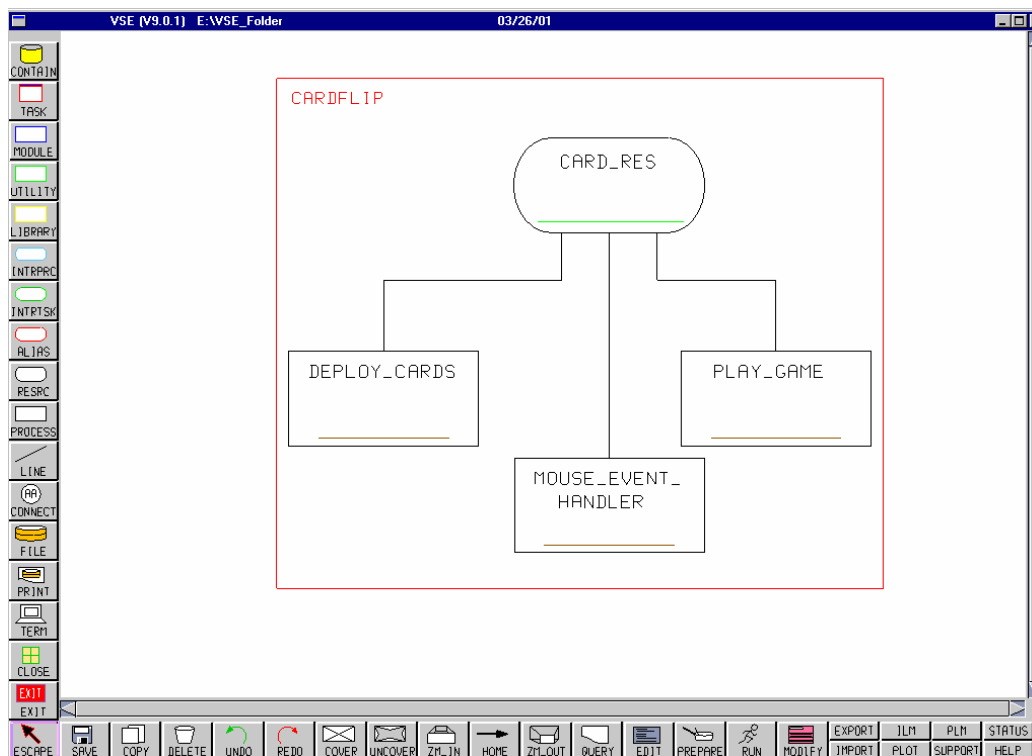


CHAPTER 7: RUN-TIME GRAPHICS— EVENT HANDLING

We are now ready to move on to a rather intensive example where we will apply everything we have learned by now (and then some!). We will implement a simple animation, which will display 2 rows of cards, flip them randomly, and then react to the user's mouse click events to toggle the cards.

For this program, we will use 3 processes, one of which is the RTG mouse event handler. It responds to mouse clicks enabling certain specified events to be triggered off. Below is an example of what the program skeleton looks like:



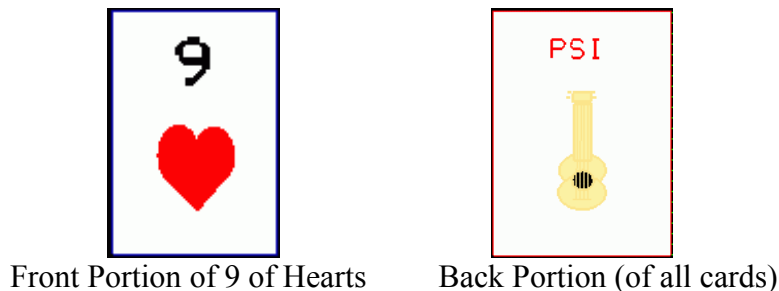
We will use 10 playing cards in this example. We have chosen: the Nine, King and Queen of Hearts, Ace, King and Queen of Clubs, 10 and Ace of Diamonds and King and Queen of Spades. You will need to create these 10 icons in the ILM. These icons will be the face portions of all the 10 cards.

Also, you will need to create a single icon to represent the back of the cards. You will have to create hierarchies of these cards – in this case, a hierarchy is the face portion of a card attached to its back portion. Let's examine the procedure for a single card. You can then repeat it for the remaining cards.

Step 1: Creating a Hierarchy for a Single Card.

- Create a single icon to show the 9 of Hearts. Call it FRONT_H9.
- Create its back portion in another icon. Call it 'BACK'.
- Back in the ILM, click on 'Create hierarchy' and then on the 'Icon' Button.
- A table of all your created icons will be displayed - in the form of a tiny representation of the icon to the left and the icon name to its right.
- Click on the tiny icon representation of ICON_H9 and click anywhere on the screen. The icon will be displayed there.
- Now click on the icon of 'BACK' and click on the screen. The BACK icon is now displayed.
- We need to attach these 2 icons together and build a hierarchy.
- Superimpose the 2 icons so that they lie coincident with each other.
- Click on ATTACH and then click on the icons. A prompt will ask you to confirm the action.
- Respond with OK.
- Save this hierarchy as ICON_H9 (for 9 of Hearts). This icon name is assigned to a variable within the control specification when we are creating the simulation. The card is now ready to be used within our program.

This is what your card should look like:



You will need to repeat this process for all of the remaining cards. The same back portion is used for each of the cards.

Step 2: Creating the Process and Resource.

When the icons are ready, create a task with one resource and three processes as illustrated. Then, add the control specification, process and resource code as detailed.

***Data for the Control Specification

```
CONTROL SECTION
  TITLE, CARD_FLIP
  SIMULATE

GRAPHICS SECTION
  ACTIVATE
  ICON CA = ICON_CA
  ICON CK = ICON_CK
  ICON CQ = ICON_CQ
  ICON D10 = ICON_D10
  ICON DA = ICON_DA
  ICON H9 = ICON_H9
  ICON HK = ICON_HK
  ICON HQ = ICON_HQ
  ICON SK = ICON_SK
  ICON SQ = ICON_SQ

  RTG_EVENT_HANDLER MOUSE_EVENT_HANDLER

MODEL SECTION
  DEPLOY_CARDS
END
```

***Data for the Resource: CARD_RES

```
*** CARDS
  1 CA ICON
  1 CK ICON
  1 CQ ICON
  1 D10 ICON
  1 DA ICON
  1 H9 ICON
  1 HK ICON
  1 HQ ICON
  1 SK ICON
  1 SQ ICON

  1 NUM_OF_CARDS INTEGER
  1 CARDS ICON QUANTITY(10)
  1 CLICKED_CARD ICON

  1 I INDEX
  1 J INDEX
  1 X REAL
  1 Y REAL
  1 DX REAL
  1 DY REAL
```

***Data for the Process: DEPLOY_CARDS

```
DEPLOY_CARDS
  EXECUTE INSERT_CARDS
  EXECUTE SHOW_CARDS
  EXECUTE AUTO_DEMO
  EXECUTE UPDATE_FUNCTION_BUTTONS
  CALL PLAY_GAME

AUTO_DEMO
  EXECUTE UNCOVER_THE_CARD
    INCREMENTING I UNTIL I IS GREATER THAN NUM_OF_CARDS
  EXECUTE COVER_THE_CARD
    INCREMENTING I UNTIL I IS GREATER THAN NUM_OF_CARDS

UNCOVER_THE_CARD
  UNCOVER_CARDS(I)

COVER_THE_CARD
  COVER_CARDS(I)

UPDATE_FUNCTION_BUTTONS
  DISPLAY F1_BUTTON 'TOGGLE'

SHOW_CARDS
  DX = 1300
  DY = 800
  I = 1
  EXECUTE MOVE_CARDS NUM_OF_CARDS TIMES

MOVE_CARDS
  IF I IS EQUAL TO 6
    DX = 1300
    DY = 400.
  X = -100
  Y = DY
  EXECUTE MOVE_A_CARD
    UNTIL X IS GREATER THAN DX
  DX = DX - 200
  INCREMENT I

MOVE_A_CARD
  UPDATE_CARDS(I) TO X,Y
  X = X+100

INSERT_CARDS
  DRAW_SCENE OFF
  X = -500
  Y = -500

  INSERT_H9_AT_X,Y
  CARDS(1) = H9
  INSERT_HK_AT_X,Y
  .
  .
  .
```

```

      .
      .
      .
CARDS (3) = HK
  INSERT HQ AT X,Y
  CARDS (5) = HQ
  INSERT CA AT X,Y
  CARDS (7) = CA
  INSERT CK AT X,Y
  CARDS (9) = CK
  INSERT CQ AT X,Y
  CARDS (2) = CQ
  INSERT D10 AT X,Y
  CARDS (4) = D10
  INSERT DA AT X,Y
  CARDS (6) = DA
  INSERT SK AT X,Y
  CARDS (8) = SK
  INSERT SQ AT X,Y
  CARDS (10) = SQ
  NUM_OF CARDS = 10

DRAW SCENE

```

The X-Y co-ordinate axis used in this environment is similar to that used in regular graphs, with the origin at the lower left corner.

To start off with, all the cards are inserted at location “-500,-500”, which is not visible in the workspace. Then each of them is displaced and moved such that they appear to be moving across the screen in two parallel rows. Then each of the cards is toggled so that, first their front portions and then their back portions are revealed. The “COVER” and “UNCOVER” functions have been used for this process. The “UNCOVER” function reveals the underlying elementary icon and the “COVER” function simply hides the lower icon and shows the higher one. In this program, the lower and higher icons are the front and back portions of the card respectively.

***Data for the Process: MOUSE_EVENT_HANDLER

```

MOUSE_EVENT_HANDLER

IF RTG_GRAPHICS_SYMBOL IS NOT ICON
  EXIT THIS RULE.

  CLICKED_CARD = RTG_ICON_POINTER
  IF RTG_SUPERIOR_ICON IS NON_EXISTENT
    UNCOVER CLICKED_CARD
  ELSE
    CLICKED_CARD = RTG_SUPERIOR_ICON
    COVER CLICKED_CARD.

```

You will notice that the one new thing we have done in this program is the utilization of the F1_BUTTON. There is a column of these buttons to the left of your interface when you run your program. Right now, we are using only one of these in order to toggle our cards.

We need to write a mouse event handler to handle the mouse click on the cards. Every time our user-defined Toggle switch is clicked (after having clicked on some card), we cover/uncover the card respectively. We, end up flipping the cards.

***Data for the Process: PLAY_CARDS

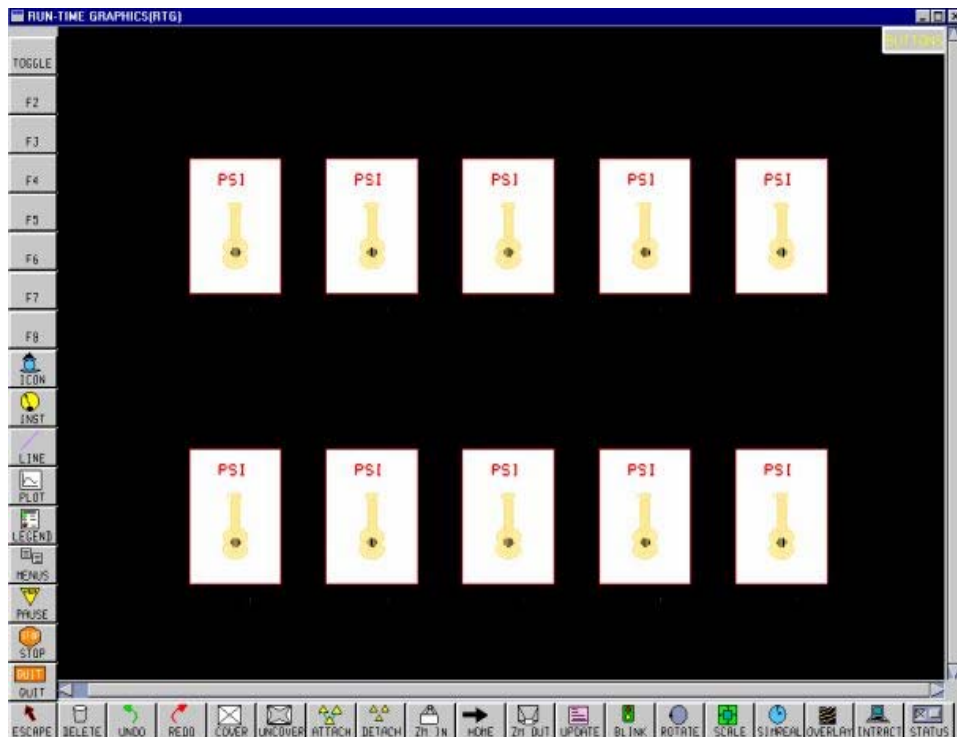
```
PLAY_GAME
  IF RTG_GRAPHICS_EVENT IS TERMINATE
    EXIT THIS RULE.

  SUSPEND THIS TASK FOR 0.5 SECOND

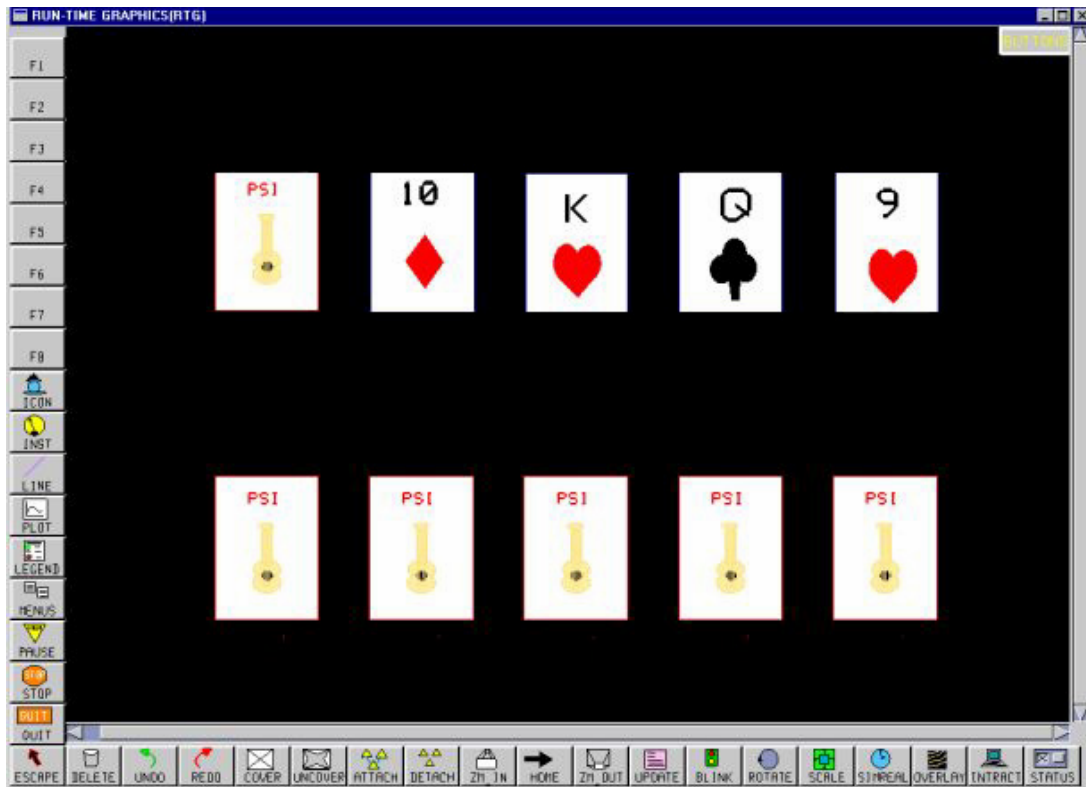
  SCHEDULE PLAY_GAME IN 0.2 SECOND
```

Step 3: Running the Program.

Now, prepare all the components and run the program. When running the program, you will see a sequence of events. First, the cards line up from the left side of the screen and position themselves. As observed, the backs of the cards can be seen.



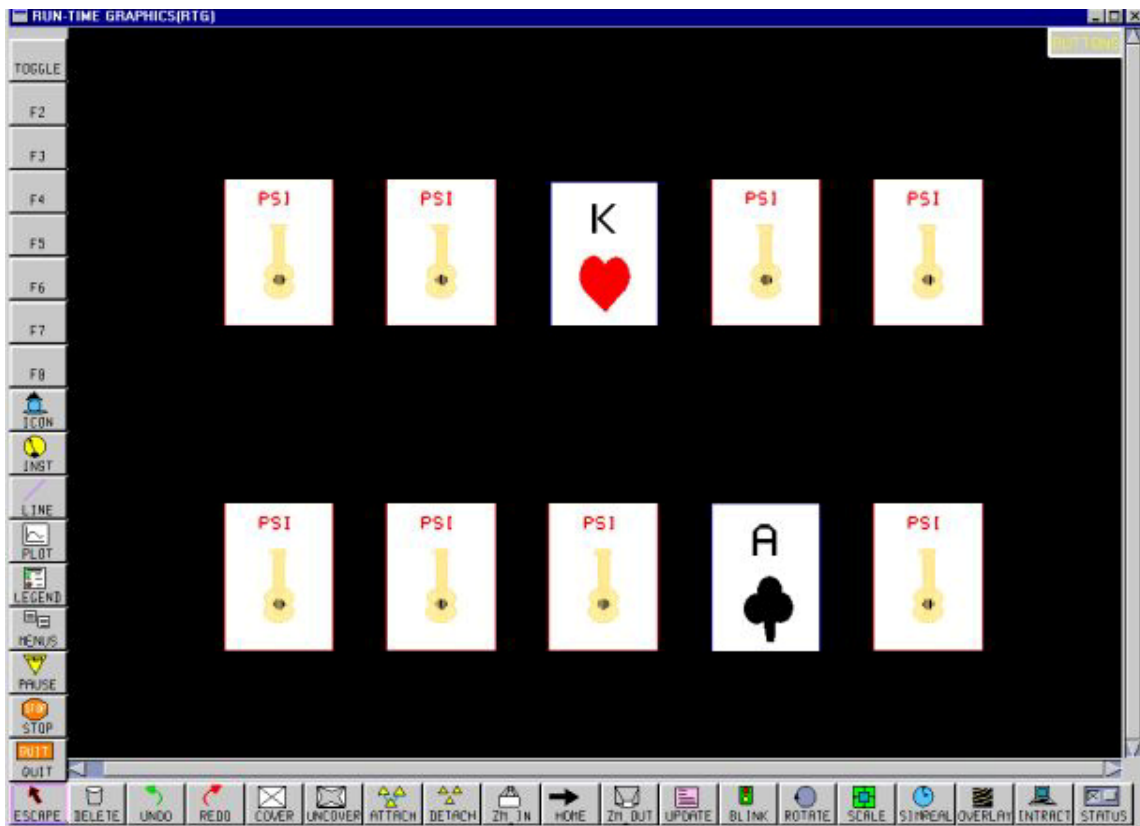
Now all the cards, in turn, uncover and cover themselves:



In the illustration, four cards are uncovered. When they are all uncovered, they will be covered again.

You will notice that one of the Function Buttons has been turned into a TOGGLE button. Click on the second last card in the bottom row.

Now, click on the TOGGLE button. The card will be uncovered. (The King of Hearts has already been uncovered in a similar manner). This is shown on the next page.



If you click on TOGGLE again, the card will be covered again. You have succeeded in creating this program!