

# **Projekt zaliczeniowy JavaScript**

**Norbert Drąg lab. 1**

# 1. Temat projektu

Ping-pong napisany w javascriptcie.

## 2. Autor

Norbert Drąg

## 3. Link do GitHuba

<https://github.com/ZurisW/Projekt-JS>

## 4. Opis

Sterujemy myszką naszą paletką, podczas gdy wrogi AI próbuje odbić piłeczkę w naszą stronę. Gdy piłeczka dotyka krańców planszy, odbija się ona i przyspiesza. Jeśli piłeczka wleci któremuś z graczy za paletkę partia kończy się a zwycięski gracz otrzymuje punkt.

## 5. Wybrane fragmenty kodu z wyjaśnieniem

```
function table() {  
    ctx.drawImage(image, 0, 0, cw, ch);  
  
    for (let linePosition = 20; linePosition < ch; linePosition += 30) {  
        ctx.fillStyle = "white";  
        ctx.fillRect(cw / 2 - lineWidth / 2, linePosition, lineWidth, lineHeight)  
    }  
}  
  
table();
```

Wyświetlenie linii przerywanej, która oddziela połówki graczy. Wypełnienie miejsca boiska zdjęciem oraz linii przerywanej kolorem białym. Rysowane są one co 10 ms.

```
const canvas = document.querySelector('canvas');  
const ctx = canvas.getContext('2d');  
canvas.width = 1000;  
canvas.height = 500;  
const cw = canvas.width;  
const ch = canvas.height;  
  
//linia przerywana  
const lineWidth = 6;  
const lineHeight = 16;  
  
const image = new Image();  
image.src = "https://img.freepik.com/darmowe-zdjecie/tlo-pola-zielon";  
  
const ballSize = 20;  
let ballX = cw / 2 - ballSize / 2;  
let ballY = ch / 2 - ballSize / 2;
```

Ustawienie canvasa, jego długości i szerokości. Ustawienie miejsca X i Y oraz wielkości piłeczki.

Stworzenie elementu Image, który użyty będzie jako pole do gry.

```

46 function ball(){
47     ctx.fillStyle = 'ffffff';
48     ctx.fillRect(ballX, ballY, ballSize, ballSize);
49     ballX += ballSpeedX;
50     ballY += ballSpeedY;
51
52     //punkty
53     if (ballX <= 0) {
54         aiptk++;
55         reset();
56         wynik();
57     }
58     if (ballX + ballSize >= cw) {
59         playerptk++;
60         reset();
61         wynik();
62     }
63
64     if (ballY <= 0 || ballY + ballSize >= ch){
65         ballSpeedY = -ballSpeedY;
66         speedUp();
67     }
68     if (ballX <= 0 || ballX + ballSize >= cw || (ballX <= playerX + paddleWidth + ballSize / 2 && ballY >= playerY && ballY <= playerY + paddleHeight) ||
69         (ballX >= aiX - paddleWidth - ballSize / 2 && ballY >= aiY && ballY <= aiY + paddleHeight) ){
70         ballSpeedX = -ballSpeedX;
71         speedUp();
72     }
73 }

```

Funkcja ball() sprawdza czy piłka wpadła za paletki graczy oraz resetuje grę i pokazuje wynik. Przyspiesza piłeczkę po odbiciu się od paletki bądź od krawędzi.

```

function reset() {
    playerY = 200;
    aiY = 200;
    ballX = cw / 2 - ballSize / 2;
    ballY = ch / 2 - ballSize / 2;
    ballSpeedX = 2;
    ballSpeedY = 2;

    console.log("Player: "+playerptk+" AI: "+aiptk);
}

```

Resetowanie gry oraz ustawianie właściwości piłki do kolejnych gier.

```

topCanvas = canvas.offsetTop;

function playerPosition(e) {
    playerY = e.clientY - topCanvas - paddleHeight / 2;

    if (playerY >= ch - paddleHeight) {
        playerY = ch - paddleHeight
    }
    if (playerY <= 0) {
        playerY = 0;
    }
}

canvas.addEventListener("mousemove", playerPosition)

```

Nastuchiwanie przy ruchu myszki o zmianie pozycji gracza. Funkcja dba też o to, by paletka gracza nie wyszła poza canvas.

```

function aiPosition() {
  const middlePaddle = aiY + paddleHeight / 2;
  const middleBall = ballY + ballSize / 2;
  if (ballX > 500) {
    if (middlePaddle - middleBall > 200) {
      aiY -= 5;
    } else if (middlePaddle - middleBall > 50) {
      aiY -= 5;
    } else if (middlePaddle - middleBall < -200) {
      aiY += 5;
    } else if (middlePaddle - middleBall < -50) {
      aiY += 5;
    }
  }
  if (ballX <= 500 && ballX > 100) {
    if (middlePaddle - middleBall > 100) {
      aiY -= 2;
    }
    if (middlePaddle - middleBall < -100) {
      aiY += 2;
    }
  }
  if (aiY >= ch - paddleHeight) {
    aiY = ch - paddleHeight;
  }
  if (aiY <= 0) {
    aiY = 0;
  }
}

```

Funkcja ta steruje wrogą paletką. Przy określonych warunkach AI porusza się o pięć pixeli, zaś w innych o dwa, przez to da się wygrać z wrogim AI przy stosownej prędkości piłeczki, gdyż w pewnym momencie jest ono za wolne.

Ostatnie dwa ify uniemożliwiają wyjście wrogiej paletce poza obszar gry.