Shadow Hearts' Judgment Ring system for RPG Maker 2003

by Avee

Contents

0. Introduction

This tutorial will teach you how to make and code your own Ring Areas (the colored areas within the Ring) to use with my SH Judgment Ring system for RPG Maker 2003.

1. Pictures

Pictures No. 1, 2, 41 to 45 and No. 50 are used in this system.
Here is the list of what on-screen images the different Pictures correspond to:

Picture No. 01 = Monster's Hit Points gauge
Picture No. 02 = Hero's Hit Points gauge
Picture No. 41 = Ring background image (the grey and black circle)
Picture No. 42 = Ring Area (the colored areas within the Ring)
Picture No. 43 = Stopped Ring Cursor no.1
Picture No. 44 = Stopped Ring Cursor no.2
Picture No. 45 = Stopped Ring Cursor no.3
Picture No. 50 = Ring Cursor (the white spinning bar)

2. Switches and Variables

Here is the list of used Switches and Variables:

Switch 0002 "Ring Cursor moves"
When ON, this makes the Cursor spin.

Switch 0003 "Stopped Cursor 1"
When ON, this shows Picture No. 43 (Stopped Ring Cursor no.1)

Switch 0004 "Stopped Cursor 2"
When ON, this shows Picture No. 44 (Stopped Ring Cursor no.2)

Switch 0005 "Stopped Cursor 3"
When ON, this shows Picture No. 45 (Stopped Ring Cursor no.3)

Switch 0008 "Hit or Miss ON"

When ON, this allows the player to press Enter to stop the Cursor while it is spinning.

Switch 0009 "Attacking ON"

When ON, this activates the Attack scene between the hero and the monster.

Switch 0101 "Attack hit no1"

When ON, this will activate the hero's first attack at normal strength during the Attack scene.

Switch 0102 "Attack hit no2"

When ON, this will activate the hero's second attack at normal strength during the Attack scene.

Switch 0103 "Attack hit no3"

When ON, this will activate the hero's third attack at normal strength during the Attack scene.

Switch 0106 "Attack strike no1"

When ON, this will activate the hero's first attack at double strength during the Attack scene.

Switch 0107 "Attack strike no2"

When ON, this will activate the hero's second attack at double strength during the Attack scene.

Switch 0108 "Attack strike no3"

When ON, this will activate the hero's third attack at double strength during the Attack scene.

Variable 0001 "Ring Cursor posi"

This keeps track of the Cursor's current position within the Ring. You won't need to edit this.

Variable 0002 "Ring Cursor speed"

This determines the spinning speed of the Cursor. You can edit this if you want but I don't recommend it.

Variable 0004 "Ring Area"

This determines what Ring Area image will be shown and used. You will have to alter this variable for every new Ring Area image you want to code.

Variable 0006 "Press Enter"

This allows the player to press Enter to stop the Cursor while it is spinning. You won't need to edit this.

Variable 0007 "No of times to press"
This keeps track of how many times the player can press Enter during the spin. This value must always be equal to the number of Area portions in the currently used Ring Area image, so make sure you edit it right.

Variable 0008 "Stopped C 1 value"
This will record the Cursor's position for Picture No. 43 (Stopped Ring Cursor no.1) when Enter is pressed. You won't need to edit this.

Variable 0009 "Stopped C 2 value"
This will record the Cursor's position for Picture No. 44 (Stopped Ring Cursor no.2) when Enter is pressed. You won't need to edit this.

Variable 0010 "Stopped C 3 value"
This will record the Cursor's position for Picture No. 45 (Stopped Ring Cursor no.3) when Enter is pressed. You won't need to edit this.

Variable 0035 "monster hp"
This keeps track of the monster's Hit Points for Picture No. 01 (Monster's Hit Points gauge). If you design your own battle system, edit this.

Variable 0036 "hero hp"
This keeps track of the monster's Hit Points for Picture No. 02 (Hero's Hit Points gauge). If you design your own battle system, edit this.

3. Events

Here is the list of used Events for both the Battle System and the Lottery:

(000,001) Cursor move speed
This sets the Cursor's spinning speed. You can edit this if you want but I don't recommend it.

(000,002) Cursor move image
This shows Picture No. 50 (Ring Cursor). You won't need to edit this.

(001,002) Cursor move image 2
This also shows Picture No. 50 (Ring Cursor). You won't need to edit this.

(002,002) Cursor move image 3
This also shows Picture No. 50 (Ring Cursor). You won't need to edit this.

(003,002) Ring Area appear
This shows Picture No. 42 (Ring Area) when called. You will have to make a new page in this event for every new Ring Area image you want to code. You will also have to edit the (001,014) Message event in order to select the right Ring Areas.

(000,004) Press

This manages the press Enter command to stop the Cursor while it is spinning. You won't need to edit this.

(000,005) Hit or Miss

This keeps track of whether you press Enter when the Cursor is over a colored Area or not. You will have to make a new page for every new Ring Area image you want to code. You will also have to edit the values according to each Ring Area. This is a big part of the coding you'll have to do.

(001,014) Message

This is the first part of the Battle/Lottery system. You will have to edit it according to your own Battle or Lottery System.

(006,014) Attacking / Get prizes

This is the second part of the Battle/Lottery system. You will have to edit it according to your own Battle or Lottery System.

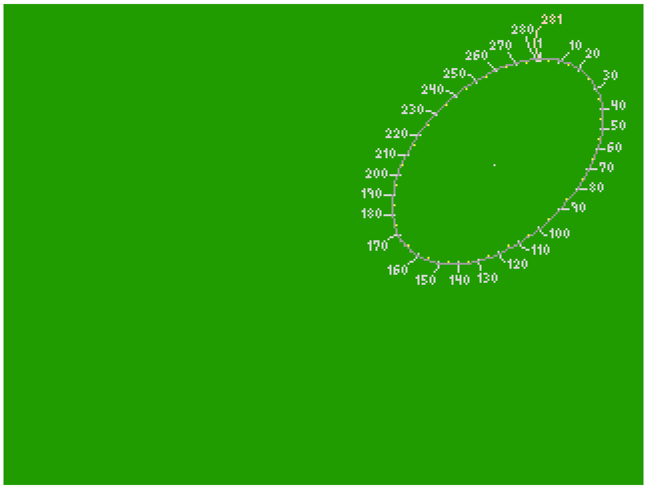
Battle System map's 9 upper right events

These show the Stopped Ring Cursor images. You won't need to edit these.

4. Drawing Ring Areas

I like using iDraw for this, but you can probably use another drawing program.

A) Open the following file: [Avee] SH Judgment Ring\Picture\judgment ring.png

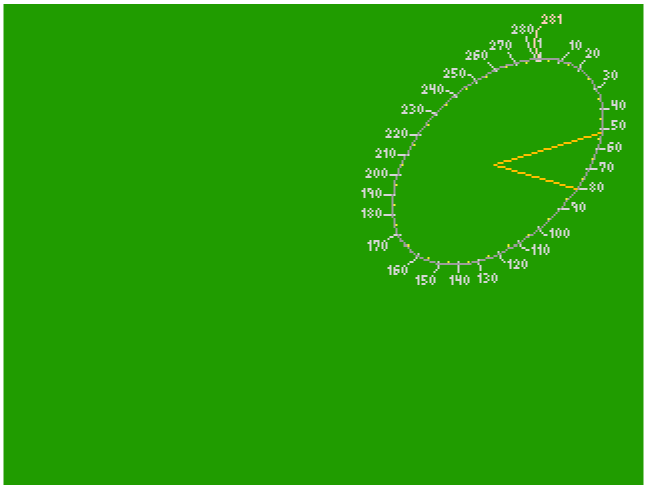


This is the template for drawing Ring Areas.

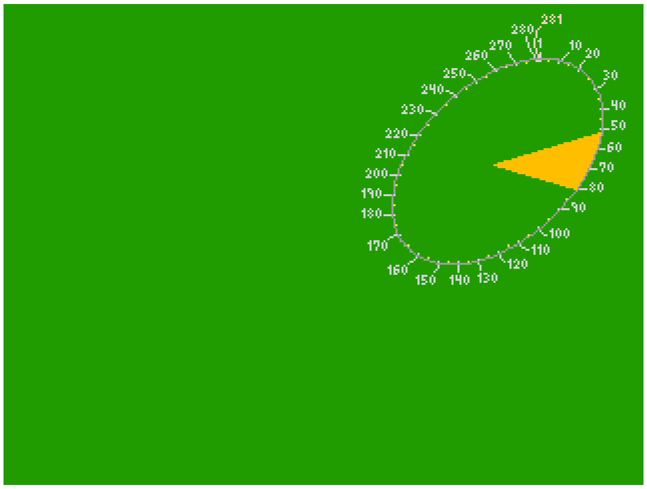
The ring's center is marked by the central light grey dot at position (245,080).
The ring's outline is a darker grey.

The light grey numbers and lines, yellow dots, and pink number and line all mark Cursor positions. These positions correspond to Variable 0001 "Ring Cursor posi" values. The Cursor always starts at value 1 and reaches value 281 after a complete spin.

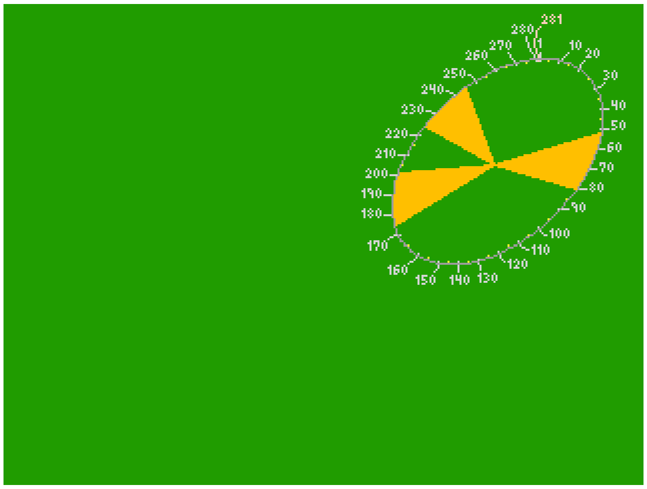
B) Select a straight line drawing tool. Draw a line starting at the central light grey dot, going all the way to a light grey, yellow or pink dot, or in-between near the dark grey outline. Then draw a second line to complete the edges of the Area portion. **Make sure you also keep track of what value (Cursor position) both lines correspond to. In the example below, the Area portion starts at value 52 and ends at value 80.**



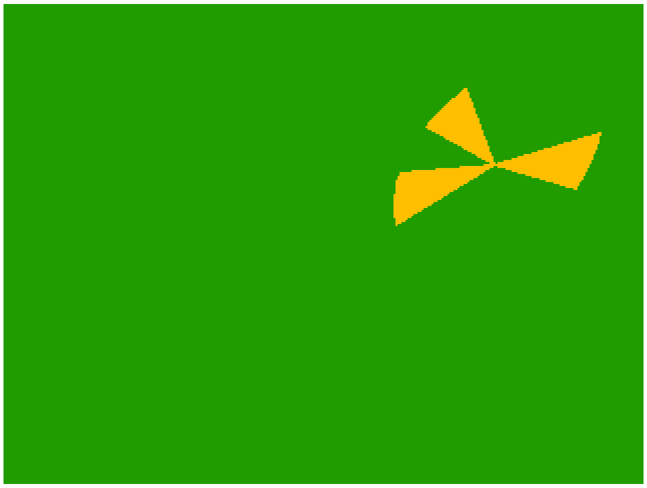
C) Fill in the Area portion with color. Add your own personal style to it. **Make sure you color over the dots but not over the dark grey outline.**



D) Repeat steps B and C if you wish. Since a maximum of 3 Area portions is coded with the system, don't make more than 3 portions unless you edit the allowed maximum. Let's make two other portions at values 175-201 and 224-245.



E) Save your file under a different name, erase everything but the Area portions and save your file once again. Your custom Ring Area is now done:



5. Coding Ring Areas

In this example, we've made a three-portion Ring Area to use in the Battle System. The portions' values are 52-80, 175-201 and 224-245.

A) Open the Sandbox Battle map from the game folder under RPG Maker 2003.

B) Open the following event to edit: (003,002) Ring Area appear

C) Copy and Paste Page 1 to make a Page 2.

D) In the Preconditions of Page 2, set Variable 0004 "Ring Area" value to 1. This value will correspond to our newly-made Ring Area.

E) In the Event Commands box, edit the Show Picture command and select our newly-made Ring Area as the Image File to be shown. Then save and exit this event by clicking OK.

F) Open the following event to edit: (001,014) Message

G) In the Event Commands box, locate the Variable Operation: [0004:Ring Area] Set, 0 and set it to 1. Also locate the Call Event: Ring Area appear[1] right below it and edit it to open Page 2 of the Ring Area appear Map Event. Then save and exit this event by clicking OK.

H) Open the following event to edit: (000,005) Hit or Miss

I) Create a new Page, Page 2.

J) In the Preconditions of Page 2, set Variable 0004 "Ring Area" value to 1, the value corresponding to our newly-made Ring Area.

K) Now we will code the hit or miss detection for our Ring Area. We will code many Conditional Branches checking the values of Variable 0010 "Stopped C 3 value" for both colored and blank portions of the Area. These values must reflect the Position values we wrote down for our three new Ring Area portions, so that the game will correctly identify if the player hits a colored portion or not when pressing Enter during the Cursor's spin.
Remember our portions' values: 52-80, 175-201 and 224-245.

k1) In the Event Command box, add a sound effect to let the player know the Cursor will begin to spin, than add a "Label 1" right below it.

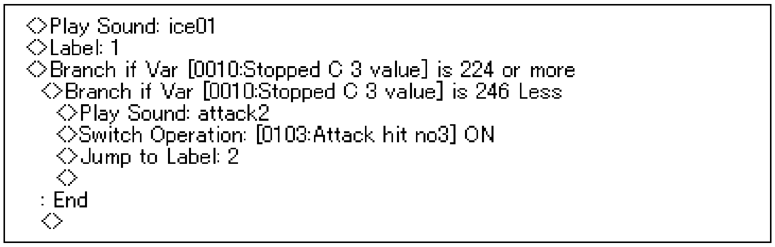
k2) Then make a Conditional Branch checking Variable 0010 "Stopped C 3 value". Set its Number to 224 and select "Greater than or Equal to" in the white box below it. **Check** the Option "Execute Custom Handler if Condition Not Met". Click OK.

k3) Right below this Conditional Branch, make a new one checking Variable 0010 "Stopped C 3 value" again. Set its Number to 246 and select "Less than" in the white box below it. **Do not check** the Option "Execute Custom Handler if Condition Not Met". Click OK. These two branches will check if the player presses Enter when the Cursor is over the colored Area portion that stretches out from value 224 to value 245.

k4) Add a sound effect to let the player know they successfully hit this colored portion.

k5) Turn ON Switch 0103 "Attack hit no3".

k6) Add a "Jump to Label 2". Your code should now look like this:

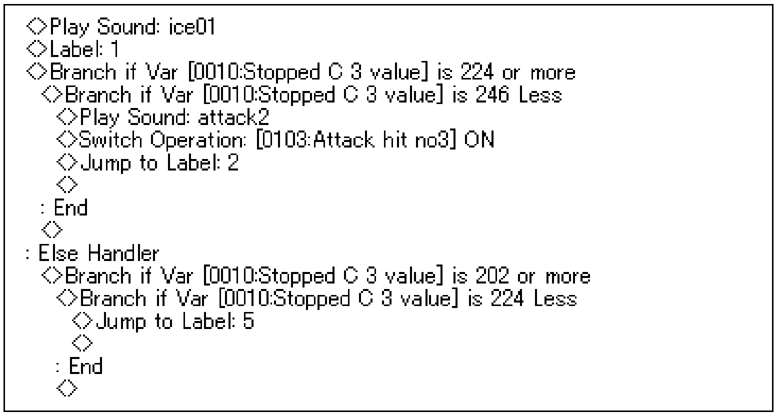


```
◇Play Sound: ice01
◇Label: 1
◇Branch if Var [0010:Stopped C 3 value] is 224 or more
  ◇Branch if Var [0010:Stopped C 3 value] is 246 Less
    ◇Play Sound: attack2
    ◇Switch Operation: [0103:Attack hit no3] ON
    ◇Jump to Label: 2
    ◇
  : End
  ◇
```

k7) We will now code the blank portion that follow this colored portion counter-clockwise. **Under the Else Handler notice** below, make a Conditional Branch checking Variable 0010 "Stopped C 3 value". Set its Number to 202 and select "Greater than or Equal to" in the white box below it. **Check** the Option "Execute Custom Handler if Condition Not Met". Click OK.

k8) Right below this Conditional Branch, make a new one checking Variable 0010 "Stopped C 3 value" again. Set its Number to 224 and select "Less than" in the white box below it. **Do not check** the Option "Execute Custom Handler if Condition Not Met". Click OK. These two branches will check if the player presses Enter when the Cursor is over the blank Area portion that stretches out from value 202 to value 223.

k9) Add a "Jump to Label 5". The whole code should now look like this:



```
◇Play Sound: ice01
◇Label: 1
◇Branch if Var [0010:Stopped C 3 value] is 224 or more
  ◇Branch if Var [0010:Stopped C 3 value] is 246 Less
    ◇Play Sound: attack2
    ◇Switch Operation: [0103:Attack hit no3] ON
    ◇Jump to Label: 2
    ◇
  : End
  ◇
: Else Handler
  ◇Branch if Var [0010:Stopped C 3 value] is 202 or more
    ◇Branch if Var [0010:Stopped C 3 value] is 224 Less
      ◇Jump to Label: 5
      ◇
    : End
    ◇
```

We just completed a first "block" of coding. You will now have to repeat steps **k2** to **k9** for all the remaining values. Don't forget to code both colored and blank portions. Also, change Switch 0103 "Attack hit no3" to Switch 0102 "Attack hit no2" for the second block and to Switch 0101 "Attack hit no1" for the third block. When done, we will code the last blank portion that goes from Cursor's position 1 to 51:

k10) Under the Else Handler notice again, make a Conditional Branch checking Variable 0010 "Stopped C 3 value". Set its Number to 52 and select "Less than" in the white box below it. **Do not check** the Option "Execute Custom Handler if Condition Not Met". Click OK.

k11) Right below this Conditional Branch, make a new one checking Variable 0010 "Stopped C 3 value" again. Set its Number to 0 and select "Greater than" in the white box below it. **Do not check** the Option "Execute Custom Handler if Condition Not Met". Click OK.

k12) Add a "Label 5". This is where hitting a blank portion will lead the code.

k13) Add a sound effect of failure.

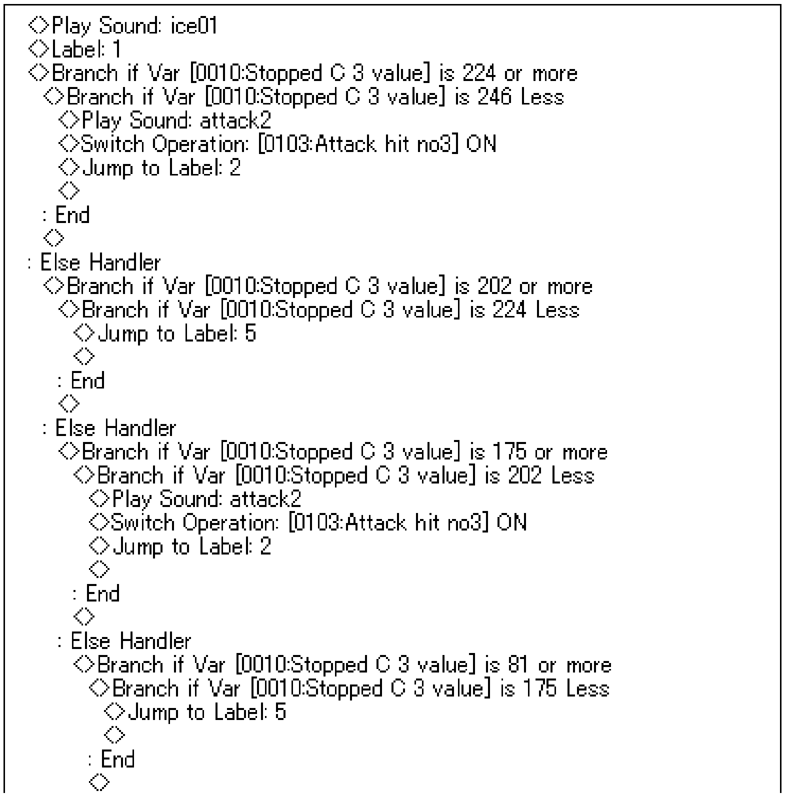
k14) Turn OFF Switch 0002 "Ring Cursor moves". Then add a Wait 0.0 seconds.

k15) Move Pictures Number 50, 45, 44, 43, 42 and 41 to 100% Transparency in 3 tenths of a second. Check the Option "Halt Other Processes During Transition" for the last Picture.

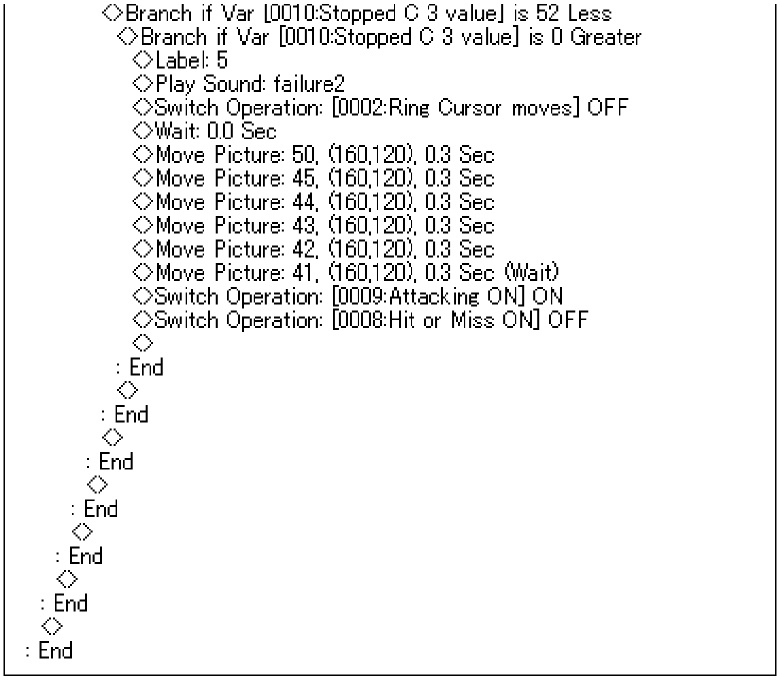
k16) Turn ON Switch 0009 "Attacking ON".

k17) Turn OFF Switch 0008 "Hit or Miss ON".

The whole code should now look like this:



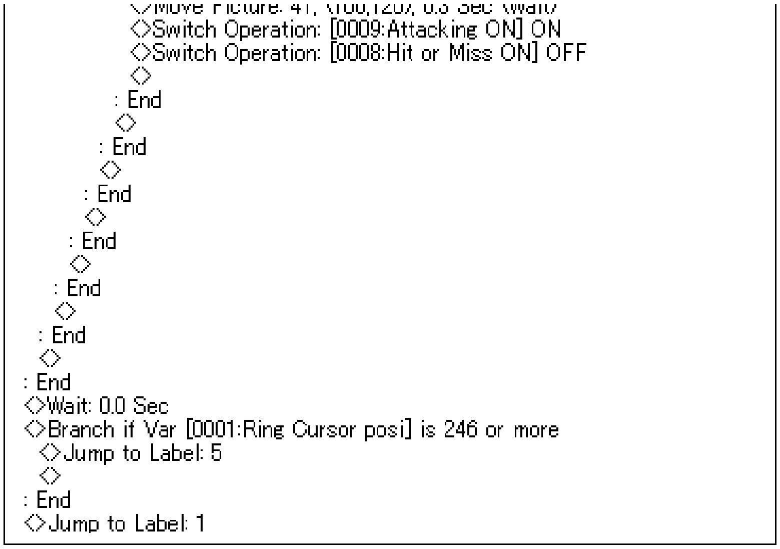
```
◇Play Sound: ice01
◇Label: 1
◇Branch if Var [0010:Stopped C 3 value] is 224 or more
  ◇Branch if Var [0010:Stopped C 3 value] is 246 Less
    ◇Play Sound: attack2
    ◇Switch Operation: [0103:Attack hit no3] ON
    ◇Jump to Label: 2
    ◇
  : End
  ◇
: Else Handler
  ◇Branch if Var [0010:Stopped C 3 value] is 202 or more
    ◇Branch if Var [0010:Stopped C 3 value] is 224 Less
      ◇Jump to Label: 5
      ◇
    : End
    ◇
  : Else Handler
    ◇Branch if Var [0010:Stopped C 3 value] is 175 or more
      ◇Branch if Var [0010:Stopped C 3 value] is 202 Less
        ◇Play Sound: attack2
        ◇Switch Operation: [0103:Attack hit no3] ON
        ◇Jump to Label: 2
        ◇
      : End
      ◇
    : Else Handler
      ◇Branch if Var [0010:Stopped C 3 value] is 81 or more
        ◇Branch if Var [0010:Stopped C 3 value] is 175 Less
          ◇Jump to Label: 5
          ◇
        : End
        ◇
```



```
          ◇Branch if Var [0010:Stopped C 3 value] is 52 Less
            ◇Branch if Var [0010:Stopped C 3 value] is 0 Greater
              ◇Label: 5
              ◇Play Sound: failure2
              ◇Switch Operation: [0002:Ring Cursor moves] OFF
              ◇Wait: 0.0 Sec
              ◇Move Picture: 50, (160,120), 0.3 Sec
              ◇Move Picture: 45, (160,120), 0.3 Sec
              ◇Move Picture: 44, (160,120), 0.3 Sec
              ◇Move Picture: 43, (160,120), 0.3 Sec
              ◇Move Picture: 42, (160,120), 0.3 Sec
              ◇Move Picture: 41, (160,120), 0.3 Sec (Wait)
              ◇Switch Operation: [0009:Attacking ON] ON
              ◇Switch Operation: [0008:Hit or Miss ON] OFF
              ◇
            : End
            ◇
          : End
          ◇
        : End
        ◇
      : End
      ◇
    : End
    ◇
  : End
  ◇
: End
```

k18) At the very last " : End" notice of the code, add a Wait 0.0 seconds followed by a Conditional Branch checking Variable 0010 "Stopped C 3 value". Set its Number to 246 and select "Greater than or Equal to" in the white box below it. **Do not check** the Option "Execute Custom Handler if Condition Not Met". Click OK. Then add a "Jump to Label 5".

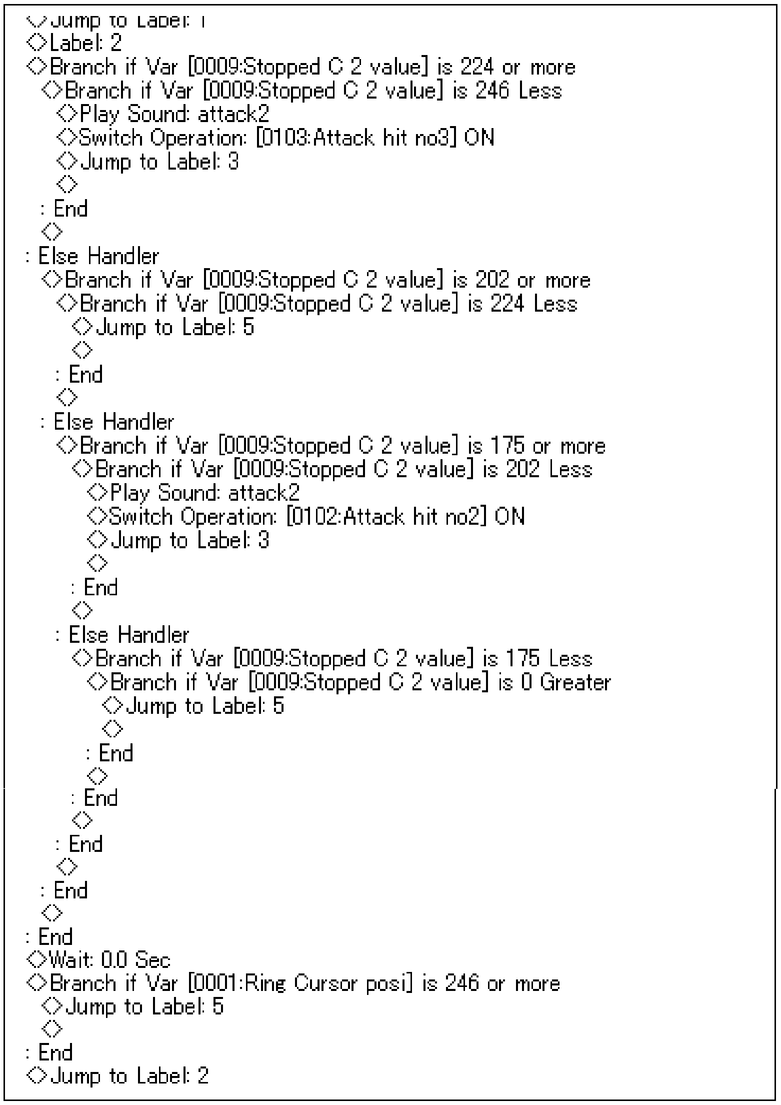
k19) At the " : End" notice below, add a "Jump to Label 1". You'll get this:



```
              ◇Move Picture: 41, (160,120), 0.3 Sec (Wait)
              ◇Switch Operation: [0009:Attacking ON] ON
              ◇Switch Operation: [0008:Hit or Miss ON] OFF
              ◇
            : End
            ◇
          : End
          ◇
        : End
        ◇
      : End
      ◇
    : End
    ◇
  : End
  ◇
: End
◇Wait: 0.0 Sec
◇Branch if Var [0001:Ring Cursor posi] is 246 or more
  ◇Jump to Label: 5
  ◇
: End
◇Jump to Label: 1
```

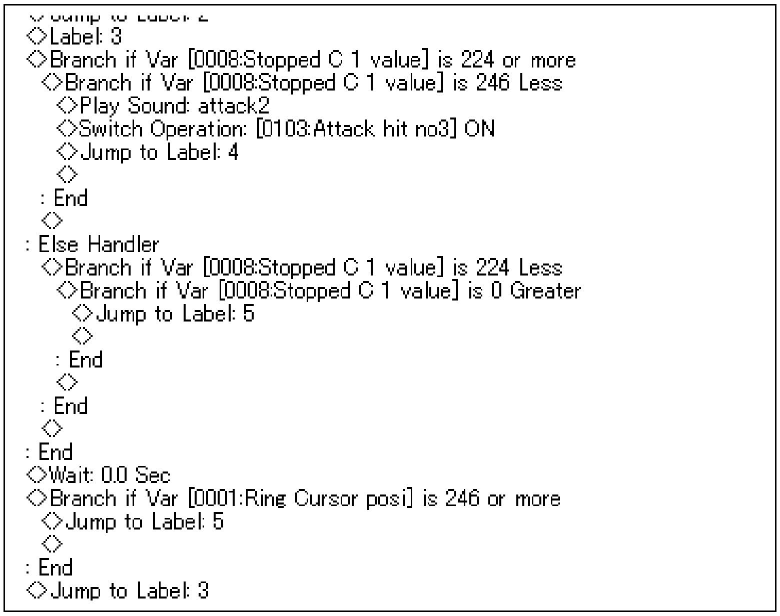
Now that might have been quite a lot of work, but we only coded the first of 3 possibilities to press Enter to stop the Cursor, as there are three colored portions to our Ring Area image. We have two more to go, but each one is a bit shorter to code than the previous one.

L) Repeat steps k1 to k19 but don't add the first sound effect in step k1. Change Label 1 to Label 2. Also change "Jump to Label 2" for "Jump to Label 3". Change Variable 0010 "Stopped C 3 value" to C 2. Only Switch 0103 "Attack hit no3" and Switch 0102 "Attack hit no2" will be used. Take a look at the image below to know if you coded everything right:



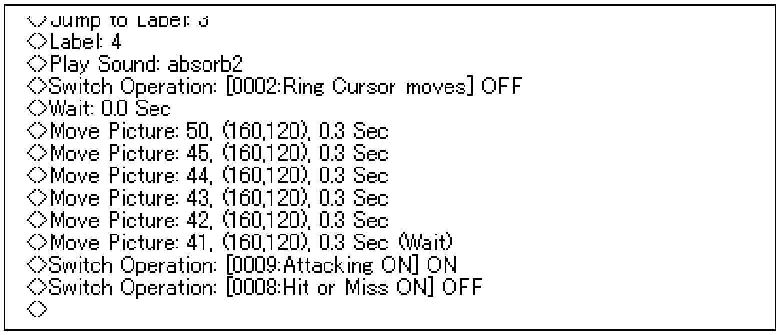
```
◇Jump to Label: 1
◇Label: 2
◇Branch if Var [0009:Stopped C 2 value] is 224 or more
 ◇Branch if Var [0009:Stopped C 2 value] is 246 Less
  ◇Play Sound: attack2
  ◇Switch Operation: [0103:Attack hit no3] ON
  ◇Jump to Label: 3
  ◇
 : End
 ◇
: Else Handler
 ◇Branch if Var [0009:Stopped C 2 value] is 202 or more
  ◇Branch if Var [0009:Stopped C 2 value] is 224 Less
   ◇Jump to Label: 5
   ◇
  : End
  ◇
 : Else Handler
  ◇Branch if Var [0009:Stopped C 2 value] is 175 or more
   ◇Branch if Var [0009:Stopped C 2 value] is 202 Less
    ◇Play Sound: attack2
    ◇Switch Operation: [0102:Attack hit no2] ON
    ◇Jump to Label: 3
    ◇
   : End
   ◇
  : Else Handler
   ◇Branch if Var [0009:Stopped C 2 value] is 175 Less
    ◇Branch if Var [0009:Stopped C 2 value] is 0 Greater
     ◇Jump to Label: 5
     ◇
    : End
    ◇
   : End
   ◇
  : End
  ◇
 : End
 ◇
: End
◇Wait: 0.0 Sec
◇Branch if Var [0001:Ring Cursor posi] is 246 or more
 ◇Jump to Label: 5
 ◇
: End
◇Jump to Label: 2
```

M) Repeat steps k1 to k19 one last time. Change Label 2 to Label 3. Also change "Jump to Label 3" for "Jump to Label 4". Change Variable 0010 "Stopped C 2 value" to C 1. Only Switch 0103 "Attack hit no3" will be used. You get the idea… Take a look at the image below to know if you coded everything right:



```
◇Jump to Label: 2
◇Label: 3
◇Branch if Var [0008:Stopped C 1 value] is 224 or more
  ◇Branch if Var [0008:Stopped C 1 value] is 246 Less
    ◇Play Sound: attack2
    ◇Switch Operation: [0103:Attack hit no3] ON
    ◇Jump to Label: 4
    ◇
  : End
  ◇
: Else Handler
  ◇Branch if Var [0008:Stopped C 1 value] is 224 Less
    ◇Branch if Var [0008:Stopped C 1 value] is 0 Greater
      ◇Jump to Label: 5
      ◇
    : End
    ◇
  : End
  ◇
: End
◇Wait: 0.0 Sec
◇Branch if Var [0001:Ring Cursor posi] is 246 or more
  ◇Jump to Label: 5
  ◇
: End
◇Jump to Label: 3
```

N) We now only have a small portion of code left to write. Add a "Label 4" at the bottom, then copy and paste the small portion of code under Label 5. Don't include the "Label 5" command and change the failure sound effect to a success sound effect. Here is what this should look like:



```
◇Jump to Label: 3
◇Label: 4
◇Play Sound: absorb2
◇Switch Operation: [0002:Ring Cursor moves] OFF
◇Wait: 0.0 Sec
◇Move Picture: 50, (160,120), 0.3 Sec
◇Move Picture: 45, (160,120), 0.3 Sec
◇Move Picture: 44, (160,120), 0.3 Sec
◇Move Picture: 43, (160,120), 0.3 Sec
◇Move Picture: 42, (160,120), 0.3 Sec
◇Move Picture: 41, (160,120), 0.3 Sec (Wait)
◇Switch Operation: [0009:Attacking ON] ON
◇Switch Operation: [0008:Hit or Miss ON] OFF
◇
```

Click OK to save this event and we are now done coding our new Ring Area for the Battle System. Run the game to test it.

Repeat this entire process if you want to add other custom Ring Areas. If you want to cut down the work, use Areas with only one colored portion in them. Feel free to experiment with creating and coding different looking Areas.

You can contact me on rpgmaker.net or rpgmakervx.net for any question regarding my Shadow Hearts' Judgment Ring system for RPG Maker 2003.

Avee