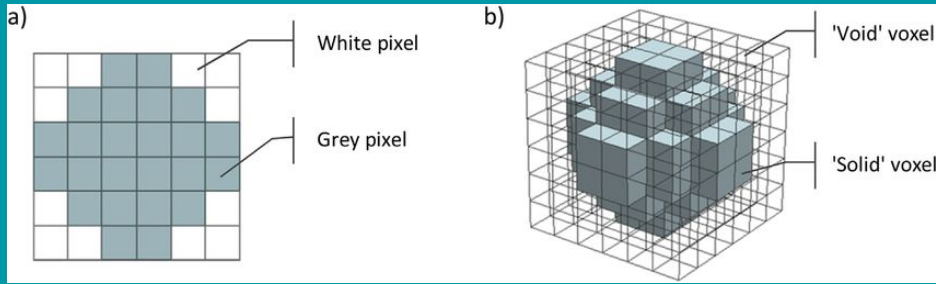


Generalities 1/3: definition

Voxels are the 3D equivalent of pixels

- Pixel = picture element - Voxel = volume element (~volumetric pixel)
- On an image, each pixel has properties, like its color and transparency
- In a volume, each voxel has properties, like its value and its material, which set its visibility and its look.

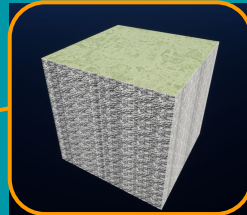
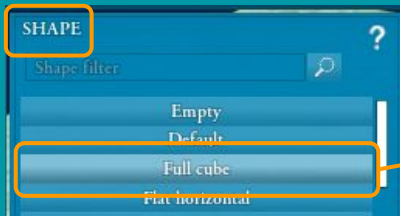


In the context of Yag, the following terms are considered equivalent:

- Voxels
- Terrain
- World

In Yag, voxels only exist inside a cube.

- This cube can be visualized with the “SHAPE/Full cube” preset:



No terrain can exist outside this cube.

Generalities 2/3: limits

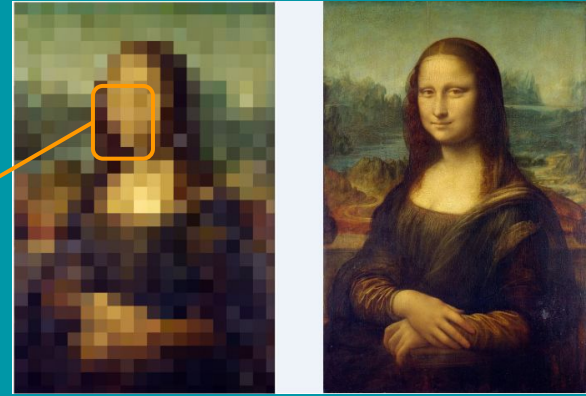
No detail smaller than the smallest unit of an environment can be created.

If each pixel of a picture is $10 \times 10 \text{ cm}^2$ no detail smaller than 10 cm can be seen.

In Yag, each pixel in the world-cube is $1 \times 1 \times 1 \text{ m}^3$

- No detail smaller than 1m can be created
- Voxels in Yag are intended to create large scale environments (plains, hills, mountains, etc.)
- No human scaled elements can be created in Yag (chairs, tables, stairs, etc.)

The meter is the lower limit of the details that can be created in Yag with the voxels.



Generalities 3/3: from mathematics to terrain

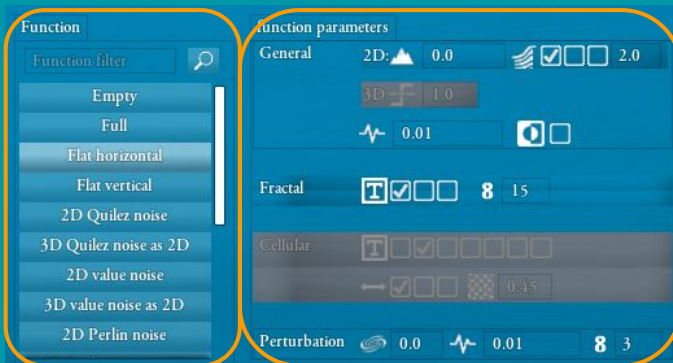
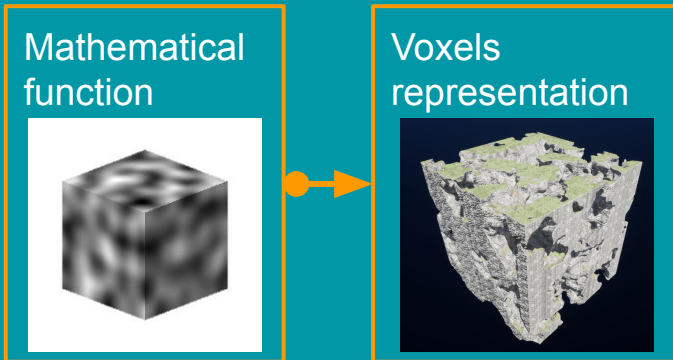
Each voxel can be assigned a value through a mathematical function.

In Yag:

- This function returns values between -1 et 1
- Voxels with a negative value are visible (= matter)
- Voxels with a positive value are invisible (= air)

The function sets the shape of the terrain:

- The more functions we have and the more modifiable they are, and the more we'll get different terrains.
- The exposed fields are the parameters allowing to modify the mathematical functions.
- It's the reason why the advanced interface for the terrains is so intimidating: there are lots of parameters.



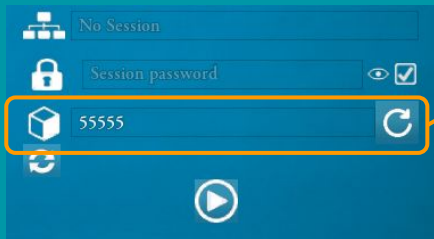
List of functions

Parameters allowing to modify the chosen function.

Prerequisites 1/2: a server dedicated to voxel data

There are 2 ways to modify the terrain in Yag:

- **Procedural: Computation of the function and its parameters**
 - Only the parameters need to be sent on the network
 - This method is very lightweight network wise: each connected computer computes its terrain and only receives the set of parameters chosen by the Dungeon Master.
 - This data uses the standard network Yag channel (the same used for all objects in the game)
 - **Manual: modifications made directly by the user (sculpting and painting the terrain)**
 - Those modifications are not computable, hence the totality of the modified data must be sent through the network.
 - This method is extremely costly network wise.
 - This data uses a separated and dedicated channel.
- Hence Yag must start a server dedicated to the synchronization of the terrain manual modifications.
- ◆ This server is automatically started by Yag when starting a standard Yag server.
 - ◆ This server is listening on a port that can be chosen before starting a server.



If you change this port, note its value carefully, it will be needed for the next step.

Prerequisites 2/2: allowing voxel data flow

IMPORTANT:

- The manipulation described here is only necessary on the server.
- Clients have nothing to do.

ONLY IF YOU ARE THE SERVER:

- Your computer is probably in the network of the router loaned by your Internet Service Provider (your “internet box”).
- It is probable that the default security rules of the router forbid the voxel server port (55555 by default, see the previous slide) to pass through.
- Hence it is imperative to configure your router to route to your computer the requests addressed to this port.
- This is called NAT (Network Address Translation) and PAT (Port Address Translation).
- You'll need to create a “NAT/PAT” rule with the following parameters:
 - Enabled
 - Name: choose any name you want
 - Internal port: the voxel port
 - External port: the voxel port (always identical to the internal port)
 - Protocole: TCP
 - Target equipment: select your computer..
- If it is not done:
 - Procedurally generated voxels will work perfectly.
 - But manual modifications will not be seen by clients.
 - Hence the only valid test is to sculpt/paint the world and check that clients receive the modifications.

The screenshot shows a router configuration page with tabs for DHCP, NAT/PAT, DNS, UPnP, DynDNS, DMZ, NTP, and IPv6. The NAT/PAT tab is active. Below the tabs, there is a warning in French: "Les règles NAT/PAT sont nécessaires pour autoriser une communication initiée depuis Internet avec un équipement particulier de votre réseau. Utiles pour certaines applications comme des jeux en lignes ou des serveurs de type FTP... Assurez-vous que cet équipement a une adresse IP statique (paramétrable dans l'onglet DHCP). Uniquement pour des équipements IPv4." Below this, there is a section titled "Vos règles personnalisées" with instructions: "Choisissez des ports qui ne sont pas bloqués par le pare-feu. Nous vous déconseillons la création d'une règle sur le port 53 (service DNS). Les équipements doivent être configurés avec une adresse IP statique pour être disponibles." There are input fields for "Telnet" (dropdown), "23" (internal port), "23" (external port), "TCP" (protocol), and a "Créer" button. Below the form is a table of rules:

Activer	Application/Service	Port interne	Port externe	Protocole	Équipement
<input checked="" type="checkbox"/>					
<input checked="" type="checkbox"/>					
<input checked="" type="checkbox"/>					
<input checked="" type="checkbox"/>	yag	55555	55555	TCP	Your computer

As an example: my own router configuration.

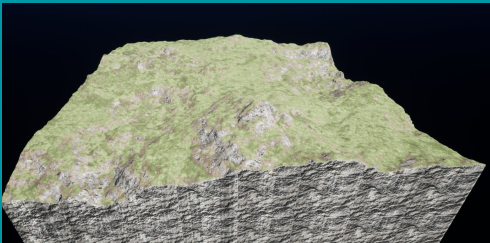
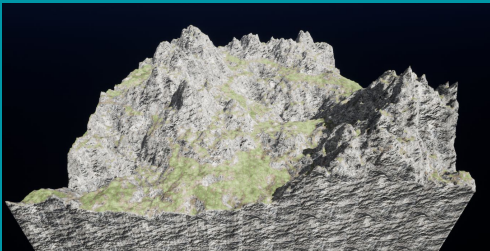
Voxels in Yag

Voxels in Yag are managed through 3 families of properties:

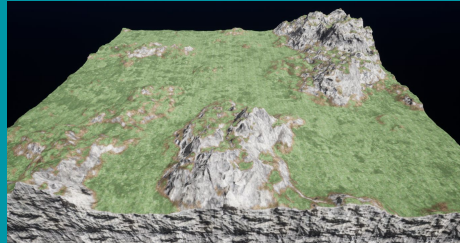
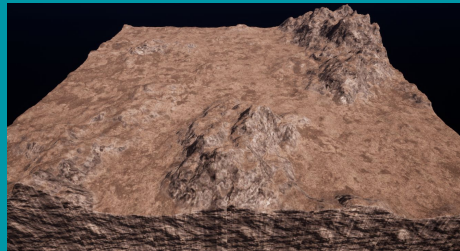
- The shape of the terrain (relief, altitude...)
- The textures of the terrain (sand, soil, snow...)
- Foliage

Those three families of properties are completely independent.

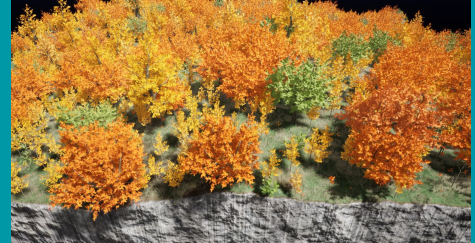
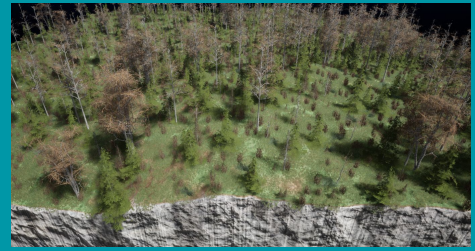
Two different **shapes**:



Two different **textures**:



Two different **foliage**:



The voxel window

Voxels are managed through a window available in the main menu:



- It contains 2 parts:
 - Simple interface
 - Advanced interface

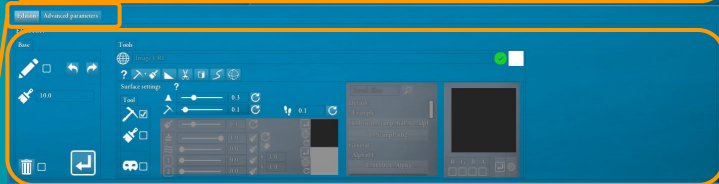
- The advanced interface also contains 2 parts:
 - Edition
 - Advanced parameters

hide/show the advanced interface



Simple interface:

- One click terrain creation
- Simple modifications



Edition/parameters



Advanced interface:

- Manual edition interface
- Parameters of the generating functions
- Foliage densities

Simple interface

The simple interface allows to create terrains very easily in a few clicks.

- To validate and propagate a modification, the corresponding “apply” button must always be pressed.



List of saved terrains

Base options:

- Name
- Seed

Choice of the shape

Choice of textures

Choice of foliage



List of 1-click presets

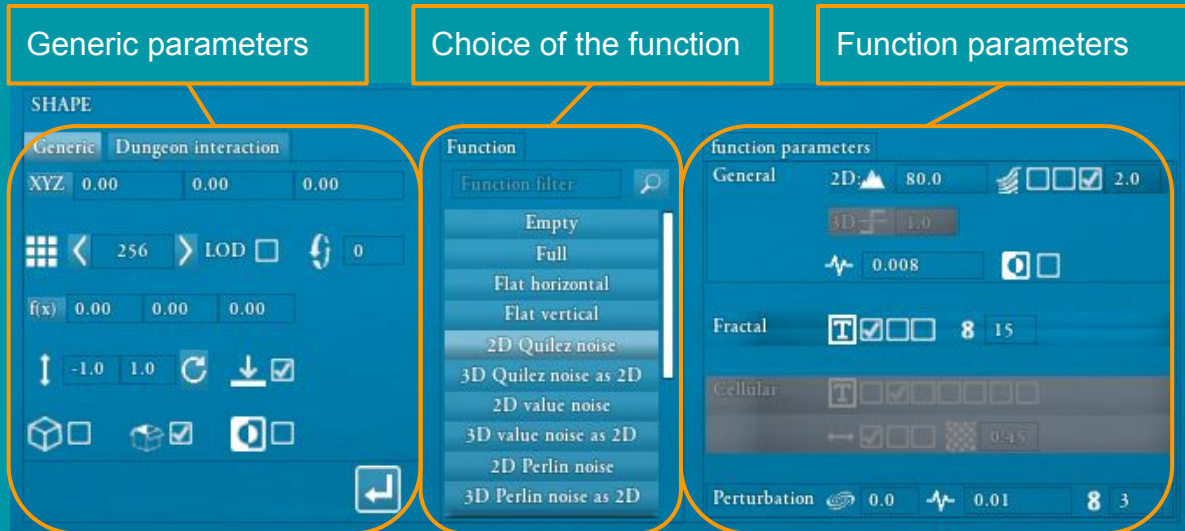
Emergency stop and terrain deletion

Number of remaining tasks

Advanced interface: shape 1/5 (generalities)

The shape advanced interface lets us fine tune the shape of the terrain.

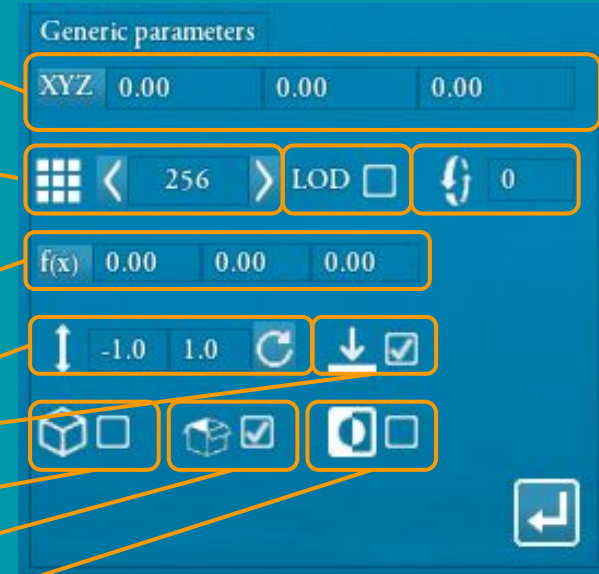
- The generic parameters are valid for all terrains, independently of the functions.
- The function is the main property, it's the one defining how the shape is computed.
- The function parameters tune the chosen function and hence depend on the function.
 - For a chosen function, non usable parameters are grayed.



Advanced interface: shape 2/5 (generic parameters)

Generic parameters concern the terrain and are valid for all functions.

- **Cube position**
 - Allow to adapt the cube position to the dungeon if needed.
- **Cube size (number of voxels)**
 - Also gives the cube size in meters.
- **LOD**
 - Slow and precise computation OR fast and approximated computations.
 - Uncheck for small terrains, check for large terrains.
- **Cube rotation**
- **Offset of the origin of the function**
 - Allows to move in the “mathematical landscape” of the function
- **Slice**
 - Allows to only build a slice of the cube (-1, 1 = percentage)
- **Ground**
 - Forces a ground at the bottom of the slice
- **Cubic mode**
 - Show each voxel as a small cube
- **Box**
 - Build the bottom of the cube as a box
- **Matter / air inversion**
 - Inverts matter and air in the cube



Advanced interface: shape 3/5 (functions)

The function is the mathematical expression of the terrain.

- It sets the essential characteristics of its shape.

Functions can be of 3 types:

- **2D**: the terrain will be a simple surface, globally horizontal.
- **3D**: the terrain will have a more complex shape that may contain holes and galleries.
- **3D as 2D**: the terrain will be globally horizontal, but the surface may have complex shapes (arches, caves...)

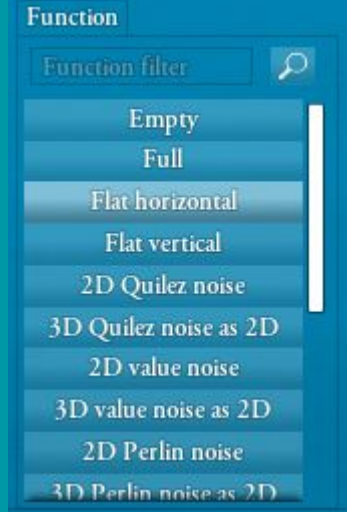
The name of a function contains useful information:

- 2D/3D: indication of the type of the function
- The name of the mathematical function (Perlin, Cellular, etc)
- “3D <name> as 2D”: 2D function created from a 3D noise.

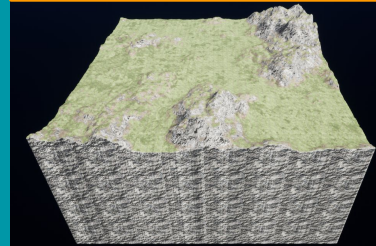
Examples:

- “2D Quilez noise”: **fonction 2D** using a 2D Quilez noise
- “3D Quilez noise”: **fonction 3D** using a 3D Quilez noise
- “3D Quilez noise as 2D”: **fonction 2D** using a 3D Quilez noise

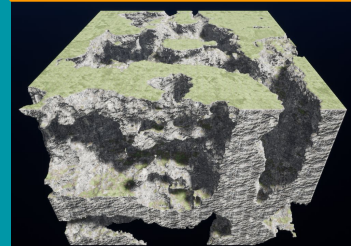
Choice of the function:



Typical 2D function:



Typical 3D function:



Advanced interface: shape 4/5 (function parameters)

The function parameters are applied to the chosen function.

- They are not explained in detail here because they can be mathematically complicated.
- They can be grayed depending on the function.

General: general parameters

- 2D:
 - Height of the terrain in meters (ex. 0 = flat, 200 = mountains)
 - Extra modifications
- 3D:
 - Threshold for the existence of matter
- Frequency: represents the roughness of the terrain
- Matter / air inversion

Fractal: fractalisation (~repetition) of the function

- Transformation of each repetition
- Octave: number of repetitions

Cellular: function made of cells

- Type of cells
- Mathematical modifiers

Perturbation: modification of the function

- Type of perturbation
- Roughness and number of repetitions



Advanced interface: shape 5/5 (dungeons integration)

Voxels can automatically adapt the geometry of the terrain to that of the dungeon.

Several options are available:

- Presence and width of a margin around the dungeon
- Altitude offset for the ground
- Height of the empty space above the dungeon
 - Allows to quickly create underground dungeons.

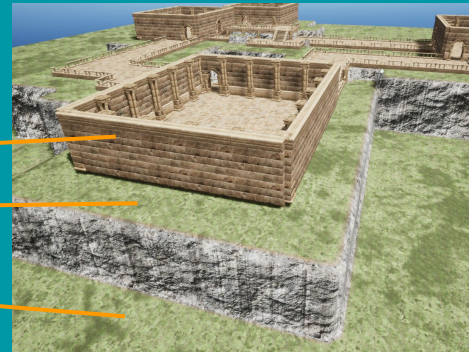
Adaptation to the dungeon or its environment can be enabled/disabled.

Foliage can be allowed or forbidden on the three geometric elements of the scene:

- Dungeon
- Margin
- Environment



This option enables or disables the adaptation to the dungeon altitude, which allows to “drive” the geometry and foliage with a dungeon that remains hidden below the ground.



Advanced interface: textures

The textures advanced interface allow to fine tune the proportion of each texture.

A terrain texture in Yag is organized in 3 layers:

- Top layer
- Intermediate layer
- Bottom layer

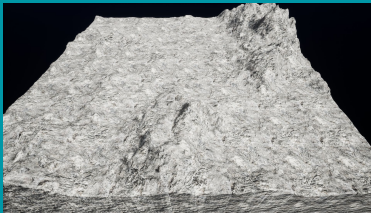
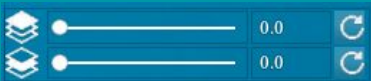
Only one layer can be seen at a time.

- Each layer covers the one that follows it.

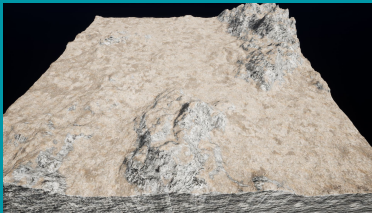
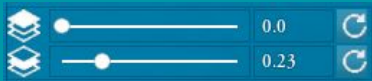
The 2 parameters set the percentages of top and intermediate layers.



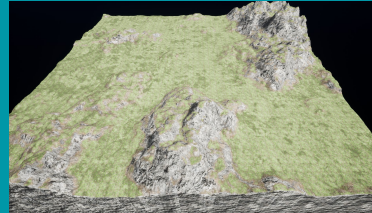
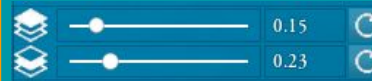
Top: 0%
Intermediate: 0%
=> only bottom layer is visible.



Top: 0%
Intermediate: 23%
=> intermediate partially covers bottom.



Top: 15%
Intermediate: 23%
=> top partially covers intermediate.

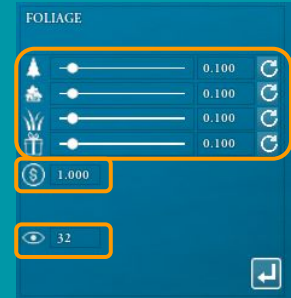


Advanced interface: foliage

The foliage advanced interface allows to fine tune the density of each type of plant.

In Yag, foliage is organized in 4 families:

- Large plants: trees
- Medium plants: bushes
- Small plants: grass
- Extra: can be anything depending on the theme

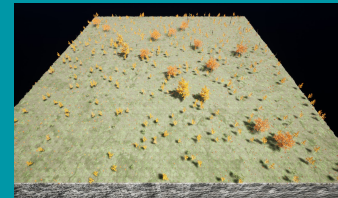
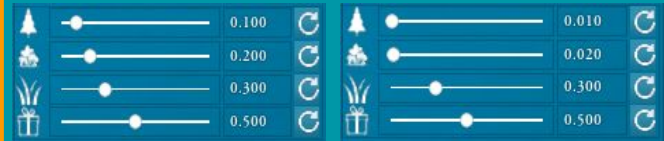


The interface has 3 families of parameters:

- Individual densities
 - Each parameters sets the density for one type of plant.
- Global density
 - A simple multiplier common to all densities
 - Allows to quickly decrease the number of polygons
 - Decrease the number if your computer is too slow
- Display distance
 - Decrease the number if your computer is too slow

Same foliage.

On the right, the densities of trees and bushes are divided by 10.



Advanced interface: manual edition / base

Edition allows to manually sculpt and paint the terrain.

This operation is only possible **on the server and for the DM.**

This interface contains 2 panels:

- Base
- Tools

Base provides a few standard elementary features of 3D edition.

- Enabling the edit mode
 - **Important:** do not forget to disable the edit mode before closing the window.
- History
 - Undo / Redo (keyboard: **Ctrl-Z / Ctrl-Y**)
 - **Important:** since history consumes a lot of memory, it is erased each time you exit the edit mode.
- Removal of all manual modifications
 - Rebuild the terrain exactly as it is computed, removing all manual modifications.

IMPORTANT REMARK:

Modifications are locally visible in real time.

However, it is essential to press the apply button to propagate them to clients.



Glossary of standard effects

Some effects are common to many tools.

Rather than repeating the explanation for each tool, here is a glossary of frequent effects and their icons.



Falloff: attenuation profile of the effect [0 - 1]

- ex: 0 => no fading, the effect stops abruptly
 - **Important:** to get the totality of a mask, you need to set the falloff to 0.
- ex: 0.1 => The effect fades quickly
- ex: 0.9 => The effect fades slowly

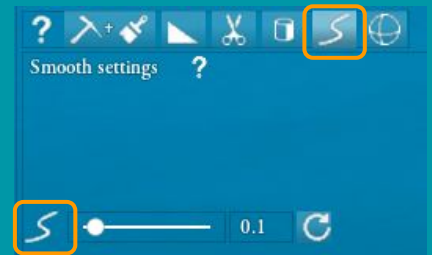


Stride: distance between 2 repetitions of the effect [> 0]

- expressed in percentage of the brush size
- ex: 0.1 => 10% => almost continuous repetition
- ex: 1 => 100% => contiguous repetition
- ex: 2 => 200% => repeated one time in two

IMPORTANT. this rule is valid for all tools:

- When an effect's icon is the same as the tool's, the effect represents the force of the tool.
- Ex: on the smoothing tool, the only parameter is the force of the smoothing.



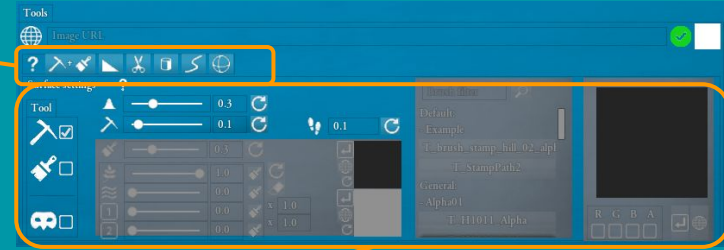
Advanced interface: manual edition / tools

Yag provides several standard tools of 3D edition.

Those tools are available through the toolbar buttons:



- Surface tools (sculpting and painting)
- Flattening
- Trimming
- Leveling
- Smoothing
- Sphere tool



Each button displays its corresponding tool panel in the central panel.
Grayed areas correspond to features that cannot be used with the options currently selected.

When edit mode is enabled and a tool is selected, the mouse cursor changes and the tool can be applied with left and right mouse buttons.

In general:

- **Left** button applies the effect while **adding** some terrain.
- **Right** button applies the effect while **removing** some terrain.



Advanced interface: manual edition / painting

To manually paint the terrain:

- Enable the edit mode on the base panel
- Select the surface tool
- Select the brush

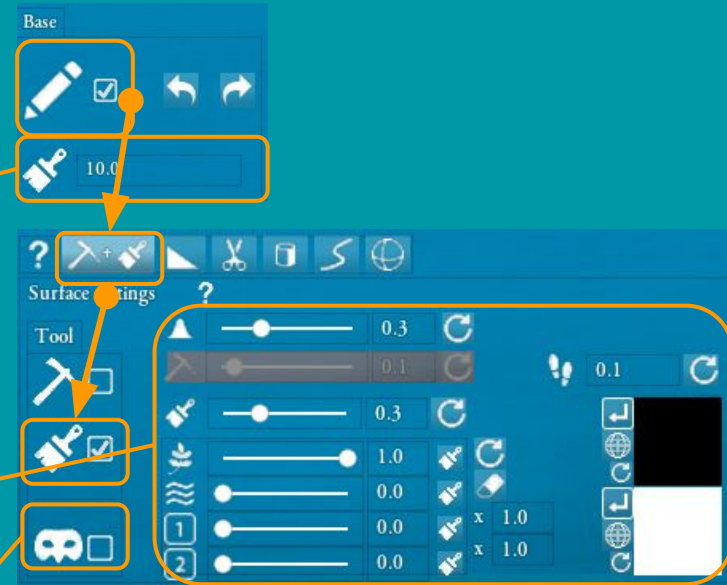
You can then paint with mouse:

- **Left click**: apply painting channel
- **Mousewheel**: change the brush size
- **Shift + Mousewheel**: rotate the brush

The following parameters are available:

- Falloff: see the effects glossary
- Stride: see the effects glossary
- Brush: force of the effect
- Painting channels (see next slide)

Masks work identically for sculpting and painting, they will be explained later.



Advanced interface: manual edition / masks

Masks can be used for sculpting and painting.

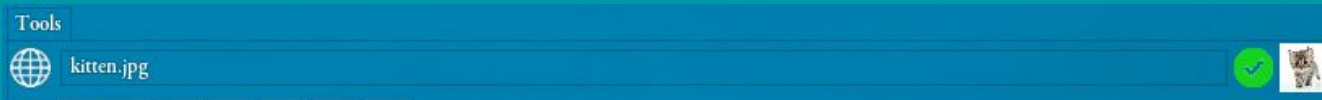
- When sculpting, they work as heightmaps
- When painting they work as brushes

To use a mask:

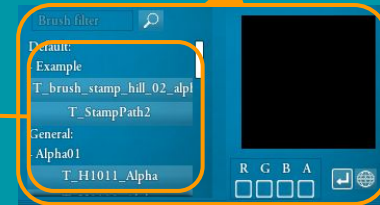
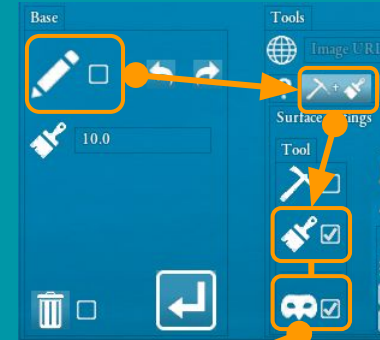
- Enable edit mode and select the sculpting or painting tool.
- Enable the option with the mask icon
- This enables the mask interface

To load a mask:

- Choose it from the native list
- Or: load it from the URL bar (local or internet address):



- Validate with the mask's "apply" button: the image appears
- Choose one of the image's RGBA channels
- The mask is ready to be used
- **Mousewheel**: change the mask size
- **Shift + Mousewheel**: rotate the mask



Advanced interface: manual edition / painting channels

Yag provides 4 channels to paint the terrain:

- 1 channel for the foliage
- 1 channel for the water
- 2 customizable channels named 1 and 2

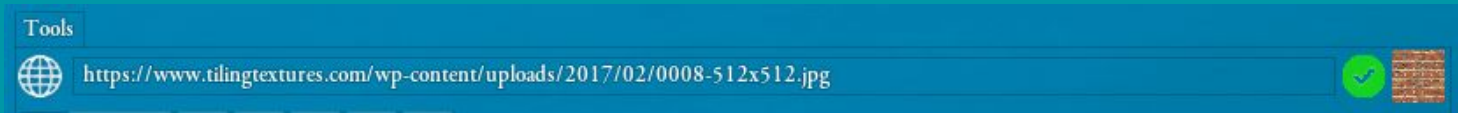
Each channel is managed through a line that allows:

- To choose its force (slider / number)
- To select it alone (brush icon button)
- The 2 channels have a multiplier to change the texture scale.
- The eraser turns off all channels.



To change the texture of a channel:

- Put a URL (or a local address) in the URL bar:



- Press the “apply” button of the chosen channel:



The two other buttons will respectively display the URL and reset the texture to its default value.

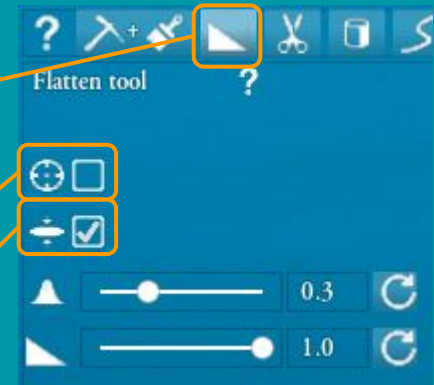
Advanced interface: manual edition / other tools

Only non trivial effects are explained here:

- Those whose icon is different from that of the tool
- Those which are neither the “falloff” nor the “stride”
- See the glossary above for more details on that matter.

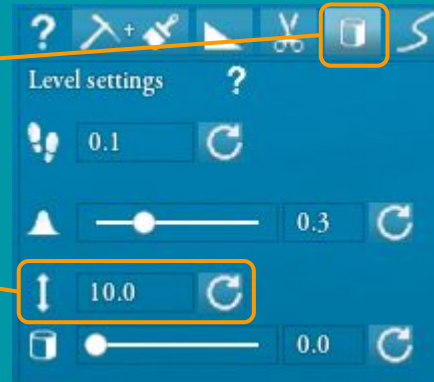
Flattening tool

- Sets whether the slope of the plane should be exactly that which is below the cursor or an average value of those hovered by the brush.
- Sets whether the slope stays constant while drawing or should follow the terrain.



Leveling tool:

- Height of the leveling cylinder



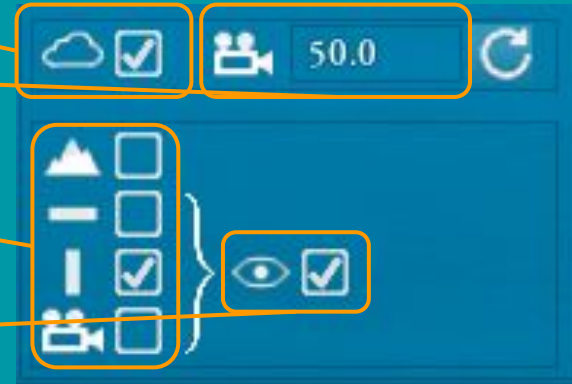
Advanced interface: manual edition / sphere tool

The “sphere” tool requires a few more explanations.

That tool is the only one allowing to create matter out of nothing because it is the only one capable of not following the existing terrain.

Its options are the following:

- 2 displacement mode **while not sculpting**:
 - Stick to the ground
 - Or stay in the air at constant distance from the camera
 - Distance can be tuned
- 4 displacement modes **while sculpting**:
 - Stick to the ground
 - Stay in the horizontal plane
 - Stay in the vertical plane
 - Stay in the plane facing the camera
 - If a plane is chosen, the eye lets us preview it



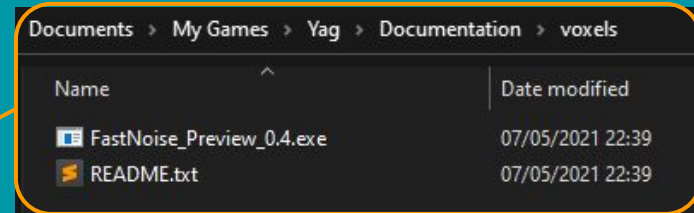
Getting familiar with fractal functions

Mathematical functions used to compute terrains are provided by an open source library (FastNoise)

FastNoise also distributes a tool to familiarize ourselves with fractal noises by experience.

You'll find this tool in the following directory:

- < my documents>|My Games|Yag|Documentation|voxels



FastNoise_Preview_0.4 lets us visualize the functions.

- The options are the ones found in Yag.
- This allows us to quickly test and prepare functions for Yag: when finding an interesting set of parameters we can copy them into Yag to get the corresponding terrain.

Functions are heightmaps

- The whiter the image, the higher the terrain
- White areas are the highest points
- Black areas are the lowest points

Hill

Hollow

