



We want **you** to create new worlds

Infinity Book

Guide For Authors

1 Welcome!

If you are reading this document you probably want to write a gamebook (or have already written it maybe) and want to turn it into an app.

Well, you're in the right place: the following instructions will allow you to edit your work in order to make it compatible with our **Infinity Creator**, the **Infinity Book** creation engine, your future app :)

In this document you will find not only the basic composition rules but also the many options at your disposal to create YOUR book, with YOUR rules, YOUR protagonists and much more... read on!

NOTE: You can comment the document to report inaccuracies or ask for clarifications.

NOTE: This documentation was translated by a *non professional translator*, please report any mistake commenting the doc!

Enjoy!

[The news of this version can be found here](#)

2 Some conventions

2.1 Examples of text (yellow)

In the yellow boxes you will find some examples, these are very important because they represent exactly how you will write your book.

Example:

#1

As you walk through the streets of the city a woman shouts "Thief!"
You look around and you see a man running and suddenly he slips into an alley.

What do you do?

There might even be some highlighted parts to draw attention to a specific aspect.

2.2 Command Syntax (blue)

The blue boxes indicate special commands that you can use to add game mechanics to the book.

Example:

@choice [CHOICE TEXT] #NEXT_PARAGRAPH

2.3 Highlights (pink)

From time to time there are some pink boxes that try to give you some important advice on a certain subject, read them!

Example:

Rember:

- The name of the paragraph MUST BE UNIQUE throughout the Book
- You can put as many spaces as you want between the # and the name of your paragraph.

3 Basics

To write your own **gamebook** (from now on also **Infinity Book**) you need to use a series of syntactical commands.

But let's start from an example... **try to guess what the following text tries to do!**

#1

As you walk through the streets of the city a woman shouts "Thief!"

You look around and you see a man running and suddenly he slips into an alley.

What do you do?

@choice [Rescue the woman] #21

@choice [Chase the thief] #46

#32

You decide there's not much to do and then you go on your way.

End of the story!

@endOfTheStory [THE END] positive

#21

«That man took my bag, please help me!»

@choice [Chase the thief] #46

@choice [You're going off on your own] #32

#46

You go into the alley, but you don't see anybody. No matter how hard you try to find the thief, he seems to have vanished.

@choice [You're going off on your own] #32

3.1 Paragraphs (#)

"A label to identify a paragraph".

The Infinity Book in the example consists of 4 paragraphs, each one marked with the # symbol and a label (which can be either text or number), for example **#56** or **#Survey**

These labels are used to refer to the paragraph and make it easily identifiable by other instructions that can be added in the book.

Of all the paragraphs you can create in your story there is one that is particularly important, this paragraph is **#1**, **it's the paragraph from which your story will start**, you will always have to insert it!

Remember:

- Paragraph #1 must always be entered!
- You can use a combination of **numbers**, **letters** and - (hyphens) to label a paragraph
- The label must be WITHOUT SPACES
- The label must be UNIQUE throughout the Book
- The order of the paragraphs in your text can be any, the important is the labels
- Examples of **correct** paragraph identifiers:
 - #123
 - #123-Attack
 - #The-Fountain
- Examples of **incorrect** paragraph identifiers:
 - # The-Fountain
- If you add text after the label will be ignored, you can use this to your advantage to add useful notes. For example:
 - #323 Clash between Lugan and the Belafus
In the above example 323 is the label, everything else is ignored
 - #AVeryLongLabel and some more text that will be ingored
In the above example AVeryLongLabel is the lable and the rest... you get it

3.2 Free Text

"What the reader will read"

In between paragraphs you can write the text of the story freely, according to your creative needs. The one highlighted below in yellow is the free text of paragraph #1.

#1

As you walk through the streets of the city a woman shouts "Thief!"
You look around and you see a man running and suddenly he slips into an alley.

What do you do?

@choice [Rescue the woman] #21

@choice [Chase the thief] #46

3.3 Comments (;)

"Commenting on your work"

If you want to insert text that you don't want to appear in the story, just use the following prefix ;
(**semicolon**)

#Attack

; Should we add a fourth choice in this key paragraph of the story?
; Think about it.

You absolutely have to make a decision!
What do you want to do?

@choice [Attack the tower] #166

@choice [Call for help] #169

@choice [Use magic] #39

3.4 Choices (@choice)

"Show a button to the reader to move through the paragraphs".

When you want to put your reader in front of a choice, all you have to do is use the following syntax:

```
@choice [CHOICE TEXT] #NEXT_PARAGRAPH
```

Simple, right?

As you can see from the example in some paragraphs we have inserted only one choice: the reader can only choose that one.

Remember:

- The text of the choice always goes in square brackets
- The paragraph to which the choice refers MUST exist
- You can enter a single choice when you want to break the paragraph into several pages to make reading easier

3.5 Endings (@endOfTheStory)

"Ending the story"

#1

Choose wisely!

@choice [I attack the dragon with my bare hands] #bad-choice

@choice [I kill him when he sleeps] #good-choice

#bad-choice

The dragon will incinerate you.

@endOfTheStory [You are dead!] negative

#good-choice

The dragon is defeated!

@endOfTheStory [You won!!!] positive

@endOfFile

Whenever a paragraph represents an ending to your book, all you have to do is use this command

@endOfTheStory [TEXT] POSITIVE|NEGATIVE

You will always have to indicate the outcome of an ending, whether it is a positive or negative ending (at the moment it is not very important to know why... you will find the explanation below).

At that point the reader will be sent back to the app's menu where they can view some statistics and maybe replay your book by making new choices.

Remember:

- You can enter as many endings as you want (one per paragraph of course)
- The text in square brackets is mandatory!
- You must always specify the outcome of an ending using **positive** or **negative**

3.6 End of file (@endOfFile)

When you've finished writing your whole book add this keyword **@endOfFile** to the end.

3.7 You are ready!

Sounds incredible, but it's true.

With these commands you can already write a wide variety of stories, you're ready to get started!

If you're at your first **Infinity Book** we strongly suggest you get familiar with these basics, write a short test story and submit it, you'll learn a lot in this process.

If you feel brave... well, go to the next section.

4 Intermediate Notions

Let's imagine, though, that you want a few more tools to make the whole thing a little more sparkling, let's see how to do it.

4.1 Full use of the command @choice

"Show a button for the reader to move through the paragraphs"

Note: *in this chapter we'll talk about how to add features to the @choice command but to do that we have to imagine that in your book there is a protagonist with some features, later you'll see how to create your character, for now we assume that our character has 4 features:*

Life = 10, Magic = 10, Gold = 10, Morale = 10

4.1.1 After choice [@afterChoice]

#TheSource

Drink from the source to refresh your energies.

@choice [Drink] #166 @afterChoice character Life +3

Pretty intuitive, right?

The **@afterChoice** keyword allows you to establish an effect that will be activated when you make your choice (i.e. when the reader clicks on the button)

In this case the character will see his **Life** growing of **3 points** before being send to paragraph **#166**.

@choice [CHOICE] #PAR @afterChoice ENTITY CHARACT +/-/=VALUE

But there's more...

4.1.2 Choice with requirement (@req)

#AnotherParagraph

You're gonna have to jump the moat, cross your fingers!
Or use your magic and prove you're a great Wizard.

@choice [Jump!] #66

@choice [Use a spell] #7 **@req character Magic greater 5**

As you can see in the second choice a keyword has been added followed by some parameters:

@req character Magic greater 5

with this command we require that the character must have more than 5 points in Magic

Note that **@req** (which stands for requirement) is only responsible for making the choice **active or inactive**, it will do nothing else (e.g. it will not take away any Magic points from the character).

With this command you can create numerous constraints (not only character-related!) with which you can build elaborate situations.

The **@req** command is made up as follows:

```
@choice [CHOICE] #PAR @req ENTITY KEY RULE VALUE
```

Let's see these parameters in more detail:

Entity

Indicates the, uh, **thing** you want to check, there are several keywords you can use, but for this introduction we'll stop at the special word **character** that indicates the protagonist of the story.

Key

Indicates any element belonging to the **thing**, in this case character characteristic: **Life**, **Magic**, **Gold** or **Morale**

Rule

It is a keyword that allows you to make a numerical comparison with the value indicated below

The keywords allowed for a rule are

- **less**
- **less_equal**
- **equal**
- **greater**
- **greater_equal**
- **not_equal**

Value

It refers to the quantity to be compared against to verify the requirement.

Let's do another example:

```
#example
...
@choice [Continue] #180
@choice [If are poor, you can click here] #81 @req character Gold less_equal 100
```

as you can see the reader can choose the second option only if has **100 Gold or less**

Rember:

- You need to be compliant with the syntax
- **@req** creates only a constraint, it does not modify any value

But there's more...

4.1.3 Hidden choice (@hide)

#nightclub

The bouncer's there and he doesn't seem inclined to let you in.
What do you do?

@choice [Go Back] #back

@choice [Attack him] #attack

@choice [Bribe him!] #bribe @hide character Gold greater_than 10

Thanks to this command the author can hide a choice, this can be useful when for narrative needs you don't want to reveal certain elements of the plot.

In the example above the reader will never know that you can bribe the bouncer to get in (he will read the option only if the requirement is respected).

This command is syntactically equal to **@req**, it only changes the initial keyword **@hide**

@choice [CHOICHE] #PAR **@hide** ENTITY KEY RULE VALUE

But there's more...

4.1.4 Let's put it all together

#FinalDuel

You're in the showdown, do your best in this magical duel!

@choice [Use the sword] #youwilldiebecauseisamagicalduel

@choice [Lightning Bolt!] #62 @req character Magic greater_than 3 @afterChoice character Magic -3

@choice [Fireball!] #262 @req character Magic greater_than 6 @afterChoice character Magic -6

@choice [KXHAS@ß#ðå!!] #614 @req character Magic greater_than 10 @afterChoice character Magic -10

In this paragraph the character will have 4 choices but not all of them will be available, depending on the amount of Magic!

With **@req** we'll check whether the choice is enabled or not, and with **@afterChoice** we'll remove the amount of Magic once the choice is made (before referring back to the appropriate paragraph).

Exciting, isn't it?

I'd be curious to know what happens to paragraph #614...

Note: The following command is a one liner (all commands are one liner!)

```
@choice [CHOICE] #PAR @req ENTITY KEY RULE VALUE @afterChoice ENTITY KEY +/-/=VALUE
```

Remember:

- You must always leave a selectable choice, otherwise the character might get stuck in the paragraph (I don't wish this to anyone).

But there's more...

No, I'm kidding :)

As for the **@choice** command there are no other options, if you've come awake up to here it means that one day you'll be one of the best Infinity Book writers on the planet, maybe in the cosmos.

And you'll be rich.

Maybe even famous.

Handsome? I don't think so.

Be satisfied :)

4.2 The @modifier command

"Changing the value of an entity".

#Beginning

You are forced to make a decisive choice.

Betray your friend James to save your own skin, try to escape by throwing yourself off the balcony, or tell the truth?

@choice [Tell the truth] #Truth

@choice [Betray James] #Betray

@choice [I Jump!] #Jump

#AnotherSimilarParagraph

Again, you are forced to make an important decision.
What are you doing?

@choice [Tell the truth] #Truth

@choice [Betray James] #Betray

#Truth

You tell the truth, it almost costs you your life, but your friend James will now trust you forever.

@modifier character Morale +7

#Betray

"James is the one who revealed the names! I had nothing to do with it, I swear, I swear!"
James looks at you in a way you'll never forget for the rest of your life...

@modifier character Morale -7

@choice [Continue] #continue

#Jump

You leap sideways, taking everyone by surprise. You jump off the balcony and land in badly, causing a terrible pain, but you're still alive.

@modifier character Life -9 @afterModify character Life lesser 0 #death

#death

The jump you make so suddenly leads you to completely lose control in the air. You try awkwardly to restore the lost balance but the asphalt gets closer and closer, inexorable. Your neck hits the road with a dull thud, breaking up instantly.

Your adventure ends here.

@endOfTheStory [The End] negative

This example highlights all possible uses of the **@modifier** keyword, this command allows you to modify a characteristic of a certain entity **when the reader goes to a certain paragraph, no matter where he comes from.**

So, if the reader goes to the paragraph where he tells the truth (#Truth), he will get +7 points to Morale.

If it reaches the paragraph where it betrays its friend James, he will lose 7 points.

In the paragraph #Jump, with the addition of the **@afterModify** subcommand, you can manage particular situations where not only a feature is modified but, once the modification is made, you can drive the character to a new paragraph without even showing the intermediate destination.

In case the character jumps will read the content of the text #Tibutti only if it has enough life points (at least 10), otherwise it will be diverted to the description of the paragraph #death in which it will face a bitter end.

Its simple syntax is

@modifier ENTITY CHARACTERISTIC +/-/=VALUE

NOTE:

- the + operator adds the VALUE to the current value of the CHARACTERISTIC
- the - operator subtracts the VALUE to the current value of the CHARACTERISTIC

- the = operator sets the VALUE to as the current value of the CHARACTERISTIC

Full syntax

```
@modifier ENTITY KEY +/-/=VALUE @afterModify ENTITY KEY RULE VALUE  
PARAGRAPH
```

Remember:

- this command applies a change to a certain characteristic when arriving at a certain PARAGRAPH, no matter where it comes from
- if you are subtracting a value that you consider vital (for example, Life) plan to use **@afterChoice** to handle any special cases

Let's also see a different example of **@afterChoice**

#Begin

Strolling around the town?

@choice [Ok] #LucyNormal

#LucyNormal

You meet Lucy, she gives you back that money she owed you and says hello.
A lucky break!

@modifier character Gold +100 @afterModify character Gold greater 500
#LucyTooRich

@choice [Back home] #BackHome

#LucyTooRich

You meet Lucy and she gives you back the money she owed you... when you open your wallet she sees all those bills and her eyes light up: "Listen, I see you're well-stocked... I have a plan..."

@choice [Tell me more...] #DifferentePlot

Basically if the character **is about to display** #LucyNormal the system will give him +100 Gold and if the total amount of his Gold is greater than 500 then the system **will NOT display** that PARAGRAPH. It will redirect the reader to #LucyTooRich.

4.3 Insert images (@pix)

"Inserting images into the story"

You can insert images at the beginning and end of a paragraph. The images will have an id name (a string).

At that point all you have to do is follow the simple syntax

```
@pix up|down ID_OF_THE_IMAGE
```

Use **up** to set the image above the paragraph, **down** to put it at the bottom.

The id name of the image must be a single word, see the examples below:

```
#ParagraphWithImage
```

In this paragraph there will be two images.

```
@pix up spaceship
```

```
@pix down laser-pistol
```

```
@choice [Continue] #anotherParagraph
```

Remember:

- You have to declare all the images you'll use in the story in a special file (Meta.txt), this will help the Infinity Creator to do checks to let you know if there are any inconsistencies. You may not be interested in this for now, but when you are ready [read this](#).

4.4 Safe Area (@save)

"Automatically save the progress of the game".

If your story is not very short and maybe not even very easy (maybe there are many paragraphs where you can find @enfOfTheStory and they are not happy endings...) there is a mechanism to help the readers.

This mechanism is called Safe Area.

Basically, when you enter the @save command in a certain paragraph, the reader will have reached a Safe Area (and the gamebook will show this information).

The reader can go to the Start menu of the Infinity Book and start from that paragraph instead of starting the story all over again.

Very useful indeed!

Remember:

- Good spots to enter a **@save** command are points where the player, even from a narrative point of view, finds himself stopping (for example at Duke Koiben's house, in a hotel in New York, or hibernating on Tropical-Alpha on a sub-light trip). In this way the concept of a safe area will be underlined.
- Only by testing and playing your book will you be able to understand if the Safe Areas you have included in the text are really functional.

4.5 Dialogue Window (@input-number e @input-text)

"Acquire information from the reader".

During reading it may be necessary to ask the reader some information.

Example:

- Enter the safe combination
- What password do you say to the guards?
- How much Gold will you offer the courtesan?

There is a command (or rather two) for this case, **@input-number** to request a number and **@input-text** to request... a text.

; one liner (as usual)

```
@input-number [Enter the safe combination] [Combination] @result equal 111333 #safe-open  
@default #safe-blocked
```

;other example

```
@input-text [Which Goddess do you worship?] [Answer] @result equal Abarath #abarath  
@result equal Balex #balex @default #unknown-goddess
```

It is always mandatory to enter the **@default**, i.e. the paragraph to which the reader should be routed in case the condition is not verified.

Syntax (**one liner!**)

```
@input-number [TEXT] [TITLE] @result RULE VALUE PAR @default PAR  
@input-text [TEXT] [TITLE] @result RULE VALUE PAR @default PAR
```

E' possibile rendere molto sofisticato il comando aggiungendo **più blocchi @result**.

; tutto su una unica riga!

@input-number [How much Gold do you offer?] [Bribe] @result less 10 #bad-reaction @result less 20 #mild-reaction @result less_equal 30 #good-reaction @default #awesome-reaction

A window will be displayed in which you can enter the value:



Digita il codice segreto

Inserisci il testo...

OK

Mostra Suggerimento

If you're a clever author you're wondering... what is Show Suggestion (Mostra suggerimento)? Well, I'm glad you asked!
If you click Show Suggestion, here's what happens:



X

Digita il codice segreto

Inserisci il testo...

Digita 700+70+7

OK

Mostra Suggerimento

The example that creates this dialogue:

#1

You're standing in front of a safe.

;One liner :)

@input-number [Enter the secret code] [Secret code] @result equal 777 #success @default
#fail @suggestion [Digit 700+70+7]

#success

Bravo!

@endOfTheStory [The End] positive

#fail

It's not working, try again

@choice [Try again] #1

@endOfFile

As you can see, all you have to do (if you want to, it's an option!) is add **@suggestion** to the command line with the suggestion text in square brackets.

If you do not indicate any kind of suggestion the button will not appear.

Remember:

- Rules are checked in order, the first that passes the test will redirect the reader to the related paragraph
- Rule for **input-number** can be
 - **less**
 - **less_equal**
 - **equal**
 - **greater**
 - **greater_than**
 - **not_equal**
- Rule for **input-text** can be
 - **equal** (the is case insensitive)
 - **not_equal**
- You must always enter a **@default!**

4.6 Badges (@badge)

"An entity for the management of player badges".

Badges (also known as achievements or goals) are a popular gaming element in digital apps and console games.

They represent a kind of reward that the player can achieve by making proper game choices.

To assign a badge the command is very simple:

```
#badge-endurance
```

```
You got a badge!
```

```
@badge endurance
```

```
@choice [Continue] #continue
```

The syntax is as follows:

```
@badge Identifier
```

The graphic representation is:



The character tab will show a badge (Distintivi) button that will allow you to access the badge gallery:



As you can see from the pictures, the badges can be associated with a **title**, a **description** and a numerical score that can be used to calculate an (optional) **game score**.

All this important information must be declared for each badge in a separate file (Meta.txt), at the moment you can skip this part but when you are ready you can go [here](#).

The badge is also an **entity**, so you can use the various commands that deal with entities to create interesting interactions, for example:

#example-99

Only if you have dragon blood you can marry Annabelle

; dragon_blood is the identifier

@choice [Marry Annabelle] #marry-annabelle @req badge dragon_blood greater_equal 1

@choice [Other choice] #other-choice

4.7 Dice (@test-dice)

"A complete system for rolling the dice and checking the result".

A gamebook system that doesn't offer a roll of the dice is not a real system.

#skill-test

The moat is very wide but you can make it!

@test-dice [Jump!] character Dexterity 3d6-3 greater 24 success #test-success fail #test-fail

With this command the reader will be given the opportunity to roll the dice and pass a test!

The system is designed to allow the author to specify the characteristic on which to test the character (in this case Dexterity) and **add a random value obtained from the dice** (in this case 3 6-sided dice minus 3).

The total **must respect a rule** (in this case 'must be greater') against a **threshold** (24).

The positive result will lead to the **first paragraph** (success) otherwise the reader will be sent back to the **second paragraph** (fail).

The syntax, always on a single line:

@test-dice [TEXT] ENTITY CHARACTERISTIC DICE RULE THRESHOLD **success**
#PARAGRAPH fail #PARAGRAPH

TEXT is the text that will appear on the button that will trigger the dice roll.

ENTITY, you should know this term by now, for more information [go here](#).

CHARACTERISTIC, is the name of the characteristic associated with an entity.

DICE, here you can express, in the canonical form known to role players, the amount of dice to roll, the type of dice to roll and any value to add or subtract.

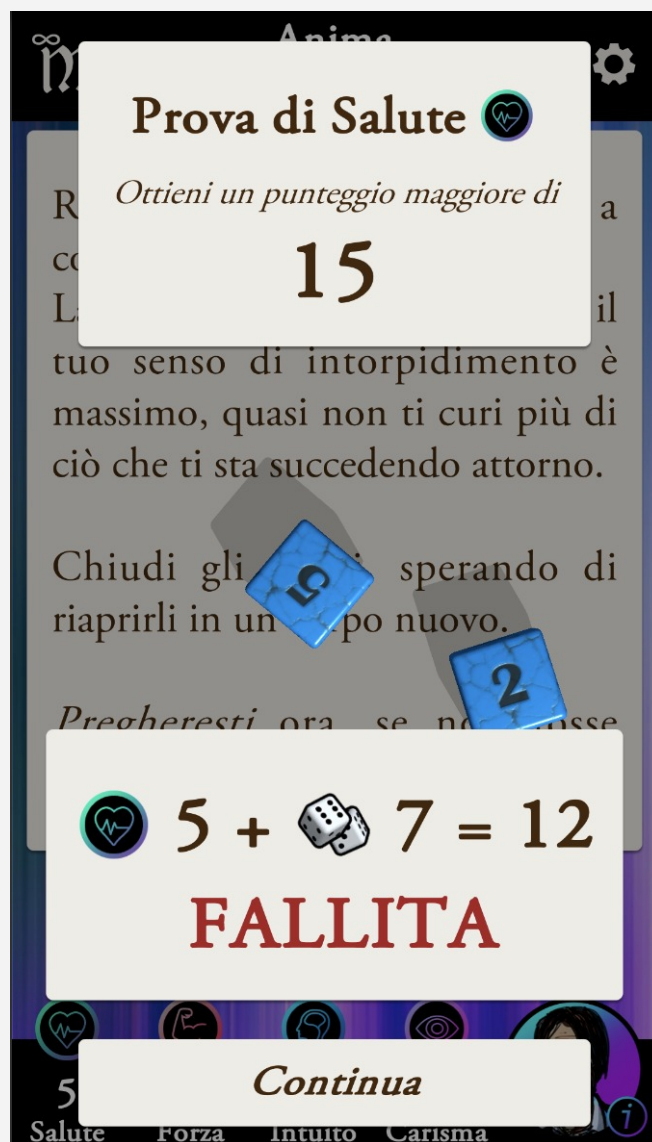
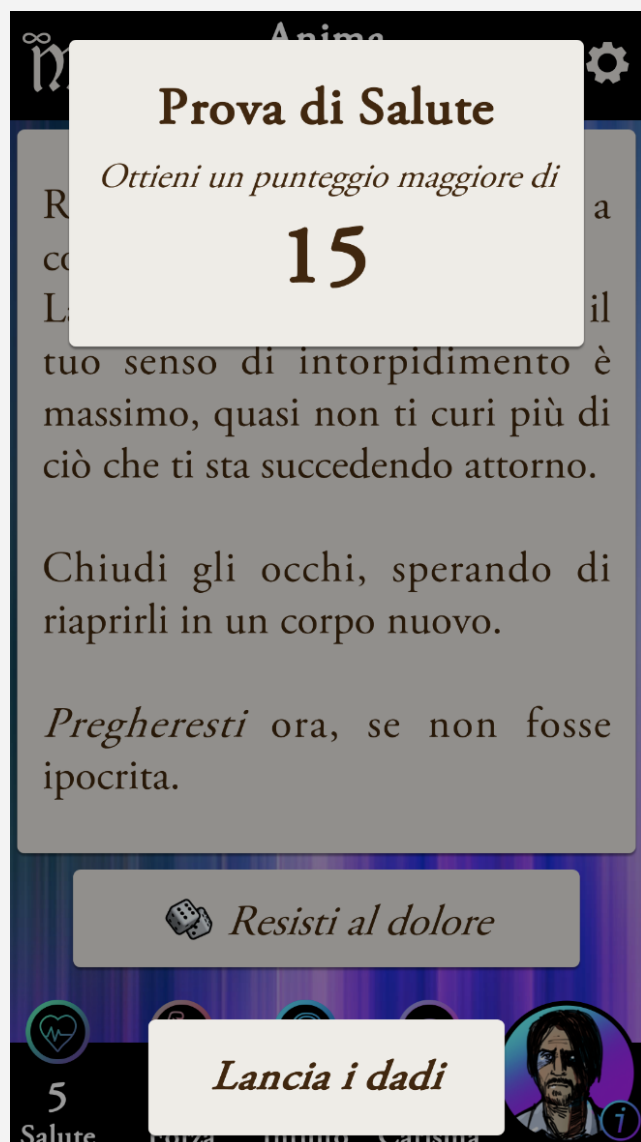
For example 2d4+1 means: roll 2 dice with 4 faces and sum 1.

For example 1d20 means: ROLL 1 die with 20 faces

At the moment only the 6-sided and 10-sided dices are supported so the wording will always be d6 or d10.

RULE, you should already know [this one too](#)

This is the graphic triggered by the command:



4.8 Spinner (@test-spinner)

In summary: *"A spinner-based minigame to perform skill tests".*



Here is another minigame for the author toolbox:

```
#spinner-test  
Spinner Test!  
@test-spinner [SPIN!] character Mana normal auto success #success-par fail #fail-par
```

```
@test-spinner [TEXT] [TITLE] ENTITY KEY DIFFICULTY AUTO|MANUAL  
success #label fail #label
```

The following entities are supported:

- character
- blackboard
- whiteboard

In the case of a **character** the test is done by transforming the value of the character characteristic into a percentage with the following formula: $\text{characteristic}/\text{max_characteristic} * 100 + \text{difficulty}$

For **blackboard and whiteboard** the value is taken as a percentage of success + difficulty (for example @modifier blackboard chance_of_success =40)

Difficulty

- **very_easy** (+50%)
- **easy** (+25%)
- **normal** (+0%)
- **medium** (-25%)
- **difficult** (-50%)
- **very_difficult** (-75%)
- **insane** (-90%)

The value is limited to a range between 5% and 95% to avoid automatic failures or successes.

- **auto or manual**, indicate if the spinner will stop automatically or if the player will perform the test, turning it into a game of skill

4.9 Timed Choices (@choice-timer)

"Create timed selectable choices"

That's funny!

An author with this option can create choices that, after a certain amount of time, will no longer be selectable. Perfect for creating narrative tension!

#1

The plane's going down!

@choice-timer [Use your parachute] #alive 5

@choice-timer[Call for help] #alive 3.5

@choice [Wait] #dead

#alive

Success!

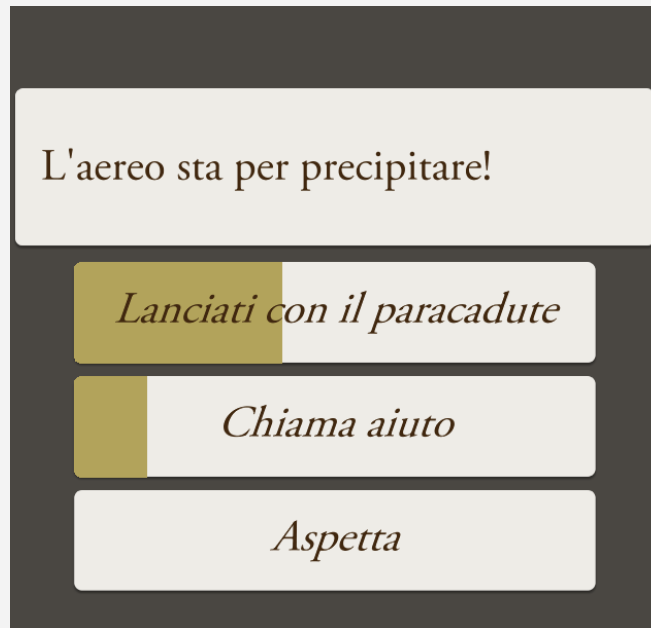
@endOfTheStory [The End] positive

#dead

Dead!

@endOfTheStory [The End] negative

@endOfFile



Syntax:

```
@choice-timer [CHOICE] #PARAGRAPH TIME
```

If you add the keyword REVEAL the choice will become pop-up: after the time indicated the choice will appear in the paragraph

```
@choice-timer [CHOICE] #PARAGRAPH TIME REVEAL
```

Remember:

- You still have to enter a **@choice** in the paragraph otherwise, the reader may not be able to select any choice!
- Time is in seconds. You can write decimals to express fractions of seconds (ie: 1.5 is 1 second and half). Use **full stop** and not **commas**.
- It's a choice, you can use it with **@afterChoice** and **@req**

4.10 Secret Choice (@choice-secret)

In summary: *"Create a selectable choice triggered by a gesture of the reader"*.

This choice will make many escape book lovers happy.

This choice remains hidden in the paragraph (not to be confused with @hide!) but, if the reader touches twice in quick succession the Infinity Mundi icon on top of the screen (the header) then the secret choice will appear as magic!

With this command you can make systems where you give the reader the responsibility to remember something (when you're in trouble you can invoke the gods or in certain paragraphs you might have a premonition...) creating fabulous game effects!

#main

You find yourself surrounded by numerous enemies, the only exit is to the north behind an evil grinning, scary, energetic man with a wicked grin.

If you venture in that direction you seriously risk discovering the side of the other opponents but see no other solution...

@choice [Try everything] #death

@choice-secret [Invoke the gods] #gods

#gods

Do you suddenly remember the promise that Zeus made to you, that one day he would save your life!

Close your eyes and try to find the right focus..

Bla bla bla...

You are safe!

@choice [Exit from the temple] #end

#end

You won!

@endOfTheStory [End] positive

#death

You try to discard the massive giant from the side, but as soon as you take a step you feel a stabbing pain in your back: a wingman has moved in your direction.

and passed you through with a spear! It's the end for you.

@endOfTheStory [End] negative



La sintassi è la seguente:

@choice-secret [TEXT] #PAR.

Rember:

- Don't be mistaken with @hide!

4.11 If...then...else (@if...@then...@else)

"Changing values to an entity or re-route to a paragraph under certain conditions".

This is a very powerful command.

With this command, the author can check a specific condition related to a specific entity (the Life of a Person, an object in the Inventory...) and if the condition is true something can be done otherwise something else can be done as well:

- changing the value of an entity
- re-route to another paragraph

Here are two examples of two options:

The first case (changing the value of an entity)

```
#1
In the next paragraph you will do a if-then-else test to change values to an entity

@choice [continue] #test-if-then-else

#test-if-then-else

If the character's Life is less than 10 then the character gets demoralized
...else Morale will be set at 15.

;the else block is optional!
@if character Life less 10 @then character Morale -1 @else character Morale 15

@choice [Try again!] #test-se-allora-altrimenti

@endOfFile
```

Syntax (one-liner):

```
@if ENTITY KEY RULE VALUE @then ENTITY KEY +/-/=VALUE @else ENTITY KEY +/-/=VALUE
```

Second case (re-routing):

#1

In the next paragraph, you'll do a if-then-else test to change paragraph

@choice [continue] #test-if-then-else

#test-if-then-else

If Life is less than 10 you'll be re-routed to #less-than-ten else #greater-or-equal-ten

;Note 1: You will never read this paragraph because of the re-routing!

#less-than-ten

@if character Life less 10 @then #less-than-ten @else #greater-or-equal-ten

@choice [Try again!] #test-if-then-else

#less-than-ten

Less than 10

@choice [Try again!] #test-if-then-else

#greater-or-equal-ten

greater or equal 10

@choice [Try again!] #test-if-then-else

@endOfFile

Syntax (one-liner)

```
@if ENTITY KEY RULE VALUE @then PARAGRAPH @else PARAGRAPH
```

Remember:

- **@then** is mandatory
- **@else** is optional
- **+** or **-** change the characteristic of the specified amount, **=** will set it to the exact value.
- You can choose **different entities** for each block.

4.12 Blackboard: Store hidden information

"Store numbers and text for granular control of the story that the reader will NOT be able to view"

Another really important tool for advanced stories.

With this entity the author can store values (both numerical and textual) and can use them to make, for example, conditional choices (see example).

The blackboard is an entity and therefore it can be used in all the commands that deal with entities.

Let's see an example:

#1

Adventurer!

You have only 4 possibilities to visit the next paragraph.

@choice [Go!] #place

#place

You are in this place.

@modifier blackboard visits +1

@if blackboard visits greater_than 4 @then #stop-visits

@choice [Back] #1

#stop-visits

You visited the place 4 times.

Now choose a potion: healing or poisonous

@choice [Healing] #healing

@choice [Poisonous] #poisonous

#healing

Setting a value in the blackboard with identifier “potion” and value “healing”

@modifier blackboard potion =healing

@choice [Continue] #what-did-you-drink

#poisonous

Setting a value in the blackboard with identifier “potion” and value “poisonous”

@modifier blackboard potion =poisonous

@choice [Continue] #what-did-you-drink

#what-did-you-drink

;You will never see this paragraph because you will always be redirected to something else based on what you drank!

;It's always good practice to enter a choice to allow the reader to continue the story

;unless you are 100% SURE that one of the cases is always verified

@ignore-no-choice

;;@ignore-no-choice is an undocumented command (yet). Validator does not throw errors when there are no choices in the paragraph.

@if blackboard potion equal vele poisonous @then #poisoned

@if blackboard potion equal healing @then #healed

#poisoned

Poisoned!

@endOfTheStory [End] negative

#healed

Healed!

@endOfTheStory [End] positive

@endOfFile

Remember:

- The blackboard is an entity, you can use it on all the commands that deal with it
- The blackboard can contain both numerical values and a **list of words**!
- [If you are interested, please read the detailed document.](#)

4.13 Whiteboard: Store public information

"Store numbers and text for granular control of the story. The reader can view them"

The Whiteboard is quite similar to a Blackboard, with the following differences:

- It is designed for the reader and not for the author (the blackboard is for the exclusive use of the author, the reader does not know that those keys and values exist)
- It will then have a graphical interface to display the information
- It's possible to customize the name in order to make it more suitable for your needs (the default is "Notes" but it can be, for example, "Clues", "Information"...).

Let's see the same example done before with the blackboard using the whiteboard:

#1

Adventurer!

You have only 4 possibilities to visit the next paragraph.

@choice [Go!] #place

#place

You are in this place.

@modifier whiteboard visits +1

@if whiteboard visits greater_than 4 @then #stop-visits

@choice [Back] #1

#basta-visite

You visited the place 4 times.

Now choose a potion: healing or poisonous

@choice [Healing] #healing

@choice [Poisonous] #poisonous

#healing

Setting a value in the blackboard with identifier “potion” and value “healing”

@modifier whiteboard potion =healing

@choice [Continue] #what-did-you-drink

#poisonous

Setting a value in the blackboard with identifier “potion” and value “poisonous”

@modifier whiteboard potion =poisonous

@choice [Continue] #what-did-you-drink

#what-did-you-drink

;You will never see this paragraph because you will always be redirected to something else based on what you drank!

;It's always good practice to enter a choice to allow the reader to continue the story

;unless you are 100% SURE that one of the cases is always verified

@ignore-no-choice

;@ignore-no-choice is an undocumented command (yet). Validator does not throw errors when there are no choices in the paragraph.

@if whiteboard potion equal vele poisonous @then #poisoned

@if whiteboard potion equal healing @then #healed

#poisoned

Poisoned!

@endOfTheStory [End] negative

#healed

Healed!

@endOfTheStory [End] positive

@endOfFile

The difference is that the whiteboard also generates a graphical interface that the reader can view:



Rember:

- The whiteboard is an entity, you can use it on all the commands that deal with it
- The whiteboard can contain both numerical values and a **list of words!**
- [If you are interested, please read the detailed document.](#)

4.14 Conditional Text (<show>)

In summary: *"Show text only if certain conditions are met"*

This is a command that acts on the text so it doesn't come with the usual @

An example is worth a thousand words:

#start

This line is always visible...

<show=character Life equal 9>This text is visible only if Helena's Life is 9</show>

<show=character Life less 5>This text is visible only if Helena's Life is less than 5</show>

More text here...

@choice [Continue] #continue

The command is one-liner, if you want to show the text on multiple lines use
:

<show=character Life equal 9>This text is on a line
 this on another line</show>

<show= ENTITY CHARACTERISTIC RULE VALUE> TEXT </show>

Remember:

- Close the command with </show>
- To write multiple lines use

4.15 Going back (@choice-back)

In summary: *"Automatically create a button to return to the last paragraph visited"*.

Sometimes the reader reaches paragraphs from different points and would like to go back to where he came from. This simple, very powerful command allows you to do this.

```
#1
Choose which planet to go to
@choice [Pluto] #pluto
@choice [Venus] #venus

#pluto
Visit the dream world and then come back here!
@choice [Go to dreamland] #dreamland

#venus
Visit the dream world and then come back here!
@choice [Go to dreamland] #dreamland

#dreamland
Bravo! Now go back where you came from!
@choice-back [Return]
```

```
@choice-back [TEXT]
```

4.16 Combat System (@combat-standard)

In summary: *"Create a fight between the protagonist and an enemy".*

The command awaited by all :)

You can create fights in your **Infinity Book**.

The topic is extensive and has been treated on a separate document, [click here!](#)



4.17 Music and Sound (@music @sfx)

In summary: "*Creating music and sounds in your story*"

Another set of commands that take your story to a higher level of entertainment

```
#1
Enter the frightening cave or listen to the howls

@choice [Cave] #cave
@choice [Howls] #howls

#cave
Listen to the background music

@music my_horror_music loop

[...]

#howls
AUUUUU!

@sfx howl

[...]
```

```
@music id (one_shot|end_of_page|loop)

@music stop

@sfx id
```

Very simple but effective:

@music plays a background music and we can decide if:

- **one_shot**: play until the end of the audio track
- **end_of_page**: sounds until you change page (**not supported yet**)
- **loop**: it plays continuously

- **stop**: stops the current track

@sfx plays a special effect, has no options

When you insert sounds in your story you must also declare them in the Meta.txt file [look here](#)

The **Validator** will play default sounds to let you know that the command has been received

If you want to listen to your real sounds you can specify the files to play by creating an **Audio** folder in the directory where the history files are located

4.18 VARS!

In summary: *"Using variables in the commands for advanced history management"*

Note: *This is an advanced and powerful command, as all advanced and powerful things can generate problems that are difficult to find in your story. If you are at your first Infinity Book we advise against using it. Use it wisely!*

Vars allow you to make more dynamic use of almost all the commands in the guide.

They are divided into 4 categories:

- **Routing Vars**, to reference a paragraph dynamically
 - **For example:** This choice when selected sends the character to a different paragraph depending on the value of a certain key in my blackboard
- **Texts Vars**, to make dynamic values appear in the paragraph
 - **For example:** This text must show the character's name, or his class, or the contents of a blackboard or whiteboard key
- **Control Vars**, to make controls in the commands dynamically
 - **For example:** This choice is selectable only based on the value of a certain blackboard key.
- **Change Vars**, to change values dynamically
 - **For example:** I want to change the value of a character characteristic according to the value of a certain key in the blackboard

The command requires a separate doc, [which you can find here](#)

4.19 Simple Roll (@roll)

In summary: "Roll the dice and save the result"...

Although the effect is very simple, I can introduce this command only now that you know the [blackboard](#) (you know it, right?). With this command the reader can roll dices whose sum will be saved inside a blackboard key.

In this way you can manage the result as you like.

```
#here
```

```
It's time to roll some dices!
```

```
@roll [Roll!] 3d6+1 blackboard my_result #analyze-result
```

```
#analyze-result
```

```
; Here I'm using some VARS, not strictly related to this command
```

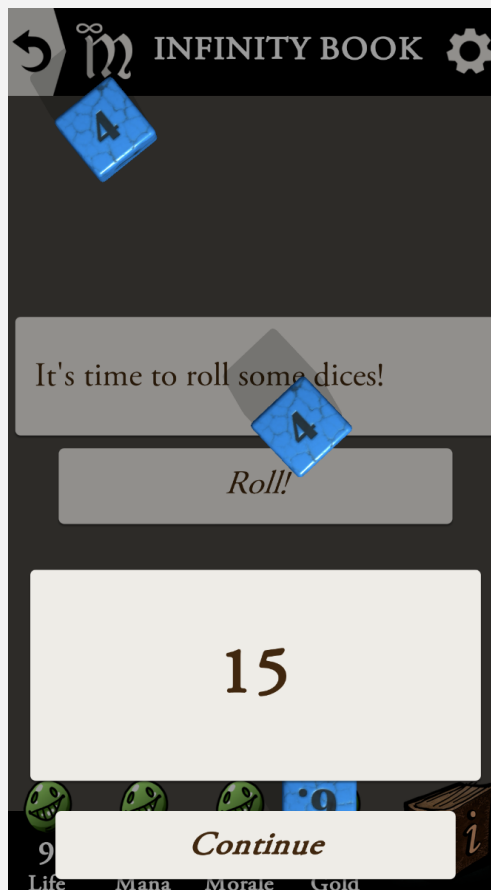
```
You scored {{blackboard-my_result}}
```

```
@if blackboard my_result less 10 @then #something
```

```
@if blackboard my_result equal 13 @then #something_with_13
```

```
...
```

```
@roll [TEXT] DICES blackboard key PARAGRAPH
```



5 Advanced notions

No one's ever gone beyond the [Intermediate Notions](#), no kidding.

You're probably here by mistake. Were you looking for the chapter on [Characters](#)? It's up ahead.

Better warn you first, in this chapter we'll be talking about the dreaded Meta.txt file. It's a necessary evil for those who want to master the deepest secrets of an Infinity Book (do you want to insert an Inventory in your story?), the choice is yours...

5.1 Structure of an Infinity Book

To write an Infinity Book you need at least one text file called **Chapter.txt** where you will write your story.

To this file you can add a **Characters.txt** file where you will write the instructions that define your characters (see the [dedicated chapter](#)).

There is a third file, called **Meta.txt**, where you will write a series of information to complement your story.

In a moment we will see in detail the commands inside the Meta.txt file but first it is the case to deal in detail with the concept of Entity.

5.2 Entity

Until now all the commands that referred to an Entity have been applied to the character. Well, the character is not the only *thing* (read Entity) we can use to interact with the mechanics of our story!

For example, we could have an Inventory in which to store the objects that our reader will find during his adventure.

Or an entity could be a Badge, i.e. a goal reached in your story (e.g. "Dragon slayer!" or "Merciful") that is unlocked as a result of certain choices.

Or Clues!

Or... uh, I think you got it :)

And then you can write instructions like this:

#jade-statues

You found 3 jade figurines, all perfectly equal!

@modifier inventory statue +3

@choice [Continue] #continue

#eat

You have complete meal.

@modifier inventory meal -1

@choice [Continue] #continue

#binoculars

You can go down to the valley or take a look first if you have binoculars.

@choice [Use binoculars] #use-binoculars @req inventory binoculars greater_than 1

@choice [Just go] #go

5.3 Meta.txt

In order to include these new **entities** in your book you must first define them so that Infinity Creator knows they exist and can allow the reader to interact with them.

The same thing is done with the character, it is defined in the Characters.txt file and then used in the ways we have seen so far.

In the Meta.txt file the same thing is done for all other entities.

5.3.1 Inventory (@inventory)

In summary: *"An entity to manage the character's possessions"*

If you want the character to have items to use during their adventure then you have to declare these in the Meta.txt file

The inventory has been completely rewritten with a new more advanced and flexible version, read the Inventory document (TODO ENG VERSION**)**

5.3.2 Images (@image-id)

In summary: "List the images that will be used in the story".

Remember the good old days of basics?

Among the various commands there was the possibility to make an image appear in the paragraph (@pix)

The images you use in that command must be listed in the Meta file, this will allow you to be sure that everything is always in order and there are no errors.

Here is, for example, a list of some images in the Meta file of the Demo:

```
@image-id logo [Logo]
@image-id lugan [Lugan]
@image-id eleus [Eleus]
@image-id belafus [Belafus (Gnam!)]
@image-id book [Book] hide
@image-id death [Death]
```

The command expects an identifier for the image and also (in square brackets) a title that will appear in the Gallery of the game.

If you are a careful observer you will have noticed that the penultimate line of the example ends with **hide**, well that is an optional command that you can use if you don't want a certain image to appear in the Gallery (for example not to show the reader special images that you want to keep in total secrecy, such as an image related to the resolution of a riddle).

The syntax is very simple

```
@image-id IMAGE_ID [TITLE GALLERY] hide (optional)
```

5.3.3 Sounds (@music-id @sfx-id)

In summary: *"List the audio that will be used in the story"*

The sounds you use with [the commands in your story](#) must be listed in the Meta file, this will allow you to be sure that everything is always in order and there are no errors.

Example:

```
@music-id intro  
@sfx-id scream
```

The names given here are exactly the names you'll use in the commands within Chapter.txt

Also if you want to listen to your sounds in the **Validator**, you have to create an **Audio** folder and insert the .wav files (only .wav files are supported in the **Validator**) with the same name used here.

```
@music-id id  
@sfx-id id
```

5.3.4 Badges (@badge-id)

In summary: *"List the badges that will be used in the story"*

As for the images of the paragraphs also for the images of the badges, it is useful to keep under control all the illustrations used.

The illustrations you use in the [command that assigns a badge](#) must be listed in the Meta file, this will allow you to be sure that everything is always in order and there are no errors.

The command expects

- an **identifier** for the image
- a **title**
- a **score** (if you don't use the scores put +0)
- a (optional) **badge description**

Here are some examples:

```
@badge-id blood [Dragon Blood] +100 [You have a superhuman endurance]
@badge-id fearless [Fearless] +100
@badge-id assassin [Assassin!!] -100 [You killed an innocent]
@badge-id friend [Friend] +0 [You are a good friend]
```

Syntax:

```
@badge-id      BADGE_ID      [TITLE]      SCORE      [DESCRIPTION]
```

Remember:

- The badge can also have a negative meaning and subtract points from the player!
 - es: @badge assassin [Assassin!] -100 [You killed an innocent]

5.3.5 Blackboard (@blackboard-id)

In summary: *"List the blackboard keys that will be used in the story"*

If you use a blackboard in your story you will need to list in the Meta file all the identifiers.

This will allow you to be sure that everything is always in order and there are no errors.

Example:

```
@blackboard-id visits  
@blackboard-id potion
```

This will allow you to use the **visits** key and the **potion** key in your story.

The syntax is very simple

```
@blackboard-id IDENTIFIER
```

5.3.6 Whiteboard (@whiteboard-id)

In summary: *"List the whiteboard keys that will be used in the story"*

If you use a whiteboard in your story you will need to list in the Meta file all the identifiers.

This will allow you to be sure that everything is always in order and there are no errors.

Example:

```
@whiteboard-id visite  
@whiteboard-id potion
```

This will allow you to use the **visits** key and the **potion** key in your story.

The syntax is very simple

```
@whiteboard-id IDENTIFIER
```

If you want to change the default name (Clipboard):

```
@whiteboard-title [Clues]
```

```
@whiteboard-title [NEW LABEL]
```

5.3.7 History (@history)

In summary: *"Allow the reader to go back of one paragraph at the time"*

This command in the Meta.txt file will allow the reader to go back to the choices made.

This is a very powerful feature!

A button to go back will appear on the graphical interface.

@history

5.3.8 Metadata

In summary: "Show credits in the relevant menu within the app".

The Meta.txt file will collect a series of (optional) information that the author might want to enter in his Infinity Book.

Example:

```
@title [Lugan the Mage]
@author [Luigi Lescarini]
@illustrator [Luigi Lescarini]
@music [Luigi Lescarini]
```

Syntax:

```
@title [TITLE]
@author [AUTHOR]
@illustrator [ILLUSTRATOR]
@music [MUSICIAN]
```

5.3.9 Interface language

In summary: *"Choose the language in which the app will interact with the reader"*.

```
@language english
```

```
@language NOME_DELLA_LINGUA
```

At the moment two languages are supported:

- italian
- english

5.3.10 Final score (@rank and the return of @endOfTheStory)

In summary: "Show a score screen at the end of the story"

Some authors may have the desire to show a final screenshot when the reader arrives at an end of the story which, as we now know, is decreed by the [@endOfTheStory](#) command

The screen looks like:

Distintivi	120
Mana	12
Gold	150
Finale	100
Totale	382
Il tuo Rank	
Grande Xaxabat	
Tabella dei Rank	
Apprendista	0-100
Illusionista	101-200
Incantatore	201-300
Grande Xaxabat	301+

TODO update image

The **Total** score is calculated by adding the following factors:

- Sum of the values of all **Badges**
- Sum of Character **Characteristics** (if the author has marked them in a special way)
- Sum of the value associated with the specific **Final**

Let's see how we do this:

5.3.10.1 Define the Rank Table

In the Meta.txt file the author defines his own **Rank Table**.

Example:

```
@rank [Apprentice] 100
@rank [Illusionist] 200
@rank [Enchanter] 300
@rank [Supreme Xaxabat] 400
```

Syntax:

```
@rank [RANK_NAME] THRESHOLD
```

5.3.10.2 Marking Character Characteristics

The author can mark some of the character characteristics that will contribute to the **Total**.

We have not yet introduced the concept of character but you can simply add the keyword **rank** to a characteristic in the Characters.txt file and this characteristics will add to the **Total**.

```
@ characteristic Life 9
@ characteristic Mana 12 rank
@ characteristic Morale 9
@ characteristic Gold 150 rank
```

See [this chapter](#) to know more about characters.

5.3.10.3 Define which endings will show the Score screen

To do so, just slightly revisit the @endOfTheStory command as follows:

```
@endOfTheStory [TEXT] POSITIVE|NEGATIVE (RANK +/-SCORE)
```

Example:

```
#ExampleWithFinalScore
```

```
Bla bla bla...
```

```
Good job!
```

```
@endOfTheStory [THE END] positive rank +100
```

Those are the steps you need to follow:

- In the Meta.txt file we define the Rank Table (from the lowest value to the highest)
- In the Characters.txt file we determine which characteristics contribute to the final score
- In the Chapter.txt file we determine which endings will show the score screen

5.3.11 Options (@options)

In summary: *"A series of options for the author"*

With this command in the Meta file you can specify a number of general options for the Infinity Creator.

At the moment there is only one option:

- font_choice_normal
 - If you set this option the choices within the Infinity Book will lose the default Italic style and appear in normal style

Example:

```
@options font_choice_normal
```

Syntax:

```
@options OPTION
```

5.3.12 Buy full version (@choice-buy)

In summary: *"Allow the reader to purchase the complete book"*

The **@choice-buy** command generates a button which, if clicked, will display a pop-up inviting the user to buy the full version of the Infinity Book.

There are two types of pop-ups: blocking pop-ups and non-blocking pop-ups (also called free pop-ups), the difference is very simple: non-blocking pop-ups allow the user to continue reading even though he does not want to buy the Infinity Book, blocking pop-ups do not offer this option.

It is **strongly recommended** to insert a free purchase choice at the beginning of the story if your intention is to sell the book.

It is also recommended to insert a blocking purchase choice in a paragraph of particular interest to the reader, in order to make the purchase more likely.

Here are the two pop-ups:

Supportaci! Puoi giocare i primi paragrafi del libro gratuitamente oppure acquista subito l'intero volume con un semplice click. Comprarlo ci aiuterà a scrivere altre storie fantastiche! Continua Acquista USD \$0.01	Acquista Il Libro X La versione gratuita del libro termina qui. Se ti è piaciuto allora aiutaci a scrivere altre storie fantastiche comprando la versione completa! Acquista USD \$0.01
--	--

Example:

#1
Welcome!

This is the first paragraph of a sample story.

@choice-buy [Continue] #continue free

#continue

The story continues and after a while you'll come to an epic moment where you have to make an epic choice.

Are you ready?

@choice-buy [Certo!] #epic-moment blocking

#epic-moment

This is the epic moment!

Congratulations

@choice [Restart] #1

@endOfFile

Syntax:

@choice-buy [TEXT] #PARAGRAPH **free | blocking**

Remember:

- Enter as soon as possible a free choice of purchase or explain to the reader that the entire book will not be free of charge
- Enter a blocking purchase choice at a narratively significant paragraph
- You can enter more @choice-buy in various paragraphs to ensure that the reader does not find a way to get to the end of the book without having had the opportunity to buy the volume
- The purchase price is determined by Infinity Mundi and cannot be configured with this command

5.3.13 Interactive maps (@map)

In summary: *"Manage an interactive map to allow the reader to move from one paragraph to another".*

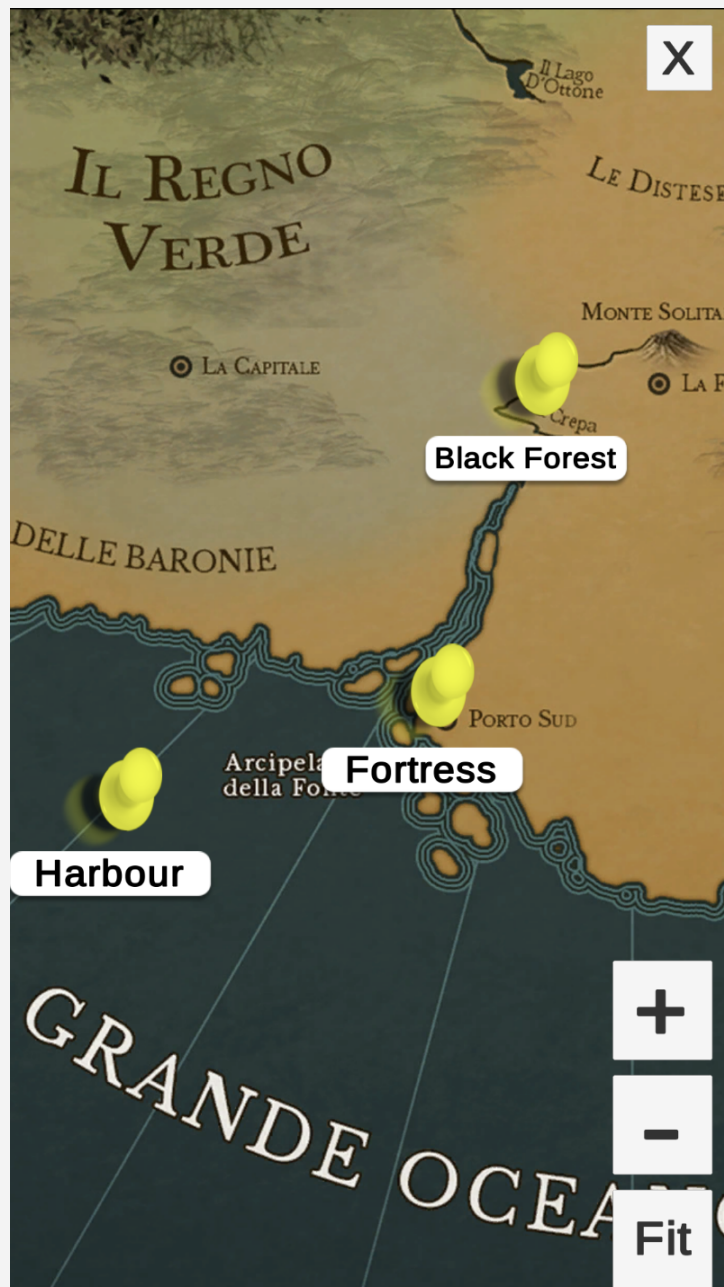
In the Meta file it is necessary to declare the existence of a map through an identification name and then locate the points on the map that will be clickable and navigable.

File Meta.txt

```
; A) Declaration of a map
@map-id my_map
; B) Declaration of the waypoints
@map-location my_map harbour [Harbour] #harbour 500 500
@map-location my_map fortress [Fortress] #fortress 800 600
@map-location my_map black_forest [Black Forest] #black-forest 900 900
```

Syntax:

```
@map-id identifier
@map-location id_map id_location [Name] #Paragraph Position(in pixel)
```



6 The Character, or better, the Characters!

We've come to an important chapter.

Let's make a little premise:

With everything you've learned in [Basics](#), you can write a complete story.

The story can be addressed to an hypothetical reader and everything would go smoothly (as in all the examples shown in that section).

If you've also read [Intermediate Notions](#) (and maybe even played [Anima](#)) you've already understood the potential of introducing a character into the story both from a narrative point of view and in terms of game mechanics.

This chapter will explain all the commands to allow your characters to come to life, let's get started!

6.1 Character creation (@character start - @character end)

Example:

```
@character start Eleus  
; here in the middle will go more information related to the character  
@character end Eleus
```

With this command you will create a character identified by the string (only one word!) Eleus. Surely having an identification string is a start but we need something more interesting.

Syntax:

```
@character start CHARACTER_IDENTIFIER  
@character end CHARACTER_IDENTIFIER
```

6.2 Character's personal data (@bio)

Well, let's start with something simple but certainly impressive, the character's biography.

Using the @bio command followed by a parameter, you can associate general information to your character. This information will be displayed on the character's tab, which more or less will look like this:



Peso: 54

Helena nasce a Sommerlund e perde all'età di 3 anni tutte e due i genitori a causa di una terribile tragedia.

Da allora è stata presa in cura dal maniscalco del Re, che le ha insegnato a forgiare le armi, sellare i cavalli e a combattere.

Helena ora è una potente guerriera che vaga per le Terre Selvagge per porre fine alle ingiustizie e, di tanto in tanto, per farsi un paio di calzari nuovi di pelle di drago...



This information will add depth to your story and your characters.

To achieve the result above is very simple

@character start Helena

@bio NAME Helena

@bio NICK The Executioner

@bio CLASS Knight

@bio MOTTO Every dragon I meet is a dead dragon...

@bio AGE 54

@bio SEX Female

@bio HEIGHT 1.65

@bio WEIGHT 54

@bio STORY START

Helena was born in Sommerlund and lost both her parents at the age of 3 due to a terrible tragedy.

Since then she has been taken care of by the King's farrier, who taught her how to forge weapons, saddle horses and fight.

Helena is now a powerful warrior who wanders the Wild Lands to end injustice and, from time to time, to make herself a new pair of dragon skin shoes...

@bio STORY END

@character end Helena

Syntax:

```
@bio NAME CHARACTER_NAME
@bio NICK CHARACTER_NICK
@bio CLASS CHARACTER_CLASS
@bio MOTTO CHARACTER_MOTTO
@bio AGE CHARACTER_AGE
@bio SEX CHARACTER_SEX
@bio HEIGHT CHARACTER_HEIGHT
@bio WEIGHT CHARACTER_WEIGHT
@bio STORY START
BIO TEXT
@bio STORA FINE
```

Rember:

- Like many commands throughout the guide you can use both upper and lower case to write them, in this case we used upper case to highlight this property
- You can use just a part of these commands!
- You can use your imagination and write, for example, **@bio AGE does not ask this an ogre**

6.3 Character's characteristics (@characteristic)

Every story, every genre, has its own peculiarities. A detective story will have a different style from a vampire story. A fantasy story will perhaps characterize its protagonists in an alternative way to a cyberpunk story...

Different genres, different characters.
Different characteristics.

What is a characteristic?

A characteristic is an attribute of your character to which you can associate a numerical value.

Life 10
Magic 5
Morale 10
Gold 20

These are the 4 characteristics that we have chosen for our story.

But they can change, for example for an investigative novel we could have a detective with

Life (maybe this is still useful) 10
Charisma 8
Dexterity 9
Intelligence 12

Formally there is no limit to the number of characteristics your character can have, to start we suggest you think of exactly 4.

Continuing with the Example above:

```
@character start Eleus
@characteristic Life 10
@characteristic Magic 5
@characteristic Morale 10
@characteristic Gold 20
@character end Eleus
```

Syntax:

```
@characteristic CHARACTERISTIC (ABBR.) INITIAL_VALUE (RANK)
```

ABBR. (abbreviation) is an **optional** command, used to indicate a 2 or 3 letters (max) abbreviation for the feature that will be used to display the feature name in certain cases.

Example:

```
@character start Eleus  
@characteristic Life LF 10  
@characteristic Magic MA 5  
@characteristic Morale MO 10  
@characteristic Gold AU 20  
@character end Eleus
```

RANK is an **optional** command that you can add to a characteristic if you plan to create a final scoring screen.

[Look here for more information about rank and score.](#)

New: maximum value.

You can now add a maximum value for a characteristic

Syntax:

```
@characteristic CHAR. (ABBR.) INITIAL_VALUE (@MAX_VALUE) (RANK)
```

In the following example we set the maximum values for Life, Magic and Morale

```
@character start Eleus  
@characteristics Life PV 10@20  
@characteristics Magic MA 5@20
```

@Moral Characteristics MO 10@15

@Features Gold AU 20

@character end Eleus

At the moment this command has no effect on the game but soon Infinity Creator will take it into account, also it is essential if you want to use the [minigame Spinner](#).

Remember:

- The name of the characteristic must be a single word
- You can create any type of characteristic with an associated numerical value
- Use a maximum of 6 characteristics for your character
- You can change the values of the characteristics during the story according to your reader's choices.
- You can condition some choices that the reader will be able to make based on the values of the characteristics
- You can even change the course of the story based on the values of certain characteristics!
- Their usage is largely explained in the [Intermediate Notions](#)

6.4 How to use the character in the story.

With the previous commands you've defined a character you can use in your story... but you haven't yet associated it explicitly to your reader!

To do this you'll have to use two commands inside a paragraph where you want to set a specific character, the two commands are **@character set** and **@character show**

6.4.1 Set a character (@character set)

With this command Infinity Creator will use this character for the rest of the game.

Then, any changes to the characteristics (or other changes) will be made to the character you just set.

#Eleus

Here we are! You've chosen Eleus!

@character set Eleus

Syntax:

```
@character set CHARACTER_IDENTIFIER
```

Once you use the command you will have created the link between player and character but **there will be no graphical change on the game interface.**

To make the character appear graphically continue reading...

6.4.2 Show a character (@character show)

With this second command you will make your character appear on the screen.



Of course the graphics of the features and avatar will be tailored to your story!

#EleusShow

Look at him!

@character show

Syntax:

```
@character show
```

6.4.3 Basic use of @character set and @character show

Typically these two commands will be paired in a paragraph, first you set a character and immediately after, in the same paragraph, you show it.

```
#Set_And_Show_Eleus
```

```
Here we are! You've chosen Eleus!  
Look at him!
```

```
@character set Eleus
```

```
@character show
```

For a basic and classic use of the character you can stop here, you have everything you need to use a character.

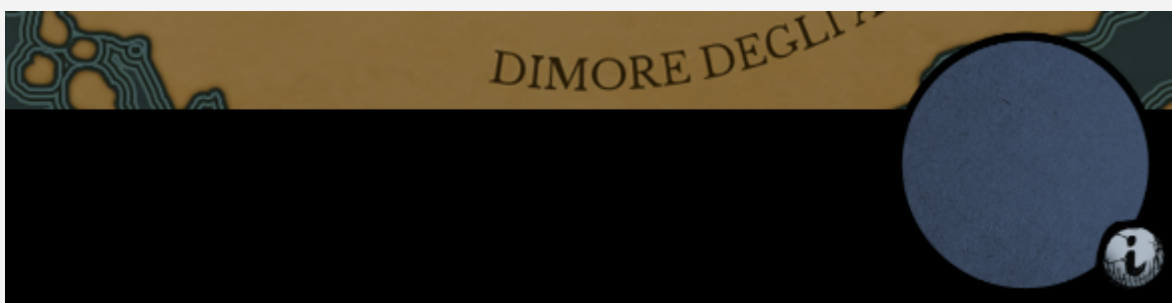
If you're a veteran Infinity Book writer, if you're feeling adventurous or just curious, keep reading, but remember: ***Infinity Creator** is a modular, expandable and flexible system, you don't have to use any commands if you don't feel confident with it.*

6.4.4 Why two separate commands?

Why make two separate commands instead of using one and make our lives easier?

Quite simply because it gives the author more creative opportunities.

- For example you can simply set the character without showing it on the screen (maybe because you want something to happen to it but still, for stylistic needs, you don't want to show it on the screen).
- Or you just want to show the empty graphics wrap without any character set, i.e. like this



6.4.5 Remove a character (@character remove)

With this command you eliminate the link between the player and the character.

Attention!

- Only use this command if you know what you're doing. If a player does not have a character associated with him, all the game mechanisms that have to do with the character will not work. You can't take 4 Life points away from your character if there is no character!

6.4.6 Hide a character (@character hide)

With this command you will hide the character graphics from the game interface.

Remember:

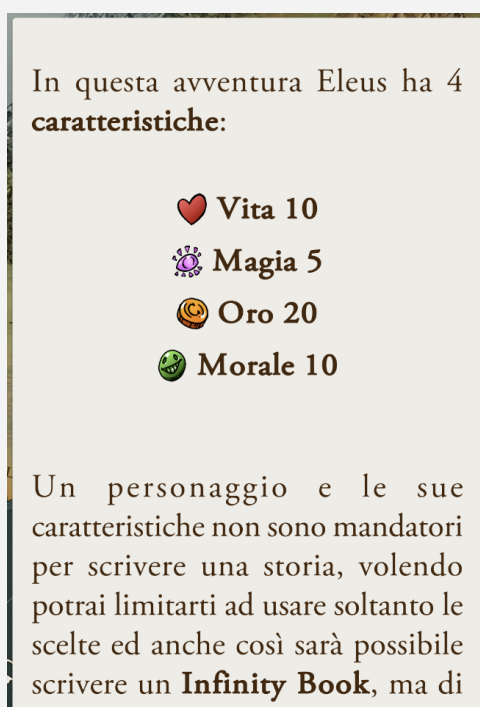
- With this command you will hide the graphic **but the character will remain associated with your player, so all game mechanics will still work.**

7 Text Formatting

7.1 Inserting icons in the text

It is possible to insert the icons of the character characteristics in the **paragraphs** and **choices** of the Infinity Book.

Example in the **paragraph**



Example in **choices**



Syntax:

```
<sprite name=CHARACTERISTIC_ID>
```

#Invisibility

```
@choice [Cast invisibility (2 <sprite name=magic>)]
```

Often you want to adjust the size of the inserted icon, to do so you add another command, called **size** (better illustrated in the next paragraph)

#Invisibility

```
@choice [Cast invisibility (2 <size=150><sprite name=magic></size>)]
```

In this way we are saying that the icon shall be 150% larger than normal, i.e. one and a half times its original size.

In addition to the feature icons you can use a special icon, called **dice**

#Test

```
@choice [<size=120><sprite name=dice></size> Resist the pain]
```

This will create the following choice button:



Remember:

- use the characteristic identifier
- if you use **<size=xxx>** remember the closing tag **</size>**

7.2 Using formatting tags

Here you will find the link to the [Full Documentation](#), while below we show you a series of examples and the most used commands:

7.2.1 Align the text

```
<align="right">Right</align>  
<align="center">Center</align>  
<align="left">Left</align>
```

		Right
	Center	
Left		

7.2.2 Color change

```
<color="red">Red</color>  
<color=#005500>Dark Green</color>  
<color=#0000FF>Blue</color>  
<color=#FF000088>Semitransparent Red</color>
```

Red	Dark Green	Blue
Semitransparent Red		

7.2.3 Bold and italic

The *<i>quick brown fox</i>* jumps over the **lazy dog**.

The *quick brown fox*
jumps over **the lazy dog**.

7.2.4 Font

Would you like `<font="Impact SDF">a different font?</font>` or just `<font="NotoSans" material="NotoSans Outline">a different material?`

Would you like **a different font?** or just a different material?

This command has some limitations, talk with us if you want to know more.

7.2.5 Indentation

1. `<indent=15%>It is useful for things like bullet points.</indent>`
2. `<indent=15%>It is handy.`

1. It is useful for things like bullet points.
2. It is handy.

7.2.6 Font size

`<size=100%>Echo <size=80%>Echo <size=60%>Echo <size=40%>Echo <size=20%>Echo`

Echo Echo Echo Echo Echo

7.2.7 Some commands (1)

Normal	
Bold	<code> </code>
<i>Italics</i>	<code><i> </i></code>
<u>Underline</u>	<code><u> </u></code>
Strikethrough	<code><s> </s></code>
Superscript - X ³ -	<code><sup> </sup></code>
Subscript - H ₂ O -	<code><sub> </sub></code>
Text Highlighting	<code><mark=#ffff8000> </mark></code>

7.2.8 Some commands (2)

ALL CAPS	<code><allcaps> </allcaps></code>
EXAMPLE OF SMALL CAPS.	<code><smallcaps> </smallcaps></code>
C h a r a c t e r s p a c i n g	<code><ospace=1em> </ospace></code>
Monospaced	<code><ospace=2em> </ospace></code>
Line height	<code><line-height=75> </line-height></code>
No Line-Break	<code><nobr> </nobr></code>
Size Smaller / Bigger	<code><size=75%> <size=1.5em> </size></code>

7.2.9 Advanced example

$$\underline{9x^3} + \underline{4x^2} + \underline{2x + 4} = 0$$
$$\underline{\textit{Propane}} = \underline{\text{C}_3\text{H}_8}$$

Whic is done as follows:

<u>9x^{<#00ff00>3</color>}</u> + <u>4x</u>^{<#00ff00>2</color>} +
<u>2x + 4 = 0</u>
<#ffff00><u><i><#ff8000>Propane</i></u></color> =
<u>C_{<#ffff00>3</color>}H_{<#ffff00>8</color>}</u>

8 FAQ

8.1 Why use Infinity Creator instead of other systems?

- **100% complete solution.** No other existing system (please tell us if you know it!) allows you to create a finalized app, publishable on a store with a graphic quality like ours.
- **Open system.** Most other systems are closed, i.e. you cannot add functionality. Infinity Creator is in full development and from time to time new mechanics and tools will be added to support the authors.
- **An author is not a programmer,** Infinity Creator was thought with the author at the center of everything. This means great flexibility for the writer and low technical complexity.
- **Infinity Creator has been translated in English,** this breaks down the entry barrier for those unfamiliar with other languages.

8.2 Is there a minimum length to respect for an Infinity Book?

Infinity Mundi plans to have a double product line, one for **Shorts** and one for **Longs**.

Shorts will consist of at least 150 paragraphs

Long will consist of at least 400 paragraphs

8.3 Validator does not display accents and other characters correctly

It's a problem of file formats and character encoding between systems.

Try opening the incriminated file with Google Docs and then saving it again by going to

File → **Share** → **Download** → **Text (.txt)**

8.4 Nested choices

Sometimes we need to use one conditional choice after another, for example we want to check if our protagonist has the badge Ofelia, in which case we send him directly to that meeting. If this condition should not be true we want to check (here is the second conditional!) if the reader has the lumberjack badge and in that case we send him to that meeting.

If neither is verified we will send him to the alternative paragraph.

Here is how it can be easily solved:

```
#start
@ignore-no-choice
;@ignore-no-choice only serves to avoid getting an error from the Validator when you don't
put choices in the paragraph

@if badge ofelia equals 1 @then #meet-ofelia @else #check-lumberjack

#meet-ofelia
blah blah blah blah blah
@choice [more...] #continue

#check-lumberjack
@ignore-no-choice

@if badge lumberjack equals 1 @then #meet-lumberjack @else #none

#meet-lumberjack
blah blah blah blah
@choice [more...] #continue

#none
blah blah blah blah
@choice [more...] #continue
```

Another example, we want to verify that the reader has exactly three objects (sword, trident and dagger)

```
#check-sword
@ignore-no-choice

@if sword inventory equal 1 @then #check-trident @else #check-failure

#check-trident
@ignore-no-choice
```

```
@if trident inventory equals 1 @then #check-dagger @else #check-failure
```

```
#check-dagger
```

```
@ignore-no-choice
```

```
@if inventory dagger equals 1 @then #you-have-all @else #control-loser
```

8.5 How do I disable some Validator warnings?

The **Infinity Validator** (don't you know it? No way! Run here www.infinitymundi.com/validator) is a very important tool for writing an **Infinity Book**. Sometimes some of the warnings it gives are excessive and so we can mute it. It must be clear that **you need to know what you're doing**, otherwise you risk to hide errors in your story!

In each paragraph you can enter the following commands:

- `@ignore-no-choice`, silences the warnings due to the fact that there are no choices in the paragraph
- `@ignore-dead-label`, silences warnings due to the fact that a certain paragraph is not (apparently!) reachable from other paragraphs

9 Log Version

9.1 Ver 1.9 (official)

- **Breaking change:** [Inventory 2.0](#)
- [Interactive Maps](#)
- [Revealing choices](#)
- **L'Infinity Validator supports these features ⇒ [Download](#).**

9.2 Ver 1.8

- [@choice-secret](#)

9.3 Ver 1.7

- **Breaking change:** files are renamed
 - [Capitolo.txt](#) => [Chapter.txt](#)
 - [Personaggi.txt](#) => [Characters.txt](#)
 - [Meta.txt](#) => [Meta.txt](#) (same)
 - [Nemici.txt](#) => [Enemies.txt](#)
- Mute some warnings from Validator [@ignore-no-choice](#), [@ignore-dead-label](#)
- Go back where you came from: [@choice-back](#)
- Multilingual interface support ([@language](#))
- Character's characteristics can be shown now with max 3 initials
- Minigame: [Spinner!](#)
- A complete module for combats! [Combat System!](#)
- [Music & SFX](#)
- Now you can add a maximum value for a [characteristic](#)
- Dynamic control of the story thanks to [VARS!](#)

9.4 Ver 1.6

- Character's characteristics can be shown with max 2 initials
- **L'Infinity Validator v1.11 supporta queste nuove funzioni**

9.5 Ver 1.5

- **Breaking change:** I distintivi vanno dichiarati nel file Meta ed il IGold utilizzo nel file Capitolo è cambiato, vedi [4.6 Distintivi \(@badge\)](#) e [5.3.3 Distintivi \(@badge-id\)](#)
- Ora le immagini in meta supportano l'opzione nascondi
- Il PARAGRAPH sui [tag di formattazione](#) è stato arricchito con degli esempi

L'Infinity Validator v1.6 supporta queste nuove funzioni

9.6 Ver 1.4

- **Breaking change:** Il comando [@input-text](#) e [@input-number](#) sono cambiati, ora accettano un testo come titolo della finestra di dialogo
- **Breaking change:** Il comando [@inventory](#) è cambiato, ora supporta singolari e plurali
- **Breaking change:** Le [Immagini usate](#) e gli [oggetti dell'inventario](#) vanno dichiarati nel file Meta per controlli d'errore più raffinati
- [Blackboard](#) rinnovata ed espansa
- [Whiteboard](#)
- [Nuovo documento di approfondimento su blackboard e whiteboard \(per i più smanettoni\)](#)

L'Infinity Validator v1.5 supporta queste nuove funzioni.

9.7 Ver 1.3

- [Se...allora...altrimenti](#), comandi condizionali per modificare entità e dirottare paragrafi
- [Blackboard](#), gestione delle informazioni nascoste
- [Ora è possibile fare i controlli con la RULE not_equal](#)
- [Scelte nascoste](#), nascondere una scelta
- [Options](#)
- [Finestra di dialogo adesso accetta anche un suggerimento](#)
- [Testo Condizionale](#), mostrare il testo sotto certe condizioni

L'Infinity Validator v1.4 supporta queste nuove funzioni.

9.8 Ver 1.2

- [Scelte a tempo](#)
- Aggiunto illustratore e musiche nei [metadati](#)
- **Breaking change:** Il comando [@endOfTheStory](#) è cambiato in modo incompatibile con le versioni precedenti.
- Aggiunta la schermata di [Punteggio Finale](#)
- [Acquisto della versione completa](#)

L'Infinity Validator v1.3 supporta queste nuove funzioni.

9.9 Ver 1.1

- [History](#)
- [Input-testo](#)
- [Inserimento di icone nel testo](#)

- I comandi [@afterChoice](#) e [@modifier](#) ora possono anche impostare il valore invece di modificarlo di una certa quantità.

9.10 Ver 1.0

- [Distintivi](#)
- [Metadati](#)
- [Comando @inventory svuota](#)
- [Minigame: lancio dei dadi!](#)

This document is a Work In
Progress

[Join the Facebook group
Infinity Mundi - Creators](#)

to follow the updates and
propose your ideas, we'll listen
to them!