

FULL HALO WIKI TUTORIALS LIST

Revival by: Nukeist

Notes: some of the tutorials did contain pictures, and reviving them would be near impossible. This list includes only the tutorials shown on the halomaps tutorial page exactly as they would have appeared a day prior to halowiki closing(without pictures). Timo12's Effective devmode tutorial could not have been found (aaah how could I let this happen?!?). Other than that, all of the tutorials on halowiki that are on <http://ghost.halomaps.org/tutorials/> are here- in the same order. —Nuke out

Creating Basic Terrain

By SomeRhino

SomeRhino@renevo.com

Note: This tutorial was originally written for the W3D engine. I cut out everything that isn't relevant to the Halo engine and have added some that is. Please don't be confused if some of the screenshots don't match up exactly with yours.

Creating terrain seems like a cumbersome job if you envision aligning planes and boxes only to yield flat ground and blocky mountains, but if you know how to use Max properly, it can be done neatly and in a relatively short amount of time. Note however that this only covers the basics of creating the terrain meshes, nothing advanced. Some more advanced techniques will be available in another upcoming tutorial. I also am assuming you know how to navigate gMax, and how to do apply basic textures. Texturing tutorials are aplenty.

Start off by creating a plane. Convert it to an editable mesh (right click) and drag the edge vertices around using soft selection (found by expanding its tab under Editable Mesh. Round it out a bit to form the base ground. Select the outer edges of the mesh.

Scroll down to Extrude on the sidebar and set the value to 0.00000001 or similar to keep the extrusion from going anywhere. Hit the Extrude button, and drag the edges vertically. Hit it again, and repeat. Use the scale tool to make the edges curve outward. Do this one or two more times until the edges cannot be seen from the ground below.

If you don't have one already, create an opening by deleting some polygons (select them and hit delete on your keyboard.) Drag a few of the vertices to make an outer angle. Now select the edges on the opening and extrude some more to create an exit to the area.

Now create a larger plane, which you will make into the main area of the map. Attach the meshes together using editable mesh.

Shape the new plane into more terrain using soft selection on the edge vertices and extruding some more.

Now it's time to connect the two objects together with polygons. Select vertices that are relatively close to each other and hit the collapse button under editable mesh. The vertices will weld, causing the mesh to become a single element.

Hit the Face branch on the Editable Mesh tree and hit the Create button. Build face by clicking on three vertices consecutively in Counter-Clockwise order. Hitting them clockwise will build a backface.

Build more faces to completely connect the objects,

If you switch your view to smooth + highlights, you'll notice that your level will be very faceted and blocky looking. Remedy this by selecting all the faces and scrolling down to the Auto Smooth button. Set the value to 180 and click it.

Now, use soft selection to drag vertices smoothly on your mountains and ground to give the terrain a more realistic shape.

Aircraftkiller taught me an easy technique for creating ridges. Select a few of your ground polygons and extrude them up a couple times scaling them down a bit each time to give the ridge a tapered look.

Now, select all the polygons that make up the ridge and find the Tessellate button. Leave it as Edge and 25, and just hit the button. It will smooth out the ridge with more polygons. Again, you need to run the Auto Smooth on your terrain.

Now it's time to divide up your meshes. Select the mountain polygons. I selected mine by going to the front view and selecting the higher polygons. Clean up by holding down the Ctrl key on your keyboard and selecting/deselecting polygons by clicking on them. Hit the detach button on the sidebar, name it, and hit OK.

You should have some nicely modelled terrain when you are done. Your design will probably be more complex than the one used in this tutorial. When you have the terrain shaped as you want, it's time to move onto the next tutorial about setting up the materials and shaders. We can't do this until the SDK comes out, however. Expect the tutorial soon after the Editing Kit comes out.

-SomeRhino

SomeRhino@renevo.com

Creating Terrain Using Displace Modifier

REQUIREMENTS: 3D Max, Photoshop.

ACKNOWLEDGEMENTS: I originally learned this technique from a tutorial by Hailfire on [Halomods](#), so great praise & thanks go to him for sorting out my terrain modelling life, so to speak!!

DESCRIPTION:

Fed up with making terrain by manually shifting vertex after vertex? Using the Displace modifier in 3DMax, you can make highly detailed terrains in a short space of time that look more realistic, more 'organic' than the blocky rendition we all got the first time we attempted it in MAX.

Using this technique, you create a heightmap in a program such as Photoshop, then import it to Max to create displaced terrain according to the map. The Heightmap is just a

greyscaled bitmap image, where white will eventually translate to high ground and black to low ground.

STEP BY STEP INSTRUCTIONS:

1.) Open up MAX. Create a box in the top window, drag it out and give it a length/width of 10,000 each, and a height of 5,000. Right click on the box and select convert to editable mesh. Now, in the edit mesh modifier section, select face selection mode (the triangle - third one along). Make sure ignore backfacing is off, as this option disables the selection of faces that are facing away from the viewer. Drag the selection lasso around the whole box, so all faces are selected. Go down to surface properties in the edit mesh dialog, and hit flip normals. This makes the faces of the box, that were all originally facing outside, to face inside, so that they will be rendered once you are inside the level.

Note: If you want to add a face to a selection, press control while selecting. If you want to remove a face from selection, press alt while selecting. This is really useful if you want to select, say, all faces except the top & sides, just select the entire whole box (face selection mode), then hold alt and then drag the selection lasso over the top & sides.

2.) Deselect all the faces, and go into the top viewport. Make sure ignore backfacing is now ON, and select the two faces that appear to make up the large flat surface of the box. If you did it right, only the bottom two faces should be selected. You can check this by looking in the edit mesh dialog and see the number of faces selected, or by rotating the perspective view. Finally, with the two ground faces selected, in surface properties (edit mesh dialog), go down to tessellate. Enter a value of 0, and hit tessellate 5 times, until the number of faces (written just under the select vertex/edge/triangle/polygon buttons) is 4096. No more than this, or your map will chug along at 1 FPS :S

3.) Now the box is ready for adding the terrain. With the bottom 4096 faces selected, carefully deselect the row of faces at the very edge, by dragging the lasso across the outside row of faces with ALT pressed down. Now, in surface properties, go to material ID, and set a material ID of 2 to these inner faces. This will help with applying the modifier later. This is how it should look so far:

(Click on thumbnail to enlarge)

4.) Open up Photoshop, and create a new canvas with an arbitrary size, like 500x500. Make sure the canvas is filled in white. For this example, we are going to make a crater like environment, so go to the ellipse tool (hold down the mouse button over the rectangle tool, and go down to ellipse), make black your foreground colour, and drag out a large black ellipse on the page. Make it slightly elliptical, rather than plain circular. Once you've done that, go to filters->blur->gaussian blur, enter a value of 5 pixels, and hit ok. It should now look something like this:

5.) Right then, in photoshop, find the smudge tool (hold down mouse over blur tool and select smudge). Select a radius of 25, a strength of 18% and make sure u have a soft edged brush selected. With the smudge tool, run around the edge of the ellipse, making small inward strokes from white to black (dont go back and forth) at irregular intervals (about 1-2 cm spacing on the screen) around the crater. Once you've gone all the way around once, make the radius, 18, and the strength something like 30%. Repeat smudging the outside of the crater from outside to inside at irregular intervals. Finally, select a smaller brush, 12 or so, and a strength of 40%, smudge again, but this time make inside to outside strokes. Feel free to experiment with sizes of brush and strengths to get the desired effect. Once you're done, perform a gaussian blur (1 pixel radius) and save. What you are aiming for is something that looks like this:

6.) Now for the actual terrain that will make up your map. Going with the crater theme, I downloaded a topology (bump) map image of the moon, available from this website: [1](approve sites) (Click number to view). Now, using the select tool, pinch a 500x500 section out of that image. I chose the area roughly to the right of centre, as it has some nice crater-style detail to it. To select an exact sized area, in the toolbar at the top of photoshop, select 'fixed size' under style for the selection tool, and enter the size as 500x500. Now click and drag anywhere in the image to place your selection box.

7.) Copy that selected area, go back to your crater image, and select Layer->New Layer. Do this twice, so two new layers are created. In the layer window at the bottom right, click on layer 1, making that one active. Paste your copied moon image into that layer. Now, click on the 'eye' next to layer 1. This defines whether it is visible or not. Hide that layer, and select the background layer with your crater pic on it. Select all, copy it, select Layer 2, and hit paste. You should have copied the background layer into layer2, and have moon data in layer one now. Finally, make layer 2 invisible for the moment (click the eye icon next to the layer) and select the background. With black as your foreground colour, press alt-backspace to fill in the background black.

8.) Now comes the clever bit. In the layer editor window again, make the two layers making up your image visible again by clicking where the eye icon once was. Now, making sure Layer 1 is moon data and Layer 2 crater (you can drag and move layers in the layer editor window), right click the crater layer and go to blending options. Under opacity, hit 70% and click ok ET VOILA!! You have a fairly realistic, craters kind of image that with a bit of luck, should look something like this:

9.) Ok, now select your Layer 1, moon data, and go to image->adjustments->auto contrast. This should lighten up those craters a bit on the moon layer only (as this is the layer selected), making them stand out a bit more. Feel free to take a white paint brush with a diameter of 50, soft brush style, opacity 80% and flow 70% and whiten up the area outside the crater a little (make sure layer 1 - moon - is selected, as the crater outside is already white!!), as this is detail you are not really going to see in-game.

10.) In a similar style to the above point, use a white brush with Radius 20, Opacity 50% and Flow 70% and just draw circles around the edges of any large craters you have on your moon data. Make sure the moon layer (Layer 1) is selected for this, and not the crater layer. Switch the colour to black, keeping the tool constraints the same, and blacken out the inside of the craters. This will make them more flat in the centre with higher ridges around the outside when you import this map into max. Similarly, blacken out a small area at either side of the map for bases, and add any more terrain features you like, such as roads (use black), hills (use white) etc... Once you have added all your features, feel free to use the smudge tool and apply a 1 pixel gaussian blur if you want to make the transistions from higher ground to lower more subtle. With a little luck, you should have something like this:

If so, save your work as a tiff (keep layers, as you can come back to edit it later) and close photoshop.

11.) Go back into MAX now, and select the faces that make up your ground, except the outside row. The simple way of doing this, if you followed the instructions from earlier properly (:P) is to select the box object, go into edit mesh mode (face select), and scroll down to surface properties. Enter 2 in the select by ID box and those faces will be highlighted. If not, no matter, just re-select them using the instructions from above. Now, in the modifier list, with those faces still selected, add a displace modifier. Everything on the modifier stack is effected by whatever is below it. If you go back into edit mesh, select a couple of faces, then click on displace, your displace modifier will now be applied to the selected faces. So, in the displace dialog, select your map that you created and saved in photoshop (go to image->bitmap-> and click on the 'none' to add the bitmap). Once you ahve done that, go to strength at the top of the dialog, and enter 2000. This corresponds to the white height - ie: white in your image will now be 2000 units high, and black 0 units.

12.) BO!! The ground terrain should magically appear. If it didnt, go back and retrace your steps until it does!!! Here is an image of what it should look like. I have added some specular to make the detail of the terrain stand out:

Final Stuff

Nice, now that you have learnt how to do your own custom terrains, you should also know that with the displace modifier in its current state, editing the map in photoshop and reloading the bitmap for the modifier will update the terrain. So, you can get it as right as possible in photoshop, then continue your editing in MAX.

Make sure you use the reference models for the MC to gauge the scale of terrain features. Also, once you are satisfied with your terrain map, if you go into the modifier stack, and add another edit-mesh modifier, you can then edit individual vertices etc... using soft selection amongst other things.

Err, Also,

Once you've completed your terrain and want to add textures for the ground etc, you can use your displace map as a basis for creating base/alpha map for the level, as it is already

lined up to the terrain. Make sure you resize the map to 512x512, or another power of two before importing it into the HEK though!!

Good luck making terrains!!!!

EViLDaVE :@

WindInSapien

How to add wind to a level

This tutorial will explain how to add wind to a level using sapien.

First of all you must understand how to use all the windows in sapien, (The Game Window, The Hierarchy View, The Tool Window, The Properties Palette) Each of these windows has its use and I will explain it in this tutorial.

Step 1. Open sapien and navigate to a Scenario file that you want to put wind it. Let it open, it will take a minute. Once it is open arrange your windows how you like them just make sure you can get to all of them.

Step 2. Once sapien opens and you have arranged your windows follow these steps. First go to your Hierarchy view and open the **Mission** folder, then open the **Cluster Properties** folder. Once you have those folders open open the **Palettes** folder under that. Click the **Weather Palette** folder.

Step 3. Once you have navigated down to your **Weather Palette** folder click your Properties Palette to select it. Next to the top box it should say (name). Click that box and name it "wind". Then scroll down to the bottom of the Properties Palette and near the bottom there should be a box that has a gray background. It should say "wind" click the 3 dots on the right side and navigate to your halo ce root folder and then to levels\test\bloodgulch. Open the multiplayer.wind tag. Then under the wind direction it should have 3 boxes. The first one should have a "i" beside it, the 2nd one should have a "j" next to it also and the 3rd box should have a "k" next to it. In the first box type 3 instead of 0 and leave the other 2 boxes with a 0 in them.

Step 4. Next click the box under the 3 boxes that says "wind Magnitude". Change the 0 to a 2. 2 is the wind magnitude for all the other halo maps. Leave the "wind scale function box like it is. Once you have a 2 in the "wind magnitude box go up to the Hierarchy View window and click "New Instance" a new instance should appear in the right side of the Hierarchy View that is labeled "wind".

Step 5. Now close the "paletts" folder after adding the wind instance. Click the "Weather" folder under the "cluster properties" folder. Then click the Properties Palette and drop down the box and select "wind" from the list. Then go into your tool window and drop down that drop down box and select "wind" again. After selecting "wind" from the list press space to apply the cluster.

Now you have added wind to a level so that during CTF your flags dont look like poles.

NetGameFlags

Net Game Flags Placement

ACKNOWLEDGEMENTS: Originally Posted by Xzzy on [Gearbox Software Forums](#). Click [HERE](#) to view the original thread.

DESCRIPTION:

The objects that determine game rules and set objective points in the level are called Netgame Flags. These Netgame Flags are used to define the spot where a flag for CTF is located at, where the Oddballs in Oddball game modes spawn, the Race Points for Race Mode, the Race Vehicle spawn points, and the region or area for the control zone for King of the Hill. In addition, Netgame Flags are used to define the entry and exit points for teleporters.

For more information on the rules for Netgame Flags see the Multiplayer Level Design Technical Rules and Guidelines section under the General Level Design Information section located in the General Reference Sections.

The following section contains tutorials and examples that will demonstrate the placement of the Netgame Flags that are necessary for the proper operation of the level in a multiplayer game.

STEP BY STEP INSTRUCTIONS:

Netgame Flags Placement

Placement of the netgame flags necessary for the Capture the Flag (CTF) game mode in the tutorial level:

1) Go to the Hierarchy view window. In the Hierarchy tree expand the "Game data" branch. Next, select the "Netgame flags" branch/

2) The netgame flags that will be added are ctf - flag objects.

In the Properties palette make sure the the "type" is set to "ctf - flag".

In the Game Window, press the Right Mouse Button to place a netgame flag object in the level on top of each base.

Align the netgame flag so that it matches the position of the previously placed "flag_stand.scenery" object.

3) In the Properties palette set the "team index" for each netgame flag. This will determine the team setting for the flag.

The "team index" follows the "team index" conventions covered previously.

The "ctf - flag" netgame flag for the Blue Base should have its "team index" set to "1".

The "ctf - flag" netgame flag for the Red Base should have its "team index" set to "0".

The ctf - flag netgame flag represents the location where the flag will appear in the game as well as the area where the opposing teams flag must be taken to in order to score.

Adding Weather Attributes using Net Game Flags

The following section contains tutorials and examples that will demonstrate the application of "Weather" settings to a portal or cluster.

Setting "Weather" attributes in the .scenario_structure_bsp portals or clusters.

For the tutorial level, the weather effect that will be added is wind. Wind affects certain objects, such as the contrail on the sniper rifle bullet as it travels and the waving of the flag used in Capture the Flag.

1) Go to the Hierarchy view window. In the Hierarchy tree expand the "Cluster properties" branch. Next, expand the "Palettes" branch and select the "weather palette" subtree object.

2) Currently, the "weather palette" is empty. To create a "weather palette" follow the procedures below.

a) Click on New Instance button in the Hierarchy view window. A new instance of a weather palette will appear in the right pane of the Hierarchy view window.

b) Select this new instance.

c) This weather palette instance must have a weather (.wind) tag set for it.

Go to the Properties palette and click on the "..." button located in the "wind" section. An Open dialog window will appear.

Most multiplayer levels have their own multiplayer.wind tag file that exists in the root directory for the specific level.

For the tutorial, the multiplayer.wind file that will be used is an example .wind file that is included in the "tutorial_examples" level directory.

3) Go to the tutorial_examples level directory located under the "levels" tags directory.

4) Select the "multiplayer.wind" tag.

5) Click on the Open button.

This tag reference should appear in the "wind" section in the Properties palette window.

6) This weather reference must now be named.

In the "name" field in the Properties palette window enter the name "forest wind".

The name should update in the right pane of the Hierarchy view window.

The name does not matter, it is just used for reference. The weather can be changed underneath this name. This enables the generic "forest wind"

weather instance to be applied to the portals or clusters in the level and then the actual wind tag reference can be changed if needed to adjust or select the appropriate wind and weather effects.

7) The wind direction and magnitude must now be set. Follow the procedures below to enter these settings

a) In the "wind direction" settings enter a "3" in the "i" field. The i, j, and k fields represent a vector.

b) In the "wind magnitude" field enter a value of "2". This will be the strength of the wind in the direction indicated in the "wind direction" fields.

8) Notice that there is also a "particle system" section in the Properties palette window for the weather palette object.

If rain or some other weather effect was needed in the level, this is the field where a particle system and associated tags would be referenced.

The particle system would be affected by the wind values set.

The first image further explains the above procedures.

The weather will now be applied to all the portals or clusters in the level.

8) Go to the Hierarchy view window. In the Hierarchy tree expand the "Cluster properties" branch. Next, expand the "Palettes" branch and select the "weather" subtree object

The portals or clusters should now appear in the Game Window. The volume for a portal or cluster is outlined. The currently applied weather is color coded for each portal.

9) In the Tool Window, the "Render planes" and "Render non painted clusters" check boxes can be enabled or disabled to help view the portals better.

Move the camera in the Game Window if it is not already clearly in a portal or cluster.

10) The current weather that is applied to the portal that the camera is in is displayed in the "weather" pull down menu located in the Properties palette window. In this case, "NONE" appears since no "weather" have been applied.

To apply "weather" to the portal, use the weather pull down menu to select the "forest wind" weather system. The weather for the portal has now been set.

Move the camera to the next portal and repeat this procedure until all the portals for the level have the "forest wind" sound environment set.

FourSeatedHog

Ang88's 4 Seated Warthog

-First make a copy of the Warthog folder which is in the tags\vehicles\ folder

-Then rename the folder, and the mp_warthog.vehicle file to something different, this helps when you populate in sapien so that the vehicles will

have different names

-Now, open up Guerilla, and goto File-Open

- from here, open up the new vehicle.vehicle tag

-Scroll about 3/4 of the way down, until you see a section called seats

- this section has all the options for the different seats

-First select the gunner seat in the drop down menu

-Then push delete to delete it, because we won't need it anymore

-Now select the passenger, and push duplicate

-These are the options we want to change for the seat we just created

- Flags: Check Third Person Camera

- Label:scorpionRB

- This tells halo how MC sits in the seat, it is a name for the type of animation

- Marker Name:gunner right foot

-This is the location of the seat, you can find all the markers if you open the warthog's model and goto the markers section

-Camera Marker Name:gunner right foot

-This is the location for rotation of the camera

-Now duplicate this seat

-In the new seat, we just want to change 3 things

-Label:scorpionLB

-Marker Name:gunner left foot

-Camera Marker Name:gunner left foot

-Now save

-Next, we want to get rid of the gun turret

-To do this, we are going to make a shader that is totally clear

-To do this, make a bitmap with a black alpha channel

-Then make a shader_enviroment tag, that has the alpha-tested option check, and the bitmap you created as the bump map

-Now open up the vehicle tag again

-where it says model, click "." and make sure that the model is the one in the new vehicle folder

-after you picked the right model tag(tags\vehicles\new_warthog\warthog.gbxmodel) push open to see the properties for the model

-Now scroll all the way down to the bottom where it says shaders

-Using the drop down list, pick the shaders that apply to the Chaingun, and change the shader to the clear one you just made

-After doing this, push save

ChoelDepthWeaponTutPT1

Custom Weapon Tutorial By Choel & DepthCharge

Section 1 - Before the HEK

REQUIREMENTS:

3DMax, HEK, a 3DS Model of a Weapon to be Injected into Halo

ACKNOWLEDGEMENTS:

This tutorial has been kindly written & submitted by **Choel & Depthcharge**, after the production of their Unreal Tournament Weapons Mod for Halo. The UT Mod threads can be viewed at the following pages. Check out their work as it's impressive stuff!!

- [UT Mod Thread 1](#) (Gearbox)
 - [UT Mod Thread 2](#) (Gearbox)
 - [Energy Sword FP In Game Pics](#) (Halomods)
 - [UT Mod Progress Update](#) (Dead link)
-

DESCRIPTION

A step by step tutorial on how to inject custom weapons into the HEK. Submitted over several parts, this tutorial will cover ..., ... and ... and some other ... too (to be added :P).

This tutorial assumes you have a model of a weapon in 3DS or GMax ready to be injected. If you don't have a weapon, try using this freeware model of a Glock-17 from Turbosquid.

- [Turbosquid Freeware Glock \(3DS\)](#)
-

Custom Weapon Tutorial - Part 1-1: Setting up Your Model

'Note:' PLEASE MAKE SURE THAT YOU KNOW ALL THE TERMINOLOGY AND HAVE READ THE GEARBOX TUTORIAL ON MAKING MAPS!!

Alright, because I have no idea what your weapons look like, were just going to assume that you've made a basic gun, like this shotgun ...

(Click Thumbnail to Enlarge)

Step 1: Rotate your gun so the part where the bullet/projectile is supposed to come out is pointing the direction of "X" on the Compass. For the Shotgun, will need to rotate 90 degrees in Z and 90 degrees in X.

Step 2: We need to scale down (or Up) your model, for reference, here is a picture of my shotgun, in max, and then a pic in sapien.

Note: Use the Master Chief Reference Model as a guide when sizing

(Click Thumbnail to Enlarge)

(Click Thumbnail to Enlarge)

Step 4: Create a Box, thats 64x64x64, name it frame gun.

Step 5: At the front of your gun, at the end of the barrel, make a shape, any shape, and name it (without quotes) "#primary trigger"

Step 5.5: If you want your weapon to own with animations and stuff use the script that James wrote. It imports the .gbxmodel tag of your choice with nodes, markers, and goemetry. The weapon that you are going to base your animations on needs to have these markers. Open up your favorite .gbxmodel tag in guerilla and go file>export and put it somewhere, don't forget where it is. The script by default is in the "Halo Custom Edition" folder so exporting it to there isn't a bad idea. So get the script (<http://files.halomods.com/viewtopic.php?p=102646#102646>) and put in your scripts

directory (it might not be a bad idea to read through the readme file it comes with) and then go from the menu at the top of 3dsmax\gmax Maxscript>Run Script>pick the script called gbxmodelimporter. It asks you to open file and then you need to open up the .txt file of a .gbxmodel tag that you had guerilla export. It will import the stuff I said above. Move the markers to where you want them to go for your weapon. You can use the geometry it imports to sort of figure out where the marker was originally and what it probably does.

Step 6: Select all parts of the model that you want exported, click select and link, then click select by name, and then click frame gun, then Link.

***Note:** IF you think you want to do some more advanced crap, then you should know that you can have more than one frame, but they should all be linked to frame gun. Some other frames that are used are: frame magazine, frame display, frame wing and many others, you can use gurilla to find more.*

Your model is now ready for exporting, move on to part 2 (for TP and FP)...

Custom Weapon Tutorial - Part 1-2, Exporting your model

Ok, you got frame, lets get it in game, and don't fap at my rap.

Step 1. ok, click on File > export.

Step 2. Navigate to your HALO CE directory, and go into data.

Step 3. If you don't have one already, make a weapons directory inside of data

Step 4. in in your /data/weapons/ create a directory for your gun, in my case, its gonna be /q3shotgun/

Step 5. now in your directory for your weapon, /data/weapons/Disco_raper/, or whatever, make ANOTHER NEW FOLDER, call it models.

Step 6. /data/weapons/YOUR_WEAPON_NAME/models/ <-- thats where you should, and remember, don't have spaces in your directory name, tool.exe doesn't like it, type in a name for your model, open the file types and find JMS, NOT JMA! Hit enter. You should now have an exported jms file.

Step 7. Reexport the model, into /data/weapons/YOUR_WEAPON_NAME/FP/models/

Step 8. If you have a texture, make sure its dimensions are powers of 2 (IE 2x4, or 16x16, or 1024x256, or 512X16 ..) and move it to /data/weapons/YOUR_WEAPON_NAME/bitmaps/

Now, lets compile the model

Custom Weapon Tutorial - Part 1-3, THE DREADED TOOL.EXE

Too.exe is not a buggy program, if it was, its Error would almost always be "USER ERROR: PLEASE REPLACE USER". With that said, the tool errors you guys get are because of the CRAP that you shove into it.

Step 1. IF your have HEKter, or any other tool.exe front end, right click it, and hit delete.

Step 2. Make a new text file, and rename it "MAKE_YOURWEAPON.bat"

Step 3. Right click it, and hit edit.

Step 4. You should see a blank text document...

TYPE IN IT (without the quotes):

```
"Tool.exe model weapons\YOURWEAPON"  
"PAUSE"  
"TOOL.EXE model weapons\YOURWEAPONS\FP"  
"PAUSE"  
"TOOL.EXE bitmaps weapons\yourweapons\bitmaps"  
"PAUSE"
```

Save and Close it

Note: If your model has no open edges, misfliped normals, or things like that, you should also make a copy of your models folder and name it physics, then at the end of your .bat file, add:

```
"TOOL.EXE collision-geometry weapons\yourweapons"  
"PAUSE"
```

Step 5: Double click the .bat file you just created It should ask you for a shader type, I recommend #2

Note: SuperSunny has said that #1 is better because it adds more options, more options ... yes, needed options for a normal model, no.

Step 6: Close the .bat window, and then re-double click it, and hit enter every time it asks you to do so.

You should now have your gbxmodel tags and bitmap tags ... ON TO SECTION 2, YOUNG GRASSHOPPER!

ChoelDepthWeaponTutPT2

PAGE UNDER CONSTRUCTION

Custom Weapon Tutorial By [Choel](#) & DepthCharge

Section 2 - ".Weapon_is_fun"

Custom Weapon Tutorial - Part 2-1: "Guerilla is like HMT, only it works"

Step 1: Open guerilla, and decide what type of weapon you want

Step 2: Find the weapon tag that is most similar to what you want your weapon to be like

Step 2: Open it and go File > save as, Save it in /tags/weapons/YourWeaponfolder/ as Your_weapon_name.weapon

" Here is a detailed list (yea right) of all the things in the .weapon tag. IF I skip something, you shouldn't mess with it, or I know nothing about it. In either case, if you feel the need to edit this page and add onto our (DepthCharge and [Choel](#)) description, PLEASE DO!

- * lets start with the top of the file ... flags, you should usually have "brigher then it should be" checked. it makes things a little brigher then normal

- * Bounding radius: Usually 0.5 - 2.0 depending on the size of the gun. This is how far away from the 0x0y0z of the gun's model in order to picked it up. For something assault rifle-ish, use 0.6. Rocket launcher-ish, use 2.0

- * Don't use bounding offset or origin offset, if you a followed the directions in section 1, you wont need them.:in case you want to though, they just control how the center of the pickup area is postioned, so if you have a sword and you want the pickup center to be at the middle of the sword, then experiment with this 'til it works

- * I have no Idea what Accekeration scale is, but it seems to have something to do with SP or AI, don't mess with it. It should always be "2"

- * Model ... use the "... " button to navigate to your .gbxmodel tag that Tool made, Don't use the FP model here.

- * Animation graph, this is for having parts of the weapon move, navigate to your Model_animations tag, if you have one, one isn't needed

- * physics ... I'm stumped, just leave this blank: this applies mostly to vehicles i think, but i bet you could put tags here and give the weapon weight and behave like something that has mass

- * modifier shader ... same as the one above: this is where you put a shader that covers the object in that shader, so if you put the jackal shield shader over a weapon the weapon will be coated in a jackal shield thingy
- * Render bounding radius, unless you got some SERIOUSLY high poly stuff, leave this at "0"
- * Export to functions ... this is for adding things like the compass on the AR to your gun. When someone figures this out, this page will be updated.:here you can define what each functions does what when, try looking at existing tags to get a feel for what this does. You could make MC glow really bright when he has high health and then glow darker when he has low health by messing around with this.
- * Attachments ... this is where flashlights for the marine and muzzle flahses are added.: when you added in markers of the previous tutorial, you put the marker in here and put what you want to attach, if you want glowing red balls around where the projectiles cum out, then put primary trigger there and add a [Light Volume](#)
- * widgets, ever wanted an energy sword like glow on your weapon, this is where you do it:

DEPTHCHARGE NEEDS TO TYPE HERE: widgets are for adding things like light volumes, make a new widget tag and set in values and then link it here, your weapon will have a funky glow, there is also a glow tag which i am in the process of figuring out, i think it is what is needed to have an animated glow like the real energy sword

- * Functions, same with the export to functions
- * Change colors ... Well, my guess is that this can change the color of the model.
- * Premutations, Beats me ...: this is for damaged versions of the weapon, the ghost has some for blown up crap, I beleive you need to have the permutations in your gbxmodel before you can use this, but its for like a weapon turning into crap when it gets hosed
- * Predicted resource ... Well, judging by halos bad netcode, my guess is that this simply doesn't work
- * More flags well "Always maintains z up", this makes the weapon work like the flag, you know, never falling over. Its spooky, but hey ...
- * "Destroid by explosions", well, I think it might have something to do with barbies.:self explanatory, it means that if you blow up the weapon it turns into banannas i.e. RYOGAAAAAAAAA!!!!!!! LONG LIVE!!!!!!!
- * "Unaffected by gravity", This button makes your mom proud to be an american, and if she isn't an american, then you probably have a bad ping time when playing haloCE.
- * The Message-Index section changes what you see when you pick up the gun/sword/pole/your_mom: check the unicode string list in the ui directory i believe to see what is what message
- * sort order ... Heck if I know ...
- * scale, your overweight, and this proves it
- * A in, B in C in, and D in have something to do with functions ...:again these appear, i think it refers to another part of the file, same as above, but maybe it deals with the export to functions part of all tags below this such as projectiles
- * Detonation delay, for gernades or guns that blow up when dropped.
- * Detonating effect, what happens when your gernade/gun/Enegy Sword blows up (if dropped)

* Detonating effect, what happens when your grenade/gun/Energy Sword blows up (if dropped)... yes, there are 2, its not a typo:some gb dev probably was messing with the source and thought he would be cool if he duplicated this

* Look, its A in, B in C in, and D in again.:look what i said above, check existing tags to see what i mean

* "ready Time", This is how many seconds after switching to this weapon till it can be fired. If zero, you can fire the instant you hit tab (or if you remaped your keys, what ever you use to switch weapons).

* Ready effect, what happens when the gun is ready. Can be a effect or sound.

* Time to PLAY WITH THE HEAT!

Note: IF your gun doesn't worry about heat, "Heat recovery threshold", "Overheated Threshold", "heat detonation threshold", "heat detonation fraction", "heat loss per second", "overheated", and "detonation" should all be blank/0.

* "Heat recovery threshold" being that 0 = 100% and 1 = 0%, this is how much cooled down does the gun have to be start firing again

* "Overheated Threshold", your weapon is over heated when you this percent of heat, Should be higer then "Heat recovery threshold". (but lower then "heat detonation threshold", if its used):unless you like things exploding in your face(oh i know you do)

* "heat detonation threshold" , once you hit this, your gun might explode in your face

* "heat detonation fraction", chance of gun exploding each time its fired after "heat detonation threshold" is hit.

* "heat loss per second", self explanitory

* "overheated", is is the effect/sound that happens when your over heated, IE: blue stuff comming out of the vents.

* "detonation", this is your explosion effect:see needler to get good ideas of fun crap

* "Player melee damage", this is the damage that your gun does to people when you melee them.

* "Player melee responce", this is damage done to you when you melee (shouldn't be any but ...) can be used for making the screen shake and shuch ...

* "Actor firing paramaters", I know its used on vehicle weapons, but I don't know what it does.:i believe this is for ai, so they know what to do

* "near reticle range", beats me

* "far reticle range", beats me

* "intersection reticle range", beats me

* "Magnification levels", should be 0(no zoom) 1 (one zoom level, like a pistol) or 2 (like a sniper rifle):could be more

* "magnification range", If you have no zoom, both of the these boxes should be 0, one zoom, they should be the same number, 2 zooms, the first box is the first zoom level, and the second box is the second zoom level.

* Choel/Depthcharge will finsh the rest later.

ChoelDepthWeaponTutPT3

PAGE UNDER CONSTRUCTION

Custom Weapon Tutorial By [Choel](#) & DepthCharge

Light Volumes - "DRUGZ MAKE YOU FEEL FUNNY"

Light Volumes - they own you: "Insert generic quote here"

Intro to light volumes Light volumes are what makes cool glow effects like the glow of the needler, not the contrail (I will go into that later).

Open up guerilla and either open an existing light volume tag or create your own by going New>light volume.

Light Volume -attachment marker-this is where you specify what marker the light volume is created at

-flags-no clue, look into that later

Brightness Scale -near fade distance-light volume is fully bright up to this distance

-far fade distance-light volume doesnt get rendered

-perpendicular brightness scale-how bright it is when you look at it right on

-parallel brightness scale- how bright it is when you look at it in a parallel fashion

-brightness scale source-where it gets its paramaters on how to be bright, the functions are probably defined in the weapon tag or whatever calls this

Bitmaps -map-the bitmap that the light volume is drawn after, the default tag works fine for me, but you could make like a cool inverted one

-sequence index-im guessing if there are multiple light volumes, the priority? maybe it deals with individual light volumes

-count- the number of bitmaps drawn, if you have a long light volume you may need to increase this, i had to increase this to 50 for the lightsaber to make it not look like bubbles, but a continous stream

Frame Animation -frame animation source-read the tip under the title in the tag, mess with functions

Frames-you use the multiple frames for changing effects -offset from marker-how far off it is place from where the marker is

-offset exponent-controls how the light is compressed like the tool tip says

-length-how long the light volume is from the first bitmap to the last

-radius hither-how big the bitmap is from the center to the outside near the marker (if you don't know what a radius is)(hither means near btw)

-radius yon-same as above, except at the end of the light volume, away from the marker, you can change these two to make it get fatter as it goes, like when ryoga goes on a walk(yon as in yonder as in far away)

-radius exponent-controls how your light volume is shaped, you can make it look fat like a pill or "tear drop shaped" to quote the tool tip

-tint color hither-what color it is at the marker(alpha is how bright it is)

-tint color yon-what color it is at the end(you have to type in the alpha manually)

-tint color exponent-how the color changes from one to the other im guessing, so if thats the case then if its high then it is exponentially one of the colors more(pure speculation havent tested)

-brightness exponent-0 defaults to 1, JK, its how white it is in the middle, so if its high, it gets really white in the middle and towards the outside and then the edge is the normal color

FPAnimations

Fp Animation Guide

By **An Emu** with Snippets Added by **BioCroc** noted by °

Note:

- .jmo = Overlay
- .jma = Base
- .jmr = Replacement

- .Jmm = Movement^o

Necessary Animations marked by a †

First-person Fire-1 †(may be substituted with fire)

- Animation Type: .jma
- Frames: -
- Description: Primary trigger firing animation

First-person Fire-2

- Animation Type: .jma
- Frames: -
- Description: secondary trigger firing animation

First-person fire † (may be substituted with fire-1)

- Animation Type: .jma
- Frames: -
- Description: firing animation (used in guns that have no secondary trigger, but fire-1 can also be used)

First-person Ready †

- Animation Type: .jma
- Frames: -
- Description: Shown when you bring out a weapon

First-person throw-grenade †

- Animation Type: .jma
 - Frames: -
 - Description: When MC throws a grenade
-

First-person throw-overheated

- Animation Type: .jma
 - Frames: -
 - Description: When weapon is overheated and MC throws a grenade
-

First-person idle †

- Animation Type: .jma
 - Frames: -
 - Description: default animation
-

First-person melee †

- Animation Type: .jma
 - Frames: -
 - Description: Shown when mc melees
-

First-person reload-empty † (can be substituted with reload)

- Animation Type: .jma
 - Frames: -
 - Description: When clip is empty and you reload
-

First-person reload-full † (can be substituted with reload)

- Animation Type: .jma
 - Frames: -
 - Description: When clip still has ammo left and you reload
-

First-person reload † (can be substituted with reload-empty and reload-full)

- Animation Type: .jma
- Frames: -

- Description: Normal reloading animation (used in guns that have same animation for reloading no matter how much ammo is left)
-

First-person Moving

- Animation Type:.jmo
- Frames: -
- Description: When Walking, the sway of the gun

<http://img146.imageshack.us/img146/9142/smgmoving2sb.gif>([approve sites](#)) - Picture by [ZMT] L33T

First-person Overlays

- Animation Type: .jmo
- Frames: 9
- Description: When Looking, the sway of the gun
- keyframe 0-Default (used when not moving)
- keyframe 1-Moving forward
- keyframe 2-Moving backward
- keyframe 3-Strafing left
- keyframe 4-Strafing right
- keyframe 5-Turning left
- keyframe 6-Turning right
- keyframe 7-Looking up
- keyframe 8-Looking down
- keyframe 9-full automatic fire

ex: Make keyframe 5 have your gun pointed slightly to the left so when you turn left your gun leads.

<http://img71.imageshack.us/img71/3927/smgoverlays6lk.gif>([approve sites](#)) - Picture by [ZMT] L33T

First-person Posing

- Animation Type: .jma
- Frames: -
- Description: Shown when idle for too long (mc checks out his gun)

First-person misfire-1

- Animation Type: .jma
- Frames: -
- Description: misfire animation for guns like the plasma rifle

First-person o-h-exit

- Animation Type: .jma
- Frames: -
- Description: Shown when gun leaves overheated animation and resumes normal animations(?)

First-person o-h-s-enter

- Animation Type: .jma
- Frames: -
- Description: (?)

First-person overheated

- Animation Type: .jma
- Frames: -
- Description: when weapon is overheated (?)

First-person overheating

- Animation Type: .jma
- Frames: -
- Description: The animation that is played when the gun starts overheating

First-person overheating-again

- Animation Type: .jma
- Frames: -

- Description: Animation that is played when switching to a gun that is overheating.
-

First-person overcharged

- Animation Type: .jma
 - Frames: -
 - Description: when weapon is charged (like when plasma pistol turns sideways) (?)
-

First-person overcharged-jitter

- Animation Type: .jmo
 - Frames: 16
 - Description: the jitter when your weapon is overcharged (like the shaking in plasma pistol) (?)
-

First-person enter

- Animation Type: .jma
 - Frames: -
 - Description: in weapons like shotgun, where each round is loaded individually, when mc begins to put shell into gun. Stops right when the shell is actually put into the gun.
-

First-person exit-empty

- Animation Type: .jma
 - Frames: -
 - Description: in weapons like shotgun, where each round is loaded individually. Shown when clip is empty and you reload. Begins after mc loads the last round into the gun. (?) (in the shotgun, this animation is where he cocks his gun after he finishes reloading all 12 rounds)
-

First-person exit-full

- Animation Type: .jma
- Frames: -

- Description: in weapons like shotgun, where each round is loaded individually. Shown when you reload when the clip still has rounds in it. Begins after mc loads the last round into the gun (?)
-

First-person ammunition

- Animation Type: .jmo
 - Frames: (Magazine Capacity +1)
 - Description: This overlay animation changes the gun's appearance depending on how much ammunition is remaining in the weapon. For example, the needler uses this animation.
 - Frame 0 - Default
 - Frame 1 - Empty
 - Frame 2 - One Round Left
 - Frame 3 - Two Rounds Left
 - Frame 4 - Three Rounds Left
 - Etc...
-

How to tell what Your Animation will look like In-Game

-According to Biocroc

- Place a camera at 0x,0y,0z and set the FOV to 70

That is what it will look like ingame.

:: How to Set up the Camera ::

- click the camera button
 - set the FOV to 70
 - click at 0x,0y,0z and move to position it just right
 - In the viewport you would like to have the camera view from, push the "c" key it should change it to Camera.01 view
-

SimpleWeaponTutorialStepbyStep

Simple Weapon Tutorial - Step by step

Author: ZiON

Acknowledgements: James @ www.halomods.com

Ok, so we wanna do a very simple weapon, an additional pistol. This is how:

Since we wanna do a pistol, we are gonna use the pistols animations and such.

Part 1 - Importing the existing pistol's markers and geometry

1. Create the following directory structure:
 1. \HaloCE\data\weapons\simplegun
 2. \HaloCE\data\weapons\simplegun\bitmaps
 3. \HaloCE\data\weapons\simplegun\physics
 4. \HaloCE\data\weapons\simplegun\shaders
 5. \HaloCE\data\weapons\simplegun\models
 6. \HaloCE\data\weapons\simplegun\fp\models
 7. \HaloCE\data\weapons\simplegun\Max
2. Open Guerilla
3. Open the tag **\HaloCE\tags\weapons\plasma pistol\plasma pistol.gbxmodel**
4. In Guerilla's menu, select *File - Export* and save the file to **\HaloCE\data\Max\plasma pistol.txt**
5. Download James model importer, install it using the instructions that comes with the script.
6. In 3D studio Max, choose *MAXScript - Run script* and select the **gbxmodelimporterX_X.ms'** script.
7. Click Open file, navigate to **\HaloCE\data\Weapons\simplegun\Max** and open the file **plasma pistol.txt**.
8. After some processing, if it all turns out good, you should have the Halo pistol imported properly into 3dsmax. There should be a few boxes and spheres there aswell. Since we aint really gonna use the pistol itself, but only the markers we will keep it all in there and go to the menu and save the file to **\HaloCE\Data\Weapons\simplegun\Max** as **simplegun.max**.

Part2 - Creating our own "gun"

1. We are gonna use the original gun as a rawmodel, but to do that we need to hide all the markers. Click the "*Select by Name*" icon in 3DS Max. Select all names that starts with a # and all that starts with Frame, then click Ok.
2. in any of the viewports, right-click and select *Hide Selection*.
3. Create a box roughly surrounding the upperpart of the plasma pistol. Name it **UpperPart**.
4. Create another box roughly surrounding the handle plasma pistol. Name it **LowerPart**.
5. Remove the old Plasma gun, all of it, but leave the markers and frame objects intact.
6. Texture the gun as you like and put all bitmaps in the **\HaloCE\data\weapons\simplegun\bitmaps** folder.
7. Remember to link all parts of your gun to the "**frame gun**" object.

Part3 - Exporting the gun

1. Right-click in any viewport, select *Unhide all*
2. Go to *File - Export*, Select *Halo Model Exporter* as the filetype.

3. Export the model to:
 1. `\HaloCE\data\weapons\simplegun\models\`
 2. `\HaloCE\data\weapons\simplegun\fp\models\`
4. Save the gun to **simplegun.max**.
5. Go to the material editor, create a new material which you name "collision-geometry" (without quotes), assign it to the whole gun (all parts).
6. Export the gun to `\HaloCE\data\weapons\simplegun\physics\`
7. Save the gun to **simplegun-collision.max**.

Part 4 - Tool

Ok, we got this far. It's this part that might not go as good as it did to this point :P

1. Bring up a command prompt.
2. Change directory to your halo folder. (`cd "\program files\Microsoft Games\HaloCE\"`).
3. At the command prompt write:
 1. **tool model weapons\simplegun**
 2. **tool model weapons\simplegun\fp**
 3. **tool bitmaps weapons\simplegun\bitmaps**
 4. **tool collision-geometry weapons\simplegun**
4. If asks for shader type, select 2 and repeat the command once more.
5. If something failed, go check that you have linked it to the frame etc. If the Collision-geometry failed (which it most likely did) either go try to fix it in 3DS Max or simple to into max, create one box, surrounding the whole weapon, put the collision-geometry "material" on it and export it to the Physics folder. It's a gun we are making and not a vehicle =)

Models:

No response at all indicates it couldn't find the jms file. Check the path and that the file really exist.

```
### simplegun.JMS
```

Only this and nothing else indicates that the gun wasn't linked to the "frame gun". I.e. no model to convert.

```
### simplegun.JMS
```

detail level	worst-case vertices	worst-case triangles
-----	-----	-----
superlow	87	75
low	87	75
medium	87	75
high	87	75
superhigh	87	75

This is a success.

Part 5 - Guerilla

There.. We're almost done actually. Just a few things to do in Guerilla. We are gonna be lazy, but remember... Lazy is good! =)

1. Start Guerilla
2. Open up the shader tags and set the basemap, exactly as in when doing it on a map.
3. Open up the Plasma Pistols weapon tag. (**\HaloCE\tags\weapons\plasma pistol\plasma pistol.weapon**)
4. In the menu, click *File - Save as...*
5. Navigate to **\HaloCE\tags\weapons\simplegun** and save the weapon tag as **simplegun.weapon**, but don't close the tag yet.
6. At the top of the tag in Guerilla, there are a few things we want to change:
 1. *Model* - Change this to **"weapons\simplegun\simplegun"**
 2. *Collision model* - Change this to **"weapons\simplegun\simplegun"**
7. Scroll down to *"Interface"* (a bit lower than middle in the tag).
 1. *First person model* - Change this to **"weapons\simplegun\fp\fp"**
8. Save the tag in the menu - Save
9. In the menu, click *File - New*
10. In the "New tag" window that pops up, select *"Item_collection"*
 1. *Item* - Change to **"Weapon"** and enter **"weapons\simplegun\simplegun"** in the field to the right (or browse to the tag)
11. In the menu, select *File - Save as...*
12. Save it in **"HaloCE\tags\item collections\single weapons"** as **simplegun.item_collection**

That's it!

Just add it in Sapien as all the other guns. Enjoy!

Footnote: This tutorial is more or less the most limited gun you can do. Now it's up 2 you to experiment with all then endless options in Guerilla. Have fun!

Download the James Model Importer v5.4 here:

http://wiki.fpsgamers.se/uploads/Halo/jmi-version5_4.rar

User comments:

CustomSoundsTutorial

SuperSunny's Tutorial on how to add custom sounds

Before using OGG files, read [this](#).

Okay, first of all, download GoldWave (do a google on it). It's an AWESOME program that allows so many different types of exports and sound editing features. You need this for easier work.

Now, Go find any sound or music you like that is not a midi file. Copy this to your C:/Program Files/Microsoft Games/Halo Custom Edition/data/sound/test/test folder (C:/ is representing your drive, and you must create the test/test folders)...

Now, open the copied file in GoldWave. Go to file, save as, change the file type to a wav, and change the file type for the wav into a PCM 16 bit stereo file (you can do Mono, but it usually sounds like crap). Save it. Then, in tool.exe, type in:

(tool.exe sounds sound\test\test platform wav)

tool.exe sounds sound\test\test wav 0 <--- Won't work within Halo of some reason

tool.exe sounds sound\test\test xbox 1

It should say importing "sound name". If not, you messed up somewhere along the directions.

Then, you can take create new .sound_looping files in Geurilla or change existing ones to play your custom sound. In order to confirm your song converted correctly, open Geurilla, go to file, open, find your converted .sound file, open it, scroll all the way done, and select any of the three buttons that say play.

I've already been able to add some of my songs, the Forest Temple Music from Zelda, and more all into Halo. This can be used for custom sound effects also.

How To Make Your Custom Sounds Play When You Fire Your New Gun by: CptLag

Ok guys, I've been asked numerous times by friends and halo fans alike about how to add custom sounds to your gun. So I'll take you through how to make your gun use the new sound. First of all, you need to make your .sound tag. In order to do this (I'm sure you all have, but just to reiterate {sp?}) check out SuperSunny's tut above this little update. Now that you've made your .sound tag it's time to begin to add it to your .weapon tag. Now I'm going to assume you've edited an existing weapon tag and resaved it as your own. Open the .weapon of your gun (lets call it the Magnum for easy typing purposes). Scroll down to the very bottom of the Magnum tag and you should see a block named "Firing Effects". It'll look like the picture below:

<http://cptlag.haloimpulse.com/customsnd1.JPG>

Now that your at the Firing Effect block it's time to create your Magnum firing effect. Choose the open button on the right of the "..." button in the firing effect line (see above picture if you do not have the tag in front of you). Once you have opened the firing effect tag choose File > Save As and browse to the Magnum directory (normally tags\weapons\magnum). Once in there, create a new folder named "effects" so you now have "tags\weapons\magnum\effects". Save your Firing Effect into there (I suggest keeping the name so you won't confuse yourself). Alright, now you have a Magnum firing effect tag. In the new tag, scoll down to either the 3rd or 4th block titled "Parts". It should say "Firing" in the dropdown menu. Now where it says "type" make sure you have "Sound" selected in the dropdown menu. Then choose the "..." button and point to your custom sound (indicated below).

<http://cptlag.haloimpulse.com/customsnd2.JPG>

Now return to your .weapon tag and change where the firing tag points to point to the one you just created. Save all tags and close. Then build ur map cache file and your all set!

EDIT: sry guys but you hafta copy/paste the pix into your browser since idk the Wiki password to allow me to add them.

EDIT: The URL is approved now, np ;) /Z

Note: You need the Xbox Audio Codec to get this to run. Get it here:

http://wiki.fpsgamers.se/uploads/Halo/xb_adpcm_codec.rar

~SuperSunny and CptLag

WeaponCustomStrings

Weapons Strings

Author: SteelixB

To create custom strings for your weapon, first of all read my tutorial [Here](#) on creating unicode string lists with tool

Adding your string to the string list

First re-create the hud_icon_messages.unicode_string_list (located inside the tags\ui\hud folder), which includes your custom strings for your weapon at the end of the list. DON'T overwrite old strings, otherwise you could have problems when tool builds your final map.

Also note I say strings. The strings should be *You picked up a <weapon name>*, and something like *You picked up %d rounds for <weapon name>*, replacing *<weapon name>* with the name of your weapon. %d is a variable that is replaced in-game to the number of round you picked up.

Changing the string of the weapon you picked up

1. Once your unicode string list has been created, and overwrites hud_icon_messages.unicode_string_list, open Guerilla and open your weapons 'weapon' tag.
2. Scroll down the tag until you find *Predicted Resource*. Below this you will find *message-index*
3. In this section change the value *message index* to the index of the string *You picked up a <weapon name>*. If you don't know the index, open the string list tag into guerilla and have a look at the number on the left of your string. That's the

- index you use. You don't define the index of the picking up ammo string, this is automatically selected when you define the index for picking up the weapon. However, for this to work properly the format of the string list must be *You picked up a <weapon name>* then straight after, the next string must be *You picked up %d rounds for <weapon name>*.
4. Finally save the weapon tag.

Now when you create your map file your weapon will have your custom string.

-SteelixB and Kiwibird

CustomSoundsTutorial

SuperSunny's Tutorial on how to add custom sounds

Before using OGG files, read [this](#).

Okay, first of all, download GoldWave (do a google on it). It's an AWESOME program that allows so many different types of exports and sound editing features. You need this for easier work.

Now, Go find any sound or music you like that is not a midi file. Copy this to your C:/Program Files/Microsoft Games/Halo Custom Edition/data/sound/test/test folder (C:/ is representing your drive, and you must create the test/test folders)...

Now, open the copied file in GoldWave. Go to file, save as, change the file type to a wav, and change the file type for the wav into a PCM 16 bit stereo file (you can do Mono, but it usually sounds like crap). Save it. Then, in tool.exe, type in:

(tool.exe sounds sound\test\test platform wav)

tool.exe sounds sound\test\test wav 0 <--- Won't work within Halo of some reason

tool.exe sounds sound\test\test xbox 1

It should say importing "sound name". If not, you messed up somewhere along the directions.

Then, you can take create new .sound_looping files in Geurilla or change existing ones to play your custom sound. In order to confirm your song converted correctly, open Geurilla, go to file, open, find your converted .sound file, open it, scroll all the way done, and select any of the three buttons that say play.

I've already been able to add some of my songs, the Forest Temple Music from Zelda, and more all into Halo. This can be used for custom sound effects also.

How To Make Your Custom Sounds Play When You Fire Your New Gun by: CptLag

Ok guys, I've been asked numerous times by friends and halo fans alike about how to add custom sounds to your gun. So I'll take you through how to make your gun use the new sound. First of all, you need to make your .sound tag. In order to do this (I'm sure you all have, but just to reiterate {sp?}) check out SuperSunny's tut above this little update. Now that you've made your .sound tag it's time to begin to add it to your .weapon tag. Now I'm going to assume you've edited an existing weapon tag and resaved it as your own. Open the .weapon of your gun (lets call it the Magnum for easy typing purposes). Scroll down to the very bottom of the Magnum tag and you should see a block named "Firing Effects". It'll look like the picture below:

<http://cptlag.haloimpulse.com/customsnd1.JPG>

Now that your at the Firing Effect block it's time to create your Magnum firing effect. Choose the open button on the right of the "... " button in the firing effect line (see above picture if you do not have the tag in front of you). Once you have opened the firing effect tag choose File > Save As and browse to the Magnum directory (normally tags\weapons\magnum). Once in there, create a new folder named "effects" so you now have "tags\weapons\magnum\effects". Save your Firing Effect into there (I suggest keeping the name so you won't confuse yourself). Alright, now you have a Magnum firing effect tag. In the new tag, scoll down to either the 3rd or 4th block titled "Parts". It should say "Firing" in the dropdown menu. Now where it says "type" make sure you have "Sound" selected in the dropdown menu. Then choose the "... " button and point to your custom sound (indicated below).

<http://cptlag.haloimpulse.com/customsnd2.JPG>

Now return to your .weapon tag and change where the firing tag points to point to the one you just created. Save all tags and close. Then build ur map cache file and your all set!

EDIT: sry guys but you hafta copy/paste the pix into your browser since idk the Wiki password to allow me to add them.

EDIT: The URL is approved now, np ;) /Z

Note: You need the Xbox Audio Codec to get this to run. Get it here:

http://wiki.fpsgamers.se/uploads/Halo/xb_adpcm_codec.rar

MoonbaseSkyTut

REQUIREMENTS: 3D Max, Photoshop, HEK

ACKNOWLEDGEMENTS:

I originally learned how to edit certain parameters of existing multiplayer skies using Mattyp001's [Custom Sky Tutorial](#) available from [Halomods](#). Also, the Photo-Realistic Earth model, renderable in 3D Max was created through the use of a 3DS Tutorial from [Noir Extreme](#).

DESCRIPTION:

I'm currently working on a [Halo CE](#) map entitled *Sea of Tranquility* - a purpose built training arena on the Moon - that requires a custom sky, altogether different to those provided with the HEK. This tutorial covers step-by-step, the process of re-creating this sky, by modifying a copy of an existing multiplayer sky & inserting custom images where appropriate. The end result is a sky tag that can be used in any level, however the process of sky modification is probably more interesting.

The Finished Sky:

(Click on thumbnail to enlarge)

STEP BY STEP INSTRUCTIONS:

Note: This sky is created by modifying copies of the original Halo skies. This is not very flexible, but it is still possible to get a great result. If you want to create your own Gearbox Model & therefore a truly custom sky, go [HERE](#) for the sky Gearbox Model tutorial.

1.) Open up your Halo CE directory and Navigate to data/sky. Add a folder called moonbase. Within that, add a folder called bitmaps. Your directory structure should then be:

...//Halo Custom Edition/data/sky/moonbase/bitmaps

2.) Now navigate to ...//Halo Custom Edition/tags/sky/. The sky we will be bootlegging for our custom sky is dusk (not space! as space has fairly rubbish uvw mapping when you look straight up & also texture quality is poorer for some reason). Copy the entire folder dusk and paste it, but renaming it after paste from 'copy of dusk' to moonbase. In tags, the directory structure should now look like this:

...//Halo Custom Edition/tags/sky/moonbase

3.) Within tags/sky/moonbase, there will be the following files that must be renamed accordingly:

- dusk.gbxmodel (file)

rename to moonbase.gbxmodel

- dusk.sky (file)

rename to moonbase.sky

- sun.lensflare (file)

delete this. Copy in instead ...//Halo CE/tags/sky/space/sun in space.lens_flare

- bitmaps (directory):

this is where your custom bitmaps will go once compiled. Leave this atm.

- shaders (directory):

rename all the shaders to be _moonbase, so basis becomes basis_moonbase.shader_transparent_chicago etc...

The reason for this renaming is just so you don't accidentally modify and mess up your dusk sky. All these files link to each other in Guerilla, so its a good idea to rename them just so you don't link to, and permanently change the wrong ones!!

4.) Open Guerilla, navigate to ...//Halo CE/tags/sky/moonbase & open up moonbase.sky. This is what you should see:

(Click on thumbnail to enlarge)

Notice that it's still pointing to the sky/dusk/dusk.gbxmodel. This is not what we want, so first off, change the model to sky/dusk/moonbase.gbxmodel.

Note: In my screens it says moonbasetut.sky - don't worry about this, I already have a moonbase.sky, so I named mine moonbase tut just so there wouldn't be a conflict. All my files are named with tut, but yours should be named just moonbase.

This is what you should be aiming for:

(Click on thumbnail to enlarge)

While you're there, scroll down and add the sun in space lensflare, and change the colour of the emitted light from the sun, angle and strength of the light:

(Click on thumbnail to enlarge)

5.) Hit file->save all to save your sky and open up Photoshop. First, open up a new file, give it a size of 32x32, make it RGB colour and hit ok. Go to the colour palette and select black as your foreground colour. With the image selected, press alt-backspace to fill in the image black. Now, go to the layers window and select the channels tab. We want to create a new channel by hitting the icon in the bottom right of the channels palette. This should create an alpha channel which will be filled in black. If it's not filled in black, fill it in, and save your image in ...//Halo CE/data/sky/bitmaps as nothing.tif (Ensure compression is OFF, and that you saved the alpha channel).

6.) If you want, you can open up Max, and play around with the photoreal earth file available from [Noir Extreme](#). I downloaded this file and played around with it a bit until I got a nice shot of the earth on a black background with the light coming from the left side. I chose Europe/Africa as the centre of this shot as its a nicely detailed piece of land with plenty of interesting features to look at. Also, I deleted the 'earth sky at night' texture from the darkened side of the earth, as this gave the impression that the side of the world in shadow was not completely dark.

Alternatively, if you can't be bothered to do all that, download the image I used!!

[\(Download Earth1024 in .tif Format\)](#)

Get the tif image into photoshop and check that it has both colour and alpha components like the images above. Alternatively, if you want to generate your own image by rendering the max file, make sure the alpha channel is white for the portions of the earth you want to be visible (opaque) and black for the portions that are transparent. I also used a slight gaussian blur on the alpha channel, to give the effect that the thin atmosphere around the earth was semi-transparent.

Once your tif is complete, you have either downloaded or created your own, place it in ...//Halo CE/data/sky/moonbase/bitmaps

Note: If you're wandering about the wierdly rotated earth in my pics, its because Guerilla is up to its old tricks and the earth appears all wierd and rotated in the final sky. So, I rotated my earth pic anti-clockwise 110 degrees to counteract this effect.

7.) Now its time to compile the bitmaps in Tool. Go to the start menu in the bottom right of Windows, and find 'run'. In run, type 'cmd' and press enter. Now in the command console, type 'cd "c:\program files\microsoft games\halo custom edition"' and hit enter - do not include the single quotes, but use the double quote around the path for your HEK installation. Finally, compile the bitmaps running the command:

tool bitmaps sky/moonbase/bitmaps

You may get a warning for the bitmaps: 'bitmap with greater than 1 bit alpha...' for earth1024.tif and 'compressed as color key transparency' for nothing.tif. The colour key transparency is a bit of a mystery. I only got this error once when compiling bitmaps & to be honest, when I made the sky I got the 'bitmap with greater than 1 bit alpha' error for both. If when you get to the end of the tutorial, there are large blacked out squares in the sky, it will be because of the colour key transparency error. If not, then hey it worked!! Either way, the earth1024 though, must be fixed.

In Guerilla, goto file->open and navigate to ...//tags/sky/moonbase/bitmaps and open earth1024.bitmap. Change the format to 'compressed with explicit alpha' and save & close the bitmap tag.

(Click on thumbnail to enlarge)

Now, recompile in tool using the same command as above, and the bitmap tag should now be saved as compressed with explicit alpha.

8.) Close Photoshop & the command prompt for the moment, and go into Guerilla. If it's not already open, open up moonbase.sky, and scroll back up to the top of that dialog. Now we need to edit the gearbox model slightly, so next to model at the top, hit open.

The gearbox model (moonbase.gbxmodel) has been compiled as a model using tool, from a JMS file produced by 3ds max. Essentially, there are a number of 'regions' that have been created in the 3DS file, each with its own shader_transparent_chicago. These shaders, be found at the bottom of the gbxmodel in a drop down box include:

- Threshold - The large gas giant next to the Halo. We will be injecting our earth picture into this shader.
- Basis - A small moon directly overhead in this gbxmodel.
- Ring - The ring of the Halo as it dissapears into the distance
- Water - The reflective & animated water leading up to the horizon
- Horizon - err, the horizon
- Sky - The sky, including the blue dusk effect, stars and star twinkle.
- Clouds - The clouds visable on one side of the sky

As this tutorial is essentially about hacking the dusk.sky, we cannot delete or move the locations of these shaders, so what we are going to do is edit the ones we don't want to have an invisible texture, & inject our earth1024.bitmap into the one we do want.

In order to edit a shader, scroll down to the bottom of the Gearbox Model file, select a shader in the drop down box, change the shader to the moonbase version (ie: from sky\dusk\shaders\threshold_dusk.shader_transparent_chicago to sky\moonbase\shaders\threshold_moonbase.shader_transparent_chicago) and then open it. We want to change the shader to the moonbase version so that we don't accidentally & irreversibly change the dusk.sky!!

(Click on thumbnail to enlarge)

Repeat this for all shaders - change them over to the moonbase versions and open them. One by one, we then need to edit the shaders we have just opened. The process for this is discribed below.

9.) Adjusting the Shader Bitmaps.

In the threshold_moonbase shader, scroll down until you find the section called bitmaps. There should be a drop-down box with 'threshold' written in it. This is the current bitmap in use. We want to inject our picture of the earth here, so at the bottom of the maps section, where the path to the current map is, browse for and open the earth1024 bitmap, in your sky\moonbase\bitmaps folder. When you have finished, close the shader and save it.

(Click on thumbnail to enlarge)

Next, open up the basis shader. This is a small moon in the sky in the dusk model, so we want to edit this out. Import the bitmap 'nothing' - with its pure black alpha channel - into this shader. When the sky renders, the black alpha channel will cause this region to be completely transparent.

(Click on thumbnail to enlarge)

Continue this for all the shaders. The remaining shaders must all be injected with 'nothing.bitmap', but note that some shaders have more than one map. A description of which shaders have which maps are as follows:

Note: You can cycle through maps on a shader by clicking on the drop-down box under the maps section. Make sure you apply 'nothing.bitmap' to all of them!!

- Threshold shader: Map is threshold_dusk.bitmap. Replace with earth1024.bitmap
- Basis shader: Map is basis_dusk.bitmap. Replace with nothing.bitmap
- Ring shader: Map is ring_dusk.bitmap. Replace with nothing.bitmap
- Water shader: Maps are water_dusk, reflection_water_dusk & waves_bump. Replace all with nothing.bitmap
- Horizon shader: Maps are detail_ground and horizon_land. Replace all with nothing.bitmap
- Sky shader: Maps are stars, stars_twinkle and sky_dusk. Replace ONLY sky_dusk with nothing.bitmap
- Clouds shader: Map is clouds_dusk. Replace with nothing.bitmap.

10.) Adjusting the Lens Flare to get a realistic sun effect.

Close all the files you have open in Guerilla and open up the sky\moonbase\sun_in_space.lens_flare tag. We want to make some simple adjustments to this tag to get a more realistic, yellowly/red, burning sun in the sky with a bright corona & lens flare effect.

First, under the bitmaps section, check the box that says 'sun'. Also, while you're there, open up the flares_generic bitmap by hitting open just above the sun checkbox. Now, scroll down to the bottom of the lens flare tag where it says 'reflections'.

(Click on thumbnail to enlarge)

A whole load of parameters are here that effect the lens flare. First off, the current `sun_in_space` flare has a sort of blue/purple star shaped tint to the edges. The sun in our solar system is much more yellow and softer at the edges, so we want to change the image index and the tint.

(Image Index 0) (Image Index 2)

For Lens Flare Reflection Block 0, we want to change the image index from 0 to 2. This has the effect of removing the harsh star effect with a purple outline & replacing it with a soft corona that we can colour as we wish. Image indexes can be viewed from within the `flares_generic` bitmap, but hitting 'show bitmap' and changing the index in that dialog. Next, we want to change the radius to 0.3, to make it a little larger, the brightness to 1, and the tint to a soft yellowish colour.

(Click on thumbnail to enlarge)

Repeat this for all lens_flare_reflection_blocks, using the following parameters:

- Block 0:

Bitmap index: 2. Position: 0. Rotation: 0. Radius 0.3/0.3. Brightness: 1. Colour: r=0.988235, g=0.996078, b=0.737255.

- Block 1:

Bitmap index: 1. Position: 0.15. Rotation: 0. Radius 0.025/0.025. Brightness: 0.2. Colour: r=1, g=0.5, b=0.

- Block 2:

Bitmap index: 2. Position: 0.18. Rotation: 0. Radius 0.02/0.02. Brightness: 0.2. Colour: r=0.18, g=0.78, b=0.

- Block 3:

Bitmap index: 0. Position: 0. Rotation: 0. Radius 0.28/0.28. Brightness: 0.9. Colour: r=1, g=0.988235, b=0.72549.

- Block 4:

Bitmap index: 1. Position: 0.26. Rotation: 0. Radius 0.06/0.06. Brightness: 0.25. Colour: r=1, g=0.5, b=0.

- Block 5:

Bitmap index: 4. Position: 0.35. Rotation: 0. Radius 0.09/0.09. Brightness: 0.25. Colour: r=1, g=0.5, b=0.

- Block 5:

Bitmap index: 4. Position: 0.35. Rotation: 0. Radius 0.09/0.09. Brightness: 0.25. Colour: r=1, g=0.5, b=0.

- Block 6:

Bitmap index: 1. Position: 0.37. Rotation: 0. Radius 0.04/0.04. Brightness: 0.35. Colour: r=1, g=0.5, b=0.

- Block 7:

Bitmap index: 1. Position: 0.43. Rotation: 0. Radius 0.03/0.03. Brightness: 0.35. Colour: r=0.3, g=0.8, b=0.25.

- Block 8:

Bitmap index: 2. Position: 0.45. Rotation: 0. Radius 0.02/0.02. Brightness: 0.35.
Colour: r=0, g=0.28, b=1.

- Block 9:

Bitmap index: 5. Position: 0.52. Rotation: 0. Radius 0.05/0.05. Brightness: 0.25.
Colour: r=1, g=1, b=1.

- Block 10:

Bitmap index: 6. Position: 0.61. Rotation: 0. Radius 0.18/0.18. Brightness: 0.15.
Colour: r=0, g=0, b=0.

- Block 11:

Bitmap index: 4. Position: -0.08. Rotation: 0. Radius 0.05/0.05. Brightness: 0.15.
Colour: r=1, g=0.5, b=0.

- Block 11:

Bitmap index: 3. Position: 0.0. Rotation: 0. Radius 0.1/0.1. Brightness: 0.15.
Colour: r=0.88, g=0.43, b=0.

Ok, save close and save that file once you're done - You're lens flare is now complete!!

11.) Once you've done that, you have completed your custom sky mod!! Save all your files and to test the sky, open up a scenario tag for a map you are working on, & inject moonbase.sky into the multiplayer skies section!!

Final Stuff

Nice, now that you have learnt how to do your own custom skies, feel free to experiment with adding different bitmaps & such. Note that if you want to change the location and size of images, this is not possible by modding current skies. What you must do is create your own sky Gearbox Model before applying bitmaps in Guerilla. This is possible using 3D Max and a copy of tool - and is described [HERE](#).

EViLDaVE :@

SkyGBXModel

:P PAGE UNDER CONSTRUCTION :P

REQUIREMENTS: 3D Max, Photoshop, HEK

ACKNOWLEDGEMENTS:

I originally learned how to edit certain parameters of existing multiplayer skies using Mattyp001's [Custom Sky Tutorial](#) available from [Halomods](#). Using this knowledge I modded an existing Halo CE sky to include custom bitmaps - a full tutorial on this can be found here: [Moonbase Custom Sky](#). This method is inflexible as it requires that you use an existing sky Gearbox Model, so I set about trying to create a gbxmodel from scratch. I got the general idea down, but it suffered some parallax errors. This is documented here: [Nearly Figured Out Custom Skies](#). Finally, Molkien to the rescue, he releases his Cliffhanger sky as an open source download, allowing me to see where I was going wrong - simply that my sky model was too small: [Cliffhanger Open Source Thread](#)

So, basically, acknowledgements to **Mattyp001** and **Molkien** for their part in helping me figure out this tutorial!!

DESCRIPTION:

How to create the Gearbox Models for custom skies. If you dont know what I'm talking about, read my [Custom Sky Tutorial](#) first and you will understand the in's and out's of a Halo CE Sky model. Once you've created your sky Gbx model, you populate it with bitmaps as in the tutorial above.

STEP BY STEP INSTRUCTIONS:

Page will be up soon, watch this space!!

|S_s| Hury up Plz! ...Is anyone even working on it?

yes plz if your working on this hurry other wise im just gonna wing it and try to make one..

ok er.. I tried and i failed.. its actualy kinda simple to make a basic sky but adding more then one object gets wierd.. stuff appears in wrong order and i cant figure out why u have to give the gbx model regions some how per each object and order them correctly

TileTextureTut

REQUIREMENTS:

Photoshop, A poorly tiling texture, Some patience

ACKNOWLEDGEMENTS:

Original sand texture taken from [HERE](#) after a google search using keywords 'sand AND texture'.

DESCRIPTION:

If you want to use a texture as a detail map, you will probably be aware that images have a course seam at the edges where the tiled map lines up. This brief tutorial discusses how you can make a poorly tiling texture such as a photograph of sand or dirt etc into a tileable one, by editing out the seams in Photoshop. This is just an example, but the technique can be applied to almost any texture. Organic textures such as sand, dirt, grass etc are easier to tile than more regular ones such as brushed steel effect,

STEP BY STEP INSTRUCTIONS:

1.) First up, download the texture below. Save it as sand_original and load it into photoshop. This texture was originally taken from [here](#). All I did was select a 256x256 portion of this texture using the selection box (hint: use the rectangle select tool, and at the top of the page select style->fixed size and enter 256x256 as the desired size). I chose a section that had as few dark regions as possible, as these would make the tile effect more difficult.

D/L Sand_Original (256x256)

(Click on thumbnail to enlarge)

2.) With that texture now in photoshop, apply a high pass filter (Filters menu->Other->Highpass). I used a value of 25 pixels to smooth out any large variations in shade over the whole image. Experiment with different values if you want. A lower value such as 1-

10 will only allow very fine details through. A higher value has virtually no effect, other than to eliminate large shading anomalies.

Notice the texture has lost some of its colour. While this cannot be restored, the effect of equalising the content of the image can be achieved using image -> adjustments -> auto contrast. If the result of the auto contrast is too harsh, undo it and try manually adjusting it using image -> adjustments -> brightness/contrast.

Sand after HPF & Contrast

(Click on thumbnail to enlarge)

To restore the yellowish colour of the sand somewhat, goto image -> adjustments -> variations and vary the average colour of the texture. I hit the magenta button once, followed by OK to get this result below:

Sand after Variations

(Click on thumbnail to enlarge)

Now the next filter we are going to apply is an offset filter. Go to Filters -> Other -> Offset and enter 128 & 128 in the horizontal & vertical pixels boxes, and make sure wrap-around is checked. This will have the effect of tiling your texture within the 256x256 image - showing up the harsh seams at the edges. Note that if the size of your image is 512x512, you want to offset at 256x256 and so on. The image should now look something like this:

Sand after Offset Filter

(Click on thumbnail to enlarge)

3.) Ok, the image is now ready to be edited to become seamless. In order to do this, we are going to be using the clone stamp tool. This is available in most packages, not just photoshop, but this example will follow the photoshop layout as that's what I'm used to :P

(Click on thumbnail to enlarge)

The parameters of the clone stamp tool that are of interest are the brush size & style, the opacity & the flow (see image above). Essentially, what the clone stamp tool does is allows you to replicate a portion of an image, and clone it elsewhere. By choosing a soft edged brush and carefully selecting source & target sections of the image, its possible to edit almost anything to be seamless!!

To edit this image, first, select the clone tool, choose a brush style with a nice soft edge, and a size of 40 pixels. Drag the opacity down to 50% and leave flow at 100%. Now, if you right-click on the image at any point while holding ALT, that point becomes your source. Once you have defined a source, clicking anywhere else without the alt will make that section the destination. With an opacity of 50% set it may require 3 or 4 clicks to transfer a section of image to another part. If the result looks dodgy, you can always use the multiple undo levels (undo levels are viewable in window -> history) and try again. Here's an example:

Before (Left) & After (Right) Clone. Click on images to enlarge

->

In these images, I alt-cloned under the cross in the first image, and clone-stamped (no alt) under the circle. When clone-stamping, I held the mouse pointer still and right-clicked 4 times to get the desired effect, as the opacity is low. Look carefully at the two images - can you see the difference that the clone stamp has made? It has copied the section from under the cross and pasted it under the circle, eliminating the seam at this precise location.

Note: The clone source (alt & clone) is set as a RATIO from the destination in pixels. If you wish to stamp on another section of image, you must re-position using your source by pressing alt & clone over the desired source section.

Repeating this clone-stamp action, pasting sections from flat portions of the image over the seams is one method of removing the edges. However, do note that if you merely paste over the seams from flat areas of the image, you will still have a visible tile effect, where the edges of the texture are flatter than others. The trick to this is to carefully pick your source and destination in order to make it look organic - like the original texture. Also, varying brush size & performing multiple clones will allow you paste whole features from one part of the image to another.

4.) Continue your clone stamping until you are happy with the image. It would be too lengthy to show the entire process from start to finish, but try this for starters:

(Click on thumbnail to enlarge)

Above: This nice feature under the X can be copied to the O location to get rid of the seam in that location. Perform a clone operation using the X and O as source and destination respectively.

(Click on thumbnail to enlarge)

Above: Repeating this theme, I cloned the following features. X is source (press ALT & Clone) O is destination (Clone). I did these stamp operations in the order Green, Purple, Blue, Cyan.

(Click on thumbnail to enlarge)

Above: The result of the above operations. Cunningly place the clone source & destination to copy features from one location of the image to another. If you don't like the outcome of a clone op, simply undo it and repeat as necessary. Notice that some operations cause the image to appear blurry. Use a nice featureless section of the image that has sharp image qualities, such as the flat sandy area in the middle-left to clone out these blurry sections.

(Click on thumbnail to enlarge)

Above: Finally, your image should be seamless to the eye. This is an example of my seamless texture, it will most likely differ from yours, but the most important thing is to make it look natural!!

5.) But it's not finished!!!! In removing the seams, you have now introduced additional seams when the tiled texture is de-offset. In order to see these, apply the offset filter once more: Filter -> Other -> Offset, and select values of 128x128 with wrap around on.

(Click on thumbnail to enlarge)

This time the seams are nowhere near as severe though. This is just a result of applying a clone stamp near the edge of the original offset image, thereby introducing seams when you revert to the original form. In order to get rid of these slight seams, repeat similar steps to those laid out above, selecting source and destination for the clone stamp, but

making sure that the stamp brush DOES NOT go off the edge of the image - keep it within the image boundaries!! Once you've finished your clone stamping, apply a weak gaussian blur (Filters -> Blur -> Gaussian Blur) with a strength of 0.5 and save.

The Final Tileable Texture

(Click on thumbnail to enlarge)

Test your texture by saving it and setting it to the background image in Windows & selecting the option 'Tile' rather than stretch or centre. If it looks something like this, then you've done well!!

Tiling the Sand Texture

(Click on thumbnail to enlarge)

Additonal information

In PSP8, seamless tiling is very easy. Check this very short tutorial about [Seamless tiling in PSP](#).

Final Stuff:

This tutorial explains the principle of making textures tileable & the best way to learn is to try it out for yourself on images/textures downloadable from the web. The sand texture above tiles pretty well - there are still some seams that need working out, but overall its a fairly decent job.

One point though, the sand texture with it's large amounts of highs and lows will make it inherently difficult to texture map, as from one direction those ridges in the sand will look raised, but if you view the texture from the other direction, the ridges will look like deep furrows in the sand. Also, this map is only 256x256, and with its large variations over the image, even though the seams are not visible, it may well be possible to see the repetition if you were to use this as a detail map.

It depends on what you consider acceptable in your map - if you dont mind this, then by all means, use this type of texture. If it's more of a problem, it may be a better idea to have a flat sand image as the detail map, and apply a greyscale bumpmap to give the appearance of ridges/furrows in the sand. This will mean that the ridges will appear as raised ridges no matter what angle you look at them, although creating a realistic bumpmap is certainly not easy!!

EViLDaVE :@

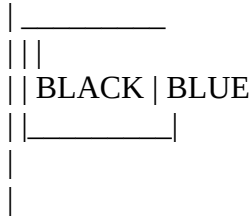
Addendum for Photoshop 7 users. The healing brush will speed this process up exponentially. It works the same way as the clone stamp tool, but it automatically blurs and does some other stuff all in one thing. Mess around with it, it can save you some time.

CompilingSprites

Creating Sprites

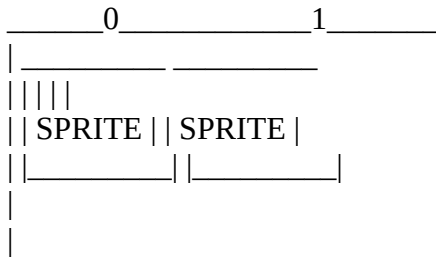
Load up your favourite image editing program.

1. Create a bitmap any size you want (unless it's a replacement bitmap, in which case you need to look at the bitmap tag's page width and height.)
2. Fill the image with the colour blue (RGB 0, 0, 255)
3. Create a black rectangle in the top left corner of the image, and make sure it is at least 1 pixel from the top left of the image. The image should look something like this: _____

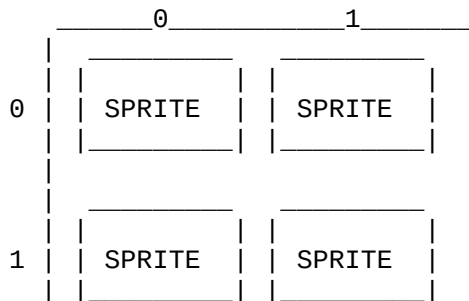


You have now created a single sprite, and a single sequence.

1. To add more sprites to this sequence, add more sprites to the right of this first sprite, for example:



2. To add a new sequence, start a new line of sprites:



(by the way, the numbers on the left represent the sequence index, whilst the numbers on the top represent the sprite index. Ignore these, they are not part of your final image)

Save the image as .tif in a folder under "Halo Custom Edition\Data"

Compiling

There's nothing to it really, simply compile this bitmap like any other bitmap.

Then open in guerilla, change the type to sprite, and possibly some other values to match up to the type of sprite you are making (for example
ui\hud\bitmaps\msg_hud_icons.bitmap - correct me if that filename is wrong).

That's it, you have a sprite bitmap.

SkinningTutorial

Here is my tutorial skinning HaloCE models

there are two parts to this: the first is on the plain skins and the second is on cubemaps.

1. okay, since the HEK doesn't provide you with skins of the vehicles and etc. you can do one of two things: use HMT to export(which doesn't work with cubemaps) or you can do it the ghetto way aka my way.
2. open up the .bitmap of the vehicle skin that you want in geurilla.
3. now open up paint.
4. now in geurilla, show the bitmap so you can see it.
5. hit print screen on your keyboard.
6. now in paint paste in the screen shot and move it into the top left corner.
7. make sure that you get the image exact.
8. now reduce the size of the bitmap down to the correct power of two(32x32,64x64,128x128,512x512, etc.) use the attributes dialog from the image menu to get it right.
9. now save this and import it into your higher end program of choice.
10. do what you want and all that and save it as a .tiff and convert it to a .bitmap like you usually do.

now with the cube maps this is a little bit more difficult.

1. you do the same thing with the opening of the .bitmap to it. you will see 6 little textures.
2. now do the same print screen thing to get that picture into paint.
3. now open up data\levels\ai10\bitmaps\cube map hazard light.tif(i use this because the black colors contrast with the green borders on it) in the higher end program that you use.
4. now one by one select the squares from the paint and put them exactly over the "template" in the same spots. *i strongly suggest using zoom to get this exact*
5. when you are all done with coloring and what not, save this as a separate map.
6. now when you run it through tool.exe it won't give you errors.
7. now open up the newly created .bitmap in geurilla.
8. make it so that it shows the map, you will notice that all the little pics are separate from each other and it only shows one at a time.
9. to fix this all you need to do is on . bitmap you have to change it to say cube instead of 2d texture(this is right below the show bitmaps button)
10. now save it.

11. re-run the tool.exe with the bitmap again and now it will be in the cubemap format.

Author: mattp001

AdvancedUVW

:P PAGE UNDER CONSTRUCTION :P

This tut will cover advanced UVW mapping using the UVW Unwrap function in 3DS Max. Far from the method previously presented where you manually move UVW's to get the desired result, this method uses a cunnninnng technique to get that mapping almost perfect - well, more perfect that it was!!

Particularly useful for skinning complex objects that don't fit the basic planar, spherical, cylindrical or box maps!!

Page will be up soon, watch this space!!

Aw Sh** I dont have much time these days. If someone feels like filling this one in, go for it, else I will do it when I get the chance!!!

Edited by Sonic Tag:

Create your mesh, if you are just going to test doing this, then create a box and include some extrudings to make it look as less like a box as possible.

Next, you can then go into the material editor and change the diffuse of a standard material to checkered and aplpy it, this is so that in the perspective you can see what it lookslike with a tessalated texture which can be of some importance, it also shows you if you have any streched areas on your model.

Go to the modifier > UVW MAP, set it first to planar, then go to Modifier > Unwrap UVW, Press Edit, at the top you will see some options, these are very useful to you, especially normal mapping and flatten mapping, unfold mapping is better for the more simple of objects, I use it alot after I have simplified a mesh into alot of smaller meshes within the UVW Wrapping co-ordinates, and then it makes life a little easier. I would reccomend setting up some shortcuts at this point.

Select Normal Mapping, then go down to Box mapping in the selections for that, and apply it. Then you should move, rotate everything appropriately, keep checking your checker map. Make sure that it is all in the box.

For the next part what you will want to do is get a tool called Texporter, what this does is take your UVW mapping, and turn it into an image, so you do not have to take a screenshot of the texture coordinates. For this part, you should make it a square, I would recommend 512 x 512, or 1024x1024 as these are squares and are multiples of 2 which most game engines support.

You can test out this by applying the UVW picture from texporter into your mesh, and taking a render. This should have no red parts, if it does then it means that there are overlappings in the mesh, and also you should see the details going along the edge of the polygons.

Now you can simply go over that in your paint program of your choice, I personally prefer Paint Shop Pro, although Photo Shop is better in the perspective of most people.

Most people think that a head is a very complicated shape to unwrap, but it is actually very very simple, all you do is select your head, go to the modify panel and scroll down on the list to UVW Map, and change it to planar, then in the Unwrap UVW modify tab, just move some vertices around and that is it, you might not even have to move any vertices around. Depending on the shape of the head it can get very complicated.

When your shape gets very complicated, what you can start to do is take rows of polygon in the same axis which don't have to much rotation from each other, maybe even columns of polygons also, and apply a UVW Map in Planar to the correct axis that you have selected, so that the planar will cover it as if it is above that section, then you do this a lot and eventually you will have created your unwrap, when you are doing this, make sure that you weld some of the vertices together in the UVW window, it will make it much easier to apply a paint to.

Automatic UVW mapping method - Click on the "Mapping" menu, then "Flatten Mapping". Make sure the three checkboxes are checked, and hit "OK". You'll see your object broken up into chunks, almost like pieces of a puzzle. If you printed this out and cut out all the pieces, you could actually assemble them into a rough version of your object - they're not perfect, but they're pretty close. Although you get a lot of odd triangles, in the perspective 3d viewport you should try and select these and put them into a planar map, which is basically a more automated version of selecting the many triangles at a time.

There is also something else you can do, basically take parts of your model with having face select on (to do this click the + on the modifier pannel next the Unwrap UVW and then start with the more central face, then basically take a few faces at a time (depends on how flat they are, but if they are not flat then you will want to select them one at a time or row at a time) and basically you planar map that whilst you are in the unwrap UVW so you just have to scroll down the modifier list a little bit, and then simply in max you can stitch it. Here I would definitely reccomend you have a stitch shortcut, because it makes life alot easier and it speeds up your modeling process by a substantial ammount. and you simply select the few verts where the part that you just planar mapped should attach to and press ctrl + s and it will automatically attach.

There are a few video tutorials I hear for this method of modeling and if you wait then I will make one soon.

CreatingNewTextures

This is a tutorial for making a new skin for the warthog. You can apply this tutorial to any object.

This tutorial will work on anything. im just using the warthog as an example because I wanted to put Quaglinoman's tron warthog into ce

Ok, first of all, you will need HEK and Photoshop or PaintShopPro with the dds reading plugin Secondly in your data folder, make the following file structure

data (should be there already, it's the folder were starting in)

```
vehicles
  warthog
    bitmaps
```

Take the texture in Photoshop or PaintShopPro and save it as a .tiff in the bitmaps folder (if it is a patriotic warthog texture call it patriothog.tiff etc.) make sure you use LZW compression or else the pictures will have a huge file size

Then open up the command prompt and type in

tool bitmaps vehicles\warthog\bitmaps and then hit enter

Once that's done, go into guerilla and open the warthog hull.shader_model. Then scroll down to where it says Maps and under that should be a selection box and it should say base map. To the left there is a little box with three dots, click it. Search for your

patriothog.bitmap which you converted from a tiff, double click it once you find it. Now save it.

Open up a .scenario map in Sapien and make a hog (mission>objects>units>vehicles and edit types and find the Warthog.vehicle) and congrats

SoloCutscene

Making a Cutscene for Solo Maps

by p0lar_bear

This is going to be complicated... grab some popcorn, 'cause you're going to be here for a while. Also, make sure you have a valid actor and actor_variant tag. This is no problem if you have the HEK+, or TheGhost's massive tagset.

There are two ways to get bipeds to move around in cutscenes. The first way is by creating custom animations in 3D Studio Max. The second way, which is a bit more complicated, is by attaching bipeds to an AI encounter, and making them execute a command list. I prefer the AI method for basic cutscenes; they will usually look better than if I got a hold of a cyborg biped rig and tried to animate it. :X

Anyway, to start, pick your characters to use in the cutscene. For this one, we'll make the Master Chief run into the red base of Blood Gulch, look at another cyborg, then kill it.

First, make a copy of the cyborg.biped tag, and rename it to something like "cinecyborg.biped." Now, open your new tag in guerilla, and scroll down until you see into about aiming and looking velocities and acceleration, which is in the "camera tracks" section of the biped tag. Set them as follows:

```
Aiming Velocity Max: 360  
Aiming Accel Max: 720  
Casual Aiming: .8  
Looking Velocity Max: 720  
Looking Accel Max: 1440
```

If we did not set these, the Master Chief's looking and aiming wouldn't look good. His head and gun would jerk around instead of smoothly gliding into place. Now, save the tag.

Go into tags\levels\test\bloodgulch, and make a copy of bloodgulch.scenario. Rename it to something like "sologulch.scenario." Open that in Guerilla, and use [my tutorial on making a solo scenario](#) to make the map work as a solo map. Save, and exit Guerilla.

Open your newly created sologulch tag in Sapien. Once it loads up, immediately add your cinecyborg tag to the biped palette. Place one cyborg outside of red base, a bit away from a side door, and place another on the flag marker. Name the one outside "chief", and the one inside "deadman".

Now, create a new AI command list set (mission/AI/command lists), and name it something like "cyborgkill". First thing to do is set some command list points for the Chief to move to. Set one in front of the side door, one at the wall inside the base, one at a doorway inside the base, one right next to deadman, and the last one you will place on deadman's spot, but use the properites palette to raise it to a height of about .6 (z axis).

Next, make a command list. To make each command, hit "new instance" on the heierarchy view, and then select the command you wish for the AI to execute in the properties palette, and how to execute it. (NOTE: Be careful when switching between commands; sometimes Sapien will crash if you switch from a command with a lot of options, like vocalize, to a command with very few or no options, like pause. When changing a command, make sure to always select the first option on the drop-down box, if there is one.)

Try to use this knowlege to create this command list; poke around a bit and you'll find the correct options:

Go to point 0 and keep moving
Go to point 1 and keep moving
Go to point 2 and stop
Look at point 5 for 1s
Wait until told to advance
Go to point 3 and stop
Wait until told to advance
Shoot at point 4 for 1.5s
wait until told to advance

Save your scenario, quit, and reload sapien. The reason for doing this is because AI doesn't recognize command list points unless you restart the engine. There may be other ways, but this is known to work 100% of the time.

Once sapien is loaded up, add an actor variant tag to it; I usually use "marine_armored assault rifle", but anything will work. create a new AI encounter and name it "cinematic1". Make sure it is not initially created, and initially blind, deaf, and braindead, then set the team index to 1 (player). Add a squad to it, and name it cineborg. Give it the actor type you added to palette, and that should do it. Save the scenario, and we'll move onto setting some camera points.

Go into sapien's camera mode, and find a good point to show the master chief running into the red base. Position the camera as needed, and hit the spacebar to save the camera's position. Name it "cam1". Now, to get good camera points, we'll attach the chief biped to the cinematic AI encounter. Bring up the console and type in (ai_attach chief cinematic1). Now, type in (ai_command_list_by_unit chief cyborgkill), to make the chief biped execute the command list. Find a good spot to show the chief running through the base that doesn't reveal the deadman biped. After getting that camera point (name it "cam2"), get another one that shows the chief looking at the deadman (name it "cam3"). To get a good close-up of the chief owning the deadman, type (ai_command_list_advance_by_unit chief) into the console, and the chief should run up to the deadman. Find your camera point and save it (as "cam4").

After this is all done, make a trigger volume (mission/game data/trigger volumes) named "cutscenestart" outside of the red base entrance the chief goes into, so the cutscene will initiate when the player walks into the trigger. Also, make one way the hell off somewhere in the map that the player can't get to without hacking, and name it something like "hide"

Finally, go into the Flags section, and add a flag somewhere that can't be seen from the cutscene area, and call it "hideplayers." Then, make one on the chief's final position and call it "showplayers".

Now that you have your AI, command lists, trigger volume, and camera points set up, let's make a script to make it work. Assuming you used all suggested names, let's go with this:

```
;;; This first script removes the bipeds used in the cutscene, so
they're not visible until the cutscene starts (script startup
hidethebipeds
(object_destroy chief)
(object_destroy deadman)
)

(script startup cutscene
(sleep_until (volume_test_objects cutscenestart (players)) 5) ; Stop
script execution until a player walks into the trigger
(fade_out 1 1 1 30)
(player_enable_input 0)
(sleep 60)
(volume_teleport_players_not_inside hide hideplayers) ; Hide the
players by teleporting them out of the area
(cinematic_start)
(cinematic_show_letterbox 1)
(object_create_anew chief)
(object_create_anew deadman)
(camera_control 1) ; Let the script control the camera
(camera_set cam1 0)
(sleep 30)
(fade_in 1 1 1 30)
(ai_attach chief cinematic1)
(ai_attach deadman cinematic1)
```

```

(ai_command_list_by_unit chief cyborgkill)
(sleep 60)
(camera_set cam2 0)
(sleep 30)
(camera_set cam3 0)
(ai_command_list_advance_by_unit chief)
(sleep 30)
(camera_set cam4 0)
(sleep 30)
(ai_command_list_advance_by_unit chief)
(sleep 30)
(unit_kill deadman) ; This is included to insure the biped dies, in
case the weapon's fire doesn't kill it
(sleep 60)
(fade_out 1 1 1 30)
(sleep 30)
(object_destroy chief)
(camera_control 0) ; Put the camera back onto the player's FPV
(volume_teleport_players_not_inside hide showplayers) ; Puts the player
back where s/he should be
(cinematic_show_letterbox 0)
(cinematic_stop)
(sleep 30)
(fade_in 1 1 1 30)
(player_enable_input 1)
)

```

Save that to a .hsc file in the folder data\levels\bloodgulch\scripts. When done, go into sapien, and click "Compile Scripts" in the File menu. If you get any errors, debug them using a little common sense. Once the scripts are compiled correctly, save the scenario, exit sapien, and compile the map. Run Halo CE, and at the main menu, bring up the console and type in "map_name sologulch." Walk into the spot where you placed your trigger volume, and the cutscene should play. Sit back, relax, and enjoy. Even if the cutscene looks like crap, just take a bit of pride in knowing that you now know HOW to make a cutscene, and you can build upon it to make it even better.

Trigger Volumes

Trigger Volumes for Dummies

by p0lar_bear

The first thing you need to know about making ANY kind of single-player map is how to use [trigger volumes](#) effectively. I will explain all of this here. All you need is Sapien, and a text editor like notepad. As an added bonus, you could also grab the HEK Plus app by

SteelIXB and rip some actor/actor variant tags out of a Halo Campaign map, or you could download TheGhost's large tagset, which includes every element used in the Halo campaign.

First, let's make an AI encounter and squad that will be placed when the player enters the trigger volume. If you need help with setting up an AI encounter, watch Aztec's video tutorial. Just make sure that when you make the encounter, the "Not initially created" flag is checked, and let's give it a name of... oh, encount_1.

Now, let's place a [trigger volume](#) in an area that the player will **have** to go through. When it is initially placed, it may be smaller than you need it to be. If you want to make it larger, click and drag on a side to move it, thus making the volume bigger. Then when done, name it something easy to remember. Let's name this one trig_1.

Now, you may notice that there is nothing you can do with the [trigger volume](#) in Sapien to make it place an encounter when a player enters it. The reason why there's nothing there in the original version and the modified version of Sapien is because trigger volumes are, always have been, and always will be, referred to in scripts.

So, let's open up notepad, and create a script. The best and neatest way to make a [trigger volume-related](#) script is to make a startup script that sleeps until the player walks into the [trigger volume](#), then continues executing. So, let's make this [trigger volume](#) create the AI encounter, wait until the encounter has been totally wiped out, then make a checkpoint and play the "killtacular" sound just for kicks.

```
(script startup h4xmb
```

```
(sleep_until (volume_test_objects trig_1 (players))15) ; Pauses script
execution, and checks every 1/2 second to see if a player walks into
the trigger "trig_1".
(ai_place encount_1) ; Places the ai encounter "encount_1"
(sleep_until (= (ai_living_count encount_1) 0) 15) ; Pauses script
execution, and checks every 1/2 second to see if the encounter
"encount_1" has been totally wiped out.
(game_save) ; Checkpoint... done.
(sound_impulse_start sound\dialog\multiplayer1\killtacular none 1) ;
Killtacular!

)
```

Save the file into your level's scripts folder (if your map's folder is in data\levels\mymap, you would save the script into data\levels\mymap\scripts) as myscript.hsc. Then go back into Sapien and compile the scripts (File | Compile scripts). There should be no errors. Now, save the scenario and compile your map in Tool.

Now, start Halo CE and when you get to the title screen, load the map by typing "map_name <your map's name here>" into the console. When you load the level and

walk into the [trigger volume](#), the encounter should be placed, and when you kill it, you will get a checkpoint, and the Killtacular voiceover should play.

NavPoints

Navpoints/Waypoints Tutorial

by p0lar_bear

Meh. Same difference. Via scripts, there are four ways to set a navpoint (the engine refers to them as waypoints), and those are by attaching the waypoint to either a cutscene flag or an object, and have it visible to either one unit, or a team. To set a waypoint, use (activate_nav_point_flag) or (activate_team_nav_point_flag) to attach the waypoint to a cutscene flag, and use (activate_nav_point_object) (activate_team_nav_point_object) to attach a waypoint to a named object. When done with a waypoint, use a (deactivate_nav_point...) or (deactivate_team_nav_point...) to turn it off. Since I don't know how to refer to the players as units, I usually set a team waypoint. The full syntax of a team waypoint command is (activate_team_nav_point_flag/object <waypoint type> <team> <flag/object> <vertical offset>).

<waypoint type> can be, by default, "default", "default_red", "flag_blue", "flag_red", "target_blue", "target_red", "skull_blue", "skull_red", "crown_blue", and "crown_red". Singleplayer maps should use only the default, and maybe, if you have to kill a certain enemy, the target nav points. the "default" type and red types of waypoints should be used on urgent things, like the exit on an area that would have a timed escape, or an enemy that is about to own you.

<team> should always be set to "player", if you want the player to see the waypoint!

<flag/object> is what you are attaching the waypoint to, which depends on the type of command you're using. Simply put in the name of the desired object or flag you wish to attach the waypoint to.

<vertical offset> is how much higher on the z-axis the waypoint is placed from the object or flag's marker in sapien. If you attach it to something on the ground, like a door, biped, or weapon, make sure to set it to about .6 or higher, but if its something like a picture on a wall, or if the flag/object is otherwise not on the ground, set it to 0.

(deactivate_team_nav_point_flag/object) is a bit different. The syntax for it is (deactivate_team_nav_point_flag/object <team> <object/flag>), where <team> is what

team you wish to not see the navpoint anymore, and <object/flag> is the object or flag the navpoint is attached to.

So, let's say we wanted to set a normal waypoint at a position we want the player to go to. Lay down a cutscene flag, name it nav_1, then add a script in your map that has this command: (activate_team_nav_point_flag default "player" nav_1 .6)

Now for something a little more useful. Let's make it so we put a waypoint on a warthog, and it turns off whenever the player is in the driver seat, and turns on if the player is ejected. Place a warthog and name it drive_me. Now, make and compile this script:

```
(script continuous hog_checker (if (vehicle_test_seat_list drive_me "w-driver" (players)) (deactivate_team_nav_point_object "player" drive_me) (activate_team_nav_point_object "default_red" "player" drive_me 1)) )
```

In theory, that will constantly deactivate the waypoint while a player is in the driver seat, and then constantly activate it when there is no player in the driver seat. I don't know if more than one waypoint can be put on the same object or flag, but if it doesn't, then this should work fine.

Anyway, read up `hs_doc`, or the HS Bible, and find out more ways on how to make and use waypoints effectively.

SoloRules

Rules of Thumb for Making Solo Maps

by p0lar_bear

Here's a list of some tips, tricks, and other stuff with scripts to make a solo map good.

Friendly AI

Whenever you make an AI encounter you want to be firendly with the player, add in the (ai_allegiance) command to make an AI team friends with the player. For example, if you make a solo map in which you are a Covenant Elite, and would like Covenant encounters to buddy up with you, use this command in a startup script: (ai_allegiance "player" "covenant"). Valid teams are "player", "human", "covenant", "flood", and "sentinel".

Checkpoints

To make a map less frustrating to play, you have to know when to make the game do a checkpoint. Usually, I'd put checkpoints before and after BIG battles, long sequences, or when you have to traverse a large distance and would like to not have to drive/walk a long ways again, like in the Silent Cartographer. The checkpoint script command is (game_save).

Objective Parameters

Let's say there's a vehicle, person, or position you have to protect, or else the mission fails, like protecting Capt. Keyes in the end segment of Truth and Reconciliation. How they did that was, they placed a Keyes biped down with a name, and attached it to the AI encounter of the marine squad that follows you around. There was a script that continually checked for Keyes's health, and if it hit 0, the game would revert to your last checkpoint.

To make a game continually check a unit's health, make a continuous script with the command (unit_get_health <unit>) in an (if) statement, so when the unit dies, the game is lost. Here's a sample, which assumes your biped's name was "protectme":

```
(if (= (unit_get_health protectme) 0) (game_lost))
```

You can also make it wake a dormant script that uses (camera_control 1) and (camera_set_dead <unit>) with (game_lost) to show how badly you phailed and rub it in your face. >:D

Friendly to all

Make sure your map can be played by all. Do not force the player to use a "skill" weapon in an area that may cause a newbie to get killed 50 times before they finally get it. You should usually provide alternative routes and strategies. For example, before an enemy-infested area, stick some dead marines with dropped pistols, ARs, shotguns, maybe rockets, and ammo. And maybe there's a cliff next to you with a sniper rifle on it? You don't have to set all of that down, but don't just lay down a plasma pistol and expect everyone to be able to get through the area. Also, do not make enemy AI overpowered or overwhelming in number without a powerful weapon or vehicle to help the player get by. Keep in mind that it takes at least 8 headshots to an fully sheilded elite with a pistol to kill, and not 3 like the cyborg biped.

MusicScripts

Scripting music in Halo: CE maps.

Warning: Scripting syncing requires use of bipeds and vehicles (or other crushing objects.) in order to work over the netcode. These methods will work for single player maps, or multiplayer maps built for just one person. Syncing over Halo's netcode will most likely not happen unless you work with the bipeds and vehicles. Remember - death's sync, most else does not.

Warning 2: This method will use AI, and will check for the status of AI. AI ***DOES NOT SYNC*** over the netcode. Don't try, you will fail. :)

Tutorial:

This tutorial will assume that you have basic scripting knowledge, basic AI knowledge, can type with a keyboard, and can read. If you do not qualify for any of these needed qualities, please, do not hesitate to smash your face into your keyboard until you can no longer see.

What you'll need:

1. A compiled .scenario tag that you can edit in Sapien. 2. A .sound tag that you have compiled with tool. This should be the music you want to play in-game. 3. AI actors and actor variants. If you don't have these, check over at the files.halomods.com forums. Search around there for "Aztec's Actors" or something like that, and download his actor and actor variant tags.

open the desired .scenario file in Sapien. Place your AI encounters and squads, and give them appropriate names. For this tutorial, I'll call my encounter "tutleets1", and the squad name will be "uber." No reason for the names. :P

Anyways, place the AI, and under the encounters block and the information for it, check the box that says "Not created automatically", or something to that effect. You can skip the checking of the box, but for this script, we will use commands that place this encounter on the map at a certain time, instead of right away as soon as the game loads. This can come in handy once you work with trigger volumes/loading zones/multiple BSPs, and that sort of thing. Make sure you give your AI spawn points, check the required boxes, and make sure they have respawn enabled, and have their max/min actors set to something other than zero.

Now, load up *gasp* Notepad, and type the following (Note: DO NOT ADD *'s.) :

(script startup musicnai

```
(ai_place tutleets1) *  
(sleep_until (= (ai_strength tutleets1/uber) 0)) **  
(sound_impulse_predict sound\music\madrigal true) ***  
(sound_impulse_start sound\music\madrigal none 1)) ****
```

* This will place the AI that we decided to not create automatically.

```
** This will pause execution of the rest of the script until the give  
condition is true, or in other words, until your AI are all dead.  
*** This script will bring your sound or music file into the sound  
cache for your map. This has to be done before actually playing the  
sound, as far as I know. I'm assuming your file is found at  
sound\music\madrigal.sound for this. If you were to name your .sound  
file anything else, change "madrigal" accordingly in the script.  
**** This will start playing your music.
```

I believe that after the last command in the script, you can do a (sound_impulse_predict [sound location] false) command, to remove the song from the cache. But, I'm not sure if you'd have to run a sleep command first, or if you could just run it right after running the sound to begin with. That could stop the sound abruptly, but I'm not sure. Go ahead and experiment with this on your own.

After typing that, save the file as scripts.hsc in your levels\[your level]\scripts directory, or something like that. Go into Sapien, and go to File > Compile scripts. If it gives you any errors... well, try and fix them. I'm not a damn troubleshooting guide. :P

Of course, to play music, you don't need AI or anything. You can use trigger volumes and things like that, but that gets just a tad more complicated. This is just basic scripting using an AI command or two.

Final note: This is a very basic way to play music in Halo: CE. I'm sure there are better, more elaborate ways, but this will get the job done. :)

EDIT: WARNING: Make sure you check for the types of sound files you are using. If it is a loop, you will need to replace sound_impulse_predict with sound_looping_predict, and you'll need to replace sound_impulse_start with sound_looping_start. Otherwise, it should just stay how it was said to be in the tutorial.

Pkmnrulz240/Dark Cloud

FixingOpenEdgeToolErrors

REQUIREMENTS: 3D Max, Tool, A Recent Tool-Compiled JMS file that Has a S*load of Errors**

ACKNOWLEDGEMENTS:

I originally learned this technique from a video tutorial by Kirk4 on [Halomods](#), so great praise & thanks to him for spreading the knowledge :D

DESCRIPTION:

Errors that are returned by Tool.exe upon level compilation such as open edges, coplanar surfaces etc... can be located and fixed easily in 3D Max using the following technique. This technique also applies to many of the tool errors listed in the HEK tutorial, including:

1. WARNING found nearly coplanar surfaces (red and green)
2. WARNING: found duplicate triangle building connected geometry.
3. WARNING found # degenerate triangles.
4. WARNING unearthed edge (magenta boxed lines)
5. WARNING: a surface clipped to no leaves (see cyan in error geometry)
6. ERROR found z buffered triangles (red).
7. ERROR edge # is open (red).
8. ERROR couldn't update edge # (red).
9. ERROR couldn't build bsp because of overlapping surfaces (orange)
10. ERROR: portal does not define two closed spaces. (see yellow in error geometry)

STEP BY STEP INSTRUCTIONS:

1.) Geometry created for use in the HEK must adhere to the sealed world rules, described in the HEK Tutorial Document, under GeneralOverview. A sealed world is one where all edges and faces continuous lead on from another, and there are no holes, gaps, or misaligned edges. In addition, a number of other factors will cause Tool errors, such as intersecting faces, two vertices occupying the same space and incorrectly applied materials/UVWMaps.

2.) Whenever Tool finds an error, (you may get hundreds), it produces an error file. This can be imported into 3DS as follows:

- In 3DS, Select file->import
- Select the file type as VRML
- Navigate to your Halo CE installation directory
- Select the file called mymap.wrl (where mymap is the name of your map)
- Hit OK to import the file

3.) Now, deselect everything in your 3DS environment and press 'h'. Up pops the objects lister. A bunch of objects have been added that highlight the exact areas where there are problems. These are called VRL01 or something like that. Select them all.

4.) This geometry is now selected and will appear to be white. In your viewports, it is now possible to locate the errored regions of your geometry by jumping to the location where the a single peice of selected error geomoetry is. Edges are quite discrete, so a thin white line may well be concealed by your objects. Coplanar surfaces result in red & green triangles being added to your 3DS document. Unearthed edges result in magenta boxed

lines being added to your 3DS file and so on (see HEK tutorial, Level Compilation->Troubleshooting, for a full description of error geometries and their manifestations).

5.) Pan & Zoom your Top, Right & Front viewports, and rotate/zoom your perspective viewport until a particular piece of error geometry is in all four view. If it is impossible to see the error geometry (sometimes it is obscured by your object), right click the move gizmo with the error objects selected, and move them 1 point in any direction. This is just to help clarity, don't worry about it if you can see the error geometry.

Example for fixing an Open Edge

7.) In the perspective view with the errored geometry in view, you have to try to locate that open edge. An open edge as described by the HEK tutorial as follows:... (Find it under level compilation in HEK tut)

There is an open edge in the level.

Typically, there is a "hole" in the level that is preventing the level from satisfying the Sealed World Rules and therefore the level cannot compile successfully.

Basically, if you have two faces that don't quite meet, a missing face, or you have accidentally moved a vertex so a hole is created, tool will flag an open edge error. The image below shows a rather blatant open edge between a box (simple base :P) and ground (green plane). The vertex that is selected is misaligned to the box, creating a visible hole. This hole must be closed in order for Tool to compile the map successfully.

(Click on thumbnail to enlarge)

8.) In this simple case, the open edge can be fixed by one of two methods. Firstly, the vertex could simply be moved to align with the appropriate vertex on the base. This alignment must be precise, so rather than moving it by eye, select the vertex on the base you want it to align with and enter the exact x, y & z locations for the vertex on the ground.

Note: Exact transformation can be performed by selecting a vertex (or object) with the move gizmo held down, and then entering the x, y, z values in the boxes at the bottom of the screen in 3DS. If the boxes are grayed out, move the vertex a little, hit undo and then they should be activated.

The second, and preferred method for fixing this and more complicated open edge problems is to delete the faces surrounding the error, and reapply them. For instance: select the plane and go into edit mesh mode, & face selection. With ignore backfacing off, in the top or perspective view, delete the faces surrounding the open edge like so:

Now, with the plane selected, and still in edit mesh mode, hit attach in the edit mesh dialog and click on the base. This will attach the base to the ground - it will become part of the ground object, inheriting the ground material:

In the edit mesh dialog, go to face selection mode, and hit create. Now create new faces around the base, so that it is properly attached to the ground.

Note: When creating faces, you click on the vertices you want to become part of the face cyclicly, ie: hit the three points of the triangle either anticlockwise, or clockwise. The direction you turn affects the normals - the direction the new face will point. Clockwise will create a face pointing away from you. Counterclockwise will create one facing towards you. If you mess up the creation, and need to invert a face's direction, hit flip-normals under surface properties with that face selected.

Finally, detach the base from the ground by selecting all the faces that make up the base, and hitting 'detach' in the edit mesh dialog. Remember that to select multiple faces, press control while selecting, or dragging a lasso. To deselect faces, press ALT.

9.) Fix all your open edges one at a time using the method presented above, by jumping to the location, inspecting the geometry and if necessary, deleting and re-creating faces. Remember to re-apply materials, smoothing groups and UVWmaps after editing faces. Sometimes a hole in the level is simply created because you accidentally moved a point. If this is the case, just move it back, precisely though, using the x, y, z boxes as explained above.

10.) Once you have fixed them all, hit 'h' again, select all the error geometry, delete it, save your 3DS file and re-export. If more errors turn up in tool you can import them as before and look for the solution in the HEK tut.

Application to Fixing Any Other Tool Errors

Each Tool error has its own little personality that must be learned & respected, and for each, the solution is different. In my experience, I have come across open edges, z-buffered triangles, coplanar surfaces, degenerate triangles and I've even had the good old 'couldn't update edge'.

While most of these can be fixed by deleting faces and re-creating them, this is both tedious and sometimes unnecessary. Look to the HEK Tutorial (Level Compilation->Troubleshooting) for a description of what your particular Tool error means. This explanation can sometimes be cryptic, for instance coplanar surfaces is explained by the HEK Tut as where 'two surfaces exist that are coplanar'. Thanks... In fact, all this means,

is that Tool got confused because two surfaces that are at an angle appear to be in the same plane, and moving a vertex that these two triangles share will disrupt the 'plane' that Tool detects, thus avoiding the error.

Dead Giveaways to Look Out For

- *Double Vertices:*

In the region of the error geometry, check there are not two or more vertices occupying the same space. You can check this by selecting a vertex by dragging, going into the edit mesh dialog, and seeing how many vertices are selected. If there is more than 1, in the edit mesh dialog, go to weld. Enter a threshold for welding (this depends on distance between the vertices, if its zero, leave the threshold at 0.1), & hit weld. Multiple pairs of vertices can be welded at once by selecting a whole area of vertices, and setting the threshold low (eg: 0.1) so that only vertices within that threshold distance are welded together.

- *Double Faces:*

Similar to double vertices. Check there are not two faces occupying the same area using the counter in the edit mesh dialog. if there are, delete one, or both, and repair any resulting holes.

- *Intersecting/Overlapping Faces:*

If faces intersect or overlap because they arn't connected to where they should be, they must be deleted and any holes repaired.

STL Check

Finally, selecting all the faces on your problem object and applying an STL Check Modifier (in Modifier list) allows you to check for a number of errors at once, such as open edges, double faces, spikes (overlapping faces) etc. Select the errors you want to check for and hit 'check'. The errors will be highlighted as before. This is often a good check to perform before attempting to compile, although, be aware that open edge errors that STL Check picks up will be fine by Tool, as STL check takes into account the fact that you have different objects. If in doubt, compile it, and import the WRL file.

Have fun fixing your geometry!!!

Keep in mind also that there is an open edge selection tool in 3ds Max. Go into edge selection mode, and below the weld toolbar on the side, there is a button that says select open edges. Click that, then move all the open edges upward so you can see where they are. Of course, you will get the obvious ones where you have attached things to your

skybox, but sometimes you can find other errors, things you forgot about etc. Undo the move, then fix the errors.

GhostPlasmaStickinessEffect

[Original Ghost Plasma Sticky Thread](#)

If anyone wants the Ghost's plasma stickiness effect, they have to add it themselves for their maps. Only Gearbox can readd it to the default maps with a patch, which they probably won't ever do.

(Why it's useful.)

It's simple.

1. Open Guerilla.
2. Open the file `\tags\vehicles\ghost\ghost bolt.damage_effect`
3. Scroll down to the "Damage" section.
4. Change the following:

Stun - 1

Maximum Stun - 1

Stun Time - 0.15 [seconds]

(Values as shown in the Vehicles\Ghost\Ghost Bolt section of a HaloPC level viewed with Halo Map Tools.)

5. Save.

That's it. The next time you compile a level that has a Ghost, that Ghost's plasma is HaloPC sticky.

AddStringsWithTool

Strings

Author: SteelixB

String Lists. No need to hex edit things, heres exactly how make them:

1. Open up WordPad
2. Type your first entry for your list
3. Either on the same line as that entry, or on the next line type '###END-STRING###'
4. Repeat the last 2 steps for your the rest of the entries for the list
5. You should have something like this:

```
You picked up a flamethrower
###END-STRING###
You picked up %d canisters for flamethrower
###END-STRING###
```

6. Save the file as either a unicode or an ansi txt file, save it in a folder in your data folder in the main halo ce directory
7. Depending on the format you saved it as, run one of the following commands in tool:

```
Tool Strings <directory of where you saved your list>
Tool Unicode-Strings <directory>
```

You should now have your string list tags created in the tags folder in your main halo ce directory.

Note that the list must also end on a new line, so that tool doesnt miss off the last string, and also don't place everything on the same line, otherwise instead of tool removing the new line characters it will remove the first 2 characters of every string after the first. Hope this helps! (And i discovered it myself)

-SteelixB
