

DANGANRONPA ONLINE

How to make your own character in 2024

aka.

“How do i make character please help i am in very serius rp”

Guide by Chrissy

Last updated March 5th, 2024

Table of Content

- The Basics
- Setting up the barebones folder
- Making sprites
- The char.ini
- The sounds.ini
- Making buttons

THE BASICS

So, you're new to DRO and want to make your very own character. In order to even start on this journey, you should probably know where you can even find characters - because that's where you will have to put yours as well.

First, go to where you have your DRO client installed, and see the folder where the .exe is placed too. It may look something like this here (You may have less though, my folder is still cluttered with stuff from many older versions):

Desktop > Danganronpa Online > !Danganronpa Online					!Danganro
Name	Änderungsdatum	Typ	Größe		
audio	25-Dec-23 17:40	Dateiordner			
base	23-Feb-24 22:20	Dateiordner			
bearer	25-Dec-23 17:40	Dateiordner			
config	30-Nov-22 22:20	Dateiordner			
designer	05-Feb-23 22:30	Dateiordner			
essentials	18-May-20 20:36	Dateiordner			
iconengines	25-Dec-23 17:40	Dateiordner			
imageformats	25-Dec-23 17:40	Dateiordner			
mediaservice	25-Dec-23 17:40	Dateiordner			
multimedia	05-Feb-23 22:30	Dateiordner			
networkinformation	05-Feb-23 22:30	Dateiordner			
packages	08-Jul-23 12:08	Dateiordner			
platforminputcontexts	06-Oct-20 05:18	Dateiordner			
platforms	25-Dec-23 17:40	Dateiordner			
playlistformats	25-Dec-23 17:40	Dateiordner			
qmltooling	09-Jun-22 18:42	Dateiordner			
scenegraph	06-Oct-20 05:18	Dateiordner			
styles	25-Dec-23 17:40	Dateiordner			
tls	05-Feb-23 22:30	Dateiordner			
translations	09-Jun-22 18:42	Dateiordner			
bass.dll	27-Aug-22 17:31	Anwendungserwe...	252 k		
bassopus.dll	27-Aug-22 17:31	Anwendungserwe...	118 k		
config.ini	26-Oct-23 21:56	Konfigurationsein...	3 k		
d3dcompiler_47.dll	27-Aug-22 17:31	Anwendungserwe...	4,802 k		
DanganronpaOnline.exe	25-Dec-23 05:01	Anwendung	1,412 k		
desktop.ini	24-Jun-19 13:25	Konfigurationsein...	1 k		
discord-rpc.dll	27-Aug-22 17:31	Anwendungserwe...	391 k		
Hardcore Mode Details.txt	20-Jul-18 16:54	Textdokument	1 k		
libcrypto-1_1.dll	08-Jun-22 00:08	Anwendungserwe...	2,466 k		
libcrypto-1_1-x64.dll	27-Aug-22 17:32	Anwendungserwe...	2,799 k		
libEGL.dll	27-Aug-22 17:32	Anwendungserwe...	25 k		
libgcc_s_dw2-1.dll	08-Jun-22 00:08	Anwendungserwe...	114 k		
libGLSv2.dll	27-Aug-22 17:32	Anwendungserwe...	3,306 k		
libssl-1_1.dll	08-Jun-22 00:08	Anwendungserwe...	520 k		
libssl-1_1-x64.dll	27-Aug-22 17:32	Anwendungserwe...	674 k		
libstdc++-6.dll	08-Jun-22 00:08	Anwendungserwe...	1,521 k		
libwinpthread-1.dll	08-Jun-22 00:08	Anwendungserwe...	46 k		
mk2-client.dll	05-Feb-23 12:24	Anwendungserwe...	207 k		
mk2-shared.dll	05-Feb-23 12:24	Anwendungserwe...	1,755 k		
opengl32.dll	23-Oct-18 22:55	Anwendungserwe...	1,650 k		
openal32sw.dll	27-Aug-22 17:32	Anwendungserwe...	20.433 k		

As you can see, my own file path for this is on the Desktop in the folder I have named Danganronpa Online, within which I have named another folder !Danganronpa Online. However, this is just my file path, and yours can be different and that's perfectly fine, this file path is just to give you a general idea of where you may roughly find this.

I have already highlighted the “base” folder in the screenshot above. This is where you need to go, and entering that folder you will be greeted with something like this:

 background	09-Jul-23 08:19	Dateiordner
 characters	28-Feb-24 18:43	Dateiordner
 configs	12-Sep-22 17:59	Dateiordner
 effects	25-Oct-23 18:10	Dateiordner
 fonts	25-Dec-23 17:40	Dateiordner
 logs	23-Feb-24 20:33	Dateiordner
 misc	12-Sep-22 17:59	Dateiordner
 notes	12-Sep-22 17:59	Dateiordner
 shouts	25-Oct-23 18:10	Dateiordner
 sounds	12-Sep-22 17:57	Dateiordner
 themes	25-Dec-23 17:40	Dateiordner
 callwords.ini	18-Jul-21 22:27	Konfigurationsein...
 config.ini	23-Feb-24 22:20	Konfigurationsein...
 favorite_servers.ini	23-Feb-24 18:39	Konfigurationsein...
 server_browser.ini	23-Feb-24 18:39	Konfigurationsein...
 serverlist.txt	07-Aug-22 17:51	Textdokument

Again, you may or may not be missing some of these and that’s okay, just in general assume there may be some obsolete stuff in my own folders that you probably won’t need anyway so don’t worry about it.

What’s important is the “characters” folder that I have once again highlighted. Enter that folder, and that’s where you will find all the characters that you have available to you.

18	18-May-20 23:10	Dateiordner
25ji Miku_PN	13-Feb-23 16:44	Dateiordner
707(MM)_TD	29-Jan-21 19:34	Dateiordner
Aaravi Mishra (Summer)	29-Nov-21 03:07	Dateiordner
Absen Mensio_DEMO	18-May-20 23:10	Dateiordner
Absen Mensio_DEMO2	09-May-21 17:04	Dateiordner
Ace Markey	20-Mar-22 09:56	Dateiordner
Adachi_DRO	26-Jul-22 18:00	Dateiordner
Aether	01-Mar-22 18:01	Dateiordner
Ai_PuyoTet	18-May-20 23:10	Dateiordner
Aigis (P3)	15-Jun-23 16:11	Dateiordner
Aigis Remake_HD	25-Jan-21 04:16	Dateiordner
Aigis_HD	25-Dec-20 17:41	Dateiordner
Aiko Umesawa	23-Apr-23 09:54	Dateiordner
Aiko Umesawa_HD	12-Sep-22 17:58	Dateiordner
Aiko_HD	12-Sep-22 17:58	Dateiordner
Aiko2.0	18-May-20 23:10	Dateiordner
Aisaka Misuzu_HD	18-May-20 23:10	Dateiordner
Aisaka Taiga_HD	18-May-20 23:10	Dateiordner
Akako Izumi_HD	12-Jul-20 17:34	Dateiordner
Akane Kotonoha_HD	19-Jul-20 17:16	Dateiordner
Akane Kurashiki_HD	18-May-20 23:10	Dateiordner
Akane Owari	23-Apr-23 09:54	Dateiordner

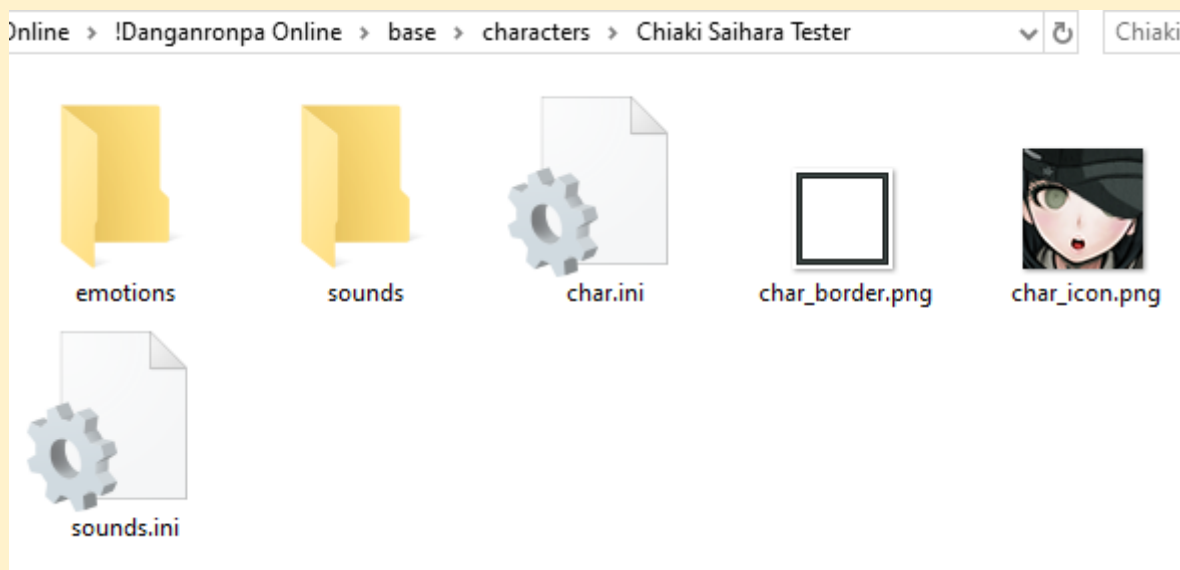
This is where you will put your own character as well. Now that you know where to go, we can start making a new character proper from here.

SETTING UP THE BAREBONES FOLDER

You should at least know how to make a new folder. Just make a new one right here in the characters folder. I will demonstrate everything from here using a tester copy of one of my custom characters, Chiaki Saihara:

Desktop > Danganronpa Online > !Danganronpa Online > base > characters		
Name	Änderungsdatum	Typ
Charuzu_HD	18-May-20 23:12	Dateiordner
Chef Hatchet	06-Feb-21 21:27	Dateiordner
Chef_Hatchet_HD	24-Apr-21 18:05	Dateiordner
Chen_HD	26-Jul-22 18:00	Dateiordner
Chesed_CF	30-Sep-21 15:13	Dateiordner
Chester_WC	10-Feb-22 22:15	Dateiordner
Chi_Aki_HD	18-May-20 23:12	Dateiordner
Chiaki But Mood_HD	18-May-20 23:12	Dateiordner
Chiaki Nanami	29-Apr-23 10:09	Dateiordner
Chiaki Nanami DR3_HD	12-Sep-22 17:58	Dateiordner
Chiaki Saihara	09-Aug-20 00:27	Dateiordner
Chiaki Saihara Tester	28-Feb-24 18:44	Dateiordner
Chiaki_HD	12-Sep-22 17:58	Dateiordner

Look, I made a new folder named “Chiaki Saihara Tester” right there in base/characters. What I have already put in there are these:

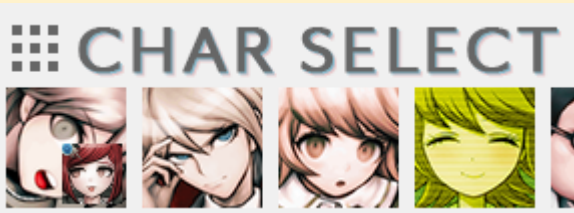


The “emotions” folder: This is where buttons will go. We will tackle that later.

The “sounds” folder: This is where you can put custom sound effects for your character, if you have any. If you do, we will go over this later. If you don’t, you may not even need this folder and can omit it from your character.

char.ini: This is where you will declare your character’s data and sprites and some such for the client to understand. We will tackle this one later also.

char_icon: The image that will appear for your character in the character select roster. Or, well, if your character is a custom and you swapped to them on a different character, they will appear in the corner of that same character’s roster image, like with Makoto here, who I currently have swapped into Himiko:

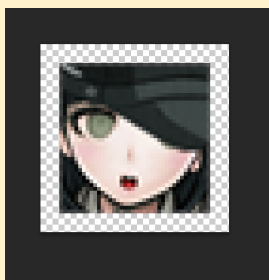


The `char_icon` needs to be a .png image and is usually 60x60 pixels big.

char_border: This is a border that will appear around your `char_icon` inside the client's list of present characters in your area.

It should be a .png image with a default size of 50x50 pixels and should be drawn to fit around a version of your `char_icon` that has been scaled down to 40x40 pixels.

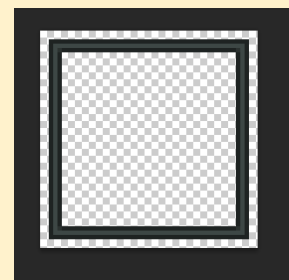
As I actually had no example of a `char_border` yet, I had to make a new one myself, so I can show you what I did for my border in Photoshop:



Made a new 50x50px file, copied my `char_icon` into it and scaled it down to 40x40px, then centered it



Drew a tight border around it on a new layer
(You don't have to, you're free to make your border free and less tight around the `char_icon`, this is just what I like)



Removed the layer the `char_icon` was on to see what my border looks like. I'm happy with this one for demonstration purposes, so this is what I saved.

Inside the client, this is where you will ultimately see that `char_border`, using the 1.5 default client (SDRO2) because that is literally the only one I have right now that can actually show this:



sounds.ini: This is where you can declare sound effects for your character's sound effect list. This is another one we will tackle later.

I know this is a lot of stuff to tackle later, but trust me, you will want at least the emotions folder and the `char.ini` in here from the start just to be set up, the `sounds.ini` is also helpful to already have in here. I guess you can put the `char_icon` in whenever, but I just like to already have an icon in there.

You can copy and paste all these from a different character folder, they should all already have these things by default. Honestly though you can just make the emotions folder a new folder anyway, as when you copy an emotions folder from another character, you may as well clear it out for your own character since you'll probably be making new buttons anyway.

NOTE: The emotions folder **NEEDS** to be named **emotions**, exactly, or the char.ini will not recognize the folder and as a result not load any buttons you may have put in there.

Now, the reason I am holding off on talking about all of the above things I just introduced is the fact that in order to meaningfully work with any of these, we first need to actually make some sprites. As such, we will talk about that first.

MAKING SPRITES

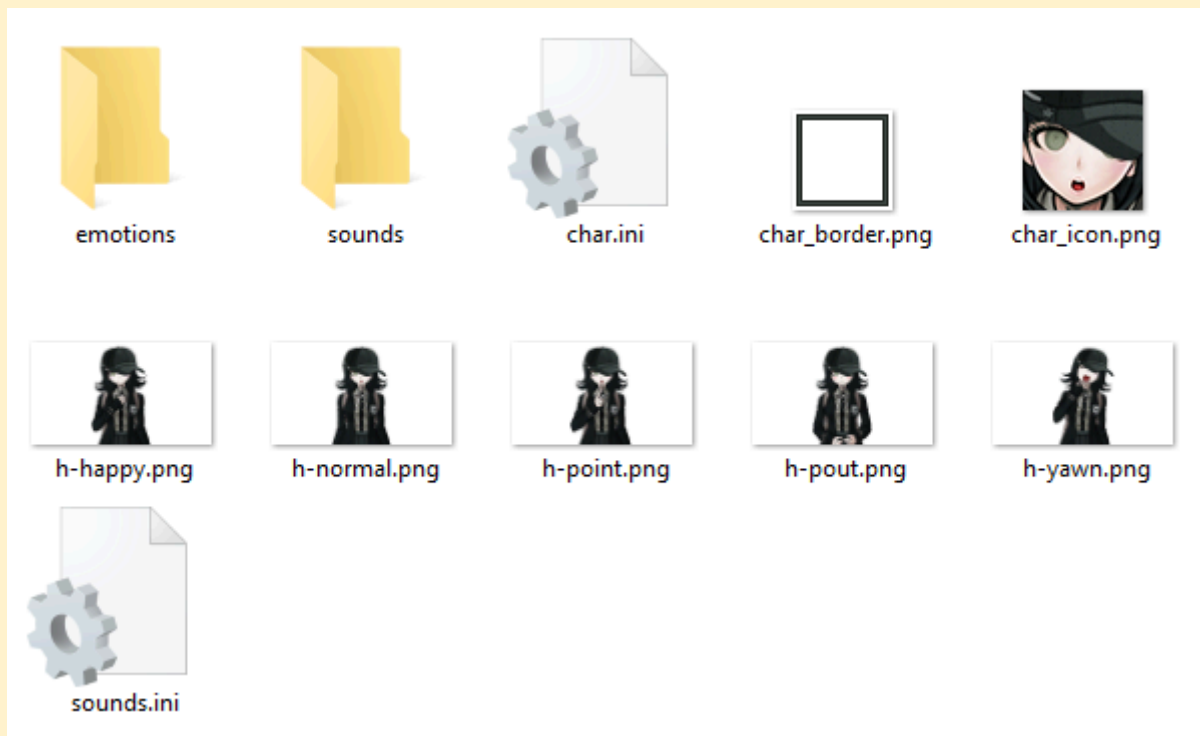
Sprites are the images and poses of your character that you will use inside the client while talking. For example, this is a sprite:



Anything you can draw in Photoshop or whatever program you use, and any image you can rip from the internet can be made into a sprite for your character as long as it is in one of the following supported file formats:

- .png
- .gif
- .webp
- .apng

Now where you put those sprites is just openly into the character folder, like so:



And just like that, your character contains sprites! It really is that easy!

NOTE: Make sure your sprites are actually decently big in pixel size so they don't appear all blurred and pixelated inside the client because the client had to upscale it. The recommended size for sprites used to be **960x544** pixels (which is why there is so much whitespace to the left and right of my character sprites to fit the client screen), but with the now available client rescaling, it is recommended you use a resolution as high as is reasonable, so feel free to make your sprites as big as **1280x720**, **1920x1080** or even bigger if you feel like it, but you're also free to stick to the old resolution. Just make sure the width to height ratio stays consistent with these kinds of resolutions (Making sprites in, say, 1080x1080 might stretch the sprites in unwanted ways).

OPTIONAL BONUS LESSON: ANIMATED CHARACTER SPRITES

Animated character sprites akin to Ace Attorney sprites aren't exactly seen very often on DRO - We actively moved away from it. However, it's still possible to bring animated sprites to Danganronpa Online using the old Attorney Online method.

Let's say you have a character like Phoenix Wright, who you want to be animated talking while your text is being written out in client, but should stop talking once the text has been written out. For this, you can have two parts of the same sprite.

Let's take Phoenix's regular sprite while not talking (left) and while talking (right):



You can name the left animated sprite **(a)normal.gif** and the right one **(b)normal.gif**. By naming them the same and putting **(a)** and **(b)** in front of their filenames, the client will recognize them as the same sprite with two states, where **(a)** is designated for **nontalking sprites** and **(b)** is designated for **talking sprites**. And that's how you set up an animated character with talking and nontalking states!

Now that we have some sprites though, we can actually move on to all the other things I said I'd talk about afterwards. Now that wasn't a very long wait, was it?

THE CHAR.INI

The char.ini can be opened by any basic code editor, like Notepad, Notepad++ or whatever code editor you prefer. Open the char.ini using the code editor of your choice now. (I am opening mine using the basic Editor that shipped with my Windows) What you will see will be something like this:

```
[Options]
showname = Chiaki Saihara|
side = hld
gender = female

[Emotions]
1 = h-normal#-#h-Normal#0#1
2 = h-yawn#-#h-Yawn#0#1
3 = h-point#-#h-Point#0#1
4 = h-pout#-#h-Pout#0#1
5 = h-happy#-#h-Happy#0#1

[SoundN]
1 = 1
2 = 1
3 = 1
4 = 1
5 = 1
```

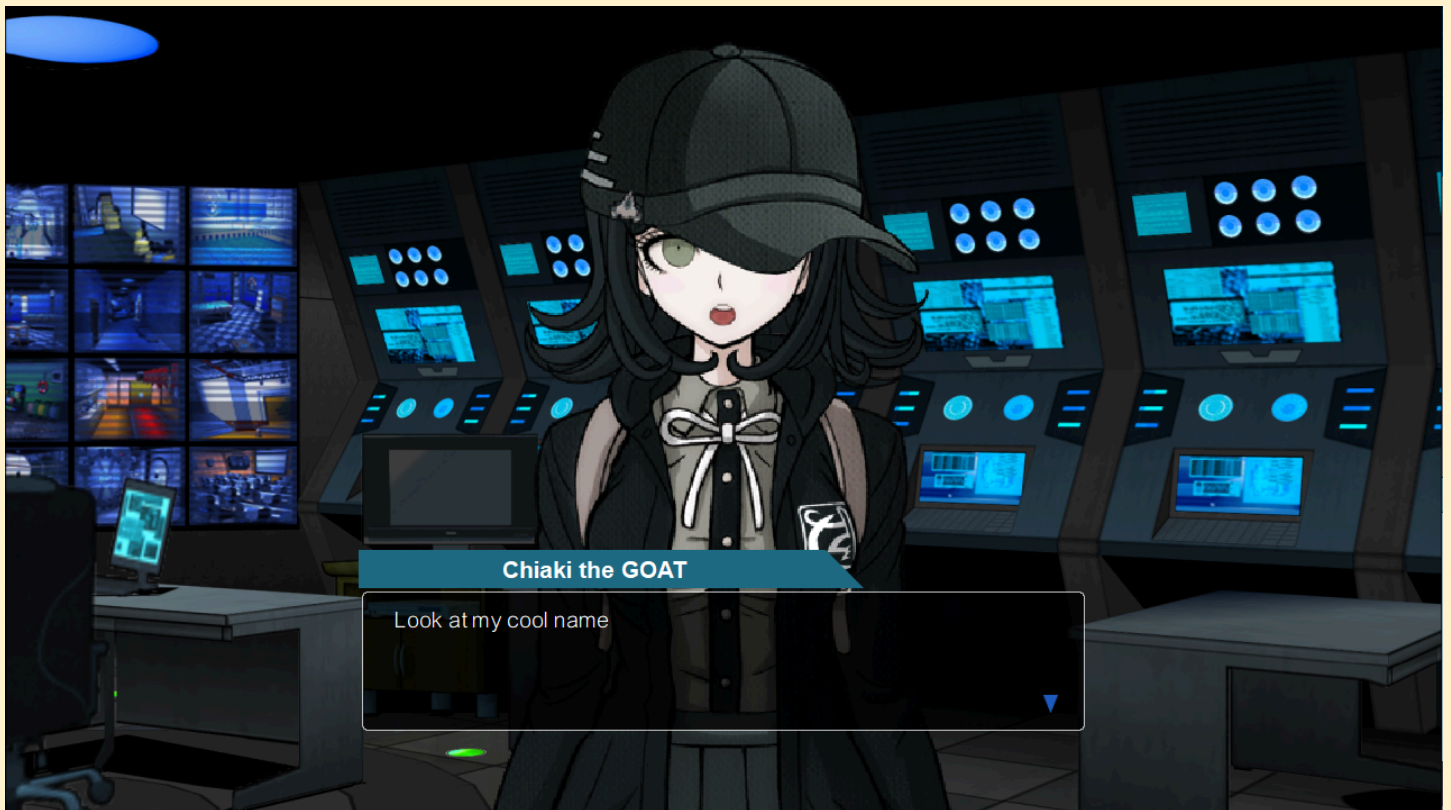
Now that's a lot of weird code at once! Let's go through it section by section, starting with the [Options] at the top.

```
[Options]  
showname = Chiaki Saihara|  
side = hld  
gender = female
```

showname: This line is optional and can be removed if not needed. This is where you can write the name that you want to show up in the client's namebox, so if you want the character folder of Makoto_HD to show up being named "Makoto Naegi", you can do that here. Something like this:

```
[Options]  
showname = Chiaki the GOAT|
```

I have changed the showname in Chiaki Saihara Tester's char.ini to "Chiaki the GOAT" for demonstration purposes. Now see what happens when I use the character in client:



Chiaki Saihara Tester now shows up as "Chiaki the GOAT" right there in the client because that's what's been declared in the showname statement.

If there is no showname statement within the char.ini, the character folder name will be used, like this:



Understood? Great. Let's get to the next line.

side: This determines where your character will “stand” in any area. If an area has several different backgrounds, this basically determines which background you will stand in front of. There are 6 different states you can choose from here (still named from Attorney Online):

wit: will make the character stand at the witness position by default.

def: will make the character stand at the defense position by default.

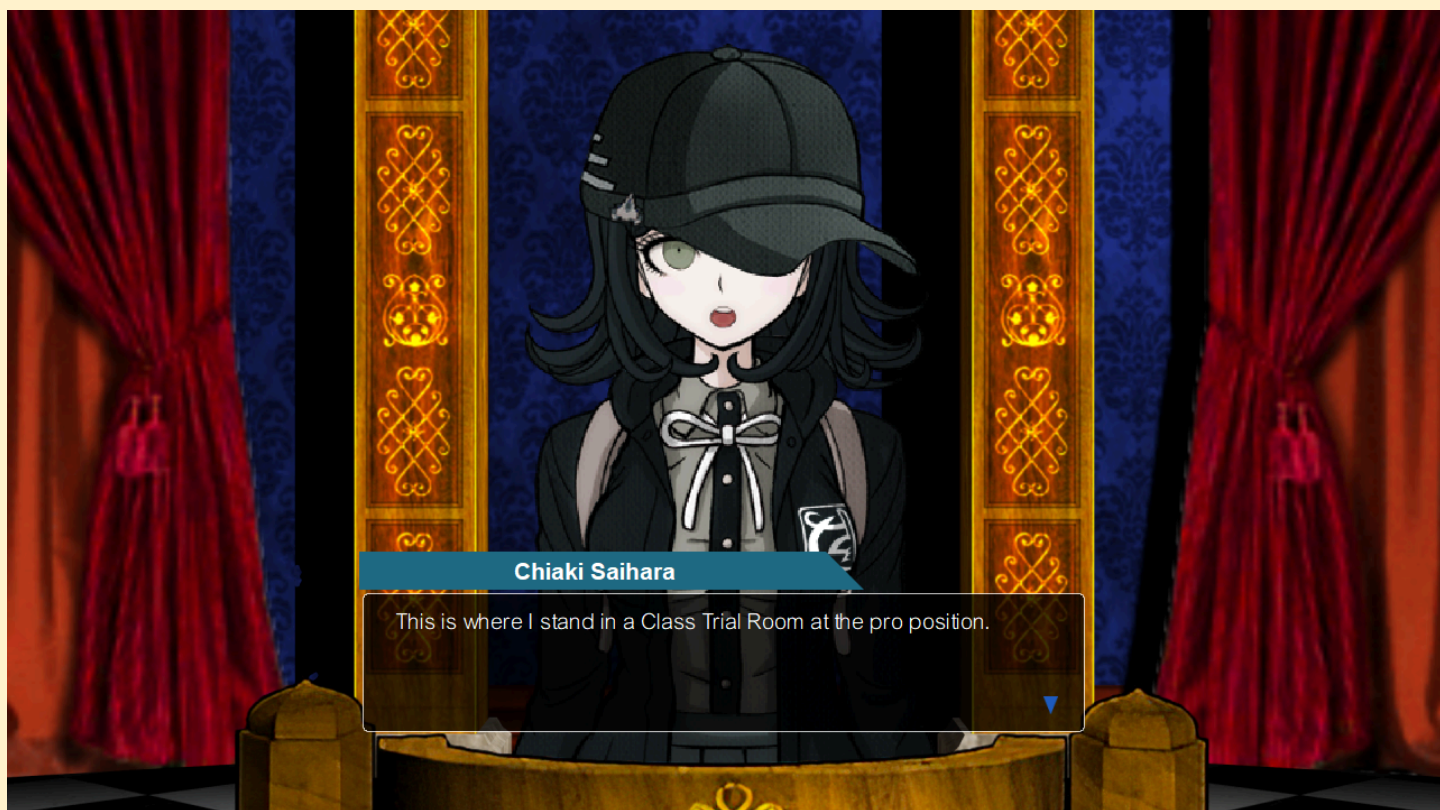
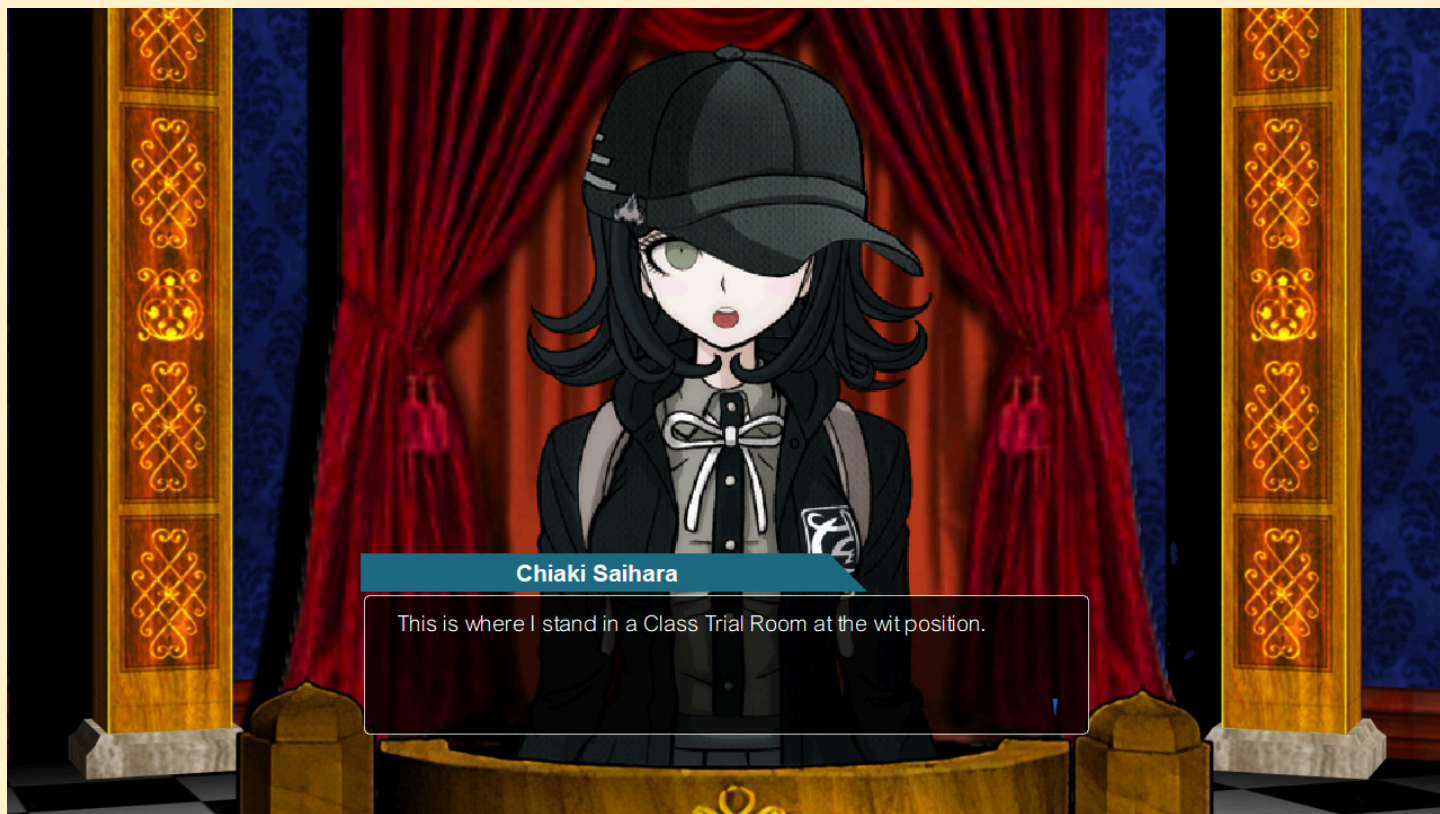
hld: will make the character stand at the defense helper position by default.

pro: will make the character stand at the prosecution position by default.

hlp: will make the character stand at the prosecution helper position by default.

jud: will make the character stand at the judge position by default. This is reserved for judge characters and characters that are often in leading positions, like the Monobears.

Let me demonstrate these positions to make it a bit clearer what editing this will do in client:

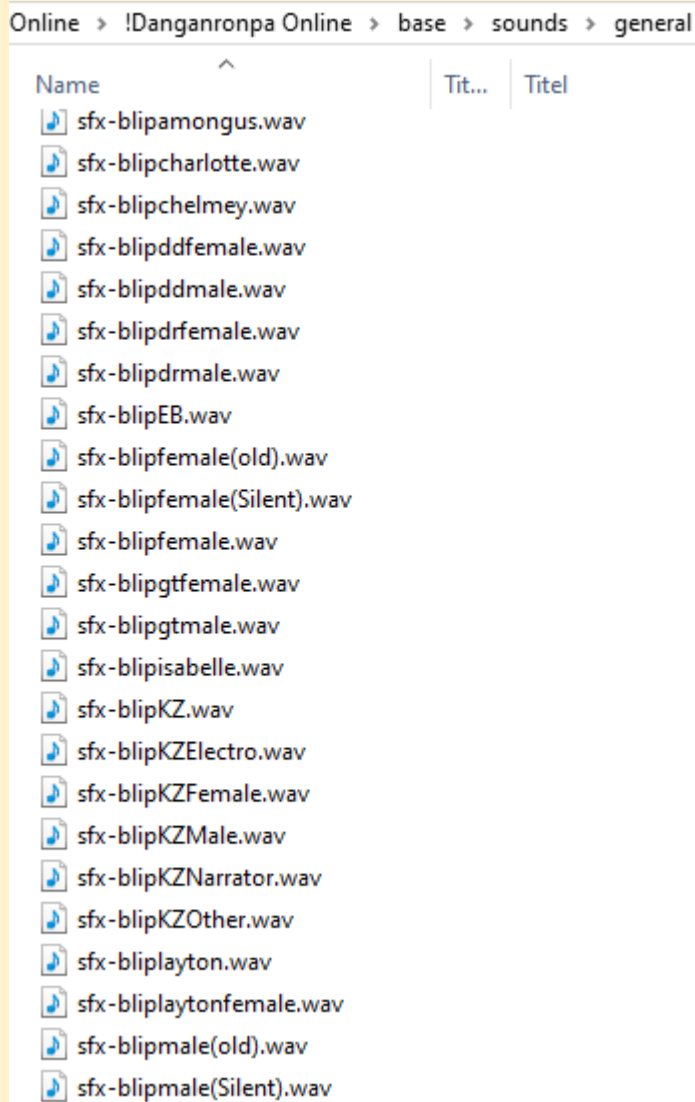




Chiaki Saihara

This is where I stand in a Class Trial Room at the jud position.
I now hand out punishments lmao.

gender: Determines the kinds of blips that will be used while you speak. You can find a few examples of blips in base/sounds/general - they will have a naming convention of “sfx_blip[name].wav”, where [name] is replaced with terms like “male”, “female” or “layton”, among others. Just look at all the ones I have collected here (file path at the top for reference):



You can make your own custom blips here using the same naming convention as well. To use a certain blip sound, you only copy the [name] part over into the char.ini's **gender =** section, meaning that if you want to use "sfx_blipmale.wav" for your character's blips, you only write **male**, if you want to use "sfx_blipsozettaslow.wav", you only write **sozettaslow**.

Now let's take a look at those [Emotions].

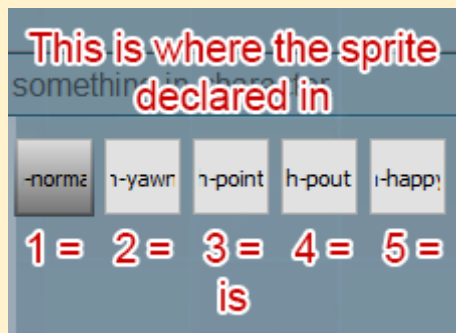
```
[Emotions]
1 = h-normal#-#h-Normal#0#1
2 = h-yawn#-#h-Yawn#0#1
3 = h-point#-#h-Point#0#1
4 = h-pout#-#h-Pout#0#1
5 = h-happy#-#h-Happy#0#1
```

Let's go through this step by step, because that's a lot of declarations per line.

1 = h-normal#-#h-Normal#0#1

This first part I have highlighted in red here is the number of the sprite. For example, this is the first sprite that will show up in the list of sprites that you will see for your character in

the client. As such, it follows that the sprite declared at **2 =** will show up in the second position, the one declared at **3 =** at the third position and so on. Make sure you don't have any of these numbers twice here, as when you accidentally declare **4 =** twice, for example, only the latest declaration of it will be used and any previous declarations of it ignored. Here is a visualization of what the numbering at the start will accomplish in client:



Now what comes after the = is the actual declaration of your sprite for that space, and there's a lot here. In fact, there are 5 more statements in total here, all separated by hashtags (#). Let's start with that first one.

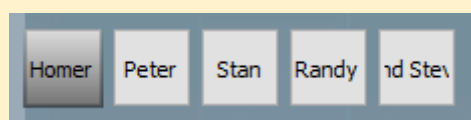
1 = **h-normal**#-#h-Normal#0#1

This part highlighted in red is where you can give your sprite any name you want. This name will mainly show up within buttons if you have no actual buttons for your character, so if you have actual buttons for your character, you will probably not see this name at all and not care what is written here. However, if you're not an artist and/or don't know how to make good buttons for your character, you may end up relying on the base buttons, which would look like in the image above. In those cases, it may make sense to take more care to put proper names here so you can grasp what button is for what sprite.

You can put pretty much anything you want here. As a demonstration, I have gone wild with my button names:

```
[Emotions]
1 = Homer#-#h-Normal#0#1
2 = Peter#-#h-Yawn#0#1
3 = Stan#-#h-Point#0#1
4 = Randy#-#h-Pout#0#1
5 = ...and Steven!#-#h-Happy#0#1
```

Now look at the buttons in the client:



So this is how you control what will be written inside the image-less buttons, though note that you would need to keep these short if you actually rely on this, as you can see with the last button there that long names can quickly trail out of bounds.

With that though, let's look at the next statement:

```
1 = h-normal#-#h-Normal#0#1
```

I have highlighted the dash after the first # here. This is the place where you would put preanimations, also known as preanims. Most of the time your character won't have preanimations though, so it's common that there's either a dash or nothing at all here.

PRO TIP AND RECOMMENDATION:

While putting a dash in this space is perfectly valid, like so...

```
1 = h-normal#-#h-Normal#0#1
```

...I actually personally recommend just putting nothing at all between those first two #'s, like this:

```
1 = h-normal##h-Normal#0#1
```

The difference here is that when a dash is put in that space, the char.ini will still look for that nonexistent "-" file.

If you put nothing in between those two #'s, therefore leaving it truly blank, the char.ini will skip that process entirely, making your char.ini more optimized as a result.

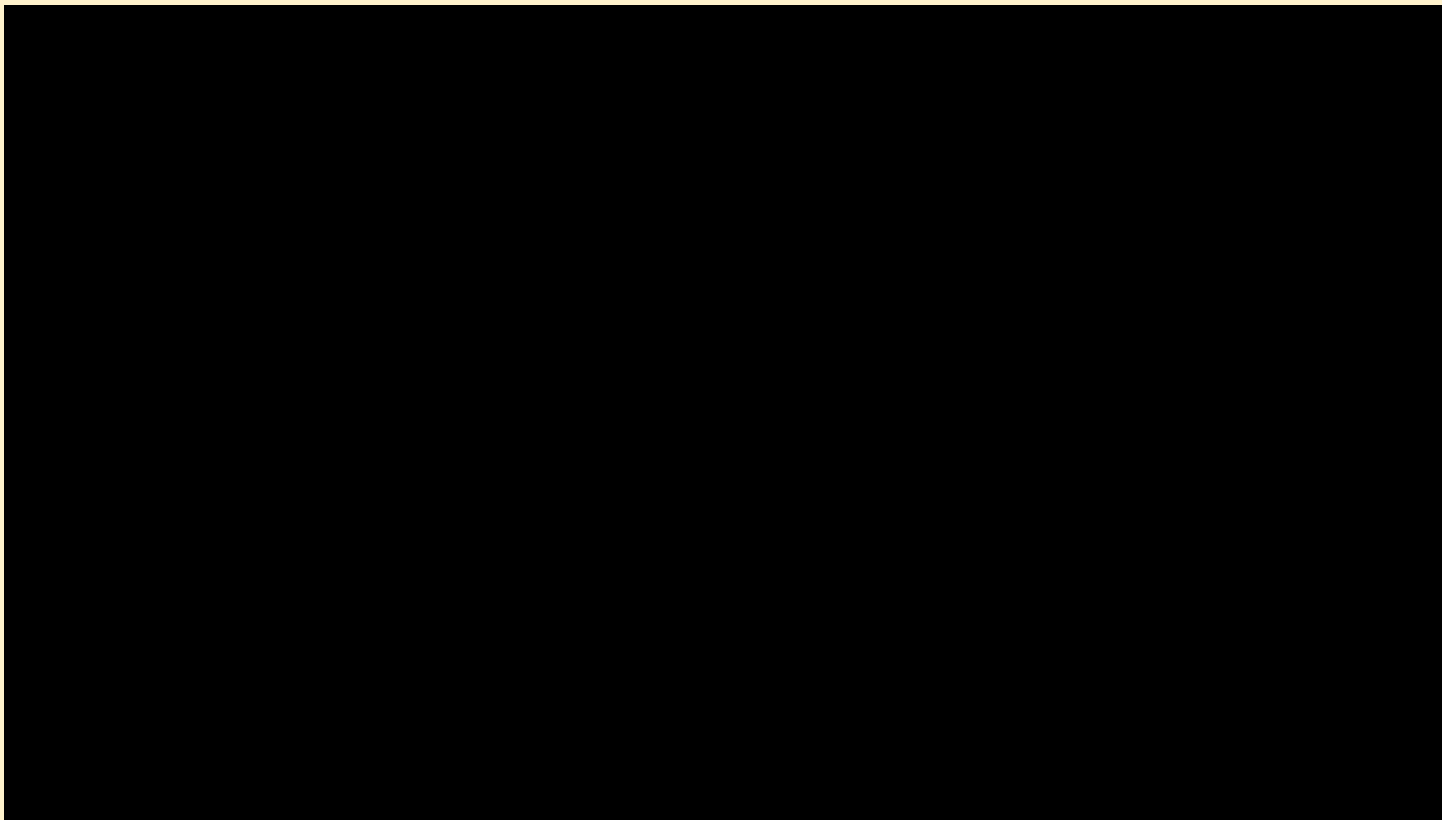
There's really no downside!

Now you may be asking: "But what if I actually do want to have a preanim on one of my sprites?" Don't worry, I will still explain this for those cases where a preanim is actually desired.

First off, if you don't know what a preanim does *exactly*, It's basically an animation that plays once before your actual sprite. A great example to use here is Monokuma's execution animation:



Now this file is named “PunishmentV3.gif” for context that will be needed in a bit. Also note that this is not the actual sprite, but the preanim before the sprite. The actual sprite is this:



This is “GameOver.png”, the actual sprite that remains after the preanim. Now in MonokumaV3_HD’s char.ini, the related line looks like this:

```
50 = PUNISHMENT#PunishmentV3#GameOver#1#0
```

Note that the preanim “PunishmentV3.gif” is noted down after the first #, in the space where preanim s go when needed. You will have to note down your own preanim in the same way, using its filename, but without the file extension. What follows after is the actual sprite, “GameOver.png”, written here as “GameOver”, meaning the same naming convention used for preanim s also applies here.

With this setup, the PunishmentV3.gif will play first when you select this specific sprite to talk, and only afterwards will you start talking with the GameOver.png sprite.

To make it more understandable in a different way via Ace Attorney, once again using Phoenix for this, this is what would happen, in order:



Phoenix’s shocked preanimation plays.
(in the previous example this was PunishmentV3.gif)



You start talking as Phoenix here.
(Previous example: GameOver.png)



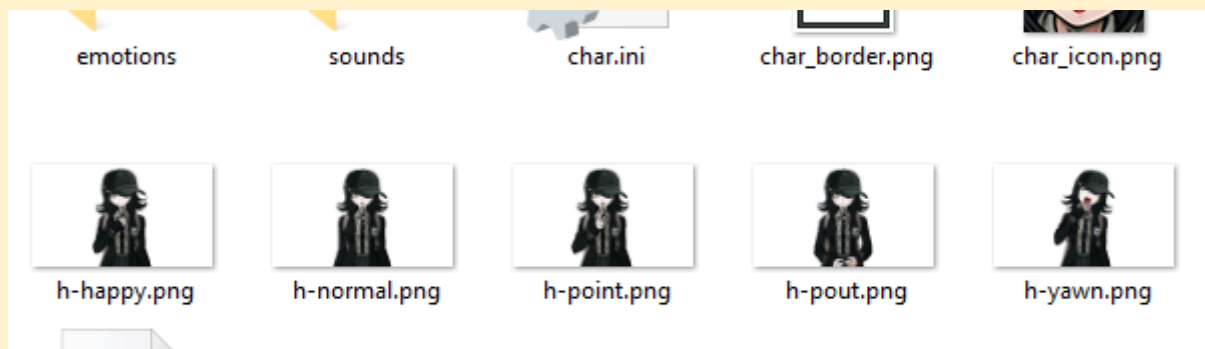
Once your text stops, this sprite plays.
(Once again GameOver.png since the previous example had no talking/nontalking variants)

Since I already started touching on the next section on this line, let’s go over that one again:

1 = h-normal#-#**h-Normal**#0#1

Like I previously mentioned, this is where you declare your actual regular sprite.

Chiaki Saihara’s first sprite is named **h-normal.png** in my folder, as you can reference here:



And since I wanted it to be my first sprite, I declared it in the section I have highlighted in red, as **h-Normal**, without the file extension, same as you would do with preanims.

```
[Emotions]
1 = Homer#-#h-Normal#0#1
2 = Peter#-#h-Yawn#0#1
3 = Stan#-#h-Point#0#1
4 = Randy#-#h-Pout#0#1
5 = ...and Steven!#-#h-Happy#0#1
```

When using animated sprites with (a) and (b) parts, don't write the (a) or (b) of the filename here. For example, if you have (a)sweat.gif and (b)sweat.gif, you simply write **sweat** here, without the (a), (b) or file extension.

This is not case-sensitive, meaning you can make capitalization mistakes here and your sprite will still work, as demonstrated by some version of past me having declared the sprite as **h-Normal** in the char.ini despite the file itself being named **h_normal**, and the sprite still works fine in the client.

Now for the last two statements, which are numbers. Let's tackle that first number first.

1 = h-normal#-#h-Normal#0#1

This number determines whether any preanim you have declared for this sprite will play before the sprite by default.

0: The preanim is off by default and you'll have to check the preanim on manually in the client every time you want to use the preanim on this sprite.

1: The preanim is on by default and will therefore play in the client by default every time you use this sprite to talk. You'll have to uncheck the preanim every time you don't want the preanim to be played.

5: This one is a leftover from Attorney Online. It shows your sprite in front of a zoom background (if available inside the current background), and removes the desk if there would usually be one, making this one great for fullscreen animations like the previous execution animation, though it's not really being used anymore due to this next one...

1 = h-normal#-#h-Normal#0#1

This last number determines whether your character will have a foreground in front of them in certain area backgrounds that come with foregrounds. These are mostly Class Trial Rooms with desks as foregrounds, so it's usually desired to turn this on for immersion's sake.

If you declare **1** here, foregrounds like desks will show in front of this sprite.

If you declare **0** here, foregrounds like desks will not be shown in front of this sprite, ever.

If you think that turning this on with 1 will show a desk in front of your character sprite in EVERY background, don't worry, that is not true at all. If turned on with 1, desks will only show in front of your character sprite if you're currently inside a background that comes with desk foregrounds. That means your character will still be nice, free and deskless in regular RP areas (unless any of those ship with special foregrounds, which usually is not the case).

There are, however, reasonable reasons to turn desks/foregrounds off for certain sprites. For example:

- It's a fullscreen sprite, like the execution animation and sprite shown earlier, and a desk would be in the way of it.

- You have a very special character, like a big-ass Exisal, that would roam freely inside a Class Trial Room and a desk would be TOO PUNY AND TINY for it!

Now that you know what each section of one of these lines in [Emotions] does, we can look at that last section, the [SoundN].

```
[SoundN]
1 = 1
2 = 1
3 = 1
4 = 1
5 = 1
```

SoundN stands for "Sound Name". This section is also optional, so if you don't need it, you can freely remove it from your char.ini entirely. Here you can declare a sound to be played every time you talk with a certain sprite. Every. Time.

For example, let's say I want to play "consume.wav" every time I talk with my first sprite, which would be h-normal. I could then write either of these here:

<pre>[SoundN] 1 = consume</pre>	<pre>[SoundN] 1 = consume.wav</pre>
---------------------------------	-------------------------------------

The sound effect will work with and without file extension, so no matter which of these you choose, you will then hear "I WILL CONSUME THE LIVING AND THE DEAD" every time you talk with this first sprite. Every. Time.

This means that this can get annoying really fast, so I really only recommend using this section for certain sprites if you're really sure you need a certain sound to play EVERY time you use that sprite, because there's definitely a better way of using sound effects in client... a way that I will talk about very shortly, as we are done talking about the char.ini! However, before we move on to that, I do need to address the issue of:

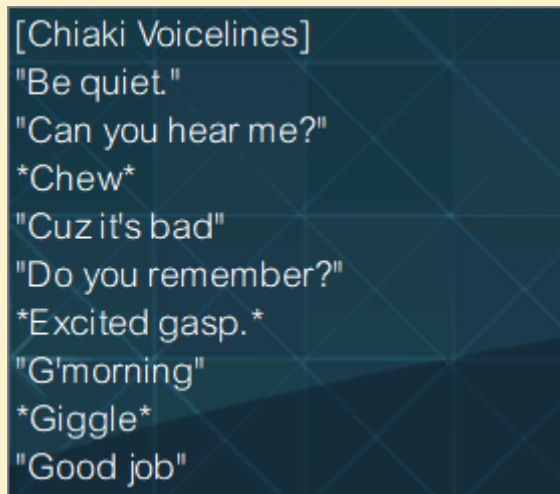
“But Chrissy, I see some other sections in my char.ini that you haven’t talked about, like name, SoundT, Time, firstmode and number!”

The reason I haven’t talked about these is because they are deprecated and no longer being used, so there is no point in educating people about sections that are no longer needed. You are free to delete any of those from your char.ini without any repercussions.

Now, to more and better sound management!

THE SOUNDS.INI

The sounds.ini is where you can declare the sounds that will show up in your in-client sfx list for your character. Your in-client sfx list may look something like this, ignoring custom theme design:



```
[Chiaki Voicelines]
"Be quiet."
"Can you hear me?"
*Chew*
"Cuz it's bad"
"Do you remember?"
*Excited gasp.*
"G'morning"
*Giggle*
"Good job"
```

You can click any of those lines before speaking, and when you then talk with one of the sounds from this sfx list selected, that sfx will play with your message. The sfx list makes playing sounds with your sprites MUCH more flexible than just using [SoundN] in your char.ini, so it's really in everyone's best interest to just set up this sfx list instead, and you do that by going into and editing the sounds.ini, which may look something like this:

```
DRgloom = Gloom
DRV3Gloom = Gloom V3
DRshock = Shock
DRshock2 = Yell
DRshock3 = Comical Shock
DRsparkle = Sparkle
sfx-deskslam = Desk Slam
sfx-explosion = Explosion
sfx-gunshot2 = Gunshot
sfx-gunshot4 = Gunshot 2
sfx-meow = Meow
sfx-stab = Damage
sfx-stab2 = Damage 2
sfx-yep = YEEEEAAAAAH

[Chiaki Voicelines]
Chiaki/Be quiet = "Be quiet."
Chiaki/Can you hear me = "Can you hear me?"
Chiaki/Chew = *Chew*
Chiaki/Cuz it's bad = "Cuz it's bad"
Chiaki/Do you remember = "Do you remember?"
Chiaki/Gasp = *Excited gasp.*
Chiaki/G'morning = "G'morning"
Chiaki/Giggle = *Giggle*
Chiaki/Good job = "Good job"
Chiaki/Good night = "Good night"
Chiaki/Hello there = "Hello there."
Chiaki/Hey = "Hey."
Chiaki/Hey hey = "Hey hey."
Chiaki/Hmph = "Hmph."
Chiaki/Hmm = "Hmm..."
Chiaki/Huh = "Huh?"
Chiaki/I'm glad = "I'm glad."
Chiaki/I think = "...I think."
Chiaki/Just kidding = "Just kidding."
Chiaki/Let's go = "Let's go."
Chiaki/Nice to meet you = "Nice to meet you."
Chiaki/No way = "No way."
Chiaki/Nope = "Nope."
Chiaki/Perhaps = "Perhaps."
```

Let's break this down. Any one line basically works like this:

File Path = Sound Name

The **file path** here may start in base/sounds/general or alternatively in your own character's sounds folder, the sounds.ini will look in both.

That means that if you have a sound in base/sounds/general called DRshock.wav, you don't have to write out the whole file path to get there, you can just write DRshock (without file extension), as the sounds.ini will pick it directly from there.

It also means that if you put custom sfx into your character's sounds folder, you can reference those the same way, as the sounds.ini will pick those out directly from there as well.

However, there is a possibility that in older versions of other characters, you may see subfolders from base/sounds/general referenced, similar to how you can see in the screenshot above. This is a good example of referencing subfolders, should you ever need it (Though adding sounds folders to your characters has made this basically obsolete, but hey, you may need to split up voice lines into separate folders or something, there could still be uses for it, so I'll still explain this anyway).

Should you have a separate subfolder or a few more for your character in base/sounds/general or your own character's sounds folder, you can reference those subfolders by adding the name of that folder in front of the sfx name you want to reference from that folder, then adding a slash and then writing the sfx name from that folder, like you can see with all of the voice lines at the bottom of the above screenshot, which are written like, for example, **Chiaki/No way**.

The **sound name** is the text that will appear for this specific sfx in the sfx list. For example, at Chiaki/No way, after the = it is labeled as **"No way."**, and that is the text that will appear in the sfx list for you to select that specific sfx.

And so we have one line like:

Chiaki/No way = "No way."

...or alternatively a line like:

DRshock = Shock

...which you can now select in the sfx list to use with any sprite you want. Just follow this same convention for every sfx you add to your sounds.ini.

Now you may have noticed that one line in brackets called [Chiaki Voicelines]. This one is of course not an sfx, but as it will show up in the sfx list like any other sfx, it makes for a great differentiator. You can use lines like these to separate general sfx from character voice lines, for example, which can be helpful if you have a lot of sfx on your character. That said, this is entirely optional and you can leave it out if you don't need it.

And that's all for the sounds.ini! It's actually pretty simple like that!

MAKING BUTTONS

Now there's one last open issue to tackle... Remember that emotions folder inside your character folder? I didn't forget about that. That one is where you will put your custom buttons for your character, and I will tell you how to properly set it up.

THE RECOMMENDED METHOD

First, you will want to actually make your buttons. You can do that in basically any image editing program. Button size starts at 40x40 pixels, but due to the client rescaling feature, it

is now recommended to draw buttons at higher resolutions as well, similar to the sprites. Just make sure that your buttons remain square. For each of your sprites, you will want to make a button that showcases what the sprite activated with that button is. As an example, let's take Chiaki Saihara's first sprite in the char.ini, h-normal, which is this sprite:



For this, I would want to make a button that shows me in a good way what kind of sprite this is, so what I did is choose a section of this sprite that made that clear for this button, and I ended up making this button:



You may also notice that I added a small black border to this button. This is just a style choice and not required for you to do as well, though many other characters do follow this same kind of style choice. You could also go wild with what you want your buttons to look like, I'm not your mom. In fact, let me go wild with the button for the second sprite in my char.ini, which is h-yawn, this one:



Look what monstrosity I have created for this sprite's button just because I could:

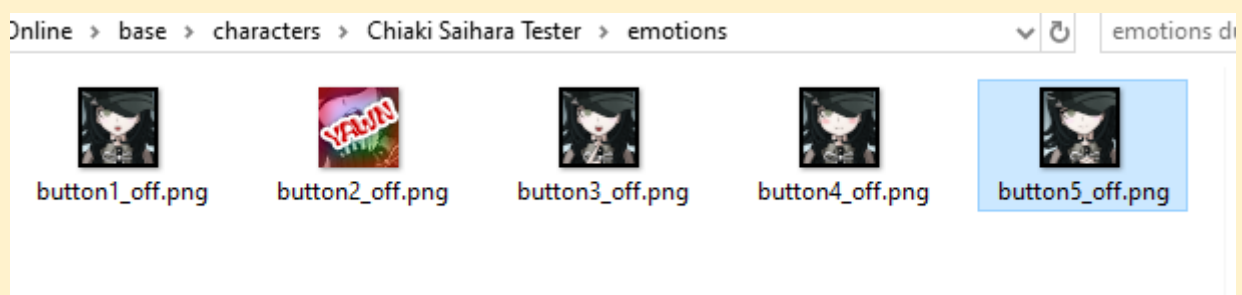


Just make sure it is recognizable from a button which sprite will be activated with it, or people may end up being confused and looking for sprites all the time because they don't know where the fuck it is.

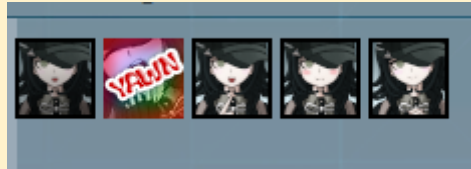
Now another thing to note is how to name your buttons. You can't really name your buttons just anything, the char.ini will only recognize certain names in order to properly order these buttons.

In this recommended method, your buttons should be named **button1_off.png** for the first sprite in your char.ini, **button2_off.png** for the second sprite in your char.ini and so on. This way, the char.ini can assign each of your buttons to the appropriate sprite.

In the end, your emotions folder so far should look something like this:



And inside the client, your button list may look something like this:

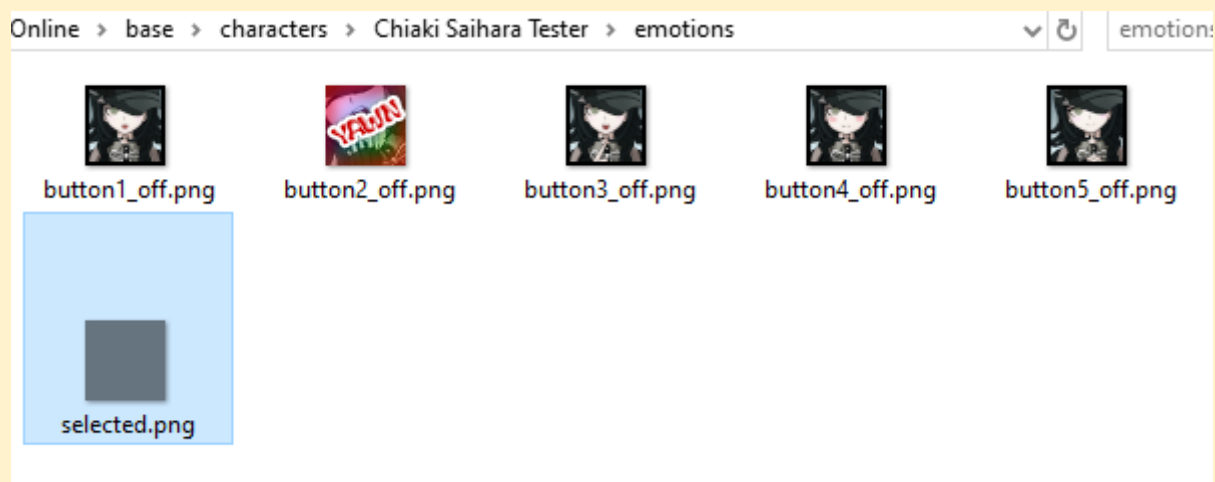


Now, you can leave it at that here and be done and happy with your character, because the client already comes with a built-in overlay for whatever button you have currently selected, making that button a bit darker for you to see which button you have currently selected. However, if you want to make your own overlay specifically for your character's buttons, you can absolutely go ahead and do that, and it's very easy to do too! Basically what you do is make another button the same size as your other buttons, and just put your overlay onto that.

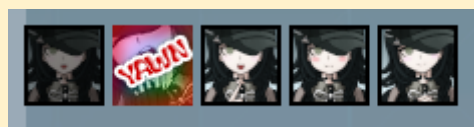


Just a very simple black overlay with 50% transparency so the sprite's button is still visible behind it.

If you decide to make your own overlay like this yourself, you need to name it **selected.png** and put it in the emotions folder with all your other buttons, as demonstrated here:



Now, when I have a button selected for this character inside the client, it will take my overlay and look like this:



My own overlay makes the difference in brightness a bit clearer, but I understand if it may be a bit difficult to tell, so let me go a bit wild with my overlay like I did with my second button, because you can do that here too, I'm not your dad:



Now this pink monstrosity will be your button overlay for whatever button you currently have selected. Look at this:

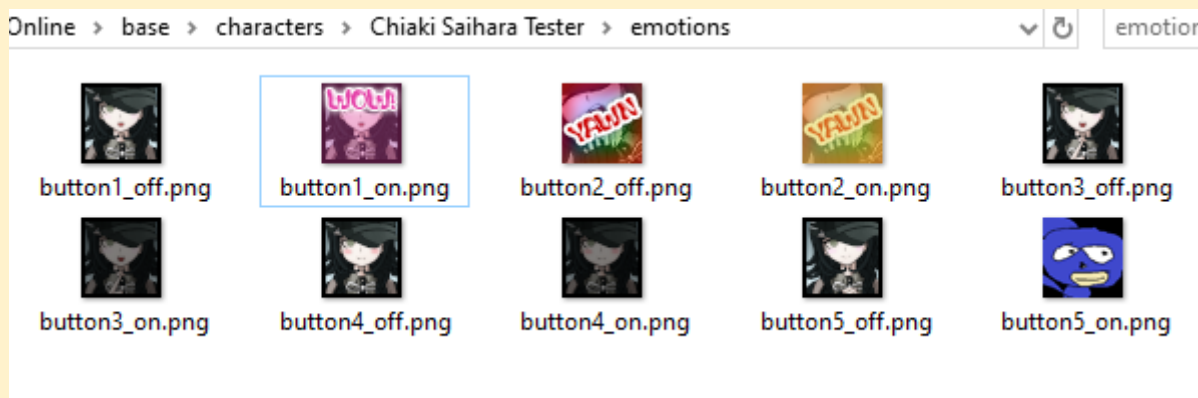


And that's how you set up your buttons using the recommended method. There is however another method you can use if you really want absolute finetuned control over every button.

THE OLD METHOD

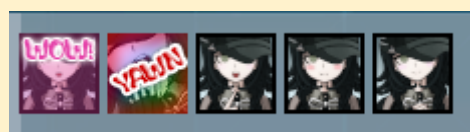
This one is the same as the above method in terms of making the `button[number]_off.png` buttons - the difference is in making the `selected` buttons.

In this method, you will not be making a singular `selected.png` for every button - Instead, you can make all of the `button[number]_off.png` buttons again, but call these `button[number]_on.png`. You would be doing something like this then:

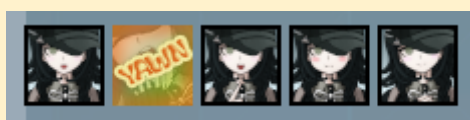


I have once again gone a bit wild with these for demonstration purposes. With this method, you can create a specific “selected” image for each button if you need to, and the `char.ini` will take these `button[number]_on` images.

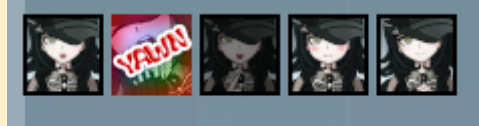
Now you would have a button list for your character inside the client that would work like this:



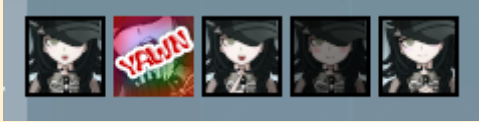
Button 1 selected



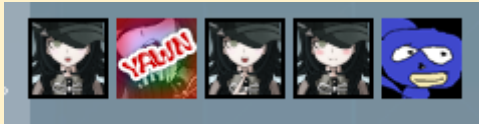
Button 2 selected



Button 3 selected



Button 4 selected



Button 5 selected

Congratulations, we have created a cursed button list!

That's all you need to know about making a character though! You have hereby graduated Character Making Academy! Here is your approval stamp:

✓ You now know how to make a character!

-Chrissy's Character Making Academy

Now go out there and create!