<!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?1592636070974"></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

                })

            }

            //建立背景物件

            exportRoot = new lib.SPACE\_WAR();

            stage = new lib.Stage(canvas);

            //顯示背景

            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;

            //螢幕上顯示飛船

            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 = 20;

            //預設鍵盤按下布林值為否

            var is\_KD = false;

            //預設分數變數為0分

            var score = 0;

            //for a1

            //預設a1子彈數為0

            var a1\_num = 0;

            //預設裝a1子彈的陣列

            var a1\_array = [0];

            //產生a1子彈的Fn

            var a1\_generate = function () {

                let a1 = new lib.a1();

                return a1;

            }

            //for a2

            //預設a2雷球數為0

            var a2\_num = 0;

            //預設裝a2雷球的陣列

            var a2\_array = [0];

            //產生a2雷球的Fn

            var a2\_generate = function () {

                let a2 = new lib.a2();

                return a2;

            }

            //for a4

            //預設a4脈衝波數為0

            var a4\_num = 0;

            //預設裝a4脈衝波的陣列

            var a4\_array = [0];

            //產生a4脈衝波的Fn

            var a4\_generate = function () {

                let a4 = new lib.a4();

                return a4;

            }

            //for a7

            //預設a7月牙天衝數為0

            var a7\_num = 0;

            //預設裝a7月牙天衝的陣列

            var a7\_array = [0];

            //產生a7月牙天衝的Fn

            var a7\_generate = function () {

                let a7 = new lib.a7();

                return a7;

            }

            //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");

                }

                //按空白鍵發射子彈

                if (e.keyCode === 32) {

                    //子彈個數加1

                    a1\_num += 1;

                    //存一個區域變數值等於當時子彈個數

                    let tmp = a1\_num;

                    //呼叫Fn a1\_generate()產生子彈後存入a1\_array內之當下a1\_num值的位置

                    a1\_array[tmp] = a1\_generate();

                    //抓取當前飛船的相對位置

                    let ship\_x = (ship.x + 12);

                    let ship\_y = (ship.y - 23);

                    //設定子彈產生的位置

                    a1\_array[tmp].x = ship\_x;

                    a1\_array[tmp].y = ship\_y;

                    //螢幕上顯示子彈

                    exportRoot.addChild(a1\_array[tmp]);

                    //利用CreateJS的TweenJS實現移動與動畫效果

                    createjs.Tween.get(a1\_array[tmp])

                        //設定子彈物件產生後自動移動到y=-500位置(2.5秒內完成)，

                        .to({

                            y: -500

                        }, 2500)

                        //當達成to指令後消除物件與動畫

                        .call(() => {

                            createjs.Tween.removeTweens(a1\_array[tmp]);

                            exportRoot.removeChild(a1\_array[tmp]);

                        })

                        //增加子彈監聽事件監聽觸發事件

                        .addEventListener("change", () => {

                            if (a1\_array[tmp].y <= -250) {

                                createjs.Tween.removeTweens(a1\_array[tmp]);

                                exportRoot.removeChild(a1\_array[tmp]);

                                a1\_array[tmp] = 0;

                            }

                        })

                }

                //按Q發射雷球

                if (e.keyCode === 81) {

                    a2\_num += 1;

                    let tmp = a2\_num;

                    a2\_array[tmp] = a2\_generate();

                    let ship\_x = (ship.x);

                    let ship\_y = (ship.y - 40);

                    a2\_array[tmp].x = ship\_x;

                    a2\_array[tmp].y = ship\_y;

                    exportRoot.addChild(a2\_array[tmp]);

                    createjs.Tween.get(a2\_array[tmp])

                        .to({

                            y: -500

                        }, 2500)

                        .call(() => {

                            createjs.Tween.removeTweens(a2\_array[tmp]);

                            exportRoot.removeChild(a2\_array[tmp]);

                        })

                        .addEventListener("change", () => {

                            if (a2\_array[tmp].y <= -250) {

                                createjs.Tween.removeTweens(a2\_array[tmp]);

                                exportRoot.removeChild(a2\_array[tmp]);

                                a2\_array[tmp] = 0;

                            }

                        })

                    createjs.Tween.get(a2\_array[tmp])

                        //讓雷球轉360度(15秒內完成)

                        .to({

                            rotation: 360,

                        }, 15000)

                }

                //按W發射月牙天衝

                if (e.keyCode === 87) {

                    a7\_num += 1;

                    let tmp = a7\_num;

                    a7\_array[tmp] = a7\_generate();

                    let ship\_x = (ship.x + 50);

                    let ship\_y = (ship.y + 10);

                    a7\_array[tmp].x = ship\_x;

                    a7\_array[tmp].y = ship\_y;

                    exportRoot.addChild(a7\_array[tmp]);

                    createjs.Tween.get(a7\_array[tmp])

                        .to({

                            y: -500

                        }, 2500)

                        .call(() => {

                            createjs.Tween.removeTweens(a7\_array[tmp]);

                            exportRoot.removeChild(a7\_array[tmp]);

                        })

                        .addEventListener("change", () => {

                            if (a7\_array[tmp].y <= -250) {

                                createjs.Tween.removeTweens(a7\_array[tmp]);

                                exportRoot.removeChild(a7\_array[tmp]);

                                a7\_array[tmp] = 0;

                            }

                        })

                }

                //按R發射脈衝波

                if (e.keyCode === 82) {

                    a4\_num += 1;

                    let tmp = a4\_num;

                    a4\_array[tmp] = a4\_generate();

                    let ship\_y = (ship.y - 20);

                    a4\_array[tmp].x = 0;

                    a4\_array[tmp].y = ship\_y;

                    exportRoot.addChild(a4\_array[tmp]);

                    createjs.Tween.get(a4\_array[tmp])

                        .to({

                            y: -500

                        }, 2500)

                        .call(() => {

                            createjs.Tween.removeTweens(a4\_array[tmp]);

                            exportRoot.removeChild(a4\_array[tmp]);

                        })

                        .addEventListener("change", () => {

                            if (a4\_array[tmp].y <= -250) {

                                createjs.Tween.removeTweens(a4\_array[tmp]);

                                exportRoot.removeChild(a4\_array[tmp]);

                                a4\_array[tmp] = 0;

                            }

                        })

                }

            }

            //鍵盤放開

            function KUF(e) {

                is\_KD = false;

                //停止移動時撥放stop狀態(飛船漂浮效果)

                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;

            }

            //建立整個遊戲環境的監聽器

            fnStartAnimation = function () {

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

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

            }

            //支援螢幕縮放

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

            AdobeAn.compositionLoaded(lib.properties.id);

            fnStartAnimation();

        }

    </script>

</head>

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

    <!-- 將遊戲畫面包在div.container內，預設置中 -->

    <div class="container">

        <!-- 利用h1.score顯示目前得分 -->

        <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>