

GruntzEdit Features

"File" Menu

File New...(Ctrl+N)

Creates a new level file (.wwd file). I strongly recommend not starting a new level this way. Instead, it is recommended that you use a "template level" to begin your new level. Simply copy the "template level" and rename it and make sure to keep a backup of your "template" levels.

File Open...(Ctrl+O)

Opens an existing level file (.wwd file).

Save (Ctrl+S)

Saves your current level.

Save As...

Saves the current level under a new file name. After opening an existing "template level", use this feature to save the level as a new level. This is the recommended method for starting a new level.

Printing to Bitmap... (Ctrl+P)

This is a VERY useful feature for analyzing the levels for playability stuff and overall layout design.

Ctrl+P (or "File | Print To Bitmap") will bring up a dialog where you can enter the bitmap filename, choose the scale value (100%, 50%, 25%, or 12%) and specify whether or not you want the objects printed as well.

The current plane/layer is then dumped (in it's entirety) to the specified file and you can then use your favorite paint program to view and/or print it.

"Edit" Menu

Undo (Ctrl+Z)

This feature is used to undo the last action. **Important:** Not all actions can be undone. (e.g. block copy) Therefore, make sure to always save your level before making any major changes.

Cut (Ctrl+X)

This feature is used to remove an object from the level. The object will be moved into the "clipboard" so the player can "paste" the object to a new location.

Copy (Ctrl+C)

This feature is used to copy an object from the level. The object will be copied into the "clipboard" so the player can "paste" the duplicate object to a new location.

Paste (Ctrl+V)

This feature is used to paste an object from the "clipboard".

Object Mode (Ctrl+M) or Tile Mode (Ctrl+M)

When in Tile Mode, pressing Ctrl+M (or "Edit | Object Mode") will enable Object Mode. While in Object Mode, you can add an object by double-clicking in the area where you want to place the object. When you double-click, a dialog box will appear.

Use the dialog box to add an object with a specific logic and image set.

While in Object Mode, pressing Ctrl+M (or "Edit | Tile Mode") will enable Tile Mode. Tile mode is the default mode when opening the editor. If you think of tiles as being similar to bathroom tiles or wallpaper, you've got the basic idea. Tiles are placed together so that they create structures or patterns. To display the tile selection window you can press Ctrl+T while in Tile Mode (or "Tile | Selection Window").

"World" Menu

Planes...(Ctrl+L)

This feature is used to add, delete, or move different planes (e.g. back, action, front). It is recommended that you use the "template levels" since doing so will prevent the you from having to use this feature. Note: In Gruntz, there are no levels that have a "front" plane. Only Area4 (High on Sweetz), Area5 (High Rollerz), and Area8 (Gruntz in Space) use this feature to create a parallaxing background plane.

Properties

The following fields are used in World Properties for "Gruntz".

Name: - This is where the user enters the level number. (e.g. "Gruntz - Level 5"). The game uses the level number to determine which area specific images to load into memory (e.g. Rocky Roadz, Gruntziclez, etc). **Important!** Make sure that you are using the appropriate Level number for your custom level.

Rez File: - This field specifies the location of the Gruntz.rez file. The default for the template levels is "C:\Games\Gruntz\Gruntz.rez". If you change the default installation directory for Gruntz, then you will need to update the path in this field for your custom levels.

Launch App: - This field specifies the location of the Gruntz.exe. The default for the template levels is "C:\Games\Gruntz\Gruntz.exe". If you change the default installation directory for Gruntz, then you will need to update the path in this field for your custom levels.

"Plane" Menu

Add...(Ctrl+A)

This feature is used to add a plane. It is recommended that you use the "template levels" since doing so will prevent the you from having to use this feature. Note: In Gruntz, there are no levels that have a "front" plane. Only Area4 (High on Sweetz), Area5 (High Rollerz), and Area8 (Gruntz in Space) use this feature to create a parallaxing background plane.

Delete...

This feature is used to delete a plane.

Goto X,Y Position... (Ctrl+Alt+P)

This feature allows the user to go to any position (defined by the X and Y coordinates) in the current level. This feature is also used to set the "Warp" coordinates to allow the player to use the "Goto Warp Position" feature (described below).

Goto Start Position... (Ctrl+Alt+P)

This feature allows the user to go to the start position (defined by the start position coordinates set in the "World | Properties" menu). Note: The "start position" in Gruntz is where the level will be centered when the level is first starts.

Go to Warp Position... (Ctrl+Alt+P)

This feature allows the user to quickly go to the position in the current level specified by the "Warp" coordinates. Note: the "Warp" coordinates are specified by using the "Goto X,Y Position..." feature.

Properties (Alt+Enter)

The following fields are used in Plane Properties for "Gruntz".

Tiles Wide: - This determines the width of the level in number of tiles.

Tiles High: - This determines the height of the level in number of tiles.

"Tile" Menu (Tile Mode Only)

Next Tile

Goes to the next tile in the "Tile Selection Window".

Previous Tile

Goes to the previous tile in the "Tile Selection Window".

Make Invisible (Ctrl+I or Del)

Changes the currently selected tile in the level to be invisible.

Make Filled (Ctrl+F)

Changes the currently selected tile in the level to be "filled" with a tile that uses the fill color defined in the "Plane | Properties" menu. Note: This feature was never used in Gruntz.

Set at Cursor (Spacebar)

This feature will change the current tile underneath the current cursor position with the tile currently selected in the "Tile Selection Window" or with the tile that was last copied into the "clipboard". Note: a tile can be copied into the "clipboard" by holding down the Ctrl key while left-clicking on any tile that you would like to copy.

Selection Window (Ctrl+T)

This feature is used to display or hide the "Tile Selection Window".

Resize Attribs

I have no idea what this feature is used for. I have never used it on any game that we have developed (Claw, Get Medieval, or Gruntz)

Validate...

This will check your level for invalid tiles. This should never be required, because the shipped game should never have invalid tiles. Note: An invalid tile is a tile that is not 32x32 pixels or 64x64 pixels.

Properties...

This feature allows the user to modify the properties for each tile. Note: This will only change the properties for the tiles in the current level. **Important:** If you are using the "template levels", do not modify the tile properties for ANY tiles since they all have already been specified. If you are starting a level from scratch, you will need to set the tile properties for approximately 300 tiles, which is why I recommend using a "template level." J

"Cursor" Menu (Tile Mode Only)

Most of the features in this menu are self-explanatory. The only feature worth describing is the "Auto Move" feature.

AutoMove

This feature specifies whether or not you would like the next tile (to the right) to be highlighted when placing a tile. For example, when you use the spacebar to place a tile, after the new tile is placed the tile to the right will then be highlighted. This allows the user to hold the spacebar down and replace multiple tiles with the one currently selected in the "Tile Selection Window."

"Objects" Menu (Object Mode Only)

Edit... (Ctrl+E or right-click on an object)

This will display the "Edit Objects" dialog box that is used to enter game logic information. See the "Game Logics" section for more information.

Insert (Ins)

I don't think this feature does anything.

Delete (Del)

This will delete a selected object.

Delete All

This will delete all objects in a level. This feature is not used often.

Delete Invalids

This will delete all invalid objects in a level. I don't think you will use this since there should be no invalid objects in the shipped version of the game.

Select Next (Tab)

This will select the next object in the level.

Select Previous (Shift+Tab)

This will select the previous object in the level.

Nudge Up (Ctrl+Up Arrow)

This will move the selected object 1 pixel in the desired direction.

Nudge Left (Ctrl+Left Arrow)

This will move the selected object 1 pixel in the desired direction.

Nudge Right (Ctrl+Right Arrow)

This will move the selected object 1 pixel in the desired direction.

Nudge Down (Ctrl+Down Arrow)

This will move the selected object 1 pixel in the desired direction.

Shove Up (Shift+Up Arrow)

This will move the selected object 5 pixels in the desired direction.

ShoveLeft (Shift+Left Arrow)

This will move the selected object 5 pixels in the desired direction.

ShoveRight (Shift+Right Arrow)

This will move the selected object 5 pixels in the desired direction.

ShoveDown (Shift+Down Arrow)

This will move the selected object 5 pixels in the desired direction.

Resize Coords...

I have no idea what this feature is used for. I have never used it on any game that we have developed (Claw, Get Medieval, or Gruntz)

Rename Logics...

This is very handy feature for renaming the logics for all objects in a level that have a common object name. **Important:** Logic names are case sensitive!

Rename Images...

This is very handy feature for renaming the images for all objects in a level that have a common image name. **Important:** Image names are case sensitive!

Convert Image Logics...

This is very handy feature for changing the logic name for all objects in a level that have a common image name.

Change Z Values...

This is very handy feature for changing the "Z-order" for all objects in a level that have a common logic name. **Important:** Logic names are case sensitive!

"Logics" Menu (Object Mode Only)

Edit Logics...

This will allow you to edit the shortcuts that are used to quickly add objects with a certain logic type to your level. Note: I never used this feature because it is easier to just copy an existing object and paste it to a new location.

Auto Images

This feature (when toggled on) will try to automatically assign an image to an object that is only given a logic name. For example, if I add a new object with the logic name "EyeCandy" and don't assign an image set, the editor will try to automatically assign an image set that was used by a previous object that had the logic name "EyeCandy". Note: This feature never seemed to be very useful, because it either didn't work or it would just assign an image set that I didn't want.

"Z-Coords" Menu (Object Mode Only)

Edit Values...

This will allow you to edit the shortcuts that are used to quickly assign a Z-order to a selected object in your level.

Auto Z-Coords

This feature (when toggled on) will automatically assign a Z-order for a new object added to the level. The Z-order value it uses is determined when an object with the same logic name is first added to the level and a Z value was assigned. For example, the very first time I add an object with the logic name "EyeCandy" to my new level I give it a value of "1". Now every time I add a new object with the logic "EyeCandy", the editor will give it the Z value of "1".

"Tools" Menu

Fill... (Tile Mode Only)

You can use the fill feature to flood-fill an entire plane with a certain tile number or type. This is useful if you want to copy an existing level, rename it, and then start fresh.

NOTE: THERE IS NO UNDO FOR THIS FEATURE! Also, this is an advanced feature that, if used incorrectly, could potentially screw up your level! Test this out in a new or test level before trying this on something you really care about.

To flood fill all tiles in a level:

- 1) Select **Fill...** from the **Tools** menu.
- 2) Select the type of fill you want to do. You can enter a tile number (which will fill the entire level with that tile), set it to invisible (useful for clearing out the tile contents of an entire level, but note that it will not clear out the objects), or set it to the default color (useful if you want to set a background layer to a solid color, such as blue for a sky). You can set the default fill color by selecting Properties from the Plane menu, and then setting the color value in the Misc area of the properties dialog.
- 3) Set the origin for the fill (it will fill down and to the right from the origin).
- 4) There is no undo for this feature!
- 5) Click **OK** when you're ready.

Block Copy... (Tile Mode Only)

The block copy feature allows you to copy a large section of a level and paste it into another portion of the level. To use the feature, follow the instructions below:

NOTE: This is an advanced feature that will **OVERWRITE** any tiles in the paste area. Test this out in a new or test level before trying this on something you really care about.

To block copy a set of tiles:

- 1) Select **Block Copy...** from the **Tools** menu.
- 2) This is where things get a little complex. There are 6 fields listed, with each field having a specific meaning. You need to enter numerical values in each of the fields. Use whole numbers.
- 3) Enter the value for the Source Column. The source column is the column number of the top left corner of the block of tiles you want to copy (think about this as a coordinate system for a rectangle, with the top-left corner of the rectangle being the origin of the coordinate system). If you're unsure as to the column number, it's the first number listed on the status bar (bottom of window) when a tile is selected.
- 4) Enter the value for the Source Row. The Source Row is the row number of the top-left corner of the block of tiles you want to copy. If you're unsure as to the row number, it's the second number listed on the status bar when a tile is selected.
- 5) Enter the value for the Dest Column and Dest Row. These are the top-left coordinates of the area where you want to paste the tiles.
- 6) Enter the values for Copy Width and Copy Height. Copy Width is how many tiles to the right you want to copy (your x-delta). Copy Height is how many tiles down you want to copy (your y-delta). Note that these are delta values, not absolute tile position values.
- 7) Click **OK**. The block of tiles you copied should appear in their new position.

Replace... (Tile Mode Only)

You can replace a tile in a level with a different tile. This is a global action, and will replace all instances of a particular tile in a level.

To use the replace feature:

- 1) Select **Replace...** from the **Tools** menu.
- 2) Enter the tile number that you want find.
- 3) Enter the tile number you want to replace it with.
- 4) This feature is global, and therefore acts on every instance of the tile you're searching for...
- 5) Click **OK**.
- 6) The tile you wanted to replace should now be replaced everywhere it appeared in the level.

Freshen Rez File

I'm not sure what this is used for. Don't worry though, I have never had to use it. J

Launch App

By selecting this, the editor will launch the game using the path specified in the "Worlds | Properties" menu.

Options...

Image Loading

Selecting this option will make the editor load images quicker since it will only load a single image for an object that may have multiple frames of images because it is an animating object. Note: this does not effect the way the game works, it is only an option to speed up the performance of the editor.

Rez File

This feature is used to set an alternate path for the game REZ file. This option will probably never be used since the user will typically only have one REZ file. Note: This feature was used for development only.

"View" Menu

Toolbar

This feature when toggled will either hide or display the toolbar.

Statusbar

This feature when toggled will either hide or display the statusbar.

Grid (Ctrl+G)

This feature when toggled will either hide or display the grid.

World (Ctrl+W)

I'm not sure what this does. I have never used this feature.

Move Up (Up Arrow)

This feature will scroll the level in the desired direction.

Move Left (Left Arrow)

This feature will scroll the level in the desired direction.

Move Right (Right Arrow)

This feature will scroll the level in the desired direction.

Move Down (Down Arrow)

This feature will scroll the level in the desired direction.

Custom Move Up (Alt+Up Arrow)

This will move the selected object TBD pixels in the desired direction. The number of pixels is determined by the "Set Custom Move Values..." feature.

Custom Move Left (Alt+Left Arrow)

This will move the selected object TBD pixels in the desired direction. The number of pixels is determined by the "Set Custom Move Values..." feature.

Custom Move Right (Alt+Right Arrow)

This will move the selected object TBD pixels in the desired direction. The number of pixels is determined by the "Set Custom Move Values..." feature.

Custom Move Down (Alt+Down Arrow)

This will move the selected object TBD pixels in the desired direction. The number of pixels is determined by the "Set Custom Move Values..." feature.

Set Custom Move Values...

This will feature will specify the number of pixels that an object is moved when using "Custom Move."

Zoom In

I don't think this feature works.

Zoom Out

I don't think this feature works.

No Draw

This is self-explanatory. I think this feature was used for development only.

Refresh (F5)

This will "redraw" the screen. This is used for those rare cases when the display of your level seems "messed" up. Using refresh will usually fix any "drawing/display" problems.

"Games" Menu

Claw

N/A for Gruntz.

Get Medieval

N/A for Gruntz.

Gruntz - Calc Cow IDs

Important! When you GruntzEdit for the very first time, make sure to select the "Games" menu and then select "Calc Cow lds" from the "Gruntz" sub-menu. This turns on a GruntzEdit feature that is required for certain game logics (e.g. switches).
Note: This step only needs to be done once. You should never have to do it again.

Basic Instruction

Step 1: Making a new level.

- 1) Double-click on the GruntzEdit.exe application.
- 2) Select "Open" from the "File" menu.
- 3) Open a template level from your "Custom" directory. See Tip #1 below.
(Note: The "Custom" directory should be located in the same directory as your Gruntz.exe)
- 4) If the level file is unable to find the REZ file, then perform *Step 2 "Setting the location of the game REZ file."*
- 5) After opening the template level, select "Save As..." from the "File" menu and save your level file with a new file name. (Note: Make sure to save your new file in the "Custom" directory.)
- 6) Now you should have a new level based on an existing template.

Tip #1: When you downloaded the GruntzEdit.zip file you should have also received "template" levels and "sample" levels for both Questz mode and Battlez mode. It is STRONGLY recommended that you make a backup of these levels and place a copy of them in your "Custom" directory. I also STRONGLY recommend that you ALWAYS make a new level by copying the appropriate template level and renaming the file.

Step 2: Setting the location of the game REZ file.

If the level file is unable to find the REZ file, then you will need to complete the following steps:

- 1) Make sure the Gruntz.rez file is installed on your hard drive. (Normal or Custom Installation)
- 2) Select "Properties" from the "World" menu.
- 3) Enter the path where your Gruntz.rez is located in the "Rez File" field.
(e.g. \GAMES\GRUNTZ\GRUNTZ.REZ)

Step 3: Setting the size of your level.

- 1) Select "Properties" from the "Plane" menu.
- 2) Enter the appropriate values in the "Tiles Wide:" and "Tiles High:" fields.
- 3) Click "OK".

Step 4: Setting the starting position of the level.

- 1) Select "Properties" from the "World" menu.
- 2) Enter the starting position (X, Y) coordinates in the (Start X, Start Y) fields.
- 3) Click "OK".

Tip #1: You can use CTRL+ALT+S to go to the start position in the level.

Tip #2: You can use CTRL+ALT+P to set a warp position in the level.

Tip #3: You can use CTRL+ALT+W to go to the warp position in the level.

Note: These tips are very handy for large levels! J

Step 5: Setting the level number (1-32) and level type.

- 1) Select "Properties" from the "World" menu.
- 2) **Single Player Levels (Questz):** Enter the level number and type by entering the text "Gruntz - Level 5" in the "Name" field.

Multiplayer Levels (Battlez): Enter the level number and type by entering the text "Gruntz - **Battlez** 5" in the "Name" field. The word "Battlez" MUST be in this field for the level to show up in the "Custom" worlds list when playing "Battlez".

Important! You must specify a level number (for Questz or Battlez) that represents the type of Area you want. Note: It doesn't matter which exact number you use, just as long as one of the numbers falls within the appropriate range. For example #1, #2, #3, or #4 can be used to make a level for Rocky Roadz.

# 1 through #4	= Rocky Roadz
# 5 through #8	= Gruntziclez
# 9 through #12	= Trouble in the Tropicz
# 13 through #16	= High on Sweetz
# 17 through #20	= High Rollerz
# 21 through #24	= Honey, I Shrunk the Gruntz!
# 25 through #28	= Miniature Masterz
# 29 through #32	= Gruntz in Space

Tip #1: If you use the supplied template levels from the Gruntz web site, you don't have to do this step since it is already done. You just need to make sure you use the appropriate template level. For example if you want to make a Questz level based on "Trouble in the Tropicz", you will need to use the QuestzTemplate3.wwd world file. You will notice that for the QuestzTemplate3.wwd file that the "Name" field already contains the text "Gruntz - Level 9".

Step 6: Place level tiles. (e.g. land, water, bridges, etc.)

- 1) Make sure you select "Tile Mode" by selecting "Tile Mode" from the "Edit" menu.
Note: You will always be in either one of 2 modes: "Object Mode" or "Tile Mode".
- 2) Activate the "Tile Selection Window" by selecting "Selection Window" from the "Tile" menu or simply press "Ctrl+T".
- 3) There are 2 different methods for placing level tiles:
Method 1: Left click on the tile in your world you would like to replace and then double click on a new tile in the "Selection Window".
Method 2: Hold the CTRL key down and left click on either an already placed tile in your world or on a new tile in the "Selection Window". This will "copy" that tile. To place the "copied" tile, simply right click in the area where you would like to insert the tile or hold the right mouse button and "paint" the area by dragging the mouse.

Step 7: Placing game logics. (e.g. pyramids, switches, powerups, etc.)

- 1) Make sure you select "Object Mode" by selecting "Object Mode" from the "Edit" menu.
Note: You will always be in either one of 2 modes: "Object Mode" or "Tile Mode".
- 2) Double click your left mouse button on the location where you would like to add the logic.
- 3) In the "Edit Objects" dialog box you will need to enter the appropriate information into certain specified fields and click "OK". (See the "Game Logics" section for game logic information.)

Tip #1: To edit an existing object, simply right click or double click with the left mouse button on the object to bring up the "Edit Objects" dialog box.

Tip #2: Don't forget that you can use CTRL-X (cut), CTRL-C (copy), and CTRL-V (paste) on game logics. This makes adding and moving triggers and switches a breeze!

Tip #3: If you notice an unusually long time to switch to object mode for the first time in a new level, this is because Gruntz is loading almost ALL the images of the game into memory. Since you only need a single image to place an object, we have added an option in GruntzEdit to force it to load only the first image of any animation. To use this option, select "Options..." from the "Tools" menu. Then check the "Single Image Loading" checkbox.

Step 8: Loading the custom level in the game.

- 1) Make sure to save the current level by selecting "Save" from the "File" menu.

Important! Make sure to save your level in the "Custom" directory. The "Custom" directory should be located in the same directory as your Gruntz.exe. If you do not do this, you will not be able to load the custom level in the game.

- 2) Start the game.
- 3) Select "Single Player"
- 4) Select "Custom Levelz"
- 5) Select your custom level from the dialog box and click OK.

Step 9: Making changes to your custom level while running the game.

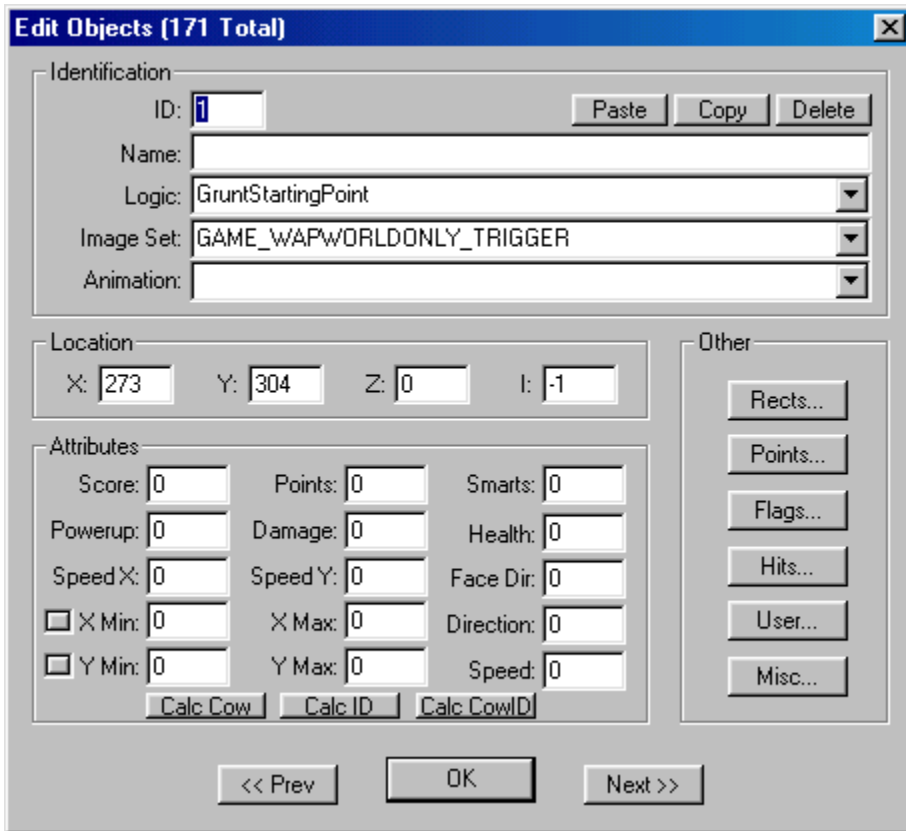
- 1) Press "Alt+Tab" while playing your custom level to return to the level editor. **Note:** GruntzEdit must be running for it to be available.
- 2) Make a change to your custom level and re-save your level.
- 3) Press "Alt+Tab" to return to your game (Gruntz.exe).
- 4) Press "Ctrl+Alt+R" to reload the custom level. That's it! J

Game Logics

Game Logics Dialog Boxes

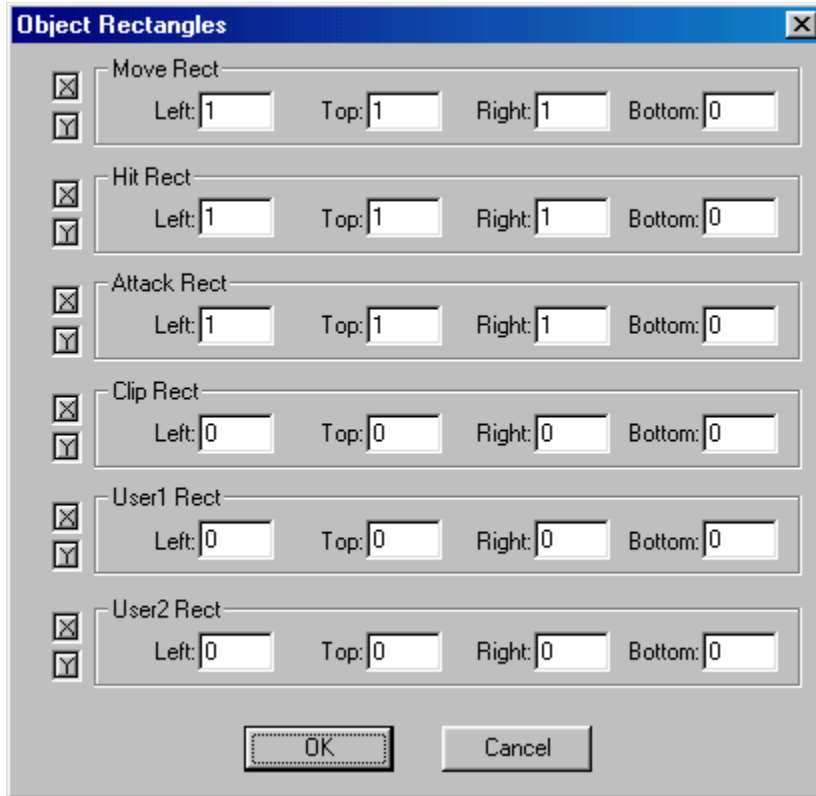
"Edit Objects" Dialog box.

Below is an image of the "Edit Objects" dialog box that is used to enter "game logic" information. You will use this dialog box MANY times throughout the development of your level. Notice the "Rects..." and "Flags..." buttons. Those buttons are used to bring up the "Object Rectangles" and "Object Flags" dialog boxes (described below) that will allow you to enter more information for certain game logics that require them.



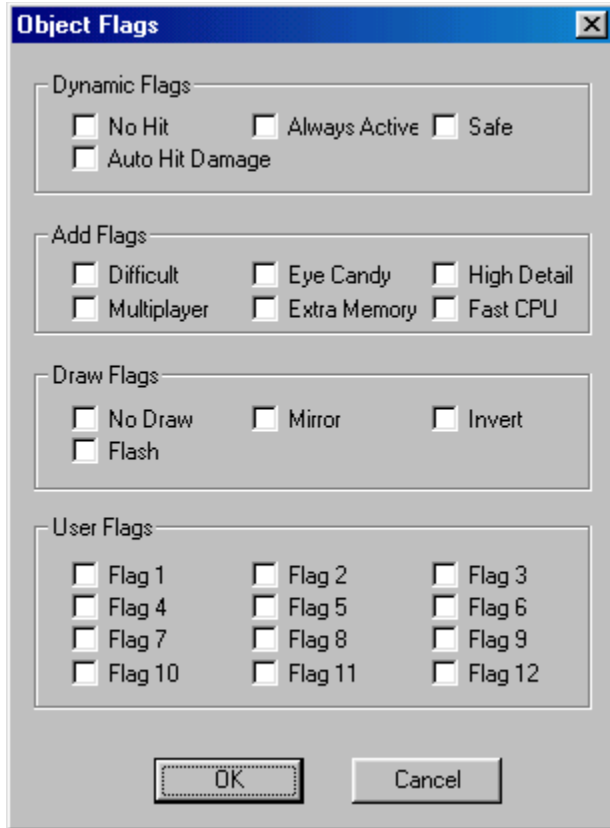
"Object Rectangles" Dialog box.

Below is an image of the "Object Rectangles" dialog box that is used for certain logics. (e.g. "GiantRock")



"Object Flags" Dialog box.

Below is an image of the "Object Flags" dialog box that is rarely used in Gruntz. The only time it is typically used is for setting the "No Draw" flag for certain logics. (e.g. "Brickz") Note: Most of the flags in this dialog box are not functional with Gruntz.



Game Logic Descriptions

Listed below are all the game logics in Gruntz grouped by category. The table is set-up to show all the possible fields in the "Edit Objects" dialog box that may or may not require information for that specific game logic. Make sure to enter all the required information in each field when adding a new logic.

Important! Make sure to read through all the game logic descriptions and become familiar with them. This information is 99% of what it takes to make a Gruntz level.

Note: Any feature/logic/image of GruntzEdit that is not listed in this document is most likely never used and shouldn't be used.

Game Logic Categories:

Brickz and Fortressez
 Gruntz
 Eye Candiez
 Hazardz
 Itemz (Toyz, Toolz, Powerupz, etc)
 Soundz and Voicez
 Switchez, Pyramidz, Bridgez, and Checkpointz
 Teleporterz
 Misc

Brickz and Fortressez

Listed below are all the game logics for placing brickz, fortz, and fort flagz in the levelz.

Brickz

Important! This logic ALWAYS needs to be placed on ANY brick tile (Tile #303-#329). If you try to break a brick that does not have the "Brickz" object, the game will exit! The other purpose of the "Brickz" object is to specify what players (Battlez) will be able to see the colored brickz (use the Move Rects fields to specify this).

Edit Objects Dialog Box

Name: N/A
Logic: "Brickz"
Image Set: "GAME_WAPWORLDONLY_BRICK"

Object Rectangles Dialog Box

Move Rect: Left "0" - Player 1 cannot see the colored brick.
"1" - Player 1 can see the colored brick.
Note: In single player, player 1 is treated as the player. Therefore using the "Move Rect: Left" field will specify whether the player sees a colored brick or not.

Move Rect: Top "0" - Player 2 cannot see the colored brick.
"1" - Player 2 can see the colored brick.

Move Rect: Right "0" - Player 3 cannot see the colored brick.
"1" - Player 3 can see the colored brick.

Move Rect: Bottom "0" - Player 4 cannot see the colored brick.
"1" - Player 4 can see the colored brick.

Object Flags Dialog Box

No Draw Checked

ExitTrigger

The ExitTrigger game logic is always assigned to the "LEVEL_FORT" and is used as the exit in a "Questz" game or as a player's fort in a "Battlez" game.

Edit Objects Dialog Box

Logic: "ExitTrigger"
Image Set: "LEVEL_FORT"
Smarts: "0" - Player 1 fort (Battlez only)
"1" - Player 2 fort (Battlez only)
"2" - Player 3 fort (Battlez only)
"3" - Player 4 fort (Battlez only)

FortressFlag

This game logic is primarily used to place fortress flags to mark a player's fortress. Fortress flags aren't really required, but they look cool. J

Edit Objects Dialog Box

Logic: "FortressFlag"
Image Set: "GAME_FORTRESSFLAGZ_KING"
"GAME_FORTRESSFLAGZ_NAPOLEAN"
"GAME_FORTRESSFLAGZ_PATTON"
"GAME_FORTRESSFLAGZ_VIKING"
Smarts: "0" - Player 1 fortress flag (Battlez and Questz)
"1" - Player 2 fortress flag (Battlez only)
"2" - Player 3 fortress flag (Battlez only)
"3" - Player 4 fortress flag (Battlez only)

Gruntz

Listed below are all the game logics for placing Gruntz (friendly or enemy), Gruntz "creation padz", and Grunt goo puddlez.

GruntCreationPoint

The GruntCreationPoint is an object that is placed that allows newly "baked" Gruntz to be dropped onto. In "Battlez", the GruntCreationPoints are owned by a player. In other words, another player cannot drop his/her Grunt onto someone else's GruntCreationPoint. Although, I don't think I ever used one, but I think you can make a special GruntCreationPoint that anyone can drop a Grunt onto by setting the "Smarts" field to "4".

Important! Placing GruntCreationPoints is the primary means of creating Gruntz in a "Battlez" map. You do NOT use GruntStartingPoint logics in a "Battlez" map.

Edit Objects Dialog Box

Logic:	"GruntCreationPoint"
Image Set:	"GAME_GRUNTCREATIONPOINT"
Smarts:	"0" - Player 1 creation point (Battlez and Questz) "1" - Player 2 creation point (Battlez only) "2" - Player 3 creation point (Battlez only) "3" - Player 4 creation point (Battlez only)

GruntPuddle

The GruntPuddle logic is used to place Grunt puddlez in your levels for GooberzGruntz to suck up.

Edit Objects Dialog Box

Logic:	"GruntPuddle"
Image Set:	"GAME_WAPWORLDONLY_TRIGGER"
Score:	This field allows you to change the color of the puddle. Note: It doesn't matter what color the puddle of goo is, anyone's Grunt can suck it up. "0" - Orange (the default Grunt color) "1" - Green "2" - Blue "3" - Red "4" - Purple "5" - Yellow "6" - Hot Pink "7" - Black "8" - Dark Blue "9" - Dark Green "10" - Turquoise "11" - Dark Red "12" - Pink "13" - Dark Yellow "14" - Grey "15" - Cyan "16" - White

GruntStartingPoint (Questz Only)

The GruntStartingPoint logic is the primary means of placing friendly or enemy Gruntz in your "Questz" levelz.

Important! You do NOT use GruntStartingPoint logics in a "Battlez" map. Instead you place GruntCreationPoints as the primary means of creating Gruntz in a "Battlez" map.

Edit Objects Dialog Box

Logic:	"GruntStartingPoint"
Image Set:	"GAME_WAPWORLDONLY_TRIGGER"
Points:	This field specifies the AI type (for enemy Gruntz only) e.g. "4" - DefenderGrunt (See "Appendix C: Enemy AI Types" for more information.)
Smarts:	This field determines what "player" the Grunt is. This value must be "0", "1", "2", or "3". If it is anything other than "0" it will be an enemy Grunt. Note: Even enemy Gruntz will attack other enemy Gruntz that have a different "player" value than they do. "0" - Player's Grunt "1" - Enemy Grunt (team 1) "2" - Enemy Grunt (team 2) "3" - Enemy Grunt (team 3)
Powerup:	This field determines what tool the Grunt has. e.g. "5" (See "Appendix B: Item ID Numbers" for a list of the ID numbers.)
Damage:	This field determines what toy the Grunt has. Note: All enemy Gruntz with the exception of the Toyer AI can not use Toyz. So giving a Toy to an AI Grunt will have no effect in most cases. e.g. "5" (See "Appendix B: Item ID Numbers" for a list of the ID numbers.)
Direction:	This field contains the radius at which the Grunt will "see" you from depending on the enemy AI type.
X Min:	In the case of an ObjectGuard AI (See "Appendix C: Enemy AI Types" for more information), this field represents the X tile coordinate of the object to be guarded. Note: If you make an ObjectGuard AI, be sure to place him on a tile that is adjacent to the location that he will be guarding.
X Max:	In the case of an ObjectGuard AI (See "Appendix C: Enemy AI Types" for more information), this field represents the Y tile coordinate of the object to be guarded. Note: If you make an ObjectGuard AI, be sure to place him on a tile that is adjacent to the location that he will be guarding.

Object Rectangles Dialog Box

Move Rect: Left	This field specifies the number of tiles to the left of the original Grunt position that the Grunt will wander randomly (applies to only certain enemy AI types)
Move Rect: Top	This field specifies the number of tiles above the original Grunt position that the Grunt will wander randomly (applies to only certain enemy AI types)
Move Rect: Right	This field specifies the number of tiles to the right of the original Grunt position that the Grunt will wander randomly (applies to only certain enemy AI types)
Move Rect: Bottom	This field specifies the number of tiles below the original Grunt position that the Grunt will wander randomly (applies to only certain enemy AI types)

Eye Candiez

Listed below are all the game logics for placing "eye candiez". The "eye candy" logics are used to simply place objects that serve no purpose, but to make the level look cool (e.g. trees, rocks, candy, plants, etc.)

BehindCandy

This logic is used for eye-candies that you always want to be displayed underneath a Grunt. For example, I used this logic for the small pebbles on the ground in the Rocky Roadz levels so that they were always displayed underneath Gruntz.

Edit Objects Dialog Box

Logic:	"BehindCandy"
Image Set:	This should be set to any image set name is an eye-candy that you want to display behind all Gruntz. e.g. "LEVEL_ROCKZ1"

BehindCandyAni

This logic is used for animating eye-candies that you always want to be displayed underneath a Grunt. For example, I used this logic for the water sparkles that I placed on the water tiles. This logic was used because I wanted the sparkles to animate and always wanted the sparkles to be displayed underneath any ToobGruntz that may be floating on the water.

Edit Objects Dialog Box

Logic: "BehindCandyAni"
Image Set: This should be set to any image set name that is an animating eye-candy
e.g. "LEVEL_WATER1"
Animation: Normally this field is left blank, but it was used for the water sparkles.
e.g. "GAME_WATER1" (Note: this is an animation file that is used to control the animation speed of the sparkles)

EyeCandy

This logic is used for non-animating eye-candies (e.g. trees, plants, etc). This logic will automatically set a "Z-order" value based on it's position such that Gruntz will always walk behind them if they are tall enough (e.g. trees). If you want an eye candy that is always displayed behind Gruntz, then use the BehindCandy logic instead. **Note:** make sure to place a solid tile (#199 or #200) underneath eye-candies that you want Gruntz to "bump" into such as treez.

Edit Objects Dialog Box

Logic: "EyeCandy"
Image Set: This should be set to any image set name that is an eye-candy
e.g. "LEVEL_TREE1"

EyeCandyAni

This logic is used for animating eye-candies that you want Gruntz to walk behind or in front of depending on the eye candy's height. This logic will automatically set a "Z-order" value based on it's position such that Gruntz will always walk behind them if they are tall enough (e.g. trees). If you want an animating eye candy that is always displayed behind Gruntz, then use the BehindCandyAni logic instead. **Note:** make sure to place a solid tile (#199 or #200) underneath eye-candies that you want Gruntz to "bump" into such as treez.

Edit Objects Dialog Box

Logic: "EyeCandy"
Image Set: This should be set to any image set name that is an eye-candy
e.g. "LEVEL_TREE1"

Hazardz

Listed below are all the game logics for placing "hazardz" in your levels (e.g. lava geysers, kitchen slime, thunder clouds, UFOs, etc).

KitchenSlime

KitchenSlime is the logic for the hazard used in "Area 6 - Honey, I Shrunk the Gruntz!" I'm pretty sure you can use this "hazard" in any area, but I have never tried it. The KitchenSlime is limited to traveling in a rectangular patch. To place a KitchenSlime, place it on the tile that is the upper left or lower right corner of it's path, then set the tile coordinates of the opposite corner of it's path in the "Speed X" and "Speed Y" fields. Note: to find the tile coordinates of a particular tile, go to Tile mode and click on a tile. Displayed in the bottom right of the screen should be the tile coordinates (e.g. 25, 50).

Edit Objects Dialog Box

Logic: "KitchenSlime"
Image Set: "LEVEL_KITCHENSLIME_WEST" (Note: I think it will start in the direction depending on which image you use (e.g. _NORTH, _SOUTH, _EAST, _WEST)).

- Speed X:** This field should contain the X coordinate (tile coordinates) of the tile that defines the opposite corner of the path that the slime travels.
e.g. "25" (The 25th tile from the very left of the map)
- Speed Y:** This field should contain the X coordinate (tile coordinates) of the tile that defines the opposite corner of the path that the slime travels.
e.g. "50" (The 50th tile from the very top of the map)

ObjectDropper

This ObjectDropper is a generic logic that is assigned to the bird in "Area 3 - Trouble in the Tropicz" and the airplane in "Area4 - High on Sweetz". The ObjectDropper is limited to flying in one direction only. To place an ObjectDropper, simply place the logic in the path you would like it to travel and pick the appropriate image that is pointed in the direction you would like it to travel.

Edit Objects Dialog Box

- Logic:** "ObjectDropper"
- Image Set:** e.g. "LEVEL_OBJECTDROPPER_NORTH" (travels in the north direction)
e.g. "LEVEL_OBJECTDROPPER_SOUTH" (travels in the south direction)
e.g. "LEVEL_OBJECTDROPPER_EAST" (travels in the east direction)
e.g. "LEVEL_OBJECTDROPPER_WEST" (travels in the west direction)
- Speed:** The value in this field will control the speed of the ObjectDropper. The default speed will be used if this value is "0". Speed is defined as the amount of time it takes for an object to travel the distance of one tile in any direction.
e.g. "1000" (Travels one tile in 1 second)

RainCloud

The RainCloud object is hazard that was used in "Area 7 - Miniature Masterz". The logic is always assigned to the "LEVEL_RAINCLOUD" image and is given a path that is defined by the Rects fields. To place a RainCloud, first position the cloud on a tile that is its starting "waypoint". Then using the Rects fields, specify the X and Y tile coordinates for each successive waypoint. Note: to find the tile coordinates of a particular tile, go to Tile mode and click on a tile. Displayed in the bottom right of the screen should be the tile coordinates (e.g. 25, 50).

Edit Objects Dialog Box

- Logic:** "RainCloud"
- Image Set:** "LEVEL_RAINCLOUD"
- Speed:** The value in this field will control the speed of the RainCloud. The default speed will be used if this value is "0". Speed is defined as the amount of time it takes for an object to travel the distance of one tile in any direction.
e.g. "1000" (Travels one tile in 1 second)
- Damage:** The value in this field will control how long the RainCloud stops and waits at each waypoint.
e.g. "1000" (Stops for 1 second at each waypoint)

Object RectanglesDialog Box

- Move Rect: Left** X Tile Coordinate of its 1st waypoint.
- Move Rect: Top** Y Tile Coordinate of its 1st waypoint.
- Move Rect: Right** X Tile Coordinate of its 2nd waypoint.
- Move Rect: Bottom** Y Tile Coordinate of its 2nd waypoint.
- Hit Rect: Left** X Tile Coordinate of its 3rd waypoint.
- Hit Rect: Top** Y Tile Coordinate of its 3rd waypoint.
- Hit Rect: Right** X Tile Coordinate of its 4th waypoint.
- Hit Rect: Bottom** Y Tile Coordinate of its 4th waypoint.
- Attack Rect: Left** X Tile Coordinate of its 5th waypoint.
- Attack Rect: Top** Y Tile Coordinate of its 5th waypoint.
- Attack Rect: Right** X Tile Coordinate of its 6th waypoint.
- Attack Rect: Bottom** Y Tile Coordinate of its 6th waypoint.

Clip Rect: Left	X Tile Coordinate of its 7th waypoint.
Clip Rect: Top	Y Tile Coordinate of its 7th waypoint.
Clip Rect: Right	X Tile Coordinate of its 8th waypoint.
Clip Rect: Bottom	Y Tile Coordinate of its 8th waypoint.
User1 Rect: Left	X Tile Coordinate of its 9th waypoint.
User1 Rect: Top	Y Tile Coordinate of its 9th waypoint.
User1 Rect: Right	X Tile Coordinate of its 10th waypoint.
User1 Rect: Bottom	Y Tile Coordinate of its 10th waypoint.
User2 Rect: Left	X Tile Coordinate of its 11th waypoint.
User2 Rect: Top	Y Tile Coordinate of its 11th waypoint.
User2 Rect: Right	X Tile Coordinate of its 12th waypoint.
User2 Rect: Bottom	Y Tile Coordinate of its 12th waypoint.

RollingBall

The RollingBall logic is used to control the rolling ball hazards in each area (e.g. the boulder in "Area 1 - Rock Roadz"). RollingBalls will always travel in one direction until it hits an arrow that may or may not change its current direction. If a RollingBall hits any solid tiles, it will be destroyed. To place a RollingBall, simply place the logic in the path you would like it to travel and pick the appropriate image that is pointed in the direction you would like it to travel (e.g. LEVEL_ROLLINGBALL_SOUTH).

Edit Objects Dialog Box

Logic:	"RollingBall"
Image Set:	"LEVEL_ROLLINGBALL_NORTH" (starts traveling in the north direction) "LEVEL_ROLLINGBALL_SOUTH" (starts traveling in the south direction) "LEVEL_ROLLINGBALL_EAST" (starts traveling in the east direction) "LEVEL_ROLLINGBALL_WEST" (starts traveling in the west direction)
Speed:	The value in this field will control the speed of the RollingBall. The default speed will be used if this value is "0". Speed is defined as the amount of time it takes for an object to travel the distance of one tile in any direction. e.g. "1000" (Travels one tile in 1 second)

SpotLight

The SpotLight is a hazard that was used in "Area 5 - High Rollerz." The SpotLight is limited to traveling in a circle with the radius of the circle being specified by the "Smarts" field. The center of the circular path is specified by the original position of the logic.

Edit Objects Dialog Box

Logic:	N/A
Image Set:	"LEVEL_SPOTLIGHT"
Smarts:	This field determines the radius of the circular path in # tiles. e.g. "4" (4 tile radius)
Damage:	The value in this field will control the speed of the SpotLight. The default speed will be used if this value is "0". The speed of a spotlight is defined as the amount of time it takes for the spotlight to do one full rotation. e.g. "1000" (Travels one full rotation in 1 second)
Direction:	The direction that the spotlight will spin. "0" - counterclockwise "1" - clockwise

StaticHazard

The StaticHazard logic is a generic logic that is used for various "static hazardz" in different Areas (e.g. Lava geysers in "Area 3 - Trouble in the Tropicz"). Typically, the StaticHazard object should be placed on a StaticHazard tile (Tile #193 or

#194). **Important!** The StaticHazard tile for Area5 - "High Rollerz" is tile #115.

Edit Objects Dialog Box

Logic: "StaticHazard"
Image Set: "LEVEL_STATICHAZARD" (Area3 - Lava Geyser)
"LEVEL_STATICHAZARD" (Area4 - Birthday Candles)
"LEVEL_STATICHAZARD" (Area5 - Trap Door)
"LEVEL_STATICHAZARD" (Area6 - Electrical Outlet)
"LEVEL_STATICHAZARD" (Area7 - Golf Ball Geyser)
"LEVEL_STATICHAZARD" (Area8 - Alien Goo Geyser)
Points: This field specifies the first-time-only initial delay before the first hazard occurrence. This is normally used if you want many hazards to all have the same inactive delays, but you want to stagger their starting times.
e.g. "5000" (a 5 second initial delay)
Damage: This field specifies the time the hazard is inactive.
e.g. "5000" (hazard is inactive for 5 seconds)

ToobSpikez

This object is used on the bridge transition tiles (Tiles #282-285) to "pop" the ToobGruntz inner-tube.

Edit Objects Dialog Box

Logic: "ToobSpikez"
Image Set: e.g. "GAME_TOOBSPIKEZHORIZ" (used on tiles #282 and #285)
e.g. "GAME_TOOBSPIKEZVERT" (used on tiles #283 and #284)

UFO

The UFO object is hazard that was used in "Area 8 - Gruntz in Space". The logic is always assigned to the "LEVEL_UFO" image and is given a path that is defined by the Rects fields. To place a UFO, first position the UFO on a tile that is its starting "waypoint". Then using the Rects fields, specify the X and Y tile coordinates for each successive waypoint. Note: to find the tile coordinates of a particular tile, go to Tile mode and click on a tile. Displayed in the bottom right of the screen should be the tile coordinates (e.g. 25, 50).

Edit Objects Dialog Box

Logic: "UFO"
Image Set: "LEVEL_UFO"
Speed: The value in this field will control the speed of the UFO. The default speed will be used if this value is "0". Speed is defined as the amount of time it takes for an object to travel the distance of one tile in any direction.
e.g. "1000" (Travels one tile in 1 second)
Damage: The value in this field will control how long the UFO stops and waits at each waypoint.
e.g. "1000" (Stops for 1 second at each waypoint)

Object RectanglesDialog Box

Move Rect: Left X Tile Coordinate of its 1st waypoint.
Move Rect: Top Y Tile Coordinate of its 1st waypoint.
Move Rect: Right X Tile Coordinate of its 2nd waypoint.
Move Rect: Bottom Y Tile Coordinate of its 2nd waypoint.
Hit Rect: Left X Tile Coordinate of its 3rd waypoint.
Hit Rect: Top Y Tile Coordinate of its 3rd waypoint.
Hit Rect: Right X Tile Coordinate of its 4th waypoint.
Hit Rect: Bottom Y Tile Coordinate of its 4th waypoint.
Attack Rect: Left X Tile Coordinate of its 5th waypoint.
Attack Rect: Top Y Tile Coordinate of its 5th waypoint.
Attack Rect: Right X Tile Coordinate of its 6th waypoint.

Attack Rect: Bottom	Y Tile Coordinate of its 6th waypoint.
Clip Rect: Left	X Tile Coordinate of its 7th waypoint.
Clip Rect: Top	Y Tile Coordinate of its 7th waypoint.
Clip Rect: Right	X Tile Coordinate of its 8th waypoint.
Clip Rect: Bottom	Y Tile Coordinate of its 8th waypoint.
User1 Rect: Left	X Tile Coordinate of its 9th waypoint.
User1 Rect: Top	Y Tile Coordinate of its 9th waypoint.
User1 Rect: Right	X Tile Coordinate of its 10th waypoint.
User1 Rect: Bottom	Y Tile Coordinate of its 10th waypoint.
User2 Rect: Left	X Tile Coordinate of its 11th waypoint.
User2 Rect: Top	Y Tile Coordinate of its 11th waypoint.
User2 Rect: Right	X Tile Coordinate of its 12th waypoint.
User2 Rect: Bottom	Y Tile Coordinate of its 12th waypoint.

Itemz (Toolz, Toyz, Powerupz, etc)

Listed below are all the game logics for placing itemz in the levelz (e.g. toolz, toyz, powerupz, cursez, etc.) as well under rockz and dirt moundz.

CoveredPowerup

This game logic is placed on top of "breakable tiles" (e.g. rockz) or "diggable tiles" (e.g. holez) to place an item underneath them or to place a specific tile that you would like other than the default ground tile.

Edit Objects Dialog Box

Logic:	"CoveredPowerup"
Image Rect:	"GAME_WAPWORLDONLY_TRIGGER"
Score:	<p>If the covered item is a megaphone then this field should contain the "order" in which the megaphone item appears in the GruntMachine on the Resource tab of the status bar.</p> <p>e.g. "1" (it will be the first item that shows up in the GruntMachine)</p> <p>If the covered item is a ToyBox then this field represents the player that owns the ToyBox.</p> <p>"0" - Player 1's team owns the ToyBox.</p> <p>"1" - Player 2's team owns the ToyBox.</p> <p>"2" - Player 3's team owns the ToyBox.</p> <p>"3" - Player 4's team owns the ToyBox.</p>
Points:	<p>If the covered item is a megaphone then this field should contain the ID number for the tool or toy that appears in the GruntMachine on the Resource tab of the status bar.</p> <p>e.g. "5" (the item that shows up in the GruntMachine will be Gauntletz)</p> <p>If the item is a ToyBox then this field represents which toy is inside. See "Appendix B: Item ID Numbers" for a list of the Toy ID numbers.</p> <p>e.g. "24" (a Beach Ball would be inside the ToyBox)</p>
Smarts:	<p>This field should contain the tile # of the tile you would like placed when the diggable tile is dug up or the breakable tile is broken.</p> <p>e.g. "241" (This will place the secret switch underneath the diggable tile or breakable tile. Important! Make sure to place a "TileTriggerSwitch" logic on top of the tile if there is to be a switch underneath it.)</p>
Powerup:	<p>This field should contain the ID number for the tool, toy, or powerup you would like to place underneath the breakable tile or digglable tile. See "Appendix B: Item ID Numbers" for a list of the ID numbers.</p> <p>e.g. "99" (covered time bomb)</p>
Face Dir:	<p>If the covered item is a timed powerup or curse then this field should contain the time the powerup will last in milliseconds.</p> <p>e.g. "30000" (30 seconds)</p>

If the covered item is a scroll or wand then this field should tell which spell it is. (See Spellz in Appendix B for a list)

Object FlagsDialog Box

No Draw

Checked

GiantRock

The GiantRock game logic is required if you want to place the "GiantRock" (Area1), "GiantCupcake" (Area4), "GiantDice" (Area5), etc. The "GiantBreakableItem" is always constructed from the tiles #286-#294. "GiantBreakableItems" when destroyed can leave behind any tiles you want by specifying the tiles in the Rects fields.

Edit Objects Dialog Box

Name: N/A

Logic: N/A

Image Set: "GAME_WAPWORLDDONLY_TRIGGER"

Score: If the item underneath the GiantRock is a megaphone then this field should contain the "order" in which the megaphone item appears in the GruntMachine on the Resource tab of the status bar.

e.g. "1" (it will be the first item that shows up in the GruntMachine)

If the item is a ToyBox then this field represents the player that owns the ToyBox.

"0" - Player 1's team owns the ToyBox.

"1" - Player 2's team owns the ToyBox.

"2" - Player 3's team owns the ToyBox.

"3" - Player 4's team owns the ToyBox.

Points: If the item underneath the GiantRock is a megaphone then this field should contain the ID number for the tool or toy that appears in the GruntMachine on the Resource tab of the status bar.

e.g. "5" (the item that shows up in the GruntMachine will be Gauntletz)

If the item is a ToyBox then this field represents which toy is inside. See "Appendix B: Item ID Numbers" for a list of the Toy ID numbers.

e.g. "24" (a Beach Ball would be inside the ToyBox)

Powerup: This field should contain the ID number for the tool, toy, or powerup you would like to place underneath the GiantRock. See "Appendix B: Item ID Numbers" for a list of the ID numbers.

e.g. "99" (covered time bomb)

Face Dir: If the item underneath the GiantRock is a timed powerup or curse then this field should contain the time the powerup will last in milliseconds.

e.g. "30000" (30 seconds)

If the item underneath the GiantRock is a scroll or wand then this field should tell which spell it is. See "Appendix B: Item ID Numbers" for a list of the Spell ID numbers.

Object RectanglesDialog Box

Move Rect: Left Upper left tile #
e.g. "1" (regular ground tile)

Move Rect: Top Upper center tile #
e.g. "1" (regular ground tile)

Move Rect: Right Upper right tile #
e.g. "1" (regular ground tile)

Hit Rect: Left Middle center tile #
e.g. "1" (regular ground tile)

Hit Rect: Top Middle right tile #
e.g. "1" (regular ground tile)

Hit Rect: Right Middle left tile #
e.g. "1" (regular ground tile)

Attack Rect: Left Bottom left tile #
e.g. "1" (regular ground tile)

Attack Rect: Top	Bottom center tile # e.g. "1" (regular ground tile)
Attack Rect: Right	Bottom left tile # e.g. "1" (regular ground tile)

Object FlagsDialog Box

No Draw	Checked
----------------	---------

InGamelcon

This is the main logic that is used for placing all toolz, toyz, powerupz, and cursez on the ground.

Edit Objects Dialog Box

Logic:	"InGamelcon"
Image Set:	The image set can be anyone of the images that starts with "GAME_INGAMEICONZ_" e.g. "GAME_INGAMEICONZ_TOOLZ_BOOMERANGZ" Note: For a ToyBox, use "GAME_TOYBOX" here.
Score:	If the item is a megaphone then this field should contain the "order" in which the megaphone item appears in the GruntMachine on the Resource tab of the status bar. e.g. "1" (it will be the first item that shows up in the GruntMachine) If the item is a ToyBox then this field represents the player that owns the ToyBox. "0" - Player 1's team owns the ToyBox. "1" - Player 2's team owns the ToyBox. "2" - Player 3's team owns the ToyBox. "3" - Player 4's team owns the ToyBox.
Points:	If the item is a megaphone then this field should contain the ID number for the tool or toy that appears in the GruntMachine on the Resource tab of the status bar. e.g. "5" (the item that shows up in the GruntMachine will be Gauntletz) If the item is a ToyBox then this field represents which toy is inside. See "Appendix B: Item ID Numbers" for a list of the Toy ID numbers. e.g. "24" (a Beach Ball would be inside)
Powerup:	This field should contain the ID number for the tool, toy, or powerup . See "Appendix B: Item ID Numbers" for a list of the ID numbers. e.g. "5" (Gauntletz)
Damage:	If the item is a re-spawning powerup or curse then this field should contain the time it takes for the item to respawn. e.g. "60000" (60 seconds)
Face Dir:	If the item is a timed powerup or curse then this field should contain the time the powerup will last in milliseconds. e.g. "30000" (30 seconds) If the item is a scroll or wand then this field should tell which spell it is. See "Appendix B: Item ID Numbers" for a list of the Spell ID numbers.

Soundz and Voicez

Listed below are all the game logics for placing soundz and Grunt voicez in the levelz.

GlobalAmbientSound

Ambient sounds are typically used for sounds that would be heard throughout the level. Although, some ambient sounds are used only in specific areas of the level. To specify a specific area in which the ambient sound is heard, use the Xmin, Xmax, Ymin, Ymax fields. **Note:** It does not matter where on the level you place this logic because the location of the sound is determined by the Xmin, Xmax, Ymin, and Ymax fields, but it is usually a good idea to place it near the area in which the sound is being played.

Edit Objects Dialog Box

Logic:	"GlobalAmbientSound"
Image Set:	"GAME_SOUNDICON"
Animation:	This field specifies which sound file to play. See "Appendix A: Ambient Sounds" for a list of available ambient sounds. e.g. "LEVEL_AMBIENT_AMBIENTLOOP1"
Damage:	This field controls the volume of the ambient sound. Valid range is "0" through "100" ("100" being the loudest)
X Min:	This field sets the minimum X coordinate of the area where the sound can be heard. e.g. "500"
X Max:	This field sets the maximum X coordinate of the area where the sound can be heard. e.g. "1000"
Y Min:	This field sets the minimum Y coordinate of the area where the sound can be heard. e.g. "250"
Y Max:	This field sets the maximum Y coordinate of the area where the sound can be heard. e.g. "1500"

Object RectanglesDialog Box

Move Rect: Left	This field sets the minimum "ON" time the sound will play. e.g. "500" (1/2 a second)
Move Rect: Top	This field sets the maximum "ON" time the sound will play. e.g. "1000" (1 second)
Move Rect: Right	This field sets the minimum "OFF" time the sound will play. e.g. "10000" (10 seconds)
Move Rect: Bottom	This field sets the maximum "OFF" time the sound will play. e.g. "20000" (20 seconds)

Object FlagsDialog Box

No Draw	Checked
----------------	---------

VoiceTrigger

This logic is used to trigger the Grunt dialogue.

Edit Objects Dialog Box

Logic:	"VoiceTrigger"
Image Set:	"GAME_WAPWORLDONLY_VOICE"
Smarts:	This field specifies which "Voice Group" to randomly play a voice from.
Health:	This field specifies which specific voice file to play from the "Voice Group". If "0", the game will randomly pick a voice file from the "Voice Group". See "Appendix D: Grunt Voice List" for the list of available Grunt dialogue.

Object RectanglesDialog Box

Move Rect: Left	This field specifies the number of tiles to the left of the original logic position that the Grunt will trigger the voice file
Move Rect: Top	This field specifies the number of tiles above the original logic position that the Grunt will trigger the voice file
Move Rect: Right	This field specifies the number of tiles to the right of the original logic position that the Grunt will trigger the voice file
Move Rect: Bottom	This field specifies the number of tiles below the original logic position that the Grunt will trigger the voice file

Object FlagsDialog Box

No Draw	Checked
----------------	---------

Switches, Pyramidz, Bridgez, and Checkpointz

Listed below are all the game logics for placing switches, pyramidz, bridgez and checkpointz.

CheckpointTrigger (Questz Only)

The CheckPoint object is used to make checkpoints. The logic CheckPoint needs to be given to the checkpoint flag image.

Edit Objects Dialog Box

Logic: "CheckpointTrigger"
Image Set: "GAME_CHECKPOINTFLAG"

Object Rectangles Dialog Box

Move Rect: Left	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5122" (switch1 ID#)
Move Rect: Top	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5123" (switch2 ID#)
Move Rect: Right	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5124" (switch3 ID#)
Move Rect: Bottom	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5125" (switch4 ID#)
Hit Rect: Left	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5126" (switch5 ID#)
Hit Rect: Top	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5127" (switch6 ID#)
Hit Rect: Right	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5128" (switch7 ID#)
Hit Rect: Bottom	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5129" (switch8 ID#)
Attack Rect: Left	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5130" (switch9 ID#)
Attack Rect: Top	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5131" (switch10 ID#)
Attack Rect: Right	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5132" (switch11 ID#)
Attack Rect: Bottom	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5133" (switch12 ID#)
Clip Rect: Left	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5134" (switch13 ID#)
Clip Rect: Top	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5135" (switch14 ID#)
Clip Rect: Right	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5136" (switch15 ID#)
Clip Rect: Bottom	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5137" (switch16 ID#)
User1 Rect: Left	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag.

	e.g. "5138" (switch17 ID#)
User1 Rect: Top	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5139" (switch18 ID#)
User1 Rect: Right	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5140" (switch19 ID#)
User1 Rect: Bottom	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5141" (switch20 ID#)
User2 Rect: Left	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5142" (switch21 ID#)
User2 Rect: Top	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5143" (switch22 ID#)
User2 Rect: Right	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5144" (switch23 ID#)
User2 Rect: Bottom	This field needs to contain the ID# of any checkpoint switch that is supposed to raise the checkpoint flag. e.g. "5145" (switch24 ID#)

TileTrigger

The TileTrigger logic is used to toggle pyramids and bridges up and down. **Note:** Tiles #257-260 are timed toggle bridges that automatically move up and down repeatedly without the need of an associated TileTriggerSwitch logic.

Important! When you place **any switch** that will trigger a TileTrigger, **MAKE SURE** to press the "Calc Cow ID" button for the "TileTriggerSwitch" logic before you copy the "TileTriggerSwitch" ID # into the "TileTrigger" Rects fields. Also, if you EVER move a switch, make sure to press the "Calc Cow ID" button AGAIN and re-copy the new "TileTriggerSwitch" ID # into the "TileTrigger" Rects field. Note: The "Calc Cow ID" button will re-calculate an ID# based on the switches position. If the switch is moved and the "Calc Cow ID" button is not pressed, the game will exit when you step on the switch.

Note: The ID # of a switch is copied from the "ID" field of the "TileTriggerSwitch" logic for the switch.

Edit Objects Dialog Box

Logic:	"TileTrigger"
Image Set:	"GAME_WAPWORLDDONLY_TRIGGER"
Points:	If this tile is a Toggle Bridge that goes up and down repeatedly without a switch (tile #257-260), or if this tile is a Silver Timed Pyramid (tile #245/246), then this field specifies the first-time-only initial delay before the tile changes into its "toggled" state.
Damage:	If this tile is a Toggle Bridge that goes up and down repeatedly without a switch (tile #257-260), or if this tile is a Silver Timed Pyramid (tile #245/246), then this field specifies the amount of time that it will stay in the "toggled" state.
Health:	If this tile is a Toggle Bridge that goes up and down repeatedly without a switch (tile #257-260), then this field specifies how long it will remain in its original state before changing to its "toggles" state.

Object Rectangles Dialog Box

Move Rect: Left	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5122" (switch1 ID#)
Move Rect: Top	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5123" (switch2 ID#)
Move Rect: Right	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5124" (switch3 ID#)
Move Rect: Bottom	This field needs to contain the ID# of any switch that is supposed to raise or lower the

	pyramid or bridge. e.g. "5125" (switch4 ID#)
Hit Rect: Left	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5126" (switch5 ID#)
Hit Rect: Top	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5127" (switch6 ID#)
Hit Rect: Right	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5128" (switch7 ID#)
Hit Rect: Bottom	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5129" (switch8 ID#)
Attack Rect: Left	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5130" (switch9 ID#)
Attack Rect: Top	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5131" (switch10 ID#)
Attack Rect: Right	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5132" (switch11 ID#)
Attack Rect: Bottom	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5133" (switch12 ID#)
Clip Rect: Left	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5134" (switch13 ID#)
Clip Rect: Top	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5135" (switch14 ID#)
Clip Rect: Right	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5136" (switch15 ID#)
Clip Rect: Bottom	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5137" (switch16 ID#)
User1 Rect: Left	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5138" (switch17 ID#)
User1 Rect: Top	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5139" (switch18 ID#)
User1 Rect: Right	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5140" (switch19 ID#)
User1 Rect: Bottom	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5141" (switch20 ID#)
User2 Rect: Left	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5142" (switch21 ID#)
User2 Rect: Top	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5143" (switch22 ID#)
User2 Rect: Right	This field needs to contain the ID# of any switch that is supposed to raise or lower the pyramid or bridge. e.g. "5144" (switch23 ID#)
User2 Rect: Bottom	This field needs to contain the ID# of any switch that is supposed to raise or lower the

pyramid or bridge.
e.g. "5145" (switch24 ID#)

Object FlagsDialog Box

No Draw Checked

TileTriggerSwitch

The TileTriggerSwitch logic is used on top of switches that are supposed to toggle pyramids and bridges up and down.

Checkpoint Switchez - Operate checkpoint pyramidz. The checkpoint switch is unique in that it can require that a certain Grunt with a tool or toy operate them. This is specified in the "Smarts" field of the "TileTriggerSwitch" logic. Checkpoint pyramids should contain the switch ID #'s of the checkpoint switches that are supposed to toggle them. **Important!** Whenever you make a set of checkpoint switchez, you **MUST** put in an associated CheckpointTrigger logic. If you forget to put in a CheckpointTrigger logic, then the game will exit when you a Grunt steps on one of the Checkpoint Switchez.

Yellow Switchez - Operate yellow toggle arrowz. Yellow toggle arrowz should contain the switch ID of the yellow switch that is supposed to toggle them.

Blue Switchez - Operate bridgez. Bridgez (except "Crumble Briggez" that don't require logics or switches) should contain the switch ID of the blue switch that is supposed to toggle them.

Silver Timer Switchez - Operate silver pyramidz. Silver pyramidz should contain the switch ID of the timer switch that is supposed to toggle them. Also the pyramidz are supposed to specify the time that they remain toggled when the switch is operated. This is specified in the "Damage" field of the "TileTrigger" logic.

Black Once-Only Switchez - Operate black pyramidz. Black pyramidz should contain the switch ID # of the black switch that is supposed to toggle them.

Green Switchez - Operate green pyramidz. Green pyramidz should contain the switch ID of the green switch that is supposed to toggle them.

Red Switchez - Operates ALL red pyramidz in the level. **Important!** Only 1 red pyramid in the level should have a "TileTrigger" logic associated with it that contains the switch ID #'s of ALL the red switchez in the entire level.

Purple MultiGrunt Switchez - Operate purple pyramidz. Purple pyramidz should contain ALL the switch ID #'s of the Purple MultiGrunt switchez that are supposed to toggle them.

Orange Up-Down Switchez - Operate orange pyramidz. The orange switches are required to have all the other ID #'s of the orange switches in the group. Orange pyramidz should contain the switch ID of the orange switch in the group that is supposed to toggle them. **Important!** Also, 1 of the orange switchez needs to be in the down position (tile #240) initially.

Gold Secret Switchez - Operate TileSecretTrigger logics. The TileSecretTrigger logics will temporarily modify the tiles that the logics are located on.

Important! Whenever you place a TileTriggerSwitch, **MAKE SURE** to press the "Calc Cow ID" button for the "TileTriggerSwitch" logic before you copy the "TileTriggerSwitch" ID # into the "TileTrigger" Rects fields. Also, if you EVER move a switch, make sure to press the "Calc Cow ID" button AGAIN and re-copy the new "TileTriggerSwitch" ID # into the "TileTrigger" Rects field. Note: The "Calc Cow ID" button will re-calculate an ID# based on the switches position. If the switch is moved and the "Calc Cow ID" button is not pressed, the game will exit when you step on the switch.

Edit Objects Dialog Box

Logic: "TileTriggerSwitch"
Image Set: "GAME_WAPWORLDONLY_SWITCH"
Smarts: If the switch is a "CheckPointSwitch" then this field should contain the ID # for the required tool or toy for the "CheckPointSwitch". Note: the appropriate image of the tool or toy will be displayed on the switch in the game. See "Appendix B: Item ID Numbers" for a list of the ID numbers.
Speed X: This field contains the X tile coordinate of the switch.
AUTOMATICALLY generated with the "Calc Cow ID" button is pressed.
Speed Y: This field contains the Y tile coordinate of the switch.
AUTOMATICALLY generated with the "Calc Cow ID" button is pressed.
Calc Cow ID button PRESS THIS BUTTON EVERYTIME YOU PLACE/MOVE A SWITCH

Object RectanglesDialog Box

Move Rect: Left	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5122" (switch1 ID#)
Move Rect: Top	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5123" (switch2 ID#)
Move Rect: Right	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5124" (switch3 ID#)
Move Rect: Bottom	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5125" (switch4 ID#)
Hit Rect: Left	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5126" (switch5 ID#)
Hit Rect: Top	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5127" (switch6 ID#)
Hit Rect: Right	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5128" (switch7 ID#)
Hit Rect: Bottom	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5129" (switch8 ID#)
Attack Rect: Left	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5130" (switch9 ID#)
Attack Rect: Top	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5131" (switch10 ID#)
Attack Rect: Right	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5132" (switch11 ID#)
Attack Rect: Bottom	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5133" (switch12 ID#)
Clip Rect: Left	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5134" (switch13 ID#)
Clip Rect: Top	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5135" (switch14 ID#)
Clip Rect: Right	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5136" (switch15 ID#)
Clip Rect: Bottom	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5137" (switch16 ID#)
User1 Rect: Left	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5138" (switch17 ID#)
User1 Rect: Top	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5139" (switch18 ID#)
User1 Rect: Right	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5140" (switch19 ID#)
User1 Rect: Bottom	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5141" (switch20 ID#)

User2 Rect: Left	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5142" (switch21 ID#)
User2 Rect: Top	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5143" (switch22 ID#)
User2 Rect: Right	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5144" (switch23 ID#)
User2 Rect: Bottom	If the switch is an "Orange Up-Down Switch" then this field needs to contain the ID# of any of the other "Orange Up-Down Switches" in the group. e.g. "5145" (switch24 ID#)

Object FlagsDialog Box

No Draw Checked

Teleporterz

Listed below are all the game logics for placing teleporterz in the levelz.

SecretTeleporterTrigger

This is the logic that is used to trigger Secret Teleporters to become active/visible.

Edit Objects Dialog Box

Logic:	"SecretTeleporterTrigger"
Image Set:	"GAME_WAPWORLDONLY_TRIGGER"
Score:	X tile coordinate of the Secret Teleporter Entrance starting position. (Where the Secret Teleporter will open)
Points:	Y tile coordinate of the Secret Teleporter Entrance starting position. (Where the Secret Teleporter will open)
Powerup:	X tile coordinate of the Secret Teleporter Exit position. (Where the return teleporter will open)
Damage:	Y tile coordinate of the Secret Teleporter Exit position. (Where the return teleporter will open)
Speed X:	X tile coordinate of the Secret Teleporter Entrance destination position. (Where the secret Teleporter will take you)
Speed Y:	Y tile coordinate of the Secret Teleporter Entrance destination position. (Where the secret Teleporter will take you)
Speed:	This field determines the number of milliseconds that the initial Secret Teleporter stays active. e.g. "5000" (5 seconds)

Object FlagsDialog Box

No Draw Checked

Teleporter

The Teleporter logic is used to place teleporters in the level.

Edit Objects Dialog Box

Logic:	"Teleporter"
Image Set:	"GAME_WORMHOLE"
Smarts:	"0" - Teleporter can always be used. "1" - Teleporter will disappear after it is used.
Speed X:	X tile coordinate of the teleporter destination.
Speed Y:	Y tile coordinate of the teleporter destination.

Misc

Listed below are miscellaneous game logics used for "Guardpointz" and "Waypointz" in Battlez maps and setting time limitz in either Questz or Battlez maps.

GuardPoint (Battlez Only)

The GuardPoint logic is used to help the "Computer Controlled Opponents" (Battlez) to defend their fort when it is being attacked. When a "GuardPoint" is placed on a tile, the logic will detect if an enemy grunt walks within a 1 tile radius of the GuardPoint and will send nearby Gruntz to attack the enemy Grunt who "triggered" the GuardPoint. I typically used GuardPoints around the outer perimeter of brickz protecting each fort. **Important!** When placing GuardPoints within the protective perimeter of a fort, the computer controlled enemy who controls that fort will break it's own brickz if they are in the way of getting to the Grunt who triggered the GuardPoint. This may or may not be what you want. It's up to you. J

Edit Objects Dialog Box

Logic: "GuardPoint"
Image Set: GAME_WAPWORLDONLY_GUARDPOINT
Smarts: "0" - Player 1 guard point (Battlez only)
"1" - Player 2 guard point (Battlez only)
"2" - Player 3 guard point (Battlez only)
"3" - Player 4 guard point (Battlez only)

Object FlagsDialog Box

No Draw Checked

LevelTime

This is a special logic that can be used to assign a time limit to any Questz or Battlez map. It will cause a timer to display in the bottom right-hand corner of the map displaying the remaining time to finish the level or battle. In Questz mode, each time a Grunt steps on a CheckpointTriggerSwitch or picks up a Stopwatch (see appendix B), the player can get more time. **Note:** It does not matter where on the level you place this logic. If this logic is found anywhere on the level, then it will be timed. Otherwise the level will not be timed. (None of the levels in the retail or demo version of Gruntz are timed – this is a special extra feature for custom level builders)

Edit Objects Dialog Box

Logic: "LevelTime"
Image Set: "GAME_WAPWORLDONLY_TRIGGER"
Score: The value in this field represents the number of minutes that the player has to finish the level initially. (e.g. 30 would be 30 minutes)
Points: The value in this field represents the number of seconds that the player has to finish the level initially. (e.g. 45 would be 45 seconds) **Note:** These seconds are in addition to any minutes that were specified in the "Score" field.

Object FlagsDialog Box

No Draw Checked

WayPoint (Battlez Only)

The WayPoint logic is used to help the "Computer Controlled Opponents" (Battlez) to navigate towards an enemy fort. When "WayPoints" are placed near a fort, the enemy "Computer Controlled Opponents" will randomly pick a WayPoint to go and from there the Grunt will find the shortest path to the enemy's fortress.

Important! It is important to understand that "Computer Controlled Opponents" will not go from waypoint to waypoint. An enemy Grunt will only use ONLY one WayPoint and from there the Grunt will go directly to the enemy fort. The best method for placing WayPoints is to place them outside of the outer perimeter of brickz that are protecting the fortress. Also, you have to make sure you place WayPoints on tiles that are "walkable" and are not covered with another object. If "Computer Controlled Opponents" are not leaving their GruntCreationPoints, then it usually means that you have placed

some WayPoints in bad or inaccessible locations.

Edit Objects Dialog Box

Logic: "WayPoint"
Image Set: GAME_WAPWORLDONLY_WAYPOINT
Smarts: "0" - Player 1 waypoint (Battlez only)
"1" - Player 2 waypoint (Battlez only)
"2" - Player 3 waypoint (Battlez only)
"3" - Player 4 waypoint (Battlez only)

Object FlagsDialog Box

No Draw **Checked**

Troubleshooting

Important! When creating a custom level, you need to be EXTREMELY careful when placing logics. Because logics are case sensitive, they can cause the game to exit if you have incorrectly spelled a logic name (certain logics...e.g. Brickz). Also it is a good idea to ALWAYS test your level before you add too much stuff to it! My technique was to start with the absolute minimum (NO switches, pyramids, bridges, hazards) and make sure that the level loads. If the game exits, then you probably added something incorrectly. If you can figure out what is wrong then obviously you can fix it. But, sometimes you can't see what is wrong and you might end having to delete ALL objects and starting over again. Which is NOT good. So once you got the basics working, try adding a few switches and pyramids and then test! Then add some more stuff, then test! If you spend all day adding stuff and not test your level, you could end up with a problem and not know which one of the hundreds of logics you just added has the problem!

Problem: "Unable to locate REZ file." message when opening a custom level using "GruntzEdit"

Possible Solution:

- Make sure the Gruntz.rez file is installed on your hard drive. (Normal or Custom Installation)
- Select "Properties" from the "World" menu.
- Enter the path where your Gruntz.rez is located in the "Rez File" field.
(e.g. \GAMES\GRUNTZ\GRUNTZ.REZ)

Problem: Game exits (or crashes) when loading the custom level.

Possible Causes:

- A "Brickz" logic was misspelled.
- A "TileTriggerSwitch" logic was placed on a tile that is NOT a switch.
- A "TileTrigger" has an invalid switch ID# in the Rects field.

Problem: Game exits when you step on a switch.

Possible Causes:

- The switch does not have a "TileTriggerSwitch" on it.

Appendix A: Ambient Sounds

Listed below are the available ambient sounds per area (e.g. Area1 - Rocky Roadz, Area2 - Gruntziclez, etc). Note: the sounds listed per area are the only sounds available. In other words, if you are making a level in Rocky Roadz then you cannot use the sound "LEVEL_AMBIENT_AMBIENT2LOOP".

Area 1 - Rocky Roadz

File Name	Description
LEVEL_AMBIENT_AMBIENT1LOOP	Looping sound for background ambience
LEVEL_AMBIENT_WATER	Looping sound for water
LEVEL_AMBIENT_GOO	Looping sound for goo (tar) pits
LEVEL_AMBIENT_AMBIENT1A	Bird sounds (should be played randomly)
LEVEL_AMBIENT_AMBIENT1B	Bird sounds (should be played randomly)
LEVEL_AMBIENT_AMBIENT1C	Bird sounds (should be played randomly)

Area 2 - Gruntziclez

File Name	Description
LEVEL_AMBIENT_AMBIENT2LOOP	Looping sound for background ambience
LEVEL_AMBIENT_WATER	Looping sound for water
LEVEL_AMBIENT_GOO	Looping sound for goo (tar) pits

Area 3 - Trouble in the Tropicz

File Name	Description
LEVEL_AMBIENT_AMBIENT3LOOP	Looping sound for background ambience
LEVEL_AMBIENT_WATER	Looping sound for water
LEVEL_AMBIENT_LAVA	Looping sound for lava pits

Area 4 - High on Sweetz

File Name	Description
LEVEL_AMBIENT_AMBIENT4LOOP	Looping sound for background ambience
LEVEL_AMBIENT_WATER	Looping sound for water

Area 5 - High Rollerz

File Name	Description
LEVEL_AMBIENT_AMBIENT5LOOP	Looping sound for background ambience
LEVEL_AMBIENT_WATER	Looping sound for water

Area 6 - Honey, I Shrunk the Gruntz!

File Name	Description
LEVEL_AMBIENT_AMBIENT6LOOP	Looping sound for background ambience
LEVEL_AMBIENT_WATERDRIP	Looping sound for water faucet dripping
LEVEL_AMBIENT_EGG	Looping sound for frying egg.

Area 7 - Miniature Masterz

File Name	Description
LEVEL_AMBIENT_AMBIENT7LOOP	Looping sound for background ambience

LEVEL_AMBIENT_WATER
LEVEL_AMBIENT_SAND

Looping sound for water
Looping sound for sand pits

Area 8 - Gruntz in Space

File Name

LEVEL_AMBIENT_AMBIENT8LOOP
LEVEL_AMBIENT_WATER

Description

Looping sound for background ambience
Looping sound for water

Appendix B: Item ID Numbers

Below is a table that contains the ID#s for all the toolz, toyz, cursez and powerupz in the game. These ID#s are used in certain game logic fields. (e.g. "Powerup" field for "CoveredPowerups")

Item ID Numbers

TOOLZ	ID #
Bombz	1
Boomerangz	2
Brickz	3
Clubz	4
Gauntletz	5
Glovez	6
Gooberz	7
Gravity Bootz	8
Gun Hatz	9
Sponge Gunz	10
Rockz	11
Shieldz	12
Shovelz	13
Springz	14
Spy Gear	15
Swordz	16
Time-Bombz	17
Toobz	18
Magic Wandz	19
Warpstonez	20
Welder's Kitz	21
Wingz	22
TOYZ	ID #
Baby-Walkerz	23
Beach Ballz	24
Monster Wheelz	25
Go-Kartz	26
Jack-In-The-Boxez	27
Jumpropez	28
Pogo-Stickz	29
Scrollz	30
Squeak Toyz	31
Yo-yoz	32
Megaphonez	50
POWERUPZ	ID #
Can of Zap Cola	51
Bottle of Zap Cola	52
Keg of Zap Cola	53
Invisibility	54
Super Speed	55
Invulnerability	56
Conversion	57
Death Touch	58

Roidz	59
Reactive Armor	60
CURSEZ	ID #
Random Colorz	61
Screen Shake	62
Black Screen	63
Mini Cam	64
OTHER	ID #
Stopwatch	75
Coin	80
Toy Box	85
Secret W	90
Secret A	91
Secret R	92
Secret P	93
Hidden Time-Bomb	99
SPELLZ	ID #
Random Spell	0
Freeze	1
Health	2
Resurrection	3
Random Toyz	4
Teleport	5
Rolling Ballz	6

Appendix C: Enemy AI Types

The table below lists the different enemy AI (Artificial Intelligence) types.

Enemy AI Types

Enemy AI Type	Description	"Points" field value
Bomberz	Bomberz always have Bombz and will hang around a location until they sense one of your Gruntz. Once a Bomber senses one of your Gruntz, the Bomber will light his Bomb and make a kamikaze run (in a straight line) for your Grunt! To avoid a kamikaze Bomber, just move your Grunt out of the way once the Bomber starts running! The range that a Bomber can sense can vary.	7
Brick Layerz	Brick Layerz always have Brick Laying Toolz and will hang around a location building Brickz anywhere they can. Once a Brick Layer senses one of your Gruntz, the Brick Layer will attack him. Brick Layerz will always go back to building Brickz if they are not attacking. The range that a Brick Layer can sense can vary.	8
Defenderz	Defenderz will stand guard at a location until they sense one of your Gruntz nearby. Once a Defender senses one of your Gruntz, the Defender will attack him. Defenderz will always return to their post if they are not attacking. Defenderz can be found carrying any of the Toolz in the game, and the range that a Defender can sense can vary.	4
Diggerz	Diggerz always have Shovelz and will hang around a location digging all Moundz nearby. Once a Digger senses one of your Gruntz, the Digger will attack him. Diggerz will always go back to digging Moundz if they are not attacking. The range that a Digger can sense can vary.	11
Dumb Chaserz	Dumb Chaserz will hang around a location until they sense one of your Gruntz. Once a Dumb Chaser senses one of your Gruntz, The Dumb Chaser will attack him. If your Grunt runs away, the Dumb Chaser will chase after him. The only way to get a Dumb Chaser to stop chasing you is to kill him, or to get far enough away from him that he can no longer sense you. Dumb Chaserz can be found carrying any of the Toolz in the game, and the range that a Dumb Chaser can sense can vary.	1
Gauntletz Gruntz	Gauntletz Gruntz always have Gauntletz and will hang around a location until they sense one of your Gruntz. Once a Gauntletz Grunt senses one of your Gruntz, the Gauntletz Grunt will attack him and break through any breakable objectz that are in the way. The range that a Gauntletz Grunt can sense can vary.	9
Goo Suckerz	Goo Suckerz always have Goober Strawz and will hang around a location sucking all Goo Puddlez nearby. Once a Goo Sucker senses one of your Gruntz, the Goo Sucker will attack him. Goo Suckerz will always go back to sucking Goo Puddlez if they are not attacking. The range that a Goo Sucker can sense can vary.	10
Hit and Runnerz	Hit & Runnerz will hang around a location until they sense one of your Gruntz. Once a Hit & Runner senses one of your Gruntz, The Hit & Runner will attack him. Each time a Hit & Runner attacks, he will try to run away to a safe location until he can attack again. If your Grunt runs away, the Hit & Runner will chase after him. The only way to get a Hit & Runner to stop chasing you is to kill him, or to get far enough away from him that he can no longer sense you. Hit & Runnerz can be found carrying any of the Toolz in the game, and the range that a Hit & Runner can sense can vary.	3
Object Guardz	Object Guardz will pace around an object until they sense one of your Gruntz. Once an Object Guard senses one of your Gruntz, the Object Guard will attack him. Object Guardz will always return	6

	to their post if they are not attacking. Object Guardz can be found carrying any of the Toolz in the game, and the range that an Object Guard can sense can vary.	
Post Guardz	Post Guardz will stand guard at a location and will not move and will not attack unless they are attacked first. Post Guardz can be found carrying any of the Toolz in the game, and they will not move under any circumstancez. The only way to remove a Post Guard from his post is to kill him or to give him a mobile Toy.	5
Smart Chaserz	Smart Chaserz are exactly like Dumb Chaserz except for one major difference. Smart Chaserz will ignore Gruntz that have stronger Toolz than they do. Smart Chaserz will still attack if they are attacked, but Smart Chaserz are smart and will not chase Gruntz that have stronger Toolz than they do. Smart Chaserz can be found carrying any of the Toolz in the game, and the range that a Smart Chaser can sense can vary.	2
Time Bomberz	Timebomberz always have Timebombz and will hang around a location until they sense one of your Gruntz. Once a Timebomber senses one of your Gruntz, the Timebomber will attack him and use Timebombz to break through and breakable objectz that are in the way. The range that a Timebomber can sense can vary.	12
Tool Thievez	Tool Thievez never have any Toolz and will hang around a location until they sense one of your Gruntz with a Tool. Once a Tool Thief senses one of your Gruntz with a Tool, he will chase your Grunt and try to steal the Tool and attack with it. If a Tool Thief steals a Tool, the only way to get it back is to kill him. The range that a Toyer can sense can vary.	13
Toyerz	Toyerz will always have a Toy and will hang around a location until they sense one of your Gruntz. Once a Toyer senses one of your Gruntz, he will chase your Grunt and try to give him the Toy. Once a Toyer gives up his Toy, he will just stand around. The range that a Toy thief can sense can vary.	14

Appendix D: Grunt Voice List

Enemy AI Types

Group #	Voice #	Description
811	0	aw meeeean... I hate these parts!!
811	1	my brainz hurting already!!
811	2	pretty tired of all switches!!
811	3	what is the deal with arrows?
811	4	rocket scientist?
811	5	don't have to go through this, right???
811	6	skip this part?
811	7	please type in a cheat code??
811	8	million games.. puzzle game!!
811	9	this'll be tough!!
811	10	this looks challenging!!
811	11	hey.. puzzle got removed??
811	12	how am i supposed to get past this??
811	13	who thought this section up.
811	14	go figure, another puzzle
812	0	better be careful around here!
812	1	more dangerous then scissors!
812	2	reminds me of Claw!
812	3	dude.. death tilez!!
812	4	ahh.. cheat code? please?
812	5	fancy footwork!!
812	6	wow.. glad i got life insurance!!
812	7	man someone could get killed around here!!
812	8	ahhh.. man.. I don't like this!!
812	9	don't let me die.. please don't let me die!!
813	0	take perfect timing!
813	1	man.. have to time perfectly!!
813	2	whoa.. looks really hard!!
813	3	hah!.. get past this in my sleep!
813	4	up..down... no up.. this is tough!
813	5	really careful around here!
813	6	you do know the timing for this part?
813	7	hope you know what you're doing?!
813	8	aww man.. I hate these parts!!
813	9	if I live thru this, remind me to send hate mail...

