# Table of Contents

Chapter 1  
**My Role on the Team**  .......................................................... 7  
  Activity 1-1 Anatomy of a Game Designer  .................................. 11  
  Activity 1-2 Personal Professional Assessment ............................ 14  
  Activity 1-3 Exploring Color  .................................................. 17  
  Activity 1-4 Multicultural Expo  ............................................. 19  
  Activity 1-5 Communication Technology  .................................. 21  

Chapter 2  
**Computerized Entertainment and Classification**  ..................... 23  
  Activity 2-1 Genre  ............................................................... 27  
  Activity 2-2 Quiz Show  ....................................................... 29  
  Activity 2-3 Cut Scene Compromise  ....................................... 42  

Chapter 3  
**Evolution of the Game** ...................................................... 53  
  Activity 3-1 Beneficial Vocabulary  ......................................... 57  
  Activity 3-2 Video Game History  .......................................... 60  
  Activity 3-3 Video Game Ratings and Immersion  ....................... 61  
  Activity 3-4 Physical Dexterity Puzzle  .................................. 63  
  Activity 3-5 Puzzle Game Construction  .................................. 74  

Chapter 4  
**Perspective, Scene Design, and Basic Animation** .................... 81  
  Activity 4-1 Landscape Design and Static Animation .................. 85  
  Activity 4-2 Active Animation  ............................................. 96  
  Activity 4-3 Creating Perspective  ........................................ 102  
  Activity 4-4 Fully Articulated Animation  ................................ 106  
  Activity 4-5 Pixel Shading and Curved Surfaces ....................... 109  
  Activity 4-6 Pixel Mapping  .................................................. 114  
  Activity 4-7 Rendering of 3D Objects  ................................... 121  

Chapter 5  
**Collision Theory and Logic** .............................................. 125  
  Activity 5-1 Absolute and Relative Location  ........................... 129  
  Activity 5-2 The Games Factory 2 Basic Skills  ....................... 136  
  Activity 5-3 Logic and Collision in Practice  ........................... 147  
  Activity 5-4 Game Review  .................................................. 169  
  Activity 5-5 Gamescape Design and Dimensions ....................... 171  
  Activity 5-6 Game Review  .................................................. 181  
  Activity 5-7 Programming a Random-Number Generator .............. 183  
  Activity 5-8 Functionality Testing  ....................................... 194
<table>
<thead>
<tr>
<th>Chapter 6</th>
<th>Game Systems, Personal Computers, and Hardware</th>
<th>197</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activity 6-1 Hardware Terminology</td>
<td>201</td>
</tr>
<tr>
<td></td>
<td>Activity 6-2 Introduction to Sound Engineering</td>
<td>205</td>
</tr>
<tr>
<td></td>
<td>Activity 6-3 Balloon Pop (2D Shooter) Game</td>
<td>211</td>
</tr>
<tr>
<td></td>
<td>Activity 6-4 Sound Engineering and Editing</td>
<td>233</td>
</tr>
<tr>
<td></td>
<td>Activity 6-5 Game Review</td>
<td>236</td>
</tr>
<tr>
<td></td>
<td>Activity 6-6 Castle Escape Platform with Depth</td>
<td>238</td>
</tr>
<tr>
<td></td>
<td>Activity 6-7 Game Review</td>
<td>253</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 7</th>
<th>Play and Game Culture</th>
<th>255</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activity 7-1 Addiction</td>
<td>259</td>
</tr>
<tr>
<td></td>
<td>Activity 7-2 Design-Centric Promotional Game</td>
<td>261</td>
</tr>
<tr>
<td></td>
<td>Activity 7-3 Game Review</td>
<td>282</td>
</tr>
<tr>
<td></td>
<td>Activity 7-4 Promotional Theme Game</td>
<td>284</td>
</tr>
<tr>
<td></td>
<td>Activity 7-5 Peer Evaluation</td>
<td>286</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 8</th>
<th>Reverse Engineering and Professional Reviews</th>
<th>289</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activity 8-1 Professional Reviews</td>
<td>293</td>
</tr>
<tr>
<td></td>
<td>Activity 8-2 Reverse Engineer and User Review</td>
<td>294</td>
</tr>
<tr>
<td></td>
<td>Activity 8-3 Personal Reviews</td>
<td>296</td>
</tr>
<tr>
<td></td>
<td>Activity 8-4 Sprite Animation with Hot Spots and Action Points</td>
<td>298</td>
</tr>
<tr>
<td></td>
<td>Activity 8-5 Retro Platform Game</td>
<td>316</td>
</tr>
<tr>
<td></td>
<td>Activity 8-6 Evaluation</td>
<td>332</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 9</th>
<th>Large-Scale Design Process</th>
<th>335</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activity 9-1 Design Documents</td>
<td>339</td>
</tr>
<tr>
<td></td>
<td>Activity 9-2 Character Design: Avatar</td>
<td>345</td>
</tr>
<tr>
<td></td>
<td>Activity 9-3 Changeable Pursuit: Enemies to Prey</td>
<td>260</td>
</tr>
<tr>
<td></td>
<td>Activity 9-4 Evaluation</td>
<td>370</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 10</th>
<th>Global Economy and Supply Chain</th>
<th>373</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activity 10-1 Video Game Economy and Market Research</td>
<td>377</td>
</tr>
<tr>
<td></td>
<td>Activity 10-2 Keytar (Keyboard + Guitar) Hero</td>
<td>381</td>
</tr>
<tr>
<td></td>
<td>Activity 10-3 Evaluation</td>
<td>396</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capstone Project</th>
<th></th>
<th>399</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activity 1 Design Submission</td>
<td>400</td>
</tr>
<tr>
<td></td>
<td>Activity 2 Design Planning</td>
<td>403</td>
</tr>
<tr>
<td></td>
<td>Activity 3 Game Build and Tuning</td>
<td>408</td>
</tr>
<tr>
<td></td>
<td>Activity 4 Evaluation</td>
<td>411</td>
</tr>
<tr>
<td></td>
<td>Activity 5 Job Evaluations</td>
<td>413</td>
</tr>
</tbody>
</table>
Chapter 5
Collision Theory and Logic

Objectives
After completing this chapter, you will be able to:
- Use computer tools to create programming and artwork for a video game.
- Create a playable game using a game engine.
- Create animated objects.
- Debug a video game build.
- Engage in constructive criticism.
- Explain different user interface designs for avatar movement.
Activity 5-3

Logic and Collision in Practice

Objective
Students will use a game engine to create a playable video game. Students will be able to apply logic and collision theory in game design and user interface design.

Situation
You are a new employee of the Really Cool Game Company. Your first task is to learn the company’s game engine technology so you can build logic and collision statements. Complete the task provided to build design skills. You will create a game where the player must “eat” fruit falling from a plane while avoiding balls.

How to Begin
Launch The Games Factory 2 (TGF2). Your instructor will provide specific instructions for doing this, if needed.

Setting the Background Color
1. Click the New button on the Standard toolbar to begin working on a new game.
2. Double-click the thumbnail for frame 1 to begin making a title page. See Figure 1.
3. In the View pull-down menu, select Toolbars and make sure all of the toolbars are checked except the Layers toolbar.
4. In the Workspace window, right-click on the Frame 1 branch and select Rename. The name is replaced with a text box with the existing name highlighted. Enter Title Frame and press the [Enter] key to change the name.

TGF2 is an object-oriented program. This means you will be referring to the names of various objects as you design a game. For this reason, it is very important to name the objects in the game build using logical names. The names Button1, Button2, and Button3 are not very meaningful. The names Close Button, Fire Button, and Move Button are easily identified.
5. With the Title Frame branch selected, pick the Settings tab in the Properties window, if it is not already displayed.
6. Locate the Background color property, which sets the background color of the frame. Then, click the small, white color swatch in the right-hand column. This displays the color palette. See Figure 2.
7. Select the bright blue color swatch. The RGB value of this color swatch is 0,0,255. You may need to resize the Properties window to see the entire color value.

Adding a Title
You have now set the background color for the first frame of the game. The entire frame should be bright blue. Now you will add a title. First, you need to create a place to write the title.
8. In the Insert pull-down menu, select New Object. The Create New Object dialog box is displayed.
Figure 1

Click to start a new game
Double-click the thumbnail

Figure 2

Background color property
Pick the bright blue color swatch
9. In the Create New Object dialog box, select All Objects in the list on the left-hand side of the dialog box. Then, select Static Text on the right-hand side of the dialog box. See Figure 3.

10. Click the OK button to close the dialog box. Then, click near the top, center of the frame to place a text box on the title page.

The exact location of the text box is not important at this time. It can be moved later. After the text box is placed, it is automatically selected. Notice that it is numbered (1). Also, notice the tree in the Workspace window has grown with the addition of a Static Text branch. This branch is the text box just added.

11. Double-click on the text box. The Enter New Text dialog box is displayed. Enter Eating on the Run and pick the OK button to add the text.

12. Click anywhere off of the frame background to deselect the text box. See how the text appears.

13. Select the text box by clicking on it once.

14. In the Settings tab of the Properties window, uncheck the Border check box. Also, change the Background color property to the same bright blue used earlier.

15. In the Text Options tab of the Properties window, click on the right-hand column for the Font property. This displays the Font dialog box. Select Arial in the Font: list, Bold in the Font Style: list, and 22 in the Size: list. Then pick the OK button to change the font.

16. Also in the Text Options tab of the Properties window, change the Color property to white.

17. In the Movement tab of the Properties window, make sure the Type property is set to Static. If not, click on the right-hand column for the property to display a drop-down list and select Static.

18. In the Runtime Options tab of the Properties window, make sure the Create at start check box is checked.

19. In the editor window, click on the text box to show the sizing handles.

20. Stretch the text box to the left or right so it is big enough to show the entire title, as shown in Figure 4.
Saving Your Work

Now is a good time to save your work. It is a good practice to save after completing a major step in a design. It is also good practice to save at a regular time interval, such as every 15 minutes.

21. Click the **Save** button on the **Standard** toolbar. Since this file has not yet been saved, the **Save As** dialog box is displayed.

22. In the **Save As** dialog box, enter **Eat and Run** in the **File name** text box. Also, navigate to the folder designated by your instructor. Finally, pick the **Save** button to save the game.

Now that the game has been saved, clicking the **Save** button on the **Standard** toolbar simply saves the file. The **Save As** dialog box will not be displayed unless **File > Save As** is selected in the pull-down menu.

Creating a Start Button

Next, you will insert a **Start** button into the game interface. This is the button the player will click to begin the game.

23. In the **Insert** pull-down menu, select **New Object**. This opens the **Create New Object** dialog box.

24. On the left-hand side of the **Create New Object** dialog box, select **Interface**. Then, on the right-hand side of the dialog box, select **Button**. Finally, pick the **OK** button to close the dialog box.
25. Click anywhere on the frame to place the button. The location is not important since you can move it later.

26. Double-click on the button just inserted to open the Enter New Button Text dialog box. Change the text to Start and pick the OK button to close the dialog box and update the button.

27. In the Workspace window, right-click on the Button branch and select Rename from the shortcut menu. Rename the branch Start Button and press the [Enter] key, as shown in Figure 5.

Is the button ready to go? Recall the discussion in the textbook reading. Just because this object looks like a Start button does not mean it will act like a Start button. It must be programmed to function as a Start button. Using the event editor in TGF2, you will program the button to move the player to the next game frame when the button is clicked. The logic statement is: IF the button is clicked, THEN move the player to the next frame.

28. Click the Event Editor button on the Navigate toolbar.

29. Click New condition in the first row.

30. In the New Condition dialog box, right-click on the button to display a shortcut menu. Select Button Clicked? in the shortcut menu, as shown in Figure 6.
Collision Theory in Practice

The first logic statement will deal with the avatar object. IF the player character collides with a banana, THEN the banana is destroyed AND the player 1 score is increased by 10 points. To make this happen, it must be programmed into the game using the event editor. Click the Event Editor button to display the event editor.

Notice there are no events, even though you already added one for the Start button. This is because frame 2 (Game Frame) is the current frame. If you right-click on frame 1 (Title Frame) in the Workspace window tree and select Event Editor from the shortcut menu, the event for the Start button is displayed. Make sure the event editor for frame 2 is current and continue as follows.

75. Click on New condition.

76. In the New Condition dialog box, right-click on the player character object and select Collisions > Another object from the shortcut menu. The Test a Condition dialog box is displayed.

77. In the Test a Condition dialog box, select the banana and click the OK button, as shown in Figure 15.

78. Right-click in the cell under the banana column and select Destroy from the shortcut menu, as shown in Figure 16. This means that IF the player character collides with the banana, THEN the banana will be destroyed, not the player character.

79. Right-click in the cell under the Player 1 column and select Score > Add to Score from the shortcut menu. In the Add to Score dialog box that appears, enter 15 and click the OK button, as shown in Figure 17.
You have added the IF side of the logic statement, or the condition. Now, you need to add the events, which are the THEN side of the logic statement.

31. In the Storyboard Controls column, right-click in the cell for the Button clicked row.
32. Select Next frame from the shortcut menu, as shown in Figure 7.

**Inserting Objects**

Now, you need to place more objects on your title frame. Switch to the frame editor by picking the Frame Editor button on the Standard toolbar. Then, continue as follows.

33. In the Library window, expand the Local Library branch by clicking on the plus sign (+).
34. Expand the Games branch and select the Miscellaneous branch.
35. Double-click on the Fruits library file in the right-hand side of the Library window. The available objects in the Fruits library file are displayed, as shown in Figure 8.
36. Drag-and-drop one each of these objects onto the title frame: SApple2, S Banana1, S Cherry2, and S Grape.

37. Place a text box (static text) below each fruit object. Remember, to insert static text, select **Insert > New Object** and then select **Static Text** in the **Create New Object** dialog box.

38. Change the text displayed in the text boxes to: Cherry: 5 points, Apple: 10 points, Banana: 15 points, and Grapes: 20 points. Remember, to change the text, double-click on the text box and enter the new text in the dialog box that appears.

39. Using the **Properties** window, change the properties of each text box so it looks good on screen. For example, you may want to change the font size or color or the background color. Also, resize the text boxes as needed using the handles.

40. Align the fruit and static text objects in a row near the top of the frame. Select each object by clicking on it, then drag it into position. Multiple objects can be selected by holding down the [Shift] key. You can also use the arrow keys to nudge the objects as needed.

41. In the **Library** window, move up one level to see all of the library files in the **Miscellaneous** branch. Then, double-click on the **Game Objects 2** file.

42. Drag-and-drop one each of these objects onto the game frame: Ball 1, Ball 4, and Ball 8.

43. Add a static text box to label these objects as **Avoid These Objects**, as shown in **Figure 9**.

**Selecting a Player Avatar**

Time to choose an avatar for the player. The avatar is how the player character appears in the game. The player will move the player character around the game frame to gobble up the fruit while avoiding the balls.

44. In the **Library** window, select the **Characters** branch in the **Games** branch. Then, double-click on the **Various Characters** library file.

45. Drag the **Soldier** object and drop it at the bottom-right corner of the game frame.

46. Resize the character by clicking on it to display the handles. Then, drag the handles to make the object bigger. Its height should be roughly 1/4 of the frame height. Be sure to use the corner handles to resize the object so it remains proportional.
80. Using the same techniques, set the events for the cherry, apple, and grapes. Add the number of points indicated in the labels on the title slide. If a different number of points is added, it would be considered a bug or glitch in the game.

The basic collision theory here will make the fruit go away when it touches the character. When this happens, points are added to the player’s score. The balls are a little different. The logic statement for these is: IF the player character collides with a ball, THEN the player loses one life AND the ball is destroyed.

81. Click on New condition.

82. In the **New Condition** dialog box, right-click on the player character and select **Collisions > Another object**.

83. In the **Test a Collision** dialog box, select the first ball and click the **OK** button.

The **IF** side of the logic statement is set. Notice that it is similar to the logic statements for the fruit objects. The **THEN** side of the logic statement is different from the fruit objects.

84. Right-click in the cell under the **Player 1** column for the collision statement between the player character and the ball. Select **Number of lives > Subtract from number of lives** from the shortcut menu, as shown in **Figure 18**.
47. In the **Properties** window for the soldier object, select the **Movement** tab. Then, click on the **Type** property, and select **Path** from the drop-down list that is displayed.

48. Click the **Edit** button next to the **Edit Movement** property. The **Path Movement Setup** dialog box is displayed, as shown in **Figure 10**.

49. Click the **New Line** button in the **Edit Movement** dialog box. A line appears attached to the cursor and the object. This will allow you to draw the path the object will travel.

50. Move the cursor near the left edge of the frame. Keep the line very straight across the screen. Click to create the path. The object will follow this line when the game is played, as shown in **Figure 11**.

51. In the **Path Movement Setup** dialog box, change the **Speed:** setting to 15.
The lower the speed number, the slower the object will move. Make sure the path is selected (flashing white) when setting the speed setting. If necessary, click the first endpoint node (on the object) to select the path.

52. Click the **Reverse at End** button to have character turn around when it reaches the end of the path.

53. Click the **Loop the Movement** button to have the character walk back and forth along the path.

54. Click the **OK** button to close the **Path Movement Setup** dialog box.

55. Test play the frame. How does it all look?

56. Click the Windows close button (red X) to close the frame preview.

Remember, you should rename each object to a logical name. Doing so will help keep all of the elements of the game organized. This is a good time to rename all objects you have added to the game. Also, if needed, adjust the location of the objects. For example, the avatar may be walking over the **Start** button. Be sure to save your work before continuing.

**Game Frame**

What have you created so far? You have a frame that looks pretty and contains some animation. Is this a video game? No, it needs to be interactive to be a game.

57. Click the **Storyboard Editor** button to display the storyboard. This is where you will add a new frame.
85. In the **Set Number of Lives** dialog box that is displayed, enter 1 and click the **OK** button.
86. Right-click in the cell under the **Ball 1** column (the name of the ball object selected earlier) and select **Destroy** from the shortcut menu.
87. Follow similar steps to create the **IF** side conditions for the collision statement for each ball.
88. Locate the check mark you created to destroy **Ball 1**. Click and hold and drag it to the **Ball 4** column in line with the collision statement for the collision between the player character and **Ball 4**. Dragging a check mark copies it and applies the programming to the other object. Also drag the check mark from the **Player 1** column into the row for the collision with **Ball 4**.
89. Drag the check marks from **Ball 1** to the correct place to finish the **Ball 8** collision statement.

**The Drop**

To have the objects drop from the sky, there needs to be an object to generate the drop. You now will add a plane to fly across the screen and drop the objects.
90. Select **Game Frame** in the **Workspace** window. Then, pick the **Frame Editor** button.
91. In the **Library** window, select **Games > Planes**. Then, double-click on the 3D Jet - **Airline** library file. Finally, select the object **Fighter 4** and drag it to the top-left corner of the game frame, as shown in **Figure 19**.
58. Click on the label for the second row to add a new frame (row), as shown in Figure 12.
59. Double-click on the thumbnail for frame 2 to open the frame in the frame editor.
60. Change the name of the frame to Game Frame.
61. Change the background color of the frame to the same blue used on the title frame.

You will use a shortcut to save time in putting your objects in the game frame. Look at the Workspace window. All of the objects you have added are listed in the Title Frame branch. You can drag any of these objects onto the current frame. Here is an example of why it is good practice to rename objects. If you did not rename the static text objects, you would have six objects all named Static Text.
62. Drag the player character object from the Workspace window into the editor window and drop it near the bottom-center of the frame.
63. Drag the apple object into the editor window, but do not put it on the actual frame. By placing it in the editor, but not on the actual frame, it can be set to drop from the top of the screen.
64. Drag the banana, cherry, grapes, and three ball objects into the editor window, but do not drop them onto the actual game frame, as shown in Figure 13.
65. Select the player character object in the editor window.
66. Click the Movement tab in the Properties window.
67. Click the Type property and select Mouse Controlled from the drop-down list.
68. Click the Edit button for the Edit movement property.

The Mouse Movement Setup dialog box is displayed. Also, a bounding box with handles is displayed around the player character object (avatar). This box defines the space within which the object can move.
69. Using the sizing handles, stretch the box to the left and right edges of the frame. Leave 1/4" space at the edge. Move the top and bottom edges of the bounding box to the top and bottom of the avatar’s head, as shown in Figure 14.
70. Click the OK button in the Mouse Movement Setup dialog box to close it.
71. Test play the frame.
72. Move the mouse to see if the player character stays on screen.
73. Press the [Alt]+[F4] key combination to close the preview window.
74. Edit the movement if needed to keep the player character on screen.
Objects placed off of the actual frame

Drag from the tree to the frame

Player character is on the actual frame

Resize the bounding box
92. Select the plane object and set its movement property to Path.
93. Draw a straight-line path for the object across the top of the game frame.
94. Set the speed to 20.
95. Click the Reverse at End and the Loop the Movement buttons. Then, close the Path Movement Setup dialog box.

Create a New Object

All of the objects needed for the game are in place. Now, you only need to set the events to make it all work. The event editor is where this is done. The first logic statement is: IF 3.4 seconds elapse, THEN create a banana that will fall from the fighter.
96. Click the Event Editor button. Make sure the event editor is displayed for the game frame. If not, right-click on the Game Frame branch in the Workspace window and select Event Editor from the shortcut menu.
97. Click on New condition.
98. In the New Condition dialog box, right-click on the timer and select Every from the shortcut menu, as shown in Figure 20.
99. In the Every dialog box that is displayed, enter 3 in the Seconds text box, 40 in the 1/100 text box, and pick the OK button.
100. Right-click in the cell in the Create New Objects column for the condition you just added and select Create Object from the shortcut menu.
101. In the Create Object dialog box that is displayed, select the banana and click the OK button.

The frame editor is displayed and a different Create Object dialog box is opened, as shown in Figure 21. Now, you need to tell the computer where the banana will start to fall. It should fall from the exact spot where the fighter is when the timer counts off 3.4 seconds.
102. In the Create Object dialog box, select the Relative to... radio button. The Create Object dialog box is hidden and the Choose an Object dialog box is displayed.
103. In the **Choose an Object** dialog box, select the plane and click the **OK** button. The **Create Object** dialog box is redisplayed.

104. Set the X and Y coordinates to 0 and 0 to make the object appear in the center of the plane. Click the **OK** button to close the **Create Object** dialog box.

You are returned to the event editor. Using similar steps, create an event for each of the fruit and ball objects. Drop a cherry every 1.10 seconds. Drop an apple every 5.00 seconds. Drop a grape every 6.80 seconds. Notice how the higher-point-value objects appear less frequently? Drop one type of ball every 7.20 seconds. Drop a second type of ball every 8.10 seconds. Drop the third type of ball every 10.30 seconds. Save your work.

**Setting the Object Speed**

Now, you need to set the speed for the fruit and ball objects. You will also set the motion for the objects. Switch to the frame editor for the game frame. Then, continue as follows.

105. Select the banana.

106. In the **Properties** window, select the **Movement** tab.

107. Change the **Type** property to **Bouncing Ball**. A new set of properties is displayed in the **Properties** window.
108. Change the **Speed** property to 12.
109. Click the **Initial Direction** property.

   A compass window is displayed, as shown in **Figure 22**. The wheel selects the direction the object will start. Multiple directions can be set to randomize the movement. To clear all the arrows, click the **Reset** button at the lower-left corner of the window.
110. Click the blocks two to the left and right of the six o’clock position. You should have two arrows.

   Using similar steps, change the remaining objects. All should be set to bouncing-ball motion with the same initial directions. The cherry speed should be 30, the apple 60, and the grape 75. The three balls should have speeds of 65, 75, and 85. Save your work.

### Setting Boundaries

Right now, the objects will fall off of the screen edges. However, they should bounce around the screen as if the edges are the bumpers on a pool table. A boundary needs to be set to keep the objects from falling off of the edge of the screen. In the event editor for the game frame, set a new condition for this logic statement: **IF** the banana leaves the play area to the left or right, **THEN** bounce.

111. Click on **New Condition**.

112. In the **New Condition** dialog box, right-click on the banana and select **Position > Test Position of SBanana1**. The **Test Position of object** dialog box is displayed, as shown in **Figure 23**.
113. Click the left and right arrows for Leaves play area on side?.
114. Click the OK button to close the Test Position of object dialog box.
115. Right-click in the cell under the banana and select Movement > Bounce from the shortcut menu.

Using similar steps, set a left and right boundary for the cherry, apple, grape, and three types of balls. To save time, you can drag and drop the check marks. This will copy the reaction programming to the other item. In this case, the properties of the fruit are changed to act like bouncing balls and have them bounce at the edge of the frame. This keeps them in play until they reach the bottom of the screen. Test play the frame to see if everything is working properly. Use the [Alt][F4] key combination to close preview window. Save your work.

**Showing the Score and Number of Lives**

On the game frame, there needs to be some way to show the player how many lives are left. There also needs to be a way of displaying the score to the player. You will insert new objects for these tasks. Switch to the frame editor for the game frame.

116. Select New Object from the Insert pull-down menu.
117. In the Create New Object dialog box, select Games > Score and then click the OK button.
118. Click in the bottom-right corner of the frame to place the score object.
119. Select New Object from the Insert pull-down menu.
120. In the Create New Object dialog box, select Games > Lives and then click the OK button.
121. Click in the bottom-left corner of the frame to place the lives object, as shown in Figure 24.
The default value for the lives object is three lives. This is indicated by the three hearts. The number of lives and the icon indicator can be changed. Test the score and lives features using the Run Application tool. Run into some scoring objects and life-removing obstacles. Of course, you still need to add programming to end the game when you are out of lives. Close the preview window and continue as follows.

122. Display the event editor for the game frame.

123. Add a new condition. In the New Condition dialog box, right-click on the Player 1 icon and select When number of lives equals 0 from the shortcut menu.

124. In the cell under the Storyboard Controls column, right-click and select Next Frame from the shortcut menu.

Hall of Fame/High Score Page

125. Switch to the storyboard editor and create a new frame. Review the previous steps if you do not recall how to do this. Rename the frame as High Score.

126. Open the frame editor for the High Score frame.

127. In the Library window, expand the Backgrounds branch, select the Other branch, and open the Games—General Playareas library file.

128. Drag the Playtime Background from the library into the editor window. Align the image so it completely fills the frame.

129. Select New Object in the Insert pull-down menu. In the Create New Object dialog box, select Games > Hi-Score and click the OK button.

130. Click in the center of the background to place the high score object.

131. Resize the high score object so it covers about 2/3 of the dark gray area of the background. Also, center the object left-to-right on the dark gray area. Leave some room at the bottom for two buttons.

132. Using the Properties window, change the font size, colors, etc., to suit your tastes. See Figure 25.

133. Insert a button and change the name to Restart. Review the previous steps if you do not recall how to do this. Place the button below the high score object.

134. Insert another button and change the name to End Game.

It is good practice to make the buttons appear the same. They should have the same font, be the same size, and their positions should be balanced on the frame. Review various software that you use. Make note of how buttons appear similar and are consistently located. Users expect certain buttons, such as OK or Cancel, to be near the bottom of a dialog box. Also, you should rename the buttons in the Workspace window. Next, use the event editor to program these two new buttons.

135. Add a new condition. In the New Condition dialog box, right-click on the Restart button icon and select Button Clicked? from the shortcut menu.

136. Right-click in the cell under the Storyboard Controls column and select Restart the Application from the shortcut menu.

137. Add another new condition. In the New Condition dialog box, right-click on the End Game button icon and select Button Clicked? from the shortcut menu.

138. Right-click in the cell under the Storyboard Controls column and select End the application from the shortcut menu.

139. Test the game using the Run Application tool. See Figure 26.
Pause or End Game Application

One of the finishing touches is to add the ability to pause or end the game with the touch of a single button. Display the event editor for the game frame. Try programming the game to meet the following logic statements. Use what you have learned in this activity. Hint: you are testing the condition of a keyboard key.

○ IF on pressing the [Esc] key, THEN end the application.
○ IF on pressing the [F] key, THEN pause the application.

Create as Stand Alone

Once you finish designing your game, you want to make it so anyone can play. You will now “package” the game so it will play on any computer. Save your work before continuing.

140. In the File pull-down menu, select Build > Application. The Save As Stand-Alone Application dialog box is displayed. This is a standard Windows save as dialog box.

141. Navigate to the folder where you want the game file saved.

142. In the File name: text box, enter Eat and Run. Since this is the name of your TGF2 file, it is the default name. You can enter a different name if you wish.

143. In the Save as type: drop-down list, make sure Stand-alone Applications (*.exe) is selected. Then, click the Save button.

The game is saved as a self-contained EXE file. See Figure 27. You can now burn the game to a CD or attach it to an e-mail for friends to play.
Review Questions

1. Write a logic statement to describe how to program a Start button to move to the next frame.
   
   IF the user clicks the Start button,
   
   THEN display the next game frame.

2. Under which event button are the controls to add points?
   
   Player 1 (any player button)

3. What is the name of the frame that should include general directions so the player knows what to do?
   
   Title Frame

4. Which button allows the designer to test play just the frame they are working on?
   
   Run Frame

5. Where do you place an object if you do not want it to show on the frame until programmed to drop?
   
   Outside of the visible game frame.

6. What type of movement allows the designer to draw a line for the object to follow?
   
   Path

7. What type of condition is used to make the banana disappear when it touches the player?
   
   IF the player collides with a banana

8. What was the user interface for this game?
   
   The mouse, keyboard, and computer monitor.

9. How do you quickly copy an event from one place to another?
   
   By dragging the check mark.

10. What object is needed to record the scores of the best players?
    
    The hi-score object.