



Guide Version 1 - 9.19.2022

Thank you for purchasing this *Time Elements* asset pack -- with this set, you can create more characters than you will need to fill an entire game world! This guide will explain how to make the best use of the assets and tools in this collection.

Included in this Pack

- *Time Elements* Character Piece Assets
- *Time Elements* Premade Full Character Sets
- *Time Elements* Emote-Popup Animation Sheets
- *Elements Character Generator* Program
- *ElementsFix* Plugin for RPG Maker MV/MZ
- Color Palette Files (.pal and .png)
- This Guide

Table of Contents

- Piece Assets and Layering
- Sprite Sheet Organization
- Animations

- Generator
- Colors
- Premade Character Sprites
- Emote-Popup Animations
- Using the ElementsFix Plugin (RPG Maker)
- Frequently Asked Questions (FAQ)
- Credits



Piece Assets and Layering

This set is designed around individual pieces that can be used to construct unique new characters. The primary intended use of this set is to create your own character sprites by assembling the included pieces.

You can create characters by copying the assets into image-editing software (such as *Photoshop* or *Aseprite*), by using the included generator program, or with a combination of those approaches.



Each piece is designed to work as a specific body-part type. Generally they would be layered in a specific order.

***Note:** Depending on the direction, the layer order might change. For example, when the character is facing north, the “back hair” should be layered above the head instead of behind it.*

The types are:

- **Weapon**
 - For weapons, tools, etc. Could potentially be used for other accessories or held items. Not all assets here would be required for every animation.
 - Some weapon assets might have unique animation uses. For more details, see the “animations” section of this guide below.
- **Hat**
 - Hats and helmets generally are layered above everything else (except weapons) regardless of facing direction.
- **Front Extra**
 - For beards, masks, or other accessories that would appear on the front of a character. This core set also includes some animal-ears on this layer.
- **Hair**
 - Main hairstyle. Called the “front” hair to make it distinct from the “back hair”, but would generally layer above the head in any facing direction. Generally these “front” hair pieces work as full standalone hairstyles, and the “back hair” can be used to extend any of them.
- **Head**
 - Head with face, mostly for different faces or types of eyes. Can potentially have variations for things like elf ears.
- **Top**
 - Torso: shirt and arms. Generally this would include any belts at the

waist, because it layers above the pants. Because of the arm movement in some animations, this is the piece that could cause the most potential layering complications (*see below*).

- **Bottom**
 - Pants and legs. Includes the feet.
- **Back Extra**
 - Layered behind the character in all directions except north-facing, where it would be above most other pieces. Used for things like wings, backpacks, or tails.
- **Back Hair**
 - Used for long hair. Could be an extension that blends into the normal hair, but also something like a ponytail, braid, etc. In many cases, it won't be visible in the south-facing direction.
- **Shadow**
 - A simple black circle that helps ground the sprite. Optional depending on your style preference. Always goes at the very bottom layer regardless of facing direction.
 - The shadow can potentially be exported separately as a single sprite, and layered uniquely in your game engine (maybe with some transparency). This usage would be particularly useful if your character jumps, and the shadow as a separate sprite can stay in place at the player's grounded location.

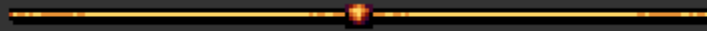
Unlike other sprite generators, which use a “paper doll” model that simply layers clothing on top of a nude base, the *Time Elements* pieces are designed to completely assemble a character. This creates some extra complexity, but without the restriction of a single base, we have a lot more freedom for possibilities for character variations.

Note: *There are a lot of moving parts (literally), and it makes the layering very difficult. For example, any animation that moves the arms in front of or behind*

*the head creates inconsistencies. Some things will even be impossible. **Inevitably, not everything is going to be perfect.***

If you're planning on using these pieces to assemble full characters using something like Photoshop, then it will be easier for you to modify and edit individual layering problems as necessary (this is how the pieces were originally designed to be used).

If you are planning on using these pieces in a game engine of your own, note that there will be complexities and edge cases: layers might need to order themselves differently depending on the animation-- I've tried to solve this in an efficient way, and unfortunately it's not realistic: it's impossible to systematically solve every possible edge case with an asset system of this size.



Sprite Sheet Organization

Full Sprite Sheet (General Use)

The default format for the full sprite sheet includes all of the animations for a single character. This is a general use sheet intended to be an easy way to organize a full character's animation frames. It is not designed for any specific single engine.

The individual piece-assets are all arranged in this format, and the generator will expect this layout by default.

Each frame is 48x48 pixels (3x3 tiles, based on a normal 16x16 grid). The character is "grounded" on the middle tile, with extra space to account for various animations in four directions.



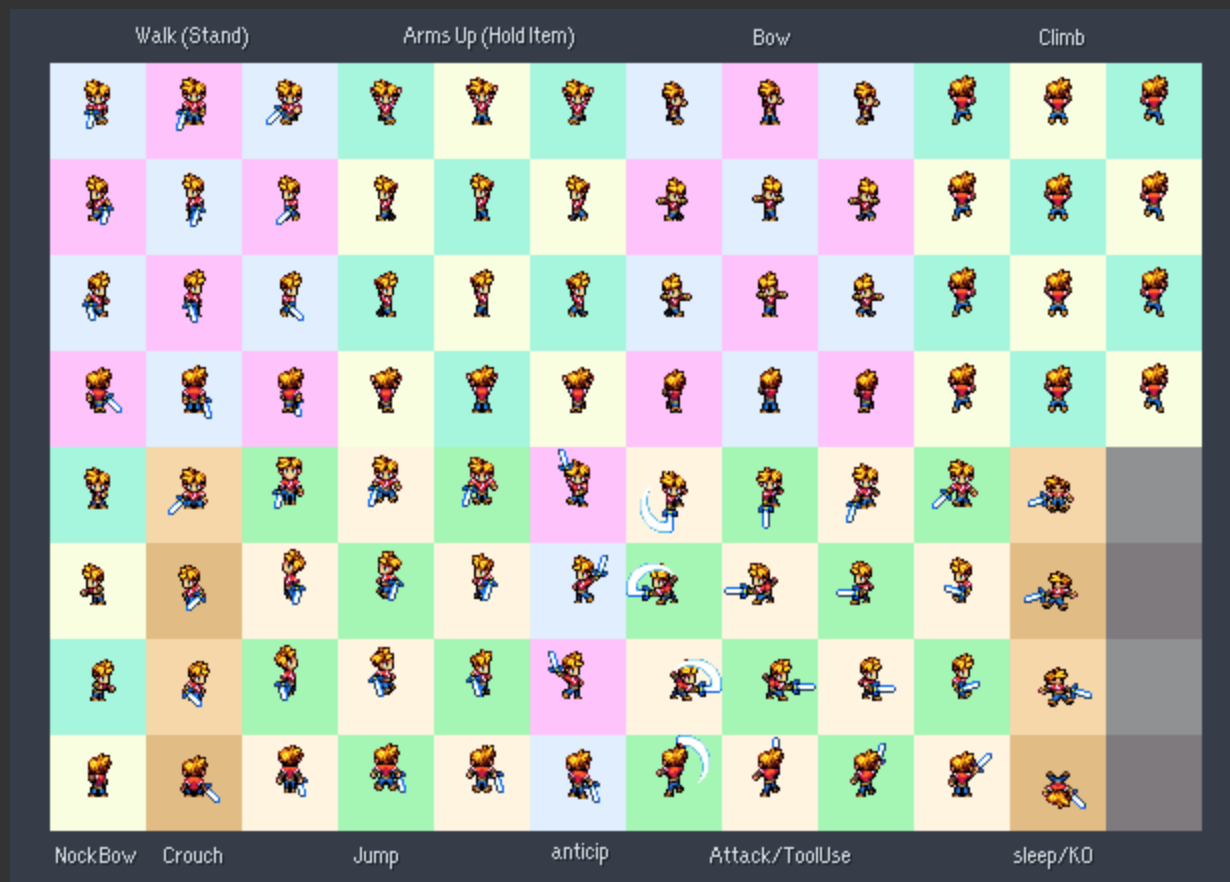
(Note: A full-sized version of this reference image is included as a separate PNG file in the “guide” folder.)

See the next section for a detailed explanation of each animation.

RPG Maker Sprite Sheets

RPG Maker sprite sheets use a specific layout. This pack includes some pre-made characters in a format for use in *RPG Maker*, and the included generator is able to export sheets formatted for *RPG Maker* (the formats included are compatible with *RPG Maker VX/Ace* and *RPG Maker MV/MZ*).

Like the sheet above, each frame here is 48x48 pixels, representing a 3x3 tile-grid, with the character “grounded” in the center tile. The only difference is in the arrangement of the frames. This is the animation layout that the generator exports for *RPG Maker*:



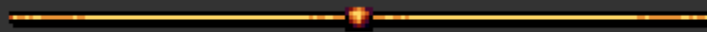
(Note: A full-sized version of this reference image is included as a separate PNG file in the "guide" folder.)

RPG Maker's sprite sheet layout is normally divided into eight characters that have four-directional walking sprites. The top row of this sheet will conform to that standard, making use of the animations that have a standard walk cycle.

The bottom is more complex: the animations that do not conform to three-frames are on the bottom half of the sheet. These animations will require additional work to display properly in *RPG Maker* (either via event commands or with a plugin).

Important: By default, the character sprites will *not* align correctly with *RPG Maker's* in-game grid. This set comes with a custom plugin that will fix this (for more details see the below section on the [ElementsFix Plugin](#)).

Additionally, the generator can export files in the Side-View (SV) battle layout used by *RPG Maker MV* and *MZ*. Please note that the original animations are designed for four-directional action-RPG gameplay, so not every SV animation is represented perfectly. I've arranged the SV layout to best map the existing animations to the actions expected by the *RPG Maker* system. If you plan to use these SV sheets in your game, additional customized changes will likely be required depending on your needs.



Animations

Some animations are more complex than others. Here are some notes for the best implementation of the sprite animations:

Walk Cycle

The three-frame walk animation makes use of the middle frame twice, in a “ping pong” cycle. It would be animated in the frame order of 1-2-3-2-1-2... When the character isn't moving, the middle frame can be used as a “stand” pose.

Other animations that use this same “walk cycle” are [Arms Up](#), [Bow](#), and [Climb](#).

Climb

All directions face north because the climb is intended for use on ladders, vines, cliff faces, etc, where the player sprite would be facing towards the wall.

In this core release with basic assets, the same animation is simply repeated for all four directions. The reason that the redundant animations are included is to allow for user edits or future assets that may have details that animate differently depending on the climbing direction.

Crouch and Jump

The single-frame crouch animation has a wide variety of uses on its own.

When combined with the jump, the crouch provides a useful “anticipation” frame that leads into the jump. Depending on the situation, you may not find it necessary.

Windup (anticipation)

The anticipation frame (wind up) is a single frame that is designed to enhance the regular attack/tool-use animation. For weapon attacks, this frame would be optional depending on how “heavy” you want the attack to feel. Additionally, this frame would be important for other actions, for example: casting a fishing rod.

Attack and Tool-Use

A four-frame sequence (five if you include the anticipation frame)

designed to work with weapons or tools on another layer or as separate sprite overlays.

This animation was designed with versatility in mind, so that different weapon or tool sprites can provide additional context to the movement (for example, the sword and the spear included in this core set each have a different “smear”, which implies unique motions even though they use this same animation).

Empty-handed, this action could potentially be used as a punch or as a generic “interact” animation (for example, pressing a switch or lever).

Nock and Bow

The “NockBow” animation is a single frame designed to lead into the main bow animation.

The bow animation itself is not actually a full bow firing animation on the character side. It's a single frame of the character's upper body in the bow position, and the other frames allowing for a walking animation while holding the bow.

This animation setup gives the user flexibility for the character to walk or strafe while having an arrow ready. If you don't want that movement (maybe you're using these for turn-based actions, for example)-- then it's perfectly fine to just use the single center standing frame from this animation. The reason this works without arm movement from the character is because:

The **bow and arrow overlay layers** do the heavy lifting for the

animation. Note that the arrow assets for this animation do *not* follow the normal “walk cycle” rules. They are intended to overlay independent from the player’s walk cycle and depend on the needs for firing the arrow: The 'nock' is a single frame that transitions into the other three frames: main, pullback, and release.

***Note:** This set includes a single-frame PNG image for use as an arrow projectile object. It can be found in the “extras” folder.*

The pullback frame could potentially be held as long as the user wants, and with this setup, the arrow could be held back while the player walks around.

For an animated reference, see the included `bowarrow_example.gif` in the guide folder.

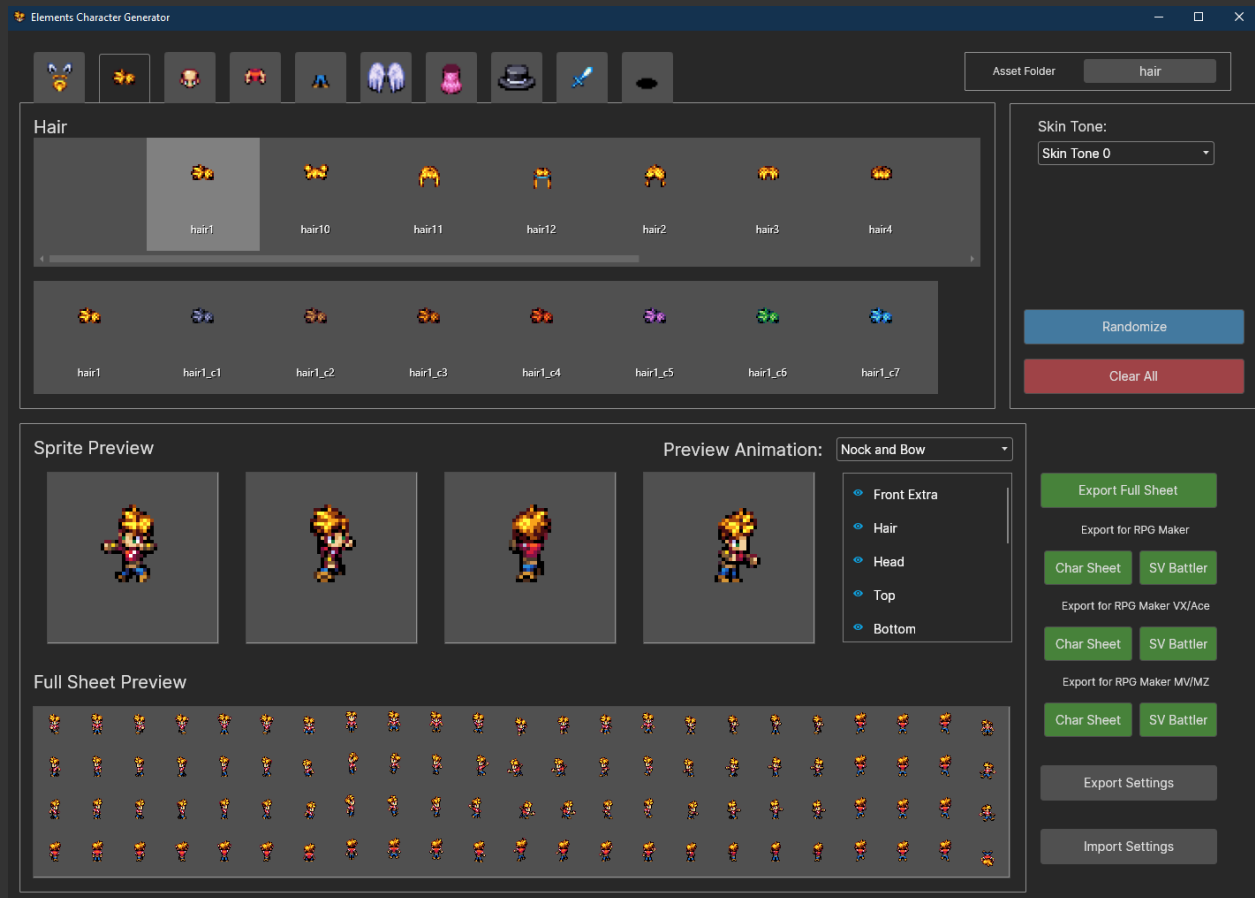
Sleep/KO

Separate single frames that do not animate. The four frames are for variety; they don’t necessarily correspond with the directions.



Generator

This pack includes a custom sprite generator program. It is designed to work specifically with these *Time Elements* assets.



Getting Started

Open the generator software from the `Elements Character Generator.exe`.

When you first open up the generator, you will notice that it is empty (*this is because I didn't want to duplicate all of the asset files, which would unnecessarily increase the size of the download*). The assets will need to be imported. There are two ways to do this:

First, you can copy and paste all of the parts from the "assets" folder into the folder with the generator itself. You will find that the generator includes empty folders for the different asset layers. You can copy the

PNG files directly into those folders, or you can copy all of the folders at once and overwrite the empty ones.

Alternatively, you can tell the generator to look into the assets folders directly, by changing the “asset folder” with the button in the top right. This would need to be done for each asset type.

Using the Generator

For the most part the generator should be easy to use. It is intended to be simple and very straightforward. It will display the assets that it finds in the folder and you can select one for each layer, with color variations for most.

***Note:** Some pieces might not show up normally in the selection area and display a blank space (specifically backhair5, some of the weapon/tools, or other future additions). This is because they aren't visible from the south-facing idle pose, so the program doesn't generate a preview icon. You can select them as normal to see the preview on the sprite itself.*

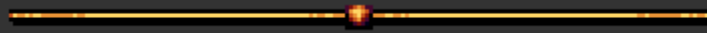
There are multiple skin tone options programmed into the generator by default. For additional details about the palettes and how the color replacement works, see the “*skin tones*” section further in this guide.

The generator can export character sheets in the formats mentioned above (see *Sprite Sheet Organization*). The first “for RPG Maker” option will export in the MV/MZ format at a 100% size, intended primarily for providing a working version for users who will modify the sprites before enlarging them for use in the engine.

The options to export/import settings are for modifying the JSON data

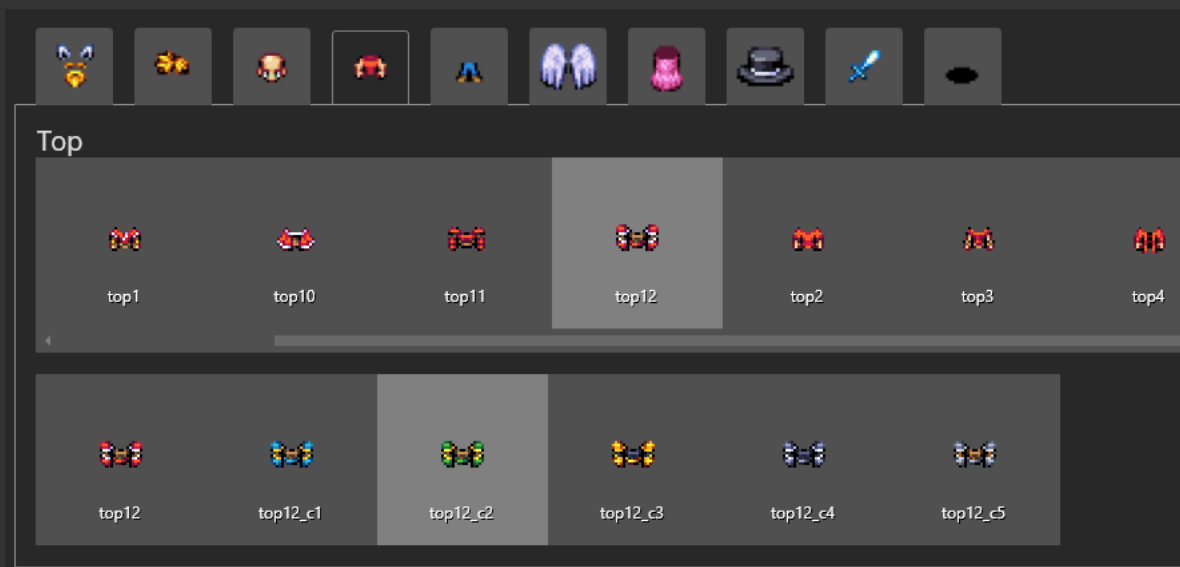
that determines things like layer order, skin tone mapping, etc. Not recommended except for advanced users.

Lastly, please respect that the generator was created (by a friend of mine) as a bonus inclusion to this pack; the Elements assets were originally designed for the user to put together in an image-editing software or in the game engine. Future support for the generator itself is unlikely.



Colors

When using the generator, you'll notice that the pieces each come in specific color variations. There are no color sliders in the generator because predefined palettes make more sense with the retro aesthetic of the *Elements* style.



Of course -- if you want additional colors, you are welcome to copy the images in the parts folders and use any graphics editing software to re-color them however works best for you.

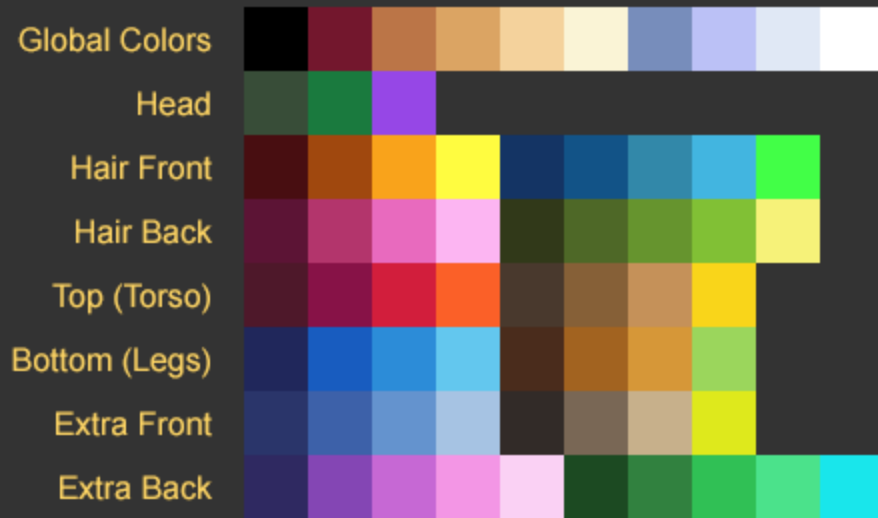
The generator will recognize alternate color variations based on filename: a “_cX” suffix, where the *X* is a number. The generator will display the color variations in the order of *X*.

For example: `hair1.png` would be a primary asset shown in the generator, and the files named `hair1_c1.png`, `hair1_c2.png`, etc, would be displayed as variations underneath it.

“Original” Color Palettes

Each asset is originally created using a color palette unique to its layer type. For example, all “top” assets use colors that are not found in any other asset type. In this way, key colors are not repeated on separate layers, allowing for easy recoloring across an entire sprite.

In addition to these layer-specific color palettes, the overall palette includes some “global” colors that are used across all layers: these are colors that are generally not intended to be recolored: solid black and a group of colors used for white across all assets. The global set also includes a palette for the skin (see next section).



This color palette is named `original`, included in the `palettes` folder.

I've included the raw assets with these layer-specific color palettes. You may find them useful if you are going to create your own edits or variants. They can be found in the folder: "`extras>originals`".

***Note:** These colors will NOT look good on a final sprite! Normally, for "good" pixel art, you want a unified palette, with a low color count that maximizes contrast. Instead, these are designed so that colors will not repeat on different layers of clothing, so it can be easily recolored.*

The colors of the final asset pieces in the pack are different from these working 'original palettes' (see "Final Color Palette" below).

Note that for the final exported sprites, the *Elements* style does NOT have a specific color-count limitation -- but in order to keep the style consistent, I recommend using colors that already exist within the finished pieces and avoid adding more colors unless they are necessary.

Skin Tones

Color palettes for skin tones are handled differently. This is for a technical reason: the palette for the skin is the only palette that is used consistently across all the layers.

Because the skin tone is potentially used on every asset, it's unrealistic to have image files for every possibility. The solution for different skin tones is to re-color the entire final sprite, which is why the skin tone uses colors unique in the global palette.

In the generator, you can change your character's skin tone using the selector in the top right. This works by recoloring the sprite as a whole, so the changes are applied across all pieces on all layers.

Note: It's possible to edit or create additional skin tones by editing the color data in the JSON settings file. I might also provide additional defaults in future updates if there is high demand.

If you are creating sprites in an image-editing software (or if you manually edit the sprites created by the generator), you might want to recolor the skin tones yourself. In order to make this easy, I've provided the pre-made color palettes for additional skin tones.

The colors below are the exact colors that come pre-programmed into the generator. The skin-tone selector finds the colors from the "default" skin palette and replaces them with the color palettes provided in the JSON settings. For this reason, any new pieces with skin should use the exact colors from Skin Tone 0.



#73172d
#bb7547
#dba463
#f4d29c
#faf4d6



#561f2d
#9d5534
#b4723c
#d49149
#f0c175



#481c0e
#774128
#955123
#b97e50
#dbaa76



#36150c
#583322
#7b4c2d
#986743
#c78e52

Skin Tone 0 (Default):

Hex	RGB	HSL
73172d	115,23,45	345,67,27
bb7547	187,117,71	24,46,51
dba463	219,164,99	32,62,62

f4d29c	244,210,156	37,80,78
faf4d6	250,244,214	50,78,91

Skin Tone 1:

Hex	RGB	HSL
561f2d	86,31,45	344,47,23
9d5534	157,85,52	19,50,41
b4723c	180,114,60	27,50,47
d49149	212, 145, 73	31,62,56
f0c175	240, 193, 117	37,80,70

Skin Tone 2:

Hex	RGB	HSL
481c0e	72, 28, 14	14,67,17
774128	119, 65, 40	19,50,31
955123	149, 81, 35	24,62,36
b97e50	185, 126, 80	26,43,52
dbaa76	219, 170, 118	31,58,66

Skin Tone 3:

Hex	RGB	HSL
36150c	54, 21, 12	13,64,13
583322	88, 51, 34	19,44,24
7b4c2d	123, 76, 45	24,46,33

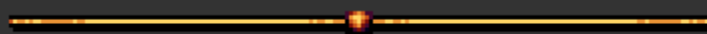
986743	152, 103, 67	25,39,43
c78e52	199, 142, 82	31,51,55

Final Color Palette

Despite the color specifics used in the advanced processes detailed above, **the final assets in this style do *not* need to conform to specific palette restrictions.**

A loose and flexible color palette will make it easier for users to create their own graphics in the style of *Time Elements*. For all the talk about it being “16bit” or “SNES”, the style does not adhere to rigid color limitations.

I've included a file for the overall color palette used for the final *Elements* assets (“general.pal” in the “extras>palettes” folder). *Note that this palette is significantly less deliberate and less organized than the other included palettes-- that's because it's pulled from the final assets rather than created before them.* If you are going to create additional edits or assets, the included palette can provide a useful starting point.



Premade Character Sprites

This collection includes a number of premade full character sprites. All of these premade sprites are in *RPG Maker* formats and are intended to kickstart your project. You can find them in the “premade” folder.

The premade characters come in two versions: with and without the shadow. You can decide which style best fits the look of your game.

Full Characters

There are five premade character sprites that have the full set of animations. They were all created using the *Elements* assets in the generator. The sheets are arranged in the *RPG Maker* format detailed above.



Note: These full character sheets have a “)” at the beginning of the filename. This is for use in *RPG Maker* with the alignment plugin (see the “Using the *ElementsFix Plugin*” section below).

I made some additional changes to these final character sheets after creating the sprites in the generator: first, I exported multiple versions with different weapons and compiled the separate weapon animations together onto a single sheet. The final sheets display the bow/nock animations and a sword attack. By default this isn't possible with generator exports (the character would need to be exported multiple times, once with each weapon-- or empty-handed to use the weapons individually as overlays).

I also manually edited the “arms up” pose for these sprites. Because of the layer ordering, by default the head and hair pieces are layered above the body and arms, which can appear strange with some combinations. For these premade characters, I edited the arms of that specific pose to appear above the head on the front and side views.

These small edits are examples of the complications mentioned above in the first section of this document. These premade characters can work as examples for when you come across situations where you might need to edit your own characters.

NPCs

I’ve created eight NPC sprites to fill out your game world. They only use the classic four-directional walk animations, and are all arranged on a single sprite sheet. A second sheet is available with alternate skin color variations, totaling sixteen premade NPC sprites.

These sprites were created using the existing pieces in the generator--so they also serve as a preview of what is possible. If you want a version of these characters with the full animations, see if you can recreate these designs in the generator!

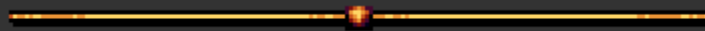
Children

The premade child sprites are unique because they are an **original asset** created to work with the *Elements* style.



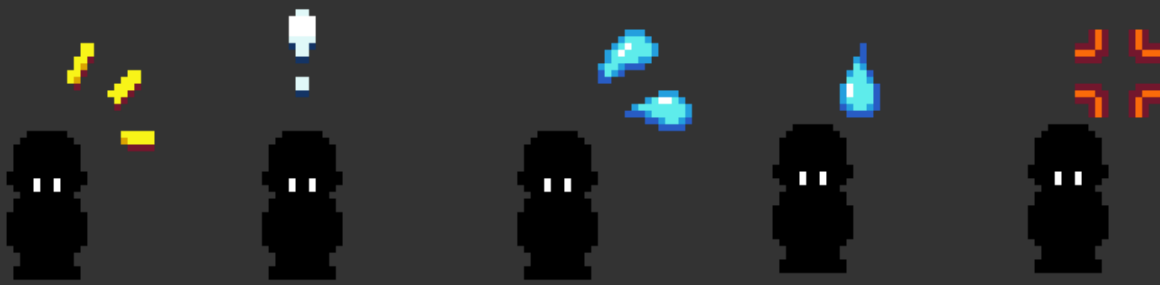
It's not possible to use generator parts for children, because the proportions are different. Therefore, I created the child sprites as unique standalone sprites, with only the basic 4-directional walking animations. The intended use for these children is as NPCs.

The sheets feature four children, two boys and two girls. Each child has a second recolor, and then both full sheets have an additional recolored sheet with alternate skin tone options. This provides a set of sixteen total child NPC sprites.



Emote-Popup Animations

I have created some custom emotion pop-ups that are designed to work with *Time Elements* character sprites. These animations are included in this pack as a special bonus to help provide more personality to your game characters!



You can find the emotion animations in the "extras>emote" folder:

The "sheets" subfolder contains individual horizontal sprite sheets for the eight unique emotion animations. These are for general use.

If you are using *RPG Maker MV/MZ*, all of the emotion animations are on a single sheet (*emote-animations-MV.png*). It's intended to be used as a *battle animation*, which you can create in the database and then play on the map using event commands.

Note: *The emotion sheet is NOT a RPG Maker 'balloon' resource, because those have some predefined limitations to them. As a 'battle animation', users have more control over the animation and its timing.*

Lastly, one of the images included is a reference/preview sheet, which includes all the animation frames of the emotes alongside character shadows to demonstrate the intended alignment.



Using the ElementsFix Plugin (RPG Maker)

By default, the *Elements* character sprites will not align perfectly to the grid in *RPG Maker*. Specifically; they will appear one tile above where they should.

This is because the *Elements* characters have an extra “tile” of space built around the sprite, which is used for some animations (such as weapon attacks).

This pack includes a plugin that fixes this problem. Once activated in your project, it will fix the alignment for all character sprites that have a closing parenthesis (`)` at the beginning of the file's name. You can find the plugin in the folder: `extras>rmfix_plugin`.

***Note:** To use the plugin in RPG Maker, first copy the .js file into your game's plugins folder. Then, in RPG Maker, activate the plugin by double-clicking the open space on the “Plugins Manager” popup window and finding it in the drop-down box.*

This plugin was created by [Hikitsune-Red](#), and I have permission to include it as a bonus in this asset pack. It should work with both *RPG Maker MV* and *MZ*.



Frequently Asked Questions

· *Are these characters compatible with your Time Fantasy assets?*

Yes and no. Both styles use the same 16x16 tilebase, so *Elements* sprites will generally work fine with *Time Fantasy* tilesets. The sizes are compatible.

However, *Time Fantasy* uses a unique desaturated color palette with a muted feel that will clash with characters in the *Elements* style. If you want to use both styles together, then it's recommended to re-color the *Time Fantasy* tilesets to have a greater range of colors. That way the *Elements* characters will not look out of place.

Alternatively, if you prefer the faded colors of the original *Time Fantasy* style, you can convert full-colored *Elements* characters to fit in with that color scheme: the biggest change will just be replacing the black color with the lighter "black" used by *TF* (RGB: 53/64/73). That alone should do most of it. If any other colors look out of place against the lighter black, then lower the saturation and brightness to match.

· *Will there be future updates or expansions with more pieces?*

Yes! In fact, I've already designed nearly twice as many pieces as the ones you can find in this set. But it takes a long time to animate them, so if we included them all, this release would have been delayed for another half a year!

I am planning some larger theme-based expansion sets, and there may be occasional small free updates in the future. I will also be sharing new pieces on *Patreon* as I fulfill requests from patrons.

You can be sure that I am looking forward to releasing more *Elements* content that you can use to create new characters.

· *Will there be new tilesets to match these character sprites?*

Yes! I plan to build up this style with a series of future asset packs and expansions, including tilesets for a variety of settings. Stay tuned for more content.

· *Will there be more animations added?*

Right now I *do not* have plans to add any additional animations to the *Elements* sheet. A lot of thought was put into this core set of animations, with the goal of settling on a set that is versatile without requiring an unrealistic amount of frames.

That doesn't mean that custom animations are impossible. I am a big fan of unique characters with custom animations-- so while the current set of animations is locked for the "generator pieces" assets, there is a good chance for new animated characters in future releases.

· *The characters are broken in RPG Maker!*

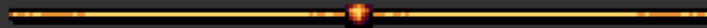
If a character is aligned one tile too high: please see the above section on the [ElementsFix Plugin](#) for *RPG Maker MV and MZ*. The plugin is included in this pack.

If a sprite appears glitchy: By default, *RPG Maker* assumes that sprite sheets have eight characters/animations on them. If your sheet contains only one character, add a "\$" to the beginning of the filename and the engine will recognize it as a single character.

If the sprites are too small or too big: You might be using the formats for the wrong version of *RPG Maker*. If you are using VX or VX Ace, then you want assets that are 200% size. If you are using MV or MZ, then you want 300% assets. The regular 100%-sized assets are best for use in non-*RPG Maker* engines, or for editing at the working size, and then upscaling to match your needs.

· *I found a mistake, error, or bug!*

Fair enough! In a complicated set like this, I probably missed something. Please get in touch via email on my website (finalbossblues.com), or send me a message on Twitter ([@finalbossblues](https://twitter.com/finalbossblues)). I appreciate the feedback!



Credits

Assets by **Jason Perry (finalbossblues)**

PDF Guide written by **Jason Perry (finalbossblues)**

Elements Sprite Generator by **Locke**

ElementsFix plugin by **Hikitsune-Red**

Export scripts by **fmoo**

Special thanks to *You*