

## ISOM Brushing

Note1: Everything mentioned here is based on a 'normal' Starcraft map, i.e. as if the map was made in Staredit. This mainly concerns the validity of the ISOM section and standard map dimensions.

Note2: I will be using windows datatypes such as WORD (unsigned short) since their names usually are shorter.

Note3: I will limit myself to examples from a 64x64 Jungle World map, with only Dirt and High Dirt used. The concepts should apply to other terrain types and tilesets, too, but I haven't tested this yet. Also, I don't claim this to really work - though I strongly believe it does. I didn't finish my coding yet, so I don't really know.

The ISOM section consists of 'IsomRects'; one IsomRect for every two tiles. An IsomRect is a structure of 4 WORDs; left, top, right, bottom, in that order. In the ISOM Section, the first IsomRect specifies the first two tiles (top left on a map). The right tile always has the index of the left tile + 16 (matching pairs).

001A	0110	0114	015C	015E	0158	01B0	01B2
0112	011C	011A	01BC	01BE			
0184	011E	0118	0020	0024	002A	002E	0020
0186	0022	0026	002C	002E	0020	0022	002C
0188	0190	0194	002E	0028	0020	0022	002C
0192	0026	0022	002C	002E	0020	0024	002E
019C	019E	0198	0020	0024	002E		
0184	019E	0198	0020	0024	002E		
0186	0022	0026					

In this picture, you can see 12 IsomRects and their IsomValues for left, top, right, bottom. In hexadecimal, of course. We can see some regularities:

The last digit (bits 13-16) are very regular. I'll come back to them later, but actually I don't have much use for these.

The first digit (bits 1-4) are all '0'. I suppose that they belong to the next group of digits. We can ignore those bits, too.

The 2 digits in the middle (bits 5-12) are the most interesting ones.

You will notice that each 2 IsomValues next to each other in the picture will always match in those 2 digits.

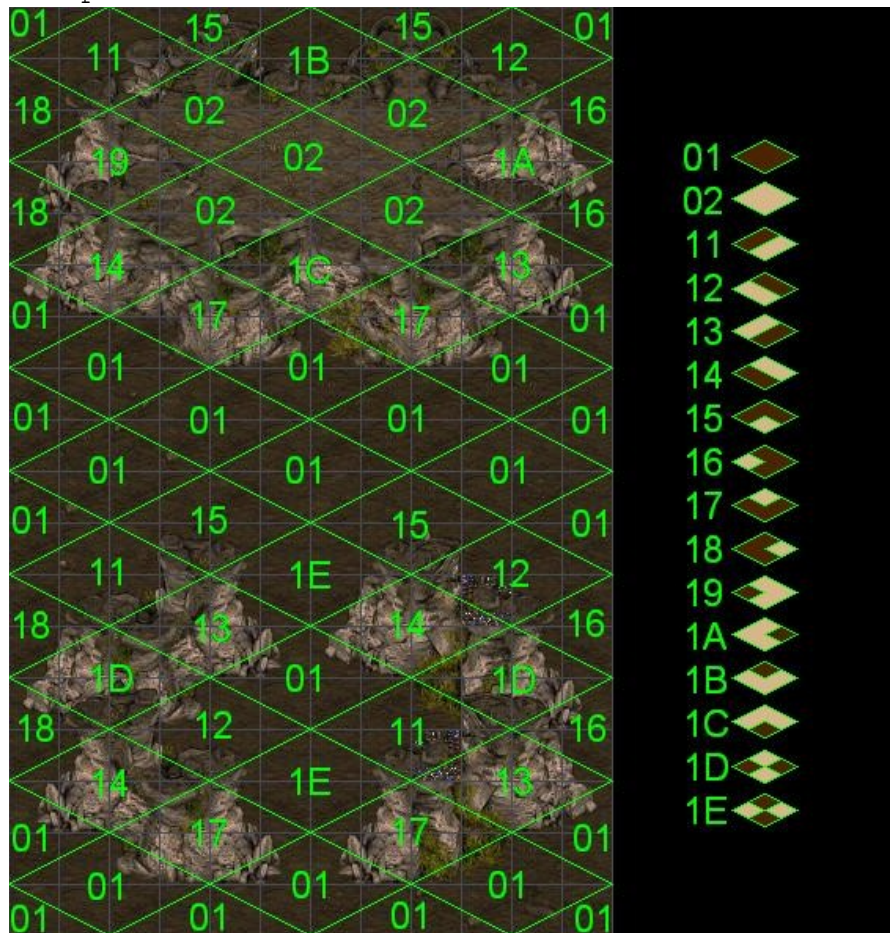
In each rectangle there are 2 pairs of values; occasionally all 4 have the same values (except for the last digit), but those can be treated as 2 pairs, too. These pairs are always next to each other (not top/bottom or right/left).

Because of that, the 8 values around each corner of each rectangle

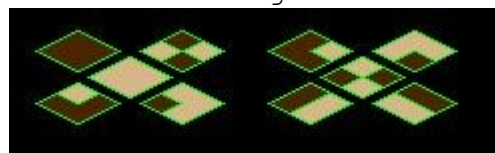
match as well.



Here you can see the 'IsomDiamonds' with their respective 'Index' in the middle. 01 stands for Dirt, 02 for HighDirt. The other values in the picture are various cliff formations.



Here you can see all the possible combinations of Dirt and HighDirt (in the ISOM Section). The column on the right shows abstract versions of each possible diamond. These diamonds can not be placed next to each other in any combination; the terrain at the edges of the diamonds must match.



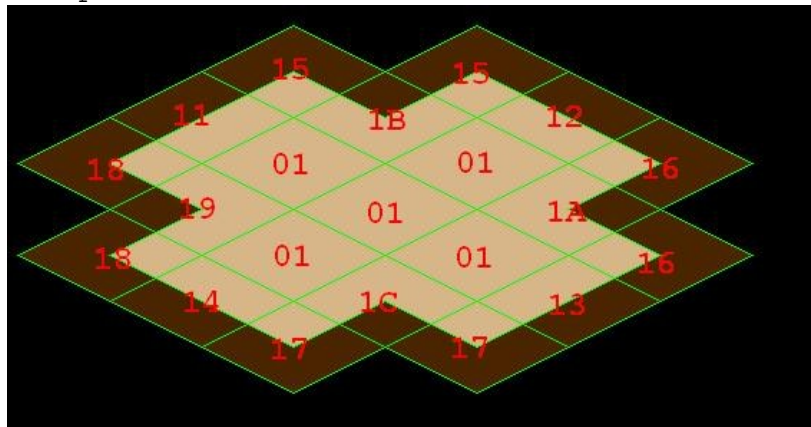
The combinations on the right are all invalid, the ones on the

right match perfectly. It's quite like a puzzle.  
 Getting TILE values from ISOM values, as well as getting ISOM Values from IsomDiamonds is simple and can be done with the help of two fairly small tables.  
 So my approach to mimic Staredit's IsomBrush is based on getting the IsomDiamonds' Indexi right.

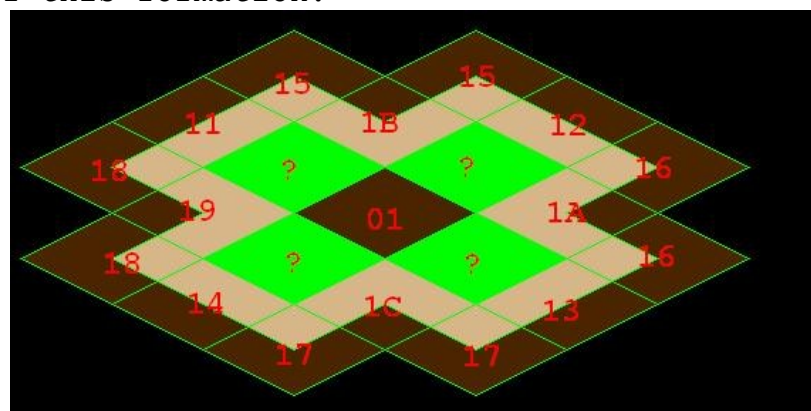
When an IsomBrush is applied to an IsomDiamond, that Diamond recieves that value. The values that can be brushed are:

01: Dirt	08: Temple
02: HighDirt	09: HighJungle
03: Water	0A: HighRuins
04: Jungle	0B: HighRaisedJungle
05: RaisedJungle	0C: HighTemple
06: RockyGround	0D: Mud
07: Ruins	

Time for an example:



This is a piece of HighDirt in a cross-shaped formation. All the IsomDiamonds that would be around this formation are 01 - Dirt. Now I will show what happens when you place a Dirt-IsomBrush in the center of this formation.



As you can see, the diamond in the middle is now flagged as Dirt. But when we apply the rule that the terrain of two touching diamonds must match, we will see that the HighDirt-Diamonds which were where now the ?-Diamonds are, cannot stay. Later on, we will see that those 4 are not the only diamonds that need to be exchanged, but we will proceed step by step.



A diamond-shaped grid of 19 cells, arranged in 5 rows. The cells are colored in a checkerboard pattern of light green and light yellow. The numbers and symbols are as follows:

		15		15	
	11		?		12
18		13		14	
	?		01		?
18		12		11	
	14		?		13
		17		17	

Mapping these diamond Indexi into the IsomRects is simple, going from there to the TILE values is not hard either, but a little tricky nevertheless because of stuff like cliffs that cover more than 2 tiles.

Well, this won't immediately give you all the knowledge you will need to write your own IsomBrush, but it should help a lot. Note: This is more than likely not the approach Staredit takes. I especially wonder if you can copy the irregularities Staredit shows at the border of a map with this algorithm. Even if you can, it seems completely illogical.

This article was written by spinesheath  
E-mail: [spines@web.de](mailto:spines@web.de)  
or contact me on blizzforums.com

Special thanks to Thomas Böhm, a good rl friend of mine.

Thanks to:

Evolipel for quite some material on ISOM, and his code for loading maps (though he won't recognize it anymore) and  
Fatimid08 for some stuff he has not done yet.

Thanks to all who read this.

Yes I know it's all sloppy and ugly.