VELEUČILIŠTE BALTAZAR ZAPREŠIĆ

Matija Sović

Projektni zadatak

Web programiranje

Sisak, 2024

VELEUČILIŠTE BALTAZAR ZAPREŠIĆ

Matija Sović

Izvanredni student

ESI:0234066308

Smijer: Informacijske tehnologije

Projektni zadatak

IGRA PAMĆENJA

Kolegij:

Web programiranje

Sisak, travanj, 2024.

SADRŽAJ

[Uvod 4](#_Toc163766367)

[HTML 4](#_Toc163766368)

[CSS 6](#_Toc163766369)

[JavaScript 9](#_Toc163766370)

# Uvod

U današnjem digitalnom dobu, interaktivne web aplikacije igraju ključnu ulogu u edukaciji, zabavi i razvoju kognitivnih vještina. Projekt "Igra pamćenja" predstavlja jednostavnu, ali izazovnu web aplikaciju razvijenu koristeći tehnologije poput HTML-a, CSS-a, i JavaScripta, s ciljem ne samo pružanja zabave korisnicima, već i poticanja razvoja memorije i pažnje.

Igra pamćenja poznata i kao igra "Pronađi par", klasična je igra koja zahtijeva od igrača da pronalaze parove identičnih kartica raspoređenih licem prema dolje. Cilj igre je pronaći sve parove koristeći što manje pokušaja, što potiče igrače na razvijanje strategija pamćenja i vizualne percepcije. Kroz interaktivno sučelje, ovaj projekt ne samo da pruža izazov i zabavu, već i služi kao alat za poboljšanje kratkotrajne memorije i koncentracije.

Razvoj ove aplikacije obuhvaća primjenu osnovnih, ali bitnih koncepta web razvoja, uključujući manipulaciju DOM-a (Document Object Model), upotrebu događaja (events) za interakciju s korisnikom, te primjenu CSS-a za stiliziranje i animacije koje igru čine vizualno privlačnijom. Osim toga, projekt demonstrira kako jednostavan gameplay može biti implementiran kroz čist i efikasan kôd, čime se potiče razumijevanje temeljnih principa programiranja i logičkog razmišljanja.

Ovaj seminarski rad detaljno će objasniti svaki aspekt razvoja „Igre pamćenja“, od ideje do realizacije, uključujući razmatranje izbora tehnologija, dizajn sučelja, implementaciju logike igre, pa sve do testiranja funkcionalnosti i korisničkog iskustva. Cilj je pružiti čitatelju uvid u proces razvoja web aplikacije, istaknuvši kako se teoretsko znanje primjenjuje u praksi na konkretnom projektu koji spaja zabavu i edukativnu vrijednost.

# HTML

Meta podatci

<meta charset="UTF-8">

Ova meta oznaka specificira set karaktera koji se koristi na web stranici. UTF-8 je široko prihvaćeni univerzalni karakter set koji podržava sve jezike i znakove. Korištenjem UTF-8 osiguravate da će svi tekstovi na vašoj stranici biti pravilno prikazani bez obzira na jezik.

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Meta oznaka viewporta ključna je za pravilno prikazivanje web stranica na mobilnim uređajima. Ova oznaka govori pregledniku da skalira širinu stranice prema širini uređaja na kojem se stranica prikazuje, s početnim mjerilom (zoom) postavljenim na 1.0, što znači da nema inicijalnog povećanja.

<meta http-equiv="X-UA-Compatible" content="IE=edge">

Ova meta oznaka osigurava da Internet Explorer koristi najnoviju dostupnu verziju render engine-a. Ovo može pomoći u sprječavanju problema s kompatibilnošću u starijim verzijama preglednika.

<meta name="description" content="Igra pamcenja - Web aplikacija za poboljšanje memorije i koncentracije.">

Opisna meta oznaka pruža kratak sažetak sadržaja stranice. Ovo je važno za SEO jer pretraživači često koriste ovaj tekst kao snippet u rezultatima pretraživanja. Dobar opis može povećati klikove na vašu stranicu.

<meta name="keywords" content="Memory, igra pamcenja, edukativne igre, igre memorije, web igra">

Iako meta oznake ključnih riječi više nisu relevantne za SEO u smislu rangiranja, neki razvijatelji ih još uvijek koriste za dokumentiranje ključnih riječi relevantnih za sadržaj stranice.

<meta name="author" content="Matija Sović">

Oznaka “author” navodi autora web stranice.

<meta name="robots" content="index, follow">

Ova meta oznaka daje upute pretraživačkim robotima da indeksiraju stranicu i prate sve poveznice na njoj. To je osnovni način da se omogući preglednicima da uključe vašu stranicu u njihove indekse i povežu je s drugim stranicama.

<meta property="og:title" content="Igra pamcenja">

og:title je Open Graph protokol koji koriste Facebook i druge društvene platforme za naslov sadržaja koji se dijeli. Ova oznaka definira naslov sadržaja za društvene mreže.

<meta property="og:description" content="Interaktivna web aplikacija koja korisnicima omogućuje da vježbaju svoje sposobnosti pamćenja kroz igru.">

Pruža opis koji se koristi kada se sadržaj dijeli na društvenim mrežama. Dobar opis može povećati angažman i interakcije na društvenim mrežama.

<title>Igra pamćenja</title>

Ovo je HTML tag koji definira naslov prozora ili kartice preglednika. Naslov je vrlo važan za SEO, jer daje kontekst o sadržaju stranice i prvi je koji korisnici vide kad pregledavaju rezultate pretraživanja.

<link rel="stylesheet" href="styles.css">

Ovo povezuje vanjsku CSS datoteku sa vašom HTML stranicom, što je ključno za stiliziranje izgleda vaše web stranice.

Body

<button id="reset-button">Resetiraj Kartice</button>

Ovo je HTML element koji definira gumb. Kada korisnik klikne na ovaj gumb, pokreće se definirana funkcija koja resetira igru na početni status. Ovo uključuje vraćanje svih kartica na početni položaj (licem prema dolje) i moguće resetiranje brojača pokušaja.

<section class="memory-game">

Ovo je kontejnerski element koji sadrži sve kartice igre. Iskoristio sam ga za grupiranje srodnih sadržaja unutar web stranice, što u ovom slučaju uključuje interaktivne kartice.

<div class="memory-card" data-framework="b">

            <img src="images/lopta.png" alt="Image 2" class="front-face">

            <img src="images/cover.png" alt="Cover" class="back-face">

</div>

*class="memory-card"*: Ova klasa koristi se za primjenu stilova i funkcionalnosti specifičnih za kartice.

*data-framework="b"*: Ovaj data atribut koristi se za povezivanje dviju kartica koje čine par.

*src="images/lopta.png"*: Atribut src specificira putanju do slike koja će biti prikazana.

*alt="Image 2"*: Atribut alt pruža alternativni tekst za sliku, koji se koristi za pristupačnost i u slučajevima kada slika ne može biti prikazana.

*class="front-face"*: Ova klasa koristi se za primjenu specifičnih stilova na lice kartice i za manipulaciju elementima u JavaScriptu.

*src="images/cover.png"*: Putanja do slike koja će biti prikazana na poleđini.

*alt="Cover"*: Alternativni tekst koji opisuje sliku, koristi se za pristupačnost.

*class="back-face"*: Klasa za primjenu stilova na poleđinu kartice

# CSS

Cascading Style Sheets (CSS) ključni su za vizualno formatiranje web aplikacije "Igra pamćenja". Ovaj segment dokumenta objašnjava kako CSS unapređuje interaktivne elemente igre kao što su kartice i gumb za resetiranje, osiguravajući estetski privlačno i intuitivno korisničko sučelje.

Ciljevi CSS-a u Projektu

CSS u "Igra pamćenja" ima višestruke funkcije:

* Stilizacija Kartica: Primjenjuje dimenzije, boje, i efekte okretanja, što je vitalno za igračko iskustvo.
* Responsive Dizajn: Osigurava da igra izgleda primjereno na svim uređajima, koristeći adaptivne layoute.
* Interaktivnost: Poboljšava korisničku interakciju kroz pseudo-klase poput :`hover` i :`active`.

.memory-game {

    display: grid;

    grid-template-columns: repeat(3, 1fr);

    gap: 10px;

    margin: 40px;

}

Ova CSS pravila definiraju grid sustav za *.memory-game* kontejner s tri jednako široke kolone, koristeći fleksibilne jedinice (fr) za ravnomjerno raspoređivanje prostora. Kontejner ima razmak od 10 piksela između kartica i uniformnu marginu od 40 piksela sa svih strana za estetski odmak od rubova stranice.

.memory-card {

    width: 180px;

    height: 180px;

    background: #f0f0f0;

    position: relative;

    cursor: pointer;

    transform: rotateY(0deg);

    transition: transform 0.5s;

    transform-style: preserve-3d;

}

Pravila CSS-a za *.memory-card* definiraju svaku karticu s fiksnom širinom i visinom od 180 piksela, te postavljaju sivu pozadinu. Stil *position: relative* i *transform-style: preserve-3d* omogućuju trodimenzionalne transformacije, dok *cursor: pointer* ukazuje da je element interaktivan. Tranzicija za rotaciju kartice (rotateY) traje 0.5 sekundi, omogućujući glatki vizualni efekt okretanja.

.memory-card.flip {

    transform: rotateY(180deg);

}

*.memory-card.flip* primjenjuje rotaciju od 180 stupnjeva oko vertikalne osi (Y) na kartice, omogućavajući im da se okrenu i pokažu drugu stranu, što je ključni vizualni efekt.

.front-face, .back-face {

    width: 100%;

    height: 100%;

    position: absolute;

    backface-visibility: hidden;

    transition: transform 0.5s, box-shadow 0.5s;

}

*.front-face* i .*back-face* postavljaju ove elemente da zauzimaju cijelu površinu svoje roditeljske kartice (100% širine i visine) i pozicioniraju ih apsolutno unutar kartice. Atribut *backface-visibility: hidden* sprječava prikaz strane kartice koja je okrenuta od korisnika, dok *transition* omogućava glatku tranziciju rotacije i promjene sjene u trajanju od 0.5 sekundi.

.front-face {

    background: url('images/cover.png') center/cover no-repeat;

    transform: rotateY(180deg);

}

Ova strana kartice inicijalno je rotirana za 180 stupnjeva duž Y osi, što znači da je okrenuta prema dolje kada je igra započeta.

.back-face {

    background: url('images/kamion.png') center/cover no-repeat;

    transform: rotateY(0deg);

}

Transformacijski atribut *transform: rotateY(0deg)* postavlja ovu stranu kartice da bude okrenuta prema gledatelju na početku igre.

#reset-button {

    margin: 20px auto;

    padding: 10px 20px;

    font-size: 16px;

    cursor: pointer;

    background-color: #4CAF50;

    color: white;

    border: none;

    border-radius: 5px;

    display: block;

}

*#reset-button* stilizira gumb za resetiranje u zelenom bojom s bijelim tekstom, centralno pozicioniran s zaobljenim rubovima i bez vanjskih granica. Gumb je postavljen da bude blok element, s jasnim vizualnim indikatorom interaktivnosti preko *cursor: pointer*.

#reset-button:hover {

    background-color: #427044;

}

*#reset-button:hover* mijenja boju pozadine gumba za resetiranje u tamniju nijansu zelene kada korisnik pređe mišem preko njega, poboljšavajući vizualni odziv i interaktivnost.

#win-message {

    color: #ff0000;

    font-size: 34px;

    text-align: center;

    margin: 20px 0;

    display: none;

}

*#win-message* stilizira završnu poruku s crvenim tekstom veličine 34px, centrirano i početno skriveno, koja se prikazuje samo pri završetku igre.

.footer {

    width: 100%;

    position: absolute;

    bottom: 0;

    background-color: #f2f2f2;

    padding: 10px 0;

    text-align: center;

}

*.footer* stilizira podnožje stranice tako da zauzima cijelu širinu, pozicionirano je na dnu stranice, s sivom pozadinom, centriranim tekstom, i blagim unutarnjim razmakom.

.footer p {

    display: inline;

    padding: 0 15px;

}

*.footer p* postavlja paragrafe unutar podnožja da se prikazuju u liniji (inline), s horizontalnim razmakom od 15 piksela.

# JavaScript

JavaScript je ključan za dodavanje interaktivnosti i dinamičnosti projektu "Igra pamćenja", omogućavajući ključne funkcionalnosti koje su potrebne za vođenje igre. Skripta reagira na korisnikove klikove na kartice, aktivirajući CSS transformacije koje okreću kartice i otkrivaju njihove slike. Kada igrač okrene dvije kartice, JavaScript provjerava podudaraju li se slike. Ako se slike poklapaju, kartice ostaju okrenute kao pronađeni par; ako se ne poklapaju, kartice se vraćaju u početni položaj. Osim toga, skripta prati napredak igre brojeći pronađene parove i završava igru kada su svi parovi uspješno pronađeni, pri čemu se prikazuje poruka pobjede. Ova logika ne samo da čini igru zabavnom i angažiranom, već i potiče korisnike na vježbanje vizualne memorije i koncentracije.

const cards = [...document.querySelectorAll('.memory-card')];

Ova linija koristi *document.querySelectorAll* da dohvati sve elemente s klasom *.memory-card* (sve kartice u igri). Spread operator (...) se koristi za pretvaranje NodeList objekta u pravi JavaScript niz, omogućujući lakše manipulacije.

const resetButton = document.getElementById('reset-button');

const winMessage = document.getElementById('win-message');

Ove linije selektiraju gumb za resetiranje i element za poruku.

console.log("Ukupan broj kartica:", cards.length);

console.log("Broj parova:", cards.length / 2);

Ove linije koda logiraju ukupan broj kartica (*cards.length*) i broj parova (*cards.length / 2*) u konzolu, što omogućava jednostavno praćenje i debugiranje stanja igre "Igra pamćenja" tijekom razvoja.

cards.forEach(card => card.addEventListener('click', flipCard));

resetButton.addEventListener('click', resetGame);

Ove linije JavaScript koda dodaju event listenere na kartice i gumb za resetiranje, aktivirajući funkcije *flipCard* i *resetGame* kada korisnik klikne na njih, omogućujući interakciju i kontrolu tijeka igre.

function flipCard() {

        if (lockBoard) return;

        if (this === firstCard) return;

        this.classList.add('flip');

        if (!hasFlippedCard) {

            hasFlippedCard = true;

            firstCard = this;

        } else {

            secondCard = this;

            checkForMatch();

        }

    }

Funkcija *flipCard* upravlja procesom okretanja kartica .Ako je ploča zaključana (lockBoard) ili igrač pokušava ponovno okrenuti već okrenutu karticu (firstCard), funkcija se prekida. Klikom na karticu, dodaje se *klasa flip* za vizualno prikazivanje njene prednje strane. Ako je to prvi klik, kartica se postavlja kao *firstCard*. Ako je drugi klik, postavlja se kao *secondCard* i izvršava se funkcija *checkForMatch* za provjeru da li kartice čine par. Ova logika osigurava glatku interakciju i sprječava greške tijekom igre.

function checkForMatch() {

   let isMatch = firstCard.dataset.framework === secondCard.dataset.framework;

   if (isMatch) {

        disableCards();

        matchCount++;

        console.log("Pronađen par, trenutni broj pronađenih parova:",matchCount);

            if (matchCount === (cards.length / 2)) {

                console.log("Svi parovi su pronađeni!");

                winMessage.style.display = 'block';

            }

        } else {

            unflipCards();

        }

    }

Nakon što igrač okrene dvije kartice, ova funkcija uspoređuje njihove *data-framework* atribute da vidi jesu li isti, što znači da je par pronađen. Ako se atributi podudaraju, funkcija *disableCards* se poziva za trajno okretanje kartica licem prema gore i inkrementira se *matchCount*, brojač pronađenih parova. Ako su svi parovi pronađeni (matchCount jednak polovici ukupnog broja kartica), ispisuje se poruka "Svi parovi su pronađeni!" u konzolu, a korisniku se prikazuje poruka "Čestitam". U suprotnom, ako kartice nisu par, poziva se funkcija *unflipCards* koja vraća kartice u početni položaj.

function disableCards() {

        firstCard.removeEventListener('click', flipCard);

        secondCard.removeEventListener('click', flipCard);

        resetBoard();

    }

Funkcija *disableCards* ima ulogu da trajno deaktivira interakciju s karticama koje čine pronađeni par. Kada dvije kartice uspješno formiraju par, ova funkcija uklanja event listenere za klikove s tih kartica, čime se sprječava njihovo daljnje okretanje ili interakcija. Nakon toga, poziva funkciju resetBoard koja priprema igru za sljedeći potez, resetirajući potrebne varijable i stanja kako bi igra mogla nastaviti glatko. Ovaj pristup osigurava da se već uparene kartice ne mogu ponovno koristiti u igri.

function unflipCards() {

        lockBoard = true;

        setTimeout(() => {

            firstCard.classList.remove('flip');

            secondCard.classList.remove('flip');

            resetBoard();

        }, 1500);

    }

Funkcija *unflipCards* se koristi za upravljanje situacijama kada dvije okrenute kartice ne čine par. Ova funkcija privremeno zaključava ploču (lockBoard = true) kako bi spriječila daljnje klikove i dala igračima vremena da zapamte lokacije kartica. Pomoću *setTimeou*t, funkcija čeka 1.5 sekunde prije nego što ukloni klasu flip s obiju kartica, čime ih vizualno vraća u početni položaj licem prema dolje. Nakon tog intervala, funkcija *resetBoard* se poziva kako bi resetirala ploču i omogućila nastavak igre, osiguravajući da se igra može nastaviti bez prekida i grešaka.

function resetBoard() {

        [hasFlippedCard, lockBoard] = [false, false];

        [firstCard, secondCard] = [null, null];

    }

Funkcija *resetBoard* je dizajnirana da re-inicijalizira ključne varijable u igri, pripremajući igru za novi potez ili za resetiranje nakon pogrešno uparenih kartica. Funkcija postavlja *hasFlippedCard* i *lockBoard* na false, omogućavajući novu seriju klikova i osiguravajući da ploča nije zaključana. Također, resetira reference *firstCard* i s*econdCard* na null, uklanjajući veze s trenutno odabranim karticama.

function resetGame() {

        cards.forEach(card => {

            card.classList.remove('flip');

            card.addEventListener('click', flipCard);

        });

        winMessage.style.display = 'none';

        matchCount = 0;

        console.log("Igra resetirana. Brojač parova resetiran.");

        resetBoard();

    }

Funkcija *resetGame* vraća sve kartice licem prema dolje i priprema igru za novi početak. Ona uklanja klasu *flip* s kartica, ponovno dodaje click listenere, skriva poruku pobjede, resetira brojač parova na nulu, i logira ove radnje u konzolu za praćenje. Dodatno, funkcija *resetBoard* je pozvana da osvježi sve pomoćne varijable, osiguravajući čist i uredan restart igre.