# Chapter 6

# Working with Graphics

## Activity 6.1: Introducing Canvas

### Problem Statement

The form for booking movie tickets has been created on the Online Booking Web page. Now, you want to display the movie ratings to the audience on that page. The movie ratings on the latest movies should be displayed in the form of bar graphs. The rating should be given on the scale of 1 ‑ 5, where 1 stands for Below Average and 5 stands for Extraordinary.

How will you perform this task?

**Prerequisite**: To perform this activity, you need to use the **ShowOnWheels** file that you have created in Activity 5.2. In addition, you need the **RGraph.bar.js** and **RGraph.common.core.js** files and save them in the **ShowOnWheels** folder. Ask your faculty to provide you the same.

### Solution

To display the movie ratings in the form of a bar graph, you need to perform the following tasks:

1. Update the OnlineBooking.html file.
2. View the movie ratings.

#### Task 1: Updating the OnlineBooking.html File

To update the **OnlineBooking.html** file, you need to perform the following steps:

1. Open the **OnlineBooking.html** file.
2. Add the following code snippet after the <P>No refund will be granted on cancellation of tickets .</P></DIV> line of code:

<BR><BR><BR>

<H1> Movie Ratings</H1>

<CANVAS width="900" height="250" id="test" style="border:1px solid black"></CANVAS>

The preceding code snippet creates a heading and a canvas on the OnlineBooking Web page along with its specified width and height.

1. Add the following code snippet after the <CANVAS width="900" height="250" id="test" style="border:1px solid black"></CANVAS> line of code:

<SCRIPT type="text/javascript" charset="utf-8">

var bar = new RGraph.Bar('test', [4,2.5,2.5,3,4,3.5]);

bar.Set('chart.colors', ['green']);

bar.Set('chart.labels', ["Source Code" , "if Only" , "Angel" , "Nightmare","Along Polly Came","A Walk to Remember" ]);

bar.Draw();

</SCRIPT>

The preceding code snippet creates a bar graph on the canvas.

1. Add the following code snippet after the <HEAD> tag:

<SCRIPT type="text/javascript" src="RGraph.common.core.js"></SCRIPT>

<SCRIPT type="text/javascript" src="RGraph.bar.js"></SCRIPT>

1. Save and Close the **OnlineBooking.html** page.

#### Task 2: Viewing the Movie Ratings

To view the movie ratings, you need to perform the following steps:

1. Open Google Chrome.
2. Type **D:\HTML\mainframe.html** in the address bar.
3. Press the **Enter** key. The Home Page appears, as shown in the following figure.

*The Home Page*

1. Click the **Online Booking** tab.
2. The movie ratings in the form of bar graph appear on the Web page, as shown in the following figure.

*The Online Booking Page*

1. Close the Web browser.

## Activity 6.2: Creating a Game

### Problem Statement

You work as a Web developer at GameMantra. The organization has a website that enables users to play online games. You have been assigned a task to create a game. The game should contain two characters, dave and monster. The dave character should try to catch as many monsters as it can in 30 seconds. After 30 seconds, the game should be over. In addition, if the dave character goes out of the game boundary, he should be dead and the game should be over. How will you create this game?

**Prerequisite**: You need to use the **Images** folder that contains the images required to create the game. Ask your faculty to provide you the same. You need to save this folder at the **D:\HTML\Game** location.

### Solution

To create the required game, you need to perform the following tasks:

1. Create the game.js file.
2. Create the home.html file.
3. Run the game.

#### Task 1: Creating the game.js File

To create the **game.js** file, you need to perform the following steps:

1. Open Notepad.
2. Add the following code snippet in Notepad:

// Create the canvas

var canvas = document.createElement("canvas");

var ctx = canvas.getContext("2d");

canvas.width = 512;

canvas.height = 480;

document.body.appendChild(canvas);

// Background image

var bgReady = false;

var bgImage = new Image();

bgImage.onload = function () {

bgReady = true;

};

bgImage.src = "images/background.png";

The preceding code snippet creates a canvas of specified height and width. In addition, it adds a background image to the canvas.

1. Add the following code snippet after the bgImage.src = "images/background.png"; line of code:

// Dave image

var daveReady = false;

var daveImage = new Image();

daveImage.onload = function () {

daveReady = true;

};

daveImage.src = "images/dave.png";

// Monster image

var monsterReady = false;

var monsterImage = new Image();

monsterImage.onload = function () {

monsterReady = true;

};

monsterImage.src = "images/monster.png";

The preceding code snippet adds the images of the characters to the canvas.

1. Add the following code snippet after the monsterImage.src = "images/monster.png"; line of code:

// Game objects

var dave = {

speed: 256 // movement in pixels per second

};

var monster = {};

var monstersCaught = 0;

var timeLeft = 30;

// Handle keyboard controls

var keysDown = {};

addEventListener("keydown", function (e) {

keysDown[e.keyCode] = true;

}, false);

addEventListener("keyup", function (e) {

delete keysDown[e.keyCode];

}, false);

The preceding code snippet creates game objects, hero and monster, and calls the key up and key down event listeners.

1. Add the following code snippet after the addEventListener("keyup", function (e) {delete keysDown[e.keyCode];}, false); line of code:

// Reset the game when the player catches a monster

var reset = function () {

dave.x = canvas.width / 2;

dave.y = canvas.height / 2;

// Throw the monster somewhere on the screen randomly

monster.x = 32 + (Math.random() \* (canvas.width - 64));

monster.y = 32 + (Math.random() \* (canvas.height - 64));

};

The preceding code snippet throws another monster anywhere on the screen when the monster is caught by the hero.

1. Add the following code snippet after the monster.y = 32 + (Math.random() \* (canvas.height - 64)); }; line of code:

// Update game objects

var update = function (modifier) {

//If the time is not over

if(timeLeft>0)

{

if(dave.y>0 && 38 in keysDown)

  { // Player holding up

    dave.y -= dave.speed \* modifier;

  }

  else if(dave.y<=0 && 38 in keysDown)

  {

    timeLeft=0;

    daveImage.src = "images/dead.png";

  }

if(dave.y <448 && 40 in keysDown) { // Player holding down

    dave.y += dave.speed \* modifier;

  }

  else if(dave.y >=448 && 40 in keysDown) { // Player holding down

    timeLeft=0;

    daveImage.src = "images/dead.png";

  }

if(dave.x>0 && 37 in keysDown) { // Player holding left

    dave.x -= dave.speed \* modifier;

  }

  else if(dave.x<=0 && 37 in keysDown)

  {

    timeLeft=0;

daveImage.src = "images/dead.png";

  }

if(dave.x<480 && 39 in keysDown) { // Player holding right

    dave.x += dave.speed \* modifier;

  }

else   if(dave.x>=480 && 39 in keysDown)

  {

    timeLeft=0;

daveImage.src = "images/dead.png";

  }

The preceding code snippet continues the game if the time is not over.

1. Add the following code snippet after the daveImage.src = "images/dead.png"; } line of code:

  // Are they touching?

  if (

    dave.x <= (monster.x + 32)

    && monster.x <= (dave.x + 32)

    && dave.y <= (monster.y + 32)

    && monster.y <= (dave.y + 32)

  ) {

    ++monstersCaught;

    reset();

  }

}

else

{

  clearInterval(timerGameMain);

  clearInterval(timerGame);

  alert("Game Over\nYou have caught " + monstersCaught + " monsters in 30 seconds.");

}

};

The preceding code snippet specifies that if the dave character touches the monster character, then the monster is caught.

1. Add the following code snippet after the alert("Game Over\nYou have caught " + monstersCaught + " monsters in 30 seconds."); } }; line of code:

// Draw

var render = function () {

if (bgReady) {

  ctx.drawImage(bgImage, 0, 0);

}

if (daveReady) {

  ctx.drawImage(daveImage, dave.x, dave.y);

}

if (monsterReady) {

  ctx.drawImage(monsterImage, monster.x, monster.y);

}

// Score

ctx.fillStyle = "rgb(250, 250, 250)";

ctx.font = "24px Helvetica";

ctx.textAlign = "left";

ctx.textBaseline = "top";

document.getElementById("infoDiv").innerHTML="<font size=+2><b>Monster Caught:</b> " + monstersCaught + "  <b>Time Left:</b> " + timeLeft + " Sec</font>";

};

The preceding code snippet shows the score of the game and the time left.

1. Add the following code snippet after the document.getElementById("infoDiv").innerHTML="<font size=+2><b>Monster Caught:</b> " + monstersCaught + "  <b>Time Left:</b> " + timeLeft + " Sec</font>"; }; line of code:

// The main game loop

var main = function () {

var now = Date.now();

var delta = now - then;

update(delta / 1000);

render();

then = now;

};

//Time Restriction

var timeRestrection= function ()

{

timeLeft--;

}

// Let's play this game!

reset();

var then = Date.now();

var timerGameMain=setInterval(main, 1); // Execute as fast as possible

var timerGame=setInterval(timeRestrection,1000);

The preceding code snippet adds the time restriction to the game and starts a new game.

1. Save the file with the name, **game.js**, at the **D:\HTML\Game** location.
2. Close the **game.js** file.

#### Task 2: Creating the home.html File

To create the **home.html** file, you need to perform the following steps:

1. Open Notepad.
2. Add the following code snippet in Notepad:

<!DOCTYPE HTML>

<HTML lang="en">

<HEAD>

  <META charset="utf-8">

  <TITLE>Simple Canvas Game</TITLE>

</HEAD>

<BODY>

<DIV id="infoDiv"></DIV>

  <SCRIPT src="game.js"></SCRIPT>

</BODY>

</HTML>

The preceding code snippet links the **game.js** file in the HTML file.

1. Save the file with the name, **home.html**, at the **D:\HTML\Game** location.
2. Close the **home.html** file.

#### Task 3: Running the Game

To run the game, you need to perform the following steps:

1. Type **D:\HTML\Game\home.html** in the address bar.
2. Press the **Enter** key. The game appears on a Web page, as shown in the following figure.

*The Game*

To move the dave character, you need to use the arrow keys. When the dave character touches the monster character, your points get increased. Ensure that the dave character does not touch the boundary as it will end your game.

1. Close the Web browser.

# Exercises

### Exercise 1

In the SkyLight University website, you need to create a Web page named Placements showing the name of the companies that are coming for students’ placements. In addition, you need to display the student placement ratio over the last six years. For this, you need to create a bar graph, as shown in the following figure.

*The Placements Web Page*

How will you perform the preceding task?

**Prerequisite**: To perform this exercise, you need to use the solution file created for Exercise 2 of Chapter 5. In addition, you need the **RGraph.bar.js** and **RGraph.common.core.js** files.

### Exercise 2

Advin is a Web designer and has been given a task to create a chess board. A chess board is a combination of eight rows and eight columns. For this, he needs to create a matrix of 8\*8 in canvas. In addition, he needs to apply alternate colors on the squares in the chess board, i.e. black and white, as shown in the following figure.

*The Chess Board*

Help Advin to perform this task?

### Exercise 3

You have been assigned a task to create a smiley on canvas. The face of the smiley should be colored in yellow, with eye balls colored in blue, and mouth outline in red, as shown in the following figure.

*The Smiley*

How will you perform this task?

### Exercise 4

Advin is a Web developer and has been assigned a task to rotate an image on canvas, as shown in the following figure.

*The Rotating Car*

Help Advin to complete this task.

**Prerequisite**: To perform this exercise, you need to use the image file, **car.gif**.

### Exercise 5

### Exercise 6