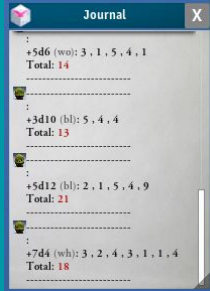# Generalities

Yag offers a realistic and customizable dice system.

The dice window is directly accessible from the menu with the "dice" button:

The results are displayed in the log:

Basic options:
- Formulas
- Colors
- Simple 1-Click rolls
- 3D dice
- Private dice

Advanced options:
- Default variations
- Custom dice

Formulas can also be found in character sheets:

Basic panel

This panel contains commonly used features.

Formula and color fields.

Simple rolls in one click:
- Each button rolls a simple formula

3D dice.
Checked:
- Standard 3D dice roll (prettier but slower)
- Non-standard dice (ex: 3d27) use a random generator
Unchecked:
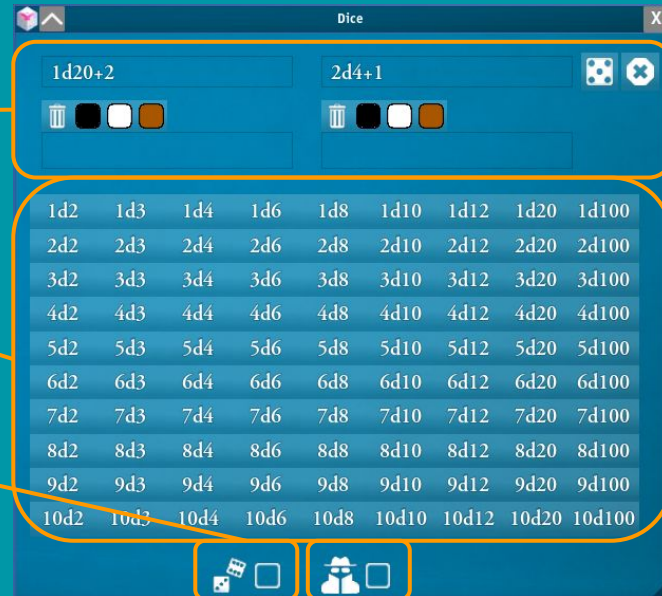- All dice use a random generator (less beautiful but instant result)

Private rolls.
Checked:
- The result is only displayed for you
Unchecked:
- The result is displayed for everyone

Formulas and language

Formulas can be typed in the 2 fields provided

| 2d20h1+1d3+2+3d6e5 | 5d10h3e9+2d4l1 | ⚙ ✖ |

A formula is a set of mini formulas separated by + and - signs.

2d20h1 + **1d3** + **2** + **3d6e5**

A mini formula is always of the form NdX[options]
- N = number of dice to roll
- X = number of faces of the die

Example:
3d8: roll 3 8-faced dice with no options

The language is as follows:
- d = dice
- h = keeps highests
- l = keep lowests
- e = explode

Examples:
- **3d10h2**: roll 3d10 and keep the 2 highests
- **4d8l1**: roll 4d8 and keep the lowest
- **2d6e5**: roll 2d6 which explode on 5 and more.
- **5d20h3e17**: roll 5d20 that explode on 17 and more, and keep the 3 highests.

The formulas are rolled 2 by 2:
- The first formula is the test of action
- The second formula is the effect of the action
- The logic is that of "hit/damage"
- If a pawn is selected, the 2nd formula triggers the attack animation

Calculations and display

The calculation and display are written in LUA and exposed to players, and can therefore be fully modified.
See the LUA API documentation to look further into this topic.
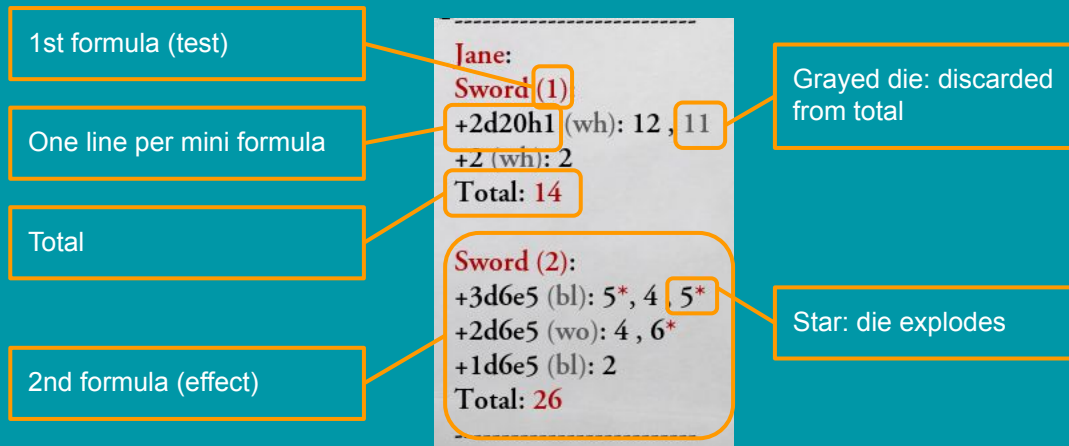We present here the native behavior of Yag.

By default Yag calculates the total of the formula.

In the illustrated roll:
Jane strikes with a sword
- She rolls 2d20 to hit
- She makes 12, 11, and keeps the best: 12
- She adds a bonus of 2
- For a total of 14

- Then she rolls 3d6 explosives for damage.
- She rolls 5, 4, 5: both 5s explode
- The reroll makes 4, 6: the 6 explodes
- The reroll makes 2
- For a total of 26

2d20h1+2   3d6e5

1st formula (test)

One line per mini formula

Total

2nd formula (effect)

Grayed die: discarded from total

Star: die explodes

Jane:
Sword (1)
+2d20h1 (wh): 12 , 11
+2 (wh): 2
Total: 14

Sword (2):
+3d6e5 (bl): 5*, 4 5*
+2d6e5 (wo): 4 , 6*
+1d6e5 (bl): 2
Total: 26

Colors

Yag offers 3 colors of dice:
- black ("bl")
- white ("wh")
- wood ("wo")

The chosen color sequence is applied to each mini formula.
- If no color is specified, the colors are chosen at random.

The color of each mini formula is given in brackets in the log

This mechanism makes it possible to distinguish the dice in the formulas.
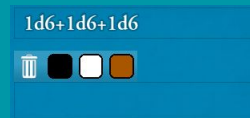For example, if a flaming sword does 2d6 damage and adds 1d6 fire, the configuration below identifies each roll:
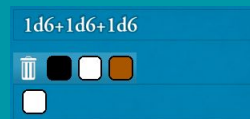- Damage dice will be white
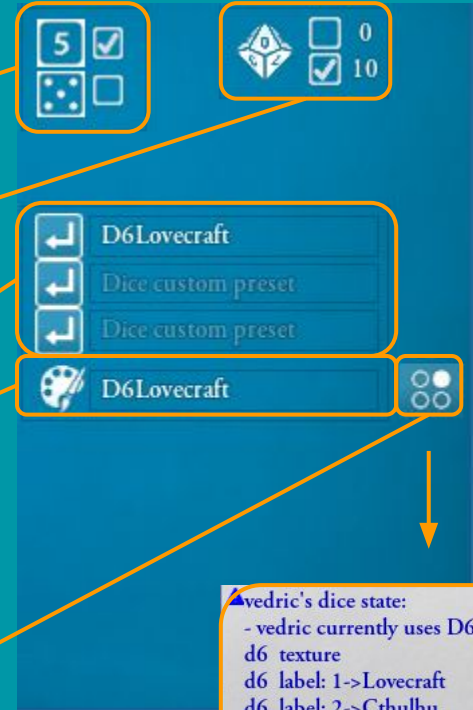- The fire die will be black
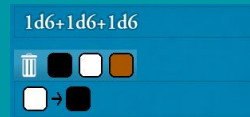
Advanced panel

This panel contains features of rare use.

- Showing d6s with numbers or dots

- Value of the 0 on the d10 (0 or 10?)

- Dice sets prepared to be loaded with one click

- Currently used dice set

- Display properties changes for the current dice set

vedric's dice state:
- vedric currently uses D6Lovecraft dice:
d6 texture
d6 label: 1->Lovecraft
d6 label: 2->Cthulhu
d6 label: 3->Tentacles
d6 label: 4->Shoggoth
d6 label: 5->Nyarlathotep
d6 label: 6->Dagon

Customizing dice 1/3: principles

Dice customization is only possible on 3D dice.
Customizing dice requires working outside of Yag.

The dice configurations are in a directory accessible from the home window.

Each sub-directory represents a dice configuration.

It specifies the differences compared to the standard dice.
- Each die can be modified by 2 files:
    - A text file for values and labels
    - A png file for the faces
- For example, if we modify only the d6, we can specify only the modification files of the d6:
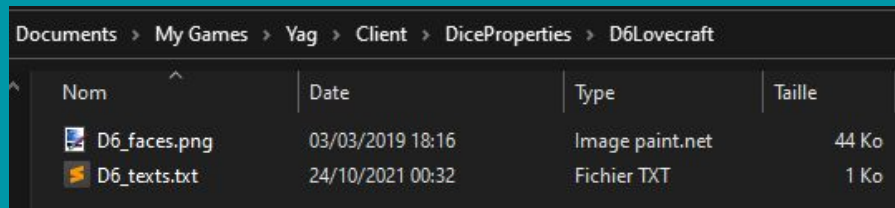
When specifying a dice set in the interface, you must use the name of the directory created for this configuration.
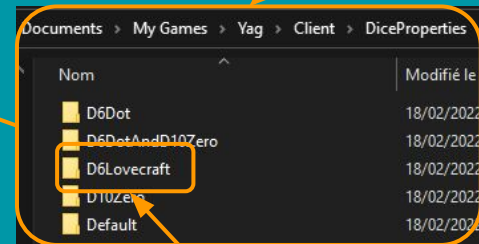
Customizing dice 2/3: values and labels

The values used for the total and the labels used for the display can be modified for each die in a text file.

This file must be called exactly "Dn_texts.txt" for die n.
For example for the d12: "D12_texts.txt"

It contains one line for each value of the die that you want to modify.
Each line contains 3 fields separated by a comma:
- 1st field: the value you want to modify
- 2nd field: the replacement value used for the total
- 3rd field: the label that will be displayed in the log

A simple example: for a game system that rolls nd6 and counts the number of 6s, one can simply modify the d6 file as follows.
- Values 1-5 count as 0 and display "fail"
- The 6 counts as 1 and displays "success !"
- The total in the log will give the number of 6s.

Example

The following file replaces the 6 of the d6 with a 2.

Display result:

Customizing dice 3/3: faces
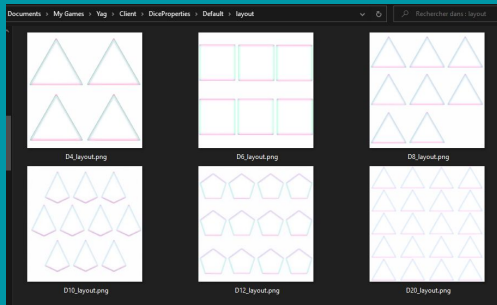
The faces of each dice are editable in a png file.

This file must be called exactly "Dn_faces.png" for die n.
For example for the d12: "D12_faces.png"

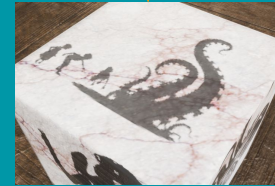It contains the image to be applied on the die.

The image must be white on black and will be applied as a mask

To create an applicable image, one can use the templates provided in the following directory:
- <My Documents>\My Games\Yag\Client\DiceProperties\Default\layout
- These templates can be used with layers to precisely position the mask for each face.

Application of the mask:

Default d6:

D6_faces.png

Modified d6:

D6_faces.png