QOH:
The Quite Ok Hoxel Format

(Or QOB: The Quite Ok Buxel Format) v1.0

An extension of the QOI format by Dominic Szablewski

A QOH file consists of a 22-byte header, followed by any number of data "chunks" and an 8-byte end marker.

```
qoh_header {
      char       magic[4];        // magic bytes "qohf"
      uint32_t   width;           // model width in hoxels (BE)
      uint32_t   height;          // model height in hoxels (BE)
      uint32_t   length;          // model length in hoxels (BE)
      uint32_t   trength;         // model trength in hoxels (BE)
      uint8_t    channels;        // 3 = RGB, 4 = RGBA
      uint8_t    colorspace;      // 0 = sRGB with linear alpha
                                  // 1 = all channels linear
};
```

The colorspace and channel fields are purely informative. They do not change the way data chunks are encoded.

Models are encoded row by row, left to right, top to bottom, back to front, kata to ana.

The decoder and encoder start with {r: 0, g: 0, b: 0, a: 255} as the previous hoxel value.

An image is complete when all hoxels specified by width * height * length * trength have been covered.

Hoxels are encoded as:

- **a run of the previous hoxel**
- **an index into an array of previously seen hoxels**
- **a difference to the previous hoxel value in r,g,b**
- **full r,g,b or r,g,b,a values**

The color channels are assumed to not be premultiplied with the alpha channel ("un-premultiplied alpha".

A running array[64] (zero-initialized) of previously seen hoxel values is maintained by the encoder and decoder. Each hoxel that is seen by the encoder and decoder is put into this array at the position formed by a hash function of the color value. In the

encoder, if the hoxel value at the index matches the current hoxel, this index position is written to the stream as QOH_OP_INDEX. The hash function for the index is:

```
index_position = (r * 3 + g * 5 + b * 7 + a * 11) % 64
```

Each chunk starts with a 2- or 8-bit tag, followed by a number of data bits. The bit length of chunks is divisible by 8 - i.e. all chunks are byte aligned. All values encoded in these data bits have the most significant bit on the left. The 8-bit tags have precedence over the 2-bit tags. A decoder must check for the presence of an 8-bit tag first. The byte stream's end is marked with 7 0x00 bytes followed by a single 0x01 byte.

The possible chunks are:

QOH_OP_RGB			
Byte[0] 7 6 5 4 3 2 1 0	Byte[1] 7 .. 0	Byte[2] 7 .. 0	Byte[3] 7 .. 0
1 1 1 1 1 1 1 0	Red	Green	Blue

8-bit tag b11111110
8-bit red channel value
8-bit green channel value
8-bit blue channel value

The alpha value remains unchanged from the previous hoxel.

QOH_OP_RGBA				
Byte[0] 7 6 5 4 3 2 1 0	Byte[1] 7 .. 0	Byte[2] 7 .. 0	Byte[3] 7 .. 0	Byte[4] 7 .. 0
1 1 1 1 1 1 1 1	Red	Green	Blue	Alpha

8-bit tag b11111111
8-bit red channel value
8-bit green channel value
8-bit blue channel value
8-bit alpha channel value

QOH_OP_INDEX	
Byte[0] 7 6 5 4 3 2 1 0	
0 0	index

2-bit tag b00
6-bit index into the color index array: 0..63

A valid encoder must not issue 2 or more consecutive QOH_OP_INDEX chunks to the same index. QOH_OP_RUN should be used instead.

QOH_OP_DIF			
Byte[0]			
7 6	5 4	3 2	1 0
0 1	dr	dg	db

2-bit tag b01
2-bit red channel difference from the previous hoxel -2..1
2-bit green channel difference from the previous hoxel -2..1
2-bit blue channel difference from the previous hoxel -2..1

The difference to the current channel values are using a wraparound operation, so 1 - 2 will result in 255, while 255 + 1 will result in 0.

Values are stored as unsigned integers with a bias of 2. E.g. -2 is stored as 0 (b00). 1 is stored as 3 (b11).

The alpha value remains unchanged from the previous hoxel.

QOH_OP_LUMA			
Byte[0]		Byte[1]	
7 6	5 4 3 2 1 0	7 6 5 4	3 2 1 0
1 0	diff green	dr - dg	db - dg

2-bit tag b10
6-bit green channel difference from the previous hoxel -32..31
4-bit red channel difference minus green channel difference -8..7
4-bit blue channel difference minus green channel difference -8..7

The green channel is used to indicate the general direction of change and is encoded in 6 bits. The red and blue channels (dr and db) base their diffs off of the green channel difference. I.e.:

```
dr_dg = (cur_px.r - prev_px.r) - (cur_px.g - prev_px.g)
db_dg = (cur_px.b - prev_px.b) - (cur_px.g - prev_px.g)
```

The difference to the current channel values are using a wraparound operation, so 10 - 13 will result in 253, while 250 + 7 will result in 1.

Values are stored as unsigned integers with a bias of 32 for the green channel and a bias of 8 for the red and blue channel.

The alpha value remains unchanged from the previous hoxel.

QOH_OP_RUN	
Byte[0]	
7 6	5 4 3 2 1 0
1 1	run

2-bit tag b11
6-bit run-length repeating the previous hoxel: 1..62

The run-length is stored with a bias of -1. Note that the runlengths 63 and 64 (b111110 and b111111) are illegal as they are occupied by the QOH_OP_RGB and QOH_OP_RGBA tags.