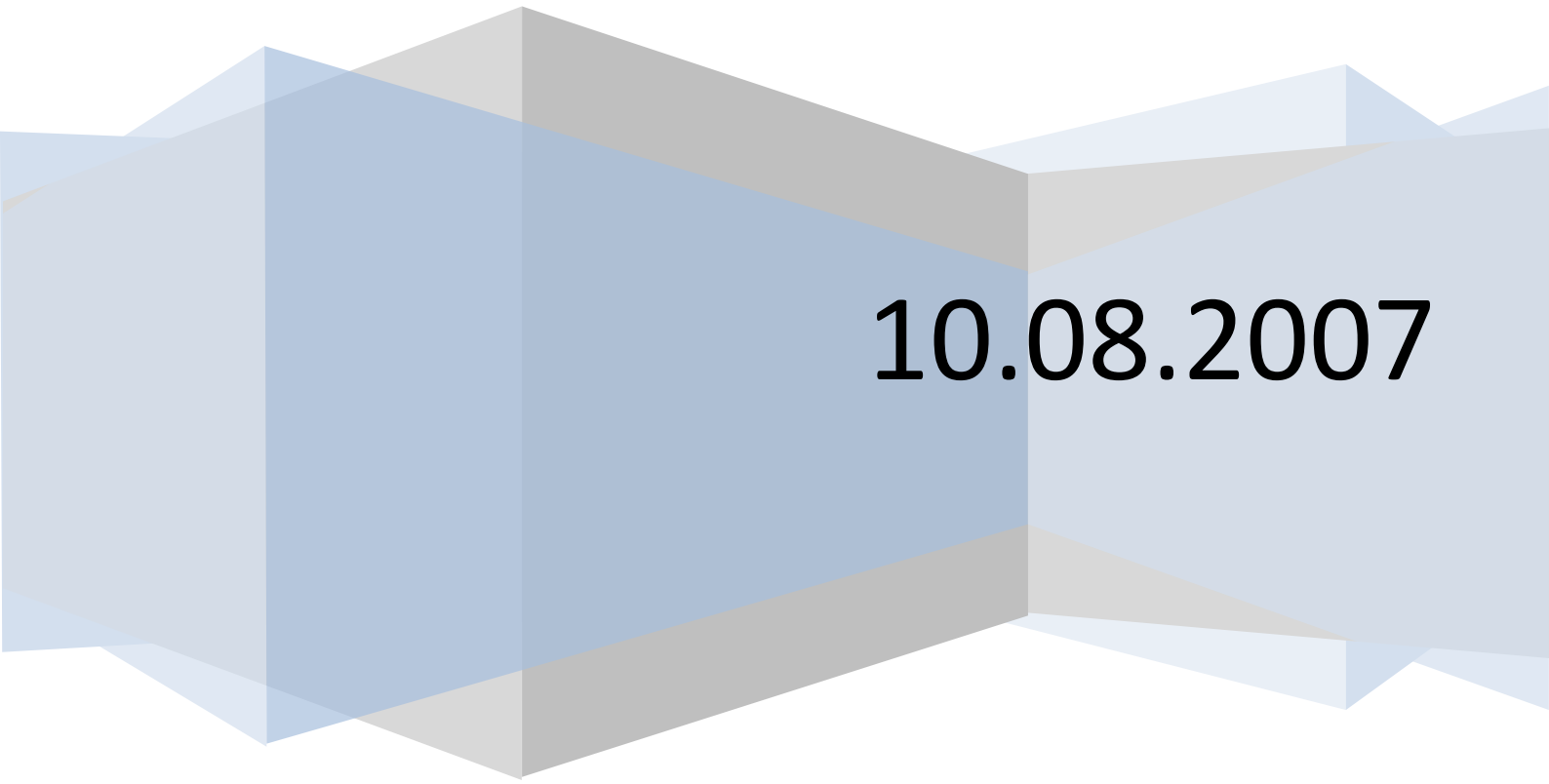
Changer de menu

Scripts de GuardianForce

Marcoest



10.08.2007

Voici un script pour changer le menu de votrer jeu.

Aperçu du script

Script testé, il fonctionne.

Vous pouvez remplacez le script Scene_Menu, ou créer un nouveau script nommé Menu_Bis. Mettez-y le code ci-dessous.

```
#==============================================
# by Xk8
# Script téléchargé sur RPG-création - www.rpg-creation.com
# Ce script ne doit pas être utilisé avec de battlers animés
# Installation : remplacez juste le script scene_menu par ce script
#==============================================

class Scene_Menu

def initialize(menu_index = 0)
@menu_index = menu_index
end

def main
@spriteset = Spriteset_Map.new
s1 = $data_system.words.item
s2 = $data_system.words.skill
s3 = $data_system.words.equip
s4 = "Statut"
s5 = "Sauvegarder"
s6 = "Quitter"
@command_window = Xcommand.new(640, [s1, s2, s3, s4, s5, s6])
@command_window.index = @menu_index
@command_window.x = -640
@command_window.opacity = 220
if $game_party.actors.size == 0

@command_window.disable_item(0)
@command_window.disable_item(1)
@command_window.disable_item(2)
@command_window.disable_item(3)
end

if $game_system.save_disabled
@command_window.disable_item(4)
end

@playtime_window = XPlayTime.new
@playtime_window.x = 640
```


RPG-LIVE.NEUF.FR

```
@playtime_window.y = 64
@playtime_window.opacity = 220

@gold_window = XGold.new
@gold_window.x = 640
@gold_window.y = 320
@gold_window.opacity = 220

@location_window = Xlocation.new
@location_window.x = 640
@location_window.y = 160
@location_window.opacity = 220

@status_window = Xstatus.new
@status_window.x = -400
@status_window.y = 64
@status_window.opacity = 220

@realtime_window = Xtime.new
@realtime_window.x = 640
@realtime_window.y = 224
@realtime_window.opacity = 220

@title_window = Xtitle.new
@title_window.x = 640
@title_window.y = 384
@title_window.opacity = 220

Graphics.transition

loop do
Graphics.update
Input.update
update

if $scene != self
break
end
end

Graphics.freeze
@command_window.dispose
@playtime_window.dispose
@gold_window.dispose
@status_window.dispose
@location_window.dispose
@realtime_window.dispose
@spriteset.dispose
@title_window.dispose
end

def update
@command_window.update
if @command_window.x < 0
@command_window.x += 40
end
@playtime_window.update
if @playtime_window.x > 400
@playtime_window.x -= 40
end
@gold_window.update
if @realtime_window.x == 400
if @gold_window.x > 400
@gold_window.x -= 40
end
end
@status_window.update
if @status_window.x < 0
@status_window.x += 20
end
@location_window.update
if @playtime_window.x == 400
if @location_window.x > 400
@location_window.x -= 40
```


RPG-LIVE.NEUF.FR

```
end
end
@realtime_window.update
if @location_window.x == 400
if @realtime_window.x > 400
@realtime_window.x -= 40
end
end
@title_window.update
if @gold_window.x == 400
if @title_window.x > 400
@title_window.x -= 40
end
end
if @title_window.x == 400
if @command_window.opacity > 180
@command_window.opacity -= 20
end
if @playtime_window.opacity > 180
@playtime_window.opacity -= 20
end
if @location_window.opacity > 180
@location_window.opacity -= 20
end
if @realtime_window.opacity > 180
@realtime_window.opacity -= 20
end
if @gold_window.opacity > 180
@gold_window.opacity -= 20
end
if @title_window.opacity > 180
@title_window.opacity -= 20
end
if @status_window.opacity > 180
@status_window.opacity -= 20
end
end
if @command_window.active
update_command
return
end

if @status_window.active
update_status
return
end
end

def update_command

if Input.trigger?(Input::B)

$game_system.se_play($data_system.cancel_se)
$scene = Scene_Map.new
return
end

if Input.trigger?(Input::C)
if $game_party.actors.size == 0 and @command_window.index < 4
$game_system.se_play($data_system.buzzer_se)
return
end

case @command_window.index
when 0
$game_system.se_play($data_system.decision_se)
$scene = Scene_Item2.new
when 1
$game_system.se_play($data_system.decision_se)
@command_window.active = false
@status_window.active = true
@status_window.index = 0
when 2
$game_system.se_play($data_system.decision_se)
```

```
@command_window.active = false
@status_window.active = true
@status_window.index = 0
when 3
$game_system.se_play($data_system.decision_se)
@command_window.active = false
@status_window.active = true
@status_window.index = 0
when 4
if $game_system.save_disabled
$game_system.se_play($data_system.buzzer_se)
return
end
$game_system.se_play($data_system.decision_se)
$scene = Scene_Save.new
when 5
$game_system.se_play($data_system.decision_se)
$scene = Scene_End2.new
end
return
end
end

def update_status
if Input.trigger?(Input::B)
$game_system.se_play($data_system.cancel_se)
@command_window.active = true
@status_window.active = false
@status_window.index = -1
return
end

if Input.trigger?(Input::C)
case @command_window.index
when 1
if $game_party.actors[@status_window.index].restriction >= 2
$game_system.se_play($data_system.buzzer_se)
return
end
$game_system.se_play($data_system.decision_se)
$scene = XskillScene.new(@status_window.index)
when 2
$game_system.se_play($data_system.decision_se)
$scene = Scene_Equip2.new(@status_window.index)
when 3
$game_system.se_play($data_system.decision_se)
$scene = Scene_Status2.new(@status_window.index)
end
return
end
end
end

class XPlayTime < Window_Base

def initialize
super(0, 0, 240, 96)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
refresh
end

def refresh
self.contents.clear
self.contents.font.color = system_color
self.contents.draw_text(4, 0, 120, 32, "Temps")
@total_sec = Graphics.frame_count / Graphics.frame_rate
hour = @total_sec / 60 / 60
min = @total_sec / 60 % 60
sec = @total_sec % 60
text = sprintf("%02d:%02d:%02d", hour, min, sec)
self.contents.font.color = normal_color
self.contents.draw_text(4, 32, 120, 32, text, 2)
```

```
end

def update
super
if Graphics.frame_count / Graphics.frame_rate != @total_sec
refresh
end
end
end


class XGold < Window_Base

def initialize
super(0, 0, 240, 64)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
refresh
end

def refresh
self.contents.clear
cx = contents.text_size($data_system.words.gold).width
self.contents.font.color = normal_color
self.contents.draw_text(4, 0, 120-cx-2, 32, $game_party.gold.to_s, 2)
self.contents.font.color = system_color
self.contents.draw_text(124-cx, 0, cx, 32, $data_system.words.gold, 2)
end
end


class Xtitle < Window_Base

def initialize
super(0, 0, 240, 96)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = "Times New Roman"
self.contents.font.size = 30
refresh
end

def refresh
self.contents.clear
self.contents.font.name = "Times New Roman"
self.contents.font.color = normal_color
self.contents.font.size = 30
self.contents.draw_text(4, 0, 120, 32, "Ton titre ici")
self.contents.font.name = $fontface
self.contents.font.color = system_color
self.contents.font.size = 14
self.contents.draw_text(4, 0, 120, 100, "Appuiez que [Echap]")
end

def update
super
refresh
end
end

class Xtime < Window_Base

def initialize
super(0, 0, 240, 96)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
refresh
end

def refresh
self.contents.clear
self.contents.font.color = system_color
self.contents.font.size = $fontsize
```


RPG-LIVE.NEUF.FR

```
self.contents.draw_text(4, 0, 120, 32, "Temps réel")

@time_string = Time.now

text = @time_string.strftime("%A %H:%M:%S")
self.contents.font.size = 18
self.contents.font.color = normal_color
self.contents.draw_text(4, 32, 120, 32, text, 2)
end
def update
super
refresh
end
end

class Xlocation < Window_Base

def initialize
super(0, 0, 240, 64)
self.contents = Bitmap.new(width - 32, height - 32)
refresh
end

def refresh
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
self.contents.clear
self.contents.font.color = system_color
self.contents.font.color = normal_color
$maps = load_data("Data/MapInfos.rxdata")
@map_id = $game_map.map_id
@currmap = $maps[@map_id].name
self.contents.font.size = 16
self.contents.draw_text(4, 0, 64, 32, @currmap)
end
end

class Xcursor < Window_Base

attr_reader :index
attr_reader :help_window

def initialize(x, y, width, height)
super(x, y, width, height)
@item_max = 1
@column_max = 1
@index = -1
end

def index=(index)
@index = index

if self.active and @help_window != nil
update_help
end

update_cursor_rect
end

def help_window=(help_window)
@help_window = help_window

if self.active and @help_window != nil
update_help
end
end

def update_cursor_rect

row = @index / @column_max
cursor_width = 26
x = @index / @column_max * 90 - self.ox
y = 2
self.cursor_rect.set(x, y, cursor_width, 26)
```


RPG-LIVE.NEUF.FR

```
end

def update
super

if self.active and @item_max > 0 and @index >= 0

if Input.repeat?(Input::RIGHT)
if (@row_max == 1 and Input.trigger?(Input::RIGHT)) or
@index < @item_max - @column_max
$game_system.se_play($data_system.cursor_se)
@index = (@index + @column_max) % @item_max
end
end

if Input.repeat?(Input::LEFT)
if (@row_max == 1 and Input.trigger?(Input::LEFT)) or
@index >= @column_max
$game_system.se_play($data_system.cursor_se)
@index = (@index - @column_max + @item_max) % @item_max
end
end
if Input.repeat?(Input::DOWN)
if @column_max >= 2 and @index < @item_max - 1
$game_system.se_play($data_system.cursor_se)
@index += 1
end
end

if Input.repeat?(Input::UP)
if @column_max >= 2 and @index > 0
$game_system.se_play($data_system.cursor_se)
@index -= 1
end
end

if Input.repeat?(Input::R)
if self.top_row + (self.page_row_max - 1) < (self.row_max - 1)
$game_system.se_play($data_system.cursor_se)
@index = [@index + self.page_item_max, @item_max - 1].min
self.top_row += self.page_row_max
end
end

if Input.repeat?(Input::L)
if self.top_row > 0
$game_system.se_play($data_system.cursor_se)
@index = [@index - self.page_item_max, 0].max
self.top_row -= self.page_row_max
end
end
end

if self.active and @help_window != nil
update_help
end

update_cursor_rect
end
end

class Xcommand < Xcursor

def initialize(width, commands)

super(0, 0, width, 64)
@item_max = commands.size
@commands = commands
row_max = 1
column_max = 7
self.contents = Bitmap.new(width - 32, @item_max - 32)
self.contents.font.name = $fontface
self.contents.font.size = 12
refresh
```


RPG-LIVE.NEUF.FR

```
self.index = 0
end

def refresh
self.contents.clear
for i in 0...@item_max
draw_item(i, normal_color)
end
end

def draw_item(index, color)
self.contents.font.color = color
rect = Rect.new(90 * index + 29, -2, self.contents.width - 8, 30)
self.contents.fill_rect(rect, Color.new(0, 0, 0, 0))
bitmap1 = RPG::Cache.icon('034-Item03')
bitmap2 = RPG::Cache.icon('045-Skill02')
bitmap3 = RPG::Cache.icon('013-Body01')
bitmap4 = RPG::Cache.icon('040-Item09')
bitmap5 = RPG::Cache.icon('038-Item07')
bitmap6 = RPG::Cache.icon('048-Skill05')
self.contents.blt(0, 2, bitmap1, Rect.new(0, 0, 24, 24), opacity)
self.contents.blt(90, 2, bitmap2, Rect.new(0, 0, 24, 24), opacity)
self.contents.blt(180, 2, bitmap3, Rect.new(0, 0, 24, 24), opacity)
self.contents.blt(270, 2, bitmap4, Rect.new(0, 0, 24, 24), opacity)
self.contents.blt(360, 2, bitmap5, Rect.new(0, 0, 24, 24), opacity)
self.contents.blt(450, 2, bitmap6, Rect.new(0, 0, 24, 24), opacity)
self.contents.draw_text(rect, @commands[index])
end

def disable_item(index)
draw_item(index, disabled_color)
end
end

class Game_Actor < Game_Battler
def now_exp
return @exp - @exp_list[@level]
end
def next_exp
return @exp_list[@level+1] > 0 ? @exp_list[@level+1] - @exp_list[@level] : 0
end
end

class Window_BaseX < Window_Base

alias :draw_actor_hp_original :draw_actor_hp
def draw_actor_hp(actor, x, y, width = 144)
if actor.maxhp != 0
rate = actor.hp.to_f / actor.maxhp
else
rate = 0
end

plus_x = 0
rate_x = 0
plus_y = 25
plus_width = 0
rate_width = 100
height = 10
align1 = 1
align2 = 2
align3 = 0
grade1 = 1
grade2 = 0
color1 = Color.new(0, 0, 0, 192)
color2 = Color.new(255, 255, 192, 192)
color3 = Color.new(0, 0, 0, 192)
color4 = Color.new(64, 0, 0, 192)
color5 = Color.new(80 - 24 * rate, 80 * rate, 14 * rate, 192)
color6 = Color.new(240 - 72 * rate, 240 * rate, 62 * rate, 192)

if actor.maxhp != 0
hp = (width + plus_width) * actor.hp * rate_width / 100 / actor.maxhp
else
```


RPG-LIVE.NEUF.FR

```
hp = 0
end

gauge_rect(x + plus_x + width * rate_x / 100, y + plus_y,
width, plus_width + width * rate_width / 100,
height, hp, align1, align2, align3,
color1, color2, color3, color4, color5, color6, grade1, grade2)
draw_actor_hp_original(actor, x, y, width)
end

alias :draw_actor_sp_original :draw_actor_sp
def draw_actor_sp(actor, x, y, width = 144)

if actor.maxsp != 0
rate = actor.sp.to_f / actor.maxsp
else
rate = 1
end

plus_x = 0
rate_x = 0
plus_y = 25
plus_width = 0
rate_width = 100
height = 10
align1 = 1
align2 = 2
align3 = 0
grade1 = 1
grade2 = 0
color1 = Color.new(0, 0, 0, 192)
color2 = Color.new(255, 255, 192, 192)
color3 = Color.new(0, 0, 0, 192)
color4 = Color.new(0, 64, 0, 192)
color5 = Color.new(14 * rate, 80 - 24 * rate, 80 * rate, 192)
color6 = Color.new(62 * rate, 240 - 72 * rate, 240 * rate, 192)

if actor.maxsp != 0
sp = (width + plus_width) * actor.sp * rate_width / 100 / actor.maxsp
else
sp = (width + plus_width) * rate_width / 100
end

gauge_rect(x + plus_x + width * rate_x / 100, y + plus_y,
width, plus_width + width * rate_width / 100,
height, sp, align1, align2, align3,
color1, color2, color3, color4, color5, color6, grade1, grade2)
draw_actor_sp_original(actor, x, y, width)
end

alias :draw_actor_exp_original :draw_actor_exp
def draw_actor_exp(actor, x, y, width = 204)

if actor.next_exp != 0
rate = actor.now_exp.to_f / actor.next_exp
else
rate = 1
end

plus_x = 0
rate_x = 0
plus_y = 25
plus_width = 0
rate_width = 100
height = 10
align1 = 1
align2 = 2
align3 = 0
grade1 = 1
grade2 = 0

color1 = Color.new(0, 0, 0, 192)
color2 = Color.new(255, 255, 192, 192)
color3 = Color.new(0, 0, 0, 192)
```


RPG-LIVE.NEUF.FR

```
color4 = Color.new(64, 0, 0, 192)
color5 = Color.new(80 * rate, 80 - 80 * rate ** 2, 80 - 80 * rate, 192)
color6 = Color.new(240 * rate, 240 - 240 * rate ** 2, 240 - 240 * rate, 192)

if actor.next_exp != 0
exp = (width + plus_width) * actor.now_exp * rate_width /
100 / actor.next_exp
else
exp = (width + plus_width) * rate_width / 100
end

gauge_rect(x + plus_x + width * rate_x / 100, y + plus_y,
width, plus_width + width * rate_width / 100,
height, exp, align1, align2, align3,
color1, color2, color3, color4, color5, color6, grade1, grade2)
draw_actor_exp_original(actor, x, y)
end

def gauge_rect(x, y, rect_width, width, height, gauge, align1, align2, align3,
color1, color2, color3, color4, color5, color6, grade1, grade2)
case align1
when 1
x += (rect_width - width) / 2
when 2
x += rect_width - width
end
case align2
when 1
y -= height / 2
when 2
y -= height
end

self.contents.fill_rect(x, y, width, height, color1)
self.contents.fill_rect(x + 1, y + 1, width - 2, height - 2, color2)
if align3 == 0
if grade1 == 2
grade1 = 3
end
if grade2 == 2
grade2 = 3
end
end
if (align3 == 1 and grade1 == 0) or grade1 > 0
color = color3
color3 = color4
color4 = color
end
if (align3 == 1 and grade2 == 0) or grade2 > 0
color = color5
color5 = color6
color6 = color
end

self.contents.gradation_rect(x + 2, y + 2, width - 4, height - 4,
color3, color4, grade1)
if align3 == 1
x += width - gauge
end

self.contents.gradation_rect(x + 2, y + 2, gauge - 4, height - 4,
color5, color6, grade2)
end
end

class Bitmap

def gradation_rect(x, y, width, height, color1, color2, align = 0)
if align == 0
for i in x...x + width
red = color1.red + (color2.red - color1.red) * (i - x) / (width - 1)
green = color1.green +
(color2.green - color1.green) * (i - x) / (width - 1)
blue = color1.blue +
```


RPG-LIVE.NEUF.FR

```
(color2.blue - color1.blue) * (i - x) / (width - 1)
alpha = color1.alpha +
(color2.alpha - color1.alpha) * (i - x) / (width - 1)
color = Color.new(red, green, blue, alpha)
fill_rect(i, y, 1, height, color)
end
elsif align == 1
for i in y...y + height
red = color1.red +
(color2.red - color1.red) * (i - y) / (height - 1)
green = color1.green +
(color2.green - color1.green) * (i - y) / (height - 1)
blue = color1.blue +
(color2.blue - color1.blue) * (i - y) / (height - 1)
alpha = color1.alpha +
(color2.alpha - color1.alpha) * (i - y) / (height - 1)
color = Color.new(red, green, blue, alpha)
fill_rect(x, i, width, 1, color)
end

elsif align == 2
for i in x...x + width
for j in y...y + height
red = color1.red + (color2.red - color1.red) *
((i - x) / (width - 1.0) + (j - y) / (height - 1.0)) / 2
green = color1.green + (color2.green - color1.green) *
((i - x) / (width - 1.0) + (j - y) / (height - 1.0)) / 2
blue = color1.blue + (color2.blue - color1.blue) *
((i - x) / (width - 1.0) + (j - y) / (height - 1.0)) / 2
alpha = color1.alpha + (color2.alpha - color1.alpha) *
((i - x) / (width - 1.0) + (j - y) / (height - 1.0)) / 2
color = Color.new(red, green, blue, alpha)
set_pixel(i, j, color)
end
end

elsif align == 3
for i in x...x + width
for j in y...y + height
red = color1.red + (color2.red - color1.red) *
((x + width - i) / (width - 1.0) + (j - y) / (height - 1.0)) / 2
green = color1.green + (color2.green - color1.green) *
((x + width - i) / (width - 1.0) + (j - y) / (height - 1.0)) / 2
blue = color1.blue + (color2.blue - color1.blue) *
((x + width - i) / (width - 1.0) + (j - y) / (height - 1.0)) / 2
alpha = color1.alpha + (color2.alpha - color1.alpha) *
((x + width - i) / (width - 1.0) + (j - y) / (height - 1.0)) / 2
color = Color.new(red, green, blue, alpha)
set_pixel(i, j, color)
end
end
end
end
end

module RPG
class Sprite < ::Sprite
def damage(value, critical)
dispose_damage
if value.is_a?(Numeric)
damage_string = value.abs.to_s
else
damage_string = value.to_s
end

bitmap = Bitmap.new(160, 48)
bitmap.font.name = "Arial Black"
bitmap.font.size = 32
bitmap.font.color.set(0, 0, 0)
bitmap.draw_text(-1, 12-1, 160, 36, damage_string, 1)
bitmap.draw_text(+1, 12-1, 160, 36, damage_string, 1)
bitmap.draw_text(-1, 12+1, 160, 36, damage_string, 1)
bitmap.draw_text(+1, 12+1, 160, 36, damage_string, 1)
```


RPG-LIVE.NEUF.FR

```
if value.is_a?(Numeric) and value < 0
bitmap.font.color.set(176, 255, 144)
else
bitmap.font.color.set(255, 255, 255)
end
bitmap.draw_text(0, 12, 160, 36, damage_string, 1)
if critical
bitmap.font.size = 20
bitmap.font.color.set(0, 0, 0)
bitmap.draw_text(-1, -1, 160, 20, "CRITIQUE", 1)
bitmap.draw_text(+1, -1, 160, 20, "CRITIQUE", 1)
bitmap.draw_text(-1, +1, 160, 20, "CRITIQUE", 1)
bitmap.draw_text(+1, +1, 160, 20, "CRITIQUE", 1)
bitmap.font.color.set(255, 255, 255)
bitmap.draw_text(0, 0, 160, 20, "CRITIQUE", 1)
end
@_damage_sprite = ::Sprite.new
@_damage_sprite.bitmap = bitmap
@_damage_sprite.ox = 80 + self.viewport.ox
@_damage_sprite.oy = 20 + self.viewport.oy
@_damage_sprite.x = self.x + self.viewport.rect.x
@_damage_sprite.y = self.y - self.oy / 2 + self.viewport.rect.y
@_damage_sprite.z = 3000
@_damage_duration = 40
end
def animation(animation, hit)
dispose_animation
@_animation = animation
return if @_animation == nil
@_animation_hit = hit
@_animation_duration = @_animation.frame_max
animation_name = @_animation.animation_name
animation_hue = @_animation.animation_hue
bitmap = RPG::Cache.animation(animation_name, animation_hue)
if @@_reference_count.include?(bitmap)
@@_reference_count[bitmap] += 1
else
@@_reference_count[bitmap] = 1
end
@_animation_sprites = []
if @_animation.position != 3 or not @@_animations.include?(animation)
for i in 0..15
sprite = ::Sprite.new
sprite.bitmap = bitmap
sprite.visible = false
@_animation_sprites.push(sprite)
end
unless @@_animations.include?(animation)
@@_animations.push(animation)
end
end
update_animation
end
def loop_animation(animation)
return if animation == @_loop_animation
dispose_loop_animation
@_loop_animation = animation
return if @_loop_animation == nil
@_loop_animation_index = 0
animation_name = @_loop_animation.animation_name
animation_hue = @_loop_animation.animation_hue
bitmap = RPG::Cache.animation(animation_name, animation_hue)
if @@_reference_count.include?(bitmap)
@@_reference_count[bitmap] += 1
else
@@_reference_count[bitmap] = 1
end
@_loop_animation_sprites = []
for i in 0..15
sprite = ::Sprite.new
sprite.bitmap = bitmap
sprite.visible = false
@_loop_animation_sprites.push(sprite)
end
```


RPG-LIVE.NEUF.FR

```
update_loop_animation
end
def animation_set_sprites(sprites, cell_data, position)
for i in 0..15
sprite = sprites[i]
pattern = cell_data[i, 0]
if sprite == nil or pattern == nil or pattern == -1
sprite.visible = false if sprite != nil
next
end

sprite.visible = true
sprite.src_rect.set(pattern % 5 * 192, pattern / 5 * 192, 192, 192)
if position == 3
if self.viewport != nil
sprite.x = self.viewport.rect.width / 2
sprite.y = self.viewport.rect.height - 160
else
sprite.x = 320
sprite.y = 240
end
else

sprite.x = self.x + self.viewport.rect.x -
self.ox + self.src_rect.width / 2
sprite.y = self.y + self.viewport.rect.y -
self.oy + self.src_rect.height / 2
sprite.y -= self.src_rect.height / 4 if position == 0
sprite.y += self.src_rect.height / 4 if position == 2
end

sprite.x += cell_data[i, 1]
sprite.y += cell_data[i, 2]
sprite.z = 2000
sprite.ox = 96
sprite.oy = 96
sprite.zoom_x = cell_data[i, 3] / 100.0
sprite.zoom_y = cell_data[i, 3] / 100.0
sprite.angle = cell_data[i, 4]
sprite.mirror = (cell_data[i, 5] == 1)
sprite.opacity = cell_data[i, 6] * self.opacity / 255.0
sprite.blend_type = cell_data[i, 7]
end
end
end
end

class Xstatusselect < Window_BaseX
attr_reader :index
attr_reader :help_window

def initialize(x, y, width, height)
super(x, y, width, height)
@item_max = 1
@column_max = 1
@index = -1
end

def index=(index)
@index = index

if self.active and @help_window != nil
update_help
end
update_cursor_rect
end

def row_max
return (@item_max + @column_max - 1) / @column_max
end

def top_row

return self.oy / 32
```


RPG-LIVE.NEUF.FR

```
end

def top_row=(row)

if row < 0
row = 0
end

if row > row_max - 1
row = row_max - 1
end

self.oy = row * 32
end

def page_row_max

return (self.height - 32) / 32
end

def page_item_max

return page_row_max * @column_max
end

def help_window=(help_window)
@help_window = help_window

if self.active and @help_window != nil
update_help
end
end

def update_cursor_rect

if @index < 0
self.cursor_rect.empty
return
end

row = @index / @column_max

if row < self.top_row
self.top_row = row
end

if row > self.top_row + (self.page_row_max - 1)
self.top_row = row - (self.page_row_max - 1)
end

cursor_width = 32
x = @index % @column_max * (cursor_width + 32)
y = @index / @column_max * 32 - self.oy
self.cursor_rect.set(x, y, cursor_width, 32)
end

def update
super

if self.active and @item_max > 0 and @index >= 0
if Input.repeat?(Input::DOWN)
if (@column_max == 1 and Input.trigger?(Input::DOWN)) or
@index < @item_max - @column_max
$game_system.se_play($data_system.cursor_se)
@index = (@index + @column_max) % @item_max
end
end

if Input.repeat?(Input::UP)

if (@column_max == 1 and Input.trigger?(Input::UP)) or
@index >= @column_max
$game_system.se_play($data_system.cursor_se)
@index = (@index - @column_max + @item_max) % @item_max
```


RPG-LIVE.NEUF.FR

```
end
end

if Input.repeat?(Input::RIGHT)
if @column_max >= 2 and @index < @item_max - 1
$game_system.se_play($data_system.cursor_se)
@index += 1
end
end

if Input.repeat?(Input::LEFT)

if @column_max >= 2 and @index > 0
$game_system.se_play($data_system.cursor_se)
@index -= 1
end
end

if Input.repeat?(Input::R)
if self.top_row + (self.page_row_max - 1) < (self.row_max - 1)
$game_system.se_play($data_system.cursor_se)
@index = [@index + self.page_item_max, @item_max - 1].min
self.top_row += self.page_row_max
end
end

if Input.repeat?(Input::L)
if self.top_row > 0
$game_system.se_play($data_system.cursor_se)
@index = [@index - self.page_item_max, 0].max
self.top_row -= self.page_row_max
end
end
end

if self.active and @help_window != nil
update_help
end

update_cursor_rect
end
end

class Xstatus < Xstatusselect

def initialize
super(0, 0, 400, 416)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
refresh
self.active = false
self.index = -1
end

def refresh
self.contents.clear
@item_max = $game_party.actors.size
for i in 0...$game_party.actors.size
x = 64
y = i * 90
actor = $game_party.actors[i]
draw_actor_face(actor, x + 200, y + 79)
draw_actor_graphic(actor, x - 50, y + 80)
self.contents.font.size = 18
draw_actor_name(actor, x - 60, y + 4)
self.contents.font.color = system_color
self.contents.draw_text(x - 5, y + 2, 120, 32, '-')
self.contents.font.color = normal_color
draw_actor_class(actor, x + 75, y + 4)
draw_actor_level(actor, x + 5, y + 4)
draw_actor_state(actor, x + 135, y + 4)
draw_actor_exp(actor, x - 35, y + 54)
draw_actor_hp(actor, x - 35, y + 32)
```


RPG-LIVE.NEUF.FR

```
draw_actor_sp(actor, x + 115, y + 32)
end
end

def update_cursor_rect
if @index < 0
self.cursor_rect.empty
else
self.cursor_rect.set(1, @index * 90 + 33, 26, 48)
end
end

def draw_actor_face(actor, x, y)
face = RPG::Cache.battler(actor.character_name, actor.character_hue)
fw = face.width
fh = 90
src_rect = Rect.new(3, -1, fw, fh)
opacity = 180
self.contents.blt(x - fw / 23, y - fh, face, src_rect, opacity)
end
end

class Window_HelpItemSkill < Window_Base

def initialize
super(0, 0, 320, 64)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
end

def set_text(text, align = 0)

if text != @text or align != @align

self.contents.clear
self.contents.font.color = normal_color
self.contents.font.size = 18
self.contents.draw_text(4, 0, self.width - 40, 32, text, align)
@text = text
@align = align
@actor = nil
end
self.visible = true
end

def set_actor(actor)
if actor != @actor
self.contents.clear
draw_actor_name(actor, 4, 0)
draw_actor_state(actor, 140, 0)
draw_actor_hp(actor, 284, 0)
draw_actor_sp(actor, 460, 0)
@actor = actor
@text = nil
self.visible = true
end
end

def set_enemy(enemy)
text = enemy.name
state_text = make_battler_state_text(enemy, 112, false)
if state_text != ""
text += " " + state_text
end
set_text(text, 1)
end
end

class Xskill < Window_Selectable

def initialize(actor)
super(0, 128, 320, 352)
@actor = actor
```


RPG-LIVE.NEUF.FR

```
@column_max = 1
refresh
self.index = 0

if $game_temp.in_battle
self.y = 64
self.height = 256
self.back_opacity = 160
end
end

def skill
return @data[self.index]
end

def refresh
if self.contents != nil
self.contents.dispose
self.contents = nil
end
@data = []
for i in 0...@actor.skills.size
skill = $data_skills[@actor.skills[i]]
if skill != nil
@data.push(skill)
end
end

@item_max = @data.size
if @item_max > 0
self.contents = Bitmap.new(width - 32, row_max * 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
for i in 0...@item_max
draw_item(i)
end
end
end

def draw_item(index)
skill = @data[index]
if @actor.skill_can_use?(skill.id)
self.contents.font.color = normal_color
else
self.contents.font.color = disabled_color
end
x = 4 + index % 1 * (288 + 32)
y = index / 1 * 32
rect = Rect.new(x, y, self.width / @column_max - 32, 32)
self.contents.fill_rect(rect, Color.new(0, 0, 0, 0))
bitmap = RPG::Cache.icon(skill.icon_name)
opacity = self.contents.font.color == normal_color ? 255 : 128
self.contents.blt(x, y + 4, bitmap, Rect.new(0, 0, 24, 24), opacity)
self.contents.draw_text(x + 28, y, 204, 32, skill.name, 0)
self.contents.draw_text(x + 232, y, 48, 32, skill.sp_cost.to_s, 2)
end

def update_help
@help_window.set_text(self.skill == nil ? "" : self.skill.description)
end
end

class Window_SkillStatus2 < Window_Base

def initialize(actor)
super(0, 64, 320, 64)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = $fontface
self.contents.font.size = 18
@actor = actor
refresh
end

def refresh
```


RPG-LIVE.NEUF.FR

```
self.contents.clear
draw_actor_name(@actor, 4, 0)
draw_actor_state(@actor, 60, 0)
draw_actor_sp(@actor, 130, 0)
end
end

class Window_Item2 < Window_Selectable

def initialize
super(0, 64, 320, 416)
@column_max = 1
refresh
self.index = 0

if $game_temp.in_battle
self.y = 64
self.height = 256
self.back_opacity = 160
end
end

def item
return @data[self.index]
end

def refresh
if self.contents != nil
self.contents.dispose
self.contents = nil
end
@data = []

for i in 1...$data_items.size
if $game_party.item_number(i) > 0
@data.push($data_items[i])
end
end

unless $game_temp.in_battle
for i in 1...$data_weapons.size
if $game_party.weapon_number(i) > 0
@data.push($data_weapons[i])
end
end

for i in 1...$data_armors.size
if $game_party.armor_number(i) > 0
@data.push($data_armors[i])
end
end
end

@item_max = @data.size
if @item_max > 0
self.contents = Bitmap.new(width - 32, row_max * 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
for i in 0...@item_max
draw_item(i)
end
end
end

def draw_item(index)
item = @data[index]
case item
when RPG::Item
number = $game_party.item_number(item.id)
when RPG::Weapon
number = $game_party.weapon_number(item.id)
when RPG::Armor
number = $game_party.armor_number(item.id)
end
```


RPG-LIVE.NEUF.FR

```
if item.is_a?(RPG::Item) and
$game_party.item_can_use?(item.id)
self.contents.font.color = normal_color
else
self.contents.font.color = disabled_color
end
x = 4 + index % 1 * (288 + 32)
y = index / 1 * 32
rect = Rect.new(x, y, self.width / @column_max - 32, 32)
self.contents.fill_rect(rect, Color.new(0, 0, 0, 0))
bitmap = RPG::Cache.icon(item.icon_name)
opacity = self.contents.font.color == normal_color ? 255 : 128
self.contents.blt(x, y + 4, bitmap, Rect.new(0, 0, 24, 24), opacity)
self.contents.draw_text(x + 28, y, 212, 32, item.name, 0)
self.contents.draw_text(x + 240, y, 16, 32, ":", 1)
self.contents.draw_text(x + 256, y, 24, 32, number.to_s, 2)
end

def update_help
@help_window.set_text(self.item == nil ? "" : self.item.description)
end
end

class XskillScene

def initialize(actor_index = 0, equip_index = 0)
@actor_index = actor_index
end

def main
@spriteset = Spriteset_Map.new

@actor = $game_party.actors[@actor_index]
@help_window = Window_HelpItemSkill.new
@help_window.back_opacity = 160
@help_window.x = -320
@status_window = Window_SkillStatus2.new(@actor)
@status_window.back_opacity = 160
@status_window.x = -320
@skill_window = Xskill.new(@actor)
@skill_window.back_opacity = 160
@skill_window.help_window = @help_window
@skill_window.x = -320
@target_window = Window_Target2.new
@target_window.visible = false
@target_window.active = false
@target_window.back_opacity = 160
@target_window.y = 480

Graphics.transition

loop do
Graphics.update
Input.update
update
if $scene != self
break
end
end

Graphics.freeze
@help_window.dispose
@status_window.dispose
@skill_window.dispose
@target_window.dispose
@spriteset.dispose
end

def update
@help_window.update
if @help_window.x < 0
@help_window.x += 10
end
```

```
@status_window.update
if @status_window.x < 0
@status_window.x += 10
end
@skill_window.update
if @skill_window.x < 0
@skill_window.x += 10
end
@target_window.update
if @target_window.y > 0
@target_window.y -= 15
end
if @skill_window.active
update_skill
return
end

if @target_window.active
update_target
return
end
end

def update_skill
if Input.trigger?(Input::B)
$game_system.se_play($data_system.cancel_se)
$scene = Scene_Menu.new(1)
return
end

if Input.trigger?(Input::C)
@skill = @skill_window.skill
if @skill == nil or not @actor.skill_can_use?(@skill.id)
$game_system.se_play($data_system.buzzer_se)
return
end

$game_system.se_play($data_system.decision_se)
if @skill.scope >= 3
@skill_window.active = false
@target_window.x = 320
@target_window.y = 480
@target_window.visible = true
@target_window.active = true

if @skill.scope == 4 || @skill.scope == 6
@target_window.index = -1
elsif @skill.scope == 7
@target_window.index = @actor_index - 10
else
@target_window.index = 0
end
else
if @skill.common_event_id > 0
$game_temp.common_event_id = @skill.common_event_id
$game_system.se_play(@skill.menu_se)
@actor.sp -= @skill.sp_cost
@status_window.refresh
@skill_window.refresh
@target_window.refresh
$scene = Scene_Map.new
return
end
end
return
end

if Input.trigger?(Input::R)
$game_system.se_play($data_system.cursor_se)
@actor_index += 1
@actor_index %= $game_party.actors.size
$scene = Scene_Skill.new(@actor_index)
return
end
```


RPG-LIVE.NEUF.FR

```
if Input.trigger?(Input::L)
$game_system.se_play($data_system.cursor_se)
@actor_index += $game_party.actors.size - 1
@actor_index %= $game_party.actors.size
$scene = Scene_Skill.new(@actor_index)
return
end
end

def update_target
if Input.trigger?(Input::B)
$game_system.se_play($data_system.cancel_se)
@skill_window.active = true
@target_window.visible = false
@target_window.active = false
if @skill_window.active = true
@target_window.y = 480
end
return
end

if Input.trigger?(Input::C)
unless @actor.skill_can_use?(@skill.id)
$game_system.se_play($data_system.buzzer_se)
return
end

if @target_window.index == -1
used = false
for i in $game_party.actors
used |= i.skill_effect(@actor, @skill)
end
end

if @target_window.index <= -2
target = $game_party.actors[@target_window.index + 10]
used = target.skill_effect(@actor, @skill)
end

if @target_window.index >= 0
target = $game_party.actors[@target_window.index]
used = target.skill_effect(@actor, @skill)
end

if used
$game_system.se_play(@skill.menu_se)
@actor.sp -= @skill.sp_cost
@status_window.refresh
@skill_window.refresh
@target_window.refresh

if $game_party.all_dead?
$scene = Scene_Gameover.new
return
end

if @skill.common_event_id > 0
$game_temp.common_event_id = @skill.common_event_id
$scene = Scene_Map.new
return
end
end

unless used
$game_system.se_play($data_system.buzzer_se)
end
return
end
end
end

class Window_Target2 < Window_Selectable
```


RPG-LIVE.NEUF.FR

```
def initialize
super(0, 0, 320, 400)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = $fontface
self.contents.font.size = 18
self.z += 10
@item_max = $game_party.actors.size
refresh
end

def refresh
self.contents.clear
for i in 0...$game_party.actors.size
x = 4
y = i * 90
actor = $game_party.actors[i]
draw_actor_face(actor, x, y + 70)
draw_actor_face2(actor, x + 200, y + 95)
draw_actor_name(actor, x + 40, y)
draw_actor_class(actor, x + 200, y + 32)
draw_actor_level(actor, x + 200, y + 64)
draw_actor_state(actor, x + 100, y)
draw_actor_hp(actor, x + 40, y + 32)
draw_actor_sp(actor, x + 40, y + 64)
end
end

def draw_actor_face(actor, x, y)
bitmap = RPG::Cache.character(actor.character_name, actor.character_hue)
cw = bitmap.rect.width / 4
ch = bitmap.rect.height / 4
src_rect = Rect.new(0, 0, cw, ch)
self.contents.blt(x - cw / 23, y - ch, bitmap, src_rect)
end

def draw_actor_face2(actor, x, y)
face = RPG::Cache.battler(actor.character_name, actor.character_hue)
fw = face.width
fh = 90
src_rect = Rect.new(3, -1, fw, fh)
opacity = 160
self.contents.blt(x - fw / 23, y - fh, face, src_rect, opacity)
end

def update_cursor_rect

if @index <= -2
self.cursor_rect.set(0, (@index + 10) * 90, self.width - 32, 96)
elsif @index == -1
self.cursor_rect.set(0, 0, self.width - 32, @item_max * 90 - 20)
else
self.cursor_rect.set(0, @index * 90 + 5, self.width - 32, 90)
end
end
end

class Scene_Item2

def main
@spriteset = Spriteset_Map.new
@help_window = Window_HelpItemSkill.new
@help_window.back_opacity = 160
@help_window.x = -320
@item_window = Window_Item2.new
@item_window.back_opacity = 160
@item_window.help_window = @help_window
@item_window.x = -320
@target_window = Window_Target2.new
@target_window.visible = false
@target_window.active = false
@target_window.back_opacity = 160
@target_window.y = 480
```


RPG-LIVE.NEUF.FR

```
Graphics.transition
loop do
Graphics.update
Input.update
update
if $scene != self
break
end
end

Graphics.freeze
@help_window.dispose
@item_window.dispose
@target_window.dispose
@spriteset.dispose
end

def update
@help_window.update
if @help_window.x < 0
@help_window.x +=10
end
@item_window.update
if @item_window.x < 0
@item_window.x +=10
end
@target_window.update
if @target_window.y > 0
@target_window.y -= 15
end
if @item_window.active
update_item
return
end

if @target_window.active
update_target
return
end
end

def update_item

if Input.trigger?(Input::B)
$game_system.se_play($data_system.cancel_se)
$scene = Scene_Menu.new(0)
return
end

if Input.trigger?(Input::C)
@item = @item_window.item
unless @item.is_a?(RPG::Item)
$game_system.se_play($data_system.buzzer_se)
return
end

unless $game_party.item_can_use?(@item.id)
$game_system.se_play($data_system.buzzer_se)
return
end
$game_system.se_play($data_system.decision_se)
if @item.scope >= 3
@item_window.active = false
@target_window.x = 320
@target_window.y = 480
@target_window.visible = true
@target_window.active = true

if @item.scope == 4 || @item.scope == 6
@target_window.index = -1
else
@target_window.index = 0
end
else
```


RPG-LIVE.NEUF.FR

```
if @item.common_event_id > 0
$game_temp.common_event_id = @item.common_event_id
$game_system.se_play(@item.menu_se)

if @item.consumable
$game_party.lose_item(@item.id, 1)
@item_window.draw_item(@item_window.index)
end

$scene = Scene_Map.new
return
end
end
return
end
end

def update_target

if Input.trigger?(Input::B)
$game_system.se_play($data_system.cancel_se)
unless $game_party.item_can_use?(@item.id)
@item_window.refresh
end

@item_window.active = true
@target_window.visible = false
@target_window.active = false
@target_window.y = 480
return
end

if Input.trigger?(Input::C)
if $game_party.item_number(@item.id) == 0
$game_system.se_play($data_system.buzzer_se)
return
end
if @target_window.index == -1
used = false
for i in $game_party.actors
used |= i.item_effect(@item)
end
end

if @target_window.index >= 0
target = $game_party.actors[@target_window.index]
used = target.item_effect(@item)
end

if used
$game_system.se_play(@item.menu_se)
if @item.consumable
$game_party.lose_item(@item.id, 1)
@item_window.draw_item(@item_window.index)
end

@target_window.refresh
if $game_party.all_dead?
$scene = Scene_Gameover.new
return
end
if @item.common_event_id > 0
$game_temp.common_event_id = @item.common_event_id
$scene = Scene_Map.new
return
end
end

unless used
$game_system.se_play($data_system.buzzer_se)
end
return
end
```


RPG-LIVE.NEUF.FR

```
end
end

class Window_Status2 < Window_Base

def initialize(actor)
super(0, 0, 430, 370)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
@actor = actor
refresh
end

def refresh
self.contents.clear
draw_actor_graphic(@actor, 20, 45)
draw_actor_face(@actor, 270, 250)
draw_actor_name(@actor, 4 + 40, 10)
draw_actor_class(@actor, 4 + 160 + 40, 10)
draw_actor_level(@actor, 96 + 40, 10)
draw_actor_state(@actor, 244+ 40, 10)
self.contents.font.size = 18
draw_actor_hp(@actor, 0, 100, 172)
draw_actor_sp(@actor, 0, 116, 172)
draw_actor_parameter(@actor, 0, 160, 0)
draw_actor_parameter(@actor, 0, 224 - 32, 1)
draw_actor_parameter(@actor, 0, 256 - 32, 2)
draw_actor_parameter(@actor, 0, 304 - 32, 3)
draw_actor_parameter(@actor, 0, 336 - 32, 4)
draw_actor_parameter(@actor, 0 + 200, 304 - 32, 5)
draw_actor_parameter(@actor, 0 + 200, 336 - 32, 6)

self.contents.font.color = system_color
self.contents.draw_text(0, 48, 80, 32, "EXP")
self.contents.draw_text(0, 64, 80, 32, "NEXT")
self.contents.font.color = normal_color
self.contents.draw_text(0 + 80, 48, 84, 32, @actor.exp_s, 2)
self.contents.draw_text(0 + 80, 64, 84, 32, @actor.next_rest_exp_s, 2)
self.contents.font.color = system_color
self.contents.draw_text(200, 48, 96, 32, "Equipment")
draw_item_name($data_weapons[@actor.weapon_id], 200 + 4, 80)
draw_item_name($data_armors[@actor.armor1_id], 200 + 4, 128)
draw_item_name($data_armors[@actor.armor2_id], 200 + 4, 304 - 128)
draw_item_name($data_armors[@actor.armor3_id], 200 + 4, 352 - 128)
draw_item_name($data_armors[@actor.armor4_id], 200 + 4, 400 - 128)
end

def draw_actor_face(actor, x, y)
face = RPG::Cache.battler(actor.character_name, actor.character_hue)
fw = face.width
fh = face.height
src_rect = Rect.new(3, -1, fw, fh)
opacity = 120
self.contents.blt(x - fw / 23, y - fh, face, src_rect, opacity)
end
end

class Scene_Status2
def initialize(actor_index = 0, equip_index = 0)
@actor_index = actor_index
end

def main
@spriteset = Spriteset_Map.new
@actor = $game_party.actors[@actor_index]
@status_window = Window_Status2.new(@actor)
@status_window.back_opacity = 160
@status_window.x = -430
Graphics.transition
loop do
Graphics.update
Input.update
update
```


RPG-LIVE.NEUF.FR

```
if $scene != self
break
end
end

Graphics.freeze
@status_window.dispose
@spriteset.dispose
end

def update
if @status_window.x < 0
@status_window.x +=10
end
if Input.trigger?(Input::B)
$game_system.se_play($data_system.cancel_se)
$scene = Scene_Menu.new(3)
return
end

if Input.trigger?(Input::R)
$game_system.se_play($data_system.cursor_se)
@actor_index += 1
@actor_index %= $game_party.actors.size
$scene = Scene_Status.new(@actor_index)
return
end

if Input.trigger?(Input::L)
$game_system.se_play($data_system.cursor_se)
@actor_index += $game_party.actors.size - 1
@actor_index %= $game_party.actors.size
$scene = Scene_Status.new(@actor_index)
return
end
end
end

class Scene_End2

def main
@spriteset = Spriteset_Map.new
s1 = "To Title"
s2 = "Shutdown"
s3 = "Cancel"
@command_window2 = Window_Command.new(192, [s1, s2, s3])
@command_window2.x = -192
@command_window2.y = 240 - @command_window2.height / 2
@command_window2.back_opacity = 160
Graphics.transition
loop do
Graphics.update
Input.update
update
if $scene != self
break
end
end
Graphics.freeze
@command_window2.dispose
@spriteset.dispose
if $scene.is_a?(Scene_Title)
Graphics.transition
Graphics.freeze
end
end

def update
@command_window2.update

if@command_window2.x < 320 - @command_window2.width / 2
@command_window2.x += 10
end
```


RPG-LIVE.NEUF.FR

```
if Input.trigger?(Input::B)
$game_system.se_play($data_system.cancel_se)
$scene = Scene_Menu.new(8)
return
end
if Input.trigger?(Input::C)
case @command_window2.index
when 0
$game_system.se_play($data_system.decision_se)
Audio.bgm_fade(800)
Audio.bgs_fade(800)
Audio.me_fade(800)
$scene = Scene_Title.new
when 1
$game_system.se_play($data_system.decision_se)
Audio.bgm_fade(800)
Audio.bgs_fade(800)
Audio.me_fade(800)
$scene = nil
when 2
$game_system.se_play($data_system.decision_se)
$scene = Scene_Menu.new(8)
end
return
end
end
end

class Window_EquipLeft2 < Window_Base

def initialize(actor)
super(0, 64, 272, 416)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
@actor = actor
refresh
end

def refresh
self.contents.clear
draw_actor_graphic(@actor, 20, 58)
draw_actor_face(@actor, 136, 360)
draw_actor_name(@actor, 52, 0)
draw_actor_level(@actor, 52, 32)
draw_actor_parameter(@actor, 4, 64, 0)
draw_actor_parameter(@actor, 4, 96, 1)
draw_actor_parameter(@actor, 4, 128, 2)
if @new_atk != nil
self.contents.font.name = "Arial"
self.contents.font.color = system_color
self.contents.draw_text(160, 64, 40, 32, "»", 1)
self.contents.font.color = normal_color
self.contents.draw_text(200, 64, 36, 32, @new_atk.to_s, 2)
end

if @new_pdef != nil
self.contents.font.color = system_color
self.contents.draw_text(160, 96, 40, 32, "»", 1)
self.contents.font.color = normal_color
self.contents.draw_text(200, 96, 36, 32, @new_pdef.to_s, 2)
end
if @new_mdef != nil
self.contents.font.color = system_color
self.contents.draw_text(160, 128, 40, 32, "»", 1)
self.contents.font.color = normal_color
self.contents.draw_text(200, 128, 36, 32, @new_mdef.to_s, 2)
end
end

def set_new_parameters(new_atk, new_pdef, new_mdef)
self.contents.font.name = $fontface
if @new_atk != new_atk or @new_pdef != new_pdef or @new_mdef != new_mdef
@new_atk = new_atk
```


RPG-LIVE.NEUF.FR

```
@new_pdef = new_pdef
@new_mdef = new_mdef
refresh
end
end

def draw_actor_face(actor, x, y)
face = RPG::Cache.battler(actor.character_name, actor.character_hue)
fw = face.width
fh = face.height
src_rect = Rect.new(3, -1, fw, fh)
self.contents.blt(x - fw / 23, y - fh, face, src_rect)
end
end

class Window_EquipRight2 < Window_Selectable

def initialize(actor)
super(272, 64, 368, 192)
self.contents = Bitmap.new(width - 32, height - 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
@actor = actor
refresh

self.index = 0
end

def item
return @data[self.index]
end

def refresh
self.contents.clear
@data = []
@data.push($data_weapons[@actor.weapon_id])
@data.push($data_armors[@actor.armor1_id])
@data.push($data_armors[@actor.armor2_id])
@data.push($data_armors[@actor.armor3_id])
@data.push($data_armors[@actor.armor4_id])
@item_max = @data.size
self.contents.font.color = system_color
self.contents.draw_text(4, 32 * 0, 92, 32, $data_system.words.weapon)
self.contents.draw_text(4, 32 * 1, 92, 32, $data_system.words.armor1)
self.contents.draw_text(4, 32 * 2, 92, 32, $data_system.words.armor2)
self.contents.draw_text(4, 32 * 3, 92, 32, $data_system.words.armor3)
self.contents.draw_text(5, 32 * 4, 92, 32, $data_system.words.armor4)
draw_item_name(@data[0], 92, 32 * 0)
draw_item_name(@data[1], 92, 32 * 1)
draw_item_name(@data[2], 92, 32 * 2)
draw_item_name(@data[3], 92, 32 * 3)
draw_item_name(@data[4], 92, 32 * 4)
end

def update_help
@help_window.set_text(self.item == nil ? "" : self.item.description)
end
end

class Window_EquipItem2 < Window_Selectable

def initialize(actor, equip_type)
super(272, 256, 368, 224)
@actor = actor
@equip_type = equip_type
@column_max = 1
refresh
self.active = false
self.index = -1
end

def item
return @data[self.index]
end
```

```
def refresh
if self.contents != nil
self.contents.dispose
self.contents = nil
end
@data = []

if @equip_type == 0
weapon_set = $data_classes[@actor.class_id].weapon_set
for i in 1...$data_weapons.size
if $game_party.weapon_number(i) > 0 and weapon_set.include?(i)
@data.push($data_weapons[i])
end
end
end

if @equip_type != 0
armor_set = $data_classes[@actor.class_id].armor_set
for i in 1...$data_armors.size
if $game_party.armor_number(i) > 0 and armor_set.include?(i)
if $data_armors[i].kind == @equip_type-1
@data.push($data_armors[i])
end
end
end
end

@data.push(nil)

@item_max = @data.size
self.contents = Bitmap.new(width - 32, row_max * 32)
self.contents.font.name = $fontface
self.contents.font.size = $fontsize
for i in 0...@item_max-1
draw_item(i)
end
end

def draw_item(index)
item = @data[index]
x = 4 + index % 2 * (288 + 32)
y = index / 2 * 32
case item
when RPG::Weapon
number = $game_party.weapon_number(item.id)
when RPG::Armor
number = $game_party.armor_number(item.id)
end
bitmap = RPG::Cache.icon(item.icon_name)
self.contents.blt(x, y + 4, bitmap, Rect.new(0, 0, 24, 24))
self.contents.font.color = normal_color
self.contents.draw_text(x + 28, y, 212, 32, item.name, 0)
self.contents.draw_text(x + 240, y, 16, 32, ":", 1)
self.contents.draw_text(x + 256, y, 24, 32, number.to_s, 2)
end

def update_help
@help_window.set_text(self.item == nil ? "" : self.item.description)
end
end

class Scene_Equip2

def initialize(actor_index = 0, equip_index = 0)
@actor_index = actor_index
@equip_index = equip_index
end

def main
@spriteset = Spriteset_Map.new
@actor = $game_party.actors[@actor_index]
@help_window = Window_Help.new
@help_window.back_opacity = 160
```


RPG-LIVE.NEUF.FR

```
@help_window.y = -100
@left_window = Window_EquipLeft2.new(@actor)
@left_window.back_opacity = 160
@left_window.x = -280
@right_window = Window_EquipRight2.new(@actor)
@right_window.back_opacity = 160
@right_window.x = 642
@item_window1 = Window_EquipItem2.new(@actor, 0)
@item_window1.back_opacity = 160
@item_window1.y = 486
@item_window2 = Window_EquipItem2.new(@actor, 1)
@item_window2.back_opacity = 160
@item_window2.y = 486
@item_window3 = Window_EquipItem2.new(@actor, 2)
@item_window3.back_opacity = 160
@item_window3.y = 486
@item_window4 = Window_EquipItem2.new(@actor, 3)
@item_window4.back_opacity = 160
@item_window4.y = 486
@item_window5 = Window_EquipItem2.new(@actor, 4)
@item_window5.back_opacity = 160
@item_window5.y = 486
@right_window.help_window = @help_window
@item_window1.help_window = @help_window
@item_window2.help_window = @help_window
@item_window3.help_window = @help_window
@item_window4.help_window = @help_window
@item_window5.help_window = @help_window
@right_window.index = @equip_index

refresh
Graphics.transition
loop do
Graphics.update
Input.update

update
if $scene != self
break
end
end

Graphics.freeze
@help_window.dispose
@left_window.dispose
@right_window.dispose
@item_window1.dispose
@item_window2.dispose
@item_window3.dispose
@item_window4.dispose
@item_window5.dispose
@spriteset.dispose
end

def refresh
@item_window1.visible = (@right_window.index == 0)
@item_window2.visible = (@right_window.index == 1)
@item_window3.visible = (@right_window.index == 2)
@item_window4.visible = (@right_window.index == 3)
@item_window5.visible = (@right_window.index == 4)

item1 = @right_window.item

case @right_window.index
when 0
@item_window = @item_window1
when 1
@item_window = @item_window2
when 2
@item_window = @item_window3
when 3
@item_window = @item_window4
when 4
@item_window = @item_window5
```


RPG-LIVE.NEUF.FR

```
end

if @right_window.active

@left_window.set_new_parameters(nil, nil, nil)
end

if @item_window.active
item2 = @item_window.item
last_hp = @actor.hp
last_sp = @actor.sp
@actor.equip(@right_window.index, item2 == nil ? 0 : item2.id)
new_atk = @actor.atk
new_pdef = @actor.pdef
new_mdef = @actor.mdef
@actor.equip(@right_window.index, item1 == nil ? 0 : item1.id)
@actor.hp = last_hp
@actor.sp = last_sp
@left_window.set_new_parameters(new_atk, new_pdef, new_mdef)
end
end

def update
if @help_window.y < 0
@help_window.y += 10
end
@left_window.update
if @left_window.x < 0
@left_window.x += 10
end
@right_window.update
if @right_window.x > 272
@right_window.x -= 10
end
@item_window.update
if @item_window1.y > 256
@item_window1.y -= 10
end
if @item_window2.y > 256
@item_window2.y -= 10
end
if @item_window3.y > 256
@item_window3.y -= 10
end
if @item_window4.y > 256
@item_window4.y -= 10
end
if @item_window5.y > 256
@item_window5.y -= 10
end
refresh

if @right_window.active
update_right
return
end

if @item_window.active
update_item
return
end
end

def update_right

if Input.trigger?(Input::B)
$game_system.se_play($data_system.cancel_se)
$scene = Scene_Menu.new(2)
return
end

if Input.trigger?(Input::C)
if @actor.equip_fix?(@right_window.index)
$game_system.se_play($data_system.buzzer_se)
```


RPG-LIVE.NEUF.FR

```
return
end

$game_system.se_play($data_system.decision_se)
@right_window.active = false
@item_window.active = true
@item_window.index = 0
return
end

if Input.trigger?(Input::R)
$game_system.se_play($data_system.cursor_se)
@actor_index += 1
@actor_index %= $game_party.actors.size
$scene = Scene_Equip.new(@actor_index, @right_window.index)
return
end

if Input.trigger?(Input::L)
$game_system.se_play($data_system.cursor_se)
@actor_index += $game_party.actors.size - 1
@actor_index %= $game_party.actors.size
$scene = Scene_Equip.new(@actor_index, @right_window.index)
return
end
end

def update_item
if Input.trigger?(Input::B)
$game_system.se_play($data_system.cancel_se)
@right_window.active = true
@item_window.active = false
@item_window.index = -1
return
end

if Input.trigger?(Input::C)
$game_system.se_play($data_system.equip_se)
item = @item_window.item
@actor.equip(@right_window.index, item == nil ? 0 : item.id)
@right_window.active = true
@item_window.active = false
@item_window.index = -1
@right_window.refresh
@item_window.refresh
return
end
end
end
```


RPG-LIVE.NEUF.FR