looks like a
curving red arrow leaving a point and touching down elsewhere, and is located midway along the
bottom row.

Select the zone you wish to make a teleport with the RIGHT mouse button. Click on the arrival
point with the LEFT mouse button. If you click with the left mouse button OUTSIDE the map
area, the teleport will be cancelled or deleted.

1.19 Define Bullet Types

HOW TO:

Define Bullet Types
===================

No doubt you will want aliens to shoot at you and vice-versa, and before you can make them do
this, you have to define some bullet types. Aliens fire selected bullet types, as do the player's
guns, and ammo you collect also gives you specified ammounts of each bullet type to shoot.
The term 'bullet' is a bit of a misnomer and should really be 'projectile' since it can refer
to bullets, mines, grenades, rockets, and, curiously enough, the bits of splatch that aliens
spurt when shot.

Load your link file into the game linker as usual.

Select BULLETS -> DEFINE BULLET DATA

To define the NAME of a bullet, click on the existing name with the LEFT mouse button and
type in your new name. You can now click on 'EDIT BULLET STATISTICS' and begin creating your
projectile.

1. DAMAGE TO TARGET PER BULLET

```
  This refers to the amount of damage the target (ie an alien, an object or you yourself) will
  take on being struck directly by the projectile. A value of zero disables collision with
  other objects, useful for stuff like the blood which spurts out of aliens.

2. VISIBLE BULLET OR INSTANT EFFECT
-----------------------------------

  Bullets can either be visible, like grenades or plasma bolts, or invisible and instantly
  effective like shotgun blasts or rifle bullets.

3. GRAVITY VALUE
----------------

  ONLY AVAILABLE FOR VISIBLE BULLETS

  This indicates the effect gravity has upon the projectile. Grenades should have a value of
  about 20 or so, with 0 representing gravity having no effect. NOTE that negative values are
  possible, and will cause the bullet to fall towards the roof!

4. BOUNCE OFF WALLS Y/N
-----------------------

  ONLY AVAILABLE FOR VISIBLE BULLETS

  Toggles bouncing off walls on/off. NB: Bullets which are subject to gravity will be slowed
  down by contact with walls, but those unaffected by gravity (Gravity Value = 0) are assumed
  to be some sort of lazer weapon and are not slowed down.

5. BOUNCE OFF FLOORS/CEILINGS
-----------------------------

  ONLY AVAILABLE FOR VISIBLE BULLETS

  Toggles bouncing off floors and ceilings. Slowdown rules same as above.

6. BOUNCE NOISE
---------------

  NOT IMPLEMENTED IN THIS VERSION.

  This setting has no effect at present, although updates may rectify this absence.

7. BULLET LIFETIME
------------------

  ONLY AVAILABLE FOR VISIBLE BULLETS
```

Bullets may last forever (a value of -1 denotes this) or may fizzle out or detonate after a certain time. This value is measured in 50ths of a second. Projectiles not defined as bouncing will detonate or fizzle prematurely if they strike a wall, and will detonate in any case if they strike an enemy target (either you or an alien depending upon who fired it).

8. EXPLOSIVE FORCE

ONLY AVAILABLE FOR VISIBLE BULLETS

When the bullet runs out of time or hits an alien or wall, it may explode. A value of zero represents no explosion, with about 40-50 representing a reasonably powerful blast.
NOTE: The visual effect of an explosion is created by producing several copies of the 'impact' frames of animation around the central blast area, so if you have an explosive bullet, ensure that the impact frames are of a suitable type to look right when the explosion occurs.

9. MOVEMENT SPEED 0-7

ONLY AVAILABLE FOR VISIBLE BULLETS

The speeds of bullets are not measured linearly, but in powers of two. Thus '0' is very, very slow (in practice, not movement at all, particularly if you use it for mines which drop and hit the ground, slowing down and therefore stopping entirely) and 7 is, well, 128 times faster than that! A speed of 7 is extremely fast, with speeds of 5 and 6 more suitable for standard fireballs and so on.

10. IMPACT NOISE

ONLY AVAILABLE FOR VISIBLE BULLETS

Click on this and you will be presented with a list of the currently loaded samples. Select one to make the bullet trigger that noise when it hits, or click at the top of the screen to select 'none'.

11. [DEFINE BULLET ANIMATION]

ONLY AVAILABLE FOR VISIBLE BULLETS

```
Here you can define the animation of your protectile. Bullets are not 'sided'; ie they look
identical from all directions, and hence you only need define one animation sequence for each
of them.

Clicking here will present you with, initially, a single, incomplete frame description and
some extra buttons. These are the values you can change in each frame:

GF: Graphic file to read the frame of animation from. Click on the number to pick
  one of your object-graphic filenames.

FN: Frame to draw from within the selected graphic file. Click here with the LEFT mouse button
  to simply type a number in, or click with the RIGHT mouse button to be shown the
  ANIMATION FRAME SELECTION SCREEN. See the document

  'HOW2-Use_Animation_Frame_Selection_Screen'

  for details, as this screen is re-used several times in the program.

SW: Scaled Width and Scaled Height of graphic on-screen. Values similar to those defined in
SH: the animation frame in the graphic file are about right, although you can squash any
  size and shape of graphic into any size and shape of box. SW and SH have a maximum value
  of 255.

VO: Vertical offset. By varying the vertical offset, you can make bullets weave up and down
  as they animate. For this to work, you will need several frames of animation.

  HINT: If you want to create an effect similar to that seen in Breathless, define only
  one frame of animation for each bullet.

BV: Brightness Value. Bullets can be defined to 'glow', lighting up their surroundings.
  A value of 20 gives a reasonably bright glow.

ADD FRAME:
  Adds a frame to the end of the list, copying all the values from the previous last in
  the list to the new last in the list. It is therefore a good idea to set up the first
  frame with the graphic file, scaled width and height and so on, then add more frames and
  just tweak the values you want to change afterwards.

DEL FRAME:
  Removes the last frame in the list.
  The bullet, when fired, will cycle through these animation frames until it  ↩
      explodes, or fizzles,
  or otherwise expires, at which point it will proceed into:

12. [ DEFINE IMPACT ANIMATION ]
-------------------------------

  Same as above, except the animation will be played through only once, before the ↩
       bullet disappears.

13. BULLET GRAPHIC TYPE
-----------------------

  There are three options, which should be selected according to what sort of  ↩
      graphic you have
  drawn:

  BITMAP:
    Simplest option of the three, this just draws the frame onto the screen scaled ↩
        appropriately.

  GLARE:
    Uses the 'GLARE/SHADOW' rules to either brighten or dim the screen behind. See

    HOW2-Draw_And_Convert_Glare_Graphics

    for instructions.

  ADDITIVE TRANSPARENCY:
    Uses a different routine again to 'add' the graphic in a transparent fashion  ↩
       to the
    graphics already on-screen, thus giving a 'firey' or 'smokey' or 'blazing  ↩
       plasma death
    bolt' impression. See

    HOW2-Draw_And_Convert_Additive_Transparency_Objects

    for instructions.

14. IMPACT GRAPHIC TYPE
-----------------------

  Same as above, but for the impact frames.
```

1.20 Define Floor Damage Values and Sound Effects.

```
HOW TO:

      Define Floor Damage Values and Sound Effects
      ============================================

  Each floor tile you draw can have a damage value and sound effect assigned to it ↩
      , so the player
```

can actually hear the difference between each tile!

To define these values, load the link file into the game linker in the usual way.

Select MISC -> DEFINE FLOOR DAMAGE VALUES

Click on each tile number and type the damage inflicted on the player by that tile. The amount you enter will be taken off the player’s health every couple of seconds, so don’t make it too big an amount; about 10 for a really damaging floor, or 4 or 5 for a slightly damaging floor.

Click on DONE when you’ve finished, and select MISC -> DEFINE FLOOR FOOTSTEP SOUNDS.

Click on the number of each tile and select the sound effect most appropriate to the floor tile.

Click on DONE, and save the link file!

1.21 Defining Objects for your game.

HOW TO:

```
            Define Objects
            ==============
```

The term ’Object’ in AB3DII has quite a broad meaning. Basically, anything which isn’t a bullet, a player, an alien or background (walls, floors and ceilings) is an object. There are several main types that an object can be, and this determines its behaviour:

1. Collectable

   The object may be collected by the player. When this happens, the player will recieve whatever ammunition or other supplies the object is defined to provide, and the object will remove itself from the map. Once so removed it will also cease to hold any doors and lifts.

2. Activatable

   The object can be ’activated’ by the player being within its activation radius and pressing the space bar. When this happens the object switches to ’active mode’. The player can switch it back, or it can ’timeout’ and return to its inactive state. Whilst active the object will cease to hold any doors or lifts.

3. Destructable.

   The object can be damaged by player bullets and explosions. Once destroyed it will cease to hold any doors or lifts, and will no longer impede the player.

4. Decoration

   The object cannot be interacted with in any way and simply sits there playing through its animation sequence.

To define an object, select OBJECTS -> DEFINE OBJECT STATS.

Click on an empty space or an object you want to redefine with the LEFT mouse button, and type the new name. Clicking on the object with the RIGHT mouse button instead takes you into the actual definiton screen. On this screen are several options, some of which are only visible for certain types of object. They are listed here in top-bottom order:

1. Type Of Object

   Click to cycle through the four types above.

2. Graphic Type

   Click to cycle through:

   BITMAP
     Simplest type of object; uses ordinary bitmap graphics
   POLYGON
     An object of this type uses frames of animation from 3D objects designed using the ANIMEDITOR program.
   GLARE/SMOKE
     The object is drawn transparently, brightening or darkening the graphics behind it.
     See HOW2-Draw_Glare_Objects for instructions.

3. Deactivate after..

   ONLY FOR ACTIVATABLE OBJECTS

   This value is the time in 50ths of a second between the player activating the object and it deactivating automatically.

3. Number Of Hit Points

   ONLY FOR DESTRUCTABLE OBJECTS

   The number of damage points the object can take before it is destroyed.

4. Explosive Potential

   ONLY FOR DESTRUCTABLE OBJECTS

```
  The explosive force with which the object detonates upon its destruction.

5. Impassable in default state

  DESTRUCTABLE AND DECORATION OBJECTS ONLY

  Does the object obstruct the player until destroyed? This value gives you the  ↩
     answer.

6. [ Define Default state Animation ]

  Clicking here provides you with a screen containing information about the  ↩
     default animation
  sequence for an object. The values andd buttons you can use are as follows:

  BITMAP AND GLARE OBJECTS
  ------------------------

  GF: Graphic file to read the frame of animation from. Click on the number to  ↩
     pick
    one of your object-graphic filenames.

  FN: Frame to draw from within the selected graphic file. Click here with the  ↩
     LEFT mouse button
    to simply type a number in, or click with the RIGHT mouse button to be shown  ↩
       the
    ANIMATION FRAME SELECTION SCREEN. See the document

    'HOW2-Use_Animation_Frame_Selection_Screen'

    for details, as this screen is re-used several times in the program.

  SW: Scaled Width and Scaled Height of graphic on-screen. Values similar to those ↩
      defined in
  SH: the animation frame in the graphic file are about right, although you can  ↩
     squash any
    size and shape of graphic into any size and shape of box. SW and SH have a  ↩
       maximum value
    of 255.

  VO: Vertical offset. By varying the vertical offset, you can make objects bounce ↩
      or wobble up
    and down as they animate. For this to work, you will need several frames of  ↩
       animation.

  NF: Next Frame Number. Click on this and then select the 'next frame' you want  ↩
     the object to
    change to after the one you just clicked on. In this way you can make objects  ↩
       perform
    a loop within their animation, or as is more usual, stop on the last frame  ↩
       rather than
    looping back to the beginning (eg switches).

  ADD FRAME:
    Adds a frame to the end of the list, copying all the values from the previous  ↩
       last in
    the list to the new last in the list. It is therefore a good idea to set up the first
    frame with the graphic file, scaled width and height and so on, then add more frames and
    just tweak the values you want to change afterwards.

  DEL FRAME:
    Removes the last frame in the list.

  VECTOR OBJECTS
  --------------

  VF: Vector file to read the frame of animation from. Click on the number to pick
    one of your vector filenames.

  FN: Frame to draw from within the selected graphic file. Click here with the LEFT mouse button
    to simply type a number in.

  AO: The amount the object should turn this frame. A value of 8192 represents 360 degrees.

  VO: Vertical offset. By varying the vertical offset, you can make objects bounce or wobble up
    and down as they animate. For this to work, you will need several frames of animation.

  NF: Next Frame Number. Click on this and then select the 'next frame' you want the object to
    change to after the one you just clicked on. In this way you can make objects perform
    a loop within their animation, or as is more usual, stop on the last frame rather than
    looping back to the beginning (eg switches). This is also extremely useful for GUN objects.
    See the file 'HOW2-Define_Gun_Objects' for fuller instructions after reading the rest of
    THIS file.

  ADD FRAME:
    Adds a frame to the end of the list, copying all the values from the previous last in
    the list to the new last in the list. It is therefore a good idea to set up the first
    frame with the graphic file, then add more frames and just tweak the values you want to
    change afterwards.

  DEL FRAME:
    Removes the last frame in the list.

7. [ Define Active/Destroyed Animation ]

  Similar to 6. except that this is the animation which will be played when the object is activated
  or destroyed. It is ALSO the sequence displayed if this object is defined as the ON-SCREEN OBJECT
```

of a player gun.

8. [Define Guns/Jetpack/Shield Given]

   Clicking here brings a list of toggles as to what the object provides the player with when he collects or activates it.

   NB: Jetpack and shield are not implemented in this version.

9. [Define Ammo/Fuel given]

   Clicking here gives a list of what ammo and health is given by the object when collected/activated. Hence a 'gun' object would give the player the appropriate gun, and also some of the ammo that the gun fires.

   NB: Fuel is not implemented in this version.

10. Collision cylinder radius

    This is the distance a player has to be within to collect or activate the object. About 100-150 is usually about right.

11. Collision cylinder height

    This is the height within which the player must be to collect the object. About 100 is usually ok.

12. Position vertically relative to:

    Either the FLOOR or the CEILING can be toggled between here, and the object will be positioned next to the one selected. Thus the Vertical Offset in the animation data is important to pull the object up or down into the room.

13. Lock to nearest wall?

    If this is enabled, then the object will 'snap' to the nearest wall when placed in the level and will face outwards into the room (vector objects only)

14. Activation/Collection/Destruction sound effect

    Click here to set the appropriate sound effect for whatever action is appropriate to the type of object.

1.22 Defining a Player's Appearance

HOW TO:

DEFINE PLAYERS' APPEARANCE
==========================

To define the appearance of PLAYER 1 and PLAYER 2 in the game, load the link file
into the game linker as usual, then select

PLAYERS -> DEFINE PLAYER STATS

Click on the player you wish to define and select an 'alien' you want him to look like in
the game. Simple as that.

1.23 Define Player Guns.

HOW TO:

Define Player Guns
==================

The player is likely to feel a little defenseless without guns to shoot, and indeed you MUST
have the first weapon defined before you can play the game (the default game link file
contains one pre-defined weapon, the shotgun).

Select GUNS -> DEFINE GUN TYPES

You will be shown a list of 10 guns, along with various items of information:

<GUN NAME>

Click to type a new name for the gun.

<BULLET TYPE>

Click, and then select a bullet from the list shown. This is the type of bullet
the gun will fire.

Bullets Fired/Shot

Click and type a number from 1 upwards. Only used by instant effect bullets, so
shotguns can be double-barrelled and fire two bullets (either of which has an
independant chance of hitting the target).

Delay between shots

A value in 50ths of a second representing the delay between the gun firing successive

times. Thus a machine gun might have a value of 5, a rocket launcher a value of 50 or
100. Of course, you could always have a rapid-fire rocket launcher....

Gun SFX

Click and select a sound effect for the gun to trigger when you shoot.

On Screen Object

Click and select an object whose COLLECTED/ACTIVATED/DESTROYED frames are set up to
work as the gun. See

HOW2-Define_Objects

for instructions.

1.24 Drawing Floor Graphics

HOW TO:

Draw Floor Graphics
===================

Floor graphics MUST be drawn individually on 32 colour screens. They must be drawn as a
64*64 pixel square located in the top left hand corner of the screen you draw them on.
Each texture picture can have its own 32 colour palette independant of the other textures.
There must be 16 of them before they can be used in the game, although you can always make
16 copies of a 'default' texture, and replace them as you design new ones.

They MUST be located, for compatability, in a single directory, and they MUST be called

FLOOR.X

where X is a number from 1 to 16.

1.25 Draw Texture Graphics.

HOW TO:

Draw Texture Graphics
=====================

Texture graphics MUST be drawn individually on 32 colour screens. They must be drawn as a
64*64 pixel square located in the top left hand corner of the screen you draw them on.

```
  Each texture picture can have its own 32 colour palette independant of the other ←
      textures.
  There must be 32 of them before they can be used in the game, although you can  ←
     always make
  32 copies of a 'default' texture, and replace them as you design new ones.

  They MUST be located, for compatability, in the directory:

  AB3:GRAPHICS/TEXTURES/

  And they MUST be called either

  TEXTURE.X or GLARE.X

  where X is a number from 1 to 32.

  GLARE textures are special. You MUST draw them in the 32 colour palette provided
  on the editor disk:

  GRAPHICS/GLAREPAL

  When drawing a GLARE texture, anything left BLACK is perfectly transparent.
  Anything drawn in shades of WHITE will appear 'added' to the background in the  ←
     game, so
  dim white will look pale and transparent, whilst bright white will look solid  ←
     and, well,
  bright.

  Shades of RED represent progressively more opaque shades of shadow, with BRIGHT  ←
     RED representing
  almost pure black. Smoke and other effects can be created using these shades.

  REMEMBER: you MUST use the palette provided for the effect to work!

  GLARE textures can only be used on GLARE polygons, and TEXTURE textures can only ←
      be used on NON-
  GLARE polygons.
```

1.26 Incorporate Converted Floor/Ceiling Gfx.

```
HOW TO:

      Incorporate Converted Floor/Ceiling Graphics
      ============================================

  PLEASE NOTE:
    You do not need to follow this procedure if you have converted your floor  ←
       graphics and
    saved them over the existing texture file. Only if you have created a new file ←
        or changed
    the name or path of the old file do you need to proceed further.

  1. Run the GAMELINKER program.

  2. Select 'LOAD LINK FILE' from the menu and select
```

MYDIR/INCLUDES/TEST.LNK

3. Select GRAPHICS->SET FLOOR TILE FILENAME

4. Select the file you wish to load, presumably created using the 'FLOORCONVERT' utility.

7. Select 'SAVE LINK FILE' and select the file

MYDIR/INCLUDES/TEST.LNK.

The new floor tile file will now be available in the level editor and the game itself!

1.27 Incorporate Object Graphics.

HOW TO:

Incorporate Object Graphics
===========================

Before any aliens or pickups can be defined, you must provide the graphics for them, and before
you can do that, they must be converted using the appropriate supplied utilities. Please refer
to the document

HOW2-Convert_Object_Graphics

for full instructions on creating these files.

Assuming you have converted some object graphics and wish to use them in the game, here is how
to go about incorporating them:

1. Load the GAMELINKER program and load the 'TEST.LNK' file in the usual way.

2. Select 'GRAPHICS -> SET OBJECT GFX FRAMES'

3. You will see a list of currently loaded graphics and a blank space for inserting a new one.
   Either click on one to replace, or click on the blank space. In either case, you must use the
   LEFT mouse button. Use the requester which appears to select EITHER the '.ptr' or
   '.wad' file produced by the convertors.

4. Assuming all is well, the filename will appear in the list on the screen. Now you must define
   the frames of animation within the file. Click on the name but this time with the RIGHT mouse
   button.

```
5. You will be shown a strip of the graphics, in the height you converted them.  ↩
    Do not worry if
   the palette is not right; this is a failing of AMOS. The following will be  ↩
      shown also,
   at the bottom of the screen:

   X,Y      These are the mouse coordinates within the graphic file. Clicking  ↩
      with the
        LEFT mouse button with the cursor within the graphic file area of the
        screen copies the X and Y value shown into the FRAME LEFT and FRAME TOP
        values below.

   SCROLL LEFT
   SCROLL RIGHT   These buttons can be clicked upon to scroll the graphic file  ↩
      across to
        display parts of it currently off-screen.

   FRAME NUMBER   This is the frame number you are currently defining, from 0 to ↩
        31.

   PREVIOUS/NEXT FRAME  Changes the current frame number.

   FRAME LEFT   These values represent the left side, top side, width and height ↩
        of the
   FRAME TOP    current frame. You can click upon and type to change any of them ↩
      , the
   FRAME WIDTH    idea being that they enclose each seperate 'frame' of  ↩
      animation you
   FRAME HEIGHT   have designed.

   Copy width/height  These options allow you to quickly copy the width and  ↩
      height you have typed
   to previous/next into the current frame into all previous or succeeding  ↩
      frames.
   frames

   GENERATE FRAMES  If you have drawn and converted the graphics in line with  ↩
      the guidelines
   FROM GRAPHIC DATA  set down in the document 'HOW2-Convert_Object_Graphics',  ↩
      you can shortcut
        this whole process and click on this button. The position and size of  ↩
           all
        the frames will be defined. However, you cannot use this function if  ↩
           some
        frames are different sizes, or located at different heights within the
        graphic file. In such cases you must simply type or click all the values ↩
             in
        manually.

   DONE     return to graphic file list
```

1.28 Incorporate Sound FX.

```
HOW TO:
        Incorporate sound effects
        =========================

  Sound effects in AB3DII are a little restricted, because of the 8-channel  ↩
     feature. They
  will only ever be played at about 8000hz, which is usually referred to as 'C2'  ↩
     in packages
  like Octamed or Protracker. They must also be a multiple of 200 bytes in length. ↩
      Finally,
  they must cover NO MORE THAN HALF of the full scale deflection available. In  ↩
     other words,
  if you have an ordinary sample, you must HALVE its volume before you can use it  ↩
     in AB3DII.
  Oh, and the sample must be saved in RAW format.

  Assuming you have, or have created, some samples in this format, you can  ↩
     incorporate them:

  1. Load the link file into the GAMELINKER as usual.

  2. Select SAMPLES -> Set SFX Filenames

  3. Click on any of the sample spaces and select your sample using the requester.

  4. Click on 'DONE' and save the link file!
```

1.29 Incorporate Converted Texture Graphics.

```
HOW TO:

      Incorporate Converted Texture Graphics
      ============================================

  PLEASE NOTE:

    If you change texture graphics, and save over your existing texture file when  ↩
       converting
    them, you DO NOT NEED to follow this proceedure. Only if the name of the  ↩
       texture file
    has changed do you need to proceed.

  1. Run the GAMELINKER program.

  2. Select 'LOAD LINK FILE' from the menu and select

  MYDIR/INCLUDES/TEST.LNK

  3. Select GRAPHICS->SET TEXTURE FILENAME

  4. Select the file you wish to load, presumably created using the '256brighten'  ↩
     utility.

  7. Select 'SAVE LINK FILE' and select the file

  MYDIR/INCLUDES/TEST.LNK.
  The new texture tile file will now be available in the level editor and the game ↩
      itself!
```

1.30 Incorporate Vector Objects

```
HOW TO:

        Incorporate Vector Objects
        ==========================

  If you've spent the last three weeks thrashing around with the unwieldy and  ↩
     unfriendly
  vector object editors (sorry Charles :->) you'll probably want to see them in  ↩
     the game, whether
  as objects or aliens or guns or whatever.

  Before you can use them, you have to load them into the game link file:

  1. Load the link file into the game linker as usual.

  2. Select VECTORS -> SET VECTOR FILENAMES

  3. Click on a name to replace OR a blank space to load a new file.

  4. Select a file written out by the ANIMEDITOR program.

  Click on 'DONE' and save the link file again!
```

1.31 Incorporate Converted Wall Graphics.

```
HOW TO:

      Incorporate Converted Wall Graphics
      ===================================

  Once a wall has been converted from IFF format to the special .256wad format  ↩
     used by the
  game, it can be incorporated into your levels. To do this, you must modify the  ↩
     gamelink
  file. Here's how to go about it:

  1. Run the GAMELINKER program.

  2. Select 'LOAD LINK FILE' from the menu and select

  MYDIR/INCLUDES/TEST.LNK

  3. Select GRAPHICS->SET WALL GFX FILENAMES

  4. You will be presented with a list of currently loaded files. Pick one to  ↩
     overwrite, or
     the last (blank) one to add a new file. Click on it with the LEFT mouse  ↩
        button.

  5. Select the .256wad file you wish to load.

  6. Repeat steps 4 and 5 until all the new walls you want to load are loaded,  ↩
     then click on
     'DONE'.

  7. Select 'SAVE LINK FILE' and select the file

  MYDIR/INCLUDES/TEST.LNK.

  The new walls will now be available in the level editor and the game itself!

  PLEASE NOTE:

    If you subsequently modify an existing wall graphic and reconvert it, you only ↩
        need
    to execute the above procedure IF:

    1. The width and/or height of the wall graphic has been changed

    OR

    2. You have converted it to a different path or filename and wish to select it
       instead of the old version.
```

1.32 Make a Clean Slate

```
HOW TO....

  Make a Clean Slate
  ==================

First of all, a bit of explanation about how Alien Breed 3D II works.

Before you can design your levels and populate them with vile denizens, you have  ↩
   to
specify the parameters within which the game will operate. These parameters  ↩
   include
such things as the types of aliens you will meet, the wall and floor graphics, the
type of weapons you can pick up and so on. This information is stored in the
GAME LINK FILE.

To begin with you will need a directory on your hard drive within which you will
place all the graphics, sounds and other files your game will use. For simplicity,
AB3DII requires the following subdirectories to be created:

MYDIR
  LEVELS
  INCLUDES
  GRAPHICS

Within the graphics drawer, you must also create two further subdirectories:
```

```
MYDIR
  LEVELS
  INCLUDES
  GRAPHICS
    FLOORS
    TEXTURES
```

Also, an assign to your directory must be made, preferably in your user-startup:

```
ASSIGN AB3: MYDIR
```

This will allow the editor programs to access your files easily and without endless prompting for filenames.

Copy all the files from the

```
  EXAMPLES/INCLUDES/
```

directory on your editor disk into

```
  MYDIR/INCLUDES
```

These are the palette files used by the game, the title screen graphic and other files used by the game when run. For instructions on changing the palette, refer to the file

```
  HOW2-Change-global-palette
```

Once such a directory structure has been made, you can start to build your game.

You cannot, however, just dive into the level editor. There are some things you must do first...

For the purpose of this tutorial, I recommend you follow this procedure:

1.
Make a directory called:

MYDIR/WALLGFX

2.
Copy the file "ExampleWall" from your editor disk (it can be found in the

EXAMPLES/WALLS/

drawer) into the new WALLGFX directory.

3.
Make another directory called:

MYDIR/WALLINC

4.
Consult the document 'HOW2-Convert_Wall_Graphics', placing the resulting files in the WALLINC directory.

5.
Copy all the files from the EXAMPLES/FLOORS directory into your FLOORS directory

6.
Consult the document 'HOW2-Convert_Floor_Graphics', placing the resulting file
in the INCLUDES directory.

7.
Make a directory called:

MYDIR/VECTOBJ

8.
Copy all the files from the EXAMPLES/VECTOBJ directory into your VECTOBJ directory.

9.
Copy the TKG file from your game disk into MYDIR.

This process gives you some example graphics to work with when designing your levels.

NB:
It is likely that at a later stage you may want to design your own graphics
instead of using those provided. It is a simple matter to remove the given
graphics and replace them, so for now it's probably best if you follow the
above procedure so you can get something up and running as quickly as possible.

What you have done so far is place some files, eg floor graphics and wall graphics,
in conveniently accessable positions. The next stage is to incorporate them into the
game.

For this purpose, the GAMELINKER program loads and saves a file called TEST.LNK which
contains all the information about alien behaviour, weapons, graphics and sounds and
where to find the relevant files on disk. Consult the following documents:

How2-Incorporate_Converted_Walls
How2-Incorporate_Converted_Floors

to incorporate the background graphics you have converted.

Once that is done, you can begin editing levels. Consult these documents:

How2-Start_A_New_Level
How2-Run_Your_Game

for further instructions. The file:

How2-Create_New_Aliens

contains information about creating new alien types, or

How2-Edit_Alien_Stats

tells you how to modify existing aliens.

How2-Design_Weapons

gives full instructions on creating new weapon types. You are given one weapon to begin with as an example; the shotgun.

1.33 Add Control Points to Levels.

HOW TO:

Add Control Points to Levels
============================

Control points are a wonderful new invention which allows aliens to follow you all over the
level without all that tedious mucking about in hyperspace. Erm.

Ok, what control points actually do is give the aliens a 'road map' of the level; sort of
invisible lines on the floor for them to trundle around. Such paths have been used in other
games to make aliens patrol around convincingly. The difference with AB3D is that the aliens
can use this map to get to you, if they know where you are, from the other side of the level,
assuming there is a path from where they are to where you are.
Also, if you make a sound somewhere near to an alien which is active, it will ' hear it' and
run to the control point it heard the sound coming from. If you continue shooting, therefore,
it will eventually make its way all the way over to where you are! In the diagram below, for
example, the player is standing near control point A. He has just fired his gun, and the
alien at point C will therefore run to control point B, which is the one towards where the player
is. When the alien reaches point B it will listen for the player to make more sounds, trying
to see if it can track him further. If it doesn't hear anything, it will wander off looking for
him.

```
A +---------------------+
                        |
                        |
                        |              B
                        |               +------------+ C
                        |               |
                          |               |
                          |               |
                          |               |
        +---------------+-----------+

PLACING CONTROL POINTS
----------------------

  Control points must be joined with straight lines and MUST NOT cross walls which the aliens
  cannot climb over. Therefore control points must be placed at all junctions and corners
  so that aliens can navigate properly. Also, if you want aliens to venture into a dead end,
  even if it is only a short one, you must put a control point near the end of the dead end.

  1. To place control points, select the icon on the right hand end of the scrollable icon bar;
     it looks like the 'add point' icon only with a pink dot rather than a green one.

  2. Move the cursor into the edit area and click where you want to position a control point
     with the mouse buttons. If you click the LEFT mouse button, the control point will be pink,
     indicating that it is in the LOWER part of a zone. If instead you use the RIGHT mouse button,
     the control point will be BLUE, indicating that it is in the upper part of the zone. Hence
     you can cause aliens to navigate across winding gantries without falling off.

     NB YOU CAN ONLY HAVE 100 CONTROL POINTS PER LEVEL SO USE THEM WISELY!

  You can also MOVE control points by clicking on the 'O->' icon, grabbing a control point by
  moving the cursor over it and pressing 'g', then placing it somewhere else by clicking with the
  left or right mouse buttons as above.

LINKING CONTROL POINTS
----------------------

  So that the aliens can move around, you must link the control points together. There are two
  ways they can be linked: PHYSICALLY or VISUALLY. Points linked PHYSICALLY describe a route for
  walking aliens. Points linked VISUALLY describe extra routes for flying aliens. Hence if you have
  a chasm with walkways on either side, the control points on each walkway would be linked visually,
  because flying aliens could get across but walking aliens could not.
  Control points which are linked PHYSICALLY are assumed to also be linked VISUALLY (but not the other
  way around).
  PHYSICAL links may be one-directional or two-directional. ONE directional links may only be
  walked in one direction but are linked VISUALLY in both directions; TWO directional links may
  be walked in both directions.

  NB: If an alien cannot physically reach you (eg an impassable chasm or an unclimbable cliff), but
      DOES know where you are, he may attempt to reach a position linked VISUALLY to your position
      so he can take pot shots at you. To take a simple example: say the player is on top of a
      cliff. Below him is a gorge, across which there is no bridge. There is a staircase leading
      up to the OTHER side of the gorge from the bottom, but not up to the side the player is on.
      Imagine there are several walking aliens with guns in the bottom of the gorge. The player
      doesn't want to jump down with them still there, so he keeps popping out and shooting, then
      ducking back out of harms way. The aliens cannot reach him, and nor can they see him properly
      from the bottom of the gorge. They will therefore climb the staircase to the other rim of the
      gorge where they can see the player and shoot at him!

  1. Select the 'link control points' icon. This is near to the 'add control point' icon and looks
     like a pink dot with three black lines radiating from it.

  2. Clicking on a control point (say point A) and then on another (say point B) with the LEFT
     mouse button toggles the PHYSICAL LINK from A to B. Clicking on them in the opposite order
     toggles the link from B to A. To link them two-directionally, therefore, it is necessary to
     click A, then B, then B again, and finally A. One-directional links are indicated by a red
     arrow indicating the possible direction of travel; two directional links are indicated by
     a blue, unarrowed line.

  3. Clicking on a control point (A) and then on another (B) with the RIGHT mouse button toggles
     the VISUAL link between the two points. If the two points are already linked visually, the
     link will be removed. If they are linked physically in one or two directions, or not linked at
     all, the existing link (or lack of it) will be replaced by a purple line indicating a VISUAL
     link.

OTHER TOOLS:
------------

  Obviously it is unreasonable to expect someone to click-click-click their way through a room
  containing 20+ control points, so it is only (usually) necessary to define the PHYSICAL links
```

by hand. The VISUAL links can be auto-generated by pressing RSHIFT and V together, with the cursor somewhere in the edit area. Visual links will be automatically created between any points with a line of sight between them which are not already linked physically. This takes a little while, but console yourself with the thought that it's about 250,000 times faster than you could do it by hand.

NOTE: If you have a tall pillar with a control point on either side, then the autolinker WILL link the two points, because there is no hole in the map seperating them. The ONLY time the autolinker will not link points across a zone is if that zone's roof and floor are at the same height; ie no travel across it is possible at ANY height.

ASSOCIATING CONTROL POINTS WITH ZONES:

Since you can have up to 256 zones but only 100 control points, it seems fairly obvious that not all zones can have a control point in them. Therefore the question of which control point an alien is nearest when standing in any particulat zone must be decided by you in advance.

This can be done in two ways, both of which begin by clicking on the 'associate zone with control point icon (which looks like a small triangle near to a pink dot). Next, you either

1. Select a zone with the RIGHT mouse button and click on the control point you want to associate with it with the LEFT mouse button. Make it a reasonable choice, ie don't pick a point on the other side of a solid wall.
   Repeat this process with each zone until you have covered the whole level.

   UPPERS: This process is VERY accurate and yields the most bug-free maps.

   DOWNERS: It takes AGES, and it's easy to forget to do certain bits of the map .

2. Press RSHIFT and A together with the cursor on the edit area. This will auto-associate zones and control points. Again, this takes a little while but speeds things up a lot.

   UPPERS: Takes no effort and doesn't miss anything out

   DOWNERS: It can quite often make mistakes; linking zones to points not in visual sight of them.

```
I prefer to initially use method 2, then tidy up with method 1 if I notice any ↩
   aliens stuck in
corners or walking into walls.

You must also link any upper parts of zones (eg gantries) with control points, ↩
   and this must be
done manually because of the error-prone nature of the process. Select the 'link ↩
    upper zone to
control point' icon (like the other one but with a double-triangle) and ↩
   associate as before.

It is perfectly OK to link a lower zone to an upper control point or an upper ↩
   zone to a lower
control point. See diagram for an explanation of why you might want to do this.


                  |     |
                  |     |
              Zone A -> |  *1 |
                  |     |
*-----------------*-----*-------------------*
|                 #     #                   |
|                 #     #                   |
|       Zone B -> #     #                   |
|                 #     #                   |
*-----------------*-----*-------------------*
|                 #     #                   |
|                 #     #                   |
|       Zone C -> #  +2 #                   |
|                 #     #                   |
*-----------------*-----*-------------------*
      |     |
      |     |
  Zone D -> |     |
      |     |


In the diagram point 2 (+) is a blue (upper zone control point), and point 1 (*) ↩
    is a
pink, lower zone control point. Zones B and C comprise a gantry above a large ↩
   room,
whilst zones A and D are corridors leading out onto that gantry. As you can see, ↩
    given
the arrangement above, the upper part of zone C and the lower part of zone D ↩
   should BOTH
be linked to point 2, and the lower part of zone A and the upper part of zone B ↩
   should
both be connected to point 1.
```

1.34 Save a Level to Disk.

```
HOW TO:

            Save a Level to Disk
      ====================
There are a number of files that AB3DII uses when loading a level, and so saving ↩
    levels to disk is not
such a simple proceedure. Here is a list of all the files involved in a level:

  TWOLEV.BIN
  TWOLEV.GRAPH.BIN
  TWOLEV.MAP
  TWOLEV.FLYMAP
  TWOLEV.CLIPS
  TWOLEV.DAT
  TWOLEV.OBJ
  TWOLEV.LINKS

And these are the files actually loaded by the game when you play it:

  TWOLEV.BIN
  TWOLEV.GRAPH.BIN
  TWOLEV.MAP
  TWOLEV.FLYMAP
  TWOLEV.CLIPS

The rest are files used by the level editor and AI-Linker.

Here's a quick run-down of what is stored in each file:

TWOLEV.BIN
  Data about the physical shape of the level and objects within it. This file is ↩
     created by
  the level editor AUTOMATICALLY when you save a level, and is loaded by the game ↩
     when you run it.
  VERY IMPORTANT: THIS FILE *MUST* BE PACKED BEFORE THE GAME CAN BE RUN! SEE BELOW ↩
      FOR DETAILS!


TWOLEV.GRAPH.BIN
  Data about the graphical appearance of the level; for example which textures ↩
     appear on which walls,
  instructions to draw water and so on are. This file is created by the level ↩
     editor AUTOMATICALLY
  when you save a level, and is loaded by the game when you run it.
  VERY IMPORTANT: THIS FILE *MUST* BE PACKED BEFORE THE GAME CAN BE RUN! SEE BELOW ↩
      FOR DETAILS!

TWOLEV.MAP
  This file tells walking aliens how to get around the level you have designed. ↩
     They use the info
  in this file to trace the sound of your gunfire, follow you around corners and ↩
     all the rest.
  This file is generated by the LINKUP program, which uses the TWOLEV.LINKS file ↩
     to generate it.
  This file can be packed to save space, but it is not necessary to test the level ↩
     .

TWOLEV.FLYMAP
  Similar to the above, but for flying aliens (which can obviously get across gaps ↩
      walking aliens
  can't).
  This file can be packed to save space, but it is not necessary to test the level ↩
      .

TWOLEV.CLIPS
  I can guarantee that this file will cause you more grief than any of the others, ↩
       because it
  certainly gave me more hassle than anything else in the entire game! Put simply, ↩
       the .CLIPS file
  contains all the information about lines of sight, obstructions and other  ↩
      important stuff. Creating
  this file takes a LONG time; for a large level it can take several minutes on  ↩
      even an accellerated
  computer. Before you complain, the equivalent file for Quake takes 4 hours on a  ↩
      network of super-fast
  power computers, so count your blessings!
  On the plus side, this file only needs recalculating under certain circumstances ↩
      . If you run the
  game, then change the level, you only need to recalculate the clips file if:

  1. You have added one or more zones to the level
     OR
  2. You have moved one or more points which lie on the border of zones

  You DO NOT have to recalculate the clips file unless you have donw one of the  ↩
      above since last
  playing your level. Changing floor and ceiling heights, adding water or objects, ↩
       changing the
  lighting in zones or changing the graphics on walls and floors does NOT require  ↩
      recalculating the
  clip file before you can play the level.

  It is therefore a very good idea to plan your level on paper, then put as much  ↩
      of it in as you
  can in one go, then calculate the clip file, then save the level to disk, then  ↩
      run the game to
  test it.

  NB: If you recalculate the clip file, you MUST save the level afterwards.  ↩
      Failure to do so will
  result in a corrupted display in the game, and mean you have to waste time  ↩
      calculating the clip file
  AGAIN, then saving the level.

  VERY IMPORTANT: THIS FILE *MUST* BE PACKED BEFORE THE GAME CAN BE RUN! SEE BELOW ↩
       FOR DETAILS!

TWOLEV.DAT
  This is the file where the level editor stores its working copy of all the info  ↩
      in the .BIN
  and .GRAPH.BIN files. DO NOT LOSE THIS FILE! It is not necessary for it to be  ↩
      included with levels
  for playing the game, eg if you pass your levels on to friends, but without it  ↩
      you cannot edit the
  level using the editor!

TWOLEV.OBJ
```

```
  This is the file where the level editor stores a working copy of the object positions and stats for
  the level. DO NOT LOSE THIS FILE EITHER! It is not necessary to include this file with levels
  you pass on to your friends, but without it you won’t be able to edit the level anymore!

TWOLEV.LINKS
  This is the file containing the raw data about which parts of the level are accessable directly from
  one another. YOU CAN LOSE THIS FILE IF YOU WANT, but if you do, you’ll have to go back and link up
  all your control points again, which is very dull, so TRY NOT TO!

PACKING FILES
=============

  Three of the five files loaded by the game MUST be packed before they can be used. These are
  the TWOLEV.BIN, TWOLEV.GRAPH.BIN and TWOLEV.CLIPS files.

  To speed up this process, a number of scripts have been thoughtfully included with your editor pack.
  These are:

  PG    Pack the .BIN and .GRAPH.BIN files ONLY.
  PGC   Pack the .BIN, .GRAPH.BIN and .CLIP files.
  PM    Pack the .MAP and .FLYMAP files only.

  USAGE: Type ’pg A’ to pack the files in level A, for example.

  IMPORTANT: DO NOT pack files twice. This most often happens accidentally when you load and save a level
  in the editor without having to recalculate the .CLIPS file, then type ’pgc A’ rather than ’pg A’.
  Since the .CLIPS file has not been changed the packer will attempt to pack it a second time, get a bit
  confused and the result will be a currupted display in the game. Unfortunately, the only way to
  recover from this is, you guessed it, recalculate the clips file, save the level, then ’pgc’ it!

CREATING THE MAP and FLYMAP FILES
=================================

  If you have put aliens in your level along with control points, you will need to  generate the MAP and
  FLYMAP files. Use the LINKUP program, entering the level letter when prompted. If you have a slow
  computer and a complicated level, this might be a good time to make some tea, watch some telly or
  perhaps look for an alternative hobby. Sorry about the time it takes, but there’s a lot of raw info
  to assimilate.

          ================
          SAVING PROCEDURE
```

================

Note: To find out how to calculate the Clip file, go back to the Index and select the Topic on Starting Up, to test the example level. Read the steps on how to start up a test of the example level for an illustration.

1. Have you just added one or more zones, or moved any points?
   YES: RECALCULATE THE CLIP FILE

2. Save the level by pressing 's', then typing the letter of the level you are saving and pressing RETURN.

3. Quit the level editor with CTRL-C.

4. Did you recalculate the clip file before saving?
   YES: type 'pgc' then the letter of the level you just saved.
   NO: type 'pg' then the letter of the level you just saved.

5. Are there any aliens in your level?
   YES: Have you put any control points in your level?
     YES: Type 'linkup' and enter the letter of your level at the prompt.

6. You are now ready to play the game!

1.35 Set Level Floor and Ceiling Graphics.

HOW TO:

Change Wall Floor and Ceiling Graphics
======================================

THIS FILE INSTRUCTS ON USAGE OF THE LEVEL EDITOR. PLEASE
READ THE LEVEL EDITOR KEY-GUIDE FOR BASIC INSTRUCTIONS.

One of the more arcane and impenetrable stages of level editing is the defining of graphics for the walls, floors and ceilings of your levels. It is therefore imperative that the proper runes are cast and the traditional ceremonies carried out TO THE LETTER to avoid the incomparable wrath of the editor gods.

Ok, so thats a lie. Graphics aren't that hard to include. However, it IS time consuming and rather dull, particularly when you're on level 15 of 16 and you really can't be arsed thinking too carefully about what you're doing, which is why I've included a shortcut or two for the lazier of you.

WALLS:

Walls are the most complicated things to define graphics for. The main ingredients cast into the cauldron for a successful wall graphic are:

1. Which wall file the graphics are to be drawn from
2. Which part of the file is to be mapped onto the wall
3. Whether the section of graphic is to be stretched so that an exact number of tiles of that strip fit onto the wall OR
4. Whether the section of graphic is to be squashed so that an exact number of tiles of that trip fit into the wall OR
5. Whether the strip of wall is to be drawn in strict 1x1 ratio regardless of the number of tilings.
6. Assorted eyes, toes, and the occsional spleen from small furry animals (optional)

The procedure you should follow is this:

1. Select the 'define wall graphic' icon. This can be found near the right hand end of the icon bar on the middle row, and looks like a letter C with a red vertical bar and grey horizontal bars.

2. Move the cursor into the edit area. The icon bar will vanish and be replaced with a rather dotty and transparent vision of a wall graphic. Press 'v' to get a better look at it.

3. Use the '<' and '>' keys to scroll along the wall graphic, or if you want to select the next or previous graphic in series, press RSHIFT along with '<' or '>'. To change the length of the tile (multiply or divide it by two, in fact), press Q or W. Once the section of graphic visible corresponds to the part of the graphic you want pasted on the wall, move on.

4. Is it important that the wall contains a whole number of tiles of the graphic selected? Such cases arise with doors, where the door graphic must be stretched to fit the exact width of the door, and so on. If you want to STRETCH OUT the wall graphic so that a whole number of tiles covers the wall, pressing '-' toggles the effect on or off. Similarly, '+' toggles SQUASHING the graphic into the wall on/off.

5. Assuming all is OK, select the zone containing the wall to be pasted onto with the RIGHT MOUSE BUTTON. Clicking on the LEFTMOST point of the wall as you would see it in the game pastes the graphic onto that wall, eg:

```
        *---------------------* <- Clicking on THIS point...
        |                     |
        |                     |
        |                     |
        |                     |
        |                     | <- Changes THIS wall.
        |                     |
        |                     |
        |                     |
        |                     |
      *---------------------*

OTHER TOOLS:
------------

  You can grab the graphic settings from a wall by selecting the zone with the  ↩
     right mouse button,
  holding the mouse pointer near to the left hand end of the wall as above, and  ↩
     pressing 'g'.

  You can fill all the walls in a zone with your graphic by selecting the zone  ↩
     with the
  right mouse button and pressing 'f'.

THINGS TO REMEMBER:
-------------------

  It is possible for a single wall to have two different textures on it, depending ↩
      upon which
  side you are on. For example, this SIDE ON VIEW of a step:


      _____________________
      |                   |
      |          Side A ->|
      |                   |_________________
      |                   .                |
      |                   .                |
      |                   .                |
      |                   .                |
      |                   .                |
      |___________________.                |
              |                |
              |<- Side B       |
              |________________|

  Hence you must define wall graphics from BOTH SIDES in cases like this.

  To help you in this task, pressing SHIFT and 'D' will redraw the level with all  ↩
     visible
  walls or steps still containing default graphics drawn in flickering grey, so  ↩
     you can see
  where you have missed.

  There is a corresponding DEFINE UPPER WALL GRAPHIC icon, which defines graphics  ↩
     for walls
  visible in the upper part of a zone (if it exists). It works in exactly the same way.

FLOORS and CEILINGS:
--------------------

  These are a little easier. Select the 'DEFINE FLOOR GRAPHIC' icon (near the right hand end of
  the scrollable icon bar; it looks like a letter C with a red bottom). Moving the cursor
  into the edit area replaces the icon bar with a picture of the graphic you are going to paste.

  Clicking in a zone with the RIGHT mouse button grabs the graphics already there.

  Q and W scale up or down the graphic by a factor of 2. This is shown next to the graphic by
  a number from -5 to +5. -5 is very, very big, and +5 very, very small.

  '<' and '>' scroll backwards and forwards through the available tiles.

  NB: When defining ceiling graphics, note that in addition to the 16 graphic textures, there
      an additional option of having no roof (ie 'sky'). This cannot be scaled.

  Clicking with the LEFT mouse button in a zone pastes the selected graphic into it. It is
  also possible to hold down the mouse button and 'smear' a texture over a number of zones without
  having to tediously click in each one. Useful for defining stairs and such like.
```

1.36 Starting a New Level.

```
HOW TO:

        Start A New Level
        =================

      THIS FILE INSTRUCTS ON USAGE OF THE LEVEL EDITOR. PLEASE
      READ THE LEVEL EDITOR KEY-GUIDE FOR BASIC INSTRUCTIONS.

First of all, create a directory:

MYDIR/LEVELS/LEVEL_A

This is where the files for the first level will be created.

Now run LEVED. You will be presented with a screen, the upper and largest part of which is
blank, the lower part of which contains icons.

USING THE ICON BAR:
===================
```

The icon bar may be scrolled left and right using the (and) keys on the keypad. Click on an icon with the LMB to select it. A short message will appear next to the icon bar to inform you which icon you have selected. For the present, DONT PRESS ANYTHING!

We shall begin by creating a very simple level consisting of two rooms or, as they shall be referred to here, 'zones'. What you may think of as a single room will frequently consist of several zones, for reasons which will become apparent.

1. Select the 'add point' icon. This is the one at the top left of the icon bar represented by

   +O

2. Move the mouse up to the top area of the screen and click four times in a square pattern, the corners about 2 inches apart. Put them towards the left hand side of the screen , like so:

```
        *               *








        *               *
```

3. To create a zone, click on the icon immediately below 'add point', the one like a triangle with dots in the corners. Move up to the edit area and select the points, one at a time, in a CLOCKWISE order, returning to the one you started from. The result should be a flashing grey box, like this:

```
        *---------------*
        |               |
        |               |
        |               |
        |               |
        |      END      |
        |               |
        |               |
        |               |
        |               |
        *---------------*
```

4. Now that you have one zone, you can add more points....

```
        *---------------*             *
        |               |
        |               |
        |               |
        |               |
        |      END      |
        |               |
        |               |
        |               |
        |               |
        *---------------*             *
```

5. ....and connect the four rightmost points, again in a clockwise order:

```
        *---------------*-------------*
        |               |             |
        |               |             |
        |               |             |
        |               |             |
        |      END      |             |
        |               |             |
        |               |             |
        |               |             |
        |               |             |
        *---------------*-------------*
```

6. Notice that the line common to both zones is dark, rather than bright. This indicates a border
   between the zones. Bright lines are solid walls which stretch from floor to ceiling.

   Note: That dark line only represents a border. That is, the zones are connected together to make
   one room, made out of 2 zones. A room can be made of many zones. I personally found out that you
   can make a nice complex room, with two levels with only 16 zones. You have up to 256 zones per
   level, which isn't bad. Stair cases can start to take up zones, but even at that, it is a good
   amount of zones to have at your disposal. If you manage to make a level that REALLY needs 256
   zones(No redundancies), doesn't take the player 2 hours of hunting to find their way though,
   and doesn't get the player hopelessly lost, then please send me a copy. (Mike B .) As you see,
   putting zones together, like this, automatically connects the Zones together, removing the wall
   between them. You must add the wall, if you want one there.

7. Now you'll have to place the player start position. Select the icon of the little man, and click
   with the LEFT mouse button in the RIGHTMOST of the two zones. You should see something like this:

```
        *---------------*-------------*
        |               |             |
        |               |             |
        |               |             |
        |               |             |
        |      END      |      P1     |
        |               |             |
        |               |             |
        |               |             |
        |               |             |
        *---------------*-------------*
```

If you want to test the level out in the game, read the document

  HOW2-Save_Levels_To_Disk

for full instructions.

When you run the game, notice that the game ends as soon as you step into the 'end' zone.

BUILDING LEVEL OUTLINES

Personally, I prefer the following method of level design:

1. Design the level outline, that is, the plan view of the level.

2. Run the game on the level and have a wander around.

3. Assuming all is OK, I put in the floor and ceiling heights.

4. Have another quick run around to make sure all is well.

5. Next, I put in all the wall and floor and ceiling graphics

6. Check out what it looks like.

7. Put in all the snazzy lighting effects

8. Have another look.

9. Put in objects like keys, medipacks and guns, together with doors and lifts.

10. play the level through, ensuring that it can be completed by collecting the right keys and so on.

11. Put in the control points for aliens to use

12. Put in aliens.

13. Final playtest.

This is also a very good way for you to learn how to use the features of the game, as it introduces new
concepts one step at a time.

See these files for information about later stages of level development:

HOW2-Change_Floor_and_Ceiling_Heights
HOW2-Set_Wall_Floor_and_Ceiling_Graphics
HOW2-Change_Level_Lighting
HOW2-Add_Doors_And_Lifts
HOW2-Add_Objects_To_Levels
HOW2-Add_Control_Points
HOW2-Add_Aliens_To_Levels

This document will now concentrate on the rules of designing level outlines.

RULES and REGULATIONS:

1.
Zones MUST ALWAYS be defined by selecting points in CLOCKWISE ORDER.

2.
Zones can have a MAXIMUM of ten sides.

3.
Zones MUST ALWAYS be convex.

Several definitions of convexity (pick the one which makes sense to you):

a. A convex shape is one where you can walk all the way around the edge and only have to turn in
ONE direction.

b. If you draw a straight line between any two points in a convex shape, that line will always lie
ENTIRELY within the shape, ie it will never cross out of the shape and back in again.

4.
No zone must ever be visible on more than ONE side of a hole in the map from any other zone.

This is a bit trickier to explain, but I'll do my best:

```
*------------*
|            |
|     A      |
|            |
*===*----*===*
|   |    |   |
| B |HOLE| C |
|   |    |   |
*===*----*===*
|            |
|     D      |
|            |
*------------*
```

The shape above consists of FOUR zones A,B,C and D; a large one at the top and at the bottom, and two

```
smaller connecting ones on either side, with a hole inbetween. The hole DOES NOT ↩
    count as a zone,
even though it is completely surrounded by zones. You can tell that it is a hole ↩
    and not a zone in the
level editor because the walls around it are BRIGHT WHITE, indicating impassable ↩
    walls, whereas the
borders between zones (shown as double lines '=' above) will be in grey.
The above design is NOT ALLOWED, because zone A is visible from zone D on both  ↩
   sides of the hole.
There are TWO ways of fixing this situation:

a. Delete zones A and D, then add points and zones to get the following:

*---*----*---*
|   #    #   |
| A # D  # F |
|   #    #   |
*===*----*===*
|   |    |   |
| B |HOLE| C |
|   |    |   |
*===*----*===*
|   #    #   |
| G # H  # I |
|   #    #   |
*---*----*---*

This is legal, because no zone is visible on either side of the hole from any  ↩
   other zone.

b. Add one extra zone, covering the hole:

*------------*
|            |
|     A      |
|            |
*===*====*===*
|   #    #   |
| B # E  # C |
|   #    #   |
*===*====*===*
|            |
|     D      |
|            |
*------------*

then change the floor height of zone E to be the same as its roof height, or  ↩
   vice versa.

Method a. is good because HOLES in the map allow the game to ignore anything on  ↩
   the other side.
Thus, if the player stands in zone H, zone D will not be drawn because it is  ↩
   invisible from anywhere
in zone H.
However, method a. is not always feasable except in very simple or regular  ↩
   patterns of holes. The
```

difficulty involved in making sure this rule is obeyed in complex rooms is prohibitive. That is
why method B is often used, despite the fact that it prevents the game eliminating invisible zones;
in this case, zone C would be drawn if the player was in zone B, despite the fact that zone E
completely obscures it.

5.
Try to avoid using sharp, pointy zones when designing the level; for various reasons the game doesn’t
really like them. If you find yourself about to add an excessively pointed zone, try rethinking the
way you have designed the room.

6. Zones MUST NOT contain within them any other zones, or any points be they connected or not.
One of the most tempting things people try to do is to create a raised pedastle in the following
manner:

```
*-----------------------*
|                       |
|                       |
|     A                 |
|                       |
|                       |
|     *-----------*     |
|     |           |     |
|     |           |     |
|     |    B      |     |
|     |           |     |
|     |           |     |
|     |           |     |
|     *-----------*     |
|                       |
|                       |
|                       |
|                       |
|                       |
*-----------------------*
```

This WILL NOT WORK. If you want such a pedastle you must create it in the following way:

```
*-----------------------*
|#                     #|
| #                   # |
|  #        B        #  |
|   #               #   |
|    #             #    |
|     *-----------*     |
|     |           |     |
|     |           |     |
| A   |    E      |  C  |
|     |           |     |
|     |           |     |
|     |           |     |
|     *-----------*     |
|    #             #    |
|   #               #   |
|  #        D        #  |
| #                   # |
|#                     #|
*-----------------------*
```

1.37 Use the Animation Frame Selection Screen.

HOW TO:

```
      Use the Animation Frame Selection Screen
      ========================================
```

The Animation Frame Selection screen gives a more graphically meaningful way of selecting
frames from your object graphic files.

You will see an image (if one is defined for the graphic file you selected) in the top
left hand corner, and some buttons below:

[PREV FRAME] [NEXT FRAME]

Clicking on these cycles through the frames defined in the graphic file.

[FLIP HORIZONTALLY]

clicking on this flips the graphic. Useful for re-using side-on movement frames
for both left and right movement.

Current Frame:

Shows the number of the current frame in the graphic file being displayed.

DONE   Return to wherever you called the screen up from, using the frame and orientation you selected.

1.38 Adding your own sky or backdrops.....

The file is called 'rawbackpacked' and is made from a RAW image...ie.
palette numbers only...what I did was load the Bril.pal palette into
brilliance on a 256 colour screen...then imported a brush with the wrap
around image for the sky so that it maps to the actual TKG palette.
the image has to be a particular size. The dimension is 656x240
now youve got your 256 colour sky image in brilliance (or whatever) and
its the correct way up ie.

```
                +--------------------------------------+
                |                            \|/       |
                |                            -O-       |
                |    ______                  /|\       |
                |---/      \__________/\_________------|
                +--------------------------------------+
```

then you have to make the palette into a 256 greyscale with black at colour
0 and white at colour 255 (using the spread command in brilliance palette menu)
....ok now save the 256 grey image as ’picture.grey’ or whatever...
now it has to be made into a RAW file oh and rotated 270 degrees too
clockwise...sorry this is so tricky :(
Photogenics will save in Raw format which is what i used so load the
’picture.grey’ file and bring up the savers menu and choose RAW and select
8bit GREYSCALE in the prefs and save the file as ’picture.raw’

righty ho...um now run the raw file through the LHApack program that came
with the TKG editors which will compress it properly
’PackLHA blah-blah:picture.raw’
then rename the file to ’rawbackpacked’ and plonk it in the ab3:includes drawer
and test it out...

in case you wish to know the game maps the pixels from your image like so...

```
   8 16     (and so on so if your height is wrong the pikzels will be put in
   7 15      the wrong line horizontally so you will definately know if you
   6 14      have it right...)
   .  .
   .  .
   .  .
   2 10
   1  9

  ---Nick McKenzie
```