RMVA 清 Esc 窗口三件套[RGSS][脚本整理]

制作更原创的 RPG 内容的时候，需要这个脚本。
网上不好找，因为太简单。
我做这个不够精简，将就用，懒得删东西。
1.window kill ：去掉金钱和右侧主角团队状态栏

```
#==============================================================================
# ** Scene_Menu    修改  by:洮蘋芝闇
#------------------------------------------------------------------------------
#   This class performs the menu screen processing.
#==============================================================================

class Scene_Menu < Scene_MenuBase
  #--------------------------------------------------------------------------
  # * Start Processing
  #--------------------------------------------------------------------------
  def start
    super
    create_command_window
    create_gold_window
    create_status_window
  end
  #--------------------------------------------------------------------------
  # * Create Command Window
  #--------------------------------------------------------------------------
  def create_command_window
    @command_window = Window_MenuCommand.new
    @command_window.set_handler(:item,      method(:command_item))
    @command_window.set_handler(:skill,     method(:command_personal))
    @command_window.set_handler(:equip,     method(:command_personal))
    @command_window.set_handler(:status,    method(:command_personal))
    @command_window.set_handler(:formation, method(:command_formation))
    @command_window.set_handler(:save,      method(:command_save))
    @command_window.set_handler(:game_end,  method(:command_game_end))
    @command_window.set_handler(:cancel,    method(:return_scene))
  end
  #--------------------------------------------------------------------------
  # * Create Gold Window
  #--------------------------------------------------------------------------
  def create_gold_window
    # @gold_window = Window_Gold.new
    # @gold_window.x = 0
    # @gold_window.y = Graphics.height - @gold_window.height
  end
```

```
#--------------------------------------------------------------------------
# * Create Status Window
#--------------------------------------------------------------------------
def create_status_window
  # status_window = Window_MenuStatus.new(@command_window.width, 0)
end
#--------------------------------------------------------------------------
# * [Item] Command
#--------------------------------------------------------------------------
def command_item
  SceneManager.call(Scene_Item)
end
#--------------------------------------------------------------------------
# * [Skill], [Equipment] and [Status] Commands
#--------------------------------------------------------------------------
def command_personal
  @status_window.select_last
  @status_window.activate
  @status_window.set_handler(:ok,     method(:on_personal_ok))
  @status_window.set_handler(:cancel, method(:on_personal_cancel))
end
#--------------------------------------------------------------------------
# * [Formation] Command
#--------------------------------------------------------------------------
def command_formation
  @status_window.select_last
  @status_window.activate
  @status_window.set_handler(:ok,     method(:on_formation_ok))
  @status_window.set_handler(:cancel, method(:on_formation_cancel))
end
#--------------------------------------------------------------------------
# * [Save] Command
#--------------------------------------------------------------------------
def command_save
  SceneManager.call(Scene_Save)
end
#--------------------------------------------------------------------------
# * [Exit Game] Command
#--------------------------------------------------------------------------
def command_game_end
  SceneManager.call(Scene_End)
end
#--------------------------------------------------------------------------
# * [OK] Personal Command
```

```
#--------------------------------------------------------------------------
def on_personal_ok
  case @command_window.current_symbol
  when :skill
    SceneManager.call(Scene_Skill)
  when :equip
    SceneManager.call(Scene_Equip)
  when :status
    SceneManager.call(Scene_Status)
  end
end
#--------------------------------------------------------------------------
# * [Cancel] Personal Command
#--------------------------------------------------------------------------
def on_personal_cancel
  @status_window.unselect
  @command_window.activate
end
#--------------------------------------------------------------------------
# * Formation [OK]
#--------------------------------------------------------------------------
def on_formation_ok
  if @status_window.pending_index >= 0
    $game_party.swap_order(@status_window.index,
                           @status_window.pending_index)
    @status_window.pending_index = -1
    @status_window.redraw_item(@status_window.index)
  else
    @status_window.pending_index = @status_window.index
  end
  @status_window.activate
end
#--------------------------------------------------------------------------
# * Formation [Cancel]
#--------------------------------------------------------------------------
def on_formation_cancel
  if @status_window.pending_index >= 0
    @status_window.pending_index = -1
    @status_window.activate
  else
    @status_window.unselect
    @command_window.activate
  end
end
```

```
end
```

2.command kill 去掉除了退出和道具以外的所有选项

```
#==============================================================================
# ** Window_MenuCommand   修改  by:洮豲芝闇
#------------------------------------------------------------------------------
#   This command window appears on the menu screen.
#==============================================================================

class Window_MenuCommand < Window_Command
  #--------------------------------------------------------------------------
  # * Initialize Command Selection Position (Class Method)
  #--------------------------------------------------------------------------
  def self.init_command_position
    @@last_command_symbol = nil
  end
  #--------------------------------------------------------------------------
  # * Object Initialization
  #--------------------------------------------------------------------------
  def initialize
    super(0, 0)
    select_last
  end
  #--------------------------------------------------------------------------
  # * Get Window Width
  #--------------------------------------------------------------------------
  def window_width
    return 160
  end
  #--------------------------------------------------------------------------
  # * Get Number of Lines to Show
  #--------------------------------------------------------------------------
  def visible_line_number
    item_max
  end
  #--------------------------------------------------------------------------
  # * Create Command List
  #--------------------------------------------------------------------------
  def make_command_list
    add_main_commands
    #add_formation_command
    #add_original_commands
    #add_save_command
    add_game_end_command
  end
```

```
#--------------------------------------------------------------------------
# * Add Main Commands to List
#--------------------------------------------------------------------------
def add_main_commands
  add_command(Vocab::item,    :item,    main_commands_enabled)
  #add_command(Vocab::skill,   :skill,   main_commands_enabled)
  #add_command(Vocab::equip,   :equip,   main_commands_enabled)
  #add_command(Vocab::status, :status, main_commands_enabled)
end
#--------------------------------------------------------------------------
# * Add Formation to Command List
#--------------------------------------------------------------------------
def add_formation_command

  #add_command(Vocab::formation, :formation, formation_enabled)

end
#--------------------------------------------------------------------------
# * For Adding Original Commands
#--------------------------------------------------------------------------
def add_original_commands
end
#--------------------------------------------------------------------------
# * Add Save to Command List
#--------------------------------------------------------------------------
def add_save_command
  add_command(Vocab::save, :save, save_enabled)
end
#--------------------------------------------------------------------------
# * Add Exit Game to Command List
#--------------------------------------------------------------------------
def add_game_end_command
  add_command(Vocab::game_end, :game_end)
end
#--------------------------------------------------------------------------
# * Get Activation State of Main Commands
#--------------------------------------------------------------------------
def main_commands_enabled
  $game_party.exists
end
#--------------------------------------------------------------------------
# * Get Activation State of Formation
#--------------------------------------------------------------------------
def formation_enabled
```

```
    $game_party.members.size >= 2 && !$game_system.formation_disabled
  end
  #--------------------------------------------------------------------------
  # * Get Activation State of Save
  #--------------------------------------------------------------------------
  def save_enabled
    !$game_system.save_disabled
  end
  #--------------------------------------------------------------------------
  # * Processing When OK Button Is Pressed
  #--------------------------------------------------------------------------
  def process_ok
    @@last_command_symbol = current_symbol
    super
  end
  #--------------------------------------------------------------------------
  # * Restore Previous Selection Position
  #--------------------------------------------------------------------------
  def select_last
    select_symbol(@@last_command_symbol)
  end
end
```

3.statu kill：单纯去掉主角团队状态，单独使用时仍会显示右侧大窗口

```
#==============================================================================
# ** Window_MenuStatus          修改  by:洮瓹芝闇
#------------------------------------------------------------------------------
#   This window displays party member status on the menu screen.
#==============================================================================

class Window_MenuStatus < Window_Selectable
  #--------------------------------------------------------------------------
  # * Public Instance Variables
  #--------------------------------------------------------------------------
  attr_reader   :pending_index            # Pending position (for formation)
  #--------------------------------------------------------------------------
  # * Object Initialization
  #--------------------------------------------------------------------------
  def initialize(x, y)
    super(x, y, window_width, window_height)
    @pending_index = -1
    refresh
  end
  #--------------------------------------------------------------------------
  # * Get Window Width
```

```
#--------------------------------------------------------------------------
def window_width
  Graphics.width - 160
end
#--------------------------------------------------------------------------
# * Get Window Height
#--------------------------------------------------------------------------
def window_height
  Graphics.height
end
#--------------------------------------------------------------------------
# * Get Number of Items
#--------------------------------------------------------------------------
def item_max
  $game_party.members.size
end
#--------------------------------------------------------------------------
# * Get Item Height
#--------------------------------------------------------------------------
def item_height
  (height - standard_padding * 2) / 4
end
#--------------------------------------------------------------------------
# * Draw Item
#--------------------------------------------------------------------------
def draw_item(index)
  # actor = $game_party.members[index]
  # enabled = $game_party.battle_members.include?(actor)
  # rect = item_rect(index)
  # draw_item_background(index)
  # draw_actor_face(actor, rect.x + 1, rect.y + 1, enabled)
  # draw_actor_simple_status(actor, rect.x + 108, rect.y + line_height / 2)
end
#--------------------------------------------------------------------------
# * Draw Background for Item
#--------------------------------------------------------------------------
def draw_item_background(index)
  # if index == @pending_index
  #    contents.fill_rect(item_rect(index), pending_color)
  # end
end
#--------------------------------------------------------------------------
# * Processing When OK Button Is Pressed
#--------------------------------------------------------------------------
```

```
  def process_ok
    super
    $game_party.menu_actor = $game_party.members[index]
  end
  #--------------------------------------------------------------------------
  # * Restore Previous Selection Position
  #--------------------------------------------------------------------------
  def select_last
    select($game_party.menu_actor.index || 0)
  end
  #--------------------------------------------------------------------------
  # * Set Pending Position (for Formation)
  #--------------------------------------------------------------------------
  def pending_index=(index)
    last_pending_index = @pending_index
    @pending_index = index
    redraw_item(@pending_index)
    redraw_item(last_pending_index)
  end
end
```

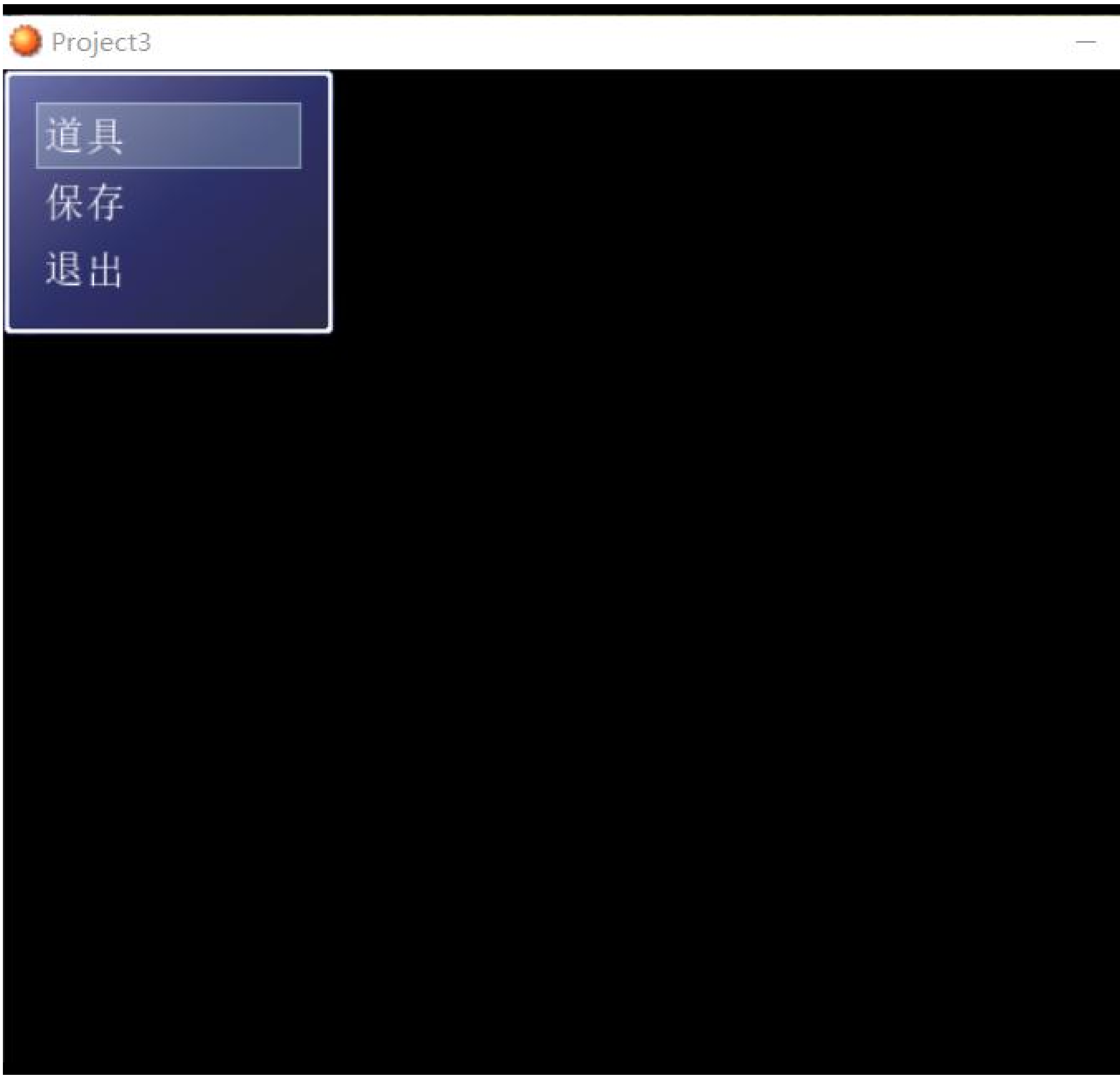
RMXP 纯净 Esc 菜单脚本[RGSS][脚本整理]
如题：
注意：清掉了默认的菜单里的队伍状态相关内容

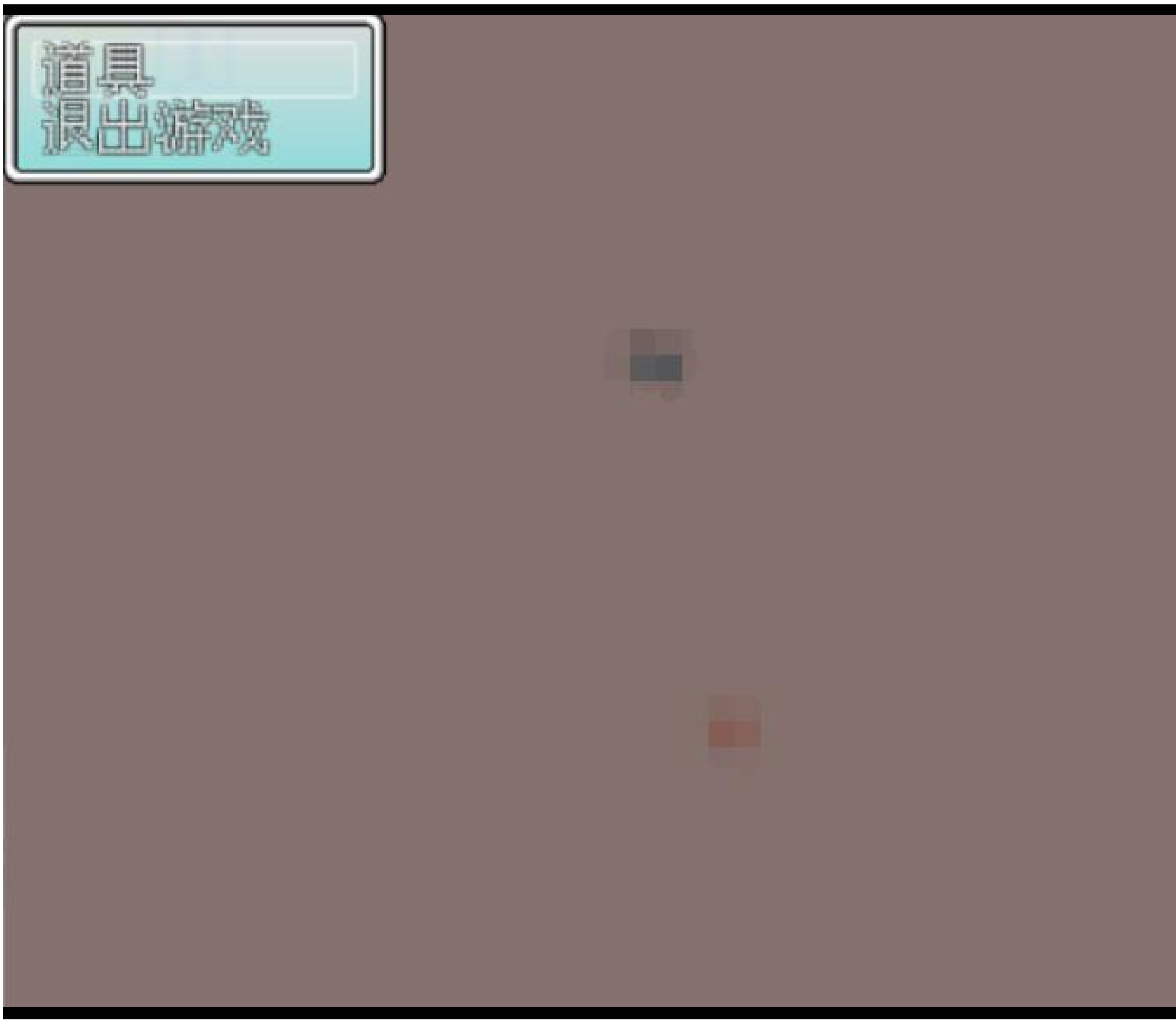
```
#==============================================================================
# ** Scene_Menu   修改 by：洮瘲芝闇
#------------------------------------------------------------------------------
#   This class performs menu screen processing.
#==============================================================================

class Scene_Menu
  #--------------------------------------------------------------------------
  # * Object Initialization
  #       menu_index : command cursor's initial position
  #--------------------------------------------------------------------------
  def initialize(menu_index = 0)
    @menu_index = menu_index
  end
  #--------------------------------------------------------------------------
  # * Main Processing
  #--------------------------------------------------------------------------
  def main
    # Make command window
    s1 = $data_system.words.item
    s2 = $data_system.words.skill
    s3 = $data_system.words.equip
    s4 = "Status"
    s5 = "保存"
    s6 = "退出"
    @command_window = Window_Command.new(160, [s1,   s5, s6])
    @command_window.index = @menu_index
    # If number of party members is 0
    if $game_party.actors.size == 0
      # Disable items, skills, equipment, and status
      @command_window.disable_item(0)

    end
    # If save is forbidden
    if $game_system.save_disabled
      # Disable save
      @command_window.disable_item(4)
    end

    # Execute transition
    Graphics.transition
    # Main loop
    loop do
      # Update game screen
      Graphics.update
      # Update input information
      Input.update
      # Frame update
      update
      # Abort loop if screen is changed
      if $scene != self
        break
      end
    end
    # Prepare for transition
    Graphics.freeze
    # Dispose of windows
    @command_window.dispose

  end
  #--------------------------------------------------------------------------
  # * Frame Update
  #--------------------------------------------------------------------------
  def update
  # Update windows
  @command_window.update

  # If command window is active: call update_command
  if @command_window.active
    update_command
    return
  end
  # If status window is active: call update_status
  if @status_window.active
    update_status
    return
  end
end
#--------------------------------------------------------------------------
# * Frame Update (when command window is active)
#--------------------------------------------------------------------------
def update_command
  # If B button was pressed
  if Input.trigger?(Input::B)
    # Play cancel SE
    $game_system.se_play($data_system.cancel_se)
    # Switch to map screen
    $scene = Scene_Map.new
    return
  end
  # If C button was pressed
  if Input.trigger?(Input::C)
    # If command other than save or end game, and party members = 0
    if $game_party.actors.size == 0 and @command_window.index < 4
      # Play buzzer SE
      $game_system.se_play($data_system.buzzer_se)
      return
    end
    # Branch by command window cursor position
    case @command_window.index
    when 0   # item
      # Play decision SE
      $game_system.se_play($data_system.decision_se)
      # Switch to item screen
      $scene = Scene_Item.new
    when 1# save
      # If saving is forbidden
      if $game_system.save_disabled
```

```
          # Play buzzer SE
          $game_system.se_play($data_system.buzzer_se)
          return
        end
        # Play decision SE
        $game_system.se_play($data_system.decision_se)
        # Switch to save screen
        $scene = Scene_Save.new
      when 2   # end game
        # Play decision SE
        $game_system.se_play($data_system.decision_se)
        # Switch to end game screen
        $scene = Scene_End.new
      end
      return
    end
  end
  #--------------------------------------------------------------------------
  # * Frame Update (when status window is active)
  #--------------------------------------------------------------------------
  def update_status
    # If B button was pressed
    if Input.trigger?(Input::B)
      # Play cancel SE
      $game_system.se_play($data_system.cancel_se)
      # Make command window active
      @command_window.active = true
      @status_window.active = false
      @status_window.index = -1
      return
    end
    # If C button was pressed
    if Input.trigger?(Input::C)
      # Branch by command window cursor position
      case @command_window.index
      when 1   # skill
        # If this actor's action limit is 2 or more
        if $game_party.actors[@status_window.index].restriction >= 2
          # Play buzzer SE
          $game_system.se_play($data_system.buzzer_se)
          return
        end
        # Play decision SE
        $game_system.se_play($data_system.decision_se)
        # Switch to skill screen
        $scene = Scene_Skill.new(@status_window.index)
      when 2   # equipment
        # Play decision SE
        $game_system.se_play($data_system.decision_se)
        # Switch to equipment screen
        $scene = Scene_Equip.new(@status_window.index)
      when 3   # status
        # Play decision SE
        $game_system.se_play($data_system.decision_se)
        # Switch to status screen
        $scene = Scene_Status.new(@status_window.index)
      end
      return
    end
  end
end

class Scene_Save < Scene_File
  #--------------------------------------------------------------------------
  # * Object Initialization
  #--------------------------------------------------------------------------
  def initialize
    super("Which file would you like to save to?")
  end
  #--------------------------------------------------------------------------
  # * Decision Processing
  #--------------------------------------------------------------------------
  def on_decision(filename)
    # Play save SE
    $game_system.se_play($data_system.save_se)
    # Write save data
    file = File.open(filename, "wb")
    write_save_data(file)
    file.close
    # If called from event
    if $game_temp.save_calling
      # Clear save call flag
      $game_temp.save_calling = false
      # Switch to map screen
      $scene = Scene_Map.new
      return
    end
    # Switch to menu screen
  $scene = Scene_Menu.new(1)
end
#--------------------------------------------------------------------------
# * Cancel Processing
#--------------------------------------------------------------------------
def on_cancel
  # Play cancel SE
  $game_system.se_play($data_system.cancel_se)
  # If called from event
  if $game_temp.save_calling
    # Clear save call flag
    $game_temp.save_calling = false
    # Switch to map screen
    $scene = Scene_Map.new
    return
  end
  # Switch to menu screen
  $scene = Scene_Menu.new(1)
end
#--------------------------------------------------------------------------
# * Write Save Data
#     file : write file object (opened)
#--------------------------------------------------------------------------
def write_save_data(file)
  # Make character data for drawing save file
  characters = []
  for i in 0...$game_party.actors.size
    actor = $game_party.actors[i]
    characters.push([actor.character_name, actor.character_hue])
  end
  # Write character data for drawing save file
  Marshal.dump(characters, file)
  # Wrire frame count for measuring play time
  Marshal.dump(Graphics.frame_count, file)
  # Increase save count by 1
  $game_system.save_count += 1
  # Save magic number
  # (A random value will be written each time saving with editor)
  $game_system.magic_number = $data_system.magic_number
  # Write each type of game object
  Marshal.dump($game_system, file)
  Marshal.dump($game_switches, file)
  Marshal.dump($game_variables, file)
  Marshal.dump($game_self_switches, file)
```

```
    Marshal.dump($game_screen, file)
    Marshal.dump($game_actors, file)
    Marshal.dump($game_party, file)
    Marshal.dump($game_troop, file)
    Marshal.dump($game_map, file)
    Marshal.dump($game_player, file)
  end
end
class Scene_End
  #--------------------------------------------------------------------------
  # * Main Processing
  #--------------------------------------------------------------------------
  def main
    # Make command window
    s1 = "To Title"
    s2 = "Shutdown"
    s3 = "Cancel"
    @command_window = Window_Command.new(192, [s1, s2, s3])
    @command_window.x = 320 - @command_window.width / 2
    @command_window.y = 240 - @command_window.height / 2
    # Execute transition
    Graphics.transition
    # Main loop
    loop do
      # Update game screen
      Graphics.update
      # Update input information
      Input.update
      # Frame Update
      update
      # Abort loop if screen is changed
      if $scene != self
        break
      end
    end
    # Prepare for transition
    Graphics.freeze
    # Dispose of window
    @command_window.dispose
    # If switching to title screen
    if $scene.is_a?(Scene_Title)
      # Fade out screen
      Graphics.transition
      Graphics.freeze
  end
end
#--------------------------------------------------------------------------
# * Frame Update
#--------------------------------------------------------------------------
def update
  # Update command window
  @command_window.update
  # If B button was pressed
  if Input.trigger?(Input::B)
    # Play cancel SE
    $game_system.se_play($data_system.cancel_se)
    # Switch to menu screen
    $scene = Scene_Menu.new(2)
    return
  end
  # If C button was pressed
  if Input.trigger?(Input::C)
    # Branch by command window cursor position
    case @command_window.index
    when 0   # to title
      command_to_title
    when 1   # shutdown
      command_shutdown
    when 2   # quit
      command_cancel
    end
    return
  end
end
#--------------------------------------------------------------------------
# * Process When Choosing [To Title] Command
#--------------------------------------------------------------------------
def command_to_title
  # Play decision SE
  $game_system.se_play($data_system.decision_se)
  # Fade out BGM, BGS, and ME
  Audio.bgm_fade(800)
  Audio.bgs_fade(800)
  Audio.me_fade(800)
  # Switch to title screen
  $scene = Scene_Title.new
end
#--------------------------------------------------------------------------
```

```
  # * Process When Choosing [Shutdown] Command
  #--------------------------------------------------------------------------
  def command_shutdown
    # Play decision SE
    $game_system.se_play($data_system.decision_se)
    # Fade out BGM, BGS, and ME
    Audio.bgm_fade(800)
    Audio.bgs_fade(800)
    Audio.me_fade(800)
    # Shutdown
    $scene = nil
  end
  #--------------------------------------------------------------------------
  # *   Process When Choosing [Cancel] Command
  #--------------------------------------------------------------------------
  def command_cancel
    # Play decision SE
    $game_system.se_play($data_system.decision_se)
    # Switch to menu screen
    $scene = Scene_Menu.new(2)
  end
end
```


Project3
道具
保存
退出


道具
退出游戏