Project Source Code

```
<!DOCTYPE html>
<html>
   <head><title>DOM JavaScript</title>
       <style>
       .game{
           height: 400px;
           width:100%;
          background-color: #333;
       }

       .enemy{
           border: 1px solid white;
           border-radius: 50%;
           font-size: 0.8em;
           padding:2px;
           cursor: pointer;
           text-align: center;
           color:white;
           line-height: 30px;
           position: absolute;
           top:10px;
           left:10px;
           height: 30px;
           width:30px;
           background-color: blue;
       }
   </style>
   </head>
   <body>
       <h1>Hello World</h1>
       <div class="game">
       <div class="box"></div></div>
       <script src="app.js"></script>
   </body>
</html>

const box = document.querySelector('.box');
const h1Maker = document.querySelector('h1');
const gameBoard = document.querySelector('.game');
const dim = gameBoard.getBoundingClientRect();
```

```
const btn = document.createElement('button');
btn.textContent = 'START';
document.body.append(btn);
//console.log(dim);
const game = {
   size: 30,
   x: dim.left,
   y: dim.top,
   speed: 10,
   counter: 0,
   max: 10,
   score: 0,
   status: false
};
const enes = [];
const keyz = {
   ArrowRight: false,
   ArrowLeft: false,
   ArrowUp: false,
   ArrowDown: false
};
box.style.position = 'absolute';
box.style.border = '1px solid black';
box.style.width = game.size + 'px';
box.style.height = game.size + 'px';
box.style.top = game.y + 'px';
box.style.left = game.x + 'px';
box.style.backgroundColor = 'red';
let move = {};
btn.addEventListener('click',(e)=>{
   if(btn.textContent =='START'){
       btn.textContent = 'STOP';
       move = window.requestAnimationFrame(updatePos);
       game.status = true;
   }else{
       btn.textContent = 'START';
       game.status = false;
   }

})

updateScore();
```

```
window.addEventListener('keydown', (e) => {
   if (e.code in keyz) {
       keyz[e.code] = true;
   }
})
window.addEventListener('keyup', (e) => {
   if (e.code in keyz) {
       keyz[e.code] = false;
   }
})

h1Maker.addEventListener('click', (e) => {
   ////console.log(dim);
   const el = maker('div');
   ////console.log(el);
})

function updateScore(){
   h1Maker.innerHTML = `Score : ${game.score}`;
}

function isCol(aEl, bEl) {
   //console.log('checking');
   let a = aEl.getBoundingClientRect();
   let b = bEl.getBoundingClientRect();
   //console.log(a);
   //console.log(b);
   //let xAxis = !((a.right < b.left)||(a.left > b.right));
   //let yAxis = !((a.bottom < b.top)||(a.top > b.bottom));
   let overTop = !((a.right < b.left) || (a.left > b.right || (a.bottom < b.top) ||
(a.top > b.bottom)));
   //console.log(overTop);
   return overTop;
}



function maker(eleType) {
   game.counter++;
   const ele = document.createElement('div');
   ////console.log(ele);
   ele.textContent = game.counter;
```

```
    ele.classList.add('enemy');
    enes.push(ele);
    ele.style.left = ranNum(dim.left, dim.right - game.size) + 'px';
    ele.style.top = ranNum(dim.top, dim.bottom - game.size) + 'px';
    ele.style.backgroundColor = 'rgb(' + ranNum(0, 255) + ',' + ranNum(0, 255) + ',' +
ranNum(0, 255) + ')';
    ele.dirX = ranNum(1, 8);
    ele.addEventListener('click', (e) => {
        ////console.log(ele.offsetLeft);
        ////console.log(ele.offsetTop);
        //console.log(ele.dirX);
        ////console.log(enes);
        isCol(ele, box);
    })
    return gameBoard.appendChild(ele);
}

function ranNum(min, max) {
    return Math.floor(Math.random() * (max - min + 1) + min);
}



function updatePos() {
    ////console.log(keyz);
    if (keyz.ArrowLeft && game.x > dim.left) {
        game.x -= game.speed;
    } else if (keyz.ArrowRight && game.x < (dim.right - game.size - game.speed)) {
        game.x += game.speed;
    }
    if (keyz.ArrowUp && game.y > dim.top) {
        game.y -= game.speed;
    } else if (keyz.ArrowDown && game.y < (dim.bottom - game.size - game.speed)) {
        game.y += game.speed;
    }
    box.style.left = game.x + 'px';
    box.style.top = game.y + 'px';
    if (ranNum(0, 40) == 10 && game.max >= enes.length) {
        const newEle = maker('div');
        ////console.log(newEle);
    }
    //moveEnemies
```

```
    enes.forEach((enemy, index) => {
        ////console.log(enemy);
        let x = enemy.offsetLeft;
        let y = enemy.offsetTop;
        if (x > (dim.right - 30) || x < 0) {
            enemy.dirX *= -1;
        }
        x += enemy.dirX;

        if (isCol(enemy, box)) {
            console.log('hit');
            enemy.remove();
            game.score++;
            updateScore();
            enes.splice(index, 1);
        };

        enemy.style.left = x + 'px';
    })
    if(game.status){
    move = window.requestAnimationFrame(updatePos);
}
}
```