

LUCRARE DE ATESTAT LA INFORMATICĂ

Joc de tenis în HTML, CSS și JavaScript

Elev: Dan Teodora-Alexandra

Clasa: 12A

Profesor îndrumător: Rad Lenuța

Câmpia Turzii, 2018

Conținut

1. Argument
2. Generalități despre HTML, CSS și JavaScript
3. Prezentarea generală a temei alese
4. Utilizarea aplicației
5. Realizarea aplicației
 - 5.1. Descriere cod utilizat în HTML
 - 5.2. Descriere cod utilizat în CSS
 - 5.3. Descriere cod utilizat în JavaScript
6. Extinderi posibile ale aplicației
7. Resurse Hard și Soft necesare
8. Bibliografie

1. Argument

Este bine cunoscut faptul că jocurile pe calculator au devenit foarte populare atât în rândul copiilor, cât și al adolescenților sau chiar al adulților.

Opinia mea este că, privind evoluția tehnologiei din ultimele decenii, jocurile pe calculator vor fi mereu îndrăgite de oamenii de pretutindeni.

În primul rând, am ales să realizez un joc mai ușor pentru a înțelege bazele programării în JavaScript și funcțiile de baza pe care fiecare programator le folosește. În documentația de față, la secțiunea Realizarea aplicației este descris codul mai detaliat, pentru a înțelege structura, funcțiile utilizate și rolul fiecăreia.

În al doilea rând, am ales să realizez jocul utilizând limbajul JavaScript deoarece conform statisticilor, este printre cele mai utilizate limbaje în momentul de față. Grafica pe care am ales să o utilizez este minimalistă, grafică pe care se bazează majoritatea jocurilor de acest tip. Codul `<canvas>` din HTML a fost folosit pentru fundalul jocului, unde dimensiunile au fost alese corespunzător.

În concluzie, privind amploarea pe care au luat-o jocurile pe calculator, limbajul de programare ales JavaScript și inteligența artificială a calculatorului, sunt de părere că acest mini-joc ar putea reprezenta o bază pentru viitoarele creații, având majoritatea funcțiilor necesare pentru implementarea codului oricărui joc de acest tip.

2. Generalități despre HTML, CSS și JavaScript

HTML

HyperText Markup Language (HTML) este un limbaj de marcare utilizat pentru crearea paginilor web ce pot fi afișate într-un browser (sau navigator). Scopul HTML este mai degrabă prezentarea informațiilor – paragrafe, fonturi, tabele ș.a.m.d. – decât descrierea semanticii documentului.

Specificățiile HTML sunt dictate de World Wide Web Consortium (W3C).

HTML este o formă de marcare orientată către prezentarea documentelor text pe o singură pagină, utilizând un software de redare specializat, numit agent utilizator HTML, cel mai bun exemplu de astfel de software fiind browserul web. HTML furnizează mijloacele prin care conținutul unui document poate fi adnotat cu diverse tipuri de metadate și indicații de redare. Indicațiile de redare pot varia de la decorațiuni minore ale textului, cum ar fi specificarea faptului că un anumit cuvânt trebuie subliniat sau că o imagine trebuie introdusă, până la scripturi sofisticate, hărți de imagini și formulare. Metadatele pot include informații despre titlul și autorul documentului, informații structurale despre cum este împărțit documentul în diferite segmente, paragrafe, liste, titluri etc. și informații cruciale care permit ca documentul să poată fi legat de alte documente pentru a forma astfel hyperlink-uri (sau web-ul).

CSS

CSS (Cascading Style Sheets) este un standard pentru formatarea elementelor unui document HTML. Stilurile se pot atașa elementelor HTML prin intermediul unor fișiere externe sau în cadrul documentului, prin elementul `<style>` și/sau atributul `style`. CSS se poate utiliza și pentru formatarea elementelor XHTML, XML și SVG.

JavaScript

JavaScript (JS) este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul JavaScript din aceste pagini fiind rulat de către browser. Limbajul este binecunoscut pentru folosirea sa în construirea siturilor web, dar este folosit și pentru accesul la obiecte încastate (embedded objects) în alte aplicații. A fost dezvoltat inițial de către Brendan Eich de la Netscape Communications Corporation sub numele de Mocha, apoi LiveScript, și denumit în final JavaScript.

În ciuda numelui și a unor similarități în sintaxă, între JavaScript și limbajul Java nu există nicio legătură. Ca și Java, JavaScript are o sintaxă apropiată de cea a limbajului C, dar are mai multe în comun cu limbajul Self decât cu Java.





3. Prezentarea generală a temei alese

Jocul de tenis este un joc arcade clasic, considerat printre cele mai usoare jocuri din punct de vedere al înțelegerii codului. Cei mai mulți începători aleg să învețe principiile fundamentale ale programării jocurilor din aceste jocuri clasice.

Epoca de aur a jocurilor arcade a durat de la sfârșitul anilor 1970 până la mijlocul anilor 1990. În timp ce jocurile arcade erau încă relativ populare la sfârșitul anilor 1990, această metodă de divertisment a intrat într-un declin continuu de popularitate în Occident, odată cu trecerea consolelor de jocuri video pentru acasă de la grafica 2D la grafica 3D.

4. Utilizarea aplicației

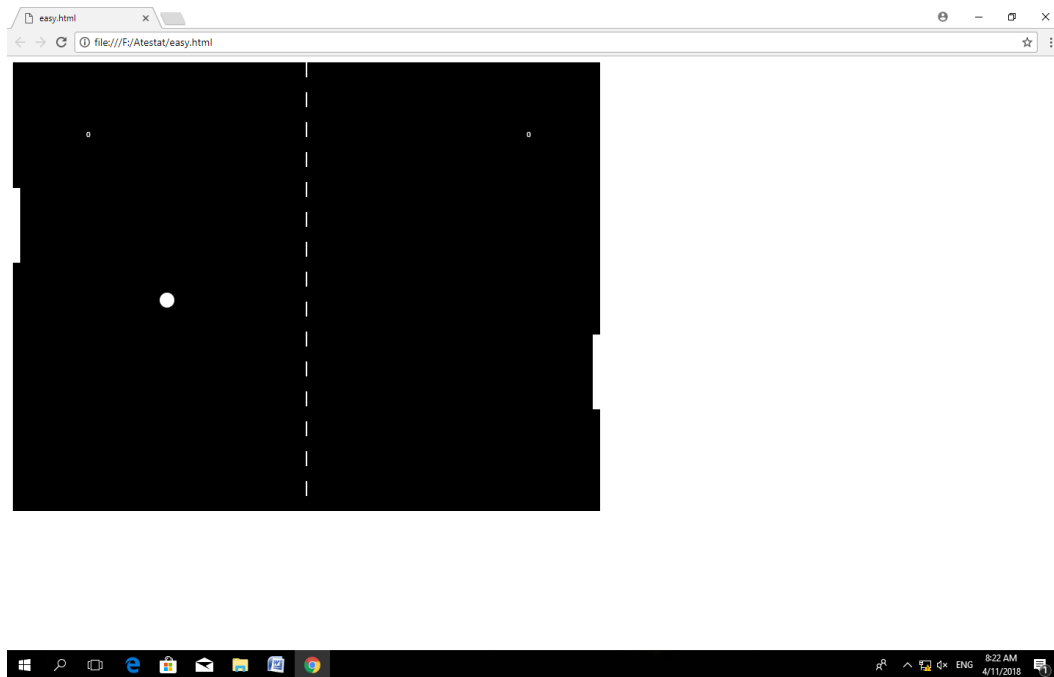
Aplicația este ușor de utilizat. În folder, există 5 fișiere HTML: index, easy, medium, hard.

Name	Date modified	Type	Size
 easy	3/30/2018 4:51 PM	Chrome HTML Do...	6 KB
 hard	3/30/2018 4:51 PM	Chrome HTML Do...	6 KB
 index	3/30/2018 3:57 PM	Chrome HTML Do...	2 KB
 medium	3/30/2018 4:51 PM	Chrome HTML Do...	6 KB

Pentru a intra în pagina principală, deschdeți cu dublu-click fișierul HTML denumit index.html. În pagina browser-ului utilizat va apărea următoarea pagină:



Aplicația conține 3 nivele de dificultate, iar fiecare dreptunghi, colorat după culorile unui semafor (verde- ușor, galben- mediu, roșu- greu), conține o ancoră către jocul propriu-zis.



Restul paginilor (easy.html, medium.html, hard.html) sunt asemănătoare, singura diferență fiind viteza mingii și direcția de deplasare. Codul în JavaScript responsabil pentru aceasta este următorul:

```
var deltaY = ballY - (paddle1Y+PADDLE_HEIGHT/2);  
    ballSpeedY = deltaY * 0.25;  
var deltaY = ballY - (paddle2Y+PADDLE_HEIGHT/2);  
    ballSpeedY = deltaY * 0.25;
```

Mai exact, într-o variabilă deltaY se atribuie valoarea obținută din scăderea dintre coordonata Y a mingii și suma dintre variabila paddle1Y(250) și PADDLE_HEIGHT/2 (100/2). Apoi, viteza mingii pe verticală se actualizează, fiind produsul dintre această variabilă deltaY și un număr mai mic decât 1, în funcție de dificultate (easy- 0.25, medium- 0.45, hard- 0.65). Aceste valori reprezintă unghiul de deplasare al mingii, calculat în pixeli.

5. Realizarea aplicației

5. 1. Descriere cod utilizat în HTML

În pagina principală (index.html) am utilizat un cod minimalist în HTML, utilizat pentru realizarea unui meniu, care conține legături pentru paginile propriu-zise ale jocului, alese în funcție de dificultate.

COD UTILIZAT:

```
<html>
<head>
<title>Atestat</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>

<h1><p align="center">Alege&#355i nivelul de dificultate al jocului: </p></h1>
<a href="easy.html"><div class="easy"><p id="easy">EASY</p></div></a>
<a href="medium.html"><div class="medium"><p id="medium">MEDIUM</p></div></a>
<a href="hard.html"><div class="hard"><p id="hard">HARD</p></div></a>

</body>
</html>
```

În paginile care conțin nivelurile jocului de tenis, codul HTML conține eticheta `<canvas></canvas>`, alături de atributele aferente (lungime de 800 pixeli și lățime de 600 pixeli). Acest cod reprezintă „pânza” pe care se desfășoară întregul joc, vizual reprezentând chiar fundalul negru al paginii. Atributele nu sunt alese întâmplător, acestea fiind valorile cel mai des utilizate la crearea jocurilor de acest tip.

COD UTILIZAT:

```
<html>
<canvas id="gameCanvas" height="600" width="800"></canvas>
</html>
```

5. 2. Descriere cod utilizat în CSS

În CSS, am realizat design-ul pentru pagina principală (index.html), unde am ales un fundal gri, care se potrivește cu dreptunghiurile colorate. Fontul utilizat este Lucinda Console, font pe care l-am considerat potrivit pentru această temă.

În secvențele din cod se poate observa design-ul ales pentru fiecare dreptunghi.

SECVENȚE REPREZENTATIVE DIN CODUL UTILIZAT:

```
<style>
```

```
body {background-color: #C7CEC5;}
```

```
body, html {
```

```
overflow: hidden;
```

```
font-family: "Lucida Console", monospace;
```

```
}
```

```
.easy {
```

```
position: absolute;
```

```
height: 80%;
```

```
width: 32%;
```

```
display: inline-block;
```

```
margin: 0 0 0 0;
```

```
background: green;
```

```
border: 3px solid darkgreen;
```

```
}
```

```
#easy {
```

```
margin-top: 60%;
```

```
margin-bottom: 50%;
```

```
margin-left: 40%;
```

```
margin-right: 50%;
```

```
-webkit-transform: rotate(330deg);
```

```
-moz-transform: rotate(330deg);
```

```
-o-transform: rotate(330deg);
```

```
writing-mode: lr-tb;
```

```
font-size: 40px;
```


```
}
```

```
</style>
```


5.3. Descriere cod utilizat în JavaScript

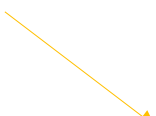
În prima parte a codului am declarat toate variabilele necesare, anume: fundalul jocului, mingea, paletele, scorul și viteza mingii. Cele mai importante funcții utilizate sunt descrise detaliat în cele ce urmează:

```
function calculateMousePos(evt) {  
  var rect = canvas.getBoundingClientRect();  
  var root = document.documentElement;  
  var mouseX = evt.clientX - rect.left - root.scrollLeft;  
  var mouseY = evt.clientY - rect.top - root.scrollTop;  
  return {  
    x: mouseX,  
    y: mouseY  
  };  
}
```



Această funcție are rolul de a afla coordonatele mouse-ului utilizatorului în timp real.

```
function handleClick(evt) {  
  if(showingWinScreen) {  
    player1Score = 0;  
    player2Score = 0;  
    showingWinScreen = false;  
  }  
}
```



Această funcție, care conține o structură decizională, are rolul de a afișa scorul. În momentul în care unul dintre jucători ajunge la 5 puncte (constanta declarată în prima parte a codului), funcția resetează jocul și atribuie fiecărui jucător un scor de 0 puncte.

```
window.onload = function() {  
  canvas = document.getElementById('gameCanvas');  
  canvasContext = canvas.getContext('2d');  
  
  var framesPerSecond = 30;  
  setInterval(function() {  
    moveEverything();  
    drawEverything();  
  }, 1000/framesPerSecond);
```

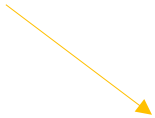
```
canvas.addEventListener('mousedown',handleMouseClicked)
```

```
canvas.addEventListener('mousemove',  
function(evt) {
```

```

var mousePos = calculateMousePos(evt);
    paddle1Y = mousePos.y-(PADDLE_HEIGHT/2);
});
}

```

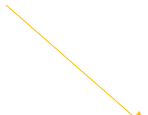


În momentul în care browserul s-a încărcat, funcția aceasta va face legătura dintre elementul canvas din HTML și funcțiile deja existente în codul din JavaScript.

```

function ballReset() {
if(player1Score >= WINNING_SCORE || player2Score >= WINNING_SCORE) {
    showingWinScreen = true;
}
    ballSpeedX = -ballSpeedX;
    ballX = canvas.width/2;
    ballY = canvas.height/2;
}

```

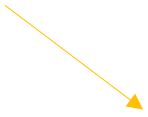


Când unul dintre jucători ajunge la constanta scorului, se va afișa textul corespunzător („Ai câștigat” sau „Ai pierdut”), iar pentru a ridica nivelul de dificultate al jocului, am ales ca viteza mingii să se schimbe de fiecare dată când unul dintre jucători „marchează”.

```

function computerMovement() {
    var paddle2YCenter = paddle2Y + (PADDLE_HEIGHT/2);
    if (paddle2YCenter < ballY-35) {
        paddle2Y += 6;
    } else if (paddle2YCenter > ballY+35){
        paddle2Y -= 6;
    }
}

```



Această funcție coordonează mișcările computerului și totodată reprezintă un cod de bază care este utilizat pentru inteligența artificială. Paleta computerului va merge în sus sau în jos câte 6 pixeli, în funcție de coordonata Y a mingii (ballY).

```

function moveEverything() {
if (showingWinScreen) {
    return;
}
    computerMovement();

    ballX += ballSpeedX;
    ballY += ballSpeedY;
    if (ballX < 0) {
if(ballY > paddle1Y && ballY < paddle1Y+PADDLE_HEIGHT) {
        ballSpeedX = -ballSpeedX;
var deltaY = ballY - (paddle1Y+PADDLE_HEIGHT/2);

```

```

        ballSpeedY = deltaY * 0.25;
    } else {
        player2Score += 1;
        ballReset();
    }
}
if (ballX > canvas.width) {
    if (ballY > paddle2Y && ballY < paddle2Y + PADDLE_HEIGHT) {
        ballSpeedX = -ballSpeedX;
var deltaY = ballY - (paddle2Y + PADDLE_HEIGHT/2);
        ballSpeedY = deltaY * 0.25;
    } else {
        player1Score += 1;
        ballReset();
    }
}
}

if (ballY < 0) {
    ballSpeedY = -ballSpeedY;
}
if (ballY > canvas.height) {
    ballSpeedY = -ballSpeedY;
}
}

```




Funcția `moveEverything` (mișcăTotul) are rolul de a apela funcția care calculează mișcările computerului și de a afla exact unde se află mingea (atât coordonatele X, cât și Y). Structurile decizionale IF au două instrucțiuni de bază: în cazul în care mingea nu atinge paleta unui utilizator, celălalt primește un punct și jocul se reia, iar în cazul în care mingea atinge paleta, mingea își schimbă direcția și jocul nu este oprit.

```

function drawNet() {
    for(var i=0; i<= canvas.height; i+=40) {
        colorRect(canvas.width/2-1,i,2,20,'white');
    }
}

```



Această funcție, printr-o structură repetitivă, afișează dreptunghiuri pe verticală, fiecare de 20 pixeli lungime și 2 pixeli lățime.

```

function drawEverything() {
    colorRect(0, 0, canvas.width, canvas.height, 'black');

    if (showingWinScreen) {
        canvasContext.fillStyle = 'white';
    }
    if (player1Score >= WINNING_SCORE) {

```

```

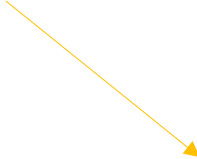
        canvasContext.fillText("Ai castigat!", 350, 200);
    } else if (player2Score >= WINNING_SCORE) {
        canvasContext.fillText("Ai pierdut...", 350, 200);
    }

    canvasContext.fillText("Click pentru a continua", 350, 500);
return;
}

drawNet();
//paleta din stanga
colorRect(0,paddle1Y,PADDLE_THICKNESS,PADDLE_HEIGHT,'white');
//paleta din dreapta
colorRect(canvas.width-
PADDLE_THICKNESS,paddle2Y,PADDLE_THICKNESS,PADDLE_HEIGHT,'white');
//mingea
colorCircle(ballX, ballY, 10,'white');

canvasContext.fillText(player1Score, 100, 100);
canvasContext.fillText(player2Score, canvas.width-100, 100);
}

```



Funcția drawEverything(deseneazăTotul) are rolul de a apela funcția drawNet descrisă mai sus și de a afișa toate elementele jocului: fundalul negru, scorul, textul de la finalul jocului (în funcție de câștigător), paletele jucătorilor și mingea.

```


function colorCircle(centerX, centerY, radius, drawColor) {
    canvasContext.fillStyle = drawColor;
    canvasContext.beginPath();
    canvasContext.arc(centerX, centerY, radius, 0,Math.PI*2, true);
    canvasContext.fill();
}

```

```

function colorRect(leftX, topY, width, height, drawColor) {
    canvasContext.fillStyle = drawColor;
    canvasContext.fillRect(leftX, topY, width, height);
}

```



Aceste două funcții au fost declarate pentru a facilita codul, reprezentând modul în care calculatorul trebuie să deseneze cercul, respectiv dreptunghiul (raza cercului, lungimea și lățimea dreptunghiului și culoarea utilizată).

6. Extinderi posibile ale aplicației

Aplicația este creată doar cu cod de bază, însă aceasta poate fi extinsă. Majoritatea jocurilor de tip Arcade folosesc aceste baze și cu ajutorul codului <canvas> din HTML se pot crea o mulțime de jocuri. Spre exemplu, utilizând cunoștințele de până acum, s-ar putea crea un joc de tip labirint, cu multe nivele.

Funcțiile ar fi asemănătoare, anume funcția drawEverything ar fi utilizată pentru a desena obstacolele labirintului și jucătorul. Funcția computerMovement ar fi utilizată pentru a coordona jucătorul printre obstacole, însă pentru facilitarea jocului, s-ar putea folosi și o funcție care ar transmite mișcările de la tastatură.

La acest exemplu, nu mai avem scor, așa că vom scăpa de câteva funcții, însă va trebui să oprim jocul de fiecare dată când jucătorul se lovește de un obstacol.

Exemplele pot continua, însă trebuie reținut faptul că aceste cunoștințe generale se pot folosi pentru crearea multor jocuri interactive, atât între 2 utilizatori sau mai mulți, cât și între utilizator și computer.

7. Resurse Hard și Soft necesare

Rularea acestui program implică puține resurse hard și soft.

Calculatorul și componentele sale reprezintă resursa hard principală. Resursa soft utilizată este browser-ul GOOGLE CHROME, însă se poate utiliza orice alt browser atât timp cât JavaScript este activat.

8. Bibliografie

https://ro.wikipedia.org/wiki/HyperText_Markup_Language

https://ro.wikipedia.org/wiki/Cascading_Style_Sheets

<https://ro.wikipedia.org/wiki/JavaScript>

https://ro.wikipedia.org/wiki/Joc_arcade