

. Victor Scripts .

Basic Settings

Sufix and Intro Fade

```
199 #-----
200 # * Animated battler sufix
201 #   When using sprites, add this to the animated sprite sheet of the battler,
202 #   that way you can keep the original battler a single sprite and make
203 #   easier to setup their position on the troop
204 #-----
205 VE_SPRITE_SUFIX = "[anim]"
206 #-----
207 # * Intro fade
208 #   When true, there will be a small fade effect on the battlers during
209 #   the battle start (like RMXB default battle)
210 #-----
211 VE_BATTLE_INTRO_FADE = true
```

On these settings you can set the Sufix for battler sprites and the fade effect shown during the battle start.

VE_SPRITE_SUFIX: The sufix is added so you can have a single normal battler as the battler you set on the database, this make easier to place enemies in position on the troop setting.

VE_BATTLE_INTRO_FADE: The fade setting, if turned on, add a small appear fade when the battle start, it's a small effect that people might like to use... or not. It's up to the user and don't have any other effect on the battle.

[» Return to top](#)

Default Sprite Settings

```
212 #-----
213 # * Default sprite settings
214 #   This is the settings for all battler graphics that doesn't have
215 #   their own custom setting
216 #-----
217 VE_DEFAULT_SPRITE = {
218 # Basic Settings
219 # name:  value,
220   frames: 4,      # Number of frames
221   rows: 14,      # Number of rows
222   mirror: false,  # Mirror battler when facing right
223   invert: false,  # Invert the battler graphic
224   mode: :sprite,  # Graphic style (:sprite or :chasert)
225   action: nil,    # Action settings
```

This is the default setting for the battlers sprites, any battler that don't have it's own settings will

use these settings.

frames: and **rows:** - These values set how many frames the battler graphic will have, it must be adjusted to the sprite/character sheet, it's important to set the right values since it's used to set the frame dimensions.

mirror: - This setting is used to automatically invert the battler graphic when the battler is facing right. If set false the battler graphic will not be inverted, this can be used if the using charsets as battler or using graphics with different direction graphics.

invert: - This setting will invert automatically the graphic. The default direction of the battlers is facing left, and since the script auto adjust the direction (when set to do this) graphics facing right might look the wrong direction. Set this true if that happen.

mode: - this is who the battle will behave to get the battler graphic, it can be two values **:sprite** and **:charset**. When set as **:sprite**, the game will look into the Graphics/Battler folder for the graphic, when set as **:charset**, the game will look into the Graphics/Charset folder for the graphic (but it will use the **Battler** filename, so it's highly advised to have charsets with the same filename of the battler for those using this setup.

action: - this is name for the custom action, you can make a specific battler to have a custom action setting by setting the custom setting name on this value. the name is as arbitrary symbol and must match with the one set when creating actions. Leave nil to use the default settings.

[» Return to top](#)

Main Pose Settings

```

227 # Main Poses
228 # name:      row,
229 idle:        1, # Idle pose
230 guard:       2, # Guard pose
231 evade:       2, # Evade pose
232 danger:      3, # Low HP pose
233 hurt:        4, # Damage pose
234 attack:      5, # Physical attack pose
235 use:         6, # No type use pose
236 item:        6, # Item use pose
237 skill:       7, # Skill use pose
238 magic:       8, # Magic use pose
239 advance:     9, # Advance pose
240 retreat:    10, # Retreat pose
241 escape:     10, # Escape pose
242 victory:    11, # Victory pose
243 intro:      12, # Battle start pose
244 dead:       13, # Incapacited pose
245 ready:      nil, # Ready pose
246 itemcast:   nil, # Item cast pose
247 skillcast:  nil, # Skill cast pose
248 magiccast:  nil, # Magic cast pose
249 command:    nil, # Command pose
250 input:      nil, # Input pose
251 cancel:     nil, # Cancel pose
252 # You can add other pose names and call them within the action settings
253 # use only lowercase letters
254 } # Don't remove

```

This setting allows to setup wich row will be used with each pose. In most cases, these values are just used as reference, since the actual sprite row used is set on the pose setting itself, but in some cases (specifically the ready, command, input, cancel and cast poses) the existance of these values decide if the pose will be played. So if they're nil, they will be skiped

[» Return to top](#)

Custom Sprite Settings

```

249 #-----
250 # * Custom sprite settings
251 #   Theses settings are set individually based on the battler graphic
252 #   filename (even if using the charset mode, the setting will be based
253 #   on the battler filename, so it's suggested to use the same filename for
254 #   the battler and charset graphic when using charset mode)
255 #   Any value from the default setting can be used, if a value is not set
256 #   it's automatically uses the value from basic setting.
257 #-----
258 VE_SPRITE_SETTINGS = {
259 # 'Filename' => {settings},
260 #
261 # 'Sample 1' => {frames: 4, rows: 14, mirror: true, mode: :sprite,
262 #               action: nil},
263 # 'Sample 2' => {frames: 3, rows: 4, mirror: true, invert: false,
264 #               mode: :charset, action: :charset},
265 # 'Sample 3' => {frames: 3, rows: 4, mirror: false, invert: false,
266 #               mode: :charset, action: :kaduki},
267 # 'Scorpion'  => {frames: 4, rows: 14, mirror: true, mode: :sprite,
268 #               action: nil},
269 # 'Warrior_m' => {frames: 4, rows: 14, mirror: false, mode: :sprite,
270 #               action: :default},
271 # 'Actor1'   => {frames: 3, rows: 4, mirror: false, invert: false,
272 #               mode: :charset, action: :charset},
273 # '$Actor4'  => {frames: 3, rows: 4, mirror: true, invert: false,
274 #               mode: :charset, action: :kaduki},
275
276 } # Don't remove

```

This is the individual custom setting for battlers, the setting is made based on the **battler filename** (remember that actors only have battler graphic when using the script 'VE – Actor Battlers').

You must setup the **filename** and then add to the **settings** any value valid for the Default Settings. The values set here overwrites the value set on the Default Settings for the battler with the graphic that have the filename.

Filename	Settings
'Scorpion'	=> {frames: 4, rows: 14, mirror: true, mode: :sprite, action: nil},

[» Return to top](#)

Actions Settings

Action Settings Basic

```

329
330     # Pose displayed when evading attacks
331     <action: evade, reset>
332     pose: self, row evade, all frames, wait 4;
333     wait: 16;
334     </action>
335
336     # Pose displayed when a attack miss
337     <action: miss, reset>
338     </action>
339
340     # Pose displayed when reviving
341     <action: revive, reset>
342     </action>
343
344     # Pose displayed when dying
345     <action: die, reset>
346     </action>
347
348     # Make the target inactive
349     <action: inactive>
350     inactive;
351     </action>
352
353     # Set action advance
354     <action: advance, reset>
355     action: self, move to target;
356     wait: action;
357     </action>
358
359     # Movement to target
360     <action: move to target, reset>
361     wait: animation;
362     wait: targets, movement;
363     direction: targets;
364     move: self, move to;
365     jump: self, move, height 7;
366     pose: self, row advance, all frames, wait 4;
367     wait: movement;
368     </action>
369

```

The action setup are made by writting arbitrary text commands following the patterns set on the script. The script interprets these commands and execute the actions accordinally. The action flows exactly as set on the sequence. it won't do anything outside of the pose setting. You must be careful when setting up the poses. Since all commands are user written and it's very sensitive, be careful with typos.

[» Return to top](#)

Action Settings Structure

The settings are composed basically by 4 parts:

```

<action: action name, next action>
pose name: pose values;
</action>

```

Action name: The name of the current action

Next action: How the battler will behave after finishing the current action.
can be omitted (recommended omit only for victory pose)

Pose Name: The current pose name, each step of the pose must be set and will
plays in sequence

Pose Values: The settings of the current pose. They're arbitrary values that
changes according to the pose name.

[» Return to top](#)

Action Name

The name of the action, it's called
by the battle to execute the
animations, there is some default
ones that are preset on the script,
you can make custom names too,
and call them within specific actions:

```

      Action Name  Next Action
<action: attack, reset>
Pose Name pose: user, row attack, all frames, wait 4;
           wait: 4;
           anim: target, effect;
           wait: 8;
           effect: 100%;
           wait: 20;
           </action>
      Pose Values
```

```

      Action Name
<action: attack, reset>
pose: user, row attack, all frames, wait 4;
wait: 4;
anim: target, effect;
wait: 8;
effect: 100%;
wait: 20;
</action>
```

idle default idle pose

dead idle pose when dead

danger idle pose when the hp is low

guard idle pose when guarding

intro pose played at the battle start

victory pose played at the battle end when victorious

ready idle pose played after selecting an action

item cast idle pose played after selecting an item action

magic cast idle pose played after selecting a magic action

skill cast idle pose played after selecting a skill action

hurt pose when receive damage

evade pose when evades an attack

advance pose played at the movement to target

retreat pose played at the action end when returning to the position

escape pose played when escaping battles

defend pose played when using the defend command

attack pose used for the basic attack

dual attack pose used for normal attacks with 2 weapons

use pose used quando using actions without type

item pose used for item use

skill pose used for skill use

magic pose used for magic use

You can create you own names and use them calling them with other actions

[» Return to top](#)

Next Action

How the battler will behave after finishing the current action

Next Action

```
<action: attack, reset>
pose: user, row attack, all frames, wait 4;
wait: 4;
anim: target, effect;
wait: 8;
effect: 100%;
wait: 20;
</action>
```

loop the action loops indefinitely until another action take place

reset reset the pose after the end of action

X times repeat the action X times then reset

custom it's possible to add another action name, so that action will plays afterward.

[» Return to top](#)

Pose Name

The current pose name, each step of the pose must be set and will plays in sequence

Pose Name

```
<action: attack, reset>
pose: user, row attack, all frames, wait 4;
wait: 4;
anim: target, effect;
wait: 8;
effect: 100%;
wait: 20;
</action>
```

pose change the sprite pose
wait wait for the next action
move move to a certain position
jump jump to a certain position
anim displays an battle animation, vital for skills
icon displays an icon or image
loop displays a loop animation (requires the 'VE - Loop Animation' script)
tone changes the screen tone
hide hide some battlers display during the action
throw throws a object (icon or loop animation) on the target
sound plays a sound effect
plane display a custom looping picture located on the Graphics/Pictures folder
flash flash effect on the screen or targets
shake shake effect on the screen or targets
movie displays a movie
count setup a numeric counter to make some controls in the pose.
 The results of the counter can be used as condition for some poses.
action calls another action inside the current action
freeze halts movement temporaly
effect apply the action effect (damage, states, buffs...), vital for skills
picture display a custom picture located on the Graphics/Pictures folder
direction changes the direction the battler is facing
transition display a transition effect

» [Return to top](#)

Pose Values

The settings of the current pose.

They're arbitrary values that changes according to the **Pose Name**. Each Pose Name have it's own list of valid values. This is probably the most complex part of the settings, there's a lot of

different values you can apply to a pose to define how it will behave. You can add how many values you want by separating them with commas ",". Repeating values **have no effect**, and **some are self-exclusives**. To finish the setup of the values, use a semicolon ";". Here's a list of the possible values for each pose name, the * means values that can vary, genreally numbers or specific text:

```

<action: attack, reset>
pose: user, row attack, all frames, wait 4;
wait: 4;
anim: target, effect;
wait: 8;
effect: 100%;
wait: 20;
</action>
  
```

Pose Values

pose

target setting ([click here for details](#))

row * set the row, * can be a numeric value or the name of the row set on VE_POSE_SETTINGS

sufix * pose sufix, if added it will use a different graphic with the sufix name added to it.

x +-* horizontal position offssset of *

y +-* vertical position offssset of *

- # **angle** * incline the battler to the angle *
- # **spin** +-* spin the battler, * is the spin speed, if negative change the direction
- # **invert** mirror the battler graphic while displaying the pose
- # **all frames** plays the all frames in sequence
- # * **frames** plays * frames in sequence
- # **frame** * display the frame * of the current row, if used with all frames or * frames, will set the start of the display.
- # **wait** * work only with all frames or * frames, it'd the wait times for frame change
- # **return** work only with all frames or * frames, after reaching the last frame, the animation goes backward until reaches the first frame
- # **revert** work only with all frames or * frames, the animation plays backward, starting from the last frame going to the first
- # **loop** makes the current pose loops indefinitely until the pose changes

[» Return to top](#)

wait

- # **target setting** [\(click here for details\)](#)
- # * a numeric value, will wait * frames
- # **action** used only when calling a action inside another action, will wait that action ends before advancing
- # **animation** will wait until the end of any battle animation being displayed.
- # **movement** wait until the end of movement
- # **origin** wait until return to the original default position
- # **throw** wait until the end of any throwing animation
- # **tone** wait until the end of any tone change
- # **pose** wait until the end of the subjects pose
- # **freeze** wait until the end of the subjects freeze time

[» Return to top](#)

move

- # **move to** move to the front of the targets of the action
- # **setp forward** give a small step forward
- # **setp backward** give a small step backward
- # **retreat** return to the original position
- # **escape** move outised of the screen
- # **x** +-* adjust on the coordinate X of the target position
- # **y** +-* adjust on the coordinate Y of the target position
- # **height** +-* adjust on the coordinate Z of the target position
- # **speed** * movement speed
- # **teleport** teleport instantly to the target position

[» Return to top](#)

jump

move the jump follows the movement, must be used after the "move" pose command
height * height of the jump
speed * speed of the jump

[» Return to top](#)

anim

target setting [\(click here for details\)](#)
cast display cast animation (requires the script 'VE - Cast Animation')
effect display the current action animation, effective only during actions
id * display the animation of ID = *
weapon * display the animation of the weapon, * is the weapon slot

[» Return to top](#)

icon

target setting [\(click here for details\)](#)
icon * display icon ID *
weapon * display weapon icon in the slot *
armor * display armor icon in the slot *
shield display shield icon
action display current action icon
index * icon identifier for multiple icon display
image '*' use a custom image located on the Graphics/Pictures folder
delete remove the icon display
x +/- * icon x adjust
y +/- * icon y adjust
angle * icon angle
opacity * icon opacity
above display icon above battler
spin +/- * spin the icon, * is the spin speed, if negative change the direction
fade in * fade in effect, * is the final opacity
fade out * fade out effect, * is the final opacity
init zoom * initial zoom value, * is the zoom rate
end zoom * final zoom value, * is the zoom rate
zoom spd * speed of the zoom change

[» Return to top](#)

loop

target setting ([click here for details](#))

anim * Animation ID

[» Return to top](#)

tone

red +-* red tone vaule. * (0-255), can be negative

blue +-* blue tone vaule. * (0-255), can be negative

green +-* green tone vaule. * (0-255), can be negative

gray * gray tone vaule. * (0-255)

white shortcut to set the tone white, ignore other color settings

black shortcut to set the tone black, ignore other color settings

duration * duration of the tone change process, * time in frames

high priority makes the tone change on the upper layer, covering all the screen, including the windows and pictures

low priority makes the tone change on the lower layer, affecting only the battleback

clear return the tone to the value before change

[» Return to top](#)

hide

all battlers hides all actors and enemies

all enemies hides all enemies

all friends hides all allies

all targets hides all targets, only for attacks, skills and items

not targets hides all non-targets, only for attacks, skills and items

exclude user exclude the user from the hide list

include user include the user on the hide list

unhide cancel hide effect

[» Return to top](#)

throw

target setting ([click here for details](#))

icon * display icon ID *

weapon * display weapon icon in the slot *

armor * display armor icon in the slot *

shield display shield icon

action display current action icon

index * icon identifier for multiple icon display

image '*' use a custom image located on the Graphics/Pictures folder

x +-* icon x adjust

y +-* icon y adjust

z +-* icon z adjust

anim * display loop animation (requires the 'VE - Loop Animation' script)
arc * throw object in angled arc
speed * throw speed
revert revert the arc direction
return make the object go from target to the user
angle * icon angle
opacity * icon opacity
spin +-* spin the icon, * is the spin speed, if negative change the direction
fade in * fade in effect, * is the final opacity
fade out * fade out effect, * is the final opacity
init zoom * initial zoom value, * is the zoom rate
end zoom * final zoom value, * is the zoom rate
zoom spd * speed of the zoom change

[» Return to top](#)

sound

name '*' sound effect filename
volume * sound effect volume (0-100)
pitch * sound effect pitch (50-150)

[» Return to top](#)

plane

name '*' use a custom image located on the Graphics/Pictures folder
duration * duration of the appear and delete effect
move x +-* horizontal plane movement, * is the move speed, if negative change the direction
move y +-* vertical plane movement, * is the move speed, if negative change the direction
zoom x * horizontal zoom of the picture
zoom y * vertical zoom of the picture
opacity * picture opacity * (0-255)
blend * picture blen type (0: normal, 1: add, 2: subtract)

[» Return to top](#)

flash

target setting [\(click here for details\)](#)
screen this value is used, the flash target the screen
red * flash red tone vaule. * (0-255)
blue * flash blue tone vaule. * (0-255)
green * flash green tone vaule. * (0-255)
alpha * flash alpha vaule. * (0-255)

duration * flash effect duration

[» Return to top](#)

shake

target setting ([click here for details](#))

screen this value is used, the flash target the screen

power * shake effect power

speed * shake effect speed

duration * shake effect duration

[» Return to top](#)

movie

**Note: the video call have a tone change call that allows to apply a tone color for a smooth transition at the movie end

name '*' movie file located on the Movies folder

#

red +-* red tone vaule. * (0-255), can be negative

blue +-* blue tone vaule. * (0-255), can be negative

green +-* green tone vaule. * (0-255), can be negative

gray * gray tone vaule. * (0-255)

white shortcut to set the tone white, ignore other color settings

black shortcut to set the tone black, ignore other color settings

duration * duration of the tone change process, * time in frames

high priority makes the tone change on the upper layer, covering all the screen, including the windows and pictures

low priority makes the tone change on the lower layer, affecting only the battleback

[» Return to top](#)

count

**Note: the counter value is used as a condition for the count pose condition, [click here for details](#)

add * adds * to the counter

rand * adds a random value from 0 to *

max * max counter value, it return to zero after reaching the max

[» Return to top](#)

action

target setting ([click here for details](#))

action name of the next action

[» Return to top](#)

freeze

target setting [\(click here for details\)](#)

duration * freeze duration

[» Return to top](#)

effect

target setting [\(click here for details\)](#)

*% change the damage of the action to *%

weapon * damage caused will be based only on the stats of the weapon on slot *

[» Return to top](#)

picture

** Note: to use the move command you need before display the picture with the show value

id * set picture ID = *

show used to display the picture

move used to move the picture

name '*' use a custom image located on the Graphics/Pictures folder

center set the picture origin to the center

pos x picture x coordinate

pos y picture y coordinate

zoom x * horizontal zoom of the picture

zoom y * vertical zoom of the picture

opacity * picture opacity * (0-255)

blend * picture blen type (0: normal, 1: add, 2: subtract)

duration * duration of the picture move

[» Return to top](#)

direction

targets face the targets of the current action

return face the original position

down face down

left face left

right face right

up face up

[» Return to top](#)

transition

**** Note:** to execute a transition, you first must prepare it, any pose called after that will be still be processed, but no visual effect will be displayed until you call the execute value.

Use this to make graphical changes before the transition.

prepare freezes the screen to prepara for transition

execute execute the transition effect

duration * duration of the transition effect

name '*' transition file located on the Graphics/System folder

[» Return to top](#)

Target Setting

The tharget setting is a value allows to apply poses into another targets besides the one using the actions. It's a common value valid for most poses and works the same for all. Only one of the following value can be used.

the following values can be used as target setting:

targets pose targets the action targets

actor * pose targets enemy index = *

friend * pose targets friend index = *

enemy * pose targets enemy index = *

random pose targets a random target

random enemy pose targets a random enemy

random friend pose targets a random friend

all enemies pose targets all enemies

all friends pose targets all friends

self pose targets the battler calling the action

user pose targets the battler who started the action

in most cases user and self works the same, but if you call an action for another target from inside another action, user will be the one who called the first action

[» Return to top](#)

Pose Conditions

Pose Conditions are specific values valid for any pose, when added, the pose will be processed only if the condition is met.

hit only action will be played only if the attack hits, this works only with attacks, skills and items, and will always be false until the first hit

miss only action will be played only if the attack misses, this works only with attacks, skills and items, and will always be false until the first hit

count * this value should be used with the count pose name, the count pose name allows to setup a numeric counter, with that you can make a list of values that if matches the counter will validate the pose with this value

[» Return to top](#)

Note Tags

Weapons note tags

Tags to be used on Weapons note boxes.

action>

Changes the normal attack pose when using a weapon with this tag

action : action name

action>

Changes the double attack pose when using a weapon with this tag

action : action name

action>

Changes the physical skill pose when using a weapon with this tag

action : action name

action>

Changes the magical skill pose when using a weapon with this tag

action : action name

action>

Changes the item pose when using a weapon with this tag

action : action name

action>

Changes the movement for actions when using a weapon with this tag
this change the movement of all actions that have movement (by default only normal attacks and physical skills)

action : movement type name

Skills note tags

Tags to be used on Skills note boxes.

action>

Changes the pose of the skill with this tag

action : action name

action>

Changes the movement for actions when using a weapon with this tag
this change the movement of all actions that have movement (by default only normal attacks and physical skills)

action : movement type name

By default, only physical skills have movement. So, if you want to add movement to non-physical skills and items, add this tag.

action>

Changes the cast pose for items of the skill with this tag

action : action name

action>

Changes the cast pose for magic skills of the skill with this tag

action : action name

action>

Changes the cast pose for physical skills of the skill with this tag

action : action name

Actors note tags

Tags to be used on Actors note boxes.

<no intro>

This tag will make the actor display no intro pose at the battle start.

By default, all actor display intro pose

<no victory>

This tag will make the actor display no victory pose at the battle start.

By default, all actors display victory pose.

Enemies note tags

Tags to be used on Enemies note boxes.

<intro pose>

This tag will make the enemy display intro pose at the battle start.

By default, no enemy display intro pose

<victory pose>

This tag will make the enemy display victory pose at the battle start.

By default, no enemy display victory pose

<weapon x: y>

This allows to display weapons for enemies when using the pose value

'icon: weapon *'.

x : the slot index of the weapon (1: right hand, 2: left hand)

y : the incon index

<armor x: y>

This allows to display armors for enemies when using the pose value

'icon: armor *'.

x : the slot index of the armor (1: shield, 2: helm, 3: armor, 4: acc)

y : the incon index

States note tags

Tags to be used on States note boxes.

action>

Changes the idle pose while under the state with this tag

action : action name

Actors, Enemies, Classes, States and Armors note tags

Tags to be used on Actors, Enemies, Classes, States, Weapons and Armors note boxes.

<unmovable>

This tag allows to make a totally unmovable battler. The battler will not move to attack, neither be can forced to move by any action.

Comment calls note tags

Tags to be used in events comment box, works like a script call.

<no intro>

When called, the next battle will have no intro pose.

<no victory>

When called, the next battle will have no victory pose.

[» Return to top](#)

Examples, Samples and Tips

Step by Step Example

Making custom actions isn't something hard, but demand some work for good results. First of all you must plan your action, step by step, the script won't do anything alone. So you must plan beforehand each detail from the action.

So in this example I will make a custom pose for the RTP Default Skill 'Aura Blade', for the Kaduki template.

The animation will work as follow: The user will move to the target but will stay at a certain distance from it. There will be a casting animation, the user will rise the weapon, and then there will be another animation, after that, he will slice through the target.

So lets tart with the planning:

First the movement, the action will use a diferent move pattern so we will need to make a custom movment for it. Using the default move to target as base, we will add a distance between the user and the target.

- **Make a copy** of the default action: move to target (don't change the original!)

```
<action: move to target, reset>
wait: animation;
wait: targets, movement;
direction: targets;
move: self, move to;
pose: self, row 4, all frames, suffix _1, return, wait 8;
wait: movement;
</action>
```

- Change the action name, so we can assign it to the action. The name is totally arbitrary, just don't use a name used by other poses.

- Add the move distance. Notice that I added a x +64, this will make the battler move 64 pixels away from the target.

```
<action: aura blade move to, reset>
wait: animation;
wait: targets, movement;
direction: targets;
move: self, move to;
pose: self, row 4, all frames, suffix _1, return, wait 8;
wait: movement;
</action>
```

Now that we made the movement, we will need to make the skill pose itself, this one we will made from scratch. We must remember each step: first the cast animation, then rise the weapon, then another animation, then slash through the target.

```
<action: aura blade move to, reset>
wait: animation;
wait: targets, movement;
direction: targets;
move: self, move to, x +64;
pose: self, row 4, all frames, suffix _1, return, wait 8;
wait: movement;
</action>
```

- First let's name our action, since it's the name of the 'Aura Blade' skill, let's name it **aura blade**. As the next action we will use **reset**, since we want the user to end the skill after this action.

```
<action: aura blade, reset>
```

- Now, actually making the action, at the start we will be displaying the cast animation, so we need to change the battler pose to the cast pose.

self, row 4, suffix_3, all frames, wait 8, loop;

The value **self** is to set the user as target of the change.

The values **row 4** and **suffix_3**, is to set the pose as the "rising hand" pose from kaduki template.

The value **all frames** is set so the animation will use all frames from the row.

The **wait 8** is the time, in frames, for each frame change.

The **loop** value is to set the pose to repeat

```
<action: aura blade, reset>  
</action>
```

- Now let's set the cast battle animation. For that i use the value **anim: id 81;**, to show the 'Skill Activation' animation. I also added the pose **wait: animation;** so no action is done until the animations ends.

```
<action: aura blade, reset>  
pose: self, row 4, suffix_3, all frames, wait 8, loop;  
</action>
```

The next step is to raise the weapon, to make this we need two things: the user graphic animating 'rising' the weapon, and obviously the weapon. So we will use the **pose** and **icon** commands.

```
<action: aura blade, reset>  
pose: self, row 4, suffix_3, all frames, wait 8, loop;  
anim: id 81;  
wait: animation;  
</action>
```

- To make the pose animated, i use the following values: **self, row 2, suffix_3, all frames, wait 3.**

```
<action: aura blade, reset>  
pose: self, row 4, suffix_3, all frames, wait 8, loop;  
anim: id 81;  
wait: animation;  
pose: self, row 2, suffix_3, all frames, wait 3;  
</action>
```

Now the icon display, we need it to display the icon of the weapon following the rising movement. This part can take sometime since it will need some trial and error until you find the best position and angle for the icon.

- To set the first weapon icon to show, i use the value weapon 1, and adjust the angle and position. I also added a small wait so the battler keeps the weapon raised for a time.

```
<action: aura blade, reset>  
pose: self, row 4, suffix_3, all frames, wait 8, loop;  
anim: id 81;  
wait: animation;  
pose: self, row 2, suffix_3, all frames, wait 3;  
icon: weapon 1, angle 45, x -6, y +6;  
icon: weapon 1, angle 30, x -10, y +0;  
icon: weapon 1, angle 15, x -14, y -6;  
icon: weapon 1, angle 0, x -10, y -10;  
wait: 30;
```

- Now set the second animation, it's done the same as the first, but this time i added a wait time besides the animation wait so there is a delay between the animation and the attack

Now it's time to make the attack pose, the attack will be a slice though the target so we need to

```
</action>
```

through the target, so we need to make the battler move through the target, display the attack pose and the weapon icon.

- First the movement, the value **x -144** will make it move 144 pixels toward the target, the value **speed 15** is to make the slice move faster, the default move speed is 10.

- Now it's the attack pose, similar to the weapon rising pose, but using a different row.

- After the sprite animation, i add the weapon icon, i used the same values of the default physical attack

- Now to finalize the action, we display the acition animation and set the action effects with the values **anim: targets, effect;** and **effect: 100%;**, and add a small delay after the action with the value **wait: 20;**

To finish the settings all you need now is to add the custom poses to the skill with Notetags, just add the following tags to the skill note box:

<advance pose: aura blade move to>

<action pose: aura blade>

Here is the full code:

And here the in game animation:

```
<action: aura blade, reset>
pose: self, row 4, suffix _3, all frames, wait 8, loop;
anim: id 81;
wait: animation;
pose: self, row 2, suffix _3, all frames, wait 3;
icon: weapon 1, angle 45, x -6, y +6;
icon: weapon 1, angle 30, x -10, y +0;
icon: weapon 1, angle 15, x -14, y -6;
icon: weapon 1, angle 0, x -10, y -10;
wait: 30;
anim: id 110;
wait: animation;
wait: 30;

</action>
```

```
<action: aura blade, reset>
pose: self, row 4, suffix _3, all frames, wait 8, loop;
anim: id 81;
wait: animation;
pose: self, row 2, suffix _3, all frames, wait 3;
icon: weapon 1, angle 45, x -6, y +6;
icon: weapon 1, angle 30, x -10, y +0;
icon: weapon 1, angle 15, x -14, y -6;
icon: weapon 1, angle 0, x -10, y -10;
wait: 30;
anim: id 110;
wait: animation;
wait: 30;
move: self, x -144, speed 15;

</action>
```

```
<action: aura blade, reset>
pose: self, row 4, suffix _3, all frames, wait 8, loop;
anim: id 81;
wait: animation;
pose: self, row 2, suffix _3, all frames, wait 3;
icon: weapon 1, angle 45, x -6, y +6;
icon: weapon 1, angle 30, x -10, y +0;
icon: weapon 1, angle 15, x -14, y -6;
icon: weapon 1, angle 0, x -10, y -10;
wait: 30;
anim: id 110;
wait: animation;
wait: 30;
move: self, x -144, speed 15;
pose: self, row 1, suffix _3, all frames, wait 3;

</action>
```

animation:

```
<action: aura blade, reset>
pose: self, row 4, suffix _3, all frames, wait 8, loop;
anim: id 81;
wait: animation;
pose: self, row 2, suffix _3, all frames, wait 3;
icon: weapon 1, angle 45, x -6, y +6;
icon: weapon 1, angle 30, x -10, y +0;
icon: weapon 1, angle 15, x -14, y -6;
icon: weapon 1, angle 0, x -10, y -10;
wait: 30;
anim: id 110;
wait: animation;
wait: 30;
move: self, x -144, speed 15;
pose: self, row 1, suffix _3, all frames, wait 3;
icon: weapon 1, angle -90, x +12, y -16;
icon: weapon 1, angle -45, x +6, y -16;
icon: weapon 1, angle 0, x -6;
icon: weapon 1, angle 45, x -10, y +8;

</action>
```

```
<action: aura blade, reset>
pose: self, row 4, suffix _3, all frames, wait 8, loop;
anim: id 81;
wait: animation;
pose: self, row 2, suffix _3, all frames, wait 3;
icon: weapon 1, angle 45, x -6, y +6;
icon: weapon 1, angle 30, x -10, y +0;
icon: weapon 1, angle 15, x -14, y -6;
icon: weapon 1, angle 0, x -10, y -10;
wait: 30;
anim: id 110;
wait: animation;
wait: 30;
move: self, x -144, speed 15;
pose: self, row 1, suffix _3, all frames, wait 3;
icon: weapon 1, angle -90, x +12, y -16;
icon: weapon 1, angle -45, x +6, y -16;
icon: weapon 1, angle 0, x -6;
icon: weapon 1, angle 45, x -10, y +8;
anim: targets, effect;
effect: 100%;
wait: 20;

</action>
```

Movement to target for the skill 'Aura Blade'

```
<action: aura blade move to, reset>
wait: animation;
move: self, move to, x +64;
direction: targets;
pose: self, row 4, all frames, suffix _1, return, wait 8;
wait: movement;

</action>
```

Pose for the skill 'Aura Blade'

```
<action: aura blade, reset>
pose: self, row 4, suffix _3, all frames, wait 8, loop;
anim: id 81;
wait: animation;
pose: self, row 2, suffix _3, all frames, wait 3;
```

```

icon: weapon 1, angle 45, x -6, y +6;
icon: weapon 1, angle 30, x -10, y +0;
icon: weapon 1, angle 15, x -14, y -6;
icon: weapon 1, angle 0, x -10, y -10;
wait: 30;
anim: id 110;
wait: animation;
wait: 30;
move: self, x -144, speed 15;
pose: self, row 1, suffix _3, all frames, wait 3;
icon: weapon 1, angle -90, x +12, y -16;
icon: weapon 1, angle -45, x +6, y -16;
icon: weapon 1, angle 0, x -6;
icon: weapon 1, angle 45, x -10, y +8;
anim: targets, effect;
effect: 100%;
wait: 20;
</action>

```



[» Return to top](#)

Samples

On the script, there's some sample actions, like bow and gun attacks. It's possible to use them together with some note tags to achieve some fast result for some actions. Study these samples to use them as reference for new custom actions

Bow Attacks

The Bow sample is a action sequence for ranged attacks with bow and arrow. This sample have a arrow thown on the target, and for charsets battlers it have a bow graphic.

To use it add the following tags to the weapon textbox:

<advance pose: step forward> # To change the advance movement to a step forward

<attack pose: bow> # To add the bow animation for normal attacks

<skill pose: bow> # To add the bow animation for physical skills

If using charsets (both default and kaduki template) you will need also the following images on the **Graphics/Pictures** folder:



You can find this sample by seaching through the settings (with ctrl + f) for (available for default, charset and kaduki templates)

[» Return to top](#)

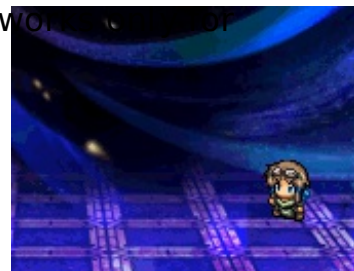
Spear Attacks

This is a sample for spear attacks, wich is a stabbing animation, works only for charset battlers.

To use it add the following tags to the weapon notebbox:

<attack pose: spear> # To add the stab animation for normal attacks

<skill pose: spear> # To add the stab animation for physical skills



You can find this sample by seaching through the settings (with ctrl + f) for (available for charset and kaduki templates)

[» Return to top](#)

Claw Attacks

This is a sample for claw attacks, it's a slashing strike but with different angles compared to the default strike, works only for charset battlers.

To use it add the following tags to the weapon notebbox:

<attack pose: claw> # To add the claw animation for normal attacks

<skill pose: claw> # To add the claw animation for physical skills



You can find this sample by seaching through the settings (with ctrl + f) for (available for charset and kaduki templates)

[» Return to top](#)

Gun Attacks

The Gun sample is a action sequence for ranged attacks with guns. This animation includes a shot sound (using the RTP 'Gun1' SE), works only for charset battlers. For non-charset battlers, just change the movement to step forward.

To use it add the following tags to the weapon notebbox:

<advance pose: step forward> # To change the advance movement to a step forward

<attack pose: gun> # To add the gun animation for normal attacks

<skill pose: gun> # To add the gun animation for physical skills



You can find this sample by seaching through the settings

(with ctrl + f) for (available for charset and kaduki templates)

[» Return to top](#)



Drain Magic

The Drain magic sample shows how projectiles can be also set from the target to the actor. On this sequence, the "globe" icon goes to the target to the user to show the drain effect. Also for the drain to have effect it's important to add the drain value on the settings.

To use it add the following tag to the skill notebook:

<action pose: drain> # To add the drain animation to the skill

You can find this sample by seaching through the settings (with ctrl + f) for

[» Return to top](#)



Movie Magics

It's possible to add videos to be displayed during actions, so i've made two examples of video magics. You will need the video files '[Tempest.ogv](#)' and '[Meteor.ogv](#)' on your Movies folder.

To use it add one of following tags to the skill notebook:

<action pose: tempest> # For the tempest magic

<action pose: meteor> # For the meteor magic

You can find this sample by seaching through the settings (with ctrl + f) for or (available for default, charset and kaduki templates)

[» Return to top](#)

Remember: these samples are just for study, use them as reference to make your own custom action sequences

Frequently Asked Questions

These are some questions that people might have frequently.

If someone is stupid enough to ask something that is already covered by this guide, expect a very bad treatment.

This faq will be expanded as it needed.

Q: Can you make a demo?

A: No, no and **NO!** I didn't spend a lot of time making this huge guide for some lazy ass to just use a demo to make their game, because they're too lazy to actually read the manual. Don't ask, I don't care about why the hell you can't make the system work without a demo. If you have any issue ask here (but first read the damn manual!)

[» Return to top](#)

Q: I'm having compatibility issue, how can I solve that?

A: You can't solve, unless you're a good scripter. So, depending on the script, I **might** make a compatibility patch. I have no responsibility on compatibility issues. Neither do I have an obligation to solve any issue with third party scripts. If I don't agree with making a patch to solve your issue and you can't do it by yourself you have only one option: **DEAL WITH IT!**

[» Return to top](#)

Q: The actors are invisible, what can I do to show them?

A: The basic function of this script is adding animated graphics for the **existing battlers** and control them, it doesn't add display for invisible battlers. So, actors are invisible since they're not displayed by default. To display the actors you must use the script '[VE – Actors Battler](#)'

[» Return to top](#)

Q: The battlers are floating on the battleback, how can I fix that?

A: Both enemies and actors positions are not set on the Animated Battle script. The enemies positions are set on the database Troops tab. Actors positions are set on the script '[VE – Actors Battler](#)', read the instructions of that script to know how to setup them.

[» Return to top](#)

Q: I can't use charsets in battle, did I do something wrong?

A: The settings are based on the **battler filename**, so even if you decide to use charsets for the battlers, you **still need to assign a battler graphic to the actor**, and you **actually need a battler graphic with that name**. So you will need a battler and a charset with the **same filename** for the battlers that will use charset. In case of actors, you still need to setup the '[VE – Actors Battler](#)' script properly to display actors battlers

[» Return to top](#)

Q: The custom sprites work, did I do something wrong?

A: The first setting of the script is the **VE_SPRITE_SUFFIX**, this is the suffix for animated battler. So the animated battlers using custom sprites will need this suffix. So you will need two files: one without this suffix that will be the one you will assign on the database (it's highly advised to use a small battler with 1 frame, it will make easier to set positions in the troops tab), and another with the suffix, that will be the one used during the battle.

[» Return to top](#)

Q: Can you give me or tell me where i can find some graphics for the battlers?

A: No. This is a script, not a resource package. Graphics for the battle system aren't my responsibility. I did the hard part making the script, the rest is up to the users.

[» Return to top](#)

Q: When a battler dies it disappear, how can i make it stay there with the dead pose?

A: This is the default behavior for battlers. To avoid that, you must use the Trait: "Collapse Effect: Not Disappear". If you want it to remove the fade from all battlers, just add the trait do the dead state, if you want to add it only to specific battler, add it to the battler traits.

[» Return to top](#)