# Advanced topics

## Contents

## 1 Overview

This document goes into more advanced notions. It is intended for those wanting to understand what is happening behind the scenes, or who want to contribute to the project. It is **not** intended for end-users.

## 2 Running The Sleeping Lion on Linux

The general way of using The Sleeping Lion on Linux is by running the command `thesleepinglion`. This will show the graphical interface.

However, one can also bypass the user interface by specifying a GML file to parse. Additionnaly, one can also give a target path to which the PDF file will be saved (if this path is not given, the resulting PDF will be placed at the same location as the GML file). For example, one can run:

`thesleepinglion Spellweaver.gml` (will create the "Spellweaver.pdf" file)

`thesleepinglion path/to/gml path/to/pdf`

## 3 In depth study of the GML format

### 3.1 Overall structure and syntax

A GML file can essentially be divided into three parts:

- a part where you will be defining a few general properties about the custom class you are creating, such as the background color used for the cards or the class's symbol
- a part where you will be defining each card separatly. For each card, you will be filling out fields corresponding to the name of the card, the initiative, the actions in the top part of the card...
- shortcuts to make writing a GML file even easier (aliases).

Whenever writing a GML file, you will often be assigning values to certain fields: if you already know YAML, JSON, Ansible or RAML, then this should feel familiar: in fact, the GML format is built on YAML. The general way to do this is by using the following syntax: `field: value`. For example, when writing down the properties of a card, you will have to write `initiative: 14` if you want the card to display an initiative of 14. Note that the fields must have a specific name: The Sleeping Lion will not recognise the words `initiativ`, `initiatve` or any other word which hasn't been thought of when designing the parser. Finally, note that **no field is mandatory** in GML. The cards may be a bit ugly, but there is no field that you must write down for the parser to work properly.

In GML, indentation matters. The identation in itself should always be the same and should be of 2 blank spaces: its presence or absence matters. Writing the following

```
initiative: 14
level: 1
```

or

```
initiative: 14
  level: 1
```

does not mean the same thing, and can generate errors.

*Note*: When using the graphical interface, The Sleeping Lion replaces every tab by two white spaces. When editing manually a GML file, take care to add only white spaces (two white spaces per level on indentation) and not tabs.

Overall, your GML file should have the following structure.

```
class:
  field1: value1
  field2: value2
    ...

card_name1:
  field1: value1
  field2: value2
    ...

card_name2:
  field1: value1
  field2: value2
    ...
...

aliases:
  field1: value1
  field2: value2
    ...
```

Note the indentation, and the colons. Detailed information about each part (class, cards and aliases) is given in the following sections.

## 3.2 General class properties

The first part allows you to define a few class-defining parameters such as the background color for cards. Here is how you would write those properties for the Spellweaver.

```
class :
  name: Spellweaver
  color: 125,0,125
  path_to_icon: path/to/icon/
```

Let's break down each line in this example:

- `class` means you are defining the global properties for this class. Note that every other value in this part is indented one level compared to the keyword `class`.
- `name` is the name of your class.
- `color` is the background color used for every card for your class. You should write three values between 0 and 255, separated by comas. These values correspond to the RGB (red, blue, green) color used for your card.
- `path_to_icon` is the path to an image (preferably a .svg file, but you can also put a .png) which will be used as an icon for your class and will be

displayed at the bottom of every card. The path should be a relative path from the gml file (something like `../../Gloomhaven/icon.svg`).

## 3.3 Creating cards

You may now create as many cards as you want. Each card should have the following syntax:

```
Fire Orbs:
  level: 1
  initiative: 69
  ID: 061
  top: |2
    \attack{3}
    ...
  bottom: |2
    \move{3}
```

Note that the names of the cards must be at the same indentation level as the keyword `class`, defined in the section above.

Let's break down each line in this example:

- `Fire Orbs` is the name of the card we are currently creating. Note that every other value in this part is indented one level compared to the name of card.
- `level` corresponds to the level of the card and will be shown at the top, in a small crown. You may also give letters, such that `level: X` is valid.
- `initiative` corresponds to the initiative of the card and will show in the middle. You may also give letters or numbers with more than two digits, although it may look ugly on the card.
- `ID` corresponds to the card ID and will be shown in small at the bottom of the card. You may also give letters or numbers, such that `S001` is valid.
- `top` **must** be followed by the "pipe" symbol `|` as well as the number `2`. Also note that the text is again indented one level compared to the keyword `top`. This is where you will be describing everything the top part of the card does. The syntax for describing the top (or bottom) half of a card is described in the tutorial.
- `bottom` is the same as `top` only the actions listed here will be displayed on the bottom of the card.

## 3.4 Adding your own aliases

The Sleeping Lion allows you to define your own aliases to write down simply your custom actions. To do this, simply add a new part to your GML file called aliases, as such:

```
aliases: |
  custom_alias1 = something
  ...
```

Note the "pipe" symbol | after the keyword aliases. You may define as many aliases as you wish: each alias should be on its own line.

*Note*: to allow a better understanding of your cards to someone reading your GML file, it is highly recommended to put the aliases section at the top of your GML file.