

Web App Development Quick Start Guide (WebAppSample_TicTacToe)

Version 1.0.0 – December 2012

LGDEV-086

Home Entertainment Company
LG Electronics, Inc.

Copyright

Copyright © 2012 LG Electronics, Inc. All Rights Reserved.

Though every care has been taken to ensure the accuracy of this document, LG Electronics, Inc. cannot accept responsibility for any errors or omissions or for any loss occurred to any person, whether legal or natural, from acting, or refraining from action, as a result of the information contained herein. Information in this document is subject to change at any time without obligation to notify any person of such changes.

LG Electronics, Inc. may have patents or patent pending applications, trademarks copyrights or other intellectual property rights covering subject matter in this document. The provision of this document does not give the recipient or reader any license to these patents, trademarks copyrights or other intellectual property rights.

No part of this document may be communicated, distributed, reproduced or transmitted in any form or by any means, electronic or mechanical or otherwise, for any purpose, without the prior written permission of LG Electronics, Inc.

This document is subject to revision without further notice.

All brand names and product names mentioned in this document are trademarks or registered trademarks of their respective owners.

About This Document

Revision History

Document Version	Date	Comment
1.0.0	2012, December 13	Initial Release

Purpose

This document is a quick guide intended to describe how developers develop and test web applications using the WebAppSample_TicTacToe example. This will help developer to understand how to use Sound Plugin and API and background music in the application.

Reference Documents

Refer to the following documents:

- Developer Library in LG Developer (<http://developer.lge.com>) website

Conventions

Codes

Source code and examples are indicated in the `grey Courier New` font.

Note, Caution

Note and caution are used to emphasize information. The following samples describe when each is used.

NOTE

Contains information about something that is helpful to you.

CAUTION

Contains important information about something that you should know.

Abbreviation

The following table defines the abbreviations used in this document.

Abbreviation	Description
CSS	Cascading Style Sheet
SDK	Software Development Kit

Contents

1	Introduction.....	5
1.1	Overview of WebAppSample_TicTacToe	6
1.2	Structure and Resources of Sample Project.....	7
2	Setting Up the Project	8
3	Implementing the Code	9
3.1	Example Source Code.....	10
3.2	Descriptions on Source Codes	28
3.3	Expected Result	34
4	Testing the Application	37

Figures

[Figure 1]	WebAppSample_TicTacToe Player Selection	6
[Figure 2]	WebAppSample_TicTacToe Game Play	6
[Figure 3]	Structure and Resources of Sample Project.....	7

Tables

[Table 1]	Structure of Sample Project.....	7
-----------	----------------------------------	---



1 Introduction

This chapter describes WebAppSample_TicTacToe and its structure.

- 1.1 Overview of WebAppSample_TicTacToe
- 1.2 Structure and Resources of Sample Project

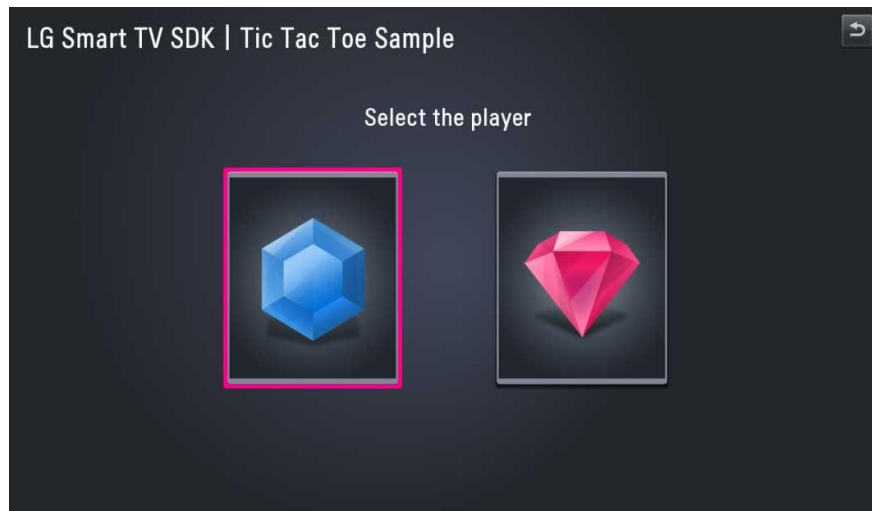
1.1 Overview of WebAppSample_TicTacToe

A web application that is developed based on general HTML standards has the following structure for its operation. If you separate each component independently, it is easy to manage application sources. Developers need to implement each component as necessary.

- **HTML: Defines application structure**
- **JavaScript: Defines application operation**
- **CSS: Defines application design elements**

In this guide, we are going to learn how to develop and test a web application using a simple development example that uses Sound Plugin and API and background music in the application.

The operation of WebAppSample_TicTacToe application is shown below. User can select the player and the other player is set to TV. The Media plays in the background and the click sound effect plays on click.



[Figure 1] WebAppSample_TicTacToe Player Selection

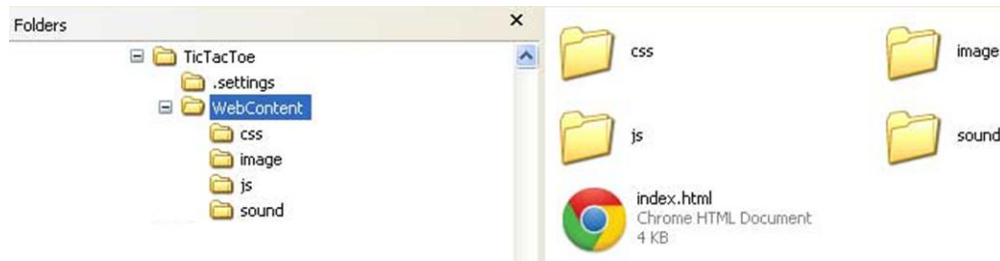


[Figure 2] WebAppSample_TicTacToe Game Play

1.2 Structure and Resources of Sample Project

The file structure of WebAppSample_TicTacToe that is provided as a sample with this guide is shown below. You can download the sample codes from **[Resource Center > Smart TV > SDK & Tools > Tools & Samples]** menu in LG Developer (<http://developer.lge.com>) website.

You can either copy this sample source into the WebContent folder of the project that is created in LG IDE or enter the code directly by referring to the description in this guide.



[Figure 3] Structure and Resources of Sample Project

[Table 1] Structure of Sample Project

Folder	File	Description
WebAppSample _TicTacToe	index.html	Main screen file of an application
	main.css	Design elements of an application
	keycode.js	Key code definitions and value assignments
	js keynavigation.js	Key navigation for main screen
	tic.js	Tic Tac Toe game logic.

2 Setting Up the Project

Although it is okay to use a general editor to develop a web application, you can easily develop, debug, and test the application if you use the LG IDE. This chapter describes how to set up a project using the LG IDE.

Note

- For detailed information on how to start using LG IDE, refer to **Developing > Using SDK** section in Developer Library in LG Developer (<http://developer.lge.com>) website.

1. Open LG IDE and click **[File > New > LG Web Project]**.
2. In the 'New LG Web Project' window, give a project name and select 'SDK Version for Web Application'. Then, click **[Finish]**.
3. Add JavaScript, CSS, and image files.

Create each folder for JavaScript, CSS, or image files that will be used in this example in the WebContent folder. Although folders are not mandatory, it will be helpful to manage resources using folders when considering resources increase in the future.

Select and right-click the WebContent node and select **[File]** and then enter a file name including its extension in the file name field to add JavaScript or a CSS file.

4. Now, a basic foundation is prepared to implement a sample example.



3 Implementing the Code

This chapter describes the source codes, description, and expected results of WebAppSample_TicTacToe example.

- 3.1 Example Source Code
- 3.2 Descriptions on Source Codes
- 3.3 Expected Result

Note

Refer to **Developing > API** section in Developer Library in LG Developer (<http://developer.lge.com>) website for detailed information on how to use Web APIs.

3.1 Example Source Code

The entire source codes of WebAppSample_TicTacToe sample application are as follows: Following is the description related to Main Screen.

[[index.html](#)]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>LG SmartTV : Broadcast</title>
<link type="text/css" href="css/style.css" rel="stylesheet"/>
<script src="js/jquery-1.8.2.min.js"></script>
<script src="js/channelList.js"></script>
<script src="js/keycode.js"></script>
<script src="js/common.js"></script>
<script src="js/displaychannedetails.js"></script>
<script src="js/keynavigation main.js"></script>
<script>
        new keyNavigation_main();
    </script>
</head>
<body>
<!-- Header -->
<div class="header header broad">
    <p class="title">
        LG Smart TV SDK | Broadcast Sample
    </p>
    <!-- main_BTN -->
    <ul class="main btn">
        <li id="back" class="btn default">
            <p class="btn_icon img_default">
                
            </p>
            <p class="btn_icon img_focus">
                
            </p>
        </li>
    </ul>
    <!-- main_BTN -->
</div>
<!-- //Header -->
<!-- contents -->
<div class="container broad">
    <!-- Contents_Left -->
    <div class="contents left">
        <!-- Left List -->
        <div class="broad list">
            <!-- ArrowUp -->
            <ul id="prev-page" class="btn dim">
                <li class="arrow default">
                    
                </li>
                <li class="arrow focus">
                    
                </li>
                <li class="arrow dim">
                    
                </li>
            </ul>
            <!-- //ArrowUp -->

            <div id="slider" class="ch list">
                <ul id="channelUL">
                </ul>
            </div>
            <!-- //ChannelList -->
            <!-- ArrowDown:btn focus/btn default -->
            <ul id="next-page" class="btn default">
```

```

        <li class="arrow_default">
            
        </li>
        <li class="arrow_focus">
            
        </li>
        <li class="arrow_dim">
            
        </li>
    </ul>
    <!-- //ArrowDown -->
</div>
<!-- //Left List -->
<!-- Left Button:ch btn focus/ch btn default -->
<ul id="navigation" class="list_btn">
    <!-- ChannelUp -->
    <li id="channelUP" class="ch_btn_default">
        <p class="up_default">
            <span class="up_focus bg">Channel Up<span>
        </p>
        <p class="up_focus">
            <span class="up_focus bg">Channel Up<span>
        </p>
    </li>
    <!-- //ChannelUp -->
    <!-- ChannelDown -->
    <li id="channelDOWN" class="ch_btn_default">
        <p class="down_default">
            <span class="down_focus bg">Channel Down<span>
        </p>
        <p class="down_focus">
            <span class="down_focus bg">Channel Down<span>
        </p>
    </li>
    <!-- //ChannelDown -->
</ul>
<!-- //Left Button -->
</div>
<!-- //Contents_Left -->
<!-- Preview:preview_focus/preview_default -->
<div class="broad preview preview focus">
    <div class="outline">
    </div>
    <div id="broadcastHolder" class="preview">
        <p id="channelDetailsHolder" class="info">
            <span class="title">Default</span>
            <span class="date">Default</span>
        </p>
        <object id="broadcast" class="broadcastIndex" type="application/x-
netcast-broadcast"></object>
        <div class="noSignalHolder" id="noSignalHolder">
        </div>
    </div>
</div>
<!-- //Preview -->
</div>
<!-- //contents -->
</body>
</html>

```

[keycode.js]

```
/**
**
// LCD TV LABORATORY, LG ELECTRONICS INC., SEOUL, KOREA
// Copyright(c) 2009 by LG Electronics Inc.
//
// All rights reserved. No part of this work may be reproduced, stored in a
// retrieval system, or transmitted by any means without prior written
// permission of LG Electronics Inc.
//
// Following key definitions and value assignments are for Netcast Browser.
//
//*****
**

var VK_ENTER           = 13;
var VK_PAUSE          = 19;
var VK_PAGE_UP        = 33;
var VK_PAGE_DOWN      = 34;
var VK_LEFT           = 37;
var VK_UP             = 38;
var VK_RIGHT          = 39;
var VK_DOWN           = 40;
var VK_0              = 48;
var VK_1              = 49;
var VK_2              = 50;
var VK_3              = 51;
var VK_4              = 52;
var VK_5              = 53;
var VK_6              = 54;
var VK_7              = 55;
var VK_8              = 56;
var VK_9              = 57;
var VK_RED            = 403;
var VK_GREEN          = 404;
var VK_YELLOW         = 405;
var VK_BLUE           = 406;
var VK_REWIND         = 412;
var VK_STOP           = 413;
var VK_PLAY           = 415;
var VK_FAST_FWD       = 417;
var VK_INFO           = 457;
var VK_BACK           = 461;
```

[keynavigation.js]

```

/*
File keynavigation.js
@Description - tic tac toe game logic and implementaion

*/
var selplayer;
var tv;
var allMenuObject = [];
var cntIndex = 0;
var gameStarted = false;
var gameEnded = false;
var backHighlight = 1;
//backHighlight - 1 : not highlighted,2:highlighted,3:gamestarted
/*
Function ready()
@param
@Description - This function is called when DOM is ready.This has the
implimentation of keydown
Returns -void
*/
$('document').ready(function () {

    $(".main btn").on("click", function (event) {
        backButtonSelect();
    });
    $(".main btn").on("mouseenter mouseleave", function (event) {
        highlightBack();
    });
    $("#players li img").each(function (i) {
        $(this).addClass("playerImgNormal");
        $(this).on("mouseenter mouseleave", function (event) {
            resetHighlight();
            $(this).toggleClass("playerImgHover");
            cntIndex = i;
        });
    });
    var elem = $("#players li img");
    $("#players li img").click(function () {
        playerClicked(this);
    });
    for (var i = 0; i < elem.length; i++) {
        allMenuObject.push(elem[i]);
    }

    $(allMenuObject[0]).toggleClass("playerImgHover");
    /*
Function keydown()
@param -event
@Description - This has the implementation of keydown like left right up
and down
Returns -void
*/
$(document).keydown(function (event) {

    var key = event.keycode || event.which;
    switch (key) {
    case VK_BACK:
        backButtonSelect();
        break;
    case VK_RIGHT:
        if (gameStarted) {
            squareMove("right");
        } else if (gameEnded) {
            if (backHighlight == 2) {
                highlightPrevious();
            } else if (backHighlight == 1) {
                highlightBack();
            }
        }
        } else {

```

```
        if (allMenuObject[cntIndex + 1] != undefined &&
backHighlight != 2) {
            cntIndex++;
            resetHighlight();
            $(allMenuObject[cntIndex]).toggleClass("playerImgHover");
        }
    }
    break;

    case VK LEFT:
        if (gameStarted) {
            squareMove("left");
        } else if (gameEnded) {
            if (backHighlight == 2) {
                highlightPrevious();
            } else if (backHighlight == 1) {
                highlightBack();
            }
        }

        } else {
backHighlight != 2) {
            cntIndex--;
            resetHighlight();
            $(allMenuObject[cntIndex]).toggleClass("playerImgHover");
        }
    }
    break;
    case VK UP:

        if (backHighlight == 1) {
            highlightBack();
        } else {
            squareMove("up");
        }
        break;
    case VK DOWN:
        if (backHighlight == 2) {
            highlightPrevious();
        } else {
            squareMove("down");
        }

        break;
    case VK ENTER:
        if (backHighlight == 2) {
            backButtonSelect();
        } else if (gameStarted) {
            selectSquare();
        } else if (gameEnded) {
            restart();
        } else {
            playerClicked(allMenuObject[cntIndex]);
        }
        break;
    }

    });

});
/*
Function backButtonSelect()
@param - none
@Description - This function implements back button functionality at
different stages.
Returns -void
*/

function backButtonSelect() {
    if (gameStarted || gameEnded) {
        history.go(0);
    } else {
```

```

        if (window.NetCastBack) {
            window.NetCastBack();
        }
    }
}
/*
Function resetHighlight()
@param -
>Description - This function implements resetting of the highlightig of
button on key press.
>Returns -void
*/
function resetHighlight() {
    $("#players li img").each(function (i) {
        $(this).removeClass("playerImgHover").addClass("playerImgNormal");
    });
    $(".main_btn li").removeClass("btn focus");
    $(".main_btn li ").addClass("btn_default");
}

/*
Function highlightBack()
@param -
>Description - This function implements highlightig of back button on down
key press.
>Returns -void
*/
function highlightBack() {
    backHighlight = 2;
    $(".main_btn li ").removeClass("btn_default");
    $(".main_btn li ").addClass("btn focus");
    if (gameStarted) {
        $("#highlight").css("display", "none");
    } else if (gameEnded) {
        $("#btn_restart").removeClass("buttonSelected").addClass("button");
    } else {
        $("#players li img").each(function (i) {

            $(this).removeClass("playerImgHover").addClass("playerImgNormal");
        });
    }
}

/*
Function highlightPrevious()
@param -
>Description - This function implements highlightig of the previous button
from back button on down key press.
>Returns -void
*/
function highlightPrevious() {
    backHighlight = 1;
    $(".main_btn li ").removeClass("btn_focus");
    $(".main_btn li ").addClass("btn_default");
    if (gameStarted) {
        backHighlight = 3;
        $("#highlight").css("display", "block");
        $("#highlight").css("left", position_arr[position_index].x);
        $("#highlight").css("top", position_arr[position_index].y);
    } else if (gameEnded) {
        $("#btn_restart").removeClass("button").addClass("buttonSelected");
    } else {
        $("#players li img").each(function (i) {

            $(this).removeClass("playerImgHover").addClass("playerImgNormal");
        });
        $(allMenuObject[cntIndex]).toggleClass("playerImgHover");
    }
}

/*
Function playerClicked()
@param -obj-the current selected HTML object

```

3 Implementing the Code

```
@Description - This function check the current selected item and marks the
other as TV.
Returns -void
*/
function playerClicked(obj) {
    backHighlight = 3;
    var dir = "Left"
    var selItem = obj;
    selplayer = selItem;
    playSound(1);
    $("#players li img").each(function (i) {
        var curItem;
        $(this).addClass("animateclass" + dir);
        $(this).off("mouseenter mouseleave click");
        $("#selectText h2").html("Play Now");
        if (this.alt != selItem.alt) {
            curItem = this;
            tv = curItem;
        }
        dir = "Right";
        this.addEventListener('webkitAnimationEnd',

            function (event) {
                $("#game").css("display", "block");

                if (curItem) {
                    $(".player_name").css("display", "block");
                    $(".player_name").css("position", "absolute");
                    $(".player_name").css("width", $(curItem).width());
                    $(".player_name").css("left", $(curItem).offset().left - 10);
                    $(".player_name").css("top", $(curItem).offset().top + 170);

                    paintBoard();
                }
            }, false);

    });
    allMenuObject = [];
}
/*
Function playSound()
@param -num -number of the sound element
@Description - This function is called for playing the sound effect..
Returns -void
*/
function playSound(num) {
    var sound = document.getElementById("sound" + num);
    sound.play();
}
```


[tic.js]

```

/*
File tic.js
@Description - tic tac toe game logic and implementaion

*/
var xBoard = 0;
var oBoard = 0;
var begin = true;
var context;
var width, height;
var position_arr =
[ {x:444,y:60}, {x:577,y:60}, {x:706,y:60}, {x:444,y:190}, {x:577,y:190}, {x:706,y:
190}, {x:444,y:323}, {x:577,y:323}, {x:706,y:323} ];
var bit_arr =
[ {x:0,y:0}, {x:1,y:0}, {x:2,y:0}, {x:0,y:1}, {x:1,y:1}, {x:2,y:1}, {x:0,y:2}, {x:1,y
:2}, {x:2,y:2} ];
var position_index = 0;

/*
Function paintBoard()
@param -none
@Description -creates the tic tac toe game board and marks the first play by
TV.
Returns -void
*/
function paintBoard() {
    gameStarted = true;
    gameEnded = false;
    $("#highlight").css("display", "block");
    $("#highlight").css("left", position_arr[position_index].x);
    $("#highlight").css("top", position_arr[position_index].y);
    $("#highlight").on({
        click : function (event) {
            selectSquare();
        }
    })

    $("#board").on({
        mousemove : function (e) {

            var y = Math.floor(e.offsetY / (height / 3));
            var x = Math.floor(e.offsetX / (width / 3));
            if (y == 0) {
                position_index = x;
            } else if (y == 1) {
                position_index = x + y + 2;
            } else if (y == 2) {
                position_index = x + y + 4;
            }
            if (!gameEnded) {
                $("#highlight").css("display", "block");
                $(".main_btn li").removeClass("btn_focus");
                $(".main_btn li ").addClass("btn_default");
            }
            $("#highlight").css("left", position_arr[position_index].x);
            $("#highlight").css("top", position_arr[position_index].y);

        }
    })
    var board = document.getElementById('board'); // $('#board')
    width = board.width;
    height = board.height;
    context = board.getContext('2d');

    context.beginPath();
    context.strokeStyle = "rgba(255,0,0,0)";
    context.lineWidth = 1;

    context.moveTo(2, 0);
    context.lineTo(2, height);

```

```
context.moveTo((width / 3), 0);
context.lineTo((width / 3), height);

context.moveTo((width / 3) * 2, 0);
context.lineTo((width / 3) * 2, height);

context.moveTo(width - 2, 0);
context.lineTo(width - 2, height);

context.moveTo(0, 2);
context.lineTo(width, 2);

context.moveTo(0, (height / 3));
context.lineTo(width, (height / 3));

context.moveTo(0, (height / 3) * 2);
context.lineTo(width, (height / 3) * 2);

context.moveTo(0, height - 2);
context.lineTo(width, height - 2);

context.stroke();
context.closePath();
if (begin) {
    var ini = Math.abs(Math.floor(Math.random() * 9 - 0.1));
    markBit(1 << ini, 'O');
    begin = false;
} else {
    begin = true;
}
}
/*
Function checkWinner()
@param -board : its either xBoard or oBoard
@Description -checks the winner .
Returns -result :Boolean
*/
function checkWinner(board) {

    var result = false;

    if ((board | 0x1C0) == board) || ((board | 0x38) == board) || ((board |
0x7) == board) || ((board | 0x124) == board) || ((board | 0x92) == board) ||
(board | 0x49) == board) || ((board | 0x111) == board) || ((board | 0x54) ==
board)) {

        result = true;
    }
    return result;
}
/*
Function paintX()
@param - x,y : position of selected box.
@Description - marks user play on the Board in the beginX,beginY .
Returns void
*/
function paintX(x, y) {
    var offsetX = (width / 3) * 0.01;
    var offsetY = (height / 3) * 0.01;
    var beginX = (x * (width / 3) + x) + offsetX;
    var beginY = (y * (height / 3) + y) + offsetY;
    // Create a new image.
    var imgX = new Image();

    // Once it's loaded draw the image on the canvas.
    imgX.addEventListener('load', function () {

        // Now resize the image: x, y, w, h.
        context.drawImage(this, beginX, beginY, 126, 126);
    });
}
```

```

    }, false);

    imgX.src = 'image/play ' + selplayer.alt + '.png';
}
/*
Function paintO()
@param - x,y : position of selected box.
@Description - marks TV play on the Board in the beginX,beginY .
Returns void
*/
function paintO(x, y) {

    var offsetX = (width / 3) * 0.01;
    var offsetY = (height / 3) * 0.01;
    var beginX = (x * (width / 3) + x) + offsetX;
    var beginY = (y * (height / 3) + y) + offsetY;
    // Create a new image.
    var imgO = new Image();

    // Once it's loaded draw the image on the canvas.
    imgO.addEventListener('load', function () {

        // Now resize the image: x, y, w, h.
        context.drawImage(this, beginX, beginY, 126, 126);

    }, false);

    imgO.src = 'image/play ' + tv.alt + '.png';

}
/*
Function clickHandler()
@param - e : event.
@Description - Click on the box handler .
Returns -void
*/
function clickHandler(e) {

    var y = Math.floor(e.offsetY / (height / 3));
    var x = Math.floor(e.offsetX / (width / 3));

    var bit = (1 << x + (y * 3));
    clickSquare(bit)

}
/*
Function checkNobody()
@param -none
@Description - Checks for winner and displays the appropriate message .
Returns -Boolean
*/
function checkNobody() {
    if ((xBoard | oBoard) == 0x1FF) {
        if (checkWinner(xBoard)) {
            playSound(4);
            $("#selectText p").html("Congrats You win!!!");
            removeHandlers();
            return;
        }
        if (checkWinner(oBoard)) {
            playSound(3);
            $("#selectText p").html("Sorry You Lose!");
            removeHandlers();
            return;
        }
        $("#selectText p").html("Nobody wins!!!");
        playSound(3);
        removeHandlers();
        return true;
    }
    return false;
}
/*

```

```
Function removeHandlers()
@param -none
@Description - Removes all the handlers and resets the display. .
Returns -none
*/
function removeHandlers() {
    $("#selectText").addClass("shake");
    $("#board").off("click");
    $("#restart").css("display", "block");
    backHighlight = 1;
    $("#highlight").off("click");
    $("#restart").off("click");
    $("#highlight").css("display", "none");
    gameStarted = false;
    gameEnded = true;
    $("#btn restart").removeClass("button").addClass("buttonSelected");
    $("#restart").on("click", function (event) {
        restart();
    });
    $("#restart").on("mouseenter mouseleave", function (event) {
        highlightPrevious();
    });
}
/*
Function restart()
@param -none
@Description - Restarts the game and draws the fresh board .
Returns -none
*/
function restart() {
    $("#restart").css("display", "none");
    $("#selectText").removeClass("shake");
    $("#selectText p").html("Play Now");
    context.clearRect(0, 0, width, height);
    xBoard = 0;
    oBoard = 0;
    backHighlight = 3;
    paintBoard();
}
/*
Function isEmpty()
@param -xBoard :user box, oBoard:TV box, bit:current box
@Description - Checks the selected box in the boards is empty or not .
Returns -Boolean
*/
function isEmpty(xBoard, oBoard, bit) {
    return (((xBoard & bit) == 0) && ((oBoard & bit) == 0));
}
/*
Function simulate()
@param -xBoard :user box, oBoard:TV box, bit:current box
@Description - simulate the next move by TV.
Returns -bit : the next location to mark the play.
*/
function simulate(oBoard, xBoard) {

    var ratio = 0;

    var bit = 0;
    for (var i = 0; i < 9; i++) {

        var cBit = 1 << i;

        if (isEmpty(xBoard, oBoard, cBit)) {

            if (checkWinner(oBoard | cBit)) {
                bit = cBit;
                break;
            } else if (checkWinner(xBoard | cBit)) {
                bit = cBit;
            }
        }
    }
}
```

```

    }
}

if (bit == 0) {
    for (var i = 0; i < 9; i++) {
        var cBit = 1 << i;

        if (isEmpty(xBoard, oBoard, cBit)) {
            var result = think(oBoard, xBoard, 'X', 0, 1)
            if (ratio == 0 || ratio < result) {
                ratio = result;
                bit = cBit;
            }
        }
    }
}
return bit;
}
/*
Function think()
@param -xBoard :user box, oBoard:TV box,player:curPlayer, bit:current
box ,ratio
@Description - simulate the next move by TV.
Returns - best : the next best location to mark the play.
*/
function think(oBoard, xBoard, player, bit, ratio) {

    if (player == 'O') {
        oBoard = oBoard | bit;
    } else {
        xBoard = xBoard | bit;
    }

    if (checkWinner(oBoard)) {
        ratio *= 1.1;
        return ratio;

    } else if (checkWinner(xBoard)) {

        ratio *= 0.7;
        return ratio;

    } else {
        var best = 0;
        ratio *= 0.6;

        for (var i = 0; i < 9; i++) {

            if (isEmpty(xBoard, oBoard, 1 << i)) {

                var newRatio = think(oBoard, xBoard, player == 'O' ? 'X' : 'O',
1 << i, ratio);

                if (best == 0 || best < newRatio) {
                    best = newRatio;
                }
            }
        }

        return best;
    }
}
/*
Function markBit()
@param -markBit :the bit to be marked, player:curPlayer,
@Description - mark the actual play.
Returns - void.
*/
function markBit(markBit, player) {

    var bit = 1;
    var posX = 0,

```

```
posY = 0;
while ((markBit & bit) == 0) {
    bit = bit << 1;
    posX++;
    if (posX > 2) {
        posX = 0;
        posY++;
    }
}

if (player == 'O') {
    oBoard = oBoard | bit;
    paintO(posX, posY);
} else {
    xBoard = xBoard | bit;
    paintX(posX, posY);
}
}
/*
Function play()
@param -none
@Description - start play function.
Returns - void.
*/
function play() {
    var bestBit = simulate(oBoard, xBoard);
    markBit(bestBit, 'O');
}
/*
Function squareMove(dir)
@param -dir :direction of the movememnt
@Description - Moves around the selected box highlight when mouse if off.
Returns - void.
*/
function squareMove(dir) {
    if (backHighlight != 2) {

        switch (dir) {
            case "up":
                position_index = position_index - 3;
                if (position_index < 0) {
                    position_index = position_index + 3;
                    highlightBack();
                }
                break;
            case "down":
                position_index = position_index + 3;
                if (position_index > position_arr.length - 1) {
                    position_index = position_index - position_arr.length;
                }
                break;
            case "left":
                if (position_index > 0) {
                    position_index--;
                } else {
                    position_index = position_arr.length - 1;
                }
                break;
            case "right":
                if (position_index < position_arr.length - 1) {
                    position_index++;
                } else {
                    position_index = 0;
                }
                break;
        }
        $("#highlight").css("left", position_arr[position_index].x);
        $("#highlight").css("top", position_arr[position_index].y);
    }
}
}
```

```

/*
Function selectSquare()
@param -none
@Description - select the current square when mouse is off and enter is
pressed.
Returns - void
*/
function selectSquare() {
    var y = bit arr[position index].y;
    var x = bit arr[position index].x;
    var bit = (1 << x + (y * 3));
    playSound(2);
    clickSquare(bit);
}
/*
Function clickSquare()
@param -xBoard : bit:current box;
@Description - The actual selection happens here when mouse is off.
Returns - best : the next best location to mark the play.
*/
function clickSquare(bit) {
    if (isEmpty(xBoard, oBoard, bit)) {

        markBit(bit, 'X')

        if (!checkNobody()) {
            if (checkWinner(xBoard)) {
                playSound(4);
                $("#selectText p").html("Congrats You win!!");
                removeHandlers()

            } else {

                play();
                if (!checkNobody()) {

                    if (checkWinner(oBoard)) {
                        playSound(3);
                        $("#selectText p").html("Sorry You Lose!");
                        removeHandlers()

                    }

                }

            }

        }

    }

}

```

[Style.css]

```
/* Page-level */
ul, ol, li, h1, h2, h3, h4, h5, h6, pre, form, div, body, html, p,
blockquote, fieldset, input {
    margin:0;
    padding:0;
}
body {
    overflow:hidden;
    font-family: "LG Display";
    width:1280px;
    height:720px;
    color:#FFFFFF;
    -webkit-user-select:none;
}
.main {
    background:-webkit-gradient(radial, center center, 30, 420 216, 860,
color-stop(0,#404559), color-stop(0.5,#24272C));
    width:1280px;
    height:720px;
}
.page {
    position:absolute;
    top:134px;
    width:1280px;
    height:539px;
    left:0px;
}
/* Note: Make HTML 5 elements block-level for consistent styling */
header, nav, article, footer, address {
    display:block;
}
/* highlight red box for game key navaigation */
.highlight {
    position:absolute;
}
/* Canvas holder */
.title{
    float:left;
    font-size:40px;
    margin:11px 0 0 8px;
}
.gameText{
    height:40px;
    font-size:36px;
}
.game {
    position:absolute;
    width:1280px;
    text-align:center;
    top:60px;
    display:none;
}
/* Select the player holder */
.player {
    width:inherit;
    text-align:center;
}
/* header holder */
header {
    width:1247px;
    padding:10px 16px 0 17px;
}
/* Footer Holder */
footer {
}
/* Content */
h1 {
    font-size: 40px;
}
```



```

h2 {
    font-size: 36px;
}
.selectText{
    padding-bottom:60px;
}
/* button class for the back button */
.button {
    width:178px; height:47px; background:-webkit-gradient(linear, left top,
left bottom, color-stop(0%,#FDFDFD),color-stop(11%,#FDFDFD),color-
stop(49%,#D0D0D0),color-stop(50%,#BFBFBF),color-stop(91%,#A2A2A2)); box-
shadow:0px 0px 1px 1px rgba(0,0,0,0.05),0px 2px 1px rgba(0,0,0,0.4),inset 0px
0px 0px 2px rgba(255,255,255,0.1); border-radius:4px; color:#343434; line-
height:47px;
    font-size:24px; text-align:center;
}
/* button selected class for the back button */
.buttonSelected {
    width:180px; height:50px; background:-webkit-gradient(linear, left top,
left bottom, color-stop(0%,#DB96AC),color-stop(49%,#AF325A),color-
stop(50%,#A61C47),color-stop(100%,#881D3F)); box-shadow:0px 0px 0px 1px
rgba(0,0,0,0.15),0px 2px 1px rgba(0,0,0,0.55),inset 0px 0px 0px 1px
rgba(255,255,255,0.1),inset 0px -3px 3px rgba(255,255,255,0.2); border-
radius:4px; color:#FFFFFF; line-height:50px;
    font-size:24px; text-align:center;
}

.restartHolder {
    margin-top:70px;
    margin-left:550px;
    display:none;
}

/* The player ul class */
.players {
    list-style:none;
}
/* The player image li class*/
.players li {
    display:inline;
}

/* player image normal state*/
.playerImgNormal {
    width:243px;
    height:284px;
    padding:8px 3px;
    margin:0px 70px;
    background-color:#818895;
    border-radius:4px;
    box-shadow:inset 0 1px 0 rgba(255,255,255,0.55),0 4px 3px 2px
rgba(0,0,0,0.5),0 2px 1px rgba(0,0,0,0.5);
}
/*player image normal class*/
.playerImgHover {
    box-shadow:inset 0 1px 0 rgba(255,255,255,0.55),0 1px 0 rgba(0,0,0,0.4),0
0 0 6px rgba(255,0,138,1);
}
.board{
    position:absolute;
    left:445px;
}
/*left motion animation class for the player*/
@-webkit-keyframes aniLeft {
100% {
left:-227px;
}
}
.animateclassLeft {
    position:relative;
    -webkit-transition-property: transform;
    -webkit-transition-duration: 1s;
}

```

```
-webkit-animation-timing-function: ease-in;
-webkit-animation: aniLeft 1s linear;
-webkit-animation-fill-mode: forwards;
}
/*right motion animation class for the player*/
@-webkit-keyframes aniRight {
100% {
left:227px;
}
}
.animateclassRight {
position:relative;
-webkit-transition-property: transform;
-webkit-transition-duration: 1s;
-webkit-animation-timing-function: ease-in;
-webkit-animation: aniRight 1s linear;
-webkit-animation-fill-mode: forwards;
}

/*Teh shaking animation for the result message*/
.shake {
position:relative;
-webkit-animation-name: shake;
-webkit-animation-duration: 0.8s;
-webkit-transform-origin:50% 50%;
-webkit-animation-iteration-count: infinite;
-webkit-animation-timing-function: linear;
}
@-webkit-keyframes shake {
0% {
-webkit-transform: translate(2px, 1px) rotate(0deg);
}
10% {
-webkit-transform: translate(-1px, -2px) rotate(-1deg);
}
20% {
-webkit-transform: translate(-3px, 0px) rotate(1deg);
}
30% {
-webkit-transform: translate(0px, 2px) rotate(0deg);
}
40% {
-webkit-transform: translate(1px, -1px) rotate(1deg);
}
50% {
-webkit-transform: translate(-1px, 2px) rotate(-1deg);
}
60% {
-webkit-transform: translate(-3px, 1px) rotate(0deg);
}
70% {
-webkit-transform: translate(2px, 1px) rotate(-1deg);
}
80% {
-webkit-transform: translate(-1px, -1px) rotate(1deg);
}
90% {
-webkit-transform: translate(2px, 2px) rotate(0deg);
}
100% {
-webkit-transform: translate(1px, -2px) rotate(-1deg);
}
}
/* Screen game*/
ol,ul,li{list-style:none;}
.screen game .tictactoe screen{overflow:hidden; float:left; width:390px;
height:390px; margin:0 445px;}
.screen game .tictactoe screen li div{width:126px; height:126px; margin:0 3px
3px 0; padding:1px; border-radius:5px; box-shadow:inset 0 2px 4px
rgba(0,0,0,0.75);}
.screen game .tictactoe screen li .cell right{margin:0;}
```

```

.screen_game .tictactoe_screen li .cell_focus{display:block;
position:relative;}
.screen_game .tictactoe_screen li .cell_focus::before{display:block;
content:''; position:absolute; width:119px; height:119px; margin:4px; box-
shadow:0 0 4px #FF008A; border-radius:3px;}
.screen_game .tictactoe_screen li .cell_focus::after{display:block;
content:''; position:absolute; width:119px; height:119px; margin:4px;
background:-webkit-gradient(linear,left top, left bottom, color-
stop(0%,#22252E), color-stop(100%,#191B21));}

.screen_game .tictactoe_screen li .cell_focus_img{display:block;}

.screen_game .tictactoe_screen li,.screen_game .tictactoe_screen li
div{float:left;}

.screen_game .tictactoe_screen .cube_first .cell_default{background:-webkit-
gradient(linear,left top, left bottom, color-stop(0%,#292C37), color-
stop(100%,#22252E));}
.screen_game .tictactoe_screen .cube_second .cell_default{background:-webkit-
gradient(linear,left top, left bottom, color-stop(0%,#22252E), color-
stop(100%,#191B21));}
.screen_game .tictactoe_screen .cube_last .cell_default{background:-webkit-
gradient(linear,left top, left bottom, color-stop(0%,#191B21), color-
stop(100%,#121418));}
.player_name{font-size:44px; color:#818895; text-align:right;}

/* MainButton */
.main_btn{position:relative; overflow:hidden; float:right; height:72px; z-
index:140;}
.main_btn li{float:left; width:51px; height:72px; margin-left:3px;}
.main_btn .btn_name{width:51px; height:22px; font:18px normal "LG Display";
text-align:center; color:#FFFFFF;}
.main_btn .btn_default .btn_icon{width:45px; height:45px; margin:2px 3px 0
3px; background:-webkit-gradient(linear, left top, left bottom, color-
stop(0%,rgba(211,214,215,0.35)), color-stop(49%,rgba(115,118,122,0.35)),
color-stop(50%,rgba(86,91,96,0.3)), color-stop(100%,rgba(53,56,60,0.3)));
border-radius:3px; box-shadow:inset 0 2px 2px rgba(255,255,255,0.25),inset
2px 0 2px rgba(255,255,255,0.15),inset 0 -3px 2px
rgba(255,255,255,0.05),inset -3px 0 2px rgba(255,255,255,0.05),0px 2px 5px
rgba(0,0,0,0.45); text-align:center;}
.main_btn .btn_default .btn_icon_img{margin-top:7px;}
.main_btn .btn_focus .btn_icon{width:51px; height:51px; background:-webkit-
gradient(linear, left top, left bottom, color-stop(0%,#D898AC), color-
stop(49%,#A83257), color-stop(50%,#A0234A), color-stop(100%,#A01842));
border-radius:3px; box-shadow:inset 0 2px 1px rgba(255,255,255,0.25),inset
2px 0 2px rgba(255,255,255,0.15),inset 0 -4px 2px
rgba(255,255,255,0.05),inset -4px 0 2px rgba(255,255,255,0.05),0px 2px 3px
rgba(0,0,0,0.2), 0 4px 7px rgba(0,0,0,0.3); text-align:center;}
.main_btn .btn_focus .btn_icon_img{margin-top:10px;}
.main_btn .btn_focus .img_default,.main_btn .btn_default .btn_name,.main_btn
.btn_default .img_focus{display:none;}

```

3.2 Descriptions on Source Codes

This section describes the core part of the sample source codes. Following sources are related to index.html, which is for Main Screen.

Linking JavaScript and CSS (index.html)

Specifies the path of JavaScript and stylesheet source file that will be used in an application.

```
<script src="js/jquery-1.8.2.min.js"></script>
<script src="js/tic.js"></script>
<script src="js/keycode.js"></script>
<script src="js/keynavigation.js"></script>
```

Designing displaying structure (index.html)

Designs the structure to be displayed on an application screen using the <div> tag.

```
<div class="main">
  Main Display Area
</div>

<div id="page" class="page">
  Contents
</div>

<div id="player" class="player">
  Player Selection Area
</div>

<div id="game" class="game">
  Game Display Area
</div>
```

Displaying header structure (index.html)

Designs the structure for header using the <header> tag.

```
<header>
  <p class="title">LG Smart TV SDK | Tic Tac Toe Sample</p>
  <!-- main BTN -->
  <ul class="main_btn">
    <li class="btn_default">
      <p class="btn icon img default">
        
      </p>
      <p class="btn icon img focus">
        
      </p>
      <p class="btn name">Back</p>
    </li>
  </ul>
</header>
```

Footer (index.html)

Contains the <div> tag which holds Object for Media and the Sound plugin. Media starts automatically background music and plays in loop.

```

<div>
  <object type="application/x-netcast-av" data="sound/bk.mp3" width="0"
  height="0" playcount="0" id="media" autoStart="true"></object>
  <object type="audio/x-wav" id="sound1" data="sound/button.wav"
  width="0" height="0" autostart="false"></object>
  <object type="audio/x-wav" id="sound2" data="sound/click.wav"
  width="0" height="0" autostart="false"></object>
  <object type="audio/x-wav" id="sound3" data="sound/boo.wav" width="0"
  height="0" autostart="false"></object>
  <object type="audio/x-wav" id="sound4" data="sound/applause.wav"
  width="0" height="0" autostart="false"></object>
</div>

```

Displaying the 'Select Player' text. Also used for displaying the game results (index.html)

Contains the <p> tag where the text is included.

```

<div id="selectText" class="selectText">
  <p class="gameText">Select the player</p>
</div>

```

Creating the Player Selection Icons

```

<ul id="players" class="players">
  <li>
    
  </li>
  <li>
    
  </li>
</ul>

```

Creating Restart Button Holder (index.html)

Contains the <div> tag which holds the Restart Button.

```

<div class="restartHolder" id="restart">
  <div id='btn_restart' class='button'>Restart</div>
</div>

```

Creating the game screen (index.html)

Contains the tag which holds all the cells.

```

<div class="screen_game">
  <!-- Game -->
  <ul class="tictactoe screen" style="color:#FFFFFF;">
    <!-- FirstLine -->
    <li class="cube first">
      <div class="cell default"></div>
      <div class="cell default"></div>
      <div class="cell right cell default"></div>
    </li>
    <!-- //FirstLine -->
    <!-- SecondLine -->
    <li class="cube second">
      <div class="cell default"></div>
      <div class="cell_default"></div>
    </li>
  </ul>

```

3 Implementing the Code

```
        <div class="cell_right cell_default"></div>
    </li>
    <!-- //SecondLine -->
    <!-- LastLine -->
    <li class="cube last">
        <div class="cell default"></div>
        <div class="cell default"></div>
        <div class="cell_right cell_default "></div>
    </li>
    <!-- //LastLine -->
</ul>
<!-- //Game -->
</div>
```

Canvas Element for drawing the cell selection (index.html)

```
<canvas width="390" height="390" id="board" class="board"></canvas>
```

Creating the Highlight element for cells of the game. (index.html)

```

```

Implementing the Player Selection (keynavigation.js)

Code for selecting the player. When one player is selected, other player becomes TV. The player icons move apart. Tic Tac Toe Board is drawn.
playSound(1) will make the first Sound Effect to be played.

```
function playerClicked(obj) {
    backHighlight = 3;
    var dir = "Left"
    var selItem = obj;
    selplayer = selItem;
    playSound(1);
    $("#players li img").each(function (i) {
        var curItem;
        $(this).addClass("animateclass" + dir);
        $(this).off("mouseenter mouseleave click");
        $("#selectText h2").html("Play Now");
        if (this.alt != selItem.alt) {
            curItem = this;
            tv = curItem;
        }
        dir = "Right";
        this.addEventListener('webkitAnimationEnd',

        function (event) {
            $("#game").css("display", "block");

            if (curItem) {
                $(".player_name").css("display", "block");
                $(".player_name").css("position", "absolute");
                $(".player_name").css("width", $(curItem).width());
                $(".player_name").css("left", $(curItem).offset().left - 10);
                $(".player_name").css("top", $(curItem).offset().top + 170);

                paintBoard();
            }
        }, false);
    });
    allMenuObject = [];
}
```

Implementing the sound effect play functionality.

Code for playing Sound effects. Passed number will specify which effect to play.

```
function playSound(num) {
    var sound = document.getElementById("sound" + num);
    sound.play();
}
```

Processing key event (keynavigation.js)

Defines action according to the remote control key input.

```
$(document).keydown(function (event) {

    var key = event.keyCode || event.which;
    switch (key) {
    case VK BACK:
        backButtonSelect();
        break;
    case VK RIGHT:
        if (gameStarted) {
            squareMove("right");
        } else if (gameEnded) {
            if (backHighlight == 2) {
                highlightPrevious();
            } else if (backHighlight == 1) {
                highlightBack();
            }
        } else {
            if (allMenuObject[cntIndex + 1] != undefined && backHighlight != 2)
            {
                cntIndex++;
                resetHighlight();
                $(allMenuObject[cntIndex]).toggleClass("playerImgHover");
            }
        }
        break;
    case VK LEFT:
        if (gameStarted) {
            squareMove("left");
        } else if (gameEnded) {
            if (backHighlight == 2) {
                highlightPrevious();
            } else if (backHighlight == 1) {
                highlightBack();
            }
        } else {
            if (allMenuObject[cntIndex - 1] != undefined && backHighlight != 2)
            {
                cntIndex--;
                resetHighlight();
                $(allMenuObject[cntIndex]).toggleClass("playerImgHover");
            }
        }
        break;
    case VK UP:

        if (backHighlight == 1) {
            highlightBack();
        } else {
            squareMove("up");
        }
        break;
    case VK DOWN:
        if (backHighlight == 2) {
```

3 Implementing the Code

```
        highlightPrevious();
    } else {
        squareMove("down");
    }

    break;
case VK ENTER:
    if (backHighlight == 2) {
        backButtonSelect();
    } else if (gameStarted) {
        selectSquare();
    } else if (gameEnded) {
        restart();
    } else {
        playerClicked(allMenuObject[cntIndex]);
    }
    break;
}
});
```

Click action on the back button (keynavigation.js)

When the back button is clicked on a player selection screen, application closes. When the back button is clicked after game starts, screen goes back to the player selection screen.

```
function backButtonSelect() {
    if (gameStarted || gameEnded) {
        history.go(0);
    } else {
        if (window.NetCastBack) {
            window.NetCastBack();
        }
    }
}
```

Click action on the canvas (tic.js)

When the canvas is clicked, a cell is selected based on the position.

```
function clickHandler(e) {

    var y = Math.floor(e.offsetY / (height / 3));
    var x = Math.floor(e.offsetX / (width / 3));

    var bit = (1 << x + (y * 3));
    clickSquare(bit)
}
```

Drawing the image on the canvas based on the player (tic.js)

TV selection

```
function paintO(x, y) {

    var offsetX = (width / 3) * 0.01;
    var offsetY = (height / 3) * 0.01;
    var beginX = (x * (width / 3) + x) + offsetX;
    var beginY = (y * (height / 3) + y) + offsetY;
    // Create a new image.
    var imgO = new Image();

    // Once it's loaded draw the image on the canvas.
    imgO.addEventListener('load', function () {
```



```

        // Now resize the image: x, y, w, h.
        context.drawImage(this, beginX, beginY, 126, 126);

    }, false);

    imgO.src = 'image/play ' + tv.alt + '.png';

}

```

Player selection-

```

function paintX(x, y) {
    var offsetX = (width / 3) * 0.01;
    var offsetY = (height / 3) * 0.01;
    var beginX = (x * (width / 3) + x) + offsetX;
    var beginY = (y * (height / 3) + y) + offsetY;
    // Create a new image.
    var imgX = new Image();

    // Once it's loaded draw the image on the canvas.
    imgX.addEventListener('load', function () {

        // Now resize the image: x, y, w, h.
        context.drawImage(this, beginX, beginY, 126, 126);

    }, false);

    imgX.src = 'image/play ' + selplayer.alt + '.png';
}

```

Displaying the result of the game (tic.js)

After the completion of game, appropriate result message is displayed. Also Sound effect is played based on the result.

```

function checkNobody() {
    if ((xBoard | oBoard) == 0x1FF) {
        if (checkWinner(xBoard)) {
            playSound(4);
            $("#selectText p").html("Congrats You win!!!");
            removeHandlers();
            return;
        }
        if (checkWinner(oBoard)) {
            playSound(3);
            $("#selectText p").html("Sorry You Lose!");
            removeHandlers();
            return;
        }
        $("#selectText p").html("Nobody wins!!!");
        playSound(3);
        removeHandlers();
        return true;
    }
    return false;
}

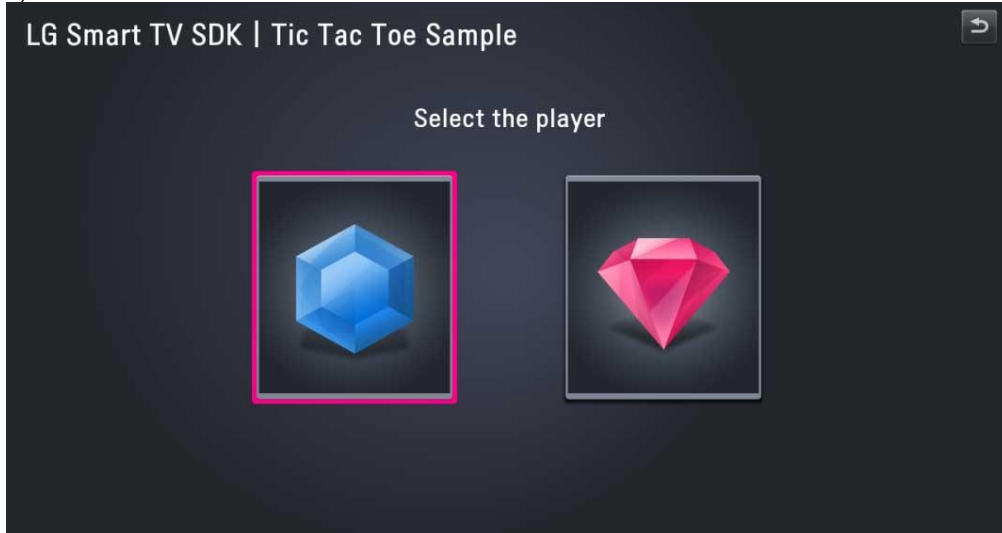
```

3.3 Expected Result

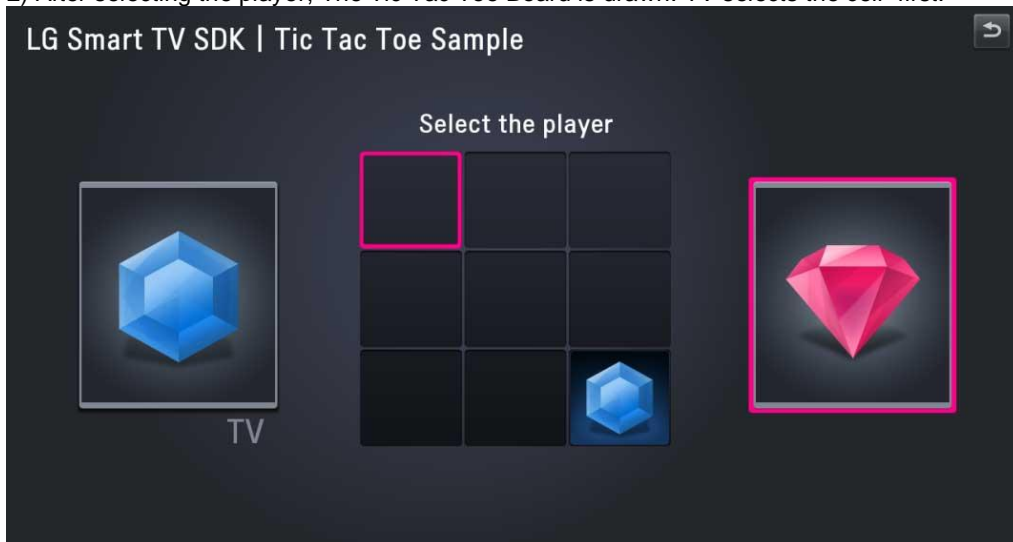
The following screens will be displayed when the sample application starts on the LG Smart TV Emulator 2012. When the application starts, the main screen is displayed. When user clicks a player, the selection is made and the other player is assigned as TV.

The emulator version is based on SDK V2.1.0 release.

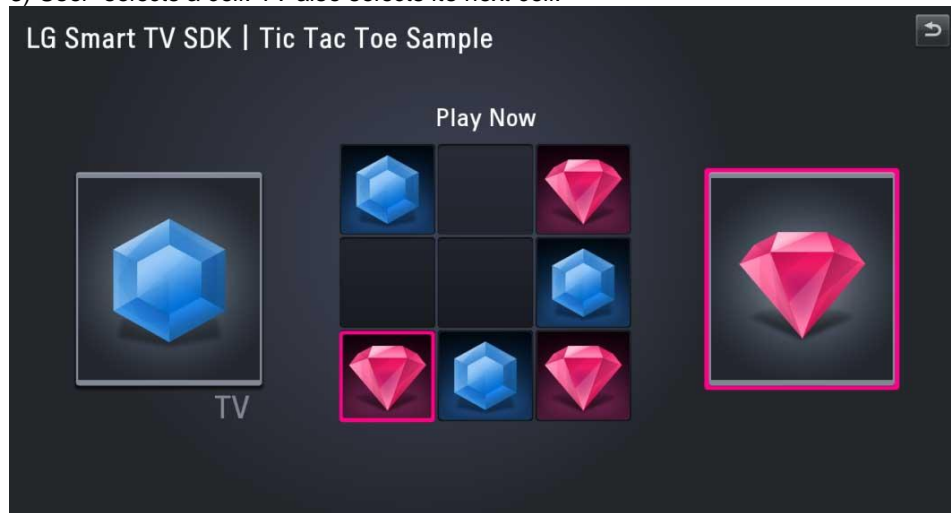
1) Main Screen



2) After selecting the player, The Tic Tac Toe Board is drawn. TV selects the cell first.

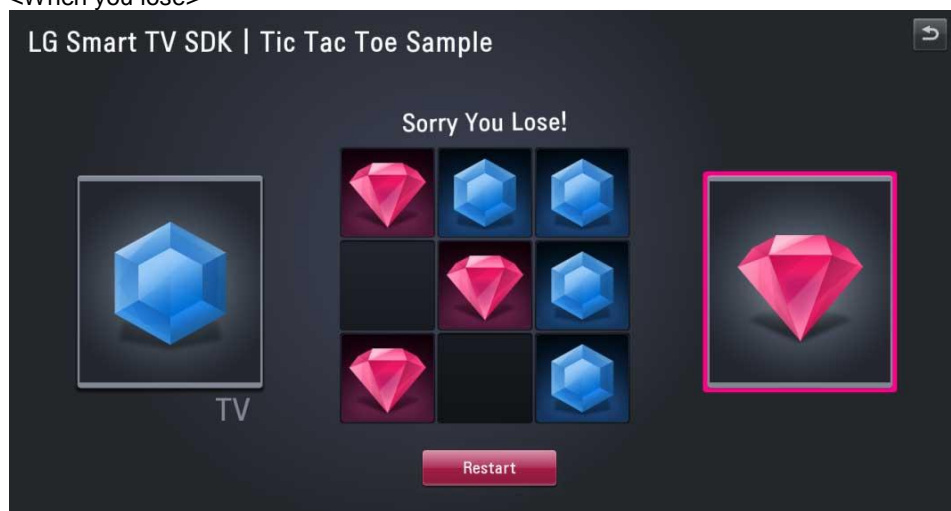


3) User selects a cell. TV also selects its next cell.

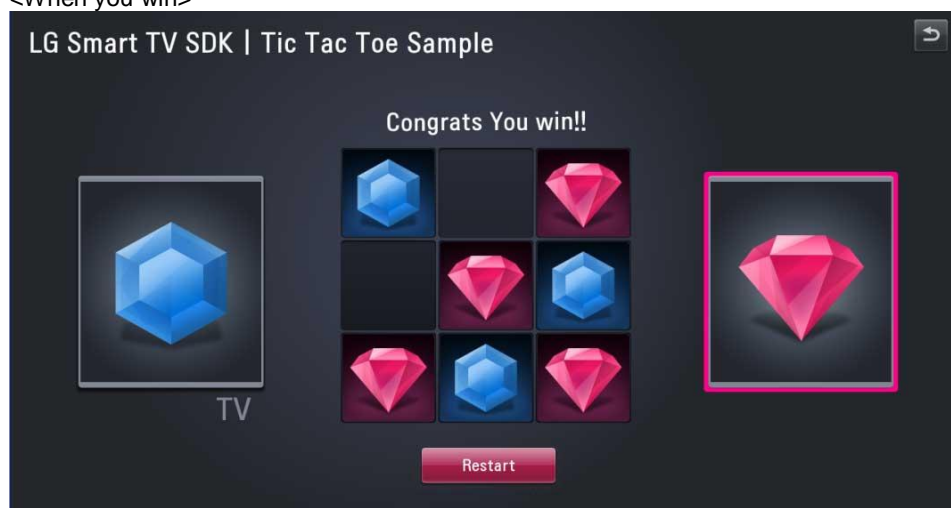


4) Game Over Screen with the result messages

<When you lose>



<When you win>



3 Implementing the Code

<When nobody wins>





4 Testing the Application

This chapter briefly describes how to test a developed application.

To debug the application on emulator, refer to **Testing > Testing App on Emulator** section in Developer Library in LG Developer (<http://developer.lge.com>) website

To run and debug the application on real TV, refer to **Testing > Deploying and Testing App on Real TV** section in Developer Library in LG Developer website.