

Memory Card Game

Autor: Delea Cynthia- Andreea, 332AB

An: 2022

Cuprins

1	Introducere	3
1.1	Prezentarea temei.....	3
1.2	Prezentarea obiectivelor.....	4
2	Prezentarea suportului tehnic	4
2.1	Noțiuni teoretice.....	4
2.2	Realizări similare	6
3	Etapa de implementare	7
4	Mod de utilizare	9
4.1	Scenariu de utilizare:.....	10
5	Concluzii	12
6	Bibliografie	13

1 Introducere

1.1 Prezentarea temei

Jocurile de memorie constituie o metodă rapidă și accesibilă prin care ne putem asigura de menținerea la un nivel cât mai sănătos al creierului. Putem spune că acest lucru este unul din beneficiile importante ale acestui tip de joc. De cele mai multe ori, jocurile de memorie sunt destinate copiilor pentru a le stimula în permanență creierul și pentru a-i ajuta să învețe lucruri noi mai simplu. S-a descoperit că jocurile din această categorie au efecte benefice asupra copiilor dislexici în ceea ce privește creșterea dezvoltării cognitive, a atenției, a activității creierului, a memoriei de lucru, a reamintirii și a performanței de citire.

Conform [1], memoria este abilitatea creierului de a codifica, stoca și de a reaminti ulterior informații, senzații, sentimente atunci când sunt necesare, în mod automat sau conștient. Altfel spus, este capacitatea de a utiliza situații învățate sau experimentate din trecut pentru a rezolva și anticipa evenimente din prezent, respectiv viitor. Memoria influențează comportamentul și este parte din procesul continuu de adaptare la mediu cu tot ce presupune acesta. Exista mai multe tipuri de memorie, în funcție de modul în care se manifesta, de reacțiile și de efectele produse la nivelul creierului:

- **memoria senzorială**, ce este capacitatea de a reține senzații provocate de stimuli vizuali, auditivi sau tactili, pentru un timp scurt, chiar după ce sursa excitației a încetat să mai acționeze.
- **memoria de scurtă durată**, folosită pentru a gestiona informațiile reținute temporar și a exersa memoria permanentă;
- **memoria de lungă durată**, ce se referă la acțiunile pe care le putem face cu precizie și rapiditate fără a urma un set de pași în acel moment;

Acuitatea vizuală joacă un rol vital în capacitatea copilului de a identifica, înțelege, evalua și interpreta un set de obiecte diferite față de altele, inclusiv simboluri, litere, culori, forme, modele, dimensiuni etc. Jocurile de memorare pot ajuta la îmbunătățirea abilităților de percepție vizuală la un copil, până la un punct în care sunt capabili să vizualizeze mental rezultatele cu puține încercări sau erori. Astfel de jocuri pot ajuta, de asemenea, la îmbunătățirea limbajului și a vocabularului.

Memory Card Game are ca scop potrivirea pe perechi a unui set de imagini într-un număr limitat de încercări, contribuind astfel la consolidarea abilităților individuale precum timpul de reacție, concentrarea, atenția, memoria și percepția.

1.2 Prezentarea obiectivelor

Obiectivele principale ale proiectului constau în:

- realizarea unei interfețe cu utilizatorul unde acesta poate vizualiza atât cărțile de joc, cât și numărul de vieți rămase;
- realizarea unei animații 3D ce apare atunci când cartea de joc este întoarsă în urma apăsării acesteia de către jucător;
- decrementarea numărului de vieți rămase de fiecare dată când jucătorul nu găsește perechea de cărți;
- implementarea opțiunii de a putea reîncepe jocul după ce jucătorul a câștigat sau a pierdut;
- amestecarea cărților de joc de fiecare dată când jucătorul pierde și reîncepe jocul;
- afișarea unui mesaj ce informează jucătorul dacă a pierdut sau a câștigat;
- potrivirea corectă de către jucător a cărților.

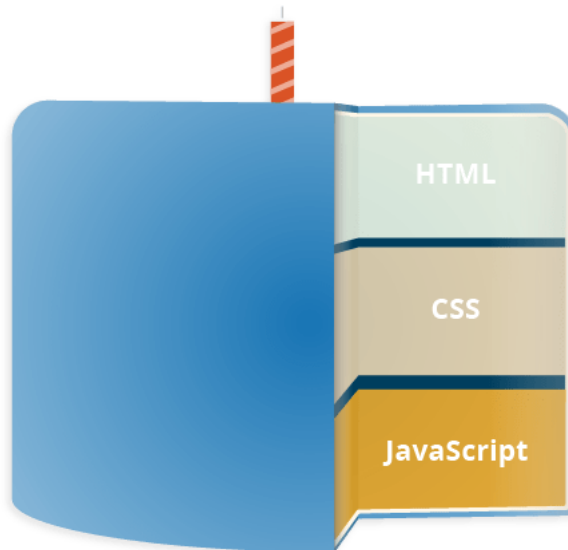
2 Prezentarea suportului tehnic

2.1 Noțiuni teoretice

Acest tip de joc este disponibil atât online, cât și ca aplicație pe dispozitivele mobile. Unul dintre mediile de dezvoltare ce favorizează crearea unui astfel de joc este, de exemplu, Visual Studio Code, fiind în legătură cu limbaje de programare precum C#, C++, JavaScript, Python, CSS, HTML, etc.

Am ales să folosesc următoarele tehnologii: JavaScript, CSS, HTML - ce fac parte din tehnologiile web standard. Pentru a învăța baza platformei VisualStudioCode în combinație cu limbajele HTML, CSS, JavaScript am urmărit tutorialul [2].

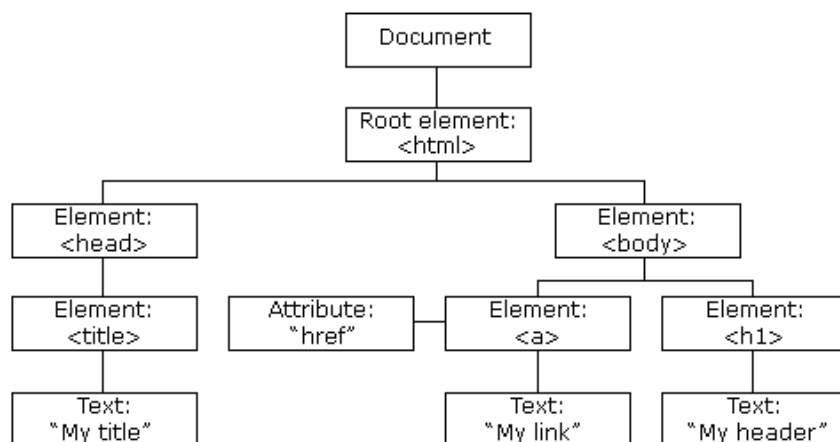
Aceste trei straturi se construiesc și completează unul peste altul, putând fi asemănate cu straturile unui tort.



Figură 1. Stratificarea tehnologiilor web standard

- a) JavaScript este un limbaj de scripting sau de programare care permite implementarea de caracteristici complexe pe paginile web (actualizare dinamică, animații 2D, grafică 3D, hărți interactive, etc) [3]
- b) CSS(Cascading Style Sheets) este un limbaj de design menit să simplifice procesul de creare a unor pagini web prezentabile. Folosind CSS, se poate controla culoarea textului, stilul fonturilor, distanța dintre paragrafe, modul în care sunt dimensionate și aranjate coloanele, ce imagini sau culori de fundal sunt folosite, modele de aspect, variațiile de afișare pentru diferite dispozitive și dimensiuni de ecran; cel mai frecvent este combinat cu limbaje de marcare HTML sau XHTML. [4]
- c) HTML (HyperText Markup Language) care alcătuiește majoritatea paginilor web și aplicațiilor online. Un hipertext este un text ce este folosit pentru a face referire la alte bucăți de text, în timp ce un limbaj de marcare este o serie de marcaje ce indică serverelor web stilul și structura unui document. Este folosit pentru a structura și a da sens conținutului web, de exemplu definirea paragrafelor, titlurilor sau încorporarea imaginilor și videoclipurilor în pagină. [5]

Documentul HTML este alcătuit dintr-o ierarhie de noduri, fiecare nod poate avea un părinte sau mai mulți copii. Astfel, prin intermediul DOM HTML, JavaScript poate accesa și modifica toate elementele unui document HTML [6]. DOM (Document Object Model) este o interfață de programare a aplicației (API) pentru documentele HTML și XML. Acesta definește structura logică a documentelor și modul în care un document este accesat și manipulat. [7].



Figură 2. Arbore de obiecte HTML DOM [7]

2.2 Realizări similare

Principala sursă de inspirație pentru acest proiect este jocul Memory pe care îl jucam în copilărie cu cărți reale de hârtie. Jocul constă într-un număr par de cărți cu o imagine specifică pe o parte a cărții și un fundal generic pe cealaltă parte. Fiecare imagine apare pe două cărți. Când începe jocul, toate cărțile sunt întoarse cu fața în jos. Apoi, jucătorul întoarce două cărți. Dacă cele două cărți alese au aceeași imagine, acestea rămân cu fața în sus. În caz contrar, cărțile se răstoarnă înapoi cu fața în jos. Scopul jocului este de a împerechea toate cărțile.



Figură 3. Jocul Memory

Implementări similare ce presupun jocul de împerechere a imaginilor pot fi găsite în proiectele [8] și [9]. De asemenea, la momentul actual, există o multitudine de jocuri Memory Card ce sunt disponibile atât ca aplicații ce pot fi descărcate din AppleStore sau PlayStore, cât și ca aplicații web ce sunt accesate dintr-un browser.

Având la bază aceste idei, am decis să implementez un Memory Card Game ce are ca tematică fructele. Astfel, partea de background a fost realizată folosind resursa [10], partea de generare de imagini și dispunerea lor sub formă de grid a fost realizată conform resursei [11].

Pentru ideea implementării logicii jocului, m-am ajutat de resursa [12], ce include atât un README ce detaliază etapele și logica din spate, cât și schelet de cod pentru a face modificări necesare proiectului.

3 Etapa de implementare

Pagina web a fost creată folosind HTML și stilizată cu ajutorul CSS. Astfel, cu ajutorul codului HTML se definesc titlul aplicației, câmpul aferent numărului de vieți rămase și cărțile de joc. Cu ajutorul codului CSS se setează fontul, culoarea, marginile, spațierea, alinierea pentru fiecare element definit în HTML.

Metodele de tip Document folosite în scriptul JavaScript sunt:

- `.querySelectorAll()` ce returnează o listă `NodeList` statică, ce conține toate elementele care se potrivesc cu un selector CSS specificat în document;
- `.querySelector()` ce returnează primul element care se potrivește cu un selector CSS specificat din document (preia clasele elementelor);
- `.createElement()` ce creează un nod element;
- `.classList()` ce returnează numele clasei CSS a unui element sub forma unui vector;
- `.setAttribute()` ce setează sau adaugă o valoare a unui atribut pe un element specificat
- `.addEventListener()` ce atașează documentului un handler de evenimente (Am folosit `EventListener` pentru click-ul efectuat de mouse pe cărțile de joc pentru a ști când o carte este întoarsă);
- `.appendChild()` ce inserează un nod (element) sau repoziționează un nod existent ca ultim copil al unui anumit nod părinte.

Funcțiile de bază sunt implementate în scriptul JavaScript:

- a) **getData()** – generează datele pe care le voi folosi pentru cărțile de joc; funcția returnează un vector de 16 obiecte ce reprezintă imaginile aflate pe partea din față a celor 16 cărți de joc (astfel, în `imgSrc` stochez sursa de unde este preluată imaginea și numele pe care acest obiect îl va avea în cadrul programului)
- b) **randomize()** – generează aleator toate cărțile; în cadrul funcției se apelează funcția `getData()` pentru a genera datele, apoi, cu ajutorul funcțiilor predefinite în JavaScript: `sort()` și `Math.random()`, sunt sortate, respectiv amestecate, returnând un vector de 16 obiecte
- c) **cardGenerator()** – generează codul HTML pentru cărțile de joc;
 - apelează funcția `randomize()` pentru a obține o versiune amestecată a cărților la fiecare refresh al paginii web;
 - generează pentru fiecare carte în parte în secțiunea `section` din documentul HTML un nod **card** de tip `div` tag, două noduri copil: unul **face** de tip `img` tag (pentru fața cărții de joc) și unul **back** de tip `div` (pentru spatele cărții de joc) cu ajutorul metodelor `.createElement()` și `.classList()`;
 - atașează informațiile pe cărți (imaginile pe partea din față a cărții) cu ajutorul metodei `.setAttribute()`;
 - atașează cărțile secțiunii folosind metoda `.appendChild()`;
 - cu ajutorul metodei `.addEventListener()`, la fiecare click pe o carte de joc se va apela metoda `.classList.toggle()` [ce fie va adăuga, fie va elimina un element din clasa `toggleCard`] și funcția **checkCards()**.
- d) **checkCards()** – verifică statusul cărților de joc;
 - cu ajutorul metodei `.classList.add()` se adaugă o clasă numită „flipped” la lista de clase a elementelor, clasă în care se adaugă numele cărților ce sunt întoarse;
 - astfel se verifică dacă numărul de cărți întoarse este egal cu 2, dacă numele primei cărți coincide cu numele celei de a doua cărți, atunci

înseamnă că a fost găsită perechea și cărțile vor rămâne pe tabla de joc cu fața în sus;

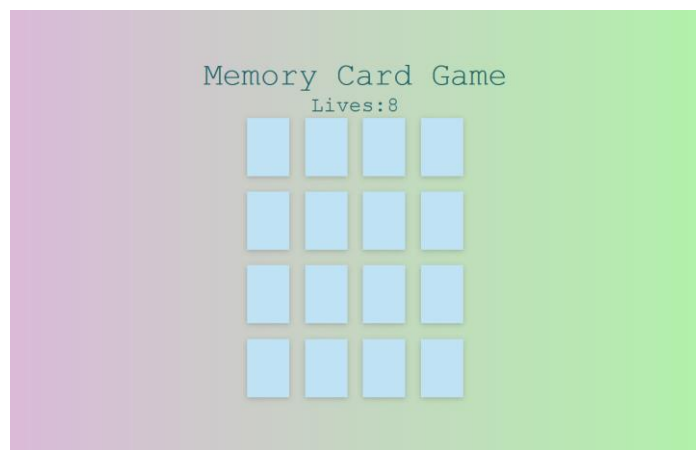
- altfel, înseamnă că nu a fost găsită perechea și cărțile vor fi întoarse din nou cu fața în jos, iar numărul de vieți rămase se va decrementa;
- dacă numărul de elemente din clasa `toggleCard` este egal cu 16, înseamnă ca toate cărțile au fost împerecheate, jucătorul a câștigat și se apelează funcția **restart()**;

e) restart() – resetează întregul joc

- apelează funcția `randomize()` pentru a genera o altă ordine a cărților de joc;
- întoarce toate cărțile cu fața în jos cu ajutorul metodei `.classList.remove()` ce va șterge toate elementele adăugate în clasa `toggleCard`;
- resetează numărul de vieți.

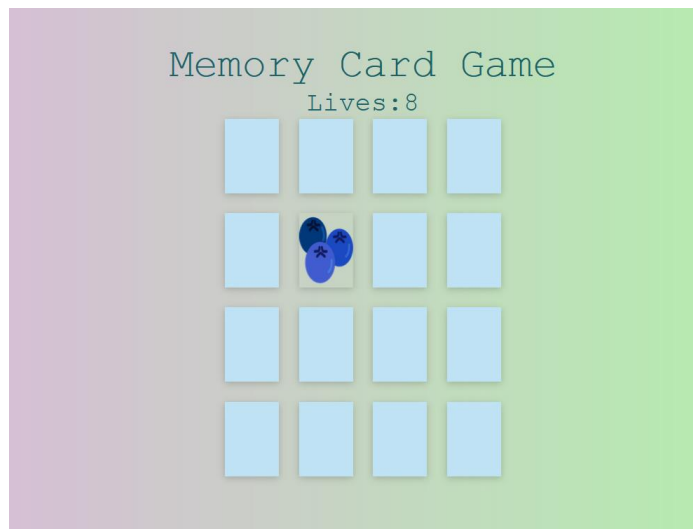
4 Mod de utilizare

La deschidere, jocul începe direct și presupune afișarea numărului de vieți și reprezentarea tablei de joc unde sunt poziționate cu fața în jos încă de la început cele 16 cărți. La apăsarea uneia dintre cărți, aceasta se va apela funcția aferentă din scriptul JavaScript.



Figură 4. Interfața cu utilizatorul a jocului

Efectul vizual generat de apăsarea unei cărți de joc se poate observa în imaginea următoare:

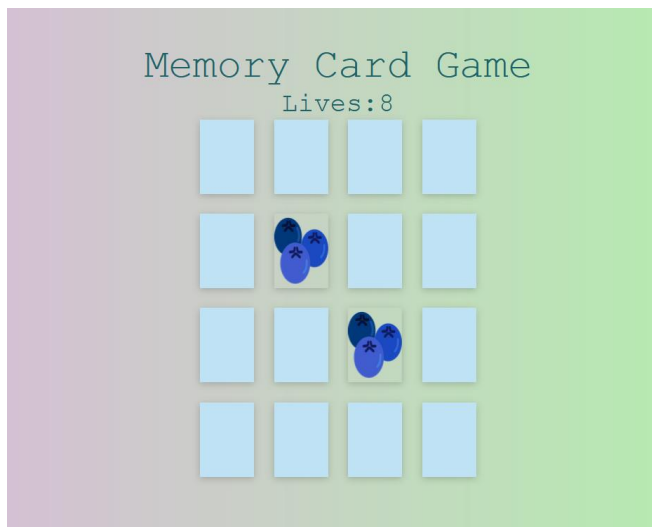


Figură 5. Efectul vizual de întoarcere a cărții apăsate

4.1 Scenariu de utilizare:

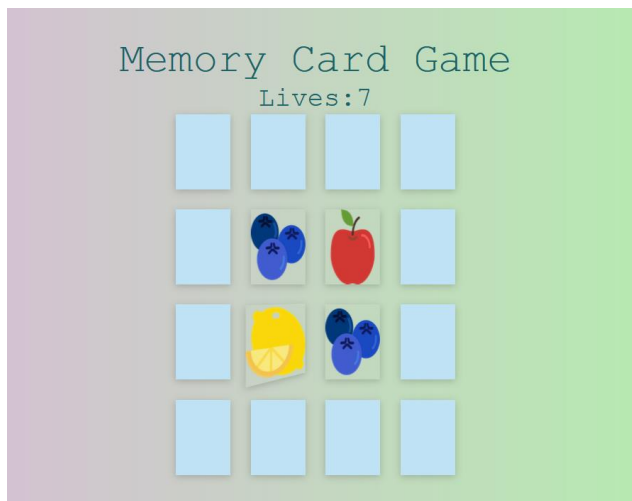
Jucătorul apasă o carte la alegere de pe tabla de joc. Acest lucru va genera întoarcerea cărții alese pe partea din față. Jucătorul apasă o altă carte, încercând să găsească perechea cărții anterior selectate. În urma selecției celei de a doua cărți, există două posibile scenarii:

- 1) Utilizatorul a găsit perechea cărții. Astfel, ambele cărți rămân întoarse cu fața în sus, iar numărul de vieți rămâne neschimbat;



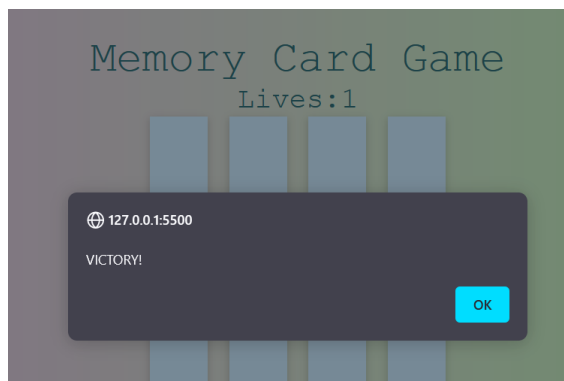
Figură 6. Interfața când numărul de vieți rămâne neschimbat

- 2) Utilizatorul nu a găsit perechea cărții. Astfel, ambele cărți vor fi întoarse înapoi cu fața în jos, iar numărul de vieți se decrementează.



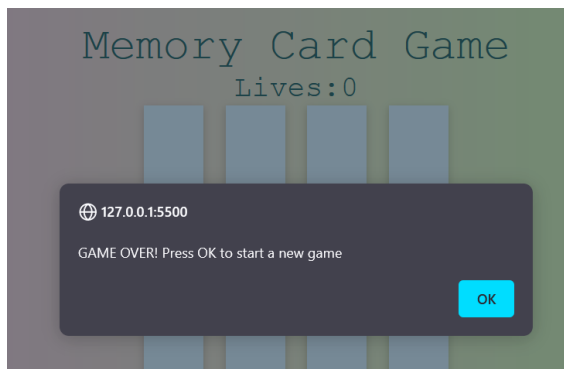
Figură 7. Interfață după ce numărul de vieți s-a modificat

Jucătorul va încerca să împerecheze toate cele 16 cărți până v-a epuiza numărul de vieți alocat. În cazul în care reușește, pe ecran va fi afișat mesajul „VICTORY!”.



Figură 8. Mesaj afișat atunci când jucătorul a câștigat

În cazul în care nu reușește, pe ecran va fi afișat mesajul „GAME OVER! Press OK to start a new game”.



Figură 9. Mesaj afișat atunci când jucătorul a pierdut

5 Concluzii

Obiectivele inițiale au fost îndeplinite și au fost realizate într-un mod optim. Am reușit să realizez un joc destinat copiilor cu scopul antrenării memoriei ce prezintă o interfață prietenoasă cu utilizatorul unde acesta poate vizualiza tabla de joc cu cărți, cât și numărul de vieți rămase. La apăsarea unei cărți de joc, o animație 3D va apărea în timp ce cartea de joc se întoarce cu fața în sus. Actualizarea numărului de vieți după fiecare mutare se realizează astfel: când perechea cărții este găsită, numărul de vieți rămâne neschimbat; când perechea cărții nu este găsită, numărul de vieți se decrementează cu 1. De fiecare dată când jucătorul pierde sau câștigă, un mesaj va fi afișat pe ecran și jocul va fi resetat, cărțile amestecându-se la fiecare resetare.

În urma realizării acestui proiect, am deprins noțiuni de bază ale tehnologiilor web standard HTML, CSS, JavaScript. Pentru a corela crearea jocului cu noțiunile asimilate în cadrul materiei Aplicații Multimedia, am adăugat imagini pe fiecare carte de joc în format PNG, deoarece, spre deosebire de JPEG, calitatea imaginii nu este degradată. PNG oferă o compresie fără pierderi (lossless) și poate gestiona elemente grafice cu fundal transparent sau semi-transparent.

Aplicația creată este una simplă, ce poate suporta modificări și îmbunătățiri prin adăugarea unor butoane prin care utilizatorul are posibilitatea de a selecta gradul de dificultate al jocului cum ar fi:

- a) Nivelul Easy – unde pe tabla de joc vor apărea doar 8 cărți de joc pe care utilizatorul trebuie să le împerecheze;
- b) Nivelul Medium – ce va fi reprezentat de stadiul actual al jocului, adică 16 cărți pe tabla de joc;
- c) Nivelul Hard – unde pe tabla de joc vor apărea 20 de cărți de joc pe care utilizatorul trebuie să le împerecheze.

6 Bibliografie

- [1] <https://www.secom.ro/blogs/ghidul-secom/memoria-ce-este-care-e-rolul-ei-si-cum-o-putem-imbunatati#2>
- [2] <https://www.youtube.com/watch?v=fWzp5IZ5CEA>
- [3] https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- [4] https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS
- [5] <https://www.hostinger.com/tutorials/what-is-html>
- [6] https://www.w3schools.com/js/js_htmlDOM.asp
- [7] <https://www.w3.org/TR/REC-DOM-Level-1/introduction.html>
- [8] <https://www.khanacademy.org/computing/computer-programming/programming-games-visualizations/memory-game/a/playing-the-game>
- [9] <https://www.c-sharpcorner.com/article/memory-game/>
- [10] https://www.w3schools.com/css/css_background.asp
- [11] https://www.w3schools.com/css/css_grid.asp
- [12] <https://github.com/ironhack-labs/lab-javascript-memory-game#readme>