Table of Contents

**Not started yet*
^Some information
#New info for version 11

Introduction

A: License

What this means is if you want to copy, upload, change, or do just about anything else with this tutorial, just give me credit near the top. I'd appreciate it.

B: Information

This tutorial is starting to include other things besides Photoshop and Dragon. Great! The more information it includes, the better. Eventually this will be covering alternative methods of creating custom maps, as well as other useful information for all aspects of world building.

C: Contact

If you'd like to contribute, I would be most grateful. Any contributions will be given full credit. If you have any questions or comments regarding this (or anything related, for that matter) you can contact me via several ways.
Email: ryandor@ryandor.com
ICQ: 2006722
IRC: irc.stratics.com in channel #uox
Website: http://www.ryandor.com

1. Getting Started and requirements

This tutorial is aimed at using Photoshop 5.5. That's what I have. If you use Paint Shop Pro or anything else, then you'll have to find the equivalent commands yourself. Sorry. Additionally, I assume you have at least some basic knowledge of how Windows and Photoshop work. This is not a "dummies" guide. However, this is set up as a step-by-step walk through. If you have questions on a specific step, feel free to contact me. Comments will be added as necessary.

Things to Note:

Color Table

Each colored square represents the tile type that it will paint. The number inside each square is the altitude (Z) of that tile type. If you draw with "Grass 10" then that spot will be grass tiles at 20 Z. When I want to refer to a particular terrain type, I will refer to it as "<Type> <altitudes>". Example: "Grass 20"

Commands

When I need to refer to commands in the program, I will use a **bold font**.
Example: "**File ✍ Open…"**

Tutorial Kit

I will be making a small download available that includes some of the things I use on a regular basis. These are custom brushes and actions for use in Photoshop.

Open Photoshop and create a new document. Set the dimensions to 6144 x 4096 pixels, resolution to 72 pixel/inch, mode to RGB, and white background.
Once created, change the document to Indexed Color:
"Image ✍ Mode ✍ Indexed Color…"

Set the Palette to Custom. It will open a new window for picking colors. Hit Load, and browse to the /dragon/addon directory. If you are using the original Dragon distribution, then the file you want to open is "COLORTABLE.ACO", however I have released a modified one, which has additional colors with which to create maps with. If you use mine it is called "DRAGON-MOD.ACT". (You'll need to change the "file type" in the drop down list box to open it) Once opened you'll see all the colors show up in the window. Hit OK twice, wait for Photoshop to convert the new document.

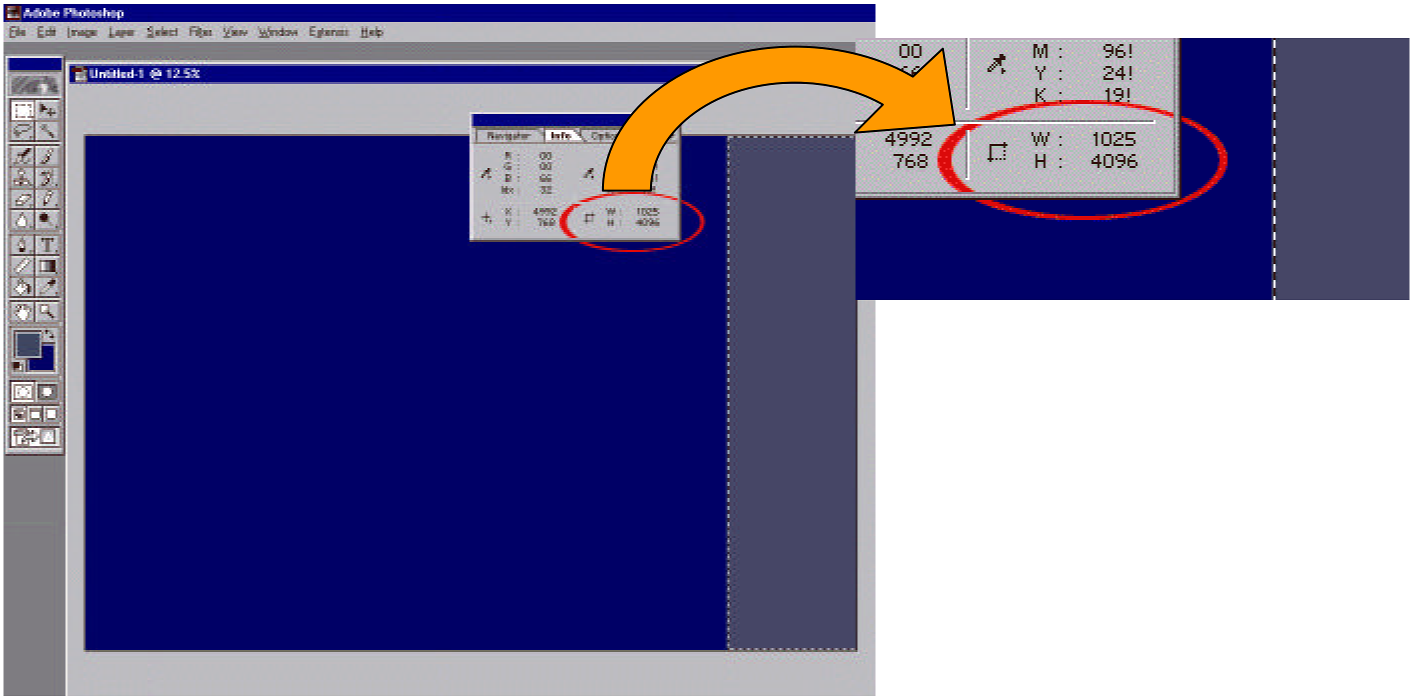
Next, open the file "colortable_eng.bmp" (or colortable_eng-mod.bmp if you are using mine) and place it next to you new blank map. You can then use the color picker tool to select the color you want based on what you are drawing. The color table is laid out by what kind of terrain you want. For example, look at the first row "grass". The whole row in this case is for grass. Each square represents the altitude (Z) of the grass.

This is all you need to get started with creating a new map from scratch. Pick a color, pick a brush, and start drawing. I recommend doing a flood fill with "water -5" to start. Then using grass to paint the "continents" and "islands".

When you are done (or need to do other things) make sure you "Save a copy" and save it as a bitmap (.bmp). The file size should show up as 24,578KB in Windows.

2. Basic Layout

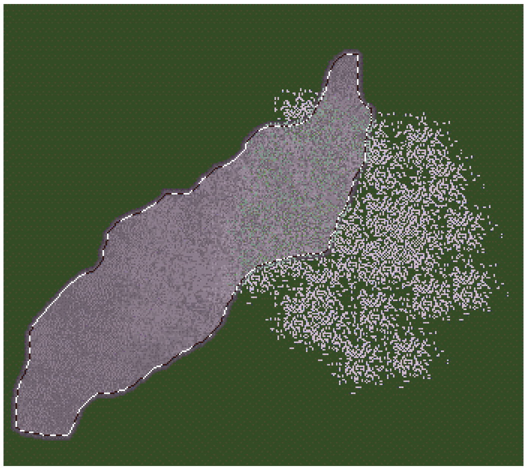
My first recommendation is to flood fill the whole map with "Water -5". If you plan on having dungeons, fill the right side (X>5119) with "Black 0" to make sure you don't put island there. The correct size of the flood fill for the dungeons area is illustrated below:



3. Basic Terrains

A: Grass

Now that you have the file for the map created, it's time to start drawing. For the first island you draw I recommend "Grass 0" to get the basic shape down. Creating a "natural looking" island is beyond this tutorial, so use your best judgment on what *you* think looks good. Pretty much what you draw is what you will get in the map.



Make a hill: pick the color "Grass 5". Draw an area on the island you made. You now have a small hill. Pick "Grass 10" and paint a smaller area in the same spot. Your hill just got higher. I think you get the picture. Grass is pretty basic.

B: Forest

Same as grass, except pick "Forest 0", etc.

C: Swamp

Swamp is considered (by me) a basic terrain type, but there is one simple rule. You must have swamp surrounded by grass. Dragon is unable to place transition tiles for swamp to anything except grass. The reason for this is there are not any other transition types available for swamp. Make sure you place at least 2 tiles of grass around the swamp, as this is needed for Dragon to properly calculate the transition.

D: Desert

Same as grass.

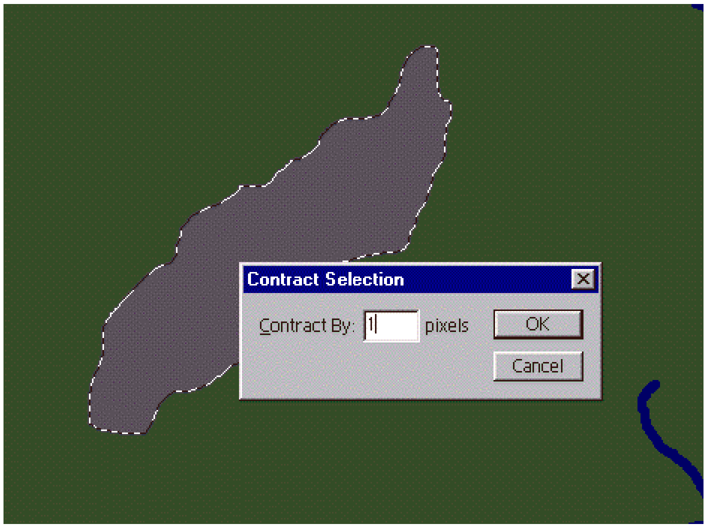
E: Jungle

Same as grass.

4: Advanced Terrains

A: Mountains

Mountains can be tricky but once you get the technique down, it will go quickly.
The first thing I recommend is to set up a couple of actions in Photoshop. These are essentially macros that can perform repetitive tasks for you with a single key press. I will not explain how set them up, but I will be providing a couple that I use in the Tutorial Kit. This method is what I use to get “OSI type” mountains. I think they look pretty good.
First, using “Rock 20”, draw the mountains on the map. Select the “Magic Wand” tool from the Toolbar. Set the Magic Wand tool options to “Contiguous” and set the Tolerance to 0. Select the “rock” you just painted. Now, either hit F3 (for the action I set up for “Contract Selection” or from the menus: “**Select ✍ Modify ✍ Contract…”** Set the contraction to 1.



Contract Selection

Contract By: 1 pixels

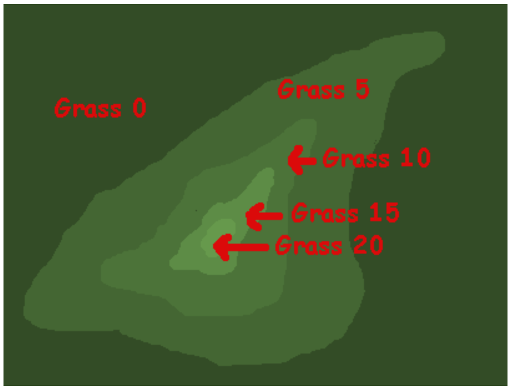
The selection will (obviously) contract by one pixel around the border. Use the color picker to select “Rock 30”. Select the “Fill” tool and set the Tolerance to 0. Now, fill the selection. Contract the selection by 1 again. Pick “Rock 50” and fill again. Contract the selection by 1 yet again. Pick “Rock 60” and fill again. Now, pick “Rock 65” but pick a brush with a random pattern and paint randomly in the selection. Pick “Rock 70” and repeat.

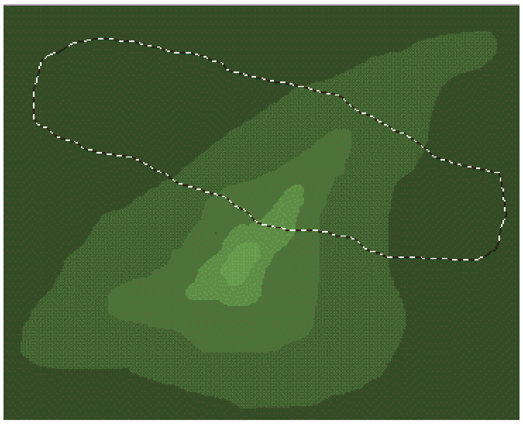
Do this as many times as you like, but I would not recommend over 80 in small spots. (OSI mountains are generally in the 60 – 70 range). You will notice as you paint the higher numbers, it will start to take on the randomness of mountains. Once you get the hang of this, turn off the “Contiguous” mode on the magic wand and fill tools, and you can do the entire map at once.

Mixed terrain:

This is another technique that is difficult to master as well, but I consider this one of the hardest things to do as well. It requires paying attention to what you are doing and remembering (or writing down) what's where.

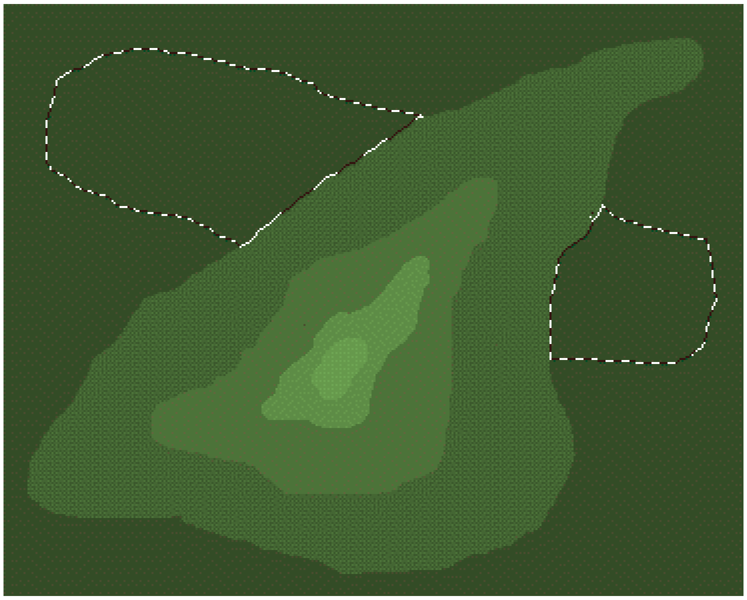
This can apply to many mixes, but I will do a grass hill with a line of forest going over the hill.
First the hill. Draw the hill using grass with different elevations. Remember the elevations you use and in which order. I usually stick to 5,10,15,20,etc, but you can use what ever elevations you want. Here's my hill:

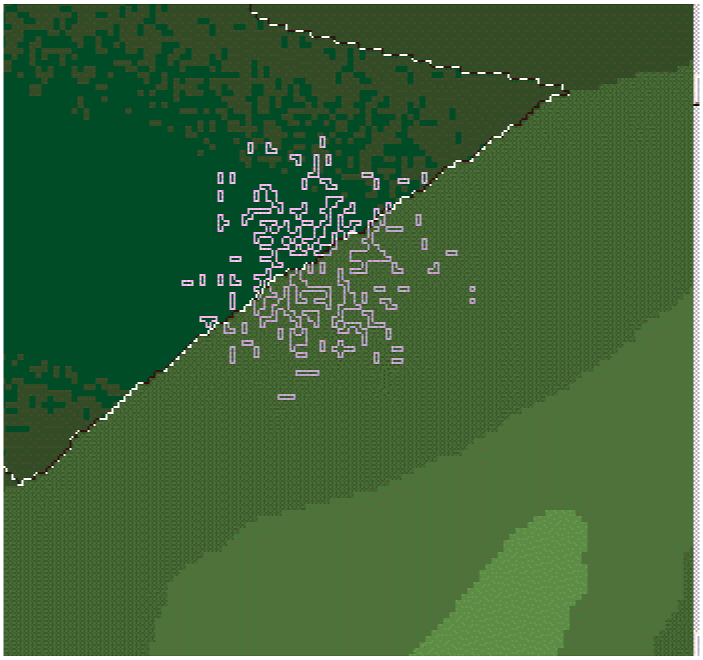




Now, using the lasso tool, select the area where you want the forest to be:

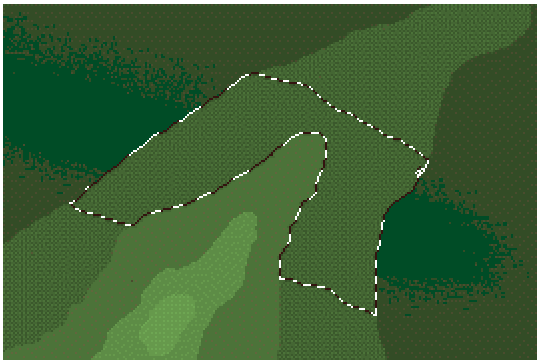
Now, save the selection with "**Select ✍ Save Selection…"** Just save it to a new document. It will make a new document with black and white representation of the selection. Now, with the Magic Wand, subtract the areas that are part of the "hill". (Use ALT-click)

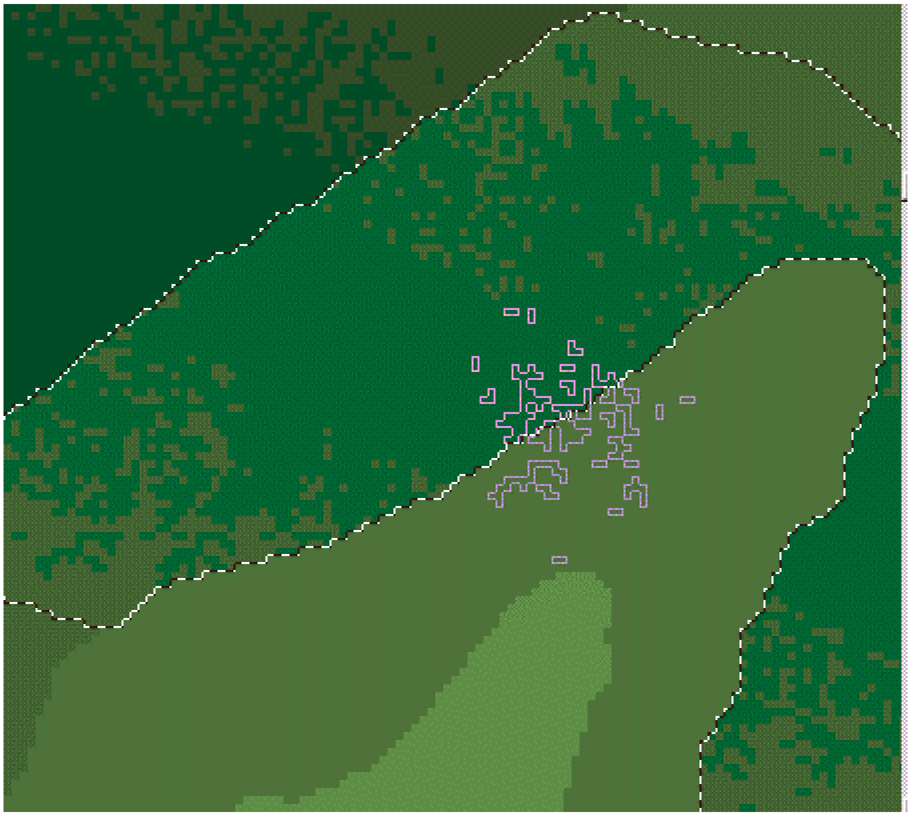




Using the “Forest 0” color and a random brush, paint in the areas of forest (Close-up of area, since “Grass 0” and “Forest 0” are hard to distinguish)

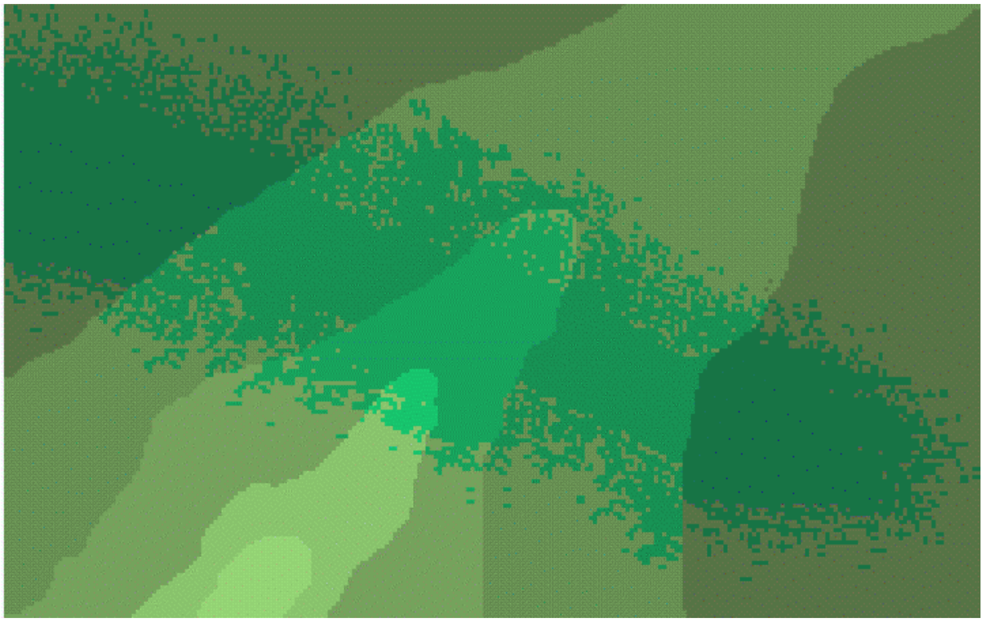
Once you have “0” done, time to move on to “5”. Deselect everything and load the selection you saved previously: **“Select ✍ Load Selection…”**
Using the Magic Wand with Contiguous unchecked and Tolerance 0, subtract the “Grass 0” and “Forest 0” and all altitudes above 5:





Select “Forest 5” and using a random brush, paint in the forest for this altitude, matching your “breaking points” to align with the previous altitude. (Close-up again)

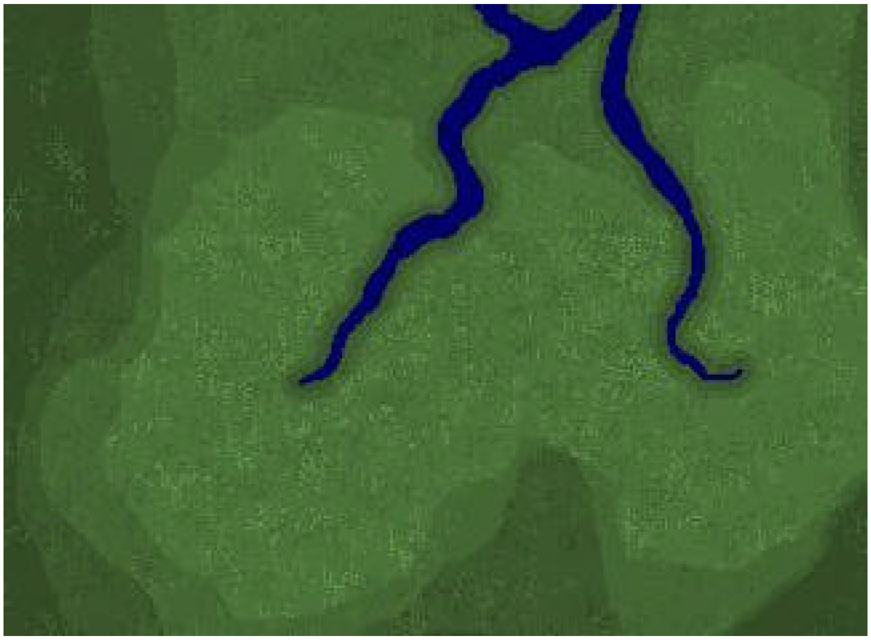
Do the same steps for altitudes 10 and 15 (the selection doesn't reach the altitude of 20 at the top of the hill in this case). When finished it'll look something like this: (enhanced to show colors better)



That's it. This technique can be applied to other terrains as well. For example: Grass/Jungle, Forest/Jungle, Snow/Grass, etc.

C: Rivers in Terrain

Using "Water –5" draw the river where you want it. Then using the lasso tool, outline where you want the terrain to be. Deselect the water. Contract selection by at least 2 (to avoid problems with translation tiles for coast), and then fill with desired terrain (for example: if the river is in "grass 0", fill with "grass 5"). Contract the selection again and fill with the next terrain ("Grass 10" in this case). Repeat as necessary. Here is an example of this:

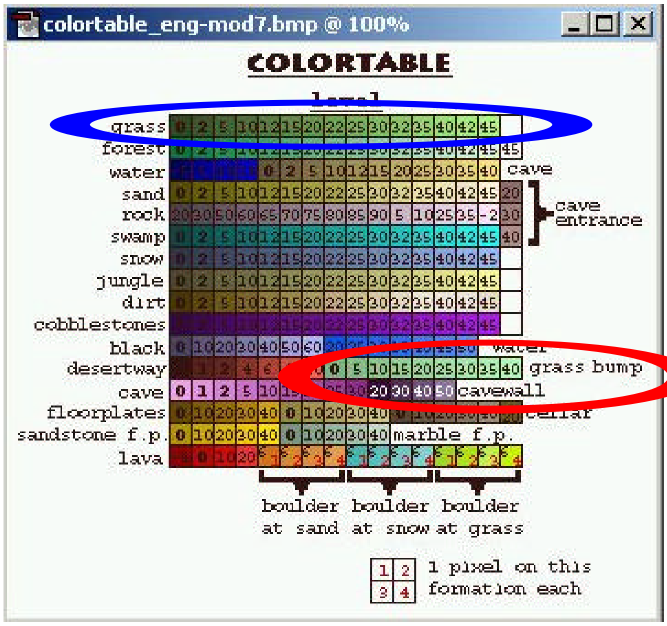


This is a river going through hills with random elevation spots and small forest areas. As you can see, I created the hill to make the rivers appear to "cut" into the hillside.
Using the magic wand and lasso with contract achieves this effect. Then a random brush to get the random elevation.

D: Grass "cliffs"

This section is for those using my Mod available for Dragon. While the name "cliffs" may not be a true representation of the tile type, it's what I will refer to it in this tutorial. The word "bump" is used in the Dragon mod, and refers to the same thing.

First the colors:



colortable_eng-mod7.bmp @ 100%

COLORTABLE

level

grass	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
forest	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45	45	
water	[illegible]	[illegible]	[illegible]	[illegible]	0	2	5	10	12	15	20	25	30	35	40		cave
sand	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45	20	cave entrance
rock	20	30	50	60	65	70	75	80	85	90	5	10	25	35	-2	30	
swamp	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45	40	
snow	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
jungle	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
dirt	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
cobblestones	[illegible]	[illegible]	[illegible]	[illegible]	12	15	20	22	25	30	32	35	40	42	45		
black	0	10	20	30	40	50	60	20	25	[illegible]	[illegible]	[illegible]	45	50			water
desertway	[illegible]	1	2	4	6	[illegible]	[illegible]	0	5	10	15	20	25	30	35	40	grass bump
cave	0	1	2	5	10	15	[illegible]	[illegible]	[illegible]	20	30	40	50				cavewall
floorplates	0	10	20	30	40	0	10	20	30	40	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	cellar
sandstone f.p.	0	10	20	30	40	0	10	20	30	40							marble f.p.
lava	[illegible]	0	10	20	1	2	3	4	1	2	3	4	1	2	3	4	

boulder at sand boulder at snow boulder at grass

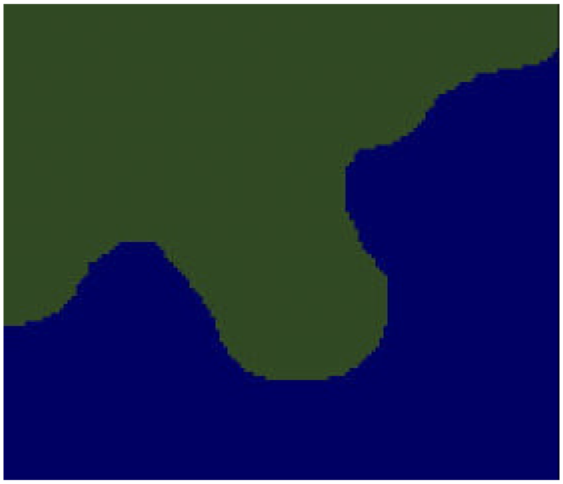
1	2
3	4

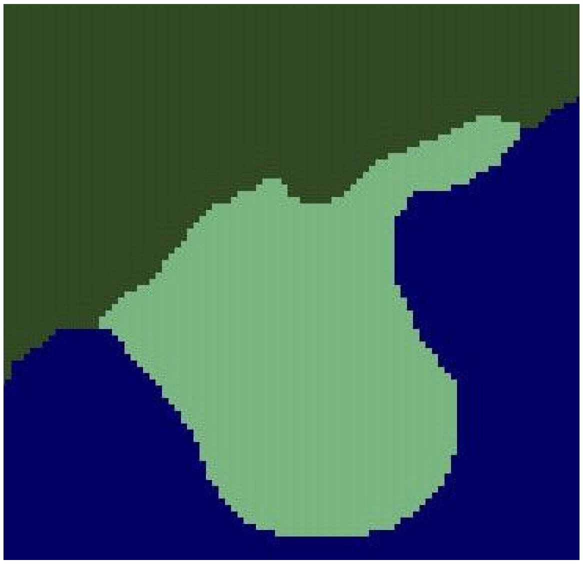
1 pixel on this formation each

The colors circled in red are the ones to be used for this effect. In order for them to work correctly, you must always use a lower number (z value) than an adjacent grass color (circled in blue). For transitions to other tile types and colors, the normal rules apply as for grass.

Putting it to use (an example):

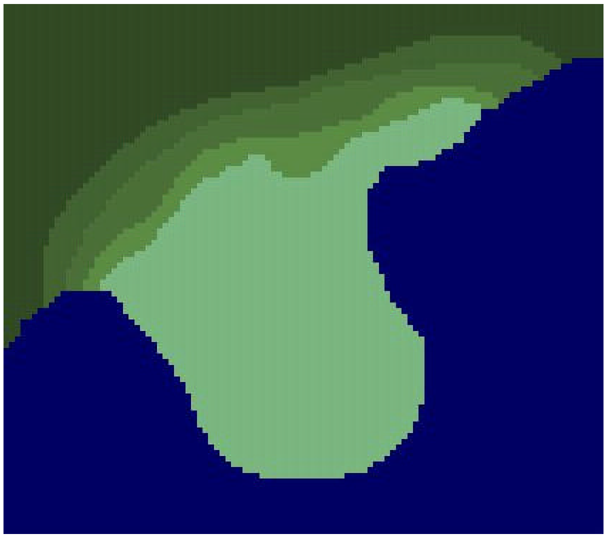
For this example I want a "low area" along the coast of this map I'm making:

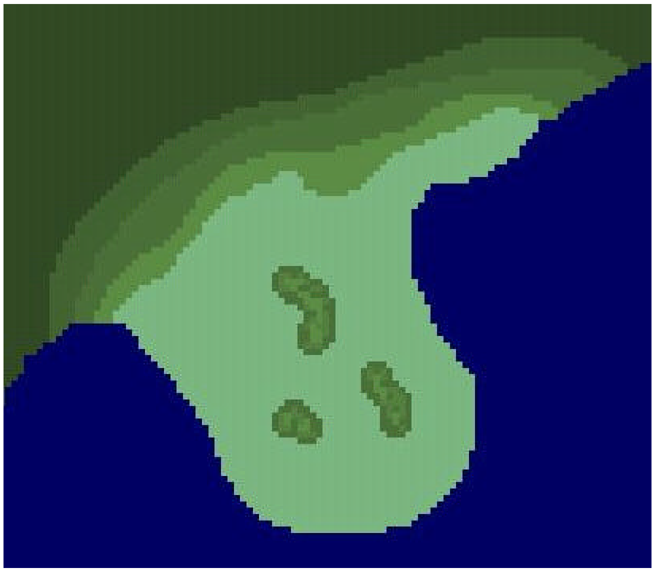




First, I'll add the new color for the low area using "grassbump 0"

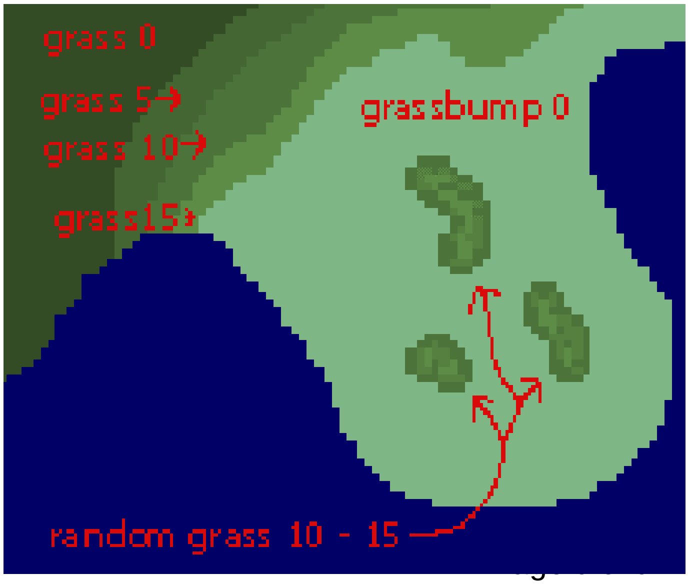
Now, for the important part. Since the original grass needs to be at a higher z (altitude) than the grassbump, I need to make the original grass higher. I'll put a gradient for the hill to make it look good.



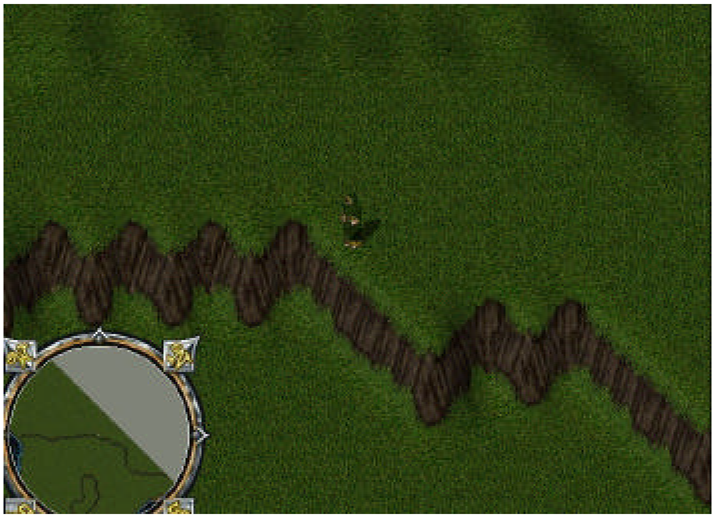


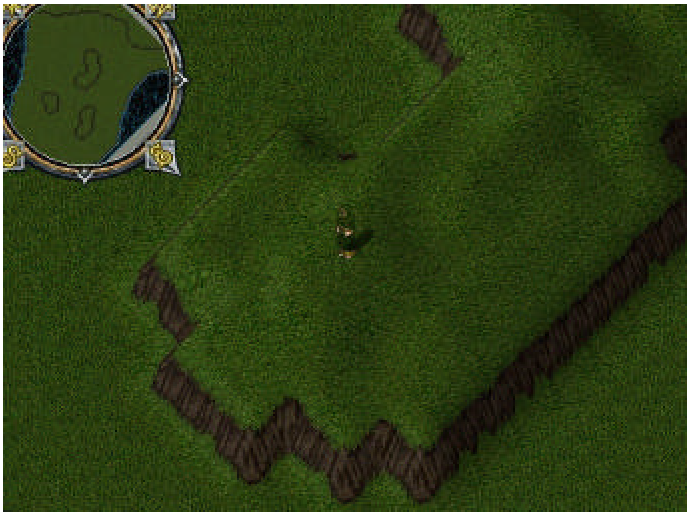
To finish it off, I'll add some little hills with in the low area, to give something to look at in game.

Here's the final map section with labels indicating tile type.



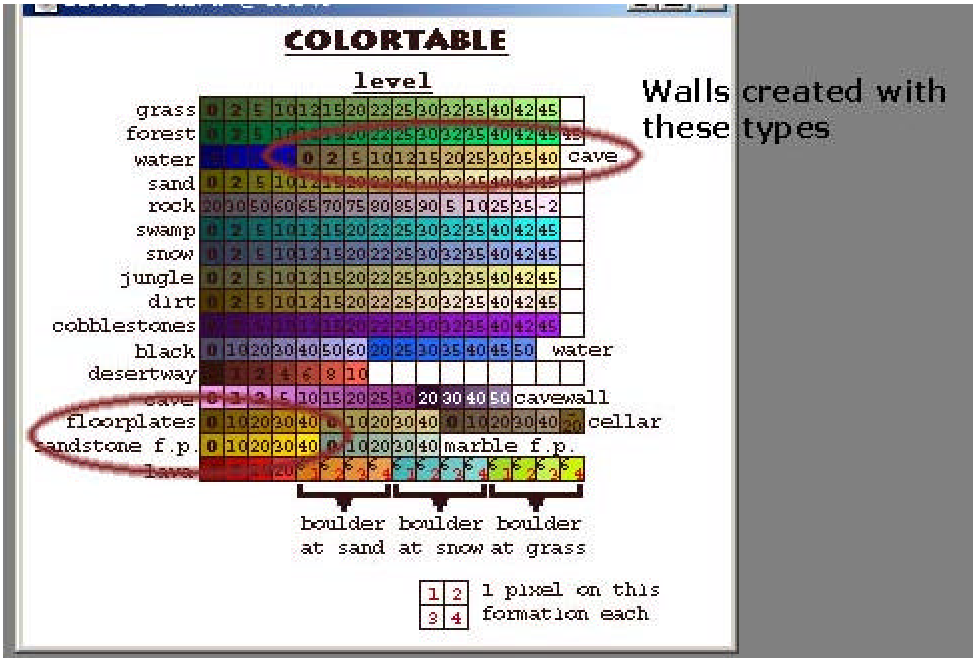
Here is what it will look like in game:





5. Dungeons

The creation of dungeons is a bit different. Most of a dungeon is made with statics, so the only major work in Photoshop will be the floors for the passageways and the rooms. It helps to draw out the basic design on paper first so you can remember where the walls are going to be. I have found that when creating dungeons that will have stairs, it is best to leave a one or two tile gap where you will have the stairs. Use the appropriate “black” level to fill the gap. Some of the tile types in Dragon will create the outside walls of the dungeon.



COLORTABLE

	level																
grass	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
forest	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45	45	
water	[illegible]	[illegible]	[illegible]	[illegible]	0	2	5	10	12	15	20	25	30	35	40	cave	
sand	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
rock	20	30	50	60	65	70	75	80	85	90	5	10	25	35	-2		
swamp	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
snow	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
jungle	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
dirt	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
cobblestones	0	2	5	10	12	15	20	22	25	30	32	35	40	42	45		
black	0	10	20	30	40	50	60	20	25	30	35	40	45	50	water		
desertway	0	1	2	4	6	8	10										
cave	0	1	2	5	10	15	20	25	30	20	30	40	50	cavewall			
floorplates	0	10	20	30	40	0	10	20	30	40	0	10	20	30	40	20	cellar
sandstone f.p.	0	10	20	30	40	0	10	20	30	40	marble f.p.						
lava	[illegible]	[illegible]	[illegible]	20	1	2	3	4	1	2	3	4	1	2	3	4	
					boulder at sand				boulder at snow				boulder at grass				

1	2
3	4

1 pixel on this formation each

Walls created with these types

If you use “cave” you will get cave walls already filled in. Also, if you use floorplates, you’ll get dungeon walls.

6. Towns and Roads

Creating towns is mainly a static creation as well, but the floors and roads are better off as part of the map. The reasoning is once you have a completed map with many towns, you'll want to keep the static files size as small as possible. If the floors are included as statics, then the size will go up correspondingly. The map0.mul file size does not change regardless of what is in it. (This, of course does not take into consideration patching aspects.)

One feature of floor tiles in the map is that they do not stretch like the landscape tiles, so you can include them at any altitude with out worrying about surrounding tiles at different altitudes.

Don't over do the "custom build from scratch" buildings. You'll get burned out real quick. Use some OSI houses with minor tweaks for some buildings. It'll go faster and you wont get burned out. Just keep the style of houses similar. I think for every large (10 x 10 in size) building you do, you should have about 3 or 4 smaller regular houses. Use the templates in Xuri's Worldbuilder to add generic houses. Go through the OSI map and extract houses you like.

If you use an OSI house, just change the furniture and move the doors and windows and maybe add an extra wall inside to break it up, or take out a wall. When you place houses that are 2 and 3 stories high, don't place them too close together; otherwise your players will have lots of hiding places (unless you want that; great effect for an alley).

7. Underground: Caves and Basements

Starting with Dragon Mod 7, I've added the ability to create "cave entrances" into a bmp for Dragon to convert for you. While this may sound great, it's mainly for reference use only, as you still have to go into WorldForge and tweak the entrance a little. Dragon does have limitations, and this is related to the main one; all transitions must have 2 tiles adjacent to create a smooth transition. Cave entrances technically have three tiles adjacent tiles, where the cave entrance meets the grass (or forest, etc.) Therefore, minor tweaking will be required for the "finish" effect of a cave entrance.

The original way I did cave entrances was with UOCopy or WorldForge, and this can actually be easier to a degree. The best way to get good looking entrances is to use WorldForge to find a cave entrance in the OSI map, and copy it to where you want it in your custom map. The main drawback to this method is you must do it after you are done compiling the map with Dragon. If later, you need to re-run Dragon for any reason, you'll have to re-copy the cave entrances back into your custom map.

8. Random Statics

Take advantage of DragonSP to create statics for Forests, Jungles, Swamps and Deserts. I would not use the Grass statics though as it may interfere with town building. The scripts are easy to edit and are well commented so take a look at them in the /Dragon/scripts/items directory. If there is a type of item you do not want, then comment it out (using "//" at the beginning of the line.) I've modified or used StormCrow's Modified scripts in my Dragon Mod. This allows for more options in the scripts.

9. How to use Dragon

First, when you run it, you'll have to wait 15 seconds. Hit OK.

The "BMP File" is for the bmp you created. Enter or browse the file name. Do not worry about the "Fix file". Leave it blank for now. An explanation of how it works is below.

The "MAP File" is what the BMP will be converted to. Make sure you have a map0.mul file in this path. Dragon cannot create a file from nothing. It needs to be able to overwrite the map0.mul. Make sure you are using a copy in case something goes wrong. I highly recommend that both the bmp file and the map file be located in the same directory as Dragon. There have been reports of Dragon not working correctly when they are elsewhere.

Hit the Config button. A new window will open.

Remove Statics Files: You want this to be checked. This controls the scripts that Dragon creates for static items. If you leave it checked, the next time you run Dragon, you will get either duplicate items (if the map is the same) or items in the wrong areas (if it's a new map).

HighPriority: This will make Dragon run just a little bit faster. When checked, the progress bar will not update and you can usually shave a few seconds off the compile time. On a 1Ghz computer with 512MB RAM, it is not noticeable. I have had occasional lockups with Dragon running in High Priority mode. Unless you've got a slow computer, leave it off. If you do use it, don't use other applications at the same time. Additionally, the amount of RAM you have can greatly affect the compile time. The more RAM you have, the better it will be. And, of course, the speed of your computer will affect it as well. I have compiled maps on a PII 233 with 144MB RAM. It took about 40 minutes. I currently use an AMD 1Ghz with 512 DDR RAM and it takes about 4 minutes.

Sphere Statics or UOX Statics: This affects the scripts that Dragon creates. Check Sphere Statics. I've been told that UOX Statics does not work correctly. The scripts that Dragon generates is usually frozen with DragonSP, so no need to worry about what server you use. Once the statics are frozen, they will work on any UO emulator.

Hit OK to exit the Config screen.

Once you've gotten everything set up, hit the Start button in Dragon. Depending on your computer's speed and resources, it can take anywhere from 3 minutes to up to an hour. Be patient. It will inform you of its progress as it works.

The Fix File: This is used for overriding the tiles in the map0.mul. You can use it if you've already edited the map with WorldForge but need to recompile the map with Dragon. You simply include a list of all tiles you want to be placed in the map instead of Dragon generating a tile. You can find an example fix file in the /Dragon/addon directory. Put the x, y, z, and item id in this file and it will place that item at that location in the map.

10. How to use DragonSP

After you've run Dragon, you may notice it tells you to run DragonSP to freeze statics. That's how you do it.

When run, DragonSP will need the paths to your statics. (Use a copy of them just in case)

On the left side are all the types of statics that you can include. This list is a set of scripts that Dragon generates based on the last map you created with it. Select the types you want to include either with the mouse or the buttons underneath the list (don't worry about the search button). Pick both Waterx2.scp and 2xWater.scp to include the coastlines. Other types should be obvious. Hit start. At one point it will ask to "Clear Current Statics?" Answer Yes. This will get rid of all the towns, forests, etc that are in the static files (staidx0.mul and statics0.mul). It will then put in the statics that you need based on the map. That's it really. If you don't answer yes to the "Clear Statics" question, it will leave the current statics in the file and add the new stuff in. This is useful for when you do the coastlines on the first run, and then you want to do other statics on a second run (forest, grass, desert, etc).

DragonSP only freezes coastlines, trees rocks, etc that are generated by the scripts that Dragon creates when converting the map. To freeze houses and buildings, you have to use a program that will freeze stuff in the items.wsc file. WorldForge, MulTool, etc do this.

11. Customizing Dragon

A: Color Table

To edit the color table to include other types of tiles, you need to edit maptrans.txt and groups.txt. Both of these are in the /Dragon/scripts directory. Open both files side by side and you can see how they are laid out. The color table is a 64 x 64 grid, with each color belonging to a group. Dragon doesn't actually care about the colors that you use. It uses the index number of where the color is. However, practically speaking, it makes sense to match a color in the colortable.bmp (in the /Dragon/addon directory) so that it is easier to visualize what you are drawing. Also, if you edit the maptrans and groups files, you should also edit the colortable.bmp so you can use the color picker to get the color easily. Once you've edited these files, you need to make sure that the actual color table of your bmp matches what you have in the txt files. In Photoshop: **"Image ✍ Mode ✍ Color Table…"** will get you to the right spot. I recommend saving the color table for future use so you can load it later. If all this sounds a bit complicated, the easier way to do this is to get my Dragon mod, which already has many additional colors, included.

B: Item Scripts

In the /Dragon/scripts/items directory are the files that control what random statics are created. These are well commented and are easy to edit. See the above section on Random Statics. For transitions, you can edit how the tiles are interpreted by editing the files in the /Dragon/scripts directory. See Appendix B for more information

12: Creating a map without Dragon

Here is a set of programs that you can use to create a custom map from scratch. It's a bit trickier than using Dragon, but can be done. Here is the list of programs you'll need:

UOConvert: used to convert a bmp made in MSPaint (or other paint program, but it has a "color table" designed for MSPaint) to map0.mul
UOCoast: Creates the water tiles for all the coast lines
Transition: Used to make transitions like grass/forest, snow/grass, etc
ItemPlacer: Used to place random trees, rocks, etc
Cartograph: Used to make the multimap.rle file for custom "in-game" maps (cartography skill)

All of these programs (and others) can be found at
http://www.eikon.tum.de/~isp/sphere/doc_index.html

WorldForge: Used to freeze the items that these programs create
WorldForge is available at www.uoxdev.com or at other sites

Additionally, UOConvert can "patch" a new area in the map0.mul if you've gone too far past the point of recompiling with Dragon.

Make sure you read the documentation that comes with these programs! It has more excellent information on how to use them.

A: UOConvert

UOConvert is a program that you can use to either create a new map with or to import a new section of map into an existing map0.mul. This can be useful if you've already created a map with Dragon but have either modified the map0 with WorldForge (and would loose your modifications) or just don't want to recompile the map.

I'll explain how to import a new section of map using UOConvert.

First thing to do is create a small map in MSPaint. I wouldn't recommend to large a bmp, since MSPaint is not designed for handling large files. The easiest way to make a map piece is to just use the colors that are there already. Use light green for grass, dark green for forest, etc. If there are areas that you do not want to overwrite, simply leave those areas blank (don't draw). This way it is possible to import areas that are not square. Once you have made a map (and saved it), open UOConvert.

When you run the program it will open up an empty window this some buttons and controls. Either use "**File ✍ Open BMP"** or use the buttons to open the bmp you just made.

"File ✍ Open Translation" and open the file "test2.tuo" which should be in the UOConvert directory. This file is what the program uses to convert the colors you draw into the appropriate tiles on the map. Explanation of the controls:

Drop down list (initially it should say "Color 00"): This is the current color you are working with. All other controls are its properties.

Color: This is the hex id of the color for the current tile IDs.

Min Tile and Max Tile: These are the tile IDs that will be drawn based on the current color. Use InsideUO to see what the tiles are. For example, 0003 and 0006 represent the grass tiles. When the program converts and places the map piece in the map0.mul, any one of these tile IDs will be randomly picked and inserted into the map, based on the color.

Min Alt and Max Alt: This controls at which altitude (z value) the tiles will be placed. This is used to create a random landscape, as whatever values you choose will be randomly picked within this range for the tile.

Color Dropper: This is used for picking the color that will represent what tile the color will use. How it works: If a color is already defined, it will go to that color in the list, and you can change its properties (same as selecting the color from the drop down list). If the color is not defined it will overwrite the currently selected color based on the drop down list. I would recommend careful examination of the currently defined colors before using this. By default, the most common colors are included in the translation table.

Delete Entry: Deletes the current color in the drop down list.

Path to map0.mul: Put the full path to the map0.mul you want the bmp to be imported into. Make sure you are using a copy, in case it doesn't import how or where you like.

Top-left X and Top-left Y: This is where in the map you want the bmp to be placed when it converts. Remember, if you have white space (non-defined) areas in your bmp, it will not change the map in these areas. If the area has white space in the upper left corner, it will "offset" the bmp that you import.

"**File ✍ Convert BMP"** will commit your bmp to the map0.mul. Use InsideUO to see the changes it made.

B: UOCoast

This is used for placing all the water coast tiles into the map. When you run it, simply place the full paths to the map0.mul and items.wsc files you want to use. If you want to create coastlines for the entire map, then leave all the x y boxes alone. Otherwise put the area you want to work with into these boxes. Select the options you want in the check boxes, and hit "Create Coast" Once it is done, you need to freeze the items into the static files with WorldForge or other appropriate program.

C: Transition

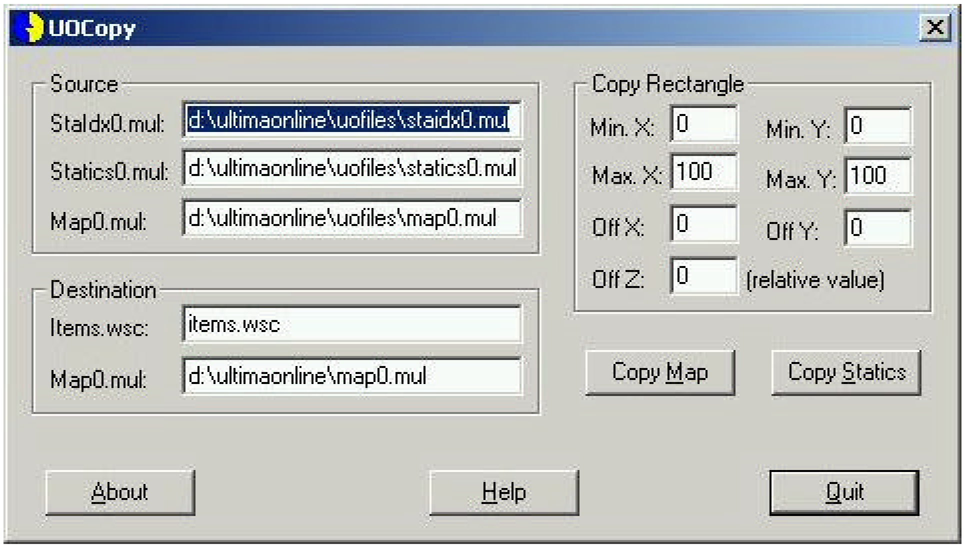
Coming Soon

D: ItemPlacer

Coming Soon

E: UOCopy

UOCopy is used to copy one part of a map to another. For instance, you want to use the OSI dungeons with your custom map. When you open UOCopy, You will see this:



Source

StaIdx0.mul: d:\ultimaonline\uofiles\staidx0.mul
Statics0.mul: d:\ultimaonline\uofiles\statics0.mul
Map0.mul: d:\ultimaonline\uofiles\map0.mul

Destination

Items.wsc: items.wsc
Map0.mul: d:\ultimaonline\map0.mul

Copy Rectangle

Min. X: 0 Min. Y: 0
Max. X: 100 Max. Y: 100
Off X: 0 Off Y: 0
Off Z: 0 (relative value)

As always, make backups of maps when using tools like this.
Source: These are the paths to the files you are going to copy **from**. For a map copy, the map0.mul path must be valid. For a static item copy, the staidx0.mul and statics0.mul paths must be valid.
Destination: These are the paths to the files you are going to copy **to**. For a map copy, the map0.mul path must be valid. For a static item copy, the items.wsc path must be valid.
Copy Rectangle: Min. X, Min. Y, Max. X and Max. Y defines the rectangle in the source map that is going to be copied. The Off X and Off Y is the **absolute** offset from (0,0) of the copied rectangle in the destination map. The Off Z is the **relative** altitude offset (original altitude is 2, Off Z is -5 -> altitude of copied tile or item is -3).
Copy Map (Button): This button will start the map copy.
Copy Statics (Button): This button will start the static item copy. Copied items are **not** directly added to a destination staidx0/statics0 file, but a file called "items.wsc" (by default, you can change the name) is created which includes all the items that were copied. You can use this file either to freeze the items with the programs Worldforge or MulTool or import it directly into SphereServer with the "import" command.

F: Cartograph

This is a tool for creating the “in-game” map when using the cartography skill. If you made a bmp of the map for Dragon, the best way is to use this map, convert it to black and white, shrink it, and run it though this program. This is a simple DOS based program, and has a readme file for it.

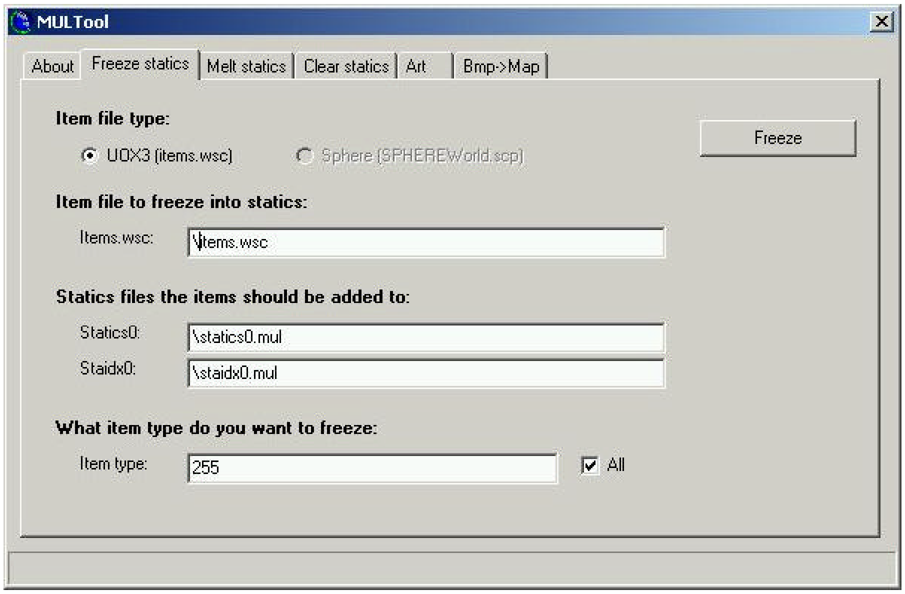
For best results, resize the bmp to 2560 x 2048, and change the color to 1-bit (black and white). **NOTE**: This program has been fixed recently. If you downloaded the program before Feb 15, 2002, re-download it.

G: WorldForge

By far the best reference for using Worldforge is at Xuri's site.
Please go here: http://homefree.sensewave.com/~s107293/wb/wforgetutorial.html

H: MulTool

MulTool can do a variety of things, but most useful for map making is Freezing and Clearing of Statics.



Item file type:

● UOX3 (items.wsc) ○ Sphere (SPHEREWorld.scp)

Item file to freeze into statics:

Items.wsc: \items.wsc

Statics files the items should be added to:

Statics0: \statics0.mul

Staidx0: \staidx0.mul

What item type do you want to freeze:

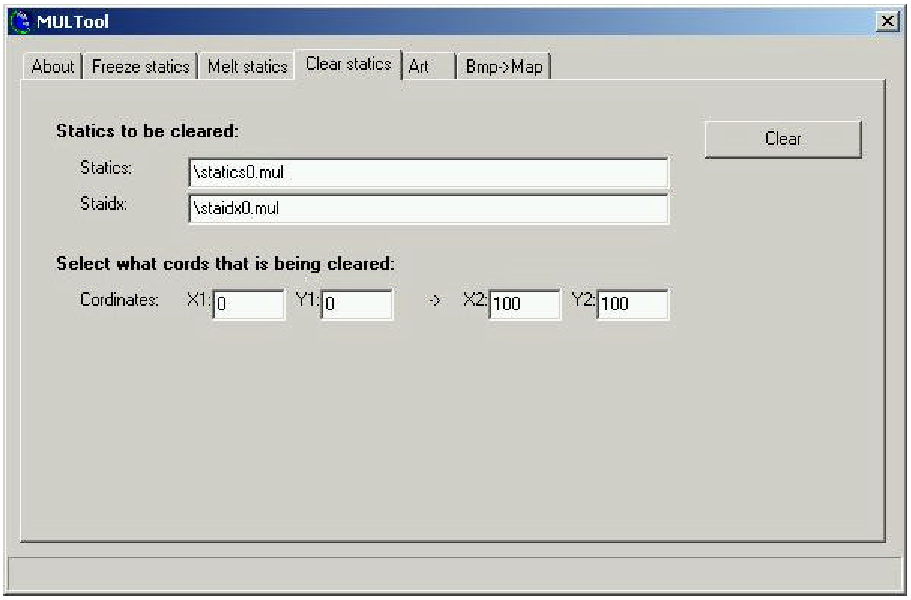
Item type: 255 ☑ All

Freezing:

This is fairly simple to use. Simply put the full paths in the appropriate fields and hit "Freeze". If the "All" checkbox is marked, then everything is Froze. If not, the only the item type specified will be froze.

Clearing:

Again, simple. Fill in the full paths of the statics, put in the coordinates you'd like to clear, and hit "Clear".
As always, backups are a good thing.



Statics to be cleared:

Statics: \statics0.mul

Staidx: \staidx0.mul

Select what cords that is being cleared:

Cordinates: X1: 0 Y1: 0 -> X2: 100 Y2: 100

Please note, these are the only parts of this program that work correctly with Win 98, ME, and XP. Not recommended for other tasks.

Appendix A: Links and Acknowledgements

Links

Dragon
http://www.uocm.de

Xuri’s Worldbuilder
http://homefree.sensewave.com/~s107293/wb/

UOCopy, UOConvert, UOCoast, Transition, ItemPlacer, Cartograph
http://www.eikon.tum.de/~isp/sphere/doc_index.html

Worldforge Tutorial
http://homefree.sensewave.com/~s107293/wb/wforgetutorial.html

Mr. Fixit’s Multool and UOGateway
http://uogateway.com

Acknowledgements

Stormcrow: parts of the Dragon and DragonSP sections
Kirre: conversations about town building
Malec: Conversation about freezing statics
Xuri: WorldBuilder and WorldForge Tutorial
Mr. Fixit: MulTool

Appendix B: Dragon Script Definitions

(Taken from www.uocm.de)

Group.txt

The file Group.txt defines groups for the individual colors in your painted BMP. A group is the definition for the range of colors. For example the group Grass (00) is all colors from 00 to 0e. This is used later with the Maptrans.

Maptrans.txt

The file Maptrans.txt contains the converting information that converts the pixel color to tiles. The first 2 characters contain the tile group number from Groups.txt. The second set of 3 characters is the altitude for this tile group. This is followed by a list of Tile Ids that this group will randomly select from to place in the map.

Betweentrans.txt

The file Betweentrans defines which file to use in creating tile transitions. For this file it is done from X to Y. The first two sets of numbers define Group X then Group Y from the Groups.txt file. This is followed by the file containing the transition table.
For example:
04 03 Black2Dun.txt
This tells when going from Black tiles (04) to Dungeon tiles (03) use Black2Dun.txt for the transition table.

Betweentrans2.txt

This file also defines which file to use for transition calculations. However this is for Group X to X, or the same type of tile. The first number then is the tile group followed by the file to use to transition it. Group X is defined as "tile A" and all other Tiles except group X "tile B" in its related transition table file.

Statbetweentrans.txt

In this file the Statics is defined. This is again a group X to group Y relationship like Betweentrans.txt but for static tiles.

Statbetweentrans2.txt

Similar again to a group X to group X relationship but for statics also.

Statbetweentrans3.txt

Similar again to a group X to group X relationship but for statics also.

Sample code for Transition Tables

If you open the file "grass2mountain.txt " you should find something like this:

AAAABBBB 0238 005 007

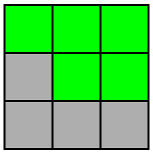
The first value (the As and Bs) describe the order of the Tiles A (grass) and B (mountain) around the Tile A found by the program (which is the center tile), beginning with the Tile in the northwest and then in a clockwise direction up to the west. For example:

1	2	3
8	A	4
7	6	5

"With Dragon - AAAABBBB (ignoring the center tile)”:

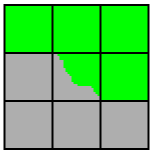
A	A	A
B	A	A
B	B	B

In our example (A - > grass - > green; B - > mountain - > grey):



If the program runs into this Tile configuration for transitions it would replace Tile A (the center tile). The second value in our above example (AAAABBBB 0238 005 007) - 0238 is the Tile ID that will replace this tile.

In order to find this ID, open " InsideUO " and open the Textures menu. Find the Tile ID that best represents what you are trying to accomplish and use its ID number. 0238 for our example:



The last two values in our example are height modifiers for our transition tile. These values take the original tile height and add a random number between the ones specified. For example in our scenario "005 007" if the original tile that is being replaced had a height of 10, the replacement tile would be set between 15 and 17. With "grass2mountain" this is quite important, so that the mountains flow gracefully into grass. In addition it is best to vary the altitude just a little. i.e. you take our transition tile, it is very close because the mountain is a larger height then the grass tiles, which are situated somewhat away from the mountain - so you need the transition tile to be between those heights so it flows well. In some cases though you do not have to adapt the height at all (ie. with edges of Dungeon) - simply use the values to " 0 0 "... for the height modifiers for it to have no change

In the same way you go through all the AABABBAA possibilities, looking for which Tile to use with InsideUO entering its ID and what height modifiers you may want.

Thanks to Wildmieze for the generation of this write-up.
And Amitiu for the English touch-up.

Note:
With Statics the program gets the highest Z from the surrounding Tiles looking 2 and 3 up in the source group and then adds the Z level.

Appendix C: Notes to include in the text eventually

*Don’t use lava, it don’t seem to work right.
*The "purple" cave (with 1,2,3,5,10,etc) is the same as the first floorplates, will be changing this soon.
*Copy and paste the color table into you new map (like down in the lower right corner) so when you convert the map, you can see exactly what each tile does.
*Unevenly.txt: how to use. Comment out with “//” to not use.
*Change of the overall structure of tutorial. Sections by apps, since this is covering more than just Photoshop and Dragon, but all aspect of mapmaking (eventually)
*UOSir
*UOSP
*Patching map and statics
*DB2M
*UOX Terrain Tool