Eggstermination

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Freek's Space Invaders</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="gamecontainer">
        <div id="player" class="player"></div>
        <div id="invaders" class="invaders"></div>
    </div>
    <div class="score">Score: <span id="score">0</span></div>
    <div class="video-overlay">
        <video autoplay muted loop>
            <source src="videos/crt.mp4" type="video/mp4">
        </video>
    <script src="script.js"></script>
</body>
</html>
```

style.css

```
body {
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-image: url(images/achtergrond2.png);
    background-size: cover;
    background-repeat: no-repeat;
    flex-direction: column;
    overflow: hidden;
}

.video-overlay {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
```

```
    z-index: -1; /* Behind other content */
    opacity: 0.1; /* Low transparency */
    pointer-events: none; /* Allow clicks to pass through */
    z-index: 1;
}

.video-overlay video {
    width: 100%;
    height: 100%;
    object-fit: cover;
}

.gamecontainer {
    position: relative;
    width: 600px;
    height: 600px;
    background-image: url(images/gras2.jpg);
    background-size: cover;
    background-repeat: no-repeat;
    overflow: hidden;
    border: 5px solid black;
}

.player {
    position: absolute;
    bottom: 10px;
    left: 50%;
    width: 80px;
    height: 60px;
    background-image: url(images/kip.png);
    background-size: cover;
    background-repeat: no-repeat;
    transform: translateX(-50%);
}

.invader {
    position: absolute;
    width: 70px;
    height: 50px;
    background-image: url(images/kat.png);
    background-size: cover;
    background-repeat: no-repeat;
}

.bullet {
    position: absolute;
    width: 20px;
    height: 20px;
```

```
    background-image: url(images/egg.png);
    background-size: cover;
    background-repeat: no-repeat;
}

.score {
    color: white;
    font-size: 24px;
    margin-top: 20px;
}
```

script.js

```
const player = document.getElementById('player');
const gameContainer = document.querySelector('.gamecontainer');
const invadersContainer = document.getElementById('invaders');
const scoreElement = document.getElementById('score');

let PLAYERPOSITION = 275; // start positie van de player
let BULLETARRAY = []; // lege array die de kogels van de speler opslaat
let INVADERSARRAY = [];
let INVADERSDIRECTION = 10;
let GAMEINTERVAL;
let lastShotTime = 0; // voorkomt dat er niet te snel achter elkaar bullets
worden geschoten
const SHOT_DELAY = 500; // aantal ms dat je moet wachten per bullet
let level = 1; // spelniveau, elk level = 5 nieuwe vijanden
let score = 0;

// beweging van de player
document.addEventListener('keydown', (e) => {
    if (e.key === 'ArrowLeft' && PLAYERPOSITION > 0) {
        PLAYERPOSITION -= 10;
        /* als jij (de player) het pijltje naar links indrukt en niet al
helemaal tegen de muur staat,
        dan verplaats je 10px naar links */
        console.log('BEWOGEN NAAR:', PLAYERPOSITION);
        // in de console wordt vermeld op welke positie de player dan op het
moment staat
    } else if (e.key === 'ArrowRight' && PLAYERPOSITION < 550) {
        PLAYERPOSITION += 10;
        console.log('BEWOGEN NAAR:', PLAYERPOSITION);
    } else if (e.key === ' ') {
        shootBullet();
        // als de spatiebalk wordt ingedrukt, wordt er een bullet afgeschoten
        // ' ' laat in code zien dat het om een spatiebalk gaat
    }
```

```
    player.style.left = PLAYERPOSITION + 'px';
});

// functie die de players bullets laat schieten
function shootBullet() {
    const currentTime = Date.now(); // haalt de huidige tijd op
    if (currentTime - lastShotTime < SHOT_DELAY) {
        return;
    }
    /* als er minder dan 500ms voorbij zijn sinds het laatste schot
    stopt de functie die voorkomt dat je niet snel achter elkaar kunt schieten
    en kan je weer een bullet afvoeren */
    lastShotTime = currentTime;

    const bullet = document.createElement('div');
    bullet.classList.add('bullet');
    // een div wordt aangemaakt en opgeslagen in de variabele bullet, en
krijgt de bullet css stijl
    bullet.style.left = PLAYERPOSITION + 'px';
    // de horizontale positie van de bullet is gebaseerd op waar de player
staat
    bullet.style.bottom = '50px';
    gameContainer.appendChild(bullet);
    /* voegt de bullet toe in de gameContainer
    voegt het element toe als een sub-element */
    BULLETARRAY.push(bullet);
    // houdt elk geschoten bullet bij en slaat deze op in een array
}

// functie die de bullets movement bepaalt
function moveBullets() {
    BULLETARRAY.forEach((bullet, index) => {
        let bulletBottom = parseInt(bullet.style.bottom); // verticale positie
van de bullet
        bullet.style.bottom = bulletBottom + 20 + 'px';
        if (bulletBottom > 600) {
            bullet.remove();
            // als de kogel de bovenkant van het scherm raakt (600), wordt hij
verwijder uit de gamecontainer
        }
    });
}

// maakt vijanden ("invaders")
function createInvaders(level) { // (level) geeft het huidige level door en
bepaald hoeveel invaders er moeten komen
    for (let i = 0; i < level * 5; i++) { // aantal invaders is afhankelijk
van het level (komen +5 bij elk level)
```

```
        const invader = document.createElement('div');
        invader.classList.add('invader');
        const row = Math.floor(i / 5); // bepaalt de rij van de vjiand (5
vijanden = 1 rij)
        const col = i % 5; // bepaalt de kolom van de vijand (elke rij heeft
er 5)
        invader.style.left = (col * 60) + 'px';
        invader.style.top = (50 + row * 50) + 'px';
        invadersContainer.appendChild(invader); // voegt de invaders toe aan
invadersContainer
        INVADERSARRAY.push(invader); // voegt de invaders toe aan de invaders
array
    }
}

// functie die de "invaders" beweegt
function moveInvaders() {
    let shouldChangeDirection = false;
    // houdt bij of de invaders van richting moeten veranderen (als ze de muur
raken)
    INVADERSARRAY.forEach(invader => {
        let invaderLeft = parseInt(invader.style.left);
        invader.style.left = invaderLeft + INVADERSDIRECTION + 'px';

        if (parseInt(invader.style.left) <= 0 || parseInt(invader.style.left)
>= 530) {
            shouldChangeDirection = true;
            // als de invader de borders van de gamecontainer raken, moeten ze
van richting veranderen
        }
    });

    if (shouldChangeDirection) {
        INVADERSDIRECTION *= -1;
        // als ^ true is, moet de richting veranderen
        let i = 0;
        while (i < INVADERSARRAY.length) {
        // na de richting veranders, beingt er een while loop die kijkt naar
de locatie van de invaders
            let invader = INVADERSARRAY[i];
            invader.style.top = parseInt(invader.style.top) + 20 + 'px';
            i++;
        }
        /* de while loop gaat door, totdat i gelijk is aan
INVADERSARRAY.length
        dan eindigt de lus en begint deze na de richting verandering weer
opnieuw */
    }
```

```
}

// functie die de collision regelt (zorgt ervoor dat de invaders geraakt
kunnen worden door bullets)
function checkCollisions() {
    BULLETARRAY.forEach((bullet, bulletIndex) => {
        INVADERSARRAY.forEach((invader, invaderIndex) => {
            const bulletRect = bullet.getBoundingClientRect();
            const invaderRect = invader.getBoundingClientRect();
            const gameRect = gameContainer.getBoundingClientRect();
            // haalt de exact positie en afmetingen van de bullet, invaders en
gamecontainer op.

            if (
                bulletRect.top <= invaderRect.bottom &&
                bulletRect.bottom >= invaderRect.top &&
                bulletRect.left <= invaderRect.right &&
                bulletRect.right >= invaderRect.left
                // checkt of de bullet positie en de invader positie
overlappen
            ) {
                bullet.remove();
                invader.remove();
                INVADERSARRAY.splice(invaderIndex, 1);
                score += 100;
                scoreElement.textContent = score;
                // verwijdert de bullet en de invader, van het spel en de
array, en verhoogt de score
                // deze score update wordt ook gedisplayed onder de
gamecontainer
            }
        });
    });
}

// functie die de game reset
function resetGame() {
    PLAYERPOSITION = 275;
    level = 1;
    score = 0;
    scoreElement.textContent = score;

    // verwijder bullets uit gamecontainer
    BULLETARRAY.forEach(bullet => bullet.remove());
    BULLETARRAY = [];

    // verwijder invaders uit 'invadersContainer'
    INVADERSARRAY.forEach(invader => invader.remove());
```

```
    INVADERSARRAY = [];
    invadersContainer.innerHTML = '';

    // nieuwe invaders, reset deze naar de startpositie
    INVADERSDIRECTION = 10;
    createInvaders(level);

    // stop en restart de gameloop
    clearInterval(GAMEINTERVAL);
    GAMEINTERVAL = setInterval(gameLoop, 50);
}

// functie die controleert of de speler heeft verloren
function checkLossCondition() {
    const playerRect = player.getBoundingClientRect();
    INVADERSARRAY.forEach(invader => {
        const invaderRect = invader.getBoundingClientRect();
    /* getBoundingClientRect zorgt ervoor dat de positie en afmeting van de
player worden afgenomen
    en opgeslagen in een DOMRect object
    in een DOMRect object staat dus al deze informatie in (x, y, top, right,
etc.)*/
        if (
            Math.abs(playerRect.bottom - invaderRect.top) < 1 &&
            // checkt of de onderkant van de player bijna gelijk is als de
bovenkant van een invader
            Math.abs(playerRect.left - invaderRect.left) < 60
            // checkt of het horizontale verschil tussen de player en invader
minder dan 60px is
            // Math.abs berekent de waarde van een getal
        ) {
            clearInterval(GAMEINTERVAL);
            alert("GAME OVER.");
            resetGame();
            // als dit het geval met een van de invaders en de player, eindigt
het spel
        }
    });
}

// functie die de game laat loopen
function gameLoop() {
    moveBullets();
    moveInvaders();
    checkCollisions();
    checkLossCondition();

    if (INVADERSARRAY.length === 0) {
```

```
        level++; // als er geen invaders meer zijn, verhoogt het level met 1
        createInvaders(level);
        console.log('LEVEL:', level);
    }
}
// GAME START! (na een loop)
createInvaders(level);
GAMEINTERVAL = setInterval(gameLoop, 50);
```