Introduction

Contents

1 Overview

This document aims at giving a broad overview of the possibilities and use cases of the Gloomhaven Markdown Language (later referred to as GML) and The Sleeping Lion. It also defines specific vocabulary and conventions which will be used throughout the documentation. It is highly recommended for every user to read it before using The Sleeping Lion.

1.1 What is the GML format?

The Gloomhaven Mardown Language is a way of representing a custom class and its cards. It was created with the following goals in mind:

- be easily editable
- be easily readable
- be precise and straight to the point.

Writing a GML file should be fast and relatively easy. Furthermore, a GML file should hold almost all the information needed to describe a character and its cards.

The GML format is an alternative to Excel, XML and text files, which are currently used to describe custom characters in their first stages of developpment. Its main advantage is that it can easily be converted into a beautiful PDF using The Sleeping Lion. GML files are saved using the **.gml** extension.

1.2 What is The Sleeping Lion?

The Sleeping Lion has two components.

The first part is a **parser** which is a software which generates a PDF file from a GML file. The action of converting a GML file is called **parsing**, and outputs normalized Gloomhaven cards, with format and quality similar to the original cards.

The second part is a **desktop app**. It is an all-in-one tool which allows to you to build from scratch, edit, save, and parse a GML file to create your own custom class. The desktop app provides a layer of abstraction from the GML files: it is much easier to edit a custom class using a graphical interface with buttons and prompts rather than writing text in a file.

1.3 How should I share my custom class?

Congratulations, you have created your first draft for your custom class! To share it with someone else, give them the GML file. The reviewer will be able to parse it using The Sleeping Lion to have the PDF version, and can very easily modify or tweak your custom class.

Note that if you share your custom class you should also include any artwork or glyphs you have created specifically for it: the GML format does not hold images for you.

2 Writing the top and bottom half of a card

Using the graphical interface, you will be able to define most of the information required to create a custom class, such as filling out fields corresponding to the cards' names, their initiative, level... However, the core of a Gloomhaven card lies in its top and bottom halves. Writing those card halves requires a special syntax which is described in the following sections. Don't hesitate to read them multiple times and experience using the visual interface.

2.1 General rules

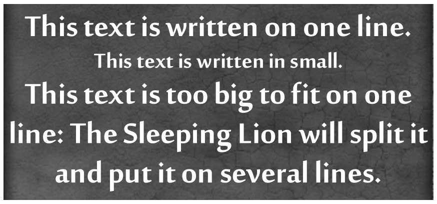
There are a few important rules to keep in mind:

- Everything you write down will appear in the output image
- Writing text on a new line will also create a new line in the output image
- **Indentation matters**. Indented text will be shown smaller than non-indented text. When indenting, it is recommended to use a tab.

- If a line you write down is too long to fit in the width of a card, it will automatically be splitted over many lines.

For example, here is how the following text is parsed by The Sleeping Lion:

```
This text is written on one line.
     This text is written in small.
This text is too big to fit on one line: The Sleeping Lion will split
it and put it on several lines.
```



This text is written on one line.
This text is written in small.
This text is too big to fit on one line: The Sleeping Lion will split it and put it on several lines.

Note that putting too many lines in a card half will cause these lines to "spill out" of the card. The Sleeping Lion does not try to fit all the lines on the card half.

2.2 Commands

The real strength of The Sleeping Lion is the way one can describe a complex action using simple functions. Commands are special instructions that are not shown as text, and instead have their own meaning. A command

- always starts with a backslash
- can have one, mutltiple or no arguments. These arguments are listed in curly brackets.

Formally, writing a command means writing something like the following:

```
\commandname{argument 1}{argument 2}...{argument n}
```

For example, the `\image` command allows you to display images on a Gloomhaven card. `\image` takes only one argument, which is the name of the image which needs to be displayed. Some images are already loaded into The Sleeping Lion, so you don't have to import them from somewhere on your computer. For example,

- `\image{attack.svg}` will show the "sword" image.
- `\image{move.svg}` will show the "boot" image.

With this very simple example, you can already create quite a few cards. For example, if the top action of card should be an "Attack 4" action and the bottom action a "Move 4" action, here is how you would write:

- the top action: `Attack \image{attack.svg} 4`
- the bottom action: `Move \image{move.svg} 4`

2.3 Aliases

With the \image command, you can already create a lot of different cards. But let’s take a look at a simple one, the Tinkerer’s Reviving Shock. The top action could be written down as:

```
Attack \image{attack.svg} 2
    Range \image{range.svg} 3
    Target \image{target.svg} 2
```

A bit tedious no? Repeating the \image command clutters up the text, and makes even the most basic action tedious to write down. Thankfully, we have defined a set of aliases to make it even simpler to write!

An alias is a short replacement for something a bit longer. For example, the alias for `Attack \image{attack.svg} 2` is `\attack{2}`. This means it is strictly equivalent for The Sleeping Lion to write down `Attack \image{attack.svg} 2` or to write `\attack{2}`.

By using aliases, we can rewrite the Tinkerer’s Reviving Shock as:

```
\attack{2}
    \range{3}
    \target{2}
```

Aliases are the best way to write a GML file are they are short and much more readable. Commonly used aliases have already been implemented: however, if needed, you can also define your own aliases.

2.4 Macros

Macros are the last type of object you might write down in a GML file. Macros are advanced and powerful objects which disrupt the typical representation of a card. For example, some cards have two separate columns, such as cards featuring area of effects like the Brute’s Leaping Cleave.

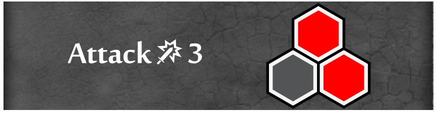
In GML, a macro

- always starts with an ”at” character (”@”)
- can have one, multiple or no arguments. These arguments as listed in curly brackets
- affects all elements placed after it in the line

A good practice is to place all macros at the beginning of the line so it is easy to see what sort of special treatment this lines getsf.

For example, the macro `@column2` tells The Sleeping Lion that this card should have two columns, and that this line should be on the second column. This is how we can write the top action for the Brute’s Sweeping Blow:

```
\attack{3}
@column2 \aoe{adjacent_two_hexes.aoe}
```



3 Additional references

You now have the basics for writing down cards using GML. A lot of commands, aliases and macros have already been implemented: you can find them in the following document. This document goes into details: see it as a dictionnary of all existing features. You will also find in the alias section explanations on how to create your own aliases.

As an example, you can find the GML file for the Spellweaver's cards (level 1 and X) here. Taking a look at how a few simple cards (like Impaling Erutpion or Fire Orbs) are transcribed in GML is also a good way to start.

You can also access those documents in the Help section of The Sleeping Lion's visual interface.