

[layout and purpose of the macro parts - the masks and such]

Introduction

All igneous rocks are composed of an assemblage of different juxtaposed and closely packed minerals, showing different forms, colours, sizes, and composition, the only exception being volcanic glass, like obsidian, which is an amorphous, i.e. uncrystallized, product of rapidly cooling magma. Granites, which is the subject which concerns use here primarily, have the advantage of being in general rather coarse-grained, thus generally showing clearly the different minerals which form their assemblages. The chemical composition of those minerals, and their relative proportions throughout the rock, determine the type of granite.

The observer is further helped along in the optical determination of a particular granite by the overall colour of the rock (white, pink, or grey) which is controlled by the proportional presence of each mineral. Sometimes the presence of particularly large crystals (phenocrysts) generally plagioclase, can help the determination.

For those interested, I include here a classification diagram for igneous rocks (Fig. 1) a QAPF diagram (also known as a Streckeisen diagram) as used by the scientific community. More information on this can be found here¹ and for granites in general, here².

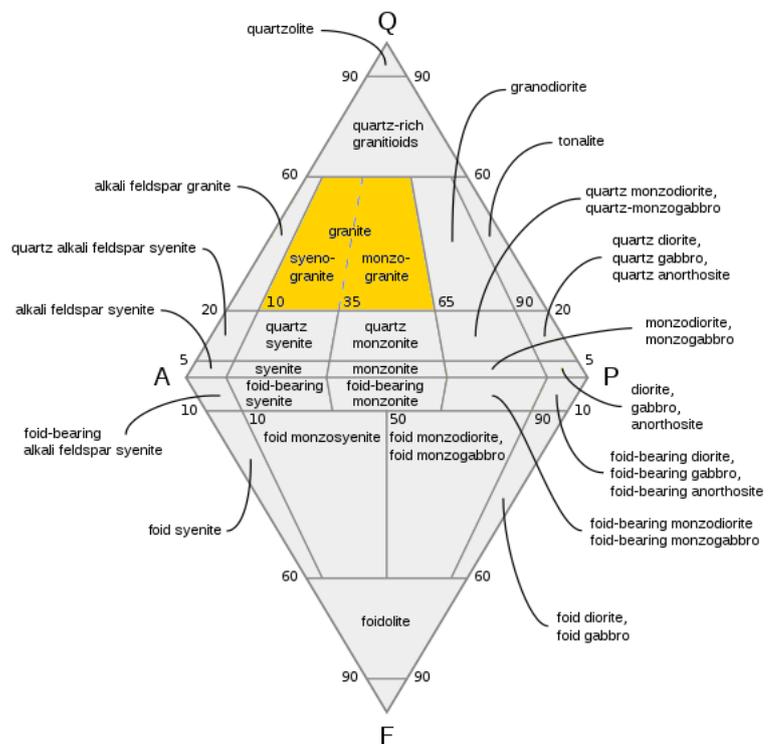


Fig. 1: QAPF diagram with granite field highlighted in yellow (after Kent G. Budge).

Granite maps and masks

To simulate real world granites, we make use of colour_maps (i.e. the different minerals/colours composing the granite pigment) and colour masks (i.e. colour_map-based pigment_patterns used within the granite texture) which control the distribution of those same minerals/colours. Necessarily, there is a close relationship between maps and masks as mineral/colour boundaries within a particular granite must correspond to their mask boundaries. The final texture is pretty complex as additional differential scaling perturbs the final aspect of the texture. In the following paragraphs we shall explain how this is achieved by the macro, illustrated by comprehensive examples and diagrams. To make things easier to understand, the different parts of the macro are described separately.

(1) https://en.wikipedia.org/wiki/QAPF_diagram

(2) <https://en.wikipedia.org/wiki/Granite>

Granite pigment creation

Sections 1 to 3 of the macro concerns the creation of granite pigments (Fig. 2). The input to this part of the macro - independently from the different parameters which will be described later - consists of three different colour map arrays (prefix A_) for each granite type file:

A_Granite_map1: describing the different minerals/colours;

A_Granite_map2: describing the veins crossing the granites (this will be discussed separately below);

A_Granite_mask_map: describing the (spatial) distribution of the different minerals/colours.

Examples are drawn from the (default) Dakota Red Granite data.

A_Granite_map1 is a mixed array and has the following format:

```
#declare Map1_entries = 18;
#declare A_Granite_map1 =
array mixed [Map1_entries][2] {
  {0.00, <not_0, not_0, not_0>},
  {0.25, <0.059, 0.059, 0.059>},

  {0.25, <0.086, 0.027, 0.059>},
  {0.35, <0.086, 0.027, 0.059>},
  /...snip.../
  {0.63, <0.769, 0.329, 0.298>},
  {0.70, <0.669, 0.229, 0.198>},
  {0.75, <0.769, 0.329, 0.298>},

  {0.75, <0.600, 0.600, 0.600>},
  {0.85, <0.550, 0.550, 0.550>},
  {1.00, <0.650, 0.650, 0.650>}
}
```

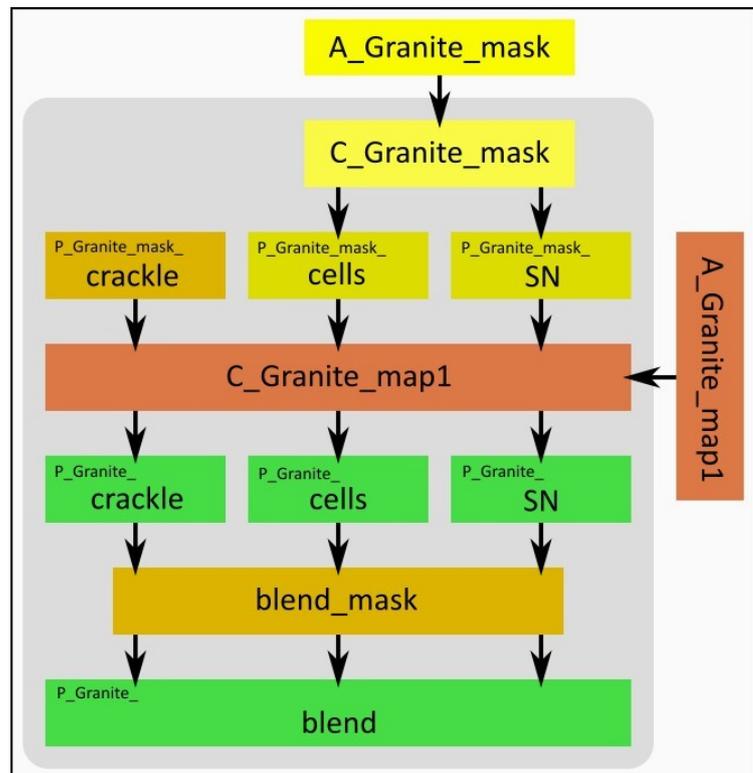


Fig. 2: Pigment generation (sections 1 to 3).

As granites consist of close-packed, dis-

crete, assemblages of minerals, this is simulated by sharply-bounded groups of colours, each group representing a different mineral. Groups/minerals are graphically shown here separated by an empty line. The array is read into the macro's C_Granite_map1 colour map, which is controlled by one of the granite pigment patterns: crackle, cells, step noise (SN). These patterns are generated by the macro's C_Granite_mask colour map, in turn fed by the A_Granite_mask mixed array, which has a similar construction to the previous array:

```
#declare Mask_entries = 14;
#declare A_Granite_mask =
array mixed [Mask_entries][2] {
  {0.00, <not_0, not_0, not_0>},
  {0.25, <not_0, not_0, not_0>},
```

```

{0.25, <0.100, 0.100, 0.100>},
{0.35, <0.100, 0.100, 0.100>},
    /...snip.../
{0.63, <0.700, 0.700, 0.700>},
{0.75, <0.700, 0.700, 0.700>},

{0.75, <0.900, 0.900, 0.900>},
{1.00, <0.900, 0.900, 0.900>}
}

```

Note that the mask map does not necessarily have the same number of entries as the colour map, but the entries representing the boundaries between each discrete group of minerals (also separated by an empty line) necessarily have identical values. Note also that the crackle pattern is internal to the macro and thus not controlled from outside.

Blend, the other internal pattern provided by the macro (*blend_mask*, Fig. 2) and developed originally by Tekno Frannansa (aka Tek) makes use of the generated pigments to provide a blended version of them. This can be very useful to simulate a weathered aspect of the granite for example.

A_Granite_map2, which controls the aspect of the (quartz) veins intersecting a given granite, is a mixed array with the following format:

```

#declare Map2_entries = 5;
#declare A_Granite_map2 =
array mixed [Map2_entries][3] {
    {0.000, <0.800, 0.800, 0.800>, 0.150},
    {0.005, <0.800, 0.800, 0.800>, 0.000},
    {0.010, <0.800, 0.800, 0.800>, 0.150},
    {0.011, <1.000, 1.000, 1.000>, 1.000},
    {1.000, <1.000, 1.000, 1.000>, 1.000}
}

```

This array describes the colour map of the quartz. It is a bit different from the minerals colour map in that it also contains transmit information. When enabled, the veins texture will be layered over the main granite texture. Remember that this feature is still in an experimental phase. It is advised to not change anything to this array and copy it from one granite include file to the next. Things may change in the future though.

In section 4 of the macro, the data from this array are fed into the internal *C_Granite_map2* colour map, which in turn is the core of the *P_Granite_veins* pigment, which uses a marble pattern.

Granite texture creation

In the texture generation, section 5, of the macro (Fig. 3) the same pigment patterns we already used in section 3 (Fig. 2). However, they are used for a different reason and in a different way. This time they control the distribution through the granite texture of different scale of the granite pigments. An array (*A_Granite_var*) serves as input for the pigment map. It has the following format:

```

#declare Var_entries = 8;
#declare A_Granite_var =
array [Var_entries][3] {
    {0.20, 0.50, 0.15},

```

```

{0.25, 1.00, 0.18},
{0.35, 1.00, 0.18},
{0.40, 0.50, 0.15},
{0.60, 1.00, 0.15},
{0.65, 0.50, 0.18},
{0.75, 1.00, 0.18},
{0.80, 0.50, 0.15}
}

```

The first entry of the array concerns the pigment map entries; the second is used in the normal map to control the strength of the granite normal pattern (frosted version of the granite); the third entry scales equally the pigments in the pigment map, as also the normal patterns in the normal map.

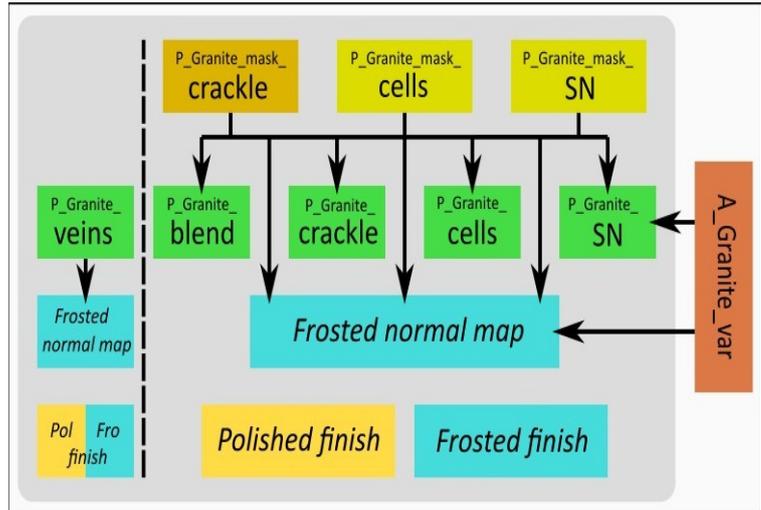


Fig 3: Texture generation (section 5).

In this section, different finish blocks are generated, corresponding to the polished or the frosted versions of the granite, but also whether granite contains veins (layered texture) or not. Finally, in this section, the normal block for the frosted version of the granite is generated.

To be complete, we need to mention here that the veins' texture (Fig. 3) has its own generation of finish blocks (polished or frosted) and a normal block where the normal map is controlled similarly by its pigment (`P_Granite_veins`) used as pigment pattern.

Still experimental is the use of subsurface light transmission. The macro parameter `SubS` switches this *on* in the finish blocks in this section. *[to be developed further]*

Last but not least about this section, the granite textures have been internally developed and scaled in such a way that they correspond globally to a fairly fine-grained real-world granite. This means that, applied without any further scale transformation to an arbitrarily-sized object, this object will show a real-world-sized granite texture. This last conforms to a dimension of 1 pov-unit being equivalent to 100 cm. The user is free to resize the texture to his own needs through the macro parameter `M_scale` which defaults to `<1,1,1>`.

Granite material creation

In section 6 of the macro, the *polished* or the *frosted* granite material blocks are generated, whether they contain veins or not. Additionally and finally, the user has the liberty to apply any transformation to the material.