

Battle Animation Commands

The first two bytes of an animation script are a header that determines the speed and alignment of the animation. The real script starts after that. Of this, animation speed is the high nibble of the first byte. Alignment is the highest 3 bits of the second byte. All of the other bits appear to be unused. After this header, the bytes of the animation script behave as follows:

\$00-\$1F	Show frame number [\$00-\$1F]
\$20-\$7F	-
\$80 xx	\$C878 Command \$80 Subcommands
	\$00 \$D9BE quadra slam/quadra slice
	\$01 \$D9A9
	\$02 \$D981
	\$03 \$D995
	\$04 \$D96E Randomize vector angle and
position (init fire dance sprites)	
	\$05 \$D938 bum rush
	\$06 \$D907 init tornado (w wind/spiraler)
	\$07 \$D8F2 move tornado to thread position
(w wind/spiraler)	
	\$08 \$D8EB move thread to vector position (w
wind/spiraler)	
	\$09 \$D879 update character/monster sprite
tile priority for tornado (w wind/spiraler)	
	\$0A \$D82B white/effect magic intro
	\$0B \$D7E3 Update Esper Pre-Animation Balls
Position	
	\$0C \$D753 move to first sprite thread
position	
	\$0D \$D7C4
	\$0E \$D79D
	\$0F \$D779
	\$10 \$D73E move to target position
	\$11 \$D727 Randomize vector angle
	\$12 \$D734 init polar movement
	\$13 \$D6E5 toggle imp graphics for target
(imp)	
	\$14 \$D6BD make target vanish (vanish)
	\$15 \$D698 move circle to thread position
	\$16 \$D68E
	\$17 \$CBC1 update sprite layer priority
based on target	
	\$18 \$D677 Load Sketched Monster Palette
	\$19 \$D62E sketch
	\$1A xx \$CB74
	\$1B \$CB5D transform into magicite
	\$1C \$CB6A decrement screen brightness
	\$1D \$CB61 transform into magicite
	\$1E \$D56B scroll bg3 for pearl

pearl (horizontal)	\$1F	\$D5FC init bg3 hdma scroll data for
pearl (vertical)	\$20	\$D59F init bg3 hdma scroll data for
Priority	\$21	\$D54E Update Rotating Sprite Layer
	\$22	\$D4F2 pearl wind
	\$23	\$D4BE pearl wind
	\$24	\$D49B Clear BG3 HDMA scroll data
	\$25	\$D4AD Clear BG1 HDMA scroll data
	\$26 xx	\$D493 Enable/Disable Character Color
Palette Updates (x: 1 = disable, 0 = enable)	\$27 xx	\$D48B Hide/Show Characters for Esper
Attack (x: 1 = hide, 0 = show)	\$28 --oo----	\$D45C Affects all characters o: sprite priority
attack)	\$29 xx	\$D454 Hide/Show Cursor sprites (esper
	\$2A xx	\$D44C load animation palette xx, sprite
(inferno)	\$2B xx	\$D43C load animation palette xx, bg1
(justice, earth aura)	\$2C xx	\$D444 load animation palette xx, bg3
	\$2D xxxx yyyy zzzz	\$D423 Jump based on battle type xxxx: jump location if normal
attack		yyyy: jump location if back
attack or (side attack and attacker is character 3 or 4)		zzzz: jump location if pincer
attack or (side attack and attacker is character 1 or 2 or monster)	\$2E xx yy	\$D3E4 Move sprite to (\$xx, \$yy)
	\$2F	\$D3AF
	\$30 xx	\$D38E load animation palette xx for
character 1	\$31 xx	\$D365 move in wide vertical sine wave
with speed xx (hope song, sea song)	\$32 xxxx yyyy	\$D33E Jump to xxxx if facing left, yyyy
if facing right	\$33 xx	\$D2D2 update rainbow gradient lines
	\$34	\$D28D copy monster palettes to
character palettes (hope song)	\$35	\$D27A use character palettes for
monster sprite data (hope song)	\$36	\$D267 restore palettes for monster
sprite data (hope song)	\$37	\$D256 clear fixed color value hdma data
(\$2132)	\$38	\$D24D enable high priority bg3
(justice)		

Cross, Carbunkl, Odin/Raiden)	\$39 xx	\$D1E6 update blue gradient lines (S.
	\$3A xx	\$D1DE set gradient line intensity to xx
animation palette	\$3B	\$D1B0 Set target's color palette to
normal	\$3C	\$D18A Set target's color palette to
	\$3D	\$D12E quadra slam/quadra slice
(\$212C)	\$3E xx	\$D126 Set main screen designation
	\$3F	\$D0E0 sonic dive
	\$40 -----mmm	\$D0D3 Set screen mode (\$2105) m: screen mode
(positive is smaller) and move (dx,dy)	\$41 cx cy dx dx	\$D06D Shrink/Grow BG1 by (cx,cy)
(\$211A)	\$42 -----vh	\$D064 Set MODE7 Settings register
		v: vertical flip h: horizontal flip
	\$43	\$D00B moon song/charm
	\$44	\$CFCC fire beam/bolt beam/ice beam
Hardware Register (\$2123)	\$45 xx	\$CFC0 Set BG1/BG2 Mask Settings
	\$46	\$CFB9
	\$47	\$CFAA
	\$48	\$CF8D clear
	\$49	\$CF7F ink hit/virite
attack for all characters	\$4A	\$CF6A disable stepping forward to
(megazerker)	\$4B	\$D2CC update red/yellow gradient lines
	\$4C	\$CF45 move triangle to thread position
target	\$4D	\$CF1C set vector from triangle to
	\$4E	\$CF15 clear frame offset
position	\$4F	\$CEF0 move to attacking character
	\$50 xx	\$CE9A
	\$51	\$CE62 rippler
	\$52	\$CE29 stone
	\$53	\$CDDF r.polarity
	\$54	\$CDC4 r.polarity
	\$55	\$CD72 quasar
	\$56	\$CD28 goner
(xx -> \$2124)	\$57 xx	\$CD1F set bg3/bg4 window mask settings
	\$58 xx	\$CD17 change circle shape to xx
	\$59	\$CD12 goner/flare star
	\$5A	\$CD0D mind blast
	\$5B	\$CD08 mind blast
	\$5C	\$CD03 mind blast

	\$5D	\$CCDF
	\$5E	\$CC98 overcast
	\$5F xx	\$CC93 Increase/Decrease Blue Backdrop
Gradient (used by Overcast)		
	\$60 aabbccdd	\$CC3F Toggle attacker status (aa =
status 1, etc.) (morph/revert)		
	\$61 xx yy zz	\$CC1A
	\$62	\$CBF5 evil toot/fader
	\$63 xx	\$D361 move in narrow vertical sine wave
with speed xx (evil toot)		
	\$64	\$CBE5 purifier/inviz edge
	\$65	\$CBE0 change rainbow palette
	\$66	\$CBDB shock wave
	\$67	\$CBD6 Load Extra Esper Palette
(purifier)		
	\$68	\$CBD1 purifier
	\$69	\$CBB6 update sprite layer priority
based on attacker		
	\$6A	\$CBAC align bottom of thread with
bottom of target (ice 3)		
	\$6B	\$CBB1 l? pearl
	\$6C	\$CB5A overcast
	\$6D	\$CB56 disable battle menu
	\$6E	\$CB51
	\$6F	\$CB4D
	\$70	\$CB43
	\$71	\$CB34 restore character palettes
(purifier/hope song)		
	\$72 xx	\$CB48 Branch forward xx if attack
didn't miss		
	\$73 xx	\$CB1D Set graphics for dice roll (xx =
die index)		
	\$74	\$CAB8
	\$75	\$CAE5 super ball
	\$76	\$CAD6 seize
	\$77	\$CADB seize
	\$78	\$CAE0 discard
	\$79	\$CAC2 Characters Run to Left Side of
Screen (takes 56 loops to reach other side)		
	\$7A	\$CAC7 Characters Run to Right Side of
Screen		
	\$7B	\$CACC Flip All Characters (after
running to opposite side of screen)		
	\$7C	\$CAD1 Swap Target and Attacker
	\$7D xx	\$CABD Branch forward xx bytes if dragon
horn effect is active		
	\$7E	\$CAA1 Flip Target Character Vertically
	\$7F	\$CA9D Hide all monsters
	\$80	\$CA65 boss death

	\$81	\$CA61
	\$82	\$CA3D boss death
	\$83	\$CA38
	\$84	\$CA29 chadarnook exit
	\$85	\$CA24 chadarnook exit
on sprite X position)	\$86 xx	\$CA0F Play Sound effect xx (pan based
on sprite Y position)	\$87 xx	\$C9F7 Play Sound effect xx (pan based
	\$88	\$C9C9
	\$89 xx	\$C9C1
	\$8A	\$C9A9 Set Target Monster Sprite
Priority to 0		
	\$8B	\$C9A5 Play Ching Sound effect
	\$8C xx	\$CA09 Play Sound effect xx (pan center)
\$81 xx yy	\$F347	Change attacking character's graphic to \$xx if facing left, \$yy if facing right
\$82 xx yy	\$F33F	Change targetted character's graphic to \$xx if facing left, \$yy if facing right
\$83 dddxxxxx	\$F377	Move Thread
		d: direction (0 = down/forward, 1 = down, 2 = down/back, 3 = forward, 4 = back, 5 = up/forward, 6 = up, 7 = up/back)
		x: distance - 1
\$84 xx	\$F7B3	Set Animation Speed to xx
\$85	\$F89D	Move Thread to Attacker Position
\$86 dddxxxxx	\$F491	Move Attacker (data same as command \$83)
\$87 dddxxxxx	\$F476	Move Target (data same as command \$83)
\$88 xx	\$F71D	Jump forward with weapon for "Fight" command (xx = frame index)
\$89 xx	\$F7BC	Loop start (loop xx times)
\$8A	\$F82F	Loop end
\$8B xx	\$F7E6	Animated loop start (loop xx times, increment frame offset each loop, start at frame 0)
\$8C	\$F84B	Animated loop end
\$8D dddxxxxx	\$F263	Move Thread if Animation is Flipped Horizontally
\$8E bf-----h	\$F27A	Show Thread Above/Below Other Sprites
		b: show below
		f: show in front
		h: 1 = show with weapon hand, 0 = show opposite weapon hand
\$8F dddxxxxx	\$F263	Move Thread if Animation is Flipped Horizontally
\$90 --oo----	\$F255	Set Thread's Sprite Tile Priority
\$91	\$F8B4	Move This Thread to Attacker Thread Position
\$92 xx yy	\$FADB	Move Thread along Vector (speed xx, code branch yy)
\$93 xx	\$FA3D	Set position on vector
\$94	\$F8E0	Set vector from attacker to a random location on the target (GP Rain, AutoCrossbow)
\$95	\$F9E6	Set vector from attacker to target
\$96 xx yy	\$FB63	Branch if ??? (xx = bytes to branch backwards)
\$97	\$FBD7	boomerang/wing edge/full moon/rising sun
\$98 xx bbbbeeee	\$FBA8	Increment graphic index offset every x frame(s),

(b..e)

\$99 ----ppp- \$FC37 Set Thread Palette to p

\$9A \$FC40 Set Thread Facing Direction to Match Attacker

\$9B xx \$F31A

\$9C xx \$F2A2

\$9D xx \$F2F1

\$9E xx \$F2B6

\$9F xx \$F7CF Animated Loop start (loop count equal to the number of active threads, xx = 0) (autocrossbow)

\$A0 xx yy \$FA4B Jump Forward Along Vector (speed xx, code branch yy)

\$A1 xx yy \$FA90 Jump Backward Along Vector (speed xx, code branch yy)

\$A2 \$F2E1 drill

\$A3 xxxx \$F1E5 Shift Color Palette Left

\$A4 xxxx \$F21D Shift Color Palette Right

 aaaabbbb ccccdddd

 a: offset color

 b: number of colors

 c: palette index

 d: speed (number of loops per shift)

\$A5 aabbccxyyyzz \$F0EC circle origin (aa,bb), (80,80) is center of screen

 cc: grow speed?

 xx

 yyyy: max size

\$A6 xx yy zz \$F094 Move circle (\$xx,\$yy), size changes by zz (signed)

\$A7 \$F088 Update Circle?

\$A8 \$F073 Move circle to attacker

\$A9 xx yy \$EFC8 Move circle (\$xx,\$yy) (signed, based on character facing direction)

\$AA rgbffffff \$EC6E Set sprite palette 3 color subtraction (absolute)

 r: affect red

 g: affect green

 b: affect blue

 f: amount to subtract

\$AB rgbffffff \$EC58 Set sprite palette 3 color addition (absolute)

 r: affect red

 g: affect green

 b: affect blue

 f: amount to add

\$AC xx yy \$EE9C set background Scroll HDMA data

 123ffffff vhaaaaaa

 1: affect BG1

 2: affect BG2

 3: affect BG3

 f: frequency

 v: vertical

 h: horizontal

 a: amplitude (max 14, must be even ???)

\$AD nnxxxxxx \$EFA3 Set BG Scroll HDMA Index

	n: BG (0,1,2)
	x: index
\$AE vh---123	\$ED86 Update Scroll HDMA data
	v: vertical
	h: horizontal
	1: affect BG1
	2: affect BG2
	3: affect BG3
\$AF rgbffffff	\$EBDA Set background palette color subtraction (absolute)
\$B0 rgbffffff	\$EBC4 Set background palette color addition (absolute)
\$B1 rgbaffff	\$ECAC Set sprite palette 1 color subtraction (absolute)
\$B2 rgbaffff	\$EC96 Set sprite palette 1 color addition (absolute)
\$B3 rgbaffff	\$EC4F Add color to sprite palette 3 (relative)
	r: affect red
	g: affect green
	b: affect blue
	a: 0 = increase addition amount, 1 = decrease addition amount
\$B4 rgbaffff (relative)	\$EC46 Subtract color from sprite palette 3 palette
	r: affect red
	g: affect green
	b: affect blue
	a: 0 = increase subtraction amount, 1 = decrease subtraction amount
\$B5 rgbaffff	\$EBB2 Add color to background palette (relative)
\$B6 rgbaffff	\$EBBB Subtract color from background palette (relative)
\$B7 rgbaffff	\$EC84 Add color to sprite palette 1 (relative)
\$B8 rgbaffff	\$EC8D Subtract color from sprite palette 1 (relative)
\$B9 rgbffffff	\$ECEA Set monster palettes color subtraction (absolute)
\$BA rgbffffff	\$ECD4 Set monster palettes color addition (absolute)
\$BB rgbaffff	\$ECCB Add color to monster palettes (relative)
\$BC rgbaffff	\$ECC2 Subtract color from monster palettes (relative)
\$BD abcd----	\$EAA1 Hide/Show BG1/BG3 Animation Thread Graphics
	a: affect bg1
	b: affect bg3
	c: bg1 (0 = show, 1 = hide)
	d: bg3 (0 = show, 1 = hide)
\$BE xx	\$EA98 Set Screen Mosaic to xx (\$2106)
\$BF xxxx	\$EA85 Jump to Subroutine \$xxxx
subroutine!	NOTE! Animations can only handle one layer of subroutines,
subroutines,	As such, animation subroutines cannot lead to other
existing animation!	meaning you can't use them as easy hooks to amend an
bytes,	If you're changing an animation and you need more
	you're going to have to relocate it to another spot in

```

the D0 bank instead.
$C0          $EA76 Return from Subroutine
$C1 xx yy    $EA05 xx = vector movement speed ???, yy = number of bytes
to branch backwards
$C2 abc----- $E9EB Unpause Animation
                a: unpause bg1
                b: unpause bg3
                c: unpause sprites
$C3          $F02F Move circle to target
$C4 ab----- $E99F Move BG1/BG3 Thread to This Thread's Position
                a: affect bg1
                b: affect bg3
$C5 4 addresses $E8FB jump based on swdtech hit
$C6 xx yy    $E830 quadra slam/quadra slice
$C7 <varies> $C873 Command $C7 Subcommands
                $00 xx      $C2C39B Change Attacking Character Facing
Direction (xx: 0 = face left, 1 = face right)
                $01          $C2C362 reset position offsets for
attacking character
                $02          $C2C31E save attacking character position
                $03          $C2C339 restore attacking character
position and reset offsets
                $04          $C2C303 restore attacking character
position
                $05 xx      $C2C2B7 (unused)
                $06 xx yy   $C2C26A
                $07          $C2C247 update character action based on
vector direction (walking)
                $08 xx yy   $C2C1D6 set vector target (xx,yy) from
attacker
                $09          $C2C1B3 update character action based on
vector direction (arms up)
                $0A xx      $C2C194 (unused)
                $0B xx yy zz $C2C171 spc command
                $0C xx yy   $C2C136 change actor xx graphic index to
yy
                $0D xx      $C2C115
                $0E xx      $C2C0F8 enable/disable screen shaking (xx
-> $6285)
                $0F          $C2C0F2 (unused)
                $10 xx      $C2C0B9
                $11          $C2C0B0 disable running with L+R
$C8 xx        $E7B1 Set attacker modified graphic index
$C9 xx        $DAE4 Play sound effect xx ($00 means play default for this
animation)
$CA xx        $E798 Set character action speed, xx=00 is normal, xx=01 is
double speed
$CB eddddddd $E779 Enable/Disable Echo Sprites (4 copies of character
sprite)

```


		e: 1 = enable, 0 = disable
		d: frame delay between echo sprites (bitmask)
\$CC rgbffffff	\$EC24	Set sprite palette 2 color subtraction (absolute)
		r: affect red
		g: affect green
		b: affect blue
		f: amount to subtract
\$CD rgbffffff	\$EC02	Set sprite palette 2 color addition (absolute)
		r: affect red
		g: affect green
		b: affect blue
		f: amount to add
\$CE rgbaffff	\$EBF0	Add color to sprite palette 2 (relative)
		r: affect red
		g: affect green
		b: affect blue
		a: 0 = increase addition amount, 1 = decrease addition amount
\$CF rgbaffff	\$EBF9	Subtract color from sprite palette 2 (relative)
		r: affect red
		g: affect green
		b: affect blue
		a: 0 = increase subtraction amount, 1 = decrease subtraction amount
\$D0 vhftpppm	\$E746	Set sprite data for all character/monster sprites
\$D1 xx	\$E73D	Validate/Invalidate Character/Monster Order Priority (0 = validate, 1 = invalidate)
\$D2 xx yy	\$F86D	Set Target Position (xx,yy) doesn't actually move target
\$D3	\$F044	Move Circle to Attacking Character
\$D4 xxxx yy	\$E722	Set Color Addition/Subtraction Data
	shbo4321 mmss--cd (+\$2130)	
		s: 0 = add, 1 = subtract
		h: 0 = full add/sub, 1 = half add/sub
		bo4321: layers affected by add/sub (b = background)
		m: 0
		s: 0
		c: 0 = fixed color add/sub, 1 = subscreen
		d: 0
	---o4321	subscreen designation (\$212D)
		o4321: layers to add/sub
\$D5 -----vh	\$E707	Flip Monster (v = vertical, h = horizontal)
\$D6 xx yy	\$E6CD	scroll background to (xx,yy), typically used to shake the screen
\$D7 xx	\$E68D	Move Fire Dance Sprites
\$D8 xx yy zz	\$E5F9	x: x speed, y: y speed, z: ???
\$D9 xx	\$E5F0	(bum rush)

\$DA xxxx	\$E528 update tornado (w wind/spiraler)
\$DB xx	\$E509 Branch forward xx bytes if character already stepped forward to attack
\$DC	\$E43A Rotate Triangle 2D
\$DD xx yy dd rr	\$E416 Init Triangle
\$DE	\$E401 move triangle to attacker position
\$DF	\$E3EC move triangle to target position
\$E0 xx yy dd rr	\$E3A0 Modify Triangle
\$E1 xx	\$E328 show/hide attacker sprite
\$E2	\$DD8D Part of Monster Steal/Jump, must be used after \$E4 and before \$E3. Leap forward on \$E4 arc.
\$E3	\$DD42 Part of Monster Steal/Jump, must be used after \$E4 and \$E2. Leap backward on \$E4 arc and restore coordinates.
\$E4	\$E286 Part of Monster Steal/Jump, saves original coordinates and calculates target-to-monster inverse parabola
\$E5 xx yy zz	\$E15D yy = number of bytes to branch backwards
\$E6 xx yy zz	\$E1B3 yy = number of bytes to branch backwards
\$E7	\$E25A calculate vector from attacking character to target
\$E8 rr tt	\$DCDF Move rr,tt in polar coordinates (radius,theta)
\$E9 xx yy	\$DC9B Move randomly (0...xx,0...yy)
\$EA 13--xxxx	\$DC81 Set BG Tile Data Quadrants 1 = affect bg1 3 = affect bg1 x = quadrant
\$EB xxxx ...	\$DC66 Jump to \$xxxx... based on thread index (number of addresses is number of threads)
\$EC xx	\$DC55 Change Thread Layer (0 = sprite, 1 = bg1, 2 = bg3)
\$ED	\$DB8F
\$EE --oo----	\$E5C5 Set Target's Sprite Tile Priority
\$EF rr tt	\$DCD9 similar to \$E8
\$F0 5 words	\$DB6C Jump based on current target index (char1, char2, char3, char4, monster)
\$F1 m-----s	\$E2C0 Hide or show specified sprite m = affect monster s = show sprite
\$F2	\$F980 Set a trajectory from target center to attacker
\$F3 5 words	\$DB64 Jump based on current attacker index (char1, char2, char3, char4, monster)
\$F4 -----t	\$F30F Set Sprite Layer Priority
\$F5	\$F7FC Loop End (loop until no threads are active)
\$F6	\$E4A2 Rotate Triangle 3D
\$F7 xx	\$DB50 Wait until vertical scanline position xx
\$F8 xxxx yyyy	\$DB31 Jump to either xxxx or yyyy if magitek mode is enabled
\$F9 xx yy zz	\$DAF9 set attacker magitek armor action to xx
\$FA xxxx	\$DB23 Jump to \$xxxx
\$FB rgbfffff	\$ED4C Set character palettes color subtraction (absolute) r: affect red g: affect green b: affect blue

		f: amount to subtract
\$FC	rgbffffff	\$ED12 Set character palettes color addition (absolute)
		r: affect red
		g: affect green
		b: affect blue
		f: amount to add
\$FD	rgbaffff	\$ED00 Add color to character palettes (relative)
		r: affect red
		g: affect green
		b: affect blue
		a: 0 = increase addition amount, 1 = decrease addition amount
		f: amount to increase/decrease
\$FE	rgbaffff	\$ED09 Subtract color from character palettes (relative)
		r: affect red
		g: affect green
		b: affect blue
		a: 0 = increase subtraction amount, 1 = decrease subtraction amount
		f: amount to increase/decrease
\$FF		End of Animation

From:

<https://www.ff6hacking.com/wiki/> - ff6hacking.com wiki

Permanent link:

https://www.ff6hacking.com/wiki/doku.php?id=ff3:ff3us:doc:asm:codes:battle_animation_script

Last update: 2023/10/10 03:51

