



CodeCamp JavaScript

Algorithmics Brasov-Smart Kids

Ziua 3

Funcții și Joculeț Pacman în JavaScript

Introducere

Astăzi vom învăța programare printr-un joc clasic și distractiv: Pac-Man.

Vom construi pas cu pas un mini-joc unde Pac-Man se mișcă prin labirint, mănâncă puncte și trebuie să evite fantoma.

Introducere

Obiectivele zilei:

- Să înțelegem cum se construiește o hartă de joc folosind o matrice 2D în JavaScript.
- Să descoperim cum folosim culorile și formele în CSS pentru a diferenția zidurile, punctele, Pac-Man și fantoma.
- Să vedem cum reacționează jocul la input-ul utilizatorului (tastele săgeți și WASD).
- Să explorăm interactivitatea vizuală: cum desenăm tabla de joc pe ecran și cum se actualizează automat mișcarea fantomei.
- Să aplicăm noțiunile de bază din JavaScript: variabile, funcții, bucle și evenimente.



⚡ Funcții în JavaScript – Noțiuni de bază

Ce este o funcție?

Este un „set de instrucțiuni” pe care îl putem folosi de mai multe ori.

Gândim o funcție ca pe o rețetă: îi dai ingrediente (parametri) și ea face ceva pentru tine.

Cum arată o funcție

```
function salut(ume) {  
  console.log("Salut, " + ume + "!");  
}  
salut("Ana"); // afișează: Salut, Ana!
```

- *function* → cuvânt cheie pentru a crea funcția
- *salut* → numele funcției
- *(ume)* → parametrul
- *{ ... }* → instrucțiunile care se execută

⚡ Funcții în JavaScript – Noțiuni de bază

De ce folosim funcții?

- Să nu scriem **același cod de mai multe ori**
- Să facem codul **mai organizat și ușor de citit**

🏃 Exerciții practice

1. Creează o funcție adauga care primește **două numere** și afișează suma lor.
2. Creează o funcție mesajBunVenit care primește **numele unui prieten** și afișează un mesaj de bun venit.
3. Creează o funcție deseneazaPatrat care primește **culoarea** și afișează în consolă un mesaj: „Desenez un pătrat <culoare>”.

✂ Funcții în JavaScript - Noțiuni de bază

JS

```
1 // Funcția care adaugă două numere și afișează suma lor
2 function adauga(a, b) {
3     let suma = a + b;
4     console.log("Suma este:", suma);
5 }
6
7 // Funcția care afișează un mesaj de bun venit
8 function mesajBunVenit(ume) {
9     console.log(`Bun venit, ${ume}!`);
10 }
11
12 // Funcția care afișează un mesaj despre desenarea unui pătrat
13 function deseneazaPatrat(culoare) {
14     console.log(`Desenez un pătrat ${culoare}`);
15 }
16
17 // Exemple de apelare a funcțiilor
18 adauga(3, 5); // Suma este: 8
19 mesajBunVenit("Andrei"); // Bun venit, Andrei!
20 deseneazaPatrat("roșu"); // Desenez un pătrat roșu
```

Cum se împart rolurile pe web

- HTML – structura paginii (elemente, containere, text)
- CSS – stiluri și layout (culori, grid, forme)
- JavaScript – comportament: desenare, mișcare, coliziuni, scor
- DOM – „arborele” de elemente la care ajungem din JavaScript

Mini PacMan

1. Obiectiv

Crearea unui mini-joc Pac-Man în browser, unde Pac-Man mănâncă puncte și evită fantoma

2. Descriere

- Labirint 12×12 cu ziduri, puncte, Pac-Man și o fantomă.
- Pac-Man se mișcă cu tastele săgeți sau WASD.
- Scorul crește când Pac-Man mănâncă puncte.
- Jocul se termină la coliziune cu fantoma sau când toate punctele sunt mâncate

3. Funcționalități

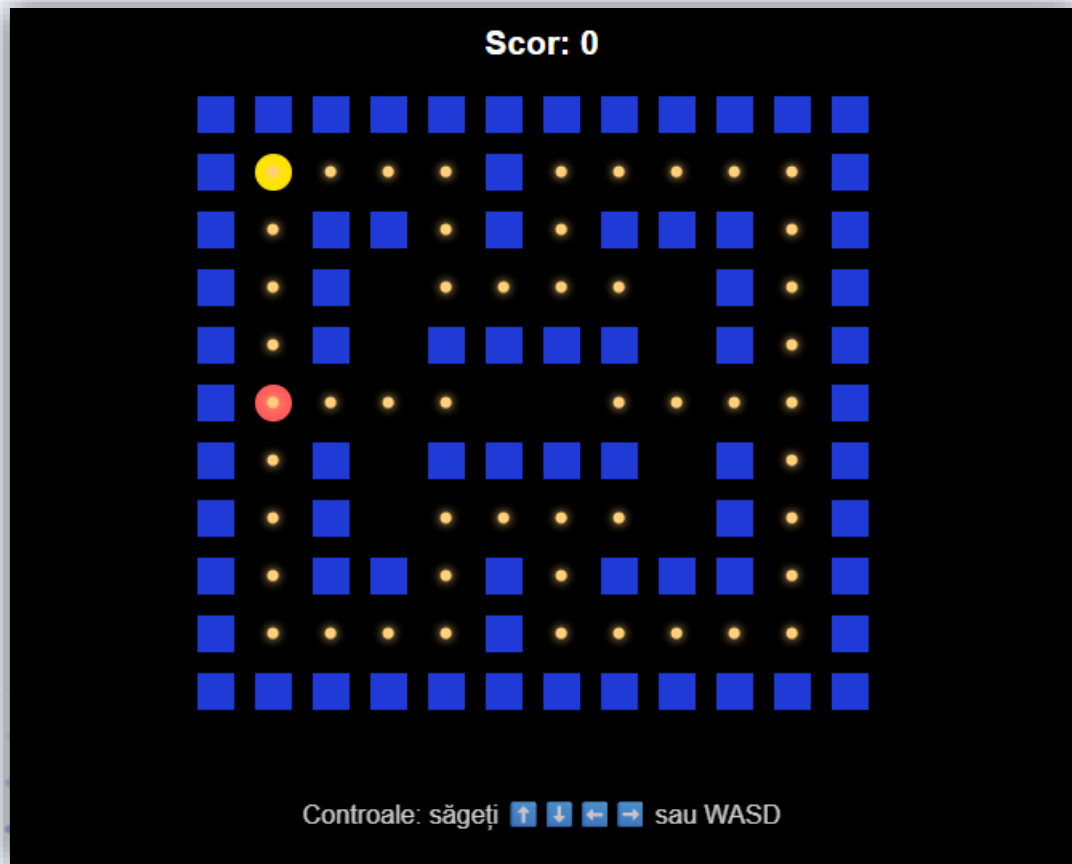
- Desenarea tablei și pozițiilor pe grilă.
- Mișcare Pac-Man la tastatură și fantomă automat.
- Colectare puncte și actualizare scor.
- Detectarea coliziunilor și condiții de câștig/pierdere

4. Tehnologii

- HTML/CSS – structură și aspect vizual.
- JavaScript – logică, mișcare, coliziuni , scor, evenimente tastatură.

Ce vom realiza

Link-ul proiectului CodePen:
<https://codepen.io/Violeta-the-sans/pen/JoYZoQx?editors=1010>







Plan de lucru în 10 pași

1. HTML: conținutul jocului și texte
2. CSS: grid-ul și aspectul (zid, punct, pacman, fantomă)
3. Harta jocului ca matrice (0/1/2)
4. Variabile: poziții, scor, constante
5. Selectarea elementelor din DOM
6. Desenul tablei (for + createElement)
7. Mișcarea jucătorului + colectarea punctelor
8. Mișcarea fantomei (aleator) + buclă de timp
9. Coliziuni și condiții de final (win/lose)
10. Controale de la tastatură (săgeți/WASD)

HTML: structura minimă

Această bucată de HTML creează „scheletul” jocului:

- **<div class="wrap">** → un container care grupează tot jocul.
- **<h3>Scor: 0</h3>** → afișează textul „Scor” și valoarea curentă a scorului; id="score" ne permite să schimbăm scorul din JavaScript.
- **<div id="game"></div>** → zona unde se va desena tabla de joc (grid-ul cu ziduri, puncte, Pac-Man și fantoma).
- **<p class="hint">...</p>** → un mic mesaj pentru utilizator, care explică ce taste poate folosi pentru a controla jocul.

```
<div class="wrap">
  <h3>Scor: <span id="score">0</span></h3>
  <div id="game"></div>
  <p class="hint">Controale: săgeți     sau WASD</p>
</div>
```

CSS: layout și grid

- `body` → fundal negru, text alb, conținutul centrat pe ecran.
- `.wrap` → centrează textul (scorul și instrucțiunile).
- `#game` → definește tabla ca un grid 12x12, cu celule pătrate.
- `.cell` → dimensiunea de bază pentru fiecare pătrățel.
- `.wall` → colorează zidurile cu albastru.
- `.dot` și `.dot::after` → un mic punct galben strălucitor (mâncarea).
- `.pacman` → cerc galben (Pac-Man).
- `.ghost` → cerc roșu (fantoma).
- `.hint` → textul de ajutor (controale) apare mai mic și puțin transparent.

👉 Pe scurt: CSS-ul face ca jocul să arate ca un labirint colorat, cu Pac-Man, fantoma și punctele vizibile.



Harta jocului: matrice 2D

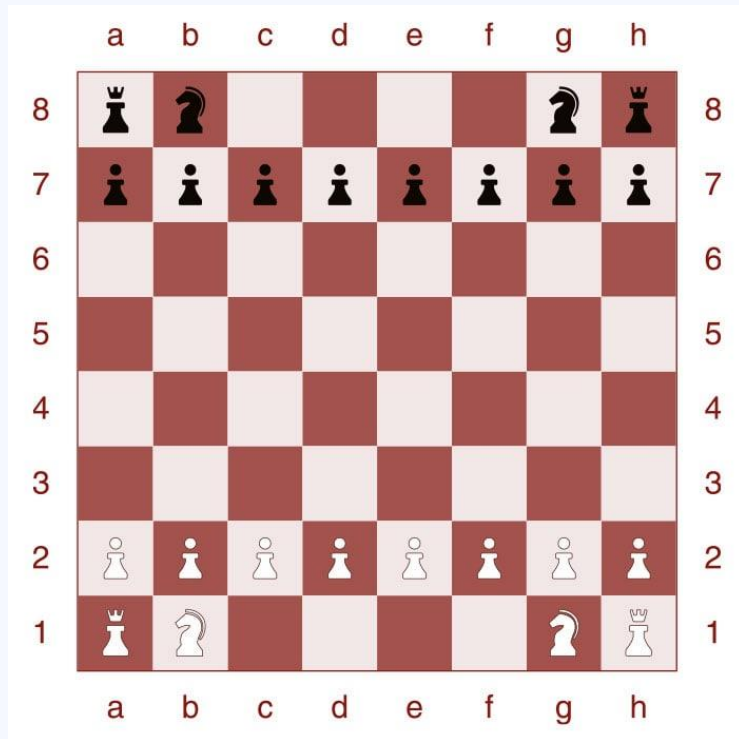
O **matrice 2D** este un **tabel cu rânduri și coloane**, unde putem stoca informații într-un mod organizat.

- **Rânduri** → orizontale
- **Coloane** → vertical

Gândește-te la **tabla de șah** sau la un **tabel de Excel**: fiecare pătrățel are o poziție unică, indicată prin rând și coloană.

	Column 1	Column 2	Column 3	Column 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

Harta jocului: matrice 2D



	Column 1	Column 2	Column 3	Column 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

Harta jocului: matrice 2D

- Reprezentăm tabla ca un tablou de tablouri (rânduri/coloane).
- Legendă: 0=gol, 1=zid, 2=punct (de mâncat).
- ROWS și COLS – derivează dimensiunile hărții.

```
const map = [  
  [1,1,1,1,1,1,1,1,1,1,1,1],  
  [1,2,2,2,2,1,2,2,2,2,2,1],  
  // ...  
  [1,1,1,1,1,1,1,1,1,1,1,1],  
];  
const ROWS = map.length;  
const COLS = map[0].length;
```


Harta jocului: matrice 2D

Legendă:

- 0=gol
- 1=zid
- 2=punct (de mâncat)

1	2	0
0	2	1
2	0	0

Unde se află punctul de mâncat?

Starea jocului

- Scorul: 'score', elementele DOM: 'gameEl', 'scoreEl'.
- Pozițiile: obiecte cu rând/coloană – 'pac = { r:1, c:1 }', 'ghost = { r:7, c:7 }'.
- 'isWall(r,c)' verifică atât zidurile cât și ieșirea din hartă.

```
const gameEl = document.getElementById('game');
const scoreEl = document.getElementById('score');
let score = 0;
const pac = { r: 1, c: 1 };
const ghost = { r: 7, c: 7 };
function isWall(r, c) {
    return !(map[r] && map[r][c] !== undefined) ||
    map[r][c] === 1;
}
```

Desenarea tablei pe DOM

- Curățăm conținutul și reconstruim grila la fiecare cadru.
- Parcurgem rândurile și coloanele, creăm un `<div>` pentru fiecare celulă.
- Adăugăm clase în funcție de map și de pozițiile pac/fantomă.

```
function draw() {  
  gameEl.innerHTML = '';  
  for (let r = 0; r < ROWS; r++) {  
    for (let c = 0; c < COLS; c++) {  
      const cell = document.createElement('div');  
      cell.className = 'cell';  
      if (map[r][c] === 1) cell.classList.add('wall');  
      else if (map[r][c] === 2) cell.classList.add('dot');  
      if (r === pac.r && c === pac.c) cell.classList.add('pacman');  
      if (r === ghost.r && c === ghost.c) cell.classList.add('ghost');  
      gameEl.appendChild(cell);  
    }  
  }  
}
```

Mișcarea jucătorului și scor

- Calculăm noua poziție cu 'dr/dc' (delta rând/coloană).
- Dacă nu este zid: actualizăm, mâncăm punctele (2→0), creștem scorul.
- Redesenăm și verificăm coliziunea și victoria.

```
function movePlayer(dr, dc) {  
  const nr = pac.r + dr, nc = pac.c + dc;  
  if (!isWall(nr, nc)) {  
    pac.r = nr;  
    pac.c = nc;  
    if (map[nr][nc] === 2) {  
      map[nr][nc] = 0;  
      score += 10;  
      scoreEl.textContent = score;  
      checkWin();  
    }  
  }  
  draw();  
  checkCollision();  
}
```

Mișcarea fantomei și bucla de timp

- Alegem o direcție aleatoare din cele 4 posibile.
- Dacă nu e perete: mutăm fantoma.
- Apelăm periodic cu `setInterval(moveGhost, 500)`.

```
function moveGhost() {  
  const dirs = [[1,0],[-1,0],[0,1],[0,-1]];  
  const [dr, dc] = dirs[Math.floor(Math.random() * dirs.length)];  
  const nr = ghost.r + dr, nc = ghost.c + dc;  
  if (!isWall(nr, nc)) { ghost.r = nr; ghost.c = nc; }  
  draw();  
  checkCollision();  
}  
setInterval(moveGhost, 500);
```

Coliziuni și condiții de final

- Coliziune: pac și fantoma pe aceeași celulă → Game Over.
- Victorie: nu mai există niciun '2' (punct) pe hartă.
- Momentan folosim 'alert' + 'location.reload()'.

```
function checkCollision() {  
  if (pac.r === ghost.r && pac.c === ghost.c) {  
    alert('Game Over! Scor: ' + score);  
    location.reload();  
  }  
}  
  
function checkWin() {  
  for (let r = 0; r < ROWS; r++) {  
    for (let c = 0; c < COLS; c++) {  
      if (map[r][c] === 2) return;  
    }  
  }  
  alert('Ai câștigat! Scor: ' + score);  
  location.reload();  
}
```

Controale de la tastatură

- Ascultăm `keydown` și prevenim scroll-ul cu săgețile.
- Suportăm săgeți și WASD pentru accesibilitate.
- Apelăm `movePlayer` cu direcția potrivită.

```
document.addEventListener('keydown', (e) => {  
  const keys =  
  ['ArrowUp', 'ArrowDown', 'ArrowLeft', 'ArrowRight', 'w', 'a', 's', 'd'];  
  if (keys.includes(e.key)) e.preventDefault();  
  if (e.key === 'ArrowUp' || e.key === 'w') movePlayer(-1, 0);  
  if (e.key === 'ArrowDown' || e.key === 's') movePlayer(1, 0);  
  if (e.key === 'ArrowLeft' || e.key === 'a') movePlayer(0, -1);  
  if (e.key === 'ArrowRight' || e.key === 'd') movePlayer(0, 1);  
});  
draw();
```


Provocări viitoare posibile (extensii)

- ✓ Mai multe fantome
- ✓ Power-up: puncte mari care încetinesc fantomele.
- ✓ Niveluri: hărți diferite și viteze crescătoare
- ✓ Pathfinding simplu: fantoma urmărește direcția spre Pac
- ✓ Sunete, animații, ecran de start/game over personalizat

Ce am învățat azi?

Recapitulare



- ❑ Hartă 2D: 0 = gol, 1 = zid, 2 = punct
- ❑ Redarea grilei și elementelor: Pac-Man, fantome, punct
- ❑ Mișcarea jucătorului și a fantomelor
- ❑ Detectarea coliziunilor și gestionarea scorului
- ❑ Control cu taste (săgeți / W-A-S-D)
- ❑ Verificarea câștigului sau pierderii

