

Pour installer le script ombre :

1/ Remplacez le code se "Sprite_Character" par :

```
#=====
=====
# ` Sprite_Character
#=====
=====

class Sprite_Character < RPG::Sprite

attr_accessor :character

def initialize(viewport, character = nil)
  super(viewport)
  @character = character
  #-----Sun edit-----
  @sunlist=[]
  if (character.is_a?(Game_Event) and character.list!=nil and character.list[0].code == 108 and
  character.list[0].parameters == ["sun"])
  if (character.list[1]!=nil and character.list[1].code == 108)
  @self_angle=character.list[1].parameters[0]
  else
  @self_angle=45
  end
  if (character.list[2]!=nil and character.list[2].code == 108)
  @self_opacity=character.list[2].parameters[0]
  else
  @self_opacity=128
  end
  for i in $game_map.events.keys.sort
  if ($game_map.events[i].is_a?(Game_Event) and $game_map.events[i].list!=nil and
  $game_map.events[i].list[0].code == 108 and $game_map.events[i].list[0].parameters == ["o"])
  @sunlist[i+1] = Sprite_Sun.new(viewport, $game_map.events[i],@self_angle,@self_opacity)
  end
  end
  @sunlist[1] = Sprite_Sun.new(viewport, $game_player,@self_angle,@self_opacity)
  end
  #-----/Sun edit-----
  #-----Shadows edit-----
  @ombrelist=[]
  if (character.is_a?(Game_Event) and character.list!=nil and character.list[0].code == 108 and
  character.list[0].parameters == ["s"])
  if (character.list[1]!=nil and character.list[1].code == 108)
  @anglemin=character.list[1].parameters[0]
```

```

end
if (character.list[2]!=nil and character.list[2].code == 108)
@anglemax=character.list[2].parameters[0]
end
if (character.list[3]!=nil and character.list[3].code == 108)
@distancemax=character.list[3].parameters[0]
end
for i in $game_map.events.keys.sort
if ($game_map.events[i].is_a?(Game_Event) and $game_map.events[i].list!=nil and
$game_map.events[i].list[0].code == 108 and $game_map.events[i].list[0].parameters == ["o"])
@ombrelist[i+1] = Sprite_Shadow.new(viewport,
$game_map.events[i],self,@anglemin,@anglemax,@distancemax)
end
end
@ombrelist[1] = Sprite_Shadow.new(viewport,
$game_player,self,@anglemin,@anglemax,@distancemax)
end
#-----/Shadows edit-----
update
end

def update
super
if @tile_id != @character.tile_id or
@character_name != @character.character_name or
@character_hue != @character.character_hue
@tile_id = @character.tile_id
@character_name = @character.character_name
@character_hue = @character.character_hue
if @tile_id >= 384
self.bitmap = RPG::Cache.tile($game_map.tileset_name,
@tile_id, @character.character_hue)
self.src_rect.set(0, 0, 32, 32)
self.ox = 16
self.oy = 32
else
self.bitmap = RPG::Cache.character(@character.character_name,
@character.character_hue)
@cw = bitmap.width / 4
@ch = bitmap.height / 4
self.ox = @cw / 2
self.oy = @ch
end
end
self.visible = (not @character.transparent)
if @tile_id == 0
sx = @character.pattern * @cw
sy = (@character.direction - 2) / 2 * @ch
self.src_rect.set(sx, sy, @cw, @ch)
end
self.x = @character.screen_x
self.y = @character.screen_y

```

```

self.z = @character.screen_z(@ch)
self.opacity = @character.opacity
self.blend_type = @character.blend_type
self.bush_depth = @character.bush_depth
if @character.animation_id != 0
animation = $data_animations[@character.animation_id]
animation(animation, true)
@character.animation_id = 0
end
#-----Shadows edit-----
if @ombrelist!=[]
for i in 1..@ombrelist.size
if @ombrelist[i]!=nil
@ombrelist[i].update
end
end
end
#-----/Shadows edit-----
#-----Shadows edit-----
if @sunlist!=[]
for i in 1..@sunlist.size
if @sunlist[i]!=nil
@sunlist[i].update
end
end
end
#-----/Shadows edit-----
end
end

```

```

*****
**

```

2/ Puis, faites un nouveau script au dessus de "main" et nommez le "Sprite_Shadow", puis collez le script ci-dessous :

```

#=====
=====
# ` Sprit_Ombre
# Basé sur les ombres de Genzai Kawakami, dynamisme par Rataime, extras par Boushy
#=====
=====

```

```

class Sprite_Shadow < RPG::Sprite

attr_accessor :character # ??????

def initialize(viewport, character = nil,source = nil,anglemin=0,anglemax=0,distancemax=0)
super(viewport)
@anglemin=anglemin.to_f
@anglemax=anglemax.to_f
@distancemax=distancemax.to_f
@character = character
@source = source
update
end

def update
super

if @tile_id != @character.tile_id or
@character_name != @character.character_name or
@character_hue != @character.character_hue
@tile_id = @character.tile_id
@character_name = @character.character_name
@character_hue = @character.character_hue
if @tile_id >= 384
self.bitmap = RPG::Cache.tile($game_map.tileset_name,
@tile_id, @character.character_hue)
self.src_rect.set(0, 0, 32, 32)
self.ox = 16
self.oy = 32
else
self.bitmap = RPG::Cache.character(@character.character_name,
@character.character_hue)
@cw = bitmap.width / 4
@ch = bitmap.height / 4
self.ox = @cw / 2
self.oy = @ch
end
end
self.visible = (not @character.transparent)
if @tile_id == 0
sx = @character.pattern * @cw
@direct=@character.direction

if self.angle>90 or angle<-90

if @direct== 6
sy = ( 4- 2) / 2 * @ch
end
if @direct== 4
sy = ( 6- 2) / 2 * @ch
end
if @direct != 4 and @direct !=6

```

```

sy = (@character.direction - 2) / 2 * @ch
end
else
sy = (@character.direction - 2) / 2 * @ch
end
self.src_rect.set(sx, sy, @cw, @ch)
end
self.x = @character.screen_x
self.y = @character.screen_y-5
self.z = @character.screen_z(@ch)-1
self.opacity = @character.opacity
self.blend_type = @character.blend_type
self.bush_depth = @character.bush_depth
if @character.animation_id != 0
animation = $data_animations[@character.animation_id]
animation(animation, true)
@character.animation_id = 0
end
@deltax=@source.x-self.x
@deltay= @source.y-self.y
self.angle = 57.3*Math.atan2(@deltax, @deltay )
@angle_trigo=self.angle+90
if @angle_trigo<0
@angle_trigo=360+@angle_trigo
end
self.color = Color.new(0, 0, 0)
@distance = ((@deltax ** 2) + (@deltay ** 2))
self.opacity = 1200000/(@distance+6000)
@distance = @distance ** 0.5
if @distancemax !=0 and @distance>=@distancemax
self.opacity=0
end
if @anglemin !=0 or @anglemax !=0
if (@angle_trigo<@anglemin or @angle_trigo>@anglemax) and @anglemin<@anglemax
self.opacity=0
end
if (@angle_trigo<@anglemin and @angle_trigo>@anglemax) and @anglemin>@anglemax
self.opacity=0
end
end
end
end
end
end

```

3/ Enfin, faites un dernier script au dessus de "main" et nommez le "Sprite_Sun", puis collez le

script ci-dessous :

```
#=====
=====
# ` Sprite_Sun
## Basé sur Sprite_Shadow, modifié par Rataime
#=====
=====

class Sprite_Sun < RPG::Sprite
  attr_accessor :character

  def initialize(viewport, character = nil, self_angle = 45, self_opacity = 128)
    super(viewport)
    @character = character
    @self_angle = self_angle
    @self_opacity = self_opacity
    update
  end

  def update
    super
    if @tile_id != @character.tile_id or
      @character_name != @character.character_name or
      @character_hue != @character.character_hue
      @tile_id = @character.tile_id
      @character_name = @character.character_name
      @character_hue = @character.character_hue
      if @tile_id >= 384
        self.bitmap = RPG::Cache.tile($game_map.tileset_name,
          @tile_id, @character.character_hue)
        self.src_rect.set(0, 0, 32, 32)
        self.ox = 16
        self.oy = 32
      else
        self.bitmap = RPG::Cache.character(@character.character_name,
          @character.character_hue)
        @cw = bitmap.width / 4
        @ch = bitmap.height / 4
        self.ox = @cw / 2
        self.oy = @ch
      end
    end

    self.visible = (not @character.transparent)

    if @tile_id == 0
      sx = @character.pattern * @cw
```

```

@direct=@character.direction

if self.angle>90 or angle<-90

if @direct== 6
sy = ( 4- 2) / 2 * @ch#@character.direction
end
if @direct== 4
sy = ( 6- 2) / 2 * @ch
end
if @direct != 4 and @direct !=6
sy = (@character.direction - 2) / 2 * @ch
end
else
sy = (@character.direction - 2) / 2 * @ch
end
self.src_rect.set(sx, sy, @cw, @ch)
end
self.x = @character.screen_x
self.y = @character.screen_y-5
self.z = @character.screen_z(@ch)-1
self.opacity = @character.opacity
self.blend_type = @character.blend_type
self.bush_depth = @character.bush_depth
if @character.animation_id != 0
animation = $data_animations[@character.animation_id]
animation(animation, true)
@character.animation_id = 0
end
self.angle = @self_angle-90
self.color = Color.new(0, 0, 0)
self.opacity = @self_opacity
end
end

```

Pour utiliser ce script :

Pour créer une source de lumière, ajoutez un événement avec un commentaire `l`
 Pour donner une ombre à un événement, ajoutez un commentaire `o`

QUESTION: pierreon@laposte.net

<http://epidemais.idoo.com/>