```
<html>
    <head>
        <title>Sub Program</title>

        <script language="javascript">

            var colorOne = "#0000FF";

            //defines the boundaries of the canvas as varaibles to make it easier to hot
            swap
            var canvasBoundX = 700;
            var canvasBoundY = 500;

            var speed = 6;

            //defines initial position of the sub
            var xSub = 350;
            var ySub = 200;

            var currentMovement = "none";
            var previousMovement = "none";

            var isOn = false;
            var time = 0;

            var health = 2000;
            //used for timer to determine if it is on or not

            //an array to hold all the fishes
            var fishes = new Array();

            class Fish{

                constructor(x, y, image, direction, speedX, speedY){
                    this.x = x;
                    this.y = y;
                    this.image = image;
                    this.direction = direction;
                    this.speedX = speedX;
                    this.speedY = speedY;
                }

                setX(newX){
                    this.x = newX - 50;
                }

                setY(newY){
                    this.y = newY - 50;
                }

                getX(){
                    return this.x + 50;
                }

                getY(){
                    return this.y + 50;
                }

                setSpeedX(speedX){
                    this.speedX = speedX;

                    if(speedX > 0){
                        this.setDirection("Right");
                    }
                    else if(speedX < 0){
                        this.setDirection("Left");
                    }
                }

                getSpeedX(){
                    return this.speedX;
                }

                setSpeedY(speedY){
                    this.speedY = speedY;
                }

                getSpeedY(){
                    return this.speedY;
                }

                setDirection(direction){
                    this.direction = direction;
                }

                getImage(){
                    return this.image + this.direction;
                }
            }

            //for loop to make a bunch of fishes
            for(var i = 0; i < 6; i++){

                var xCoord = Math.floor(Math.random() * (canvasBoundX - 200)) + 100;
                var yCoord = Math.floor(Math.random() * (canvasBoundY - 200)) + 100;

                var colorIndex = Math.floor(Math.random() * 9) + 1;
                var image = "fish" + colorIndex;

                //var direction = "Right";
                var fishSpeedX = Math.floor(Math.random() * 4) - 2;
                var fishSpeedY = Math.floor(Math.random() * 4) - 2;

                if(fishSpeedX > 0)
                    direction = "Right";
                else if(fishSpeedX < 0)
                    direction = "Left";
                else{
                    fishSpeedX = 2;
                    direction = "Right";
                }


                fishes[i] = new Fish(xCoord, yCoord, image, direction, fishSpeedX, fishSpeedY);
            }

            var smallFish = new Fish(300, 300, "littleFish", "Left", 3, 3)


            //adds keyboard listener
            window.addEventListener("keydown", function(event){
                //changes keyboard input into diretions
                switch(event.key){
                    case "a":
                        currentMovement = "left";
                        break;
                    case "d":
                        currentMovement = "right";
                        break;
                    case "s":
                        currentMovement = "down";
                        break;
                    default:
                        currentMovement = "none";
                        break;
                }

                if(currentMovement != "none" && currentMovement != "down")
                    previousMovement = currentMovement;

            }, true);

            window.addEventListener("keyup", function(event){
                currentMovement = "none";

            }, true);


            //fill background and turn  on timer when the body gets initialized
            function initialize(){
                var canvas = document.getElementById("myCanvas");
                var context = canvas.getContext("2d");

                context.fillStyle="#ADD8E6";
                context.fillRect(0, 0, canvasBoundX, canvasBoundY);

                turnOn();
            }

            //periodically called ever 20ms to update the game
            function update(){
                time += .02;
                resetBackground();
                controlSub();
                moveFish();
                decayHealth();
                writeText(Math.floor(time * 100) / 100, 20, 40);
                writeText(health, 20, 80);

                if(health < 0){
                    writeText("GAME OVER", 300, 300);
                    writeText("You lasted: " + (Math.floor(time * 100) / 100) + "
                    seconds!", 300, 400);
                    turnOff();
                }
            }

            function drawSub(x, y, direction){
                var canvas = document.getElementById("myCanvas");
                var context = canvas.getContext("2d");

                var image;

                if(direction == "right")
                    image = document.getElementById("sub right");
                else
                    image = document.getElementById("sub left");

                context.drawImage(image, x -50, y - 50, 120, 75);
            }

            function drawImage(id, x, y, width, height){
                var canvas = document.getElementById("myCanvas");
                var context = canvas.getContext("2d");

                var image = document.getElementById(id);
                context.drawImage(image, x, y, width, height);
            }

            //writes text on the screen given what to write and the location
```

```javascript
                function writeText(text, x, y){
                    var canvas = document.getElementById("myCanvas");
                    var context = canvas.getContext("2d");

                    context.font = "30px Arial";
                    context.fillStyle = "#FF0000";
                    context.fillText(text, x, y);
                }

                //moves the sub based on the given direction
                function controlSub(){

                    switch(currentMovement){
                        case "left":
                            if(xSub > 40)
                                xSub -= speed;
                            drawSub(xSub, ySub, previousMovement);
                            break;
                        case "right":
                            if(xSub < canvasBoundX - 60)
                                xSub += speed;
                            drawSub(xSub, ySub, previousMovement);
                            break;
                        case "down":
                            if(ySub <= canvasBoundY - 15)
                                ySub += speed;
                            drawSub(xSub, ySub, previousMovement);
                            break;
                        default:

                            drawSub(xSub, ySub, previousMovement);
                            break;
                    }

                    if(currentMovement != "down" && ySub > 40)
                        ySub -= speed / 5;
                }

                //used to reset the background to avoid smearing
                function resetBackground(){
                    var canvas = document.getElementById("myCanvas");
                    var context = canvas.getContext("2d");
                    //paint the background of the canvas
                    context.fillStyle="#ADD8E6";
                    context.fillRect(0, 0, canvasBoundX, canvasBoundY);
                }

                //places the defender to a given location and moves the turret based on the direction
                function moveSub(x, y, direction){
                    var canvas = document.getElementById("myCanvas");
                    var context = canvas.getContext("2d");

                    var directX = 0;
                    var directY = 0;

                    context.fillStyle = colorOne;
                    context.beginPath();
                    context.arc(x, y, 15, 0, 2 * Math.PI, true);
                    context.closePath();
                    context.fill();

                    if(direction == "left")
                        directX = -1;
                    else if(direction == "right")
                        directX = 1;

```

```
    context.fillStyle = colorOne;
    context.beginPath();
    context.arc(x + (15 * directX), y, 7.5, 0, 2 * Math.PI, true);
    context.closePath();
    context.fill();
}

function moveFish(){

    for(var i = 0; i < fishes.length; i++){
        var currentFish = fishes[i];

        if(time % 2 < 0.1){
            currentFish.setSpeedX((Math.random() - 0.5) * 4);
            currentFish.setSpeedY((Math.random() - 0.5) * 4);
        }

        if((currentFish.getX() < 25) || (currentFish.getX() > canvasBoundX - 25))
            currentFish.setSpeedX(currentFish.getSpeedX() * -1);

        if((currentFish.getY() < 10 && currentFish.getSpeedY() < 0) || (currentFish.getY() > canvasBoundY - 10 && currentFish.getSpeedY() > 0))
            currentFish.setSpeedY(currentFish.getSpeedY() * -1);


        if(Math.abs(currentFish.getX() - xSub) < 25 && previousMovement == "right")
            currentFish.setSpeedX(6);
        else if(Math.abs(currentFish.getX() - xSub) < 25 && previousMovement == "left")
            currentFish.setSpeedX(-6);

        if(Math.abs(currentFish.getY() - ySub) < 25 && currentMovement == "down")
            currentFish.setSpeedY(6);
        else if(Math.abs(currentFish.getY() - ySub) < 25 && currentMovement != "down")
            currentFish.setSpeedY(-6);

        currentFish.setX(currentFish.getX() + currentFish.getSpeedX());
        currentFish.setY(currentFish.getY() + currentFish.getSpeedY());
    }

    moveSmallFish();
    drawFish();
}

function moveSmallFish(){
    var currentFish = smallFish;

        if(time % 2 < 0.1){
            currentFish.setSpeedX((Math.random() - 0.5) * 4);
            currentFish.setSpeedY((Math.random() - 0.5) * 4);
        }

        if((currentFish.getX() < 25) || (currentFish.getX() > canvasBoundX - 25))
            currentFish.setSpeedX(currentFish.getSpeedX() * -1);

        if((currentFish.getY() < 10 && currentFish.getSpeedY() < 0) || (currentFish.getY() > canvasBoundY - 10 && currentFish.getSpeedY() > 0))
            currentFish.setSpeedY(currentFish.getSpeedY() * -1);


```

```
                        currentFish.setX(currentFish.getX() + currentFish.getSpeedX());
                        currentFish.setY(currentFish.getY() + currentFish.getSpeedY());
                }

                function drawFish(){
                    var canvas = document.getElementById("myCanvas");
                    var context = canvas.getContext("2d");

                    for(var i = 0; i < fishes.length; i++){
                        if(fishes[i])

                        var image = document.getElementById(fishes[i].getImage());

                        context.drawImage(image, fishes[i].x, fishes[i].y, 100, 100);
                    }

                    var image = document.getElementById(smallFish.getImage());
                    context.drawImage(image, smallFish.x, smallFish.y, 75, 75);
                }

                function decayHealth(){
                    for(var i = 0; i < fishes.length; i++){
                        var currentFish = fishes[i];

                        if(Math.abs(currentFish.getX() - smallFish.getX()) < 50 && Math.abs(
                        currentFish.getY() - smallFish.getY()) < 50)
                            health -= 20;
                    }
                }


                //toggles the timer
                function toggleTimer(){
                    //if the timer is on the turn it off, if its not then turn it on
                    isOn ? turnOff() : turnOn();
                }

                //manually turns on the timer
                function turnOn(){
                    //turns the timer on
                    timer = setInterval("update()", 20);
                    isOn = true;
                }

                //manually turns off the timer
                function turnOff(){
                    //turns the timer off
                    clearInterval(timer);
                    isOn = false;
                }
            </script>
        </head>

        <body onload="initialize()">
            <center>
                <h1>Sub Program</h1>

        </br>
        </br>

            <canvas id="myCanvas" width="700" height="500"
                style="border:2px solid rgb(195, 195, 195);">
                Your browser does not suppport the canvas element
            </canvas>

            <img src="sub left.png" id="sub left" style="position: absolute; padding-left:
            10000px;">
```

```html
        <img src="sub right.png" id="sub right" style="position: absolute; padding-left: 10000px;">

        <img src="fish1Left.png" id="fish1Left" style="position: absolute; padding-left: 10000px;">
        <img src="fish2Left.png" id="fish2Left" style="position: absolute; padding-left: 10000px;">
        <img src="fish3left.png" id="fish3Left" style="position: absolute; padding-left: 10000px;">
        <img src="fish4Left.png" id="fish4Left" style="position: absolute; padding-left: 10000px;">
        <img src="fish5Left.png" id="fish5Left" style="position: absolute; padding-left: 10000px;">
        <img src="fish6Left.png" id="fish6Left" style="position: absolute; padding-left: 10000px;">
        <img src="fish7Left.png" id="fish7Left" style="position: absolute; padding-left: 10000px;">
        <img src="fish8Left.png" id="fish8Left" style="position: absolute; padding-left: 10000px;">
        <img src="fish9Left.png" id="fish9Left" style="position: absolute; padding-left: 10000px;">
        <img src="fish10Left.png" id="fish10Left" style="position: absolute; padding-left: 10000px;">

        <img src="fish1Right.png" id="fish1Right" style="position: absolute; padding-left: 10000px;">
        <img src="fish2Right.png" id="fish2Right" style="position: absolute; padding-left: 10000px;">
        <img src="fish3Right.png" id="fish3Right" style="position: absolute; padding-left: 10000px;">
        <img src="fish4Right.png" id="fish4Right" style="position: absolute; padding-left: 10000px;">
        <img src="fish5Right.png" id="fish5Right" style="position: absolute; padding-left: 10000px;">
        <img src="fish6Right.png" id="fish6Right" style="position: absolute; padding-left: 10000px;">
        <img src="fish7Right.png" id="fish7Right" style="position: absolute; padding-left: 10000px;">
        <img src="fish8Right.png" id="fish8Right" style="position: absolute; padding-left: 10000px;">
        <img src="fish9Right.png" id="fish9Right" style="position: absolute; padding-left: 10000px;">
        <img src="fish10Right.png" id="fish10Right" style="position: absolute; padding-left: 10000px;">

        <img src="littleFishLeft.png" id="littleFishLeft" style="position: absolute; padding-left: 10000px;">
        <img src="littleFishRight.png" id="littleFishRight" style="position: absolute; padding-left: 10000px;">
    </br>
    </br>
        </center>
    </body>
</html>
```