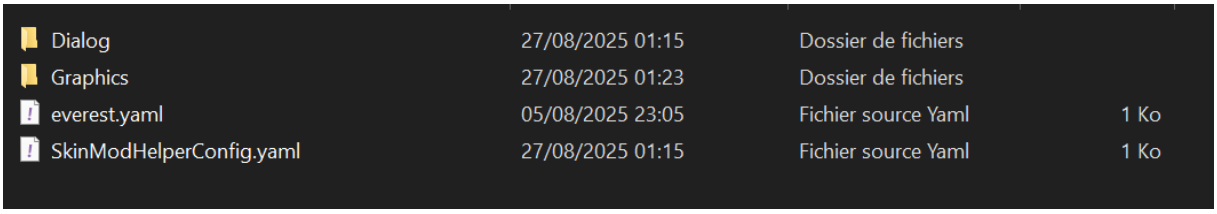
Create your own skin on *Celeste*

This documentation concerns the creation and organization of your skin on Celeste. If you do not know what to do or where to start, we will try to answer your question(s) here.

SUMMARY

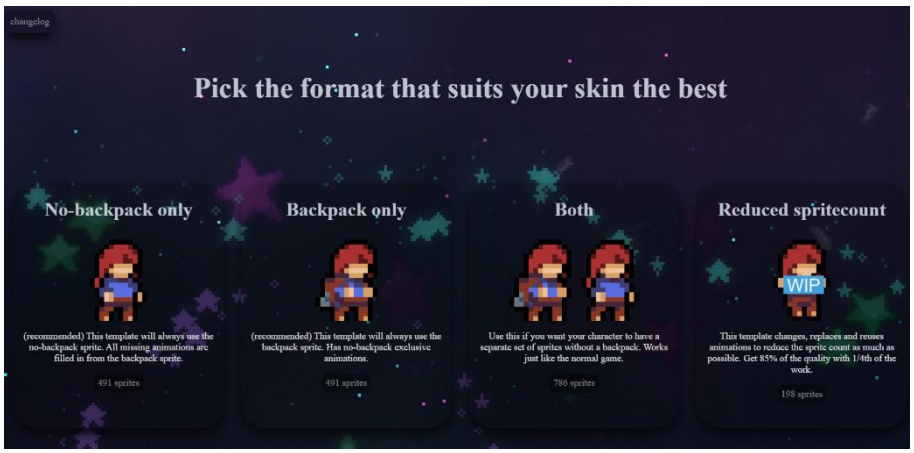
COMPOSITION OF THE SKIN FOLDER :

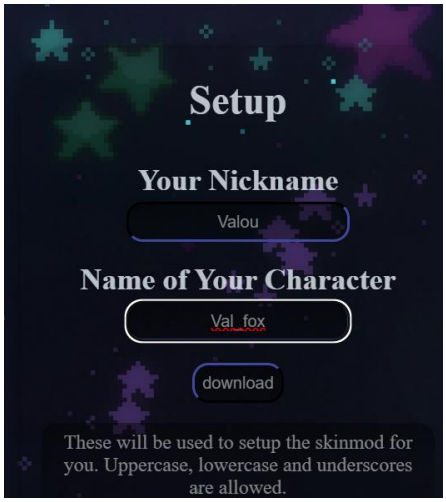


Your skin should be a zip file containing these folders and files:

To obtain this well-organized ZIP file, I recommend visiting this GitHub (Made by Kuksa), which will prepare a ZIP file for you with everything you need for your skin (some animations may be missing if you are making a skin with or without a backpack, so make sure you have everything you need !):

https://kuksattu.github.io/celeste/skinmod-template





On this GitHub, you will find a selection of four types of skins :

- 'No-backpack only' : which will create a ZIP file with Madeline's sprite without the backpack.
- 'Backpack only' : which will create a ZIP file of the sprites with backpack.
- 'Both' : which will create a ZIP file of the sprites with and without backpacks (which will be separate).
- 'Reduced Spritecount' : which will create a very small ZIP file with minimal effort required to create the skin.

For beginners, I recommend choosing the 'Reduced Spritecount' ZIP file, which will save you time. If you feel it is necessary, you can add other sprites to achieve a smoother rendering.

PLEASE NOTE: The ZIP file 'Reduced Spritecount' has been removed by its creator (it will be back later under better conditions, and this documentation will be updated when the time comes).

Your nickname and character name will be included in your skin folders. You will often see them between certain folders (in the example below, Valou/Val_fox/). To explain further,

'Valou/Val_fox/' translates to 'YourName/YourCharacter/' and is used to avoid conflicts with other sprites of the same name, but it also corresponds to the standard naming system used by the Celeste community for mod architecture.

First, we will look at the files '*everest.yaml*' and '*SkinModHelperConfig.yaml*' :

EVEREST.YAML :

```yaml
- Name: Val_Skinmod_Valou
  Version: 1.0.0
  Dependencies:
    - Name: Everest
      Version: 1.4465.0
    - Name: SkinModHelperPlus
      Version: 0.10.4
```

Just a file containing your skin's dependencies.

SKINMODHELPERCONFIG.YAML :

```yaml
- SkinName: "Valou_Val_fox"
  Player_List: true
  Silhouette_List: false
  Character_ID: "Valou_Val_fox"
  OtherSprite_Path: "Valou/Val_fox"
  Mod: "SkinModHelper_Grouping__Val_fox"

- SkinName: "Valou_Val_fox_playback"
  Player_List: false
  Silhouette_List: true
  Character_ID: "Valou_Val_fox_playback"
  OtherSprite_Path: "Valou/Val_fox_playback"
```

'SkinName' = the name of your skin

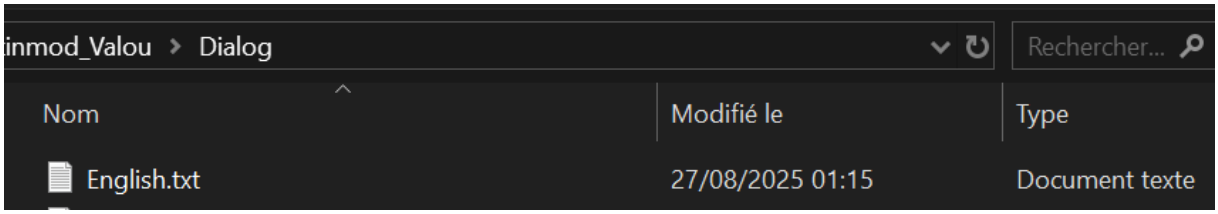
'Player_List' = activate if the skin is in the player skin list

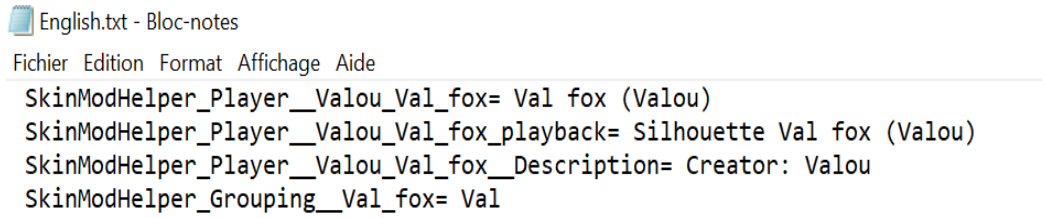
'Silhouette_List' = activate if the skin is in the silhouette skin list, also called 'playback' (**only** activate if you intend to make one)

'Character_ID' = name used in your .xml files

'OtherSprite_Path' = folder path leading to the second file 'Sprites.xml' for additional sprites.

« DIALOG » FOLDER :



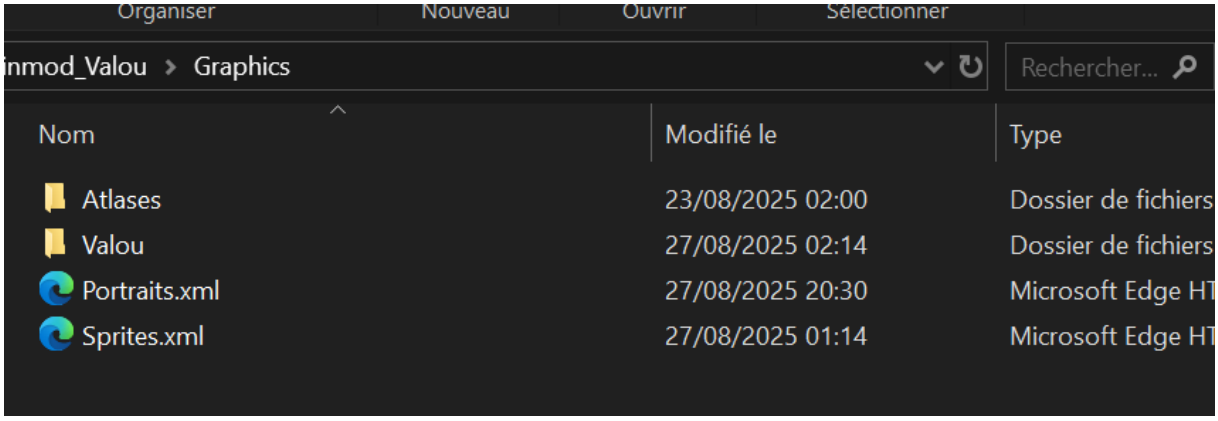


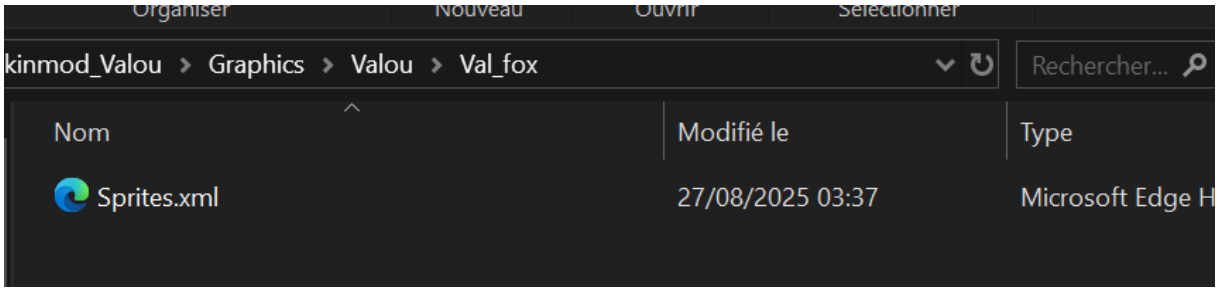
In this .txt file, you can name your skin types (normal or playback) :

- ***'SkinModHelper_Player__Valou_Val_fox'*** = player skin name
- ***'SkinModHelper_Player__Valou_Val_fox_playback'*** = playback skin name
- ***'SkinModHelper_Player__Valou_Val_fox__Description'*** = Description of the skin (name of the creator)
- ***'SkinModHelper_Grouping__Valou_Val_fox'*** = name of your skin mod

« GRAPHICS » FOLDER :

.xml files :





There are two types of .xml files: 'Portraits' (containing information about the faces and text boxes of the different characters) and 'Sprites' (containing information about each sprite in your skin). We will look at this information in more detail later in the document.

You will have noticed that there are two 'Sprites' files. The first, located in the "Graphics" folder, contains information about the animations in the base game and mods, as well as the 'playback' sprite.

The second contains information about sprites that use objects, for example, when Madeline uses the binoculars ('Lookout' animation).

Please note: if you put sprite animations using an object in the first Sprites.xml file, they will remain active even if you change skins, so keep in mind that you must separate sprites with and without objects.

The inside of an .xml file looks like this :

```
<Valou_Val_fox path="characters/Valou/Val_fox/" start="idle">
 <Origin x="16" y="32" />


 <Anim id="roll" path="roll" frames="0-12" delay="0.05"/>
 <Anim id="rollGetUp" path="roll" frames="12-16" delay=".1" goto="downed"/>
 <Loop id="downed" path="rollSit" delay="0.8" frames="0"/>

 <Loop id="idle_carry" path="idle_carry" delay="0.1" />
 <Anim id="runSlow_carry" path="run_carry" delay="0.07" />
 <Loop id="jumpSlow_carry" path="jump_carry" delay="0.1" frames="0,1" />
 <Anim id="fallSlow_carry" path="jump_carry" delay="0.1" frames="2,3" />
 <Anim id="pickUp" path="pickup" delay="0.06" />
 <Anim id="throw" path="throw" delay="0.06" goto="idle" />

 <Anim id="idle" path="idle" delay="0.1" goto="idle"/>

 <Anim id="idleA" path="idleA" delay="0.12" goto="idle"/>
 <Anim id="idleB" path="idleB" delay="0.16" goto="idle"/>
 <Anim id="idleC" path="idleC" delay="0.11" goto="idle"/>
```

```
<Metadata>
    <Frames path="idle" hair="x"/>
    <Frames path="idleA" hair="x"/>
    <Frames path="idleB" hair="x"/>
    <Frames path="idleC" hair="x"/>

    <Frames path="lookUp" hair="x"/>
    <Frames path="edge" hair="x"/>
    <Frames path="edge_back" hair="x"/>
    <Frames path="walk" hair="x"/>
    <Frames path="push" hair="x"/>
    <Frames path="runSlow" hair="x"/>
    <Frames path="runFast" hair="x"/>
    <Frames path="run_wind" hair="x"/>
    <Frames path="runStumble" hair="x"/>
    <Frames path="dash" hair="x"/>
    <Frames path="dreamDash" hair="x"/>
    <Frames path="jumpSlow" hair="x"/>
    <Frames path="jumpFast" hair="x"/>
    <Frames path="tired" hair="x"/>
    <Frames path="climb" hair="x"/>
    <Frames path="duck" hair="x"/>
    <Frames path="fallPose" hair="x"/>
    <Frames path="hug" hair="x"/>
    <Frames path="sleep" hair="x"/>
    <Frames path="flip" hair="x"/>
    <Frames path="dangling" hair="x"/>
    <Frames path="shaking" hair="x"/>
```

```
<Valou_Val_fox path="characters/Valou/Val_fox/" start="idle">
```

You have the path leading to your sprites starting directly from the 'Gameplay' folder (**do not forget the '/' at the end of your path**) and start with which animation your skin will begin with.

```
<Anim id="roll" path="roll" frames="0-12" delay="0.05"/>
```

Each line like this has an ID, which is the name of an animation also referenced in the game code itself, a path showing the folder path leading to the animation, and frames, which are the number of sprites for an animation (for example, for the 'roll' animation above, the frames range from sprite "roll0" to sprite 'roll12').

But these frames can be changed to give you other possibilities for your animations. You don't need to scroll through your animation from 0 to 12, you can do whatever you want :

```
<Anim id="bigFallRecover" path="bigFall" frames="5*5,6*4,7*3,8,8,9,9,10,10,10" delay="0.08" goto="swimIdle"/>
```

For example, in this animation with frames such as :

« 5*5, 6*4, 7*3, 8, 8, 9, 9, 10, 10, 10 »

We can translate this as :

« 5, 5, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 8, 8, 9, 9, 10, 10, 10 »

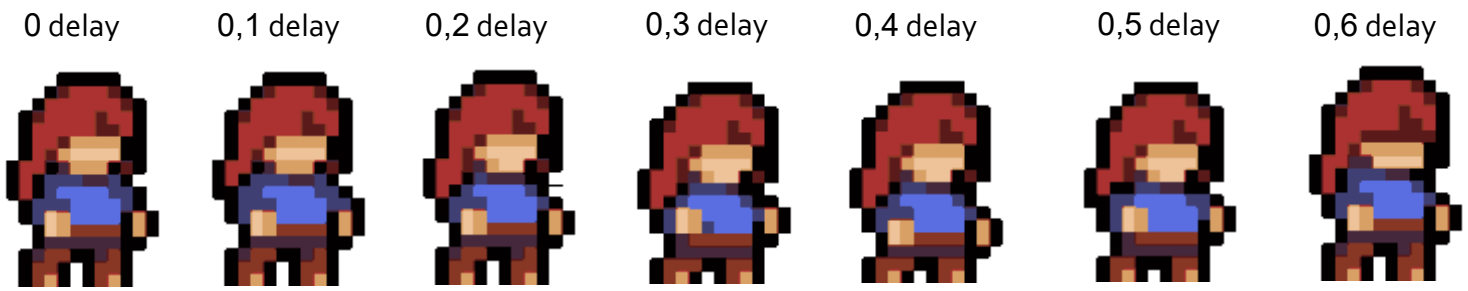
Meaning that out of the total of 19 frames :

- The first 5 are the sprite '5'
- The next 4 are the sprite '6'
- The next 3 are the sprite '7'
- The next 2 are the sprite '8'
- The next 2 are the sprite '9'
- The last 3 are the sprite '10'

The 'delay' is the time (in seconds) between each sprite in the animation (note that 0.1 delay = 10 FPS):

Example with the 'idle' animation having a 'delay' of 0.1 :

```
<Anim id="idle" path="idle" delay="0.1" goto="idle"/>
```




```
<Frames path="idle" hair="-2,4|-2,4|-2,4|-2,4|-2,4|-2,4|-2,4|-2,4|-2,4"/>
```

In the 'metadata' section, there is also a folder path and a 'hair' variable containing the x and y coordinates of the hair pixels in each frame of the animation. These coordinates are relative to the sprite's original coordinates (called 'Origin' in the .xml file) and are separated by a '|'. Lik above, you can see :

« -2,4|-2,4|-2,4|-2,4|-2,4|-2,4|-2,4|-2,4 |-2,4 »

You can view it like this :

« x, y|x, y|x, y|x, y|x, y|x, y|x, y|x, y |x, y »

The first coordinates are for frame 0, in this case the sprite 'idle0', and the next ones will be for 'idle1', "idle2" ... up to 'idle8'. Like this :

« *x, y* | *x, y* | *x, y* | *x, y* | *x, y* | *x, y* | *x, y* | *x, y* | *x, y* »

« ***frame*** 0 | ***frame*** 1 | ***frame*** 2 | ***frame*** 3 | ***frame*** 4 | ***frame*** 5 | ***frame*** 6 | ***frame*** 7 | ***frame*** 8 »

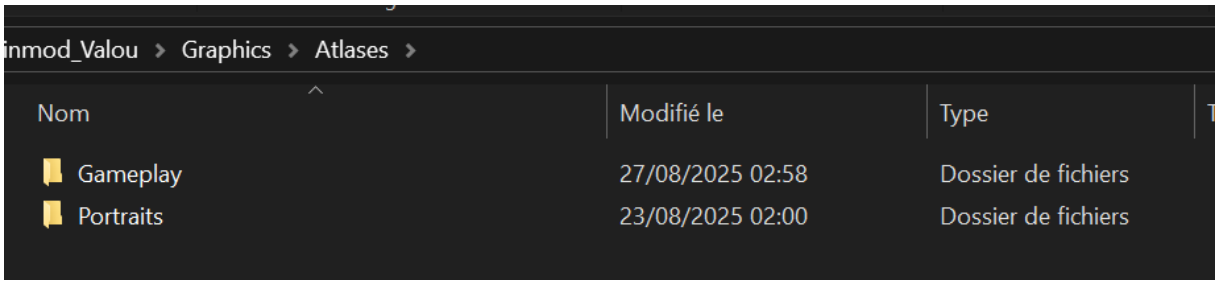
```
<Frames path="idle" hair="x"/>
```

If you put an "x" in place of the coordinates, the pixels will be removed.

```
goto="runFast"
```

Some lines have 'goto', which represents the animation that will play after the animation in question.

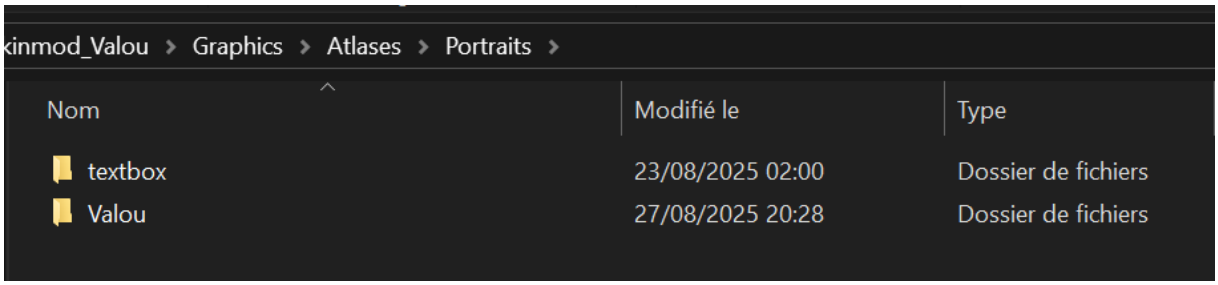
« ATLASES » FOLDER :



In the 'Atlases' folder, there are two folders : the 'Gameplay' folder, where all your skin's sprites will be located, and the 'Portraits' folder, where your skin's portrait images will be located (you are not required to have portraits for your skin).

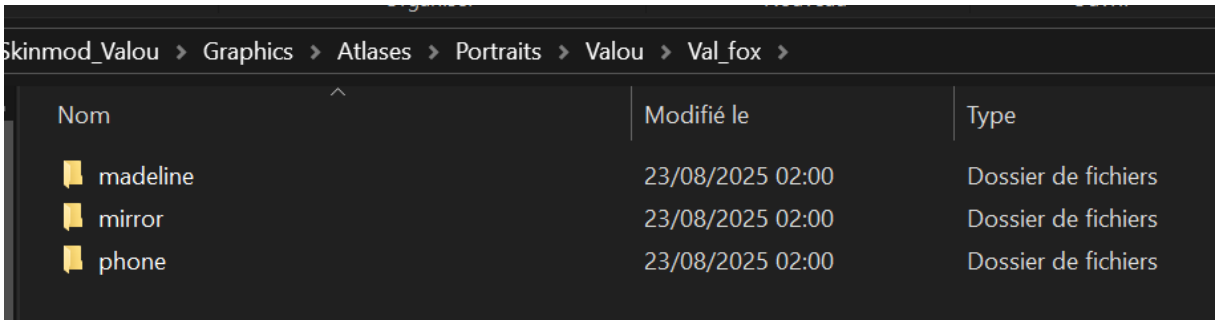
Let's quickly go over the 'Portraits' folder:

« PORTRAITS » FOLDER :



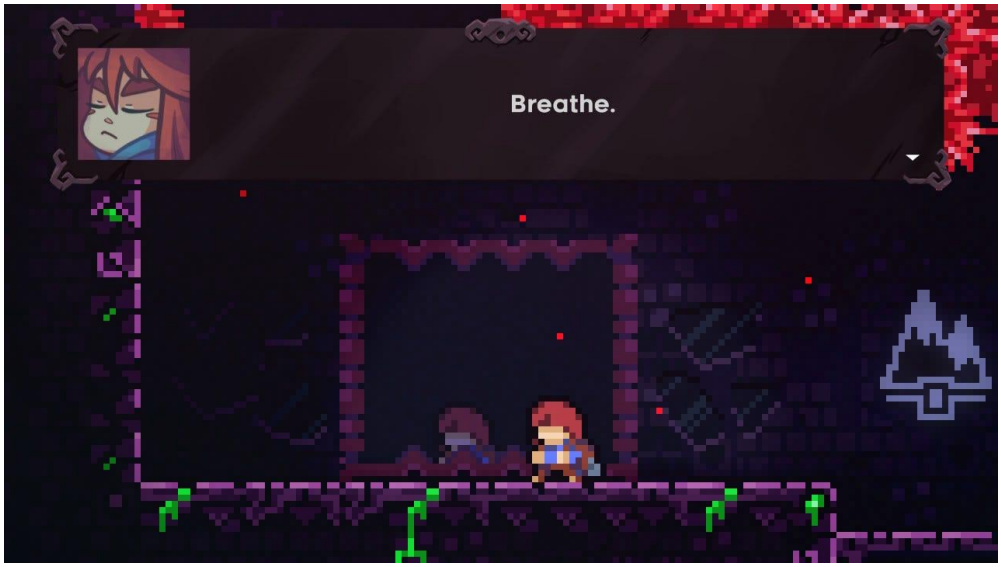
It contains a 'textbox' folder where you will find your character's dialogue box.

And the second folder corresponds to the one containing your character's facial expressions :



The ‘madeline’ folder for common expressions for your skin :

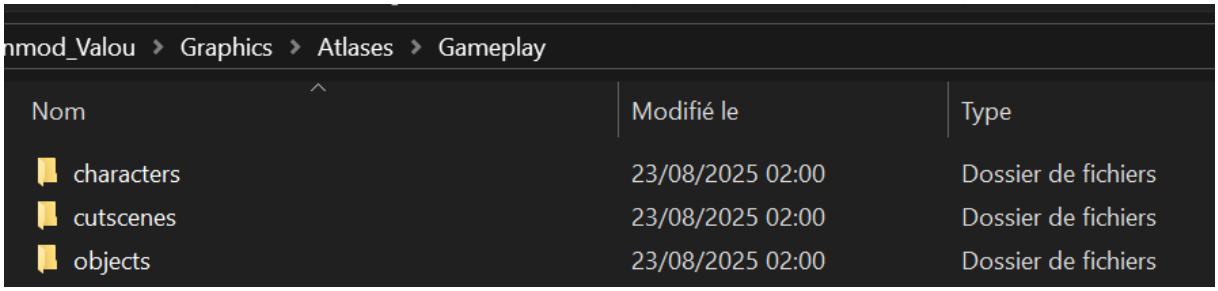
The ‘mirror’ folder for your skin expressions in Chapter 5 :



The ‘phone’ folder for your skin expressions at the end of Chapter 2 :

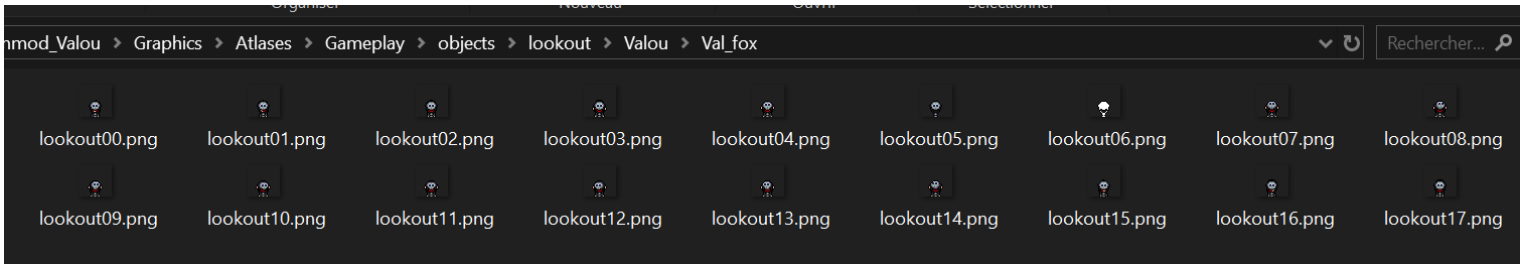
« GAMEPLAY » FOLDER :

Now let's move on to the 'Gameplay' section

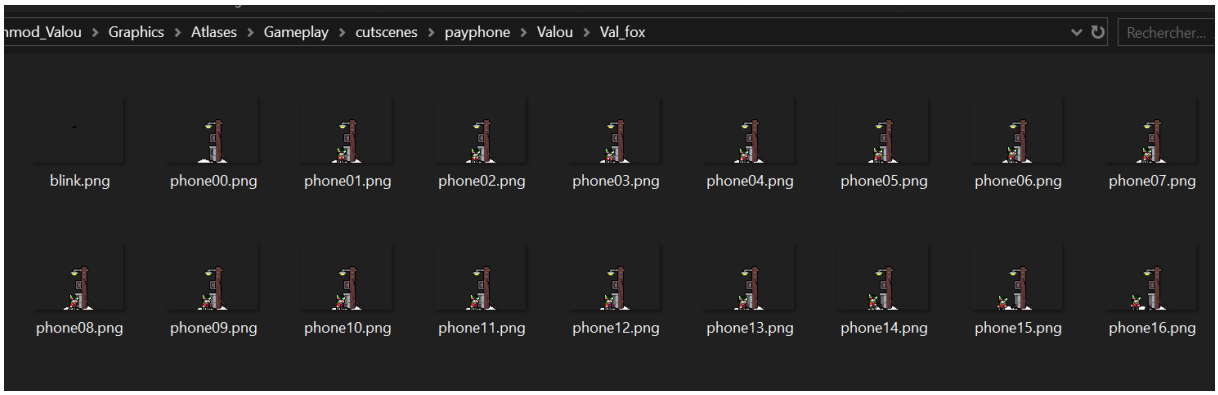


It contains a total of 3 files :

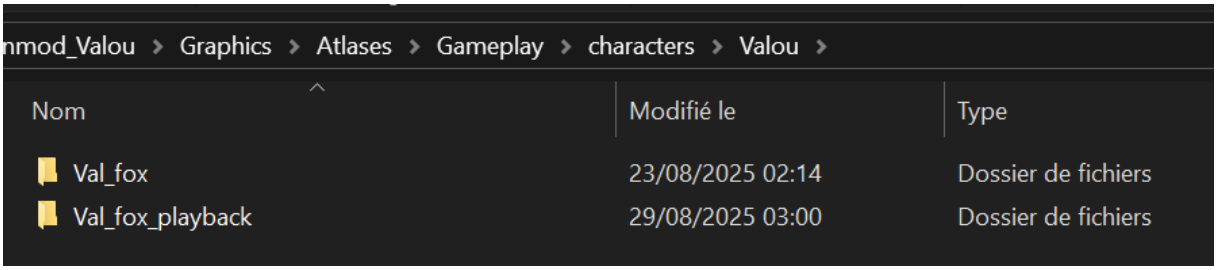
- An 'objects' folder containing the 'lookout' animation



- A 'cutscenes' folder with the phone animation from chapter 2A

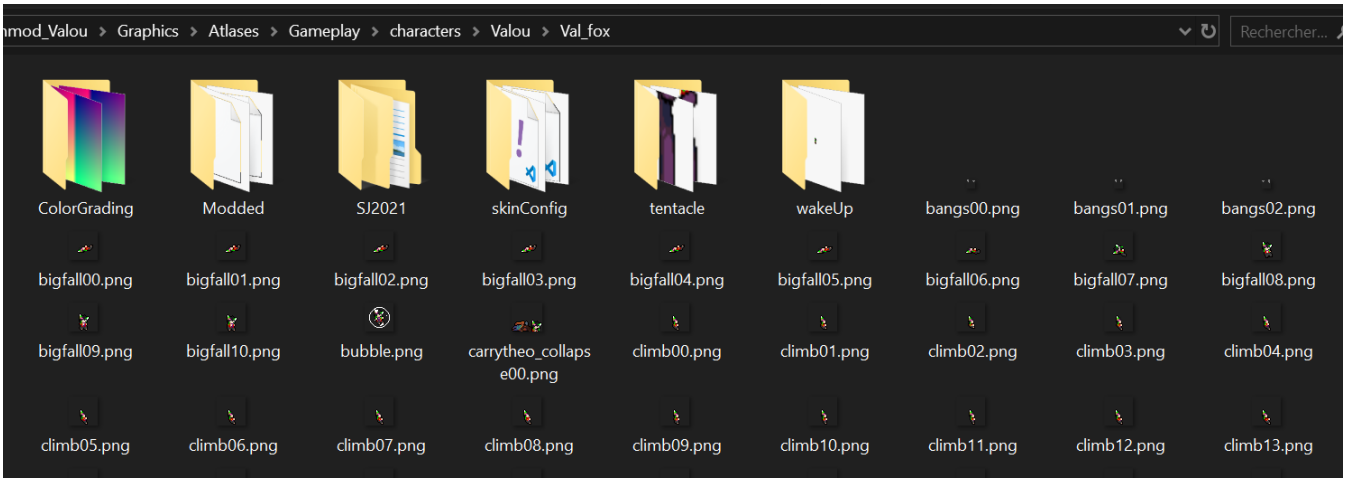


- A 'characters' folder with all the animations for the skin and playback (from the base game and mods)



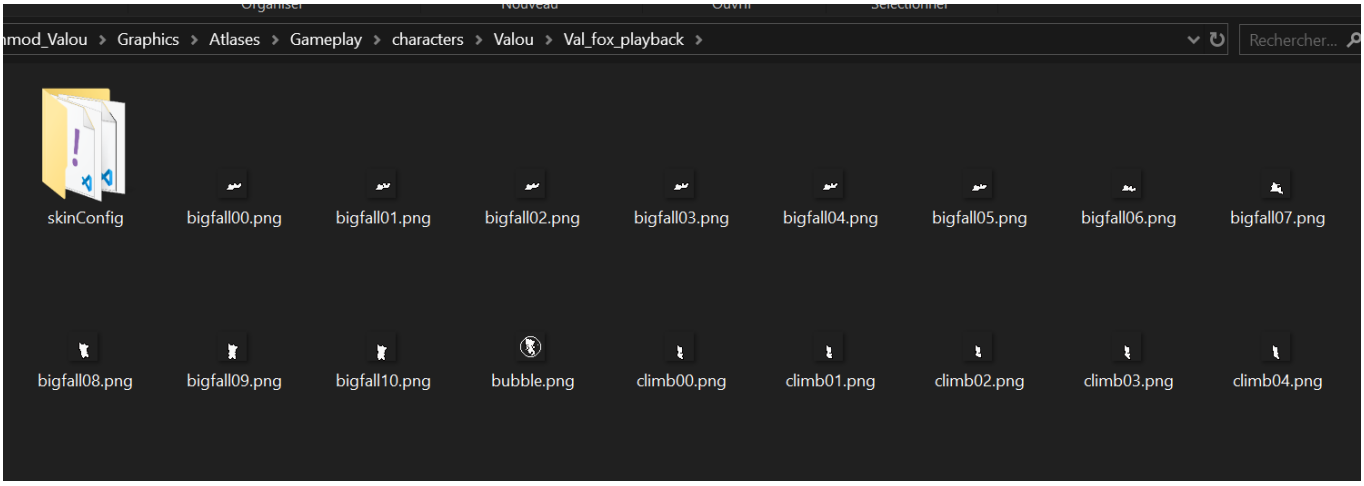
« CHARACTERS » FOLDER :

Classic skin :



Playback :

The screenshot of the folder I am going to show you is one I made myself, it will not be ready for you when you receive your ZIP file.



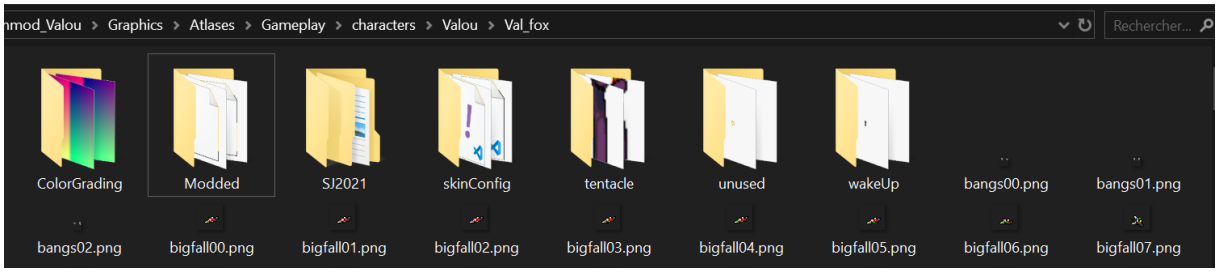
PLEASE NOTE: When you download your ZIP file, you will receive a folder containing a 'READ.me' file with the following information :

README.txt - Bloc-notes

Fichier Edition Format Affichage Aide

```
You can delete this file and folder if you aren't planning to add a playback skin.


A playback skin (also known as a silhouette skin) is the single-colored Madeline you see in the wavedash tutorial in Farewell.
Some maps also use the playback visuals. For example Chromatic Complex and Pinball Purgatory, both from the mod Strawberry Jam Collab.
(https://gamebanana.com/mods/424541)

Playback skins nearly double the image count so they're not included in this template to prevent unnecessary bloat.
After all, many people don't bother reskinning them.
Please join the Celeste discord server and ask for help in the #asset_making channel if you want to add the playback skin.
(https://discord.com/invite/celeste)

You can also look at existing skins with playback animations. I suggest looking at the Link skinmod by Linkdahl.
(https://gamebanana.com/mods/493129)

Feel free friend requesting me in discord so you can dm me if you don't want to join the server.
My handle is @kuksattu
```

Let's look at the classic skin file :



You must have the following folders:

- **'ColorGrading'** (which will manage the colour change of your dash on your skin)
- **'Modded'** (this one is not mandatory, but it contains sprites from various helpers/mods)
- **'SJ2021'** (this is not mandatory and was added manually by me; it contains animations from the Strawberry Jam Paint level)
- **'skinConfig'** (which contains two .cs files that manage your skin's graphics settings; we will come back to this later)
- **'unused'** (which contains unused frames)
- And the **'tentacle'** and **'wakeUp'** folders contain sprites from the base game for your skin (chapter 6 with Badeline's tentacles and Madeline's awakening from chapter 2)

The rest of the files are sprites from the base game for your skin, which you can also modify as you see fit.

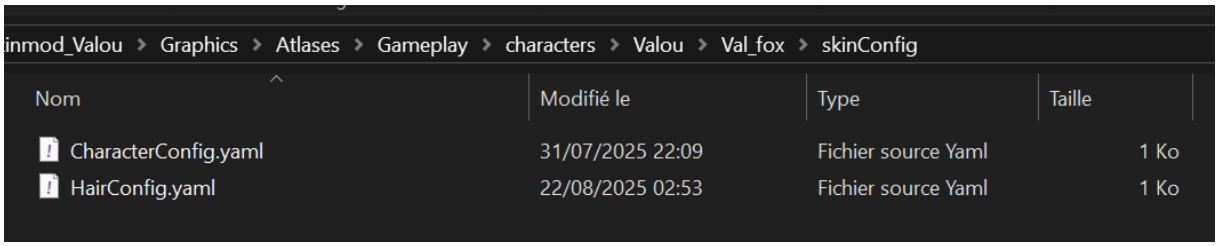
Before diving into the **'skinConfig'** and **'Colorgrading'** folders, I would like to tell you about this file called 'bang'.

Bang

Sprite (idle)

On Celeste, Madeline's hair is separate from her body. If you want to modify the hair base, then modify the 'bang' file, or you can directly place the hair on the skin's base body (some skins have the hair directly on the sprite, leave them as they are). ***But be careful, if you place the hair directly on the sprite, it will not react to wind and directional movements.***

SKIN CONFIGURATION :



Now, the « SkinConfig » folder

It contains two files :

- ***CharacterConfig***

```
SilhouetteMode: false          # If enabled, the player will be tinted to the hair color.
LowStaminaFlashColor: "FF0000"  # Specifies the color that the player will flash when their stamina is below 20.
LowStaminaFlashHair: false      # If enabled, low stamina flash will color the hair in addition to the player.
# DeathParticleColor: "FFFFFF"  # Uncomment to take effect. Changes the color of the 8 particles that flash when you die. Normally
```

'SilhouetteMode': The sprite will be tinted with the hair color if set to 'true' (this is why the skin playback sprites are white).

'LowStaminaFlashColor': Specifies the colour that will flash when the player's stamina is low/exhausted.

'LowStaminaFlashHair': Indicates low stamina, which will colour the hair in addition to the sprite.

'DeathParticleColour': Specifies the colour of the skin's particles when it dies.

```
# These are the idle animations when a map has no core mode or the core mod is set to cold.
IdleColdOptions:
- A, 3          # First the identifier, then the weight of the probability
- B, 5          # This corresponds to <Anim id="idleB" ... />
- C, 1          # This corresponds to <Anim id="idleC" ... />
```

'IdleColdOptions': Represents the probability that animation 'idle' A, B, or C will play when Core's 'cold' mode is on the map (chapter 8) or when there is no mode:

- 3 out of 9 chances of having 'idleA ' animation
- 5 out of 9 chances of having the 'idleB' animation
- 1 out of 9 chances of having the 'idleC' animation

Chapter 8 'Cold' Mode :

```
# Uncomment these two lines to enable the custom idle animations defined inside the Sprites.xml.
# - D, 1        # You can add more animations like this. This corresponds to <Anim id="idleD" ... />
# - Sit, 1      # The identifier doesn't have to be a single letter. This corresponds to <Anim id="idleSit" .
```

These two lines are there if you wish to enable custom animations that you can delete (information on Sprites.xml in the 'Graphics' folder) :

```
<!-- These two animations are custom idle animations. They don't currently do anything. -->
<!-- If you want to make them appear ingame, go to the CharacterConfig.yaml and follow the instructions there. -->
<Anim id="idleD" path="lookUp" delay="0.1" frames="0-4,5*10,5,4,3,2,1" goto="idle"/>
<Anim id="idleSit" path="sitDown" delay="0.1" frames="8-14,15*9999" goto="idle"/>
```

```
# These are the idle animations when the core mode is set to hot.
IdleWarmOptions:
- A, 3
- B, 5
- C, 1
```

'IdleWarmOptions': Represents the probability that animation 'idle' A, B, or C will play when Core's 'warm' mode is active on the map (chapter 8):

- 3 out of 9 chances of having the 'idleA' animation
- 5 out of 9 chances of having the 'idleB' animation
- 1 out of 9 chances of having the 'idleC' animation

Chapter 8 'Warm' Mode :

```
# The chance for an idle animation to play every time the idle animation loop finishes.
# The default value is 0.2=20%. This template makes it a lower 5% because vanilla 20% plays idle animations basically constantly.
IdleAnimationChance: 0.05
```

'IdleAnimationChance': Represents the probability that animations 'idleA', 'idleB', 'idleC' will be played at the end of the 'idle' animation.

- ***HairConfig***

```yaml
# Please use an online yaml validator if you encounter issues.
HairFlash: true # Enables the white flash that happens when your dash count changes.
HairFloatingDashCount: 2 # The dash count at which the hair starts foating like when you have two dashes. -1 to make the hair never
OutlineColor: "040000" # This specifies the hair outline color, not the sprite outline of the player.

# If your custom bangs or hair texture is larger than 10x10 pixels, it will get offset ingame.
# Use these offsets to move the hair back into the right position.
BangsOffset: 0, 0
HairOffset: 0, 0
```

```yaml
# Here you change the hair color/scale/length
HairAttrWithDashes:
- Dashes: -1        # -1 Means the feather tail.
  Scale: 1, 0.357   # The scale of the hair from start to end
  Length: 7

- Dashes: 0
  Color: "46C3FF"
  Scale: 1, 0.4375
  Length: 4

- Dashes: 1
  Color: "AC3232"
  Scale: 1, 0.4375
  Length: 4

- Dashes: 2
  Color: "FF7BFF"
  Scale: 1, 0.4375
  Length: 5
```

```yaml
- Dashes: 3
  Color: "FF7BFF"
  Scale: 1, 0.4375
  Length: 5
  SegmentAttrs:
  - Segment: 0
    Color: "FF7BFF"
  - Segment: 1
    Color: "B359C4"
  - Segment: 2
    Color: "733688"
  - Segment: 3
    Color: "B359C4"
  - Segment: 4
    Color: "FF7BFF"
```

'HairFlash': Activates the white flash that appears when your dash count changes.

'HairFloatingDashCount': Specifies the number of dashes at which the skin's hair floats (such as when Madeline has two dashes).

'OutlineColor': Specifies the colour of the hair outline, not the sprite.

'BangsOffset': Allows you to change the position of the 'bangs' sprite (to be done if it exceeds the size of 10x10).

'HairOffset': Allows you to change the position of the skin's hair (to be done if it exceeds the size of 10x10).

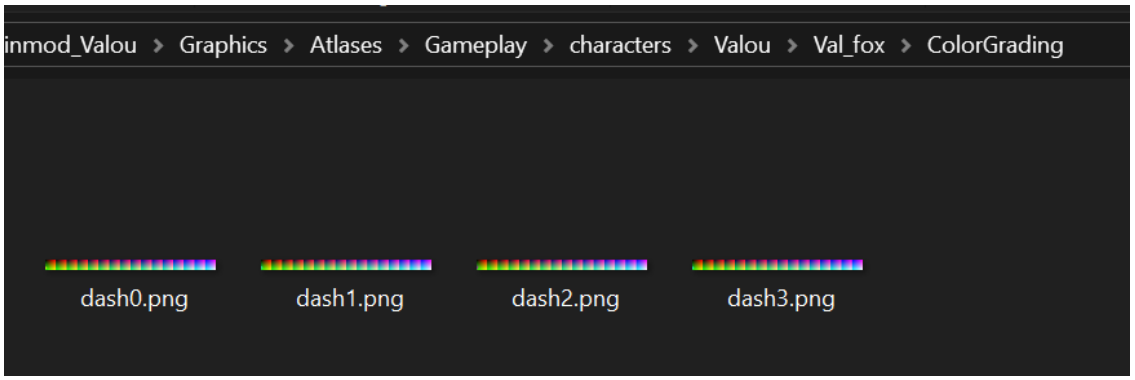
'Dashes': Represents the number of dashes on the skin ('-1' for feather, '0' for no dashes, etc.).

'Color': Changes the colour of your skin's hair and dash (visual trail) by acting as a filter (in HEX code) based on your number of dashes.

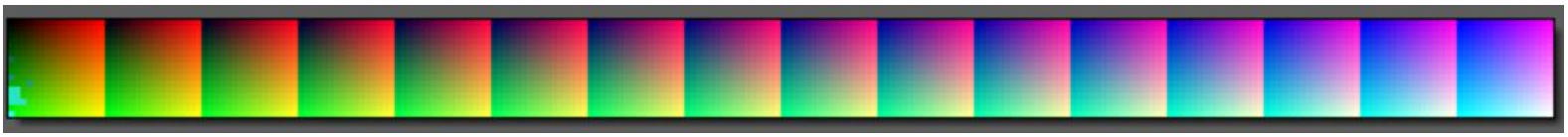
'Scale': Changes the size of your skin's hair (from start to finish) based on your number of dashes.

'Length': Changes the length of your skin's hair based on your number of dashes.

COLORGRADING :



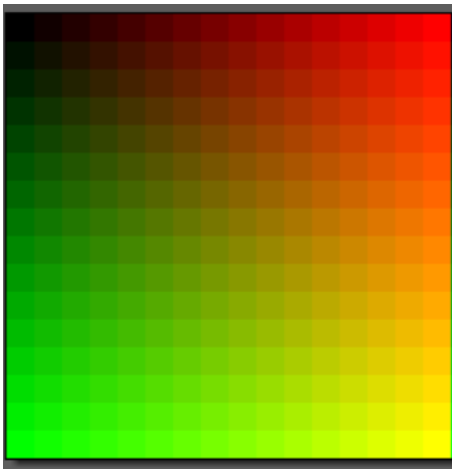
The colorgrading is a very important step in creating your skin.



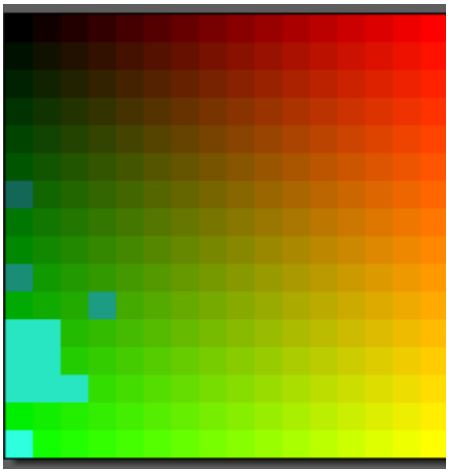
The colorgrading is a colour palette that will replace the colours on your skin with the modified colours.

Here is an example :

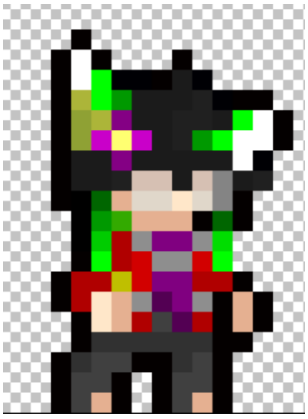
Colorgrading empty, unchanged



Colorgrading with modification



Sprite in your ZIP file



Sprite in the game

The colorgrading is simply a palette on which you place the colours you want to see instead of other colours on your skin (and this applies to each dash).

The colours in the basic palette (empty colorgrading, without modification) that you put on your skin will therefore be replaced in the game by the new colours you have placed on the palette (colorgrading with modification).

Don't forget to use the correct name for your colorgrading file ! ('dash1' for when your skin only has one dash, etc. See the screenshot of the folder above).

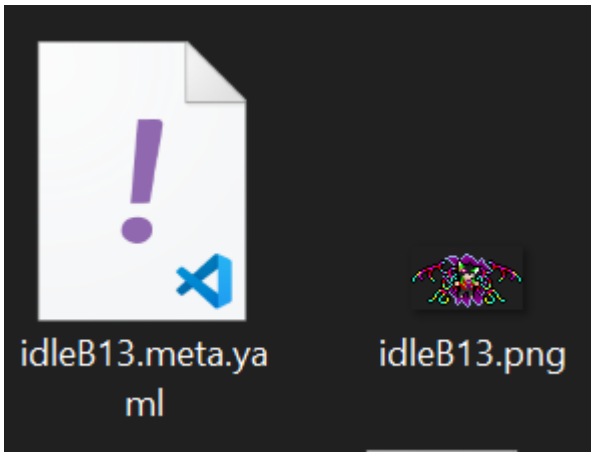
Please note: Some helper/mod and vanilla animations may not take colorgrading into account, so you will need to apply the colour of your dash directly to the sprite.

POSITION FRAMES ANIMATION :

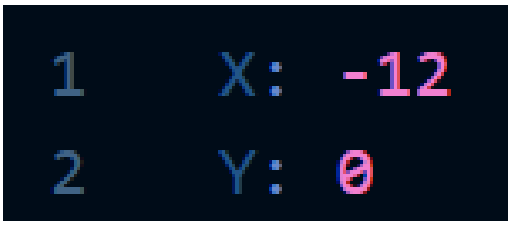
This is a specific point, but I had to mention it here. If you want to change the position of a specific sprite in an animation, you can !

Let me explain: if, for example, you want to change the position of frame 13 of the 'idleB' animation, you simply need to have a '.meta.yaml' file with the name of your frame (in this case, 'idleB13.meta.yaml') and enter the number of pixels by which you want to shift your sprite :

Inside your folder with your sprites



Inside the '.meta.yaml' file



Here, the sprite 'idleB13' will only be shifted 12 pixels to the left.

Please note: Your '.meta.yaml' file must be in the same folder as the sprite in question.

SKIN MOD HELPER PLUS SETTINGS :

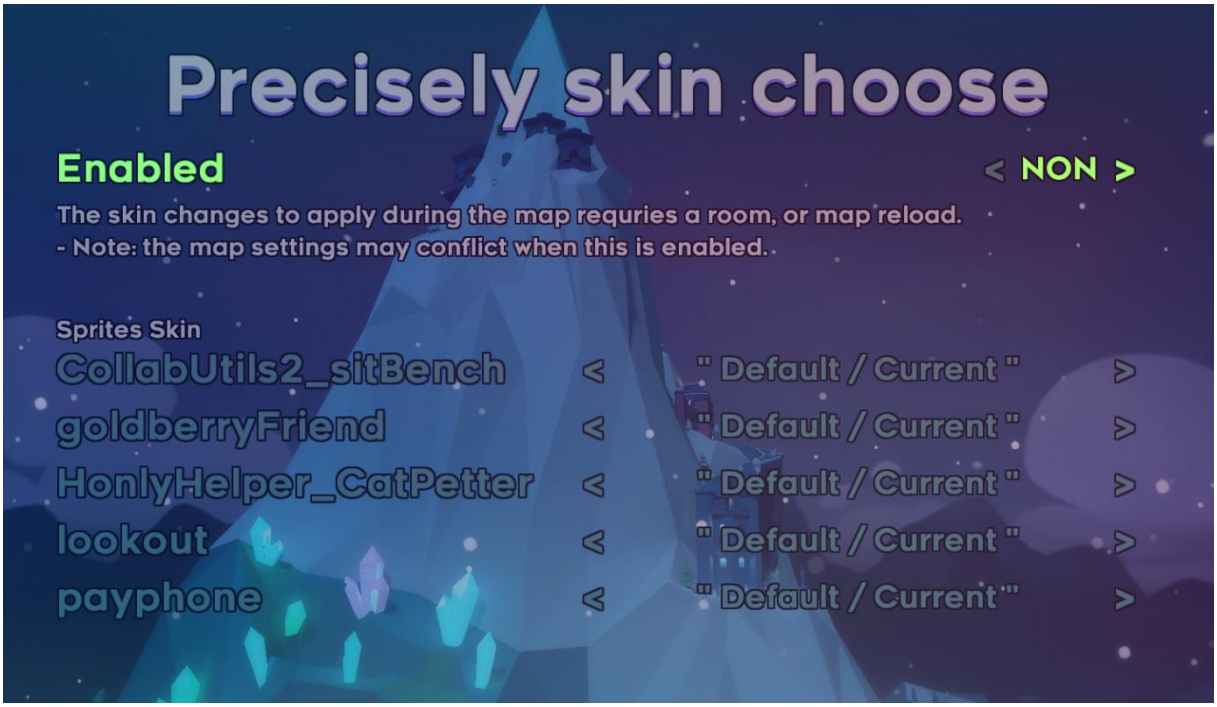
Variant	Otherself variant	Silhouette variant

https://gamebanana.com/mods/53734

PS: I don't have one for my skin (I'll leave you the link to the mod (made by Misirter E) so you can see for yourself).

« ADVANCED OPTIONS » :

The settings below are for choosing whether you want to use specific animations or the default ones (those in your skin folders).



CELESTE NET :

'Celeste NET' is a mod that allows you to play Celeste in multiplayer mode. You can also show your skin to your friends !

All you need is for the person playing with you to have your ZIP file so that Celeste NET can recognize the skin and display it on their side.

SOFTWARE RECOMMENDATIONS :

Little side note, if you don't really know where to start when it comes to creating your skin, here are some free software programs for creating sprites and other things :

- Paint.Net
- Krita
- Piskel
- GraphicsGale