**Colegiul Național „Gheorghe Șincai”, Cluj-Napoca**

**Lucrare pentru obținerea**

**atestatului profesional la**

**INFORMATICĂ**

# JoyByteGaming

**Profesor coordonator: Elev:**

**Alina Cioban Ban Mihai-Constantin**

**Clasa: a XII-a D**

**CLUJ-NAPOCA**

**2023**

**Cuprins**

[JoyByteGaming 1](#_Toc134981926)

[**1.** **Justificarea motivului alegerii temei** 3](#_Toc134981927)

[**2.** **Descrierea softului utilizat** 4](#_Toc134981928)

[**3.** **Resurse hardware și software necesare** 6](#_Toc134981929)

[**4.** **Descrierea aplicației** 7](#_Toc134981930)

[**5.** **Corpul codul sursă al paginii ScoringBubbles.html** 9](#_Toc134981931)

[**6.** **Codul sursă al paginii ScoringBubbles.js** 12](#_Toc134981932)

[**7.** **Funcționalitate jocuri** 17](#_Toc134981933)

[**8.** **Posibilități de dezvoltare** 20](#_Toc134981934)

# **Justificarea motivului alegerii temei**

Acest proiect este un website de jocuri simple pentru persoane de toate vârstele. Un joc este tradiționalul și mult-iubitul joc de Piatră-Hârtie-Foarfecă, iar celălalt este o idee personală a unui joc pentru antrenare a abilităților de folosire a mouse-ului.

Am optat pentru această idee deoarece pe parcursul copilăriei mele am fost fascinat de jocurile video, astfel că am ajuns a crea câteva pe parcursul primilor ani de liceu. Prin intermediul acestui proiect, a atestatului, am reușit a învăța sa programez în HTML, CSS și JavaScript pentru a-mi atinge obiectivul de a crea un site și 2 jocuri ce să funcționeze împreună.

Acest proiect este începutul ce m-a ajutat să-mi setez bazele cunoștințelor programării web, iar pe viitor voi putea crea proiecte mult mai complexe.

# **Descrierea softului utilizat**

HyperText Markup Language (HTML) este un limbaj de marcare utilizat pentru crearea paginilor web ce pot fi afișate într-un browser. Scopul HTML este prezentarea informațiilor – paragrafe, fonturi, tabele etc.

Introducere:

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 hiperlink-uri (sau web-ul).

HTML este un format text proiectat pentru a putea fi citit și editat de oameni utilizând un editor de text simplu. Totuși scrierea și modificarea paginilor în acest fel solicită cunoștințe solide de HTML și este consumatoare de timp. Editoarele grafice, cum ar fi Macromedia Dreamweaver, Adobe GoLive sau Microsoft FrontPage, permit ca paginile web să fie tratate asemănător cu documentele Word. HTML se poate genera direct utilizând tehnologii de codare din partea serverului cum ar fi PHP, JSP sau ASP. Multe aplicații ca sistemele de gestionare a conținutului, wiki-uri și forumuri web generează pagini HTML. HTML este de asemenea utilizat în e-mail. Majoritatea aplicațiilor de e-mail folosesc un editor HTML încorporat pentru compunerea e-mail-urilor și un motor de prezentare a e mail-urilor de acest tip.

Noțiuni de bază:

HTML este prescurtarea de la Hyper Text Mark-up Language și este codul care stă la baza paginilor web. Paginile HTML sunt formate din etichete sau tag-uri și au extensia „.html” sau „.htm”. În marea lor majoritate aceste etichete sunt pereche, una de deschidere <eticheta> și alta de închidere </eticheta>, mai există și cazuri în care nu se închid, atunci se folosește <eticheta />. Navigatorul web interpretează aceste etichete afișând rezultatul pe ecran. HTML-ul este un limbaj care nu face deosebire între litere majuscule și minuscule.

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 SVGL.

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 bine-cunoscut pentru folosirea sa în construirea site-urilor web, dar este folosit și pentru accesul la obiecte încapsulate (embedded objects) în alte aplicații.

Cea mai des întâlnită utilizare a JavaScript este în scriptarea paginilor web. Programatorii web pot îngloba în paginile HTML script-uri pentru diverse activități cum ar fi verificarea datelor introduse de utilizatori sau crearea de meniuri și alte efecte animate.

Browserele rețin în memorie o reprezentare a unei pagini web sub forma unui arbore de obiecte și pun la dispoziție aceste obiecte script-urilor JavaScript, care le pot citi și manipula. Arborele de obiecte poartă numele de Document Object Model sau DOM. Există un standard W3C pentru DOM-ul pe care trebuie să îl pună la dispoziție un browser, ceea ce oferă premiza scrierii de script-uri portabile, care să funcționeze pe toate browserele. În practică, însă, standardul W3C pentru DOM este incomplet implementat. Deși tendința browserelor este de a se alinia standardului W3C, unele din acestea încă prezintă incompatibilități majore, cum este cazul Internet Explorer.

# **Resurse hardware și software necesare**

Proiectul fiind realizat in HTML, funcționarea corectă depinde de foarte puțini factori la nivel hardware.

Resurse minime:

1. Hardware

* Procesor (CPU): 900 MHz+
* Memorie (RAM): 64 MB
* Monitor: VGA 16-bit color

2. Software

* Sistem de operare: Windows XP
* Browser internet: Google Chrome, Internet Explorer 6.0, Mozilla Firefox 3.0.4, Opera

Resurse recomandate:

1. Hardware

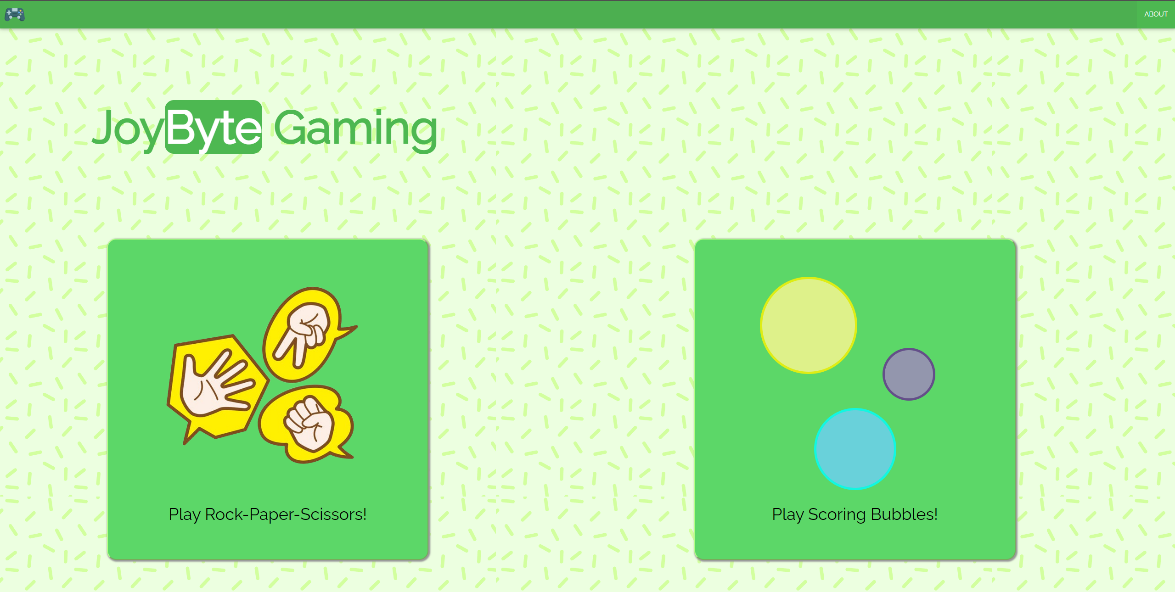
* Procesor (CPU): 1200 MHz+
* Memorie (RAM): 256 MB
* Monitor: QHD (2560x1560)

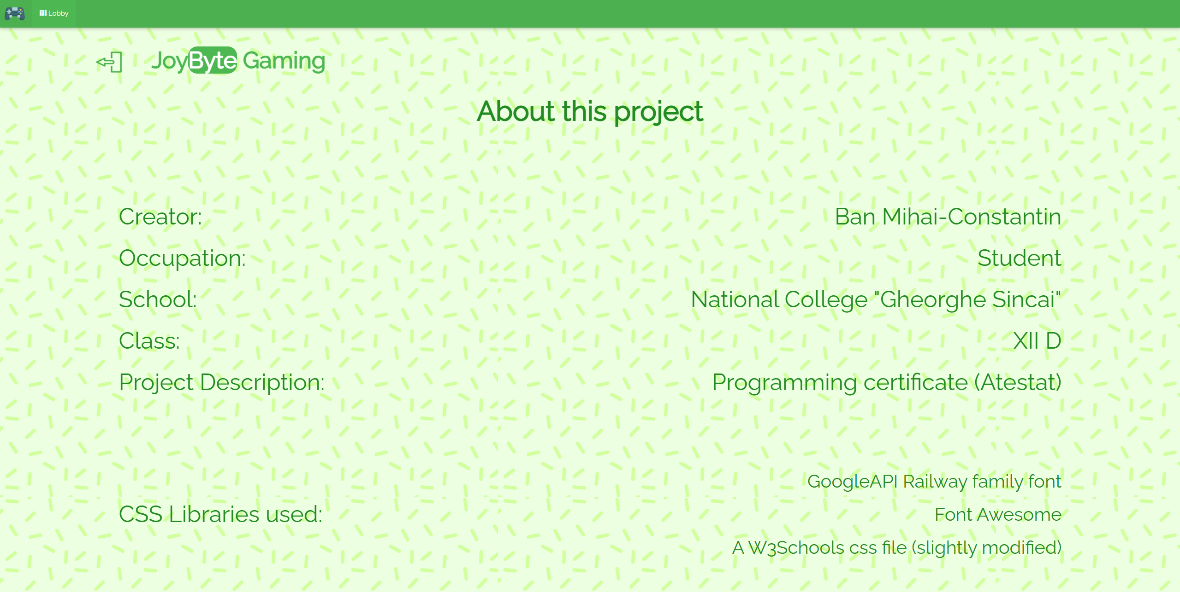
2. Software

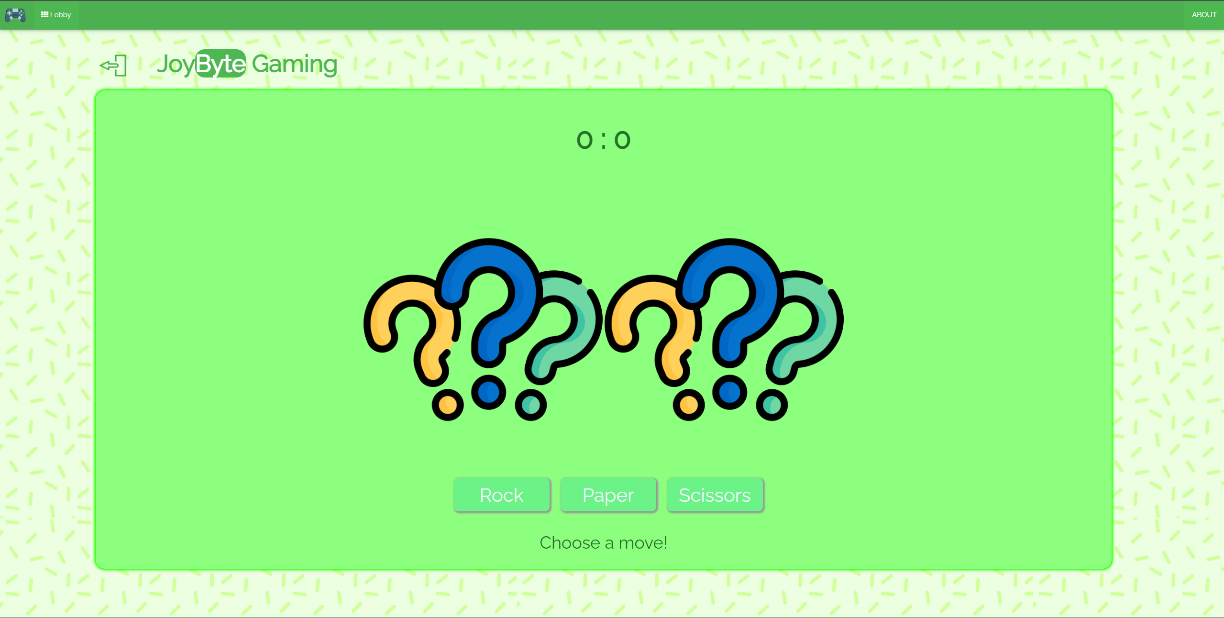
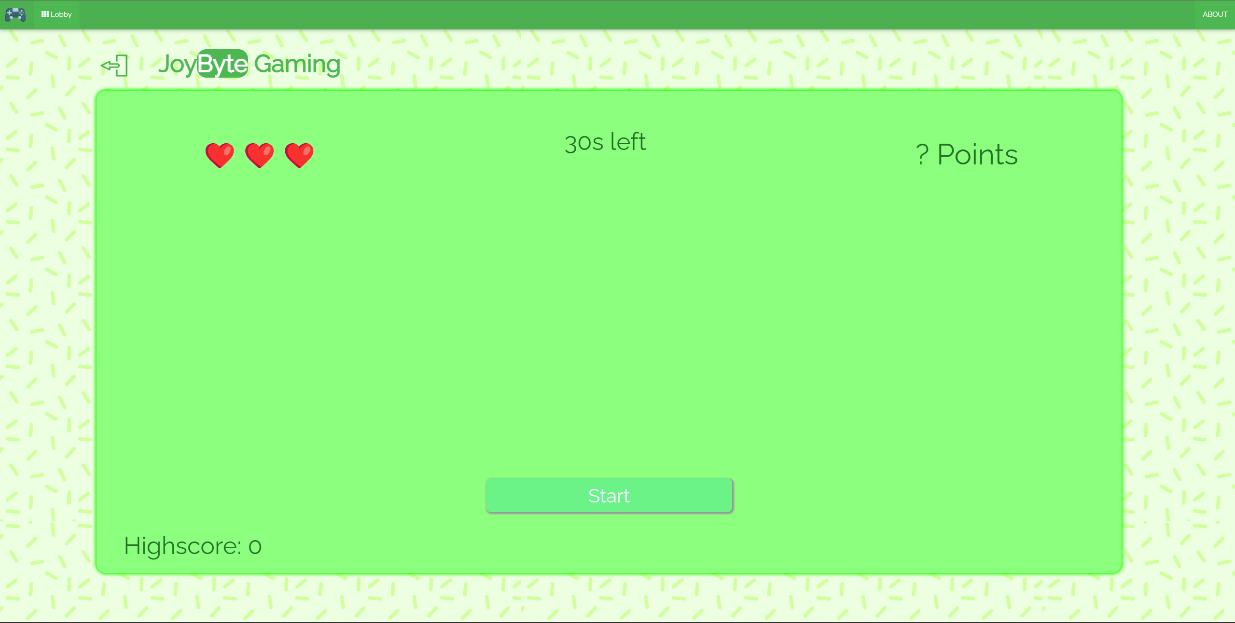
* Sistem de operare: Windows 10 sau mai nou
* Browser internet: Google Chrome

# **Descrierea aplicației**

Proiectul este format din 4 pagini web, legate între ele prin butoane intuitive.

Pagina “Lobby.html” este pagina principală de la care se poate naviga oriunde altundeva. Cu ajutorul butoanele ce conțin numele fiecărui joc se poate ajunge la paginile jocurilor “PiatraHartieFoarfeca.html” și „ScoringBubbles.html”. Butonul din dreapta-sus al paginii cu numele “About” duce utilizatorul la pagina “About.html” unde sunt prezente informații despre proiect.

Pagina “About.html” conține 2 butoane ce pot redirecționa utilizatorul către pagina principală, unul fiind în stânga-sus cu numele “Lobby”, iar celălalt în stânga numelui proiectului, un buton intuitiv de revenire la pagina anterioară.

Asemenea paginii “About.html”, paginile celor 2 jocuri (“PiatraHartieFoarfeca.html” și „ScoringBubbles.html”) conțin 2 butoane pentru a reveni la pagina principală, acestea având încă un buton, asemenea paginii principale, ce duce utilizatorul la “About.html”.

Paginile “About.html” și “Lobby.html” sunt create doar cu ajutorul limbajelor HTML și CSS, iar paginile de jocuri, “PiatraHartieFoarfeca.html” și “ScoringBubbles.html”, funcționează cu ajutorul limbajelor HTML, CSS si JavaScript, cea din urmă ocupându-se majoritar de funcționalitate.

Toate resursele sunt ordonate în foldere pentru a putea fi găsite, schimbate și folosite eficient.

# **Corpul codul sursă al paginii ScoringBubbles.html**

Acesta este doar conținutul din interiorul etichetei <body>.

<body background="Images/PatternSprinklesGreen.png">

    <!-- navbar (sit on top) -->

    <div class="w3-top">

        <div class="w3-bar w3-green w3-card" id="myNavbar">

            <img src="Images/Joystick.png" class="w3-bar-item" height="60px" width="100px" style="padding-left: 10px;">

            <a href="Lobby.html" class="w3-bar-item w3-green2 w3-button">

                <i class="fa fa-th"></i> Lobby

            </a>

            <!-- right-sided navbar links -->

            <div class="w3-right w3-hide-small">

                <a href="About.html" class="w3-bar-item w3-green2 w3-button">ABOUT</a>

            </div>

            <a href="javascript:void(0)" class="w3-bar-item w3-button w3-right w3-hide-large w3-hide-medium" onclick="w3\_open()">

                <i class="fa fa-bars"></i>

            </a>

        </div>

    </div>

    <!--top JoyByte site name and back icon-->

    <h1 style="font-size:50px; padding-top: 75px; padding-left: 200px;">

        <form action="Lobby.html">

            <button type="submit"; style="border: 0px; background: none; padding-right: 50px;">

                <img src="Images/backIcon2.png" onmouseover="hoverOverSound()" height="50px" width="60px" class="iconGreen nonSelect">

            </button>

            <b style="color: rgb(77, 184, 81);">Joy</b><b class="w3-green2" style="border-radius: 20px;">Byte</b>  <b style="color: rgb(77, 184, 81);">Gaming</b>

        </form>

    </h1>

    <div style="padding-left: 200px;">

        <div id="GameCanvas" style="background-color: rgb(141, 255, 126); border-color: rgb(180, 255, 136); color: rgb(35, 114, 39); width: 90%; height: 1000px; border-radius: 20px; border-width: 5px; padding-top: 4%;" class="bordershadow">

            <table style="width:100%;">

                <tr>

                    <td style="padding-left: 10%">

                        <h1 style="font-size: 60px;" class="hearts nonSelect" id="Hearts">

                            ❤️❤️❤️

                        </h1>

                    </td>

                    <td style="padding-right: 10%">

                        <h1 style="font-size: 60px;" class="points nonSelect" id="Score">

                            ? Points

                        </h1>

                    </td>

                </tr>

            </table>

            <h1 style="align-content: center;" class="centeredDiv nonSelect timer nonSelect" id="Timer">

                30s left

            </h1>

            <!-- don't remove, it's important -->

            <div id="BubblesPlace"></div>

            <h1 style="align-content: center; color:white; visibility: hidden;" class="gameOver centeredDiv nonSelect" id="GameOver">

                Game Over! <br>

                Play again?

            </h1>

            <div style="align-content: center;" class="playButton centeredDiv nonSelect" id="StartButton">

                <button onclick="startGame(); buttonSound();" style="border: none; border-radius: 10px; color: aliceblue; font-size: 40px;" class="choiceBorderShadow">Start</button>

            </div>

            <div style="align-content: center; font-size: 50px;" class="highscore centeredDiv nonSelect" id="Highscore">

                Highscore: 0

            </div>

        </div>

    </div>

    <script src="js\_and\_css/ScoringBubbles.js"></script>

    <script>

        var audio1 = document.getElementById("sound1");

        var audio2 = document.getElementById("sound2");

        function buttonSound(volume) {

            audio1.volume = volume || .05;

            audio1.load();

            audio1.play();

        }

        function hoverOverSound(volume) {

            audio2.volume = volume || .1;

            audio2.load();

            audio2.play();

        }

    </script>

</body>

# **Codul sursă al paginii ScoringBubbles.js**

Acest cod se ocupă de funcționalitatea codului expus anterior (al paginii „ScoringBubbles.html”)

// objects

var scoreObj = document.getElementById("Score")

var heartsObj = document.getElementById("Hearts")

var bubblesPlace = document.getElementById("BubblesPlace")

var startButton = document.getElementById("StartButton")

var gameOver = document.getElementById("GameOver")

var timerObj = document.getElementById("Timer")

var highscoreObj = document.getElementById("Highscore")

// other vars

var gameState = "canStart"; // canStart, playing

var score = 0;

var highscore = 0;

var hearts = 3;

var gameIntervalID;

var time = 60;

// tables

var bubbleValues = [

    [

        "🖤",

        80, // px

    ],

    [

        100,

        100,

    ],

    [

        100,

        120,

    ],

    [

        250,

        70,

    ],

    [

        500,

        60,

    ]

]

// functions

function startGame(){

    if(gameState!="canStart"){return};

    gameState = "playing";

    gameOver.style.visibility = "hidden";

    startButton.style.visibility = "hidden";

    score = 0;

    hearts = 3;

    updateUI();

    time = 30;

    timerObj.innerHTML = time + "s left";

    updateTimer()

    createBubble();

}

function endGame(){

    gameState = "canStart";

    clearTimeout(gameTimeoutID);

    gameOver.style.visibility = "visible";

    startButton.style.visibility = "visible";

}

function updateTimer(){

    setTimeout(function(){

        time--;

        timerObj.innerHTML = time + "s left";

        if(time<=0){

            timerObj.innerHTML = "Time's up!"

            endGame()

        }

        if(gameState=="canStart") return;

        updateTimer();

    }, 1000)

}

function manageHighscore(){

    if(score>highscore){

        highscore = score;

        highscoreObj.innerHTML = "Highscore: " + highscore;

    }

}

function updateUI(){

    scoreObj.innerHTML = score + " Points";

    heartsObj.innerHTML = "❤️".repeat(hearts);

    if(hearts == 0)

        endGame();

}

function createBubble(){

    var bubbleNr = getRndInteger(0, bubbleValues.length-1);

    var bubbleObj = document.createElement('button');

    var lastingTime = getRndInteger(20, 40)/10;

    bubbleObj.state = "ready"

    bubbleObj.classList.add('bubble');

    bubbleObj.style.height = bubbleValues[bubbleNr][1] + "px";

    bubbleObj.style.width = bubbleValues[bubbleNr][1] + "px";

    bubbleObj.style.fontSize = bubbleValues[bubbleNr][1]/2.5 + "px";

    bubbleObj.style.top = getRndInteger(30, 80) + "%";

    bubbleObj.style.left = getRndInteger(30, 70) + "%";

    bubbleObj.addEventListener("click", function(){

        clickedBubble(bubbleObj);

    });

    bubbleObj.bValueIndex = bubbleNr

    bubbleObj.innerHTML = bubbleValues[bubbleNr][0];

    giveBubbleRndColor(bubbleObj);

    bubblesPlace.appendChild(bubbleObj);

    setTimeout(function(){

        destroyBubble(bubbleObj);

    }, lastingTime.toString()\*1000);

    gameTimeoutID = setTimeout(createBubble, getRndInteger(2, 6)\*100)

}

function clickedBubble(bubbleObj){

    destroyBubble(bubbleObj);

    if(gameState=="playing"){

        if(bubbleObj.bValueIndex!=0){

            rewardSFX();

            score += bubbleValues[bubbleObj.bValueIndex][0];

            manageHighscore();

        }

        else{

            wrongSFX();

            hearts--;

        }

        updateUI();

        bubbleObj.state = "popped"

    }

}

function destroyBubble(bubbleObj){

    if(bubbleObj.state=="ready"){

        bubblesPlace.removeChild(bubbleObj);

        bubbleObj.state = "destroyed"

    }

}

function getRndInteger(min, max){

    return Math.floor(Math.random() \* (max+0.999 - min) ) + min;

}

function giveBubbleRndColor(bubbleObj){

    var h = getRndInteger(0, 360);

    var s = getRndInteger(70, 100);

    var l = getRndInteger(50, 70);

    bubbleObj.style.borderColor = "hsl(" +

        + h + ","

        + s + "%,"

        + l

    + "%)"

    bubbleObj.style.backgroundColor = "hsl(" +

        + h + ","

        + s/1.5 + "%,"

        + l

    + "%)"

}

var audio3 = document.getElementById("sound3");

function rewardSFX(volume){

    audio3.volume = volume || .1;

    audio3.load();

    audio3.play();

}

var audio4 = document.getElementById("sound4");

function wrongSFX(volume){

    audio4.volume = volume || .06;

    audio4.load();

    audio4.play();

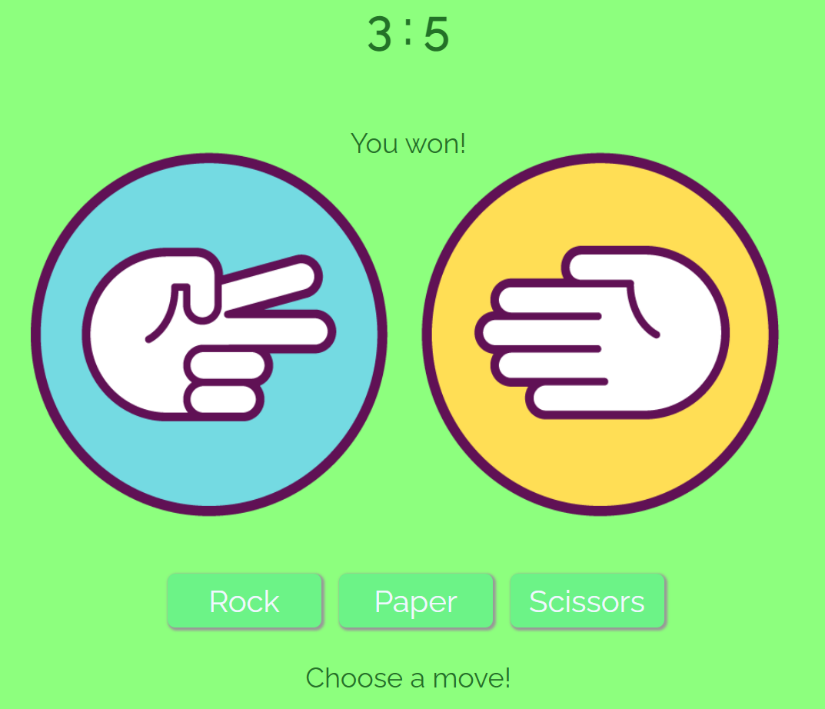
}

# **Funcționalitate jocuri**

1. Piatră-Hârtie-Foarfecă

În acest joc, utilizatorul poate alege una dintre cele 3 mutări (Piatră/Hârtie/Foarfecă), iar apoi o mutare la întâmplare dintre cele 3 este aleasă de calculator. În continuare se aplică regulile jocului tradițional.

Un text deasupra celor două mutări este afișat ce comunică jucătorului dacă a câștigat, pierdut sau este remiză, scorul schimbând-se conform rezultatului.



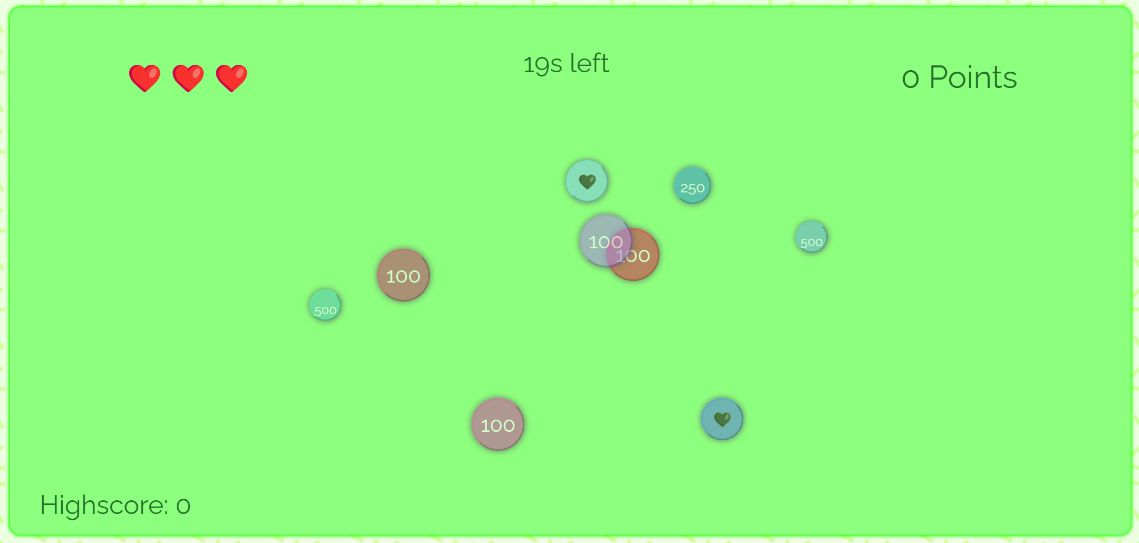
1. Scoring Bubbles

În acest joc, utilizatorul poate apăsa pe butonul Start ce începe jocul.

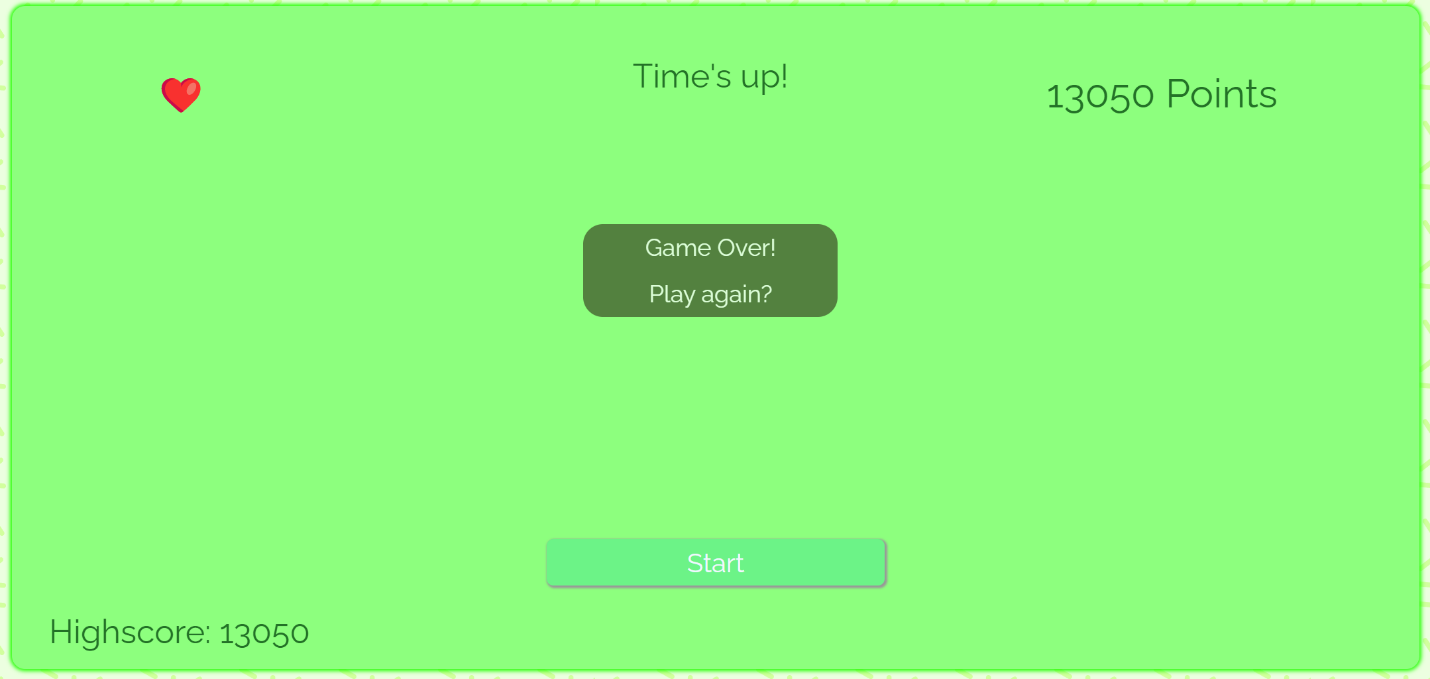


După apăsarea butonului, acesta dispare și niște obiecte circulare numite “Bubbles” (bule) apar pe ecran în diferite culori și mărimi, cu diferite texte scrise pe acestea. Textul de pe fiecare bulă determină cate puncte sunt câștigate de către jucător la spargerea bulei (click cu mouse-ul pe obiect) înainte ca aceasta să dispară. Cu cât numărul de pe bulă este mai mare, cu atât bula este mai mică pentru a fi mai dificil a o sparge.

Jucătorul trebuie, totuși, să aibă grijă deoarece bulele cu inimă neagră în loc de un punctaj îi iau o viață, iar la 3 vieți pierdute jocul se încheie.



Jucătorul are 30 de secunde de la începerea jocului pentru a strânge cât mai multe puncte, acestea fiind afișate în dreapta-sus, iar în colțul opus este afișat cel mai mare scor obținut de jucător pe parcursul tuturor jocurilor/rundelor sale. Acest “Highscore” este actualizat în timp real.

La finalizarea unui joc este afișat un mesaj pe ecran ce comunică jucătorului acest fapt, utilizatorul putând vedea scorul și numărul de inimi rămase.

# **Posibilități de dezvoltare**

Site-ul ar mai putea fi extins, astfel aș putea crea mai multe jocuri, făcându-l un site similar celor din copilăria mea cum ar fi Y8 sau Friv, aceste site-uri incluzând mii de jocuri gratis ce pot fi jucate până în ziua de azi.

Jocurile ce sunt deja create pot fi dezvoltate în sensul în care pentru ScoringBubbles ar putea, de exemplu, exista un Leaderboard unde oricine poate să se întreacă pentru cel mai bun scor în acea tură de 30 de secunde.

Fiecare joc ar putea primi rating-uri de la jucători, precum și idei, sugestii și rapoarte față de posibile probleme (bug-uri) ce se pot trimite pe email. De asemenea, utilizatorul s-ar putea loga pe această platformă, sa primească insigne pentru realizările sale. Ar putea exista un forum unde pot discuta cei pasionați de jocuri.