<!DOCTYPE html>

<!--

    NOTES:

    1. All tokens are represented by '$' sign in the template.

    2. You can write your code only wherever mentioned.

    3. All occurrences of existing tokens will be replaced by their appropriate values.

    4. Blank lines will be removed automatically.

    5. Remove unnecessary comments before creating your template.

-->

<html>

<head>

    <meta charset="UTF-8">

    <meta name="authoring-tool" content="Adobe\_Animate\_CC">

    <title>SPACE\_WAR</title>

    <!-- write your code here -->

    <script src="https://code.createjs.com/1.0.0/createjs.min.js"></script>

    <script src="SPACE\_WAR.js?1592626883617"></script>

    <link rel="stylesheet" href="./SPACE\_WAR.css">

    <script src="./ndgmr.Collision.js"></script>

    <script>

        var canvas, stage, exportRoot, anim\_container, dom\_overlay\_container, fnStartAnimation;

        function init() {

            canvas = document.getElementById("canvas");

            anim\_container = document.getElementById("animation\_container");

            dom\_overlay\_container = document.getElementById("dom\_overlay\_container");

            var comp = AdobeAn.getComposition("0356A0DF8E8F4448B3C6AC8B88E2FDE9");

            var lib = comp.getLibrary();

            var loader = new createjs.LoadQueue(false);

            loader.addEventListener("fileload", function (evt) {

                handleFileLoad(evt, comp)

            });

            loader.addEventListener("complete", function (evt) {

                handleComplete(evt, comp)

            });

            var lib = comp.getLibrary();

            loader.loadManifest(lib.properties.manifest);

        }

        function handleFileLoad(evt, comp) {

            var images = comp.getImages();

            if (evt && (evt.item.type == "image")) {

                images[evt.item.id] = evt.result;

            }

        }

        function handleComplete(evt, comp) {

            //This function is always called, irrespective of the content. You can use the variable "stage" after it is created in token create\_stage.

            var lib = comp.getLibrary();

            var ss = comp.getSpriteSheet();

            var queue = evt.target;

            var ssMetadata = lib.ssMetadata;

            for (i = 0; i < ssMetadata.length; i++) {

                ss[ssMetadata[i].name] = new createjs.SpriteSheet({

                    "images": [queue.getResult(ssMetadata[i].name)],

                    "frames": ssMetadata[i].frames

                })

            }

            exportRoot = new lib.SPACE\_WAR();

            stage = new lib.Stage(canvas);

            stage.addChild(exportRoot);

            //建立飛船物件

            var ship = new lib.sh();

            //設定飛船初始位置

            ship.x = 512;

            ship.y = 660;

            //建立飛船影像

            exportRoot.addChild(ship);

            //預設水平與垂直移動變數為0

            var H = 0;

            var V = 0;

            //設定每次移動的距離

            var D = 10;

            //預設鍵盤按下布林值為否

            var is\_KD = false;

            //設定鍵盤監聽事件

            window.addEventListener("keydown", KDF)

            window.addEventListener("keyup", KUF)

            //鍵盤按下

            function KDF(e) {

                //回傳鍵盤按鍵的keyCode

                console.log(e.keyCode);

                //鍵盤keyCode紀錄 37左 38上 39右 40下

                //左右方向控制

                if (e.keyCode === 37 || e.keyCode === 39) {

                    //執行一次後直接擋掉避免重複執行，影像一直重新loading

                    if (is\_KD) return;

                    //控制is\_KD

                    is\_KD = true;

                    //移動方向控制

                    H = e.keyCode === 39 ? 1 : -1;

                    //移動時播放run狀態

                    ship.gotoAndPlay("run");

                }

                //上下方向控制

                if (e.keyCode === 38 || e.keyCode === 40) {

                    //執行一次後直接擋掉避免重複執行，影像一直重新loading

                    if (is\_KD) return;

                    //控制is\_KD

                    is\_KD = true;

                    //移動方向控制

                    V = e.keyCode === 38 ? -1 : 1;

                    //移動時播放run狀態

                    ship.gotoAndPlay("run");

                }

            }

            //鍵盤放開

            function KUF(e) {

                is\_KD = false;

                //停止移動時撥放stop狀態

                ship.gotoAndPlay("stop");

                //將控制方向的值歸0

                H = 0;

                V = 0;

            }

            //設定監聽事件並執行Fn MOVE來達成ship移動

            createjs.Ticker.addEventListener("tick", MOVE)

            function MOVE() {

                //左右移動

                ship.x += D \* H;

                //上下移動

                ship.y += D \* V;

            }

            //Registers the "tick" event listener.

            fnStartAnimation = function () {

                createjs.Ticker.framerate = lib.properties.fps;

                createjs.Ticker.addEventListener("tick", stage);

            }

            //Code to support hidpi screens and responsive scaling.

            AdobeAn.makeResponsive(false, 'both', false, 1, [canvas, anim\_container, dom\_overlay\_container]);

            AdobeAn.compositionLoaded(lib.properties.id);

            fnStartAnimation();

        }

    </script>

    <!-- write your code here -->

</head>

<body onload="init();" style="margin:0px;">

    <div class="container">

        <h1 class="score">0</h1>

        <div id="animation\_container" style="background-color:rgba(255, 255, 255, 1.00); width:1024px; height:768px">

            <canvas id="canvas" width="1024" height="768"

                style="position: absolute; display: block; background-color:rgba(255, 255, 255, 1.00);"></canvas>

            <div id="dom\_overlay\_container"

                style="pointer-events:none; overflow:hidden; width:1024px; height:768px; position: absolute; left: 0px; top: 0px; display: block;">

            </div>

        </div>

    </div>

</body>

</html>