Intro and Prep

This guide is best viewed on PC. Some parts may be difficult to read on mobile
Discord links may require you to be part of the Hollow Knight Modding Discord

Knowing how to create mods isn't required to make your very own custom Hollow Knight skin. The mod already exists, all you need to do is create the art. Making your own skin for the knight is a big task, with over 500 sprites on just the knight sprite sheet alone, but there are tools to make it easier. This document will cover the basics of skin making, starting with the prep work.

While using the default skin is great as a base, do not use other artists' skins without their permission! You can use other skins as references, but never draw on top of or edit another skin without asking its creator first.

First make sure you have 'Custom Knight' and its dependency 'Satchel' installed. You can find these mods on the Lumafly Mod Installer. The first time you boot up the game and enter a save file with CK installed, the game will lag while it generates the default skin in the mod folder. Once it's finished and the lag stops, you can exit the game. The default skin will be your file naming guide for your custom skin. If you ever accidentally delete part of the default skin, or save over the original files, you can force it to generate again by going to the advanced settings under the custom knight mod menu and clicking 'make default'. It will generate the default skin the next time you boot up the game.

Next you will need to get the sprites. The sprite sheets (also called the atlases) are difficult to work with directly, so instead download the sprites folder found here: Sprites
Do not rename any of the files or folders inside the sprites folder. It's very important that the names of the folders stay as they are or the packing programs will not be able to pack them into sprite sheets.

To edit the sprites, any art program that can save with transparent backgrounds will do. If you do not have an art program that can handle transparency, some popular free options include: Krita, FireAlpaca, Gimp, Paint.net
I personally use Krita, so any images of an art program in this guide will be of Krita.

Note: Clip Studio Paint has been known to cause white outlines to appear around the knight in game due to how it exports. Open your edited png file in a different program, such as Paint 3D, and re-save it to remove the white fuzz. If that doesn't work, try exporting it as a psd, opening it in a different art program, and re-exporting it as a png. But the best way to avoid this problem is to just not use CSP.

The last thing you'll want to grab before you start is a sprite packing program.

Sprite Packing Programs

The packing programs are not required to make a skin, but they make the process so much easier that I highly recommend using one.
If you choose not to use a packing program, you will need to edit the sprite sheets directly.
There are 3 options to choose from: Sprite Packer (**SP**), New Sprite Packer (**NSP**), and Custom Knight Creator (**CKC**).

All packers do the following:
+Allow you to work on sprites individually and in animation order
+Check for sprites that are used in multiple animations (dupes) and replace them so they match
+Preview animations

Sprite Packer (recommended)

Download: https://github.com/magegihk/HollowKnight.SpritePacker

+Will make sure you have dealt with all dupes before it lets you pack
+Very stable and easiest to get going again after a crash
+Can pack anything that can be dumped, as long as there is a compatible spriteinfo file in the atlas folder
- Animation playback is always 12fps even if the animation plays faster in game
- Windows only. Not compatible with Mac or Linux
- Sprites must be saved in a very specific file location
- preview window does not adjust to large sprites
- default language is not english, so you need to switch it over every time you open the program (not a big deal, but slightly annoying)

If you are using **SP**, you must put your sprites folder in the following location:
C:\Users\<your username>\AppData\LocalLow\Team Cherry\Hollow Knight

So the file path for the first sprite of the knight's idle would look like this:
C:\Users\<your username>\AppData\LocalLow\Team Cherry\Hollow Knight\sprites\Knight\001.Idle\001-00-007.png

Note:Spritepacker will not launch if there is an empty folder in the sprites folder.

New Sprite Packer

Download: https://github.com/jngo102/sprite-packer (Do not download v0.0.1. Download v0.2.0)

+Dark Mode toggle, so you can check how your sprites look on a light or dark background
+Compatible with Windows, Mac, and Linux
+Preview Window adjusts to large sprites
+Can pack anything that can be dumped, as long as there is a compatible spriteinfo file in the atlas folder
+Animation playback is accurate to in game fps
- More taxing on your computer than other Packers
- Must be installed
- Animation playback can lag, especially when working on the Knight
- Crashes if folders are named wrong or lack a spriteinfo file and might not open again without a reset
- Does not always properly identify dupes if the individual images share a file name with an image in a different sprite folder.

If you are using **NSP**, you must direct it to your sprites folder so it can find your sprites. Click 'Options' -> 'Set Sprite Path' and select your sprites folder. This folder can be named anything you want, but remember you cannot change the names of the folders inside (ex: Knight, Quirrel, Hollow Shade). If the sprites do not show up after you have selected your path, exit the program and open it again and they should appear.

NSP will crash if the selected sprites folder contains a folder that does not contain a compatible spriteinfo.json file and sometimes will no longer open even if the problem folder is removed. If this happens, remove the problem folders and reset its memory by doing the following:

- Press the WindowsKey + R at the same time to bring up the "Run" window, then enter %APPDATA%/sprite-packer/config and press Enter. (For Macs, you will need to go to ~/Library/Application Support/rs.sprite-packer/)
- Delete the sprite-packer.toml file.
- Run New Sprite Packer again

Custom Knight Creator

Original Download:https://github.com/cmot17/CustomKnight-Creator
Updated Fork: https://github.com/TriMay/CustomKnight-Creator-Fixed (less crashing)

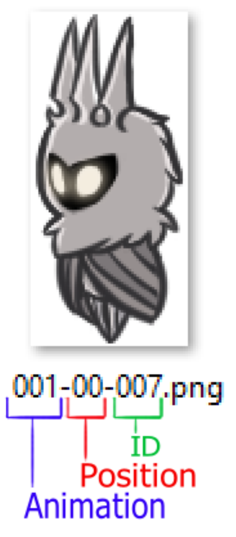
+Compatible with Windows, Mac, and Linux
+Animation playback is accurate to in game fps
+Search box allows you to filter animations by name
+Identifies unmarked dupes
- Crashes if folders are named wrong or incompatible folders are selected, and will not open again without a reset
- Can only pack certain sheets (specifically the ones related to the knight. Notably cannot pack the shade nor enemies)
- Generated sprite sheets will not contain any sprites that are not present in the program during packing. All sprites must be present, even those purposely left vanilla.

If you are using **CKC**, you must direct it to each of your sprite folders. Under Root Folders, click 'Add' and find your sprites folder. Add all of the folders you plan to work on individually (ex: Knight, Quirrel, Spider Cutscene Anim). Click on the root folder of the sprites you want to actively work on and click 'load categories'. Click on the Atlas you are currently working on and click 'Load Animations'.

CKC will crash if you don't direct it to the correct folder, or if the folders contain any extra files, and will no longer open. If it does, reset its memory by doing the following:

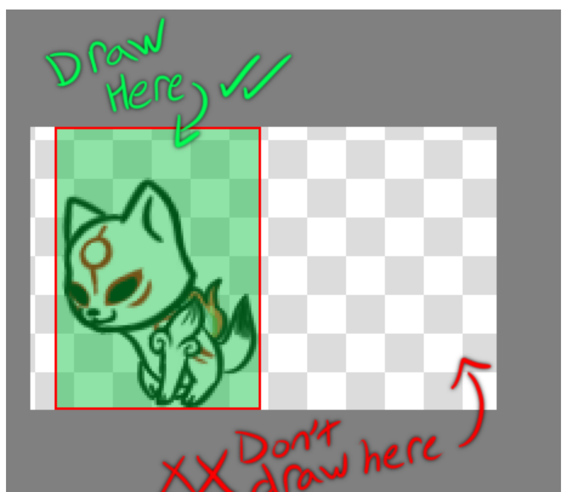
- Press the WindowsKey + R at the same time to bring up the "Run" window, then enter %appdata%\..\..\CustomKnight Creator and press Enter.
- Delete the savestate.json file.
- Run Custom Knight Creator again

Creating a Knight Skin



Before you start drawing your skin, take a look at a sprite's file name. Every sprite should have a name that looks like xxx-yy-zzz.png, with X being what animation it belongs to, Y being its position in that animation, and Z being its sprite ID. The sprite ID is important, as any sprite that has the same ID, regardless of what animation it belongs to, is a duplicate and uses the same sprite on the sprite sheet. That means they must match each other. Make sure you don't waste time and effort drawing a full set of frames, only to realize afterwards that a bunch of them are duplicates that are going to overwrite each other.

If two sprites share an ID, you only need to edit 1 of them. The packer will automatically replace the unchanged duplicates later.
NOTE: There are some dupes that do not share an ID despite being duplicates. They will be listed below in the 'unmarked dupes' section.

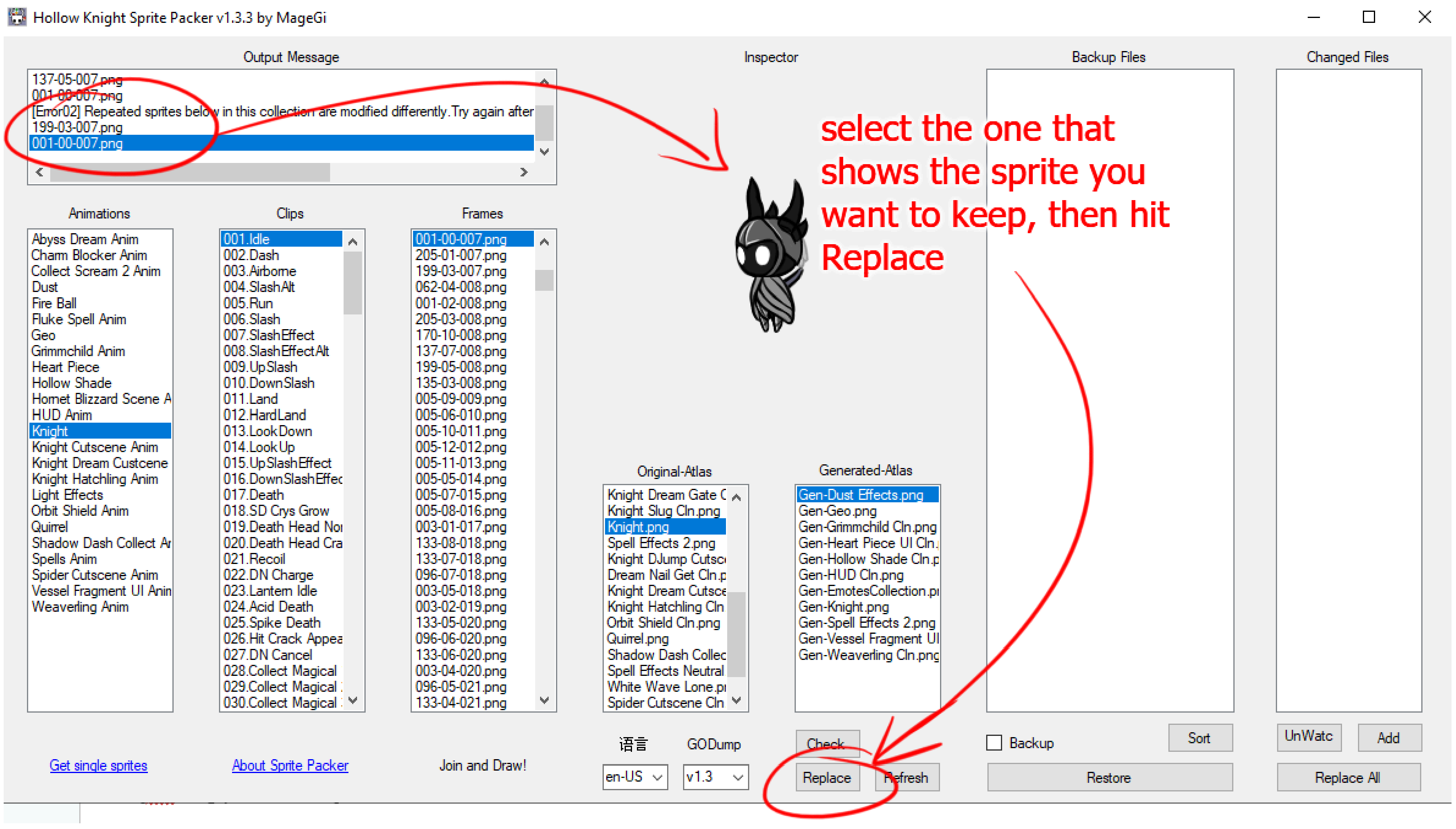


Once you open up a sprite in your art program of choice, it's important that you **do not alter the size of the canvas and do not draw on nor outside of the red borders**. Changing the canvas size will make your sprite not pack correctly and if a sprite has red borders anywhere in the canvas, anything drawn outside of those borders will get cut off. Part of the challenge of making a skin is working in a limited space. Keep that in mind when designing your skin. If you want to break these size limits, you will need to use an add-on (see Add-ons section).

NOTE: If you are editing sprites while **SP** is open, make sure you do not have the sprite you are actively working on selected in SP, or you will not be able to save it as it will be ‘open in another program’. If you get this message while trying to save a sprite, just select a different sprite and save again. If you recently packed an atlas, you may need to hit ‘refresh’ in SP and then select a sprite you’re not working on at the moment.

After you’ve edited all the non duplicate sprites of an animation, you can check to see how it looks using your preferred packing program. In your packer, click on the animation clip you want to preview. **SP** and **NSP** will automatically play the animation on loop. To pause the autoplay, click on one of the frames. To resume the autoplay, select the animation clip again. **CKC** needs to be manually told to play the animation, and will not loop unless the autoplay checkbox is checked. To stop playback, the checkbox must be unchecked.

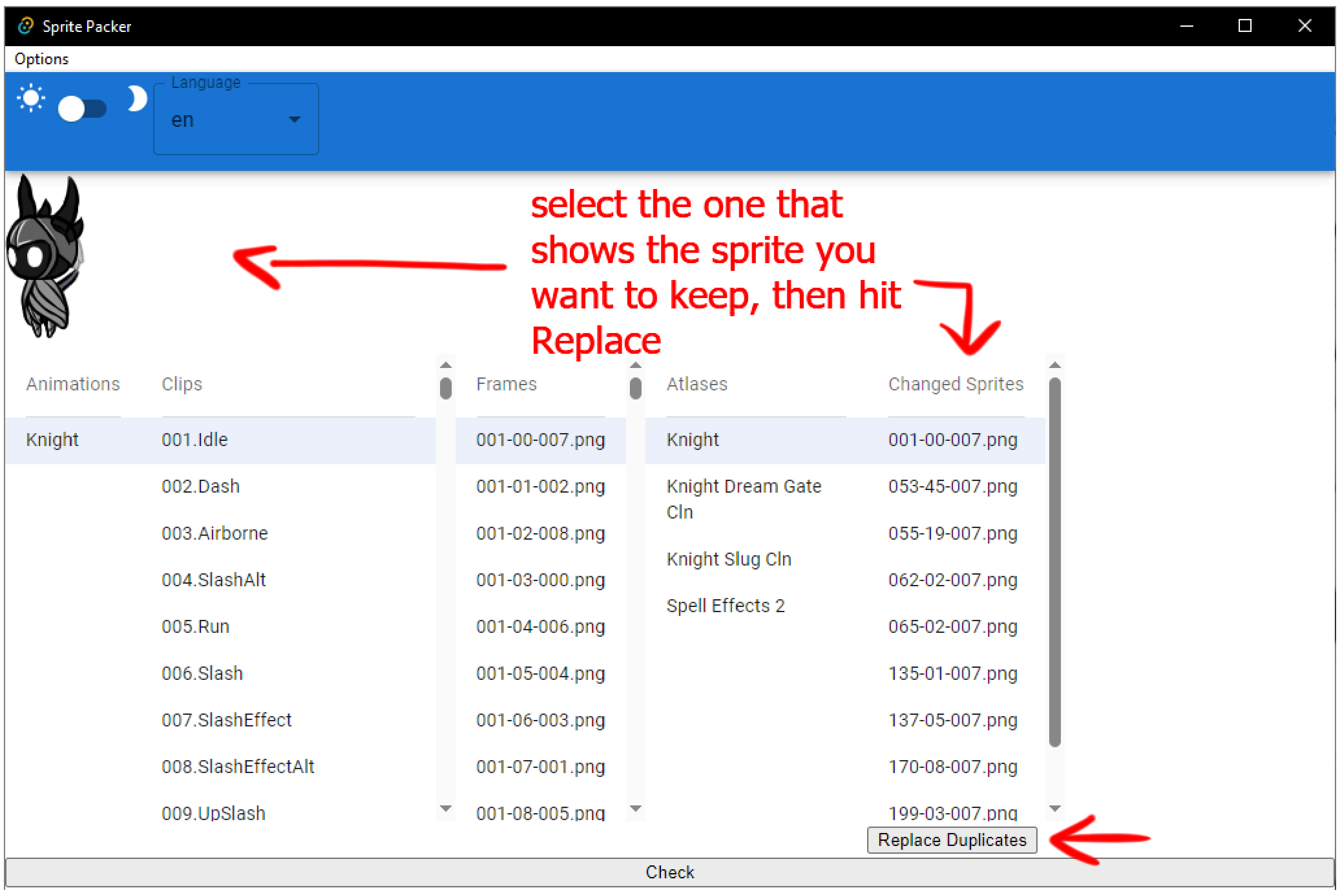
To replace duplicates in **SP** and **NSP**, click on the atlas you are working with and click ‘check’. In **SP**, duplicates that need your attention will appear in the top left. Select the sprite you want to keep. Hit replace and all sprites sharing that ID will become the sprite you selected. Continue to check and replace until the message in the top left says ‘good to go’.


Output Message

```
137-05-007.png
001-00-007.png
[Error02] Repeated sprites below in this collection are modified differently.Try again after
199-03-007.png
001-00-007.png
```

To replace a bunch of dupes at once with **SP,** select the frame of the sprite you want to keep and click ‘add’ under the ‘Changed Files’ section. Do this with all the sprites you want to keep, then select the original-atlas you’re working on, click ‘check’, and then ‘Replace All’ under the ‘Changed Files’ section. SP will automatically replace all the dupes it finds with the sprites you selected earlier. To remove a sprite from the ‘Changed Files’ section, double click it.

In **NSP** when you select the atlas, it will stretch the page. Scroll to the bottom to find ‘check’ and click it. All duplicates will appear on the right. Select the sprite you want to keep and hit ‘Replace Duplicates’. To clear your ‘Changed Sprites’ list, click ‘Options’ and hit ‘Refresh’.

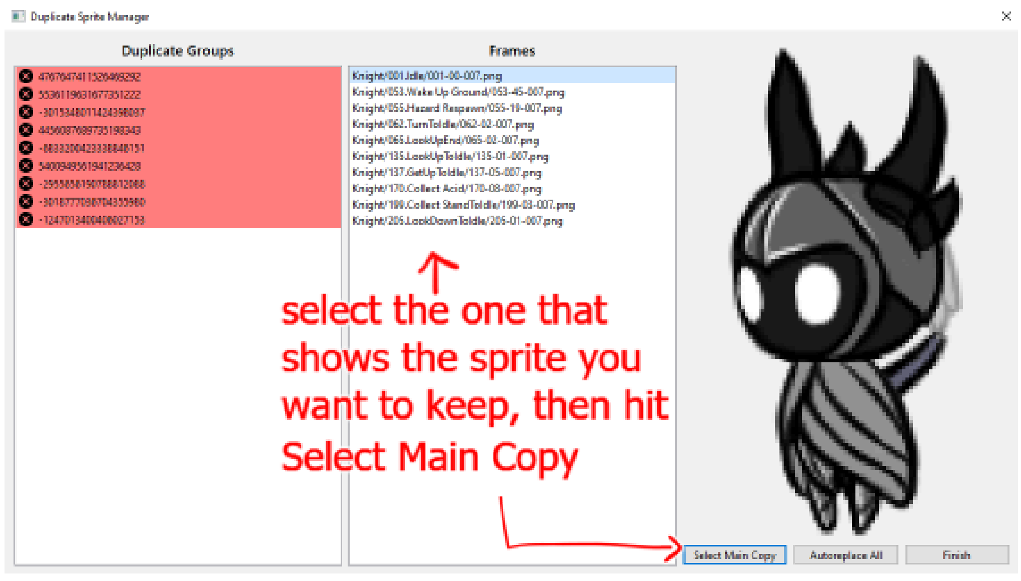


Animations	Clips	Frames	Atlases	Changed Sprites
Knight	001.Idle	001-00-007.png	Knight	001-00-007.png
	002.Dash	001-01-002.png	Knight Dream Gate Cln	053-45-007.png
	003.Airborne	001-02-008.png	Knight Slug Cln	055-19-007.png
	004.SlashAlt	001-03-000.png	Spell Effects 2	062-02-007.png
	005.Run	001-04-006.png		065-02-007.png
	006.Slash	001-05-004.png		135-01-007.png
	007.SlashEffect	001-06-003.png		137-05-007.png
	008.SlashEffectAlt	001-07-001.png		170-08-007.png
	009.UpSlash	001-08-005.png		199-03-007.png

For both **SP** and **NSP**, if nothing else needs to be replaced, ‘check’ will become ‘pack’. Clicking ‘pack’ will create your custom sprite sheet. In **SP**, it will automatically save in the 0.Atlases folder. In **NSP**, you will be asked where you want to save it.

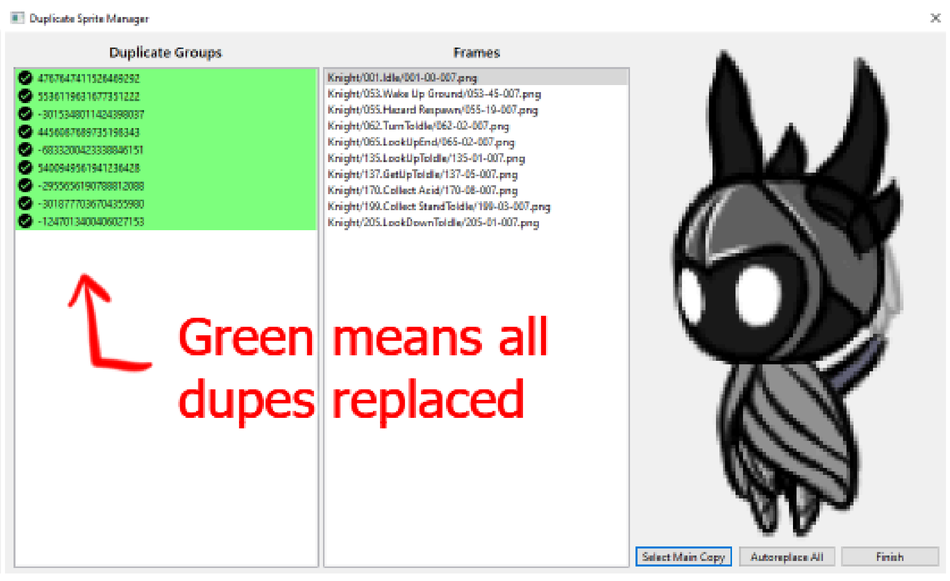
For a video tutorial on using SP, click here: How make Hollow Knight's skins with SpritePacker (with audio make by Akkitty08)
There is no video tutorial for using NSP.

To replace duplicates in **CKC**, click View Duplicate Groups. Viewing ‘Animation Duplicate Groups’ will only show dupes that have IDs matching the currently selected animation. Viewing ‘All Duplicate Groups’ will show all dupes for the currently selected atlas. If a duplicate group has a green check mark, then all dupes match. If it is a red X, then either a dupe needs to be replaced, or none of the sprites in that group have been customized yet. To replace a sprite, click on a group and select the sprite you want to use. Hit ‘Select Main Copy’. Do this for every group that has a red X (skipping groups you haven’t customized yet). Hitting ‘Autoreplace All’ will automatically select your most recently edited sprite for each group as the Main Copy. Once you’re done, hit ‘Finish’.



Duplicate Groups:
- ✗ 4767647411526469292
- ✗ 5536119631677351222
- ✗ -3015348011424398037
- ✗ 4456087689735198343
- ✗ -6833200423338846151
- ✗ 5400949561941236428
- ✗ -2955656190788812088
- ✗ -3018777036704355980
- ✗ -1247013400406027153

Frames:
- Knight/001.Idle/001-00-007.png
- Knight/053.Wake Up Ground/053-45-007.png
- Knight/055.Hazard Respawn/055-19-007.png
- Knight/062.TurnToIdle/062-02-007.png
- Knight/065.LookUpEnd/065-02-007.png
- Knight/135.LookUpToIdle/135-01-007.png
- Knight/137.GetUpToIdle/137-05-007.png
- Knight/170.Collect Acid/170-08-007.png
- Knight/199.Collect StandToIdle/199-03-007.png
- Knight/205.LookDownToIdle/205-01-007.png



Duplicate Groups:
- ✓ 4767647411526469292
- ✓ 5536119631677351222
- ✓ -3015348011424398037
- ✓ 4456087689735198343
- ✓ -6833200423338846151
- ✓ 5400949561941236428
- ✓ -2955656190788812088
- ✓ -3018777036704355980
- ✓ -1247013400406027153

Frames:
- Knight/001.Idle/001-00-007.png
- Knight/053.Wake Up Ground/053-45-007.png
- Knight/055.Hazard Respawn/055-19-007.png
- Knight/062.TurnToIdle/062-02-007.png
- Knight/065.LookUpEnd/065-02-007.png
- Knight/135.LookUpToIdle/135-01-007.png
- Knight/137.GetUpToIdle/137-05-007.png
- Knight/170.Collect Acid/170-08-007.png
- Knight/199.Collect StandToIdle/199-03-007.png
- Knight/205.LookDownToIdle/205-01-007.png

To pack with **CKC**, create a new folder outside of your sprites folder. Click ‘select output folder path’ and select this folder. Whenever you hit pack, it will save to your selected folder. Note: CKC will always pack all of the loaded atlases. If you only want it to pack specific ones, make sure to disable the atlases you don’t want under ‘atlases/collections’.

For a video tutorial on using CKC, click here: (NEW) CustomKnight Creator Tutorial

Once you’ve generated your atlases, go into the skins folder under the Custom Knight mod folder (if you can’t find it, boot up Hollow Knight and click ‘open folder’ under the custom knight mod menu), create a new folder and name your skin. Put your new atlases inside. Use the default skin as a guide and rename all of your atlases to match the names of the ones in the default skin. For any sprites in the Default skin that are single, stand alone sprites (Such as the charms, the inventory, and the particles), you can copy and paste those pngs directly into your own skin folder, then edit them directly. No packer needed!

Boot up your game and you should see your skin in the skin list.

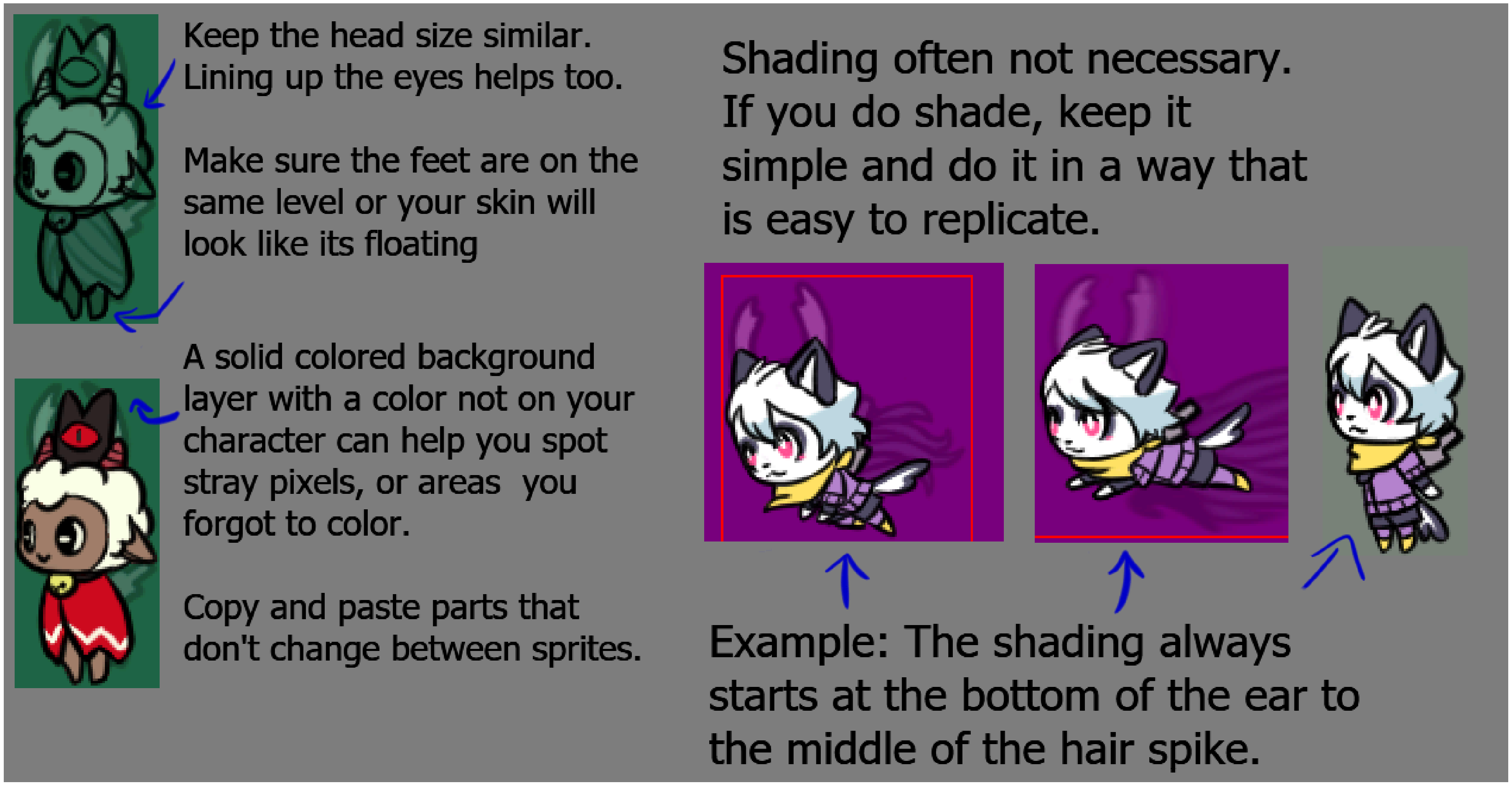
Tips and Tricks

Here is a checklist that groups the various animations together by type. You can use it to keep track of your progress.
Копия Копия Custom Knight Checklist - Template
Click File -> ‘make a copy’ to create your own private copy that you can save and edit.

When designing your skin, you might need to simplify your character to better fit the size limits and to make it easier to animate. Avoid using too many fine details or heavy shading. Consistency between sprites is easier to maintain with simpler designs. If you are using shading, try to shade in a way that’s easy to mimic in each sprite, and if your design has a pattern, try to position it in a way that is easily replicated. Patterns can start to wander or change sizes across the sprites during the animation process if you’re not careful, so design in a way to minimize the chances of that happening.

Work smarter not harder! Keep a file with a copy of your skin’s head facing different directions, and copy paste it (and anything else that doesn’t change) whenever you can rather than re-draw for every sprite.

Try to make the head the same size as the original knight’s and the eyes in the same spot. This way you can use the original knight as your guide to help you with keeping your own skin consistent between frames. The more of the original knight you can use as a guide, the easier it will be to make your skin. The hitbox won’t change in-game even if you draw a much smaller or much larger character, so matching the knight’s size will also make your skin feel good to use, as the hitbox will match better.


Keep the head size similar. Lining up the eyes helps too.

Make sure the feet are on the same level or your skin will look like its floating

A solid colored background layer with a color not on your character can help you spot stray pixels, or areas you forgot to color.

Copy and paste parts that don't change between sprites.

Shading often not necessary. If you do shade, keep it simple and do it in a way that is easy to replicate.

Example: The shading always starts at the bottom of the ear to the middle of the hair spike.

Krita tip: You can import all the frames of an animation into the program at once. Click ‘File’ -> ‘Import Animation Frames…’ and add all the frames in the animation folder. Working on the whole animation at the same time can help with making sure the animation looks smooth.

Animations that end in the air (such as 096.SD Air Brake and 098.Double Jump) go straight into the ‘003.Airborne’ animation, starting at sprite 003-03-021.png. If you are drawing a sprite that doesn’t follow the knight’s anatomy, make sure you end those animations in a way that goes smoothly into the airborne animation.

Get into the habit of checking for dupes every time you finish an animation with your packer’s check and replace function. It’s quick and easy to do, and saves you the pain of realizing you drew a frame you didn’t need to draw because you didn’t realize it was a dupe until later.

Pack your atlas periodically and check how it looks in game every now and then. It’s better to check and make adjustments to animations while you’re working on them rather than waiting until the end, and seeing your skin in action can be a nice motivation boost to keep you going.

If you open up the Custom Knight Mod menu in game, under ‘keybinds’ you can set a keybind to re-load your skins. This is extremely useful if you are making edits while leaving the game open, as you can quickly refresh the sprites and see how it looks without having to open the menu over and over again.

Unmarked Dupes

The following sprites from the Knight folder are the same sprite despite not having a matching ID. If you are using **CKC,** you do not need to worry about these sprites. CKC will automatically detect and replace them. If you are using **SP** or **NSP,** you will have to manually edit the images to make them match. Once they match, SP and NSP will take care of any remaining dupes.

005-09-009 = 006-09-156	031-00-195 = 046-20-339	053-33-349 =046-05-341	069-00-178 = 139-05-382
022-01-230 = 022-05-219	031-01-191 = 033-03-189	053-33-349 =046-06-349	071-00-476 = 116-00-734
022-02-233 = 022-06-217	039-00-291 = 040-00-296	053-33-349 =046-07-342	
069-02-176 = 068-00-357	053-00-343 = 053-25-352	053-33-349 =046-08-340	
069-01-173 = 068-01-298	053-00-343 = 053-26-346	053-33-349 =046-12-333	

Unused Sprites

The following animations are not used in-game and can be ignored.
026. Hit Crack Appear
036. Idle Wind

The following animations contain sprites that are unused. The listed sprite positions can be skipped.
040. Sit Lean (Positions 01 and 02 unused.)
098. Double Jump (Positions 04, 05, 06, and 07 unused.)

Advanced Skin Making

Along with skinning the knight, you can also change enemies and environments, including Nosk or the little knight posters and plushies in Bretta's bedroom. Even the dialogue and cinematics can be changed through Custom Knight. You can also include alternate versions of atlases that can be accessed in-game through the alternates option in the Custom Knight Mod menu as well as customize a few color settings related to the knight.

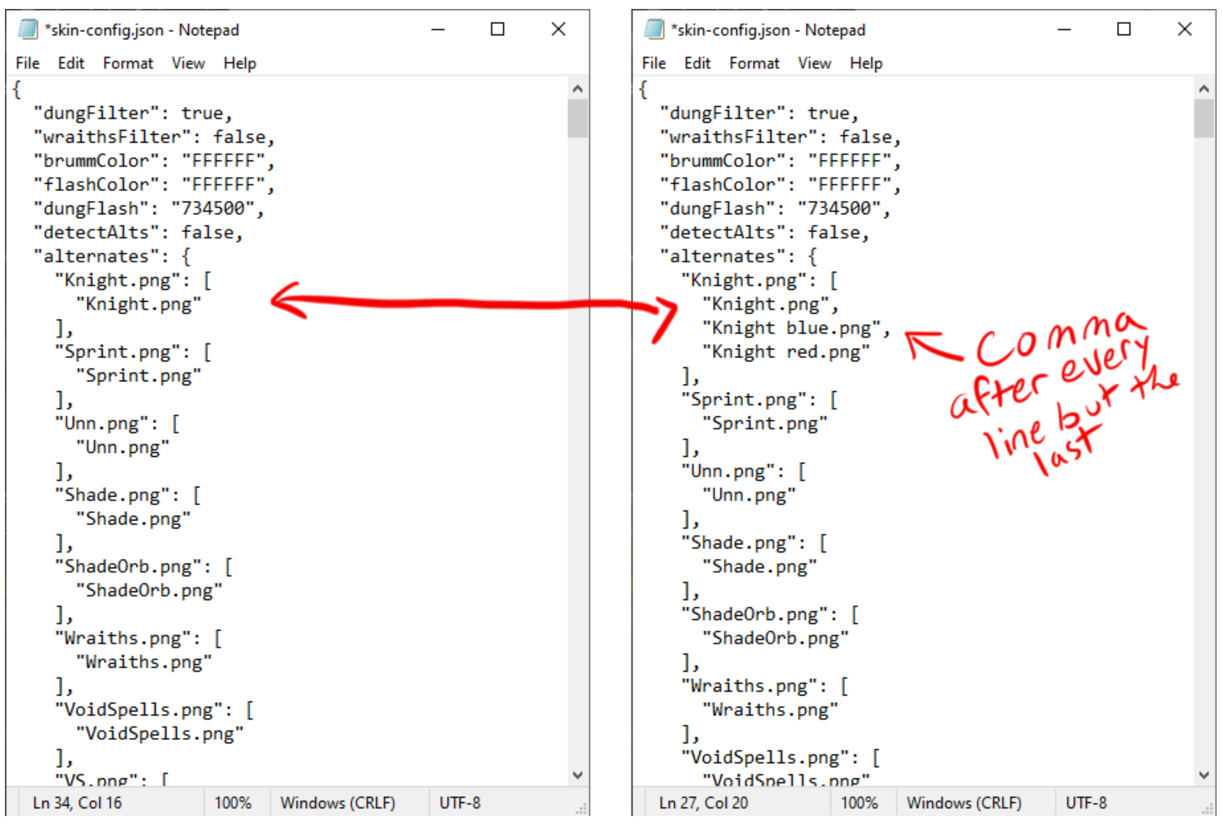
Skin Config

Once you've booted up the game at least once with your custom skin installed, when you exit the game you will find two new files in your custom skin folder: skin-config.json and skin-settings.json. The skin-config.json is the one you want to look at.

dungFilter - set to false to remove the brown filter applied to the dung cloud created by Defender's Crest.
wraithsFilter - set to true to remove the alpha filter applied to the Wraiths atlas. WARNING: This will make all the black on the sheet visible. Only do this if you are customizing the entire atlas.
brummColor - controls the color the knight flashes when Carefree Melody activates. Default is FFFFFF
flashColor - controls the color the knight flashes when using the Focus ability. Default is FFFFFF
dungFlash - controls the color enemies flash when taking damage from Defender's Crest. Default is 734500
detectAlts - set to true to automatically attempt to detect any alts in the skin

The alternates section is how you can manually set up any alternate sprite sheets you want to include in your skin. You will find the names of all the pngs that can have alts here. To set up an alt, all you need to do is put the name of the png underneath the atlas you want to associate it with. Make sure it's on its own line. If you have multiple files for the same atlas, make sure all but the last line has a comma at the end of it.

For example, say you wanted to add 3 versions of the knight sheet, so you include 3 copies of the knight png in your skin (Knight.png, Knight blue.png, and Knight red.png). This is how you would change the alternates section so that all 3 are available:


*skin-config.json - Notepad

```
{
  "dungFilter": true,
  "wraithsFilter": false,
  "brummColor": "FFFFFF",
  "flashColor": "FFFFFF",
  "dungFlash": "734500",
  "detectAlts": false,
  "alternates": {
    "Knight.png": [
      "Knight.png"
    ],
    "Sprint.png": [
      "Sprint.png"
    ],
    "Unn.png": [
      "Unn.png"
    ],
    "Shade.png": [
      "Shade.png"
    ],
    "ShadeOrb.png": [
      "ShadeOrb.png"
    ],
    "Wraiths.png": [
      "Wraiths.png"
    ],
    "VoidSpells.png": [
      "VoidSpells.png"
    ],
    "VS.png": [
```

Ln 34, Col 16 100% Windows (CRLF) UTF-8

*skin-config.json - Notepad

```
{
  "dungFilter": true,
  "wraithsFilter": false,
  "brummColor": "FFFFFF",
  "flashColor": "FFFFFF",
  "dungFlash": "734500",
  "detectAlts": false,
  "alternates": {
    "Knight.png": [
      "Knight.png",
      "Knight blue.png",
      "Knight red.png"
    ],
    "Sprint.png": [
      "Sprint.png"
    ],
    "Unn.png": [
      "Unn.png"
    ],
    "Shade.png": [
      "Shade.png"
    ],
    "ShadeOrb.png": [
      "ShadeOrb.png"
    ],
    "Wraiths.png": [
      "Wraiths.png"
    ],
    "VoidSpells.png": [
      "VoidSpells.png"
```

Ln 27, Col 20 100% Windows (CRLF) UTF-8

Comma after every line but the last

The alts can be named anything you want. Just make sure they're associated with the correct atlas in this file and make sure a default named version is included.

If you do rename all of your files in this way, make sure to also edit the skin-settings.json so that it is also using your new name.
Example:

```
{
  "selectedAlternates": {
 "Knight.png": "Knight blue.png",
 "Sprint.png": "Sprint.png",
```

If you don't do this, any renamed atlas will appear vanilla in-game, but don't worry. Just go into the in-game alternates menu and apply your renamed atlas to fix it.

Enemy and Environment Skins

Custom Knight includes a 'swapper' feature, that allows you to customize more than just the knight. Creating custom enemies and changing the backgrounds for your skin can really help with immersion, but does require a bit of extra work. You will need to dump the sprites yourself in most cases, though you can also find a lot of enemy sprites already dumped in the Hollow Knight Modding discord. The sprite dumps in the modding discord are compatible with **SP** and **NSP**. CKC cannot pack enemy atlases, so you must use a different packer if you plan to use a packing program to create skins for enemies and npcs.

You can find the enemy and npc sprite dumps here (Requires being part of the HK Modding Discord):
https://discord.com/channels/879125729936298015/1100280517808640110
If the thing you want to skin is not already dumped, you will have to dump the sprites yourself.

Dumping Sprites

There are two ways you'll be dumping sprites. Dumping with the Custom Knight mod itself will not only give you the sprites for the various backgrounds, it will also give you the names you will need for your skin. Just like how the default skin was your naming guide for the knight files, you can use the dumped files as your naming guide for your enemy and background skins.

The second dumping method is through the use of the mod 'Game Object Dump' also called 'GoDump'. GoDump is what you will use to break the atlases up into individual sprites that you can use with your packing program.

With CK

To dump sprites with Custom Knight, go to the room or enemy you want to skin. Open the custom knight mod menu, click tools, and you will be presented with two dumping styles: Directory and Names.Json.

- ~~**'Directory'** dumps the sprites by a readable name and often involves multiple sub folders being generated for the room, so you may need to do some digging to find the sprites you're looking for. Images might be dumped multiple times and appear in different sub folders if they are used for different things in the room you are dumping. This dump method is faster than the other method, and the naming style can help you figure out what you're looking at, but you will have less creative control for your skin. You will need to match the folder pathing exactly when you copy the sprites you want to edit for your skin, otherwise your skin elements will not show up.~~ (As of CK 3.5, Names.json is the only dumping method)

- '**Names.Json**' **(Recommended)** dumps the sprites by their hash name, which is a jumble of numbers and letters, and dumps everything in the room into a single folder. There will be no duplicates with this method, but the hash names make it harder to figure out what you're looking at. This style of dumping will take longer, but the names.json file it creates will give you more control over your skin than the directory method. By editing the names.json file, you will be able to name the images within the folder anything you want, organize the images into sub folders, and assign different images to background elements that would normally use the same sprite.

Once you've decided what dump method you want to use, click 'Dump sprites'. Depending on the size of the room and how many sprites there are, this can take a while and you will experience considerable lag while the sprites are being dumped. If you want to dump the dialogue, make sure to trigger it while you have dumping turned on. Dumped dialogue will appear as a txt file in the dump folder. It's easiest to dump the dialogue after waiting for the room to finish dumping, so you can move without the lag.

You will see a percentage on the left side of the screen that will tell you how far along the dump is. Once it hits 100%, it will disappear and you can turn dumping off. If it hits 100%, then drops back down to something in the 90s repeatedly, it likely means that something in the scene has a particle effect that is constantly generating new sprites. In this case, it will never hit 100%, but you can still safely turn it off. It will have dumped what you needed.

NOTE: If what you want to customize does not dump, then it likely cannot be customized with CK at this time.

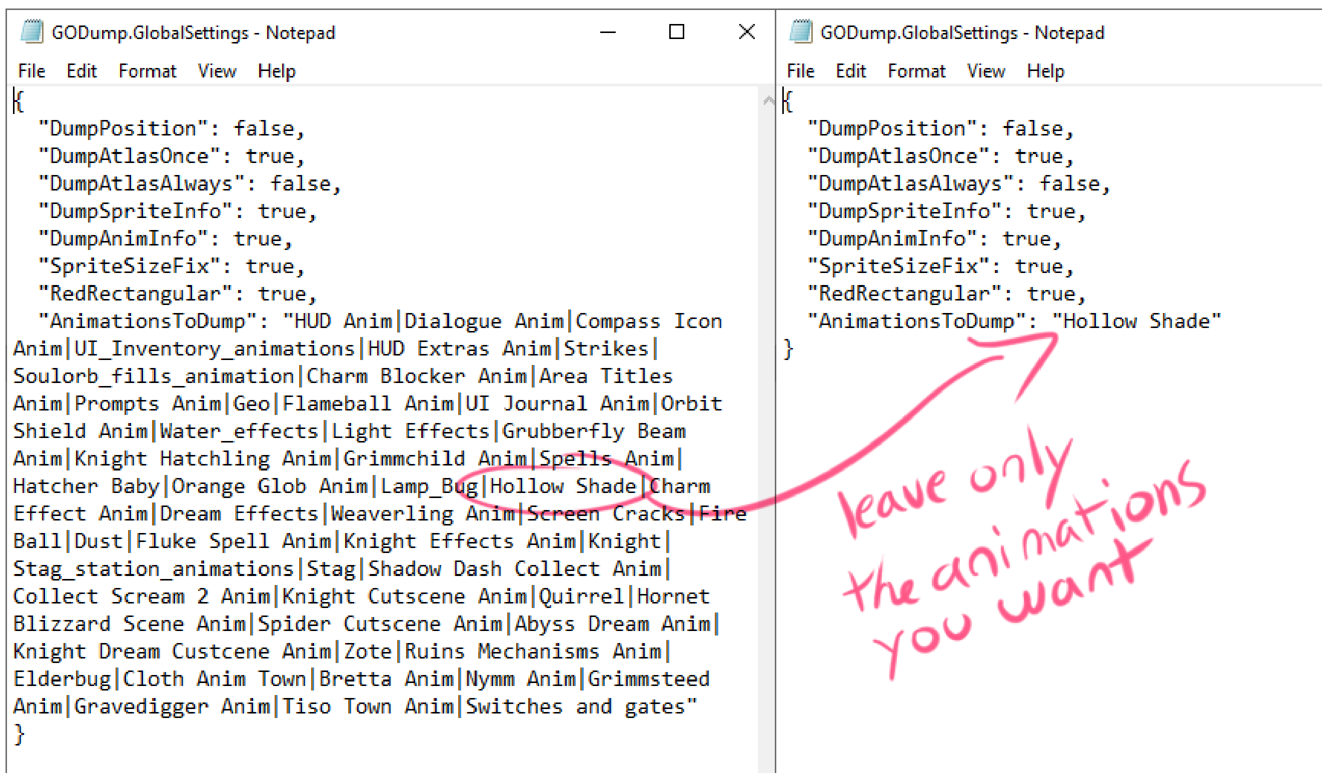
With GoDump

The GoDump that is on the Lumafly installer is not compatible with SP or NSP. You will need to download this version of GoDump from the Modding Discord and manually install it (Requires being part of the HK Modding Discord):
https://discord.com/channels/879125729936298015/879133769070706698/889708344170913802
Both **NSP** and **SP** require the sprite info from this version of GoDump. If you are unsure how to install it, you can ask for help in the discord server.

Once installed, start the game and a **GODump.GlobalSettings.json** file will be created in your 'AppData\LocalLow\Team Cherry\Hollow Knight' folder (The same location you needed to put the sprites for **SP**). If you had this file already from a previous version of GoDump, close the game, delete this file, and start the game again. A new copy of the file will be generated.

Enter the scene you want to dump and hit F2. Atlases will be generated in 'AppData\LocalLow\Team Cherry\Hollow Knight\atlases'. If you had generated atlases previously, duplicates of the previous atlases may appear. This is fine, and you can delete these images after you are finished using GoDump to free up space on your computer.

Hit F3 and the **GODump.GlobalSettings.json** file will update with the names of the animations in your current scene. Edit the file and delete any animations you do not want to dump, then save. If you're dumping multiple animations, put a | between them. If you are unsure what each animation is, you can compare the names to the dumped atlases that were generated earlier.

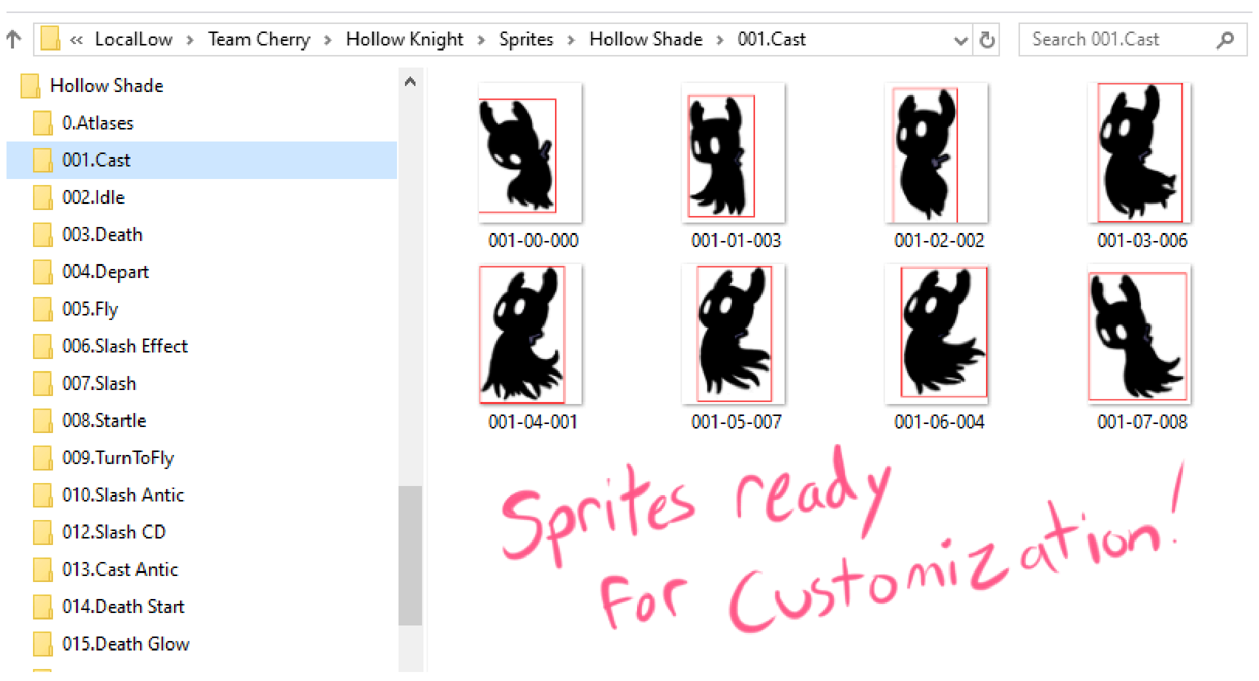

GODump.GlobalSettings - Notepad
File Edit Format View Help
GODump.GlobalSettings - Notepad
File Edit Format View Help
leave only the animations you want

```
{
  "DumpPosition": false,
  "DumpAtlasOnce": true,
  "DumpAtlasAlways": false,
  "DumpSpriteInfo": true,
  "DumpAnimInfo": true,
  "SpriteSizeFix": true,
  "RedRectangular": true,
  "AnimationsToDump": "HUD Anim|Dialogue Anim|Compass Icon
Anim|UI_Inventory_animations|HUD Extras Anim|Strikes|
Soulorb_fills_animation|Charm Blocker Anim|Area Titles
Anim|Prompts Anim|Geo|Flameball Anim|UI Journal Anim|Orbit
Shield Anim|Water_effects|Light Effects|Grubberfly Beam
Anim|Knight Hatchling Anim|Grimmchild Anim|Spells Anim|
Hatcher Baby|Orange Glob Anim|Lamp_Bug|Hollow Shade|Charm
Effect Anim|Dream Effects|Weaverling Anim|Screen Cracks|Fire
Ball|Dust|Fluke Spell Anim|Knight Effects Anim|Knight|
Stag_station_animations|Stag|Shadow Dash Collect Anim|
Collect Scream 2 Anim|Knight Cutscene Anim|Quirrel|Hornet
Blizzard Scene Anim|Spider Cutscene Anim|Abyss Dream Anim|
Knight Dream Custcene Anim|Zote|Ruins Mechanisms Anim|
Elderbug|Cloth Anim Town|Bretta Anim|Nymm Anim|Grimmsteed
Anim|Gravedigger Anim|Tiso Town Anim|Switches and gates"
}
```

```
{
  "DumpPosition": false,
  "DumpAtlasOnce": true,
  "DumpAtlasAlways": false,
  "DumpSpriteInfo": true,
  "DumpAnimInfo": true,
  "SpriteSizeFix": true,
  "RedRectangular": true,
  "AnimationsToDump": "Hollow Shade"
}
```

Hit F4 and the sprites will begin dumping in 'AppData\LocalLow\Team Cherry\Hollow Knight\sprites'. This step may take a while depending on how many animations there are and how many sprites need to be dumped. Some NPCs that appear in multiple places, such as Cornifer and the Stag, do not dump spriteinfo. If a spriteinfo file is not created in the 0.Atlases folder, unfortunately that means you cannot use the packers to pack that sprite sheet.

Once the dump is finished, you can exit the game. The sprites can now be edited and packed with **NSP** or **SP**.



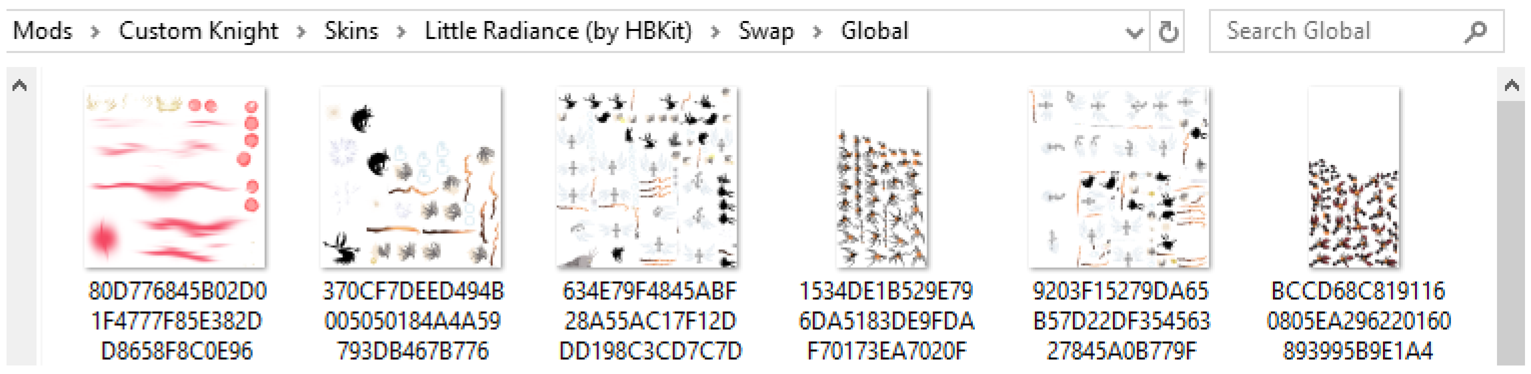
NOTE: The sprite info that dumps with the Radiance atlases does not pack correctly.
Replace the 0.Atlases folder found within Radiance Anim and its contents with the one found here to fix it. Radiance Anim.zip
If you downloaded the pre-dumped Radiance sprites from the modding discord, you do not need to do this, as they have already been fixed.

Swapper

The swapper function of Custom Knight is what changes the backgrounds and enemies. All swapper files go into the 'Swap' folder under your skin and there are two different ways to set up swaps. The global folder can be used to apply a skin to enemies globally, so the skin will affect every version of that enemy no matter where they are encountered. Path swapping is room by room and is used to swap the environment sprites.

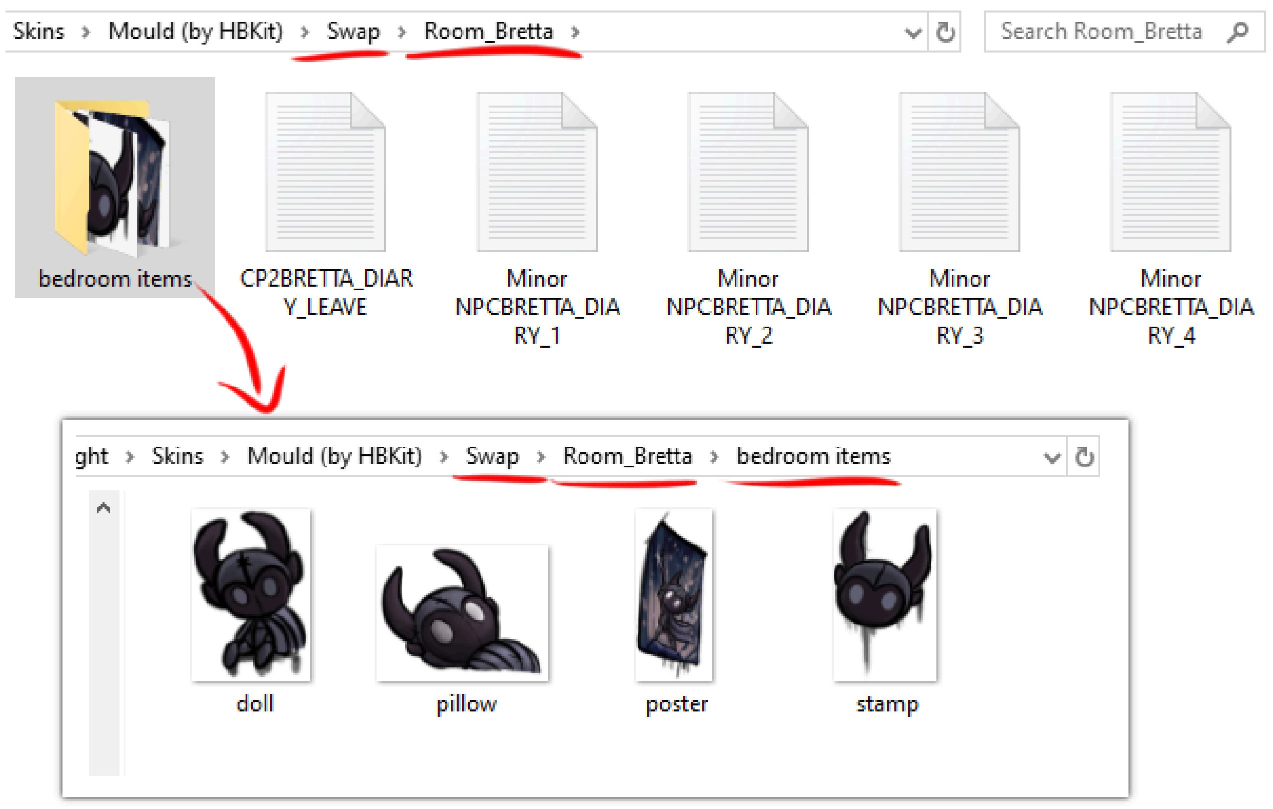
Global Swap

To do global swaps, all you need to do is put the enemy atlas you created into the 'Global' folder within the swap folder. If there is not a 'Global' folder, make one. The files here must use their hash names or the names found in the 'aliases.json' located in the 'Custom Knight' folder. If you dumped the atlas earlier using the names.json method, it will have dumped with the hash name so you can copy that name when you rename your file. If you used the directory dump method, then you will need to find the atlas in the dumped 'Global' folder and can copy the hash from there.
Example:

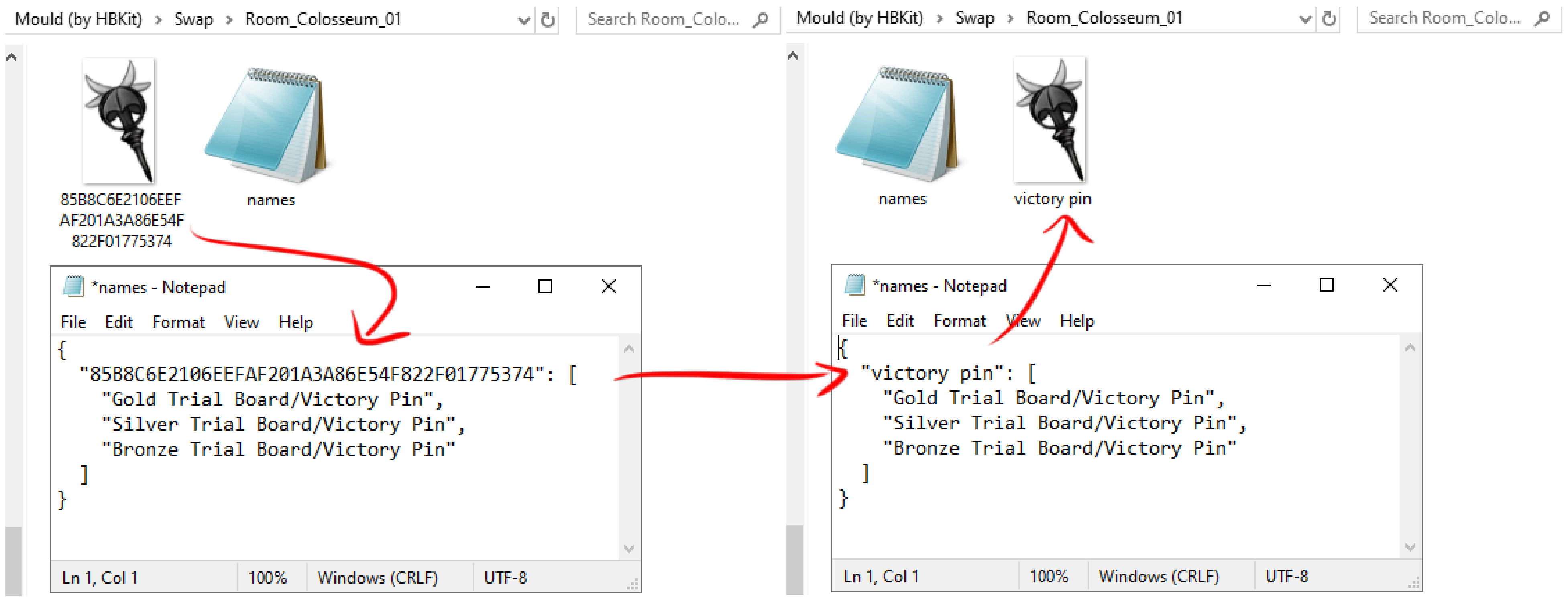

Mods › Custom Knight › Skins › Little Radiance (by HBKit) › Swap › Global
Search Global
80D776845B02D0 1F4777F85E382D D8658F8C0E96
370CF7DEED494B 005050184A4A59 793DB467B776
634E79F4845ABF 28A55AC17F12D DD198C3CD7C7D
1534DE1B529E79 6DA5183DE9FDA F70173EA7020F
9203F15279DA65 B57D22DF354563 27845A0B779F
BCCD68C819116 0805EA296220160 893995B9E1A4

Path Swap

Path swapping requires the name of the room you are customizing, which you can get by dumping that room with CK. There are two ways to swap things in the environment, depending on how you dumped the room before. If you dumped with the **Directory** method, you will need to match the names and file path of the sprite you are replacing directly. Sometimes this means you will need to make multiple sub folders in a single folder just for one image. As long as you match the file structure and the naming structure, your images will swap.
NOTE: Path Swapping is being phased out. While it still works, it is not recommended.



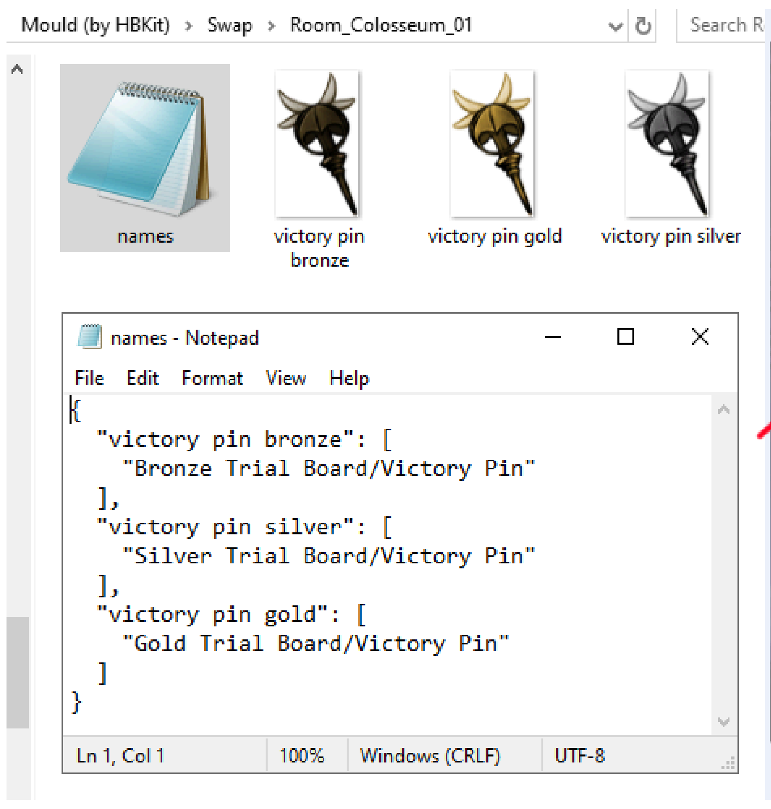
If you dumped with the **Names.json** method, then you just need to put all the images you want to swap into the room folder with the names.json file. When you open the names.json file, you can delete anything that you aren't swapping. This is also where you can set the names of your images, so that you do not need to use the hash names.


Mould (by HBKit) › Swap › Room_Colosseum_01
Search Room_Colo...
85B8C6E2106EEF AF201A3A86E54F 822F01775374
names
*names - Notepad
File Edit Format View Help
Ln 1, Col 1 100% Windows (CRLF) UTF-8
Mould (by HBKit) › Swap › Room_Colosseum_01
Search Room_Colo...
names
victory pin
*names - Notepad
File Edit Format View Help
Ln 1, Col 1 100% Windows (CRLF) UTF-8

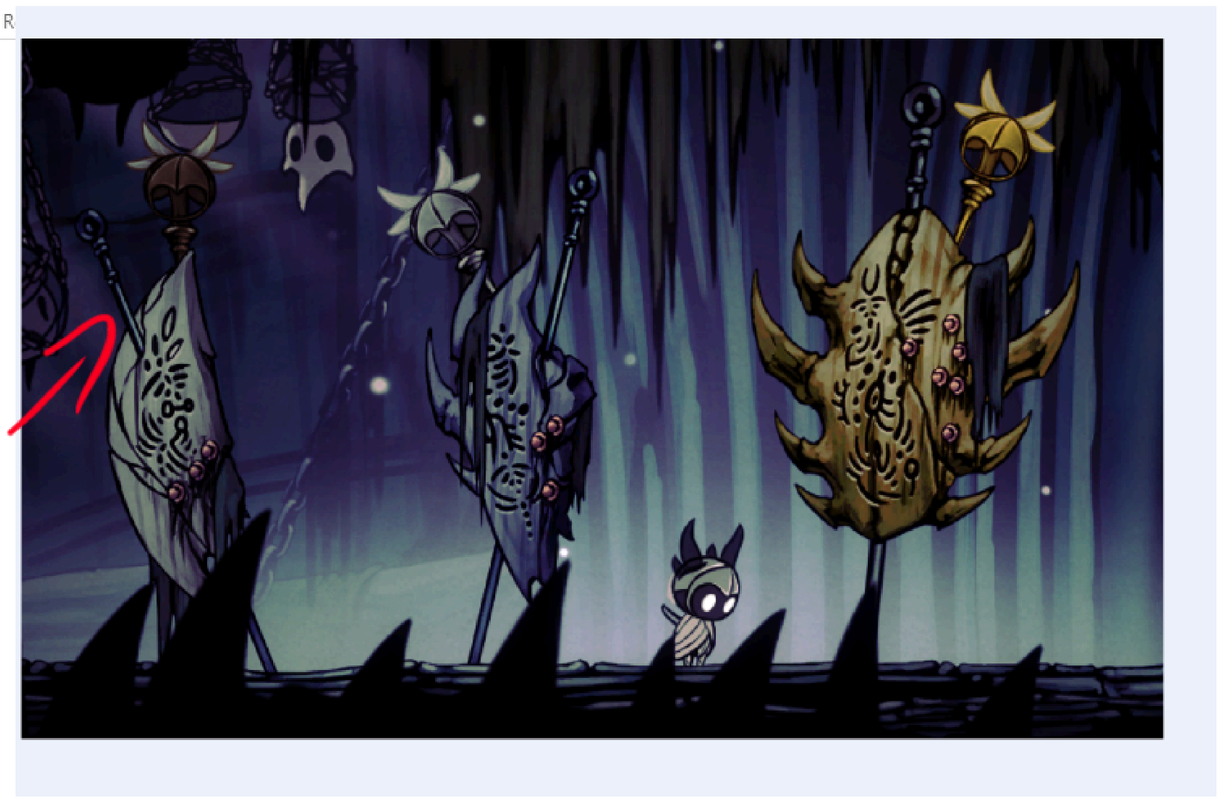
```
{
  "85B8C6E2106EEFAF201A3A86E54F822F01775374": [
    "Gold Trial Board/Victory Pin",
    "Silver Trial Board/Victory Pin",
    "Bronze Trial Board/Victory Pin"
  ]
}
```

```
{
  "victory pin": [
    "Gold Trial Board/Victory Pin",
    "Silver Trial Board/Victory Pin",
    "Bronze Trial Board/Victory Pin"
  ]
}
```

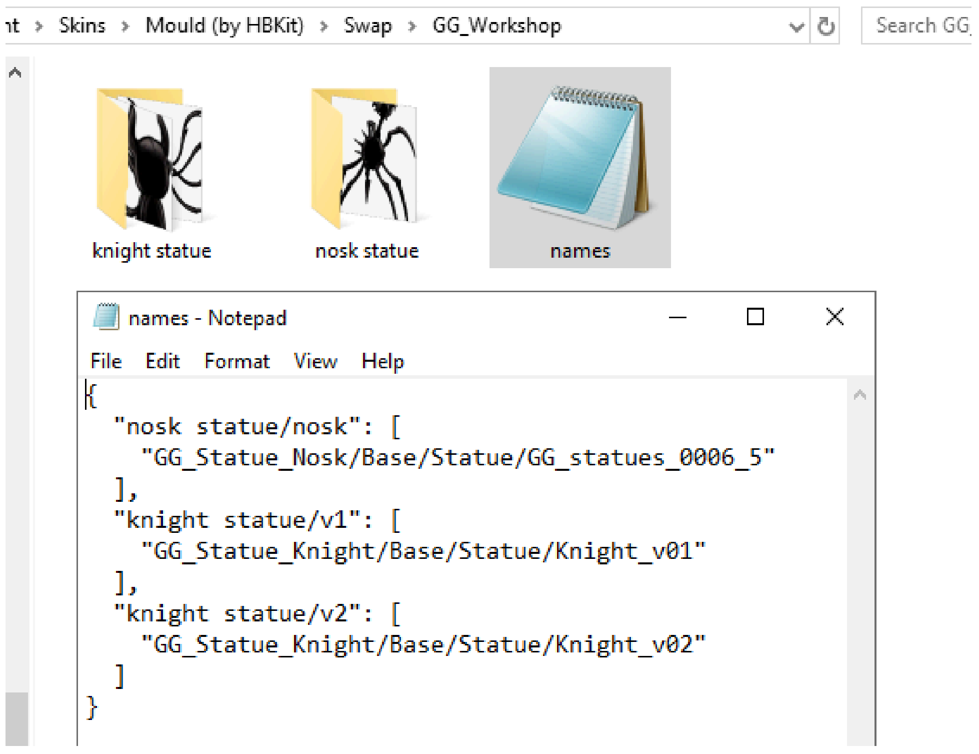
When you look at the names file, you'll notice that there might be multiple elements tied to a single image, like the above victory pin in the colosseum. By editing the names file, you can make those elements use different images instead, allowing you to use different images for something that normally uses the same one.


Mould (by HBKit) › Swap › Room_Colosseum_01
Search R
names
victory pin bronze
victory pin gold
victory pin silver
names - Notepad
File Edit Format View Help
Ln 1, Col 1 100% Windows (CRLF) UTF-8

```
{
  "victory pin bronze": [
    "Bronze Trial Board/Victory Pin"
  ],
  "victory pin silver": [
    "Silver Trial Board/Victory Pin"
  ],
  "victory pin gold": [
    "Gold Trial Board/Victory Pin"
  ]
}
```



If you want to put images into sub-folders, you can by adding the folder name before the image name and putting a forward slash between them. This is very useful if you're swapping a lot of elements in a single room and want to keep them organized.


› Skins › Mould (by HBKit) › Swap › GG_Workshop
Search GG
knight statue
nosk statue
names
names - Notepad
File Edit Format View Help

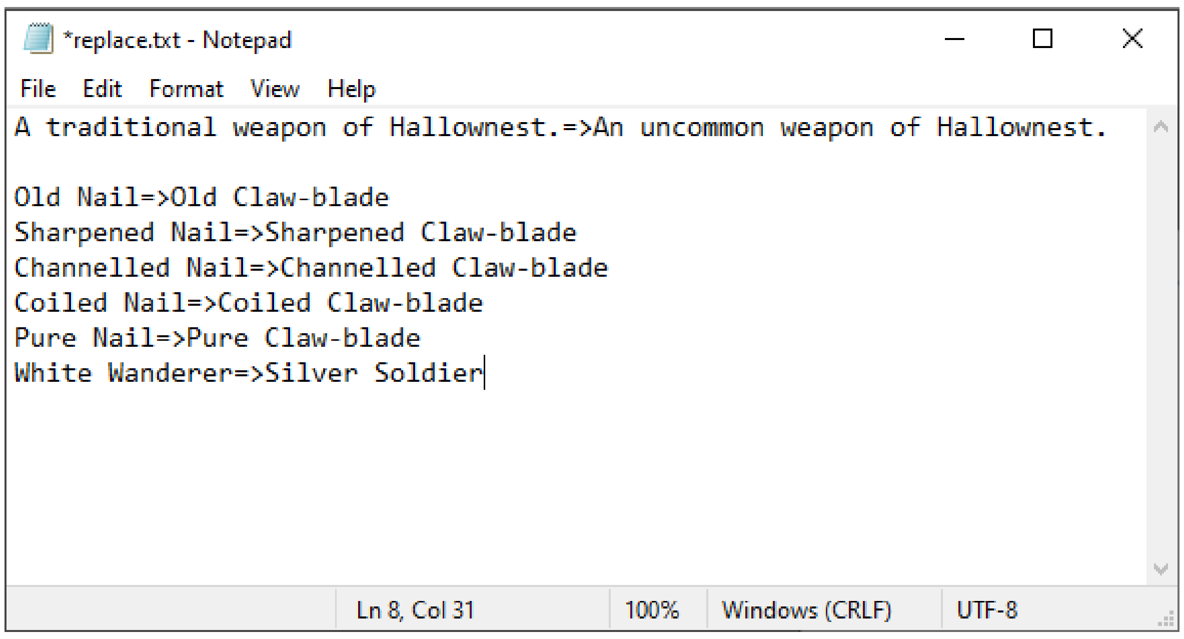
```
{
  "nosk statue/nosk": [
    "GG_Statue_Nosk/Base/Statue/GG_statues_0006_5"
  ],
  "knight statue/v1": [
    "GG_Statue_Knight/Base/Statue/Knight_v01"
  ],
  "knight statue/v2": [
    "GG_Statue_Knight/Base/Statue/Knight_v02"
  ]
}
```

When you swap a background element, make sure to actively check it in game. Sometimes sprites that dump as single images do not align perfectly and may need to be manually adjusted. Open the sprite in your art program of choice and move the image around the canvas, save, and reload your skin in game. Keep adjusting it until it lines up correctly.

Custom Text

Changing the text, whether that be dialogue, menus, or names, can be done in two different ways. The easiest way which requires no preparation is through the '**replace.txt**' file that is inside the Swap folder. The second way is to dump the individual txt files using the CK dumping method described earlier.

To text swap using the replace.txt file, all you need to do is type the text you want to replace, type '=>', then type what you want the new text to be. Every time the text on the left side of the arrow appears, it will automatically be replaced with what is on the right.



```
A traditional weapon of Hallownest.=>An uncommon weapon of Hallownest.

Old Nail=>Old Claw-blade
Sharpened Nail=>Sharpened Claw-blade
Channelled Nail=>Channelled Claw-blade
Coiled Nail=>Coiled Claw-blade
Pure Nail=>Pure Claw-blade
White Wanderer=>Silver Soldier
```

Order matters! Text at the top will be replaced before text at the bottom.

So if you type:
ghost=>little guy
Come no closer, ghost=>Hi sibling

The second line would not change, because it would have already changed to 'Come no closer, little guy' and would no longer match the text you want to replace. You would need to put 'ghost=>little guy' second to make both swap.

Keep order in mind when replacing text.

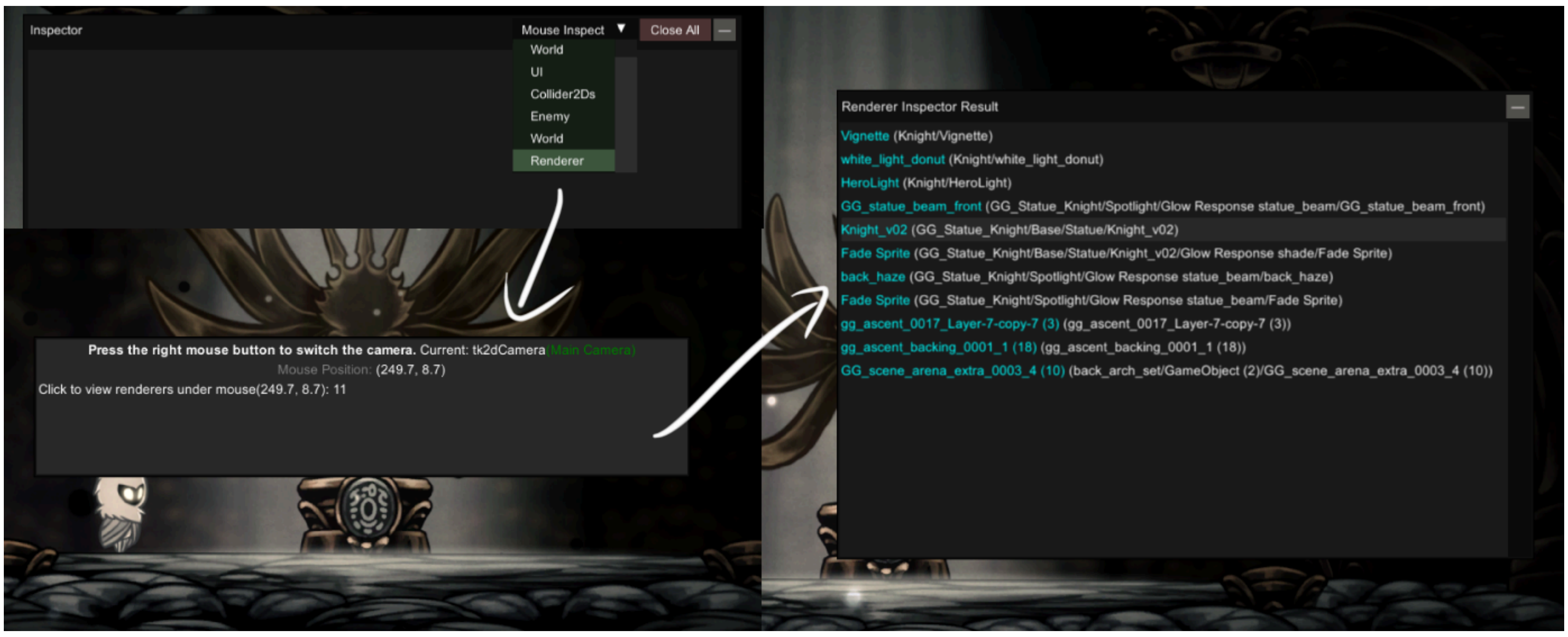
If you replace text by dumping the text file, you can edit the file directly. Only that instance of the text will change. For text that can be dumped in multiple rooms, such as menu text, journal entries, and inventory descriptions, you can put those text files in any folder within the swap folder and CK will see it. You can even create a folder called 'Text' inside of the swap folder if you want. But CK will not see any text files that are in a sub-folder inside of another folder.

Using the text files directly is useful for lengthier text that you want to swap, while using the replace file is useful to swap multiple instances of a word all at once. But if you don't want to fill your skin up with a cluster of text files, you can use the replace file to swap the majority of text in the game.

Unity Explorer

The mod '**Unity Explorer**' and its add-on '**Unity Explorer Plus**' are not required but are useful tools to help you identify and test sprites in game. Hitting F7 opens and closes the UE menu. The Inspector and Object Explorer can help you figure out the name and path of the element you want to swap.

If what you want to look at is overlapping a lot of elements or camera/respawn zones, set the inspector to 'Renderer'. It should say how many renderers are under your mouse. If it's none, right click until you reach the main camera and a number appears. Left click, and you will get a list of names. Click on any that you want to view more information about. You can check to see if it's the sprite you want by toggling the 'ActiveSelf' checkbox on and off.

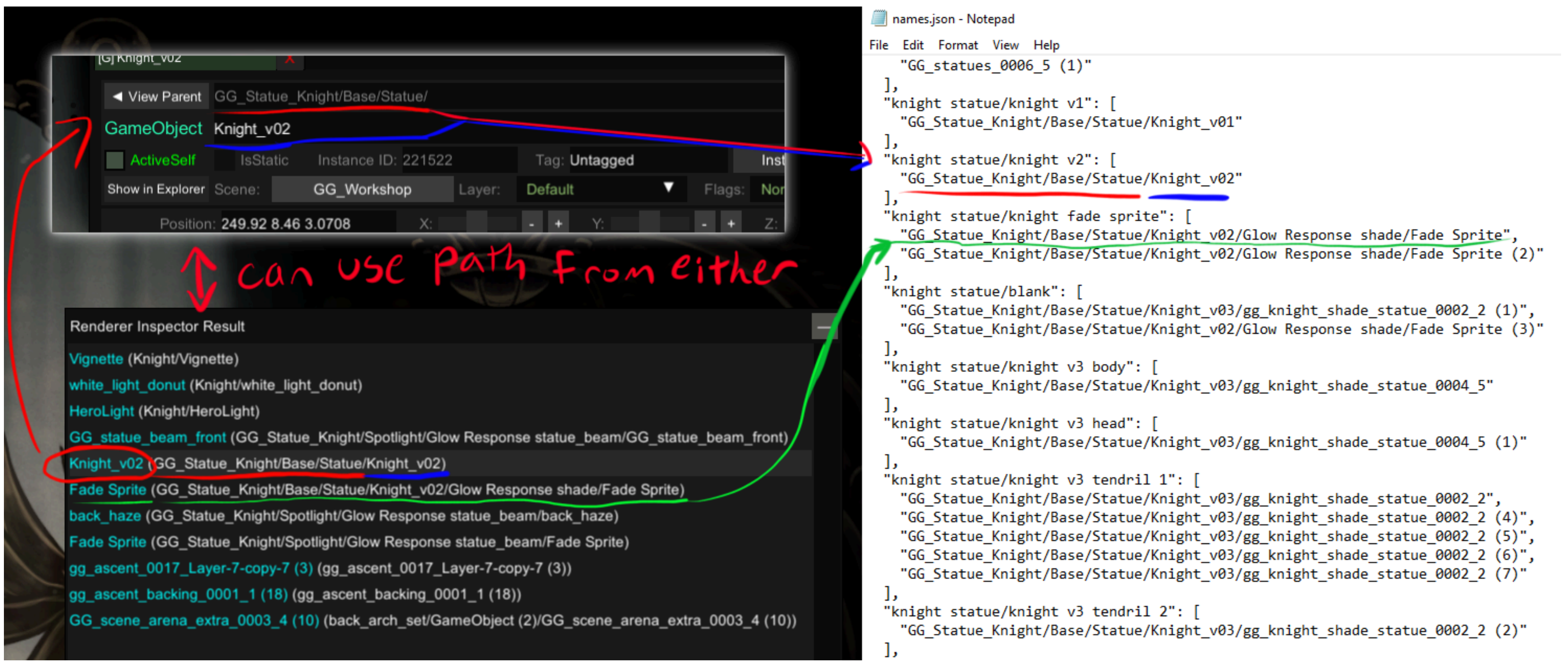


Press the right mouse button to switch the camera. Current: tk2dCamera(Main Camera)
Mouse Position: (249.7, 8.7)
Click to view renderers under mouse(249.7, 8.7): 11

Renderer Inspector Result

```
Vignette (Knight/Vignette)
white_light_donut (Knight/white_light_donut)
HeroLight (Knight/HeroLight)
GG_statue_beam_front (GG_Statue_Knight/Spotlight/Glow Response statue_beam/GG_statue_beam_front)
Knight_v02 (GG_Statue_Knight/Base/Statue/Knight_v02)
Fade Sprite (GG_Statue_Knight/Base/Statue/Knight_v02/Glow Response shade/Fade Sprite)
back_haze (GG_Statue_Knight/Spotlight/Glow Response statue_beam/back_haze)
Fade Sprite (GG_Statue_Knight/Spotlight/Glow Response statue_beam/Fade Sprite)
gg_ascent_0017_Layer-7-copy-7 (3) (gg_ascent_0017_Layer-7-copy-7 (3))
gg_ascent_backing_0001_1 (18) (gg_ascent_backing_0001_1 (18))
GG_scene_arena_extra_0003_4 (10) (back_arch_set/GameObject (2)/GG_scene_arena_extra_0003_4 (10))
```

The path associated with the image is the same as what is used in the names.json file that CK dumps, so you can use it to help you with customizing your skin in unique ways. For example, if you only want a single cloud in all of godhome to have a unique sprite amongst all the clouds, you could identify its path with UE and assign it a new image with the names.json in your skin.
As long as the sprite does not have a Tk2d component, even if it normally shares a sprite image with another object, it can be uniquely swapped. Sprites that have a Tk2d component cannot be uniquely swapped.



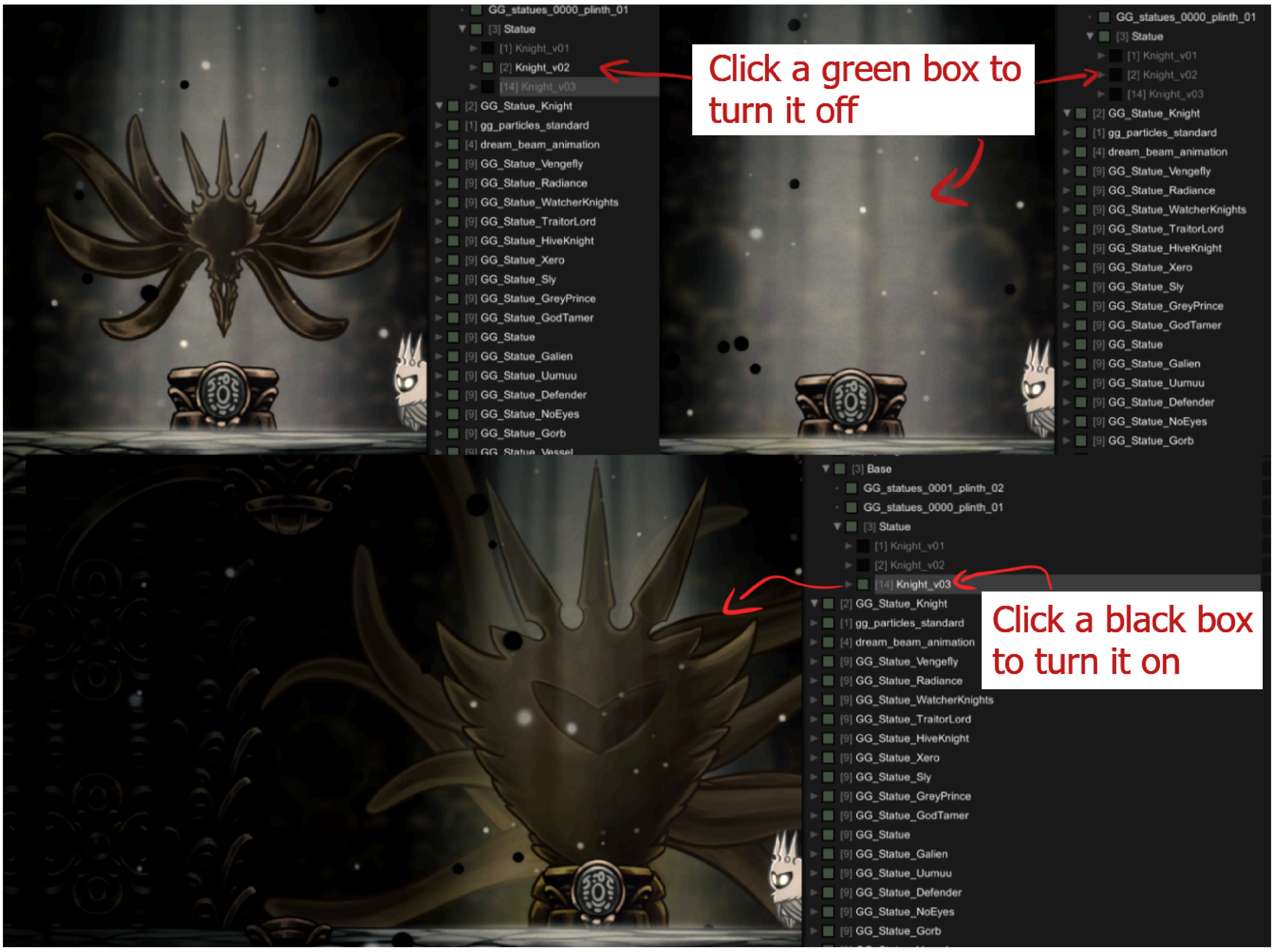
```
[G] Knight_v02
View Parent   GG_Statue_Knight/Base/Statue/
GameObject    Knight_v02
ActiveSelf    IsStatic    Instance ID: 221522    Tag: Untagged
Show in Explorer   Scene: GG_Workshop   Layer: Default   Flags: Nor
Position: 249.92 8.46 3.0708

Renderer Inspector Result
Vignette (Knight/Vignette)
white_light_donut (Knight/white_light_donut)
HeroLight (Knight/HeroLight)
GG_statue_beam_front (GG_Statue_Knight/Spotlight/Glow Response statue_beam/GG_statue_beam_front)
Knight_v02 (GG_Statue_Knight/Base/Statue/Knight_v02)
Fade Sprite (GG_Statue_Knight/Base/Statue/Knight_v02/Glow Response shade/Fade Sprite)
back_haze (GG_Statue_Knight/Spotlight/Glow Response statue_beam/back_haze)
Fade Sprite (GG_Statue_Knight/Spotlight/Glow Response statue_beam/Fade Sprite)
gg_ascent_0017_Layer-7-copy-7 (3) (gg_ascent_0017_Layer-7-copy-7 (3))
gg_ascent_backing_0001_1 (18) (gg_ascent_backing_0001_1 (18))
GG_scene_arena_extra_0003_4 (10) (back_arch_set/GameObject (2)/GG_scene_arena_extra_0003_4 (10))
```

names.json - Notepad

```
    "GG_statues_0006_5 (1)"
  ],
  "knight statue/knight v1": [
    "GG_Statue_Knight/Base/Statue/Knight_v01"
  ],
  "knight statue/knight v2": [
    "GG_Statue_Knight/Base/Statue/Knight_v02"
  ],
  "knight statue/knight fade sprite": [
    "GG_Statue_Knight/Base/Statue/Knight_v02/Glow Response shade/Fade Sprite",
    "GG_Statue_Knight/Base/Statue/Knight_v02/Glow Response shade/Fade Sprite (2)"
  ],
  "knight statue/blank": [
    "GG_Statue_Knight/Base/Statue/Knight_v03/gg_knight_shade_statue_0002_2 (1)",
    "GG_Statue_Knight/Base/Statue/Knight_v02/Glow Response shade/Fade Sprite (3)"
  ],
  "knight statue/knight v3 body": [
    "GG_Statue_Knight/Base/Statue/Knight_v03/gg_knight_shade_statue_0004_5"
  ],
  "knight statue/knight v3 head": [
    "GG_Statue_Knight/Base/Statue/Knight_v03/gg_knight_shade_statue_0004_5 (1)"
  ],
  "knight statue/knight v3 tendril 1": [
    "GG_Statue_Knight/Base/Statue/Knight_v03/gg_knight_shade_statue_0002_2",
    "GG_Statue_Knight/Base/Statue/Knight_v03/gg_knight_shade_statue_0002_2 (4)",
    "GG_Statue_Knight/Base/Statue/Knight_v03/gg_knight_shade_statue_0002_2 (5)",
    "GG_Statue_Knight/Base/Statue/Knight_v03/gg_knight_shade_statue_0002_2 (6)",
    "GG_Statue_Knight/Base/Statue/Knight_v03/gg_knight_shade_statue_0002_2 (7)"
  ],
  "knight statue/knight v3 tendril 2": [
    "GG_Statue_Knight/Base/Statue/Knight_v03/gg_knight_shade_statue_0002_2 (2)"
  ],
```

You can also click Object Explore and search for the sprite manually. This is also a great way to test sprites that might not be active in your current file without needing to leave the current game.



Creating Custom Cinematics

There are no sprite sheets for the cinematics. These are instead swapped by creating video files. Currently CK can swap the cinematics for the prologue, intro, stag, salubra's kiss, the nail arts, the nail smith, all five endings, mister mushroom ending, Lurien's telescope, the fountain, and the dreamer mask shattering. All of these can be downloaded here:
https://github.com/PrashantMohta/HollowKnight.CustomKnight/releases/download/v2.2.0/DefaultCinematics.zip
Do not change the names of these files. You can use them as a guide for what you need to name them for your skin.

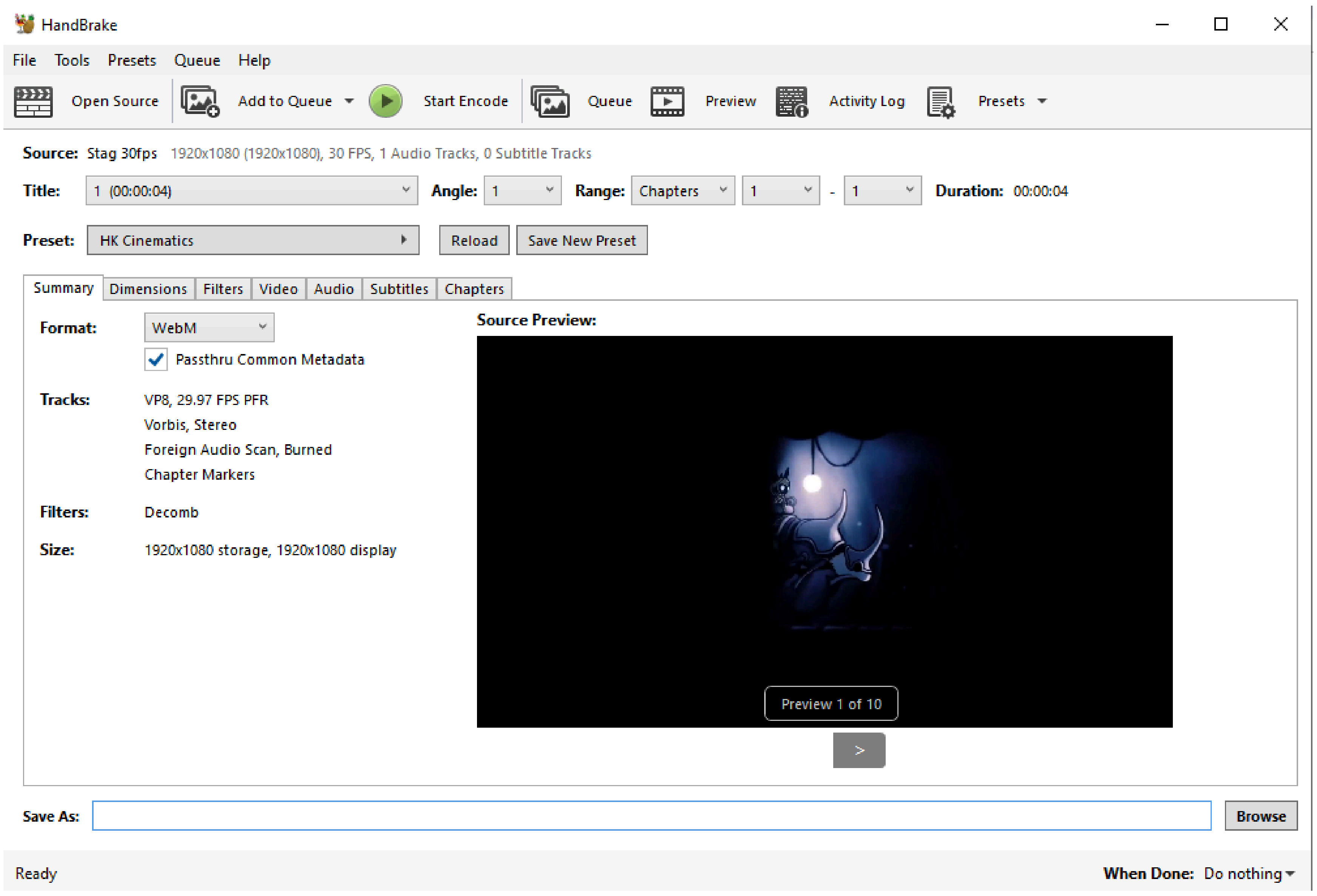
There is no easy way to make a custom cinematic, but there are some templates you can use to help you out. Most of these templates are for krita, and can be found in the art resources section of the HK Modding Discord. The templates I personally have made can be found here: HK Custom Cinematic Templates
These templates remove the knight from the cinematics for you, but you still have to draw your custom knight into every frame.

Once you've created your animation, you will need to save it as a WebM with the following settings:

Video: 1080p, 30fps, VP8 encoding
Audio: 48000hz sample rate, 2 channels, Vorbis encoding
Container: WebM

To format your video, you can download FFMPEG here:https://www.gyan.dev/ffmpeg/builds/#release-builds
Use the following command to format your video correctly:
ffmpeg -i <INFILE> -c:v libvpx -crf 10 -b:v 8M -c:a libvorbis -q 6 <OUTFILE>.webm

If you're like me and have no idea how to use FFMPEG and need something with a ui, download Handbrake here: https://handbrake.fr/
To format your video with handbrake, drag your video directly onto the program. It should default to the preset 'Fast 1080p30'. Click the dropdown menu under 'format' and select 'WebM'. Under the Dimension tab, set Cropping to None and Anamorphic Scaling to None. Under the Video tab, change the Video Encoder to VP8. Under Audio set Samplerate to 48. Then click 'Save New Preset' and name it 'HK Cinematics' so you never have to think about this again.



Source: Stag 30fps 1920x1080 (1920x1080), 30 FPS, 1 Audio Tracks, 0 Subtitle Tracks
Title: 1 (00:00:04) Angle: 1 Range: Chapters 1 - 1 Duration: 00:00:04
Preset: HK Cinematics
Format: WebM
✓ Passthru Common Metadata
Tracks: VP8, 29.97 FPS PFR
Vorbis, Stereo
Foreign Audio Scan, Burned
Chapter Markers
Filters: Decomb
Size: 1920x1080 storage, 1920x1080 display
Save As:
When Done: Do nothing

Once your preset is saved, choose a save location, then hit 'Start Encode'. Whenever you want to format your cinematics in the future, you can just drag them onto handbrake, select your 'HK Cinematic' preset, click start encode, and you're done.

To add the cinematics to your skin, place them directly into the 'Cinematics' folder inside of your skin. If it does not have one, make one. Make sure the files are named correctly. Use the default cinematics you downloaded earlier for reference if you need to.

CK Add-ons

These mods are add-ons to CK and require CK to work.

CustomKnightSuperAnimationAddon

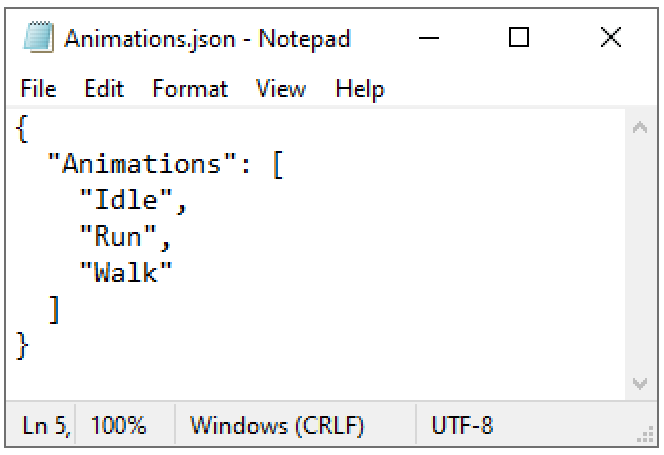
Adds support for skinning tk2d animations with new user created sprite sheets. Currently only supports the animations from Knight.png. With this add-on, you can alter the number of frames an animation has, making it longer or shorter, as well as change the canvas size limits so you can draw outside of that red box. Every individual animation that you want to alter (such as the idle animation or the airborne animation) requires its own individual sprite sheet, so it is advised to avoid doing this for every single knight animation, or you might end up with a very large skin that slows down your computer.

To see an example of a CKSAA skin, click here:
https://github.com/RedFrog6002/CustomKnightSuperAnimationAddon/tree/main/SampleSkins/Red%20Lines%20v2
You can use one of the skins here as a guide to set up your own skin.

Inside of the skin folder, create a new folder with the name of the atlas the original animation comes from (example:Knight).
Copy the 'Animation.Json' from one of the example skins and put it here. Alternatively, you can create a text document yourself and name it Animation.Json, then add the following text to the file, but instead of 'Idle', put the name of whatever animation you are changing. (For a list of animation names, just look at the names of the animations in the knight sprites folder found at the beginning of this guide):

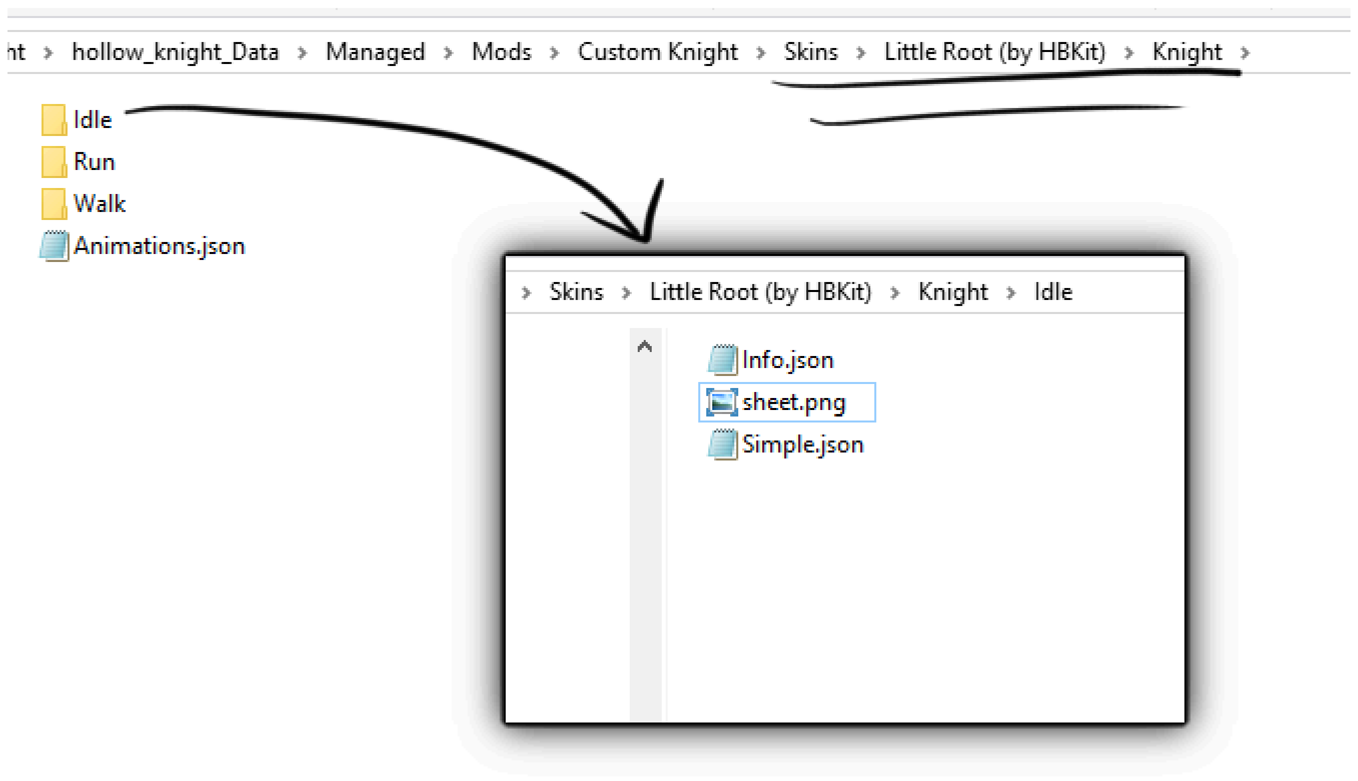
```
{
  "Animations": [
     "Idle"
  ]
}
```

If you are customizing more than one animation with a new sprite sheet, add a comma after the first animation, and add the next animation on the next line. Each one should be on its own line, with a comma after all but the last. Ex:


Animations.json - Notepad
File Edit Format View Help
Ln 5, 100% Windows (CRLF) UTF-8

```
{
  "Animations": [
    "Idle",
    "Run",
    "Walk"
  ]
}
```

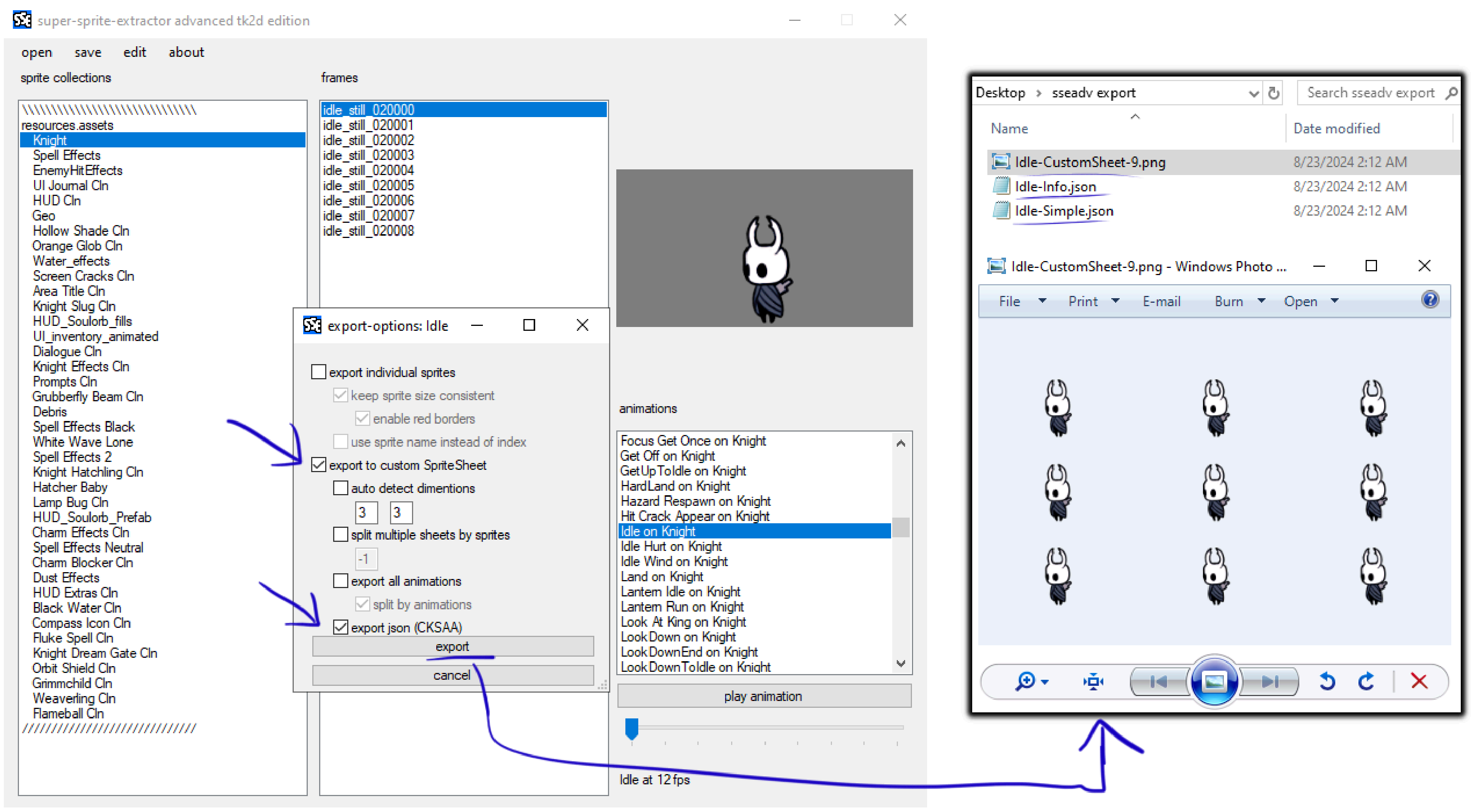
Next create a folder for each animation you are replacing (example: Inside of the Knight folder, you would make an Idle folder). Within each animation folder, you will include three files:
sheet.png, Info.json, and Simple.json (or Advanced.json, but this guide will only be covering how to set up the simple.json file. For examples of an advanced.json file, please look at the example skins mentioned previously.)


ht › hollow_knight_Data › Managed › Mods › Custom Knight › Skins › Little Root (by HBKit) › Knight ›
Idle
Run
Walk
Animations.json
› Skins › Little Root (by HBKit) › Knight › Idle
Info.json
sheet.png
Simple.json

You can either create these files from scratch, or use sseadv to make them for you.
You can find sseadv here: https://github.com/RedFrog6002/sseadv

To use sseadv, make sure you are on the computer you have hollow knight installed on.
Run the program, click open->one file->automatically. A window will pop up. Scroll down to the bottom and select 'Resources'. (NOTE: If you have a non-default install location, you might get an error and will need to direct the program to 'resources.assets' located in your hollow knight install location manually) This might take a while, so don't worry if nothing shows up right away. Once it loads, you can select the atlas containing the animation you want to edit (in this case 'Knight'). On the far right, all the animations will be listed in alphabetical order. Find the animation you're editing and select it.

Once you've selected your animation, on the top left of the program, select save->this->animation. It will ask you to select a folder to export to. You can make a new one, or use a folder you've made previously. Once selected, a pop up will appear with exporting options. Make sure 'export to custom SpriteSheet' and 'export json (CKSAA)' are both selected. If you select 'auto detect dimensions' sseadv will try to export a sheet in a way that makes a square (example: If there are 9 sprites, it will do 3 rows of 3 sprites). If you prefer, you can specify how many rows and columns you want your spritesheet to be instead. You can also choose to export the individual frames. This can be useful if you want to work on the frames one at a time, rather than on the spritesheet itself, however you will need to assemble the spritesheet yourself if you do this. There are no packing programs for these sheets.

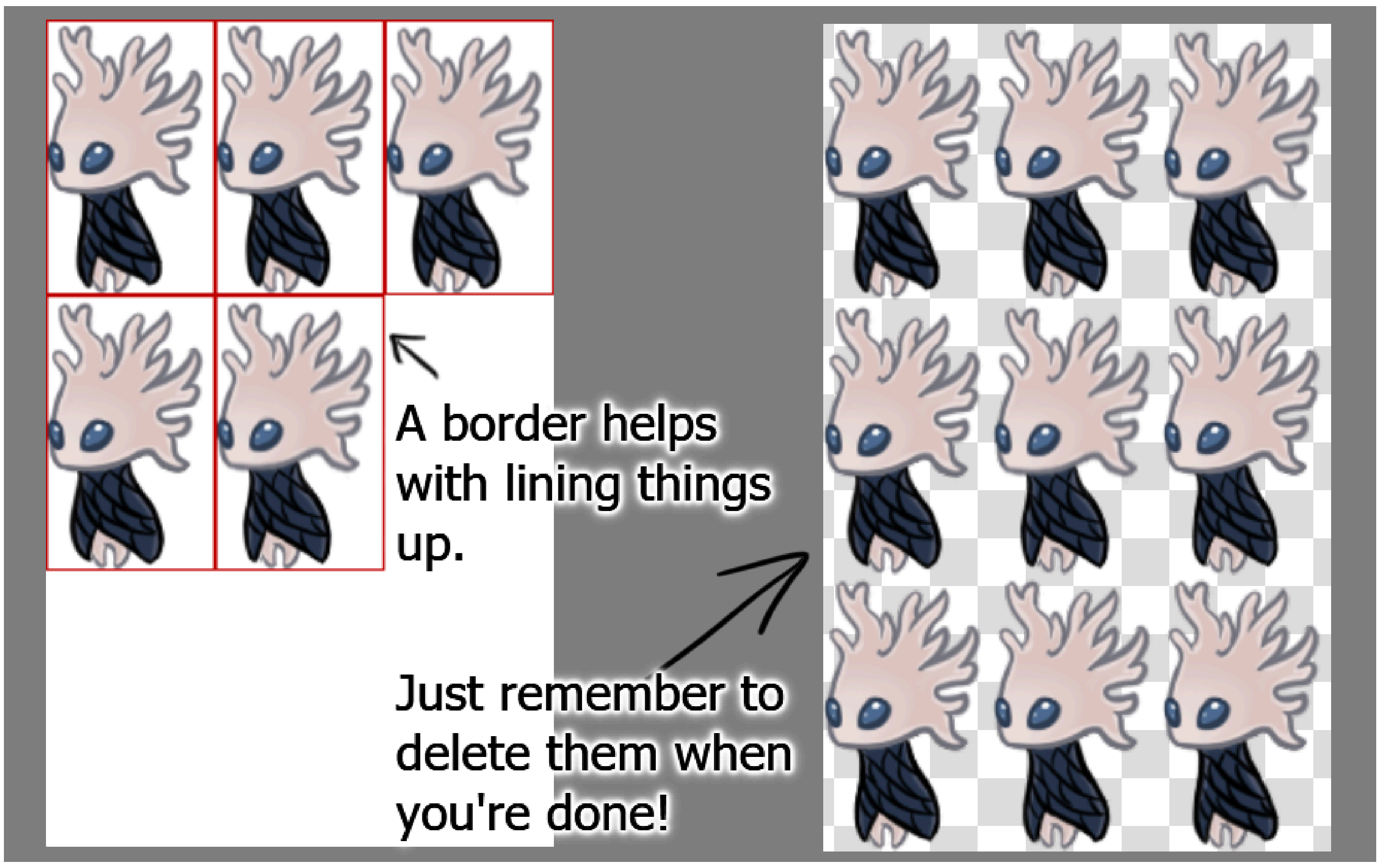


[ ] export individual sprites
[✓] keep sprite size consistent
[✓] enable red borders
[ ] use sprite name instead of index
[✓] export to custom SpriteSheet
[ ] auto detect dimentions
3 3
[ ] split multiple sheets by sprites
-1
[ ] export all animations
[✓] split by animations
[✓] export json (CKSAA)

Name	Date modified
Idle-CustomSheet-9.png	8/23/2024 2:12 AM
Idle-Info.json	8/23/2024 2:12 AM
Idle-Simple.json	8/23/2024 2:12 AM

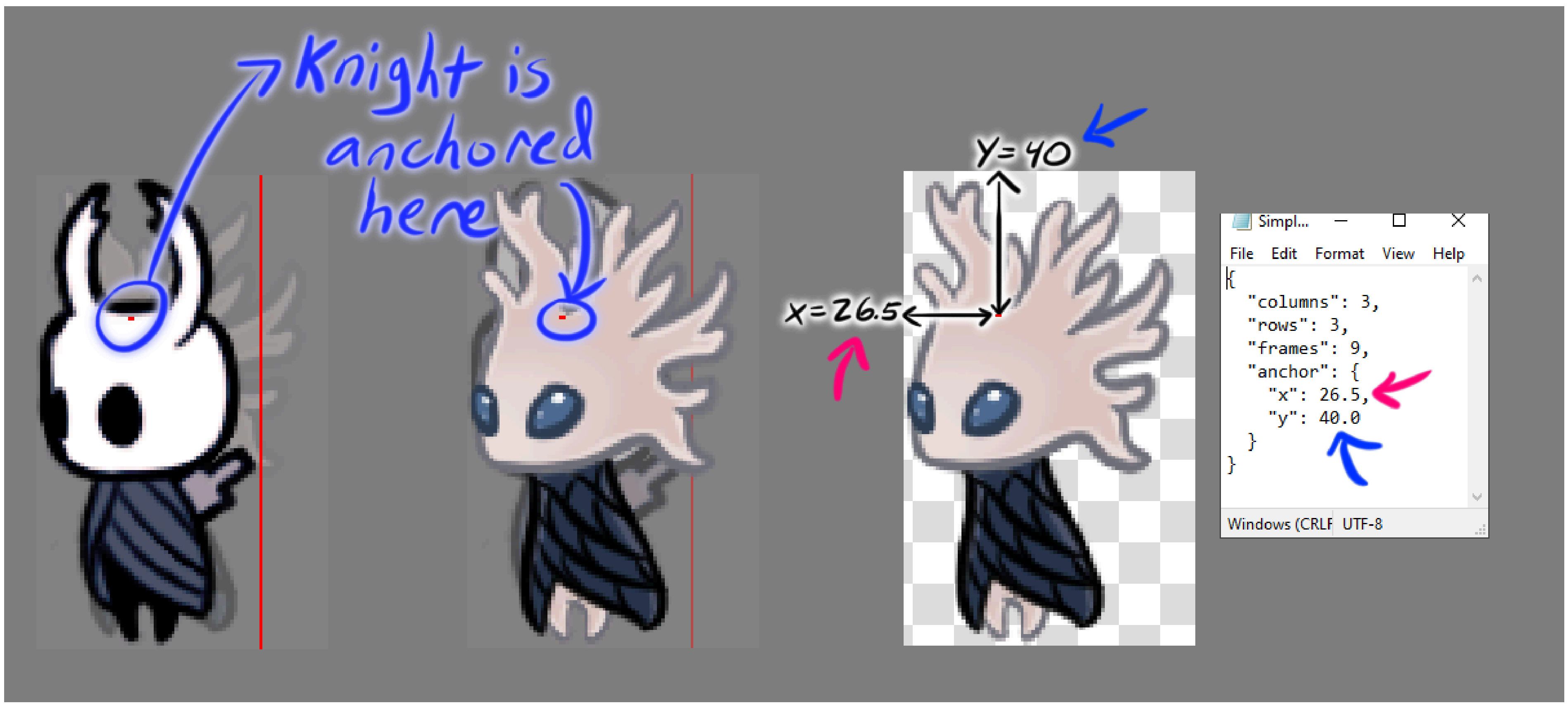
Make sure you rename the files to 'sheet.png', 'info.json', and 'simple.json' after you have moved these files into the proper folder within your skin. Now that you have a sprite sheet, you can edit it directly with your new sprites. If you find there still is not enough space in the exported spritesheet for your skin, then just copy the json files from the export and assemble the spritesheet yourself.

To make your own sprite sheet, make sure the canvas is the same size for all the individual sprites in your edited animation. If some frames don't use up all the space, that's fine. Put a single pixel thick border around the edge of the canvas. This will help you line your sprites up later. In a new file, copy and paste all of your sprites so that they are in order, starting from the top left. Make sure the borders you drew earlier are touching, but not overlapping. Trim the canvas of your sprite sheet so that there is no space outside of the borders of your sprites. Once all of your sprites are in the grid and the sprite sheet is trimmed, you can delete the borders.


A border helps with lining things up.
Just remember to delete them when you're done!

Once you have your custom sprite sheet, you need to make sure the simple.json matches it. The first 3 lines of the simple.json are pretty straight forward.
Columns: How many columns ↕ are on your sprite sheet?
Rows: How many rows ↔ are on your sprite sheet?
Frames: How many sprites are on your sprite sheet?

The anchor will determine where the sprite shows up on the screen. The default knight's anchor is in the middle of the top of their head. You can use that to help you figure out where to put the anchor on your own sprites. Use the info from sseadv to help you figure out where the anchor is on the default knight by checking the simple.json file that dumps with the animation and marking it yourself on the default sprite sheet that generates with it.
For your X coordinate, count how many pixels from the left of your sprite you want your anchor to be, and for your Y coordinate, count how many pixels from the top of your sprite.



```
{
  "columns": 3,
  "rows": 3,
  "frames": 9,
  "anchor": {
    "x": 26.5,
    "y": 40.0
  }
}
```

Incorrect anchor coordinates will make your sprites clip into the ground, float, or just not line up with the knight's hitbox. If you find your sprite isn't lined up correctly in game, try adjusting the anchor coordinates.

You only need to edit the Info.json if you are changing the speed or looping style of the animation. If you're not changing those things, using the info.json generated by sseadv is all you need. But if you do want to mess with those things, this is where you would edit that info.

An example of an info.json is as follows:

```
{
  "useOriginalData": true,
  "LoopType": "Loop",
  "loopStart": 0,
  "fps": 12.0,
  "InfoType": "Simple"
}
```

'useOriginalData'if set to true, the animation will use the same fps and loop information as the default knight. If set to false, it will use your specified loop and fps information

'LoopType' changes how the animation will play. The following are loop types you can use.

Loop - Loop indefinitely
LoopSection - Starts from the beginning, then loops a section defined by "loopStart" indefinitely
Once - Plays the clip once and stops at the last frame
PingPong - Plays the clip once forward, and then once in reverse, repeating indefinitely
RandomFrame - Chooses a random frame and stops
RandomLoop - Starts at a random frame and loops indefinitely from there.
Single - Switches to the first sprite and stops.

'loopStart' is what frame the animation will start at when it begins looping if loop type is set to 'LoopSection'. NOTE: the first frame of an animation is 0 here, so if you have a 3 sprite animation, then those sprites are 0, 1, and 2, not 1, 2, and 3.

'fps' is 'frames per second' and controls how quickly the animation will play

'InfoType' is what .json file type is being used (Simple or Advanced) for this animation.

NOTE: **Super animations will not be affected by hitting 'fix skins'. You must switch skins if you want to update cksaa while still in game.**

NOTE: **Changing the canvas size will not change the knight's hitbox.**

Asymmetric Knight

Adds support for asymmetry for the Knight, Unn, and Sprint atlases, so your knight will look different depending on if they are facing left or right. Add a Knight_left.png, Unn_left.png, and Sprint_left.png for the left facing sprites. Knight.png, Unn.png, and Sprint.png will be the right facing sprites.

CustomKnightPlus

Adds support for asymmetry as well as map zone specific sprite sheets. Asymmetry is set up the same as Asymmetric knight, with the regular names of the sheets being right facing and <name>-left.png being left facing. CustomKnightPlus however can support more atlases than Asymmetric Knight.

For Zone specific sprite sheets, add -<map zone> to the sprite sheet. For example, 'knight-deepnest.png' would use that knight sheet exclusively in deepnest. The list of locations compatible with this mod can be found in the Read Me that comes with the mod.

You can combine zone and asymmetry as well. Put direction, then zone. Example: knight-left-deepnest.png' will only use that sheet while in deepnest and facing left.

NOTE: **not compatible with Asymmetric Knight**. If you have CustomKnightPlus, you should not need Asymmetric Knight anyway.

CustomJournal

Allows customizing the Hunter's Journal. Includes a way to dump the journal under Custom Journal in the mod menu. Add a 'Journal' folder to your skin's 'Swap' folder and treat it like a Path Swap.

Mods with CK Integration

These are standalone mods that have CK integration built in.

press g to dab

This mod adds a unique animation that plays when you hit g. This animation can be customized by adding the atlas 'dab.png' to your skin. A fun way to add a special animation to your skin! (It doesn't need to be a dab).

The dab sprites and atlas can be found here and are compatible with all 3 packers: (Requires being part of the HK Modding Discord): https://discord.com/channels/879125729936298015/880166423635308655/1230175945386889277

Enemy HP Bar

This mod adds a visible hp bar to enemies and bosses. Include an 'HPbar' folder inside of your skin, and this mod will use your hp bar skin instead. You can use the default HP bar that comes with the mod as your base to create your own. You are not limited to the canvas size and can increase or decrease the canvas. Check how it looks in game after every alteration of the canvas, to make sure the bars still line up correctly with each other. Make sure the names match the default hp bar.

Note: When you swap skins, the HP Bar skin for bosses will not update until you leave the current room.
After booting up the game, you must swap skins at least once before CK integration will start working.

CustomAudio

This mod allows you to customize the sound effects that play within the game. Include an 'Audio' folder within your skin's swap folder, and this mod will use the audio files from your skin instead. You can download the audio files here: Hollow Knight audio files 1.4.3.2
Make sure the replacement audio files match the name of the original file it is replacing.

NOTE: **CK Integration is currently bugged and does not work at this time**