

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII
COLEGIUL NAȚIONAL „GH. VRĂNCEANU”, BACĂU

LUCRARE PENTRU OBȚINEREA
CERTIFICATULUI DE COMPETENȚĂ
PROFESIONALĂ ÎN INFORMATICĂ

Coordonator,
Profesor Mihaela Ciubotariu

Autor,
Butacu Ștefan-Alexandru

2020

Jocul Spânzurătoarea

- website -

CUPRINS

| | |
|--|---|
| Argument | 1 |
| Cap. I - Noțiuni generale despre HTML..... | 2 |
| HTML - Paragraf..... | 4 |
| HTML - Titluri..... | 4 |
| HTML - Linebreak..... | 4 |
| HTML - Line Orizontala | 4 |
| HTML - Liste | 5 |
| HTML - Link-uri | 5 |
| HTML - Imagini | 5 |
| HTML - Formulare..... | 5 |
| HTML - Campuri de text..... | 6 |
| HTML - Tabele | 6 |
| HTML - Bgcolor | 6 |
| HTML - Background | 7 |
| HTML - Comentarii | 7 |
| HTML - Embed (Muzica) | 7 |
| HTML - Object (Video) | 7 |
| HTML - Text ingrosat (Bold) | 7 |
| HTML - Text inclinat..... | 7 |
| HTML - Subscript, Superscript, Striketrough..... | 8 |
| HTML - Upload si Campuri ascunse | 8 |
| Cap. II - Relatia CSS-HTML | 9 |
| Introducere - De ce folosim CSS..... | 9 |
| Introducere - Ce este CSS - Generalități..... | 9 |

| | |
|--|----|
| Cap. III – Noțiuni generale despre JavaScript | 10 |
| Introducere – Ce este JavaScript - Generalități..... | 10 |
| Ce este jQuery? | 11 |
| Cap. IV – Prezentarea site-ului web..... | 12 |
| Prezentarea jocului..... | 12 |
| Dezvoltarea aplicației | 12 |
| 1. Fișierul HTML..... | 12 |
| 2. Fișierul CSS | 13 |
| 3. Fișierul JavaScript | 18 |
| Cap.V – Capturi de ecran | 24 |
| Bibliografie | 29 |

Argument

Am ales ca temă implementarea arhi-cunoscutului joc Spânzurătoarea într-o formă technologizată, sfecifică erei internetului în care trăim.

Noua generație nu mai valoarea jocului pe hârtie și consider prin această metodă că pe lângă atragerea acestora în a juca forma digitalizată, se ameliorează cantitatea imensă de hârtie procesată la nivel global. În scurt timp, totul se va transpune într-un format electronic, începând jocurile copilăriei.

În paginile ce vor urma se vor descrie succind atât povestea jocului și tehnologiile utilizate în implementarea aplicației (HTML, CSS, JavaScript), cât și pașii realizării acestui proiect.

Cap. I - Noțiuni generale despre HTML

HTML este acronimul de la **HyperText Markup Language** și reprezintă un limbaj pentru crearea și marcarea (formatare, aranjare) unui document astfel încât să poată fi publicat pe World Wide Web și vizualizat cu ajutorul unui browser (Internet Explorer, Netscape, Opera, Mozilla Firefox etc.).

Termenul de hypertext desemnează un material sub formă de text și imagine, interconectat într-o manieră complexă, nesecvențială, în care utilizatorul poate naviga, căuta informații referitoare la un obiect. Hypertext-ul trebuie interpretat ca un text care semnalează o legătură la o altă informație web, de obicei un alt document web, și este identificat prin subliniere sau culoare, pentru a-l deosebi de textul simplu.

Hypermedia este un termen aproape sinonim celui de hypertext, singura deosebire fiind faptul că subliniază prezența și a unor elemente care nu sunt de tip text, cum ar fi animații, secvențe sonore sau secvențe video.

HTML se utilizează din 1990, cunoscând câteva versiuni de dezvoltare, fiecare dintre acestea îmbunătățind performanțele limbajului. Ultima variantă (la data elaborării acestui ghid) este HTML 4.01. ce include facilitățile versiunilor anterioare (tag-uri de marcarea, tag-uri pentru hiperlegături, antete, paragrafe, liste, elemente de meniu, formatare caractere, imagini in-line și tag-uri pentru schimbul de date dinamic între utilizatori), adăugând facilități și extensii pentru numere, tabele și elemente de control.

Un document HTML este format dintr-o succesiune de blocuri de informație. Aceste blocuri pot fi incluse unul în altul. Un bloc este delimitat de simboluri speciale, numite **tag-uri** (etichete). Modul în care un document este marcat cu elemente și cu attribute ale acestor elemente se realizează în conformitate cu Document Type Definition (DTD – definiția tipului de document). Aceasta conține regulile ce caracterizează fiecare tip de document.

Sursa autorizată pentru furnizarea de informații despre HTML și HTML DTD este World Wide Web Consortium (W3C) având adresa

Tag-ul este termenul consacrat pentru a defini elementele cu care sunt marcate textul și grafica într-o pagină web. Fiecare tag este încadrat de semnele "<" (mai mic) și ">" (mai mare).

De exemplu: Acest text se va vedea îngroșat într-un browser

Primul cuvânt sau caracter ce apare în interiorul acestor paranteze poartă numele de **element**.

Majoritatea elementelor au un tag de deschidere și unul de închidere cu aceeași structură, dar cu prezența caracterului "/" în fața denumirii elementului.

De exemplu: este un tag de închidere

Cuvântul sau cuvintele ce urmează după element și despărțite de acesta printr-un spațiu, poartă de numirea de **attribute**, având rolul de descrie proprietățile elementelor.

Attributele sunt urmate de semnul "=" și pot avea diferite **valori**. Valoarea unui atribut este trecută, de obicei, între ghilimele.

De exemplu: Acest text va fi albastru

Elementele HTML pot avea unul sau mai multe atribute:

De exemplu: `Acest text va fi albastru și cu o unitate mai mare decât normal`

De remarcat că atributele nu apar și în tag-urile de închidere.

Elementele HTML specifică structura logică a unui document web și sugerează prezentarea vizuală a documentului. HTML furnizează două tipuri de elemente:

- tag-urile care permit delimitarea antetelor, paragrafelor, listelor, tabelelor, legăturilor, imaginilor, etc.;
- referințele de entitate caracter care permit utilizarea în documentele HTML a simbolurilor declarate drept caractere de control. Acestea se identifică ușor deoarece întotdeauna încep cu **ampersand** "&" . De exemplu, pentru a reprezenta simbolul "<" într-un document HTML, se va utiliza referința de entitate caracter **<** (less then).

Etichete HTML

| eticheta | semnificatie |
|-----------------------------------|---|
| <code><!-- ??? --></code> | comentariu |
| <code><!DOCTYPE ...></code> | tipul documentului |
| <code><a ...></code> | legatura (link) |
| <code><area></code> | arie, folosita la hartile de imagini |
| <code></code> | bold, text ingrosat |
| <code><bgsound></code> | adauga paginii muzica de fond |
| <code><body ...></code> | corpul paginii, contine informatia ce va fi afisata |
| <code> </code> | sfarsit de rand |
| <code><caption ...></code> | "titlul" tabelului |
| <code><code></code> | text continand cod de computer |
| <code><cite></code> | citatie (text italic) |
| <code></code> | text taiat cu o linie orizontala |
| <code><div></code> | formateaza un sir de caractere |
| <code><dl></code> | lista de definitii |
| <code><dt></code> | termen dintr-o lista de definitii |
| <code></code> | text evidentiat |
| <code></code> | font |
| <code><form ...></code> | formular |
| <code><frame ...></code> | cadru |
| <code><frameset ...></code> | defineste blocul compus din cadre |
| <code><h1 ...></code> | header sau titlu (de la h1 la h6) |
| <code><head></code> | defineste antetul paginii HTML |
| <code><hr ...></code> | introduce o linie orizontala |
| <code><html></code> | anunta ca blocul este tip HTML |
| <code><i></code> | text italic |
| <code></code> | imagine |
| <code><input ...></code> | element (camp) al formularului |
| <code><kbd></code> | text introdus de la tastatura |
| <code><li ...></code> | termen al listei |
| <code><link ...></code> | legatura |
| <code><meta></code> | metainformatii despre pagina |
| <code><noframes></code> | text alternativ pentru cadru |

| | |
|--------------|---|
| | lista ordonata |
| <option ...> | element al listei de selectii |
| <p ...> | paragraf nou |
| <select ...> | lista de selectii din formular |
| <samp> | mostra de cod |
| | formateaza mai multe elemente |
| <small> | font micorat |
| | text ingrosat |
| <table ..> | tabel |
| <td ...> | celula a tabelului |
| <textarea> | camp din formular in care se introduce text |
| <th> | headerul tabelului |
| <title> | titlul paginii HTML |
| <tr ...> | rand al tabelului |
| <tt> | text monospatiat |
| <u> | text subliniat |
| | lista neordonata |

HTML - Paragraf

<p>Acesta este un simplu paragraf in HTML</p>
 <p align="left">Acesta este un paragraf aliniat la stanga</p>

Pe langa atributul align putem folosi pentru a alinia elementele HTML urmatoarele valori: *left*, *right*, *center* si *justify*.

HTML - Titluri

<h1>Acesta este tag-ul HTML pentru cel mai mare titlu</h1>

Alte tipuri de titluri in **HTML**: h2, h3, h4, h5, h6. Unde h1 este cel mai mare si h6 cel mai mic.

HTML - Linebreak

 este un salt in linie in **HTML** - spatiul intre br si slash face diferenta intre cod **valid XHTML** si **HTML 4.1** de exemplu.

HTML - Line Orizontala

<hr /> se foloseste pentru a desena o linie orizontala in **HTML** - spatiul intre hr si slash face diferenta intre **cod valid XHTML** si **HTML 4.1** de exemplu..

HTML - Liste

O lista simpla in **HTML** este de forma

```
<ol>
<li>Primul element HTML din lista </li>
<li>Al doilea element HTML din lista</li>
<li>Al treilea element HTML din lista</li>
</ol>
```

Lista de mai sus este o lista ordonata. Alte tipuri de liste in **HTML** sunt liste neordonata si lista de definitii.

HTML - Link-uri

```
<a href="http://www.tutorialehtml.com/" title="HTML 5" target="_blank" >Html 5</a>
<a href="#top">Mergi la inceput</a> (<a name="top"></a>)
<a href="mailto:admin@exemplu.com" >Nelamuriri aici</a>
<a href="http://www.tutorialehtml.com/" title="HTML 5" target="_blank" ></a>
```

Au fost exemplificate mai sus un link normal in **HTML**, un link folosit pentru a lega doua sectiuni in aceasi pagina, un link email si un link imagine. Desi nu se mai foloseste in **HTML** dar merita mentionat.

```
<base href="http://www.tutorialehtml.com/">
Base href a fost scos din indexul tagurilor in HTML 5
```

HTML - Imagini

```
 Va recomand alinierea cu ajutorul CSS. La fel si stabilirea dimensiunilor
imaguinii. Ceea ce ar transforma codul HTML de mai sus in:

```

HTML - Formulare

Am adaugat mai jos **codul HTML** folosit pentru a crea un simplu formular. Acesta este doar formularul HTML folosit de user pentru a introduce informatia dorita. Nu uita ca pentru procesarea aceste informatii, odata ce se face click pe butonul submit al formularului HTML. Informatia va fi procesata de catre un fisier.php si eventual urcata in baza de date sau trimisa pe email.

```

<form method="post"
name="formular" action="fisier_care_proceseaza_formularul_html.php">
.....campurile de text care vor prelua informatia.....
    <input type="submit" value="Trimite">
    <input type="reset" value="Sterge" />
</form>

```

HTML - Campuri de text

Am atasat mai jos **codurile HTML** folosite pentru campuri de text in functie de cerinte.

```

<input type="text" size="10" maxlength="40" name="nume"> - camp de text normal
<input type="password" size="10" maxlength="10" name="parola"> - parola
<input type="radio" name="culoare" value="rosu"> - buton radio
<input type="checkbox" name="da" value="da"> - casuta de bifat
<select name="limbajdeprogramare"> - meniu drop-down
<option>Html 4.1</option>
<option>Html 5</option>
</select>

```

HTML - Tabele

Structura de baza a unui *tabel in HTML* este urmatoarea..

```

<table>
  <tr>
    <th>Column 1</th>
    <th>Column 2</th>
  </tr>
  <tr>
    <td>Row 1 Cell 1</td>
    <td>Row 1 Cell 2</td>
  </tr>
  <tr>
    <td>Row 2 Cell 1</td>
    <td>Row 2 Cell 2</td>
  </tr>
</table>

```

HTML - Bgcolor

Iata folosirea atributului bgcolor in **HTML**. Scopul lui este acela de a stabili o culoare de fond.

```

<body bgcolor="#efefef">
<div bgcolor="#888888">
<table bgcolor="#000000"> - etc.

```

HTML - Background

Puntru a pune o imagine background vom folosi urmatorul cod HTML.

```
<table background="img/pattern.jpg">
```

HTML - Comentarii

```
<!-- Acesta este un comentariu simplu in Html -->
```

HTML - Embed (Muzica)

Pentru a insera muzica intr-un document.html vom folosi urmatorul rand **HTML**.

```
<embed src="tu si tuborg.mp3" hidden="false" autostart="false" loop="false" volume="60" width="144" height="60" />
```

HTML - Object (Video)

Pentru a insera un video intr-un document.html vom folosi urmatoarea bucata de cod **HTML**.

```
<object width="425" height="344">
<param name="movie"
value="http://www.youtube.com/v/UAq8qHNWMNw&hl=en&fs=1"></param>
<param name="allowFullScreen" value="true"></param>
<embed src="http://www.youtube.com/v/UAq8qHNWMNw&hl=en&fs=1"
type="application/x-shockwave-flash" allowfullscreen="true" width="425" height="344">
</embed>
</object>
```

HTML - Text ingrosat (Bold)

```
<b>text ingrosat in html</b>
<strong>text ingrosat in html</strong>
```

HTML - Text inclinat

```
<i>italic</i>
<em>emphasized</em>
<blockquote>blockquote</blockquote>
<address>address</address>
```

HTML - Subscript, Superscript, Strikethrough

`_{indice!}`

`^{exponential!}`

`text taiat`

In general aceste trei tag-uri HTML nu se folosesc prea mult dar merita amintite

HTML - Upload si Campuri ascunse

Exemplul de mai jos este campul folosit pentru a urca un fisier pe server. Tine minte ca este doar *partea HTML* a formularului.

```
<form method="post" action="fisier_care_processeaza_formularul_html.php">  
  <input type="hidden" name="MAX_FILE_SIZE" value="500" />  
  <input type="file" />  
  <input type="submit" value="Trimite">  
  <input type="reset" value="Sterge" />  
</form>
```

Cap. II - Relatia CSS-HTML

Introducere - De ce folosim CSS

Cu ajutorul **CSS**-ului putem crea pagini web simple dar si complexe folosind efecte diverse. Cu **CSS** putem stabili culoarea, marimea si fontul textului, deasemenea putem crea un **layout** (un suport pentru elementele HTML) personalizat adaugand margine culoare sau imagine de fond si multe altele. In concluzie **CSS** te poate scapa de multe batai de cap, ajutandute in a mentine codul html cat mai simplu si mai ordonat.

Introducere - Ce este CSS - Generalități

Cascading Style Sheets pe numele lui mic **CSS** se foloseste pentru a personaliza tagurile HTML.

In principiu HTML a fost conceput pentru a marca elementele unei pagini:

```
<html>
<head>
<title>.....</title>
</head>
<body>
<h1>.....</h1>
<p>.....</p>
</body>
</html>
```

Odata cu introducerea HTML 3.2 au fost introduse si atributele de personalizare a tagurilor precum "font", "color" etc. A fost atunci cand limbajul de programare HTML a devenit greoi. Fiecare pagina a websitului trebuia luata separat si modificate proprietatile elementelor principale.

Aceasta problema a fost rezolvata in versiunea 4.0 a HTML-ului. Toate atributele de personalizare au fost scoase si salvate intr-un fisier extern cu extensia ".css".

In felul acesta modificand un singur fisier putem schimba forma in care sunt afisate toate paginile unui website. Putem schimba culoarea textului, fontul, marimea, putem personaliza div-uri, formulare si multe altele.

Cap. III – Noțiuni generale despre JavaScript

Introducere – Ce este JavaScript - Generalități

JavaScript (numit oficial ECMAScript) a apărut în 1995 și a fost creat cu scopul de a oferi, printre altele, un anumit nivel de interactivitate pentru dezvoltari (programatori), rămânând folosit în aceea perioadă la un nivel relativ mic în website-uri (ex: în trimiterea informațiilor din formulare de contact). Pe parcursul anilor, a tot crescut, având tot mai mulți dezvoltari (și firme mari) care adăugau diferite facilități limbajului.

În general se consideră că există zece aspecte fundamentale ale limbajului **JavaScript** pe care orice programator în acest limbaj ar trebui să le cunoască :

- JavaScript poate fi introdus în HTML - De obicei codul JavaScript este gazduit în documentele HTML și executat în interiorul lor. Majoritatea obiectelor JavaScript au etichete HTML pe care le reprezintă, astfel încât programul este inclus pe partea de client a limbajului. JavaScript folosește HTML pentru a intra în cadrul de lucru al aplicațiilor pentru web.
- JavaScript este dependent de mediu - JavaScript este un limbaj de scriptare; software-ul care rulează de fapt programul este browser-ul web (Firefox, Opera, Netscape Navigator, Internet Explorer, Safari, etc.) Este important să luăm în considerare această dependență de browser atunci când utilizăm aplicații JavaScript.
- JavaScript este un limbaj în totalitate interpretat - nu necesită compilări sau preprocesări, ci rămâne parte integrantă a documentului HTML. Dezavantajul acestui limbaj este că rularea durează ceva mai mult deoarece comenzile JavaScript vor fi citite de navigatorul Web și procesate atunci când user-ul apelează la acele funcții (prin completare de formulare, apăsare de butoane, etc). Avantajul principal este faptul că putem mult mai ușor să actualizăm codul sursă.
- JavaScript este un limbaj flexibil - în aceasta privință limbajul diferă radical de C++ sau Java. În JavaScript putem declara o variabilă de un anumit tip, sau putem lucra cu o variabilă deși nu-i cunoaștem tipul specificat înainte de rulare.
- JavaScript este bazat pe obiecte - JavaScript nu este un limbaj de programare orientat obiect, ca Java, ci mai corect, este "bazat pe obiecte"; modelul de obiect JavaScript este bazat pe instanță și nu pe moștenire.
- JavaScript este condus de evenimente - mare parte a codului JavaScript răspunde la evenimente generate de utilizator sau de sistem. Obiectele HTML, cum ar fi butoanele, sunt îmbunătățite pentru a accepta handler de evenimente.
- JavaScript nu este Java - Cele două limbaje au fost create de companii diferite, motivul denumirii asemănătoare este legat doar de marketing.
- JavaScript este multifuncțional - limbajul poate fi folosit într-o multitudine de contexte pentru a rezolva diferite probleme: grafice, matematice, și altele.

- JavaScript evoluează - limbajul evoluează, fapt pozitiv care însă poate genera și probleme, programatorii trebuind să verifice permanent ce versiune să folosească pentru ca aplicațiile să poată fi disponibile unui număr cât mai mare de utilizatori de browsere diferite.
- JavaScript acoperă contexte diverse - programarea cu acest limbaj este îndreptată mai ales către partea de client, dar putem folosi JavaScript și pentru partea de Server. JavaScript este limbajul nativ pentru unele instrumente de dezvoltare web, ca Borland IntraBuilder sau Macromedia Dreamweaver

Ce este jQuery?

După mulți ani de zile și multe framework-uri, unul a ieșit în evidență, **jQuery** (2006). jQuery a ajutat JavaScript să fie mult mai folosit în crearea de site-uri dinamice, în special din cauza Ajax-ului. Ce a oferit Ajax-ul a fost abilitatea de a schimba pagini fără full page refresh. Când navighezi pe mare parte din site-uri și dai click pe un link, vei vedea că înainte ca pagina să se încarce, totul dispare pentru o secundă sau mai puțin, după care apare pagina nouă — acesta e un full page refresh. Cu AJAX, asta nu se mai întâmplă, și se încarcă doar ce avem nevoie.

Partea mai interesantă e de ce a devenit faimos acest framework; nu pentru că a adus ceva ce nu exista deja în JavaScript, ci doar făcea toată treaba mult mai ușoară. Domeniul care a primit cel mai mult ajutor de la jQuery a fost și este cel al front-end developerilor (cei care fac partea pe care o vedem pe site-uri, cum ar fi afișarea corectă a paginilor web, adăugarea de comentarii, mesajele de notificare, site-uri responsive, etc.)

Altă confuzie e aceea că **JavaScript** este același lucru cu **jQuery**. JavaScript e limbajul pe care e construit jQuery, acesta din urmă fiind o librărie care face scrierea de cod JavaScript mult mai ușoară.

Cap. IV – Prezentarea site-ului web

Prezentarea jocului

Spânzurătoarea este un joc pentru doi jucători: utilizatorul și calculatorul sau doi utilizatori, moduri specifice pentru singleplayer și multiplayer. În cadrul modului singleplayer mecanismul aplicației alege un cuvânt, iar utilizatorul trebuie să-l ghicească sugerând literele de la tastatură. Dacă jucătorul sugerează o litera ce se află în cuvânt, aceasta se completează automat în toate pozițiile unde apare. Dacă litera nu se află în cuvânt, procesul spânzurătorii se dezvoltă. Jocul se trece la următoarea etapă când: jucătorul completează tot cuvântul sau numărul de încercări eșuate ajunge la 6. În modul singleplayer, cuvintele nu se repetă, de aceea utilizatorul finalizează întregul joc atunci când ghicește toate cuvintele stoacate în baza de date, fără a pierde la nicio etapă. De asemenea, acesta are la dispoziție câte un indiciu pentru fiecare cuvânt.

În modul multiplayer, unul dintre utilizatori tastează un cuvânt de maxim 14 caractere în căsuța de pe ecran, cuvânt pe care adversarul său trebuie să-l ghicească după mecanismul prezentat anterior. În acest mod nu există indicii.

În cazul în care utilizatorul pierde, cuvântul ascuns este afișat pe ecran.

În orice moment, utilizatorul se poate întoarce la meniul principal, unde alege modul de joc prin intermediul săgeții „Înapoi”.

Dezvoltarea aplicației

1. Fișierul HTML

```
<!DOCTYPE html>

<head>
  <meta charset="UTF-8">
  <title>Jocul Spanzuratoarea</title>
  <link href="main.css" rel="stylesheet" type="text/css" />
  <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no">
  <link href="https://fonts.googleapis.com/css2?family=Amatic+SC:wght@700&fa
mily=Oswald&display=swap" rel="stylesheet">
</head>

<body>
  <div id="topbar">Spânzurătoarea
    <hr class="hrNew">
  </div>

  <div class="spacer"></div>
  <div id="gameContent">
  </div>
  <script src="jquery.js"></script>
```



```
<script src="controller.js"></script>
</body>

</html>
```

Linia `<meta charset="UTF-8">` ajută pentru a se afișa diacriticele în browser.

Linia `<title>` este pentru a afișa în fila browser-ului titlul proiectului.

Următoarea linie atașează documentul numit „main.css” care o să conțină toate informațiile despre stilul elementelor (culori, font, vizibilitate etc).

Următoarele rânduri din `<body>` vor conține div-urile care vor reține conținutul paginii. „Topbar” va conține titlul paginii. Div-ul „spacer” va crea un spațiu gol pe pagina, „GameContent” este locul în care va fi afișat jocul.

Prima linie `<script>` spune paginii să caute un fișier Javascript numit „jquery”. Aceasta este o „extensie” standard JavaScript folosită la scară largă de dezvoltatori. A doua linie `<script>` spune paginii să caute un fișier JavaScript numit „controller.js”. Acest fișier conține toate funcționalitățile cu privire la derularea jocului. Acestea sunt adăugate la finalul documentului pentru a facilita navigarea pe site, încărcându-se prima oară detaliile vizuale, după aceea funcționalitățile.

2. Fișierul CSS

```
html,body {
  margin: 0;
  padding: 0;
  background-color: #E0E0E0;
  font-family: 'Oswald', cursive;
}

#topbar {
  width: 80%;
  height: 50px;
  margin: auto;
  margin-top: 40px;
  color: #1a1818;
  text-align: center;
  font-size: 48px;
  font-weight: bold;
  font-family: 'Amatic SC', cursive;
}
```

- **Hmtl,body:** conțin stilul paginii principale.
- **#topbar:** se va folosi pentru a afișa textul „Spânzurătoarea”.

```

.hrNew {
    border: 0;
    height: 2px;
    background-image: linear-
gradient(to right, rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.75), rgba(0, 0, 0, 0));
}

#gameContent {
    margin: auto;
    width: 950px;
    height: -webkit-fill-available;
    display: flex;
    align-items: center;
    justify-content: center;
    align-content: flex-start;
    overflow: hidden;
    background-color: #E0E0E0;
    border-radius: 10px;
    margin-top: -3px;
}

```

- **#hrNew**: este folosit pentru a se sublinia titlul jocului.
- **#gameContent**: controlează zona stilului din suprafața de joc.

```

#arrowHolder
{
    top: 150px;
    right: 150px;
    cursor: pointer;
    position: absolute;
}

#arrow
{
    height: 50px;
    width: 50px;
}

.spacer {
    height: 20px;
    margin-bottom: 20px;
}

```

- **#arrowHolder, #arrow:** este folosit pentru a stiliza săgeata de reincepere a jocului.
- **.spacer:** este folosit pentru a delimita titlul de zona de joc.

```
#gameTitle {
    margin-top: 70px;
    text-align: center;
    font-size: 40px;
    color: #404040;
}

#startButtonSP, #startButtonMP, #startMP{
    margin: 0 auto;
}
```

- **#gameTitle:** este folosit pentru a se stiliza titlul jocului.
- **#startButtonSP, #startButtonMP, #startMP:** sunt ID-uri folosite pentru butoanele prin care se alege tipul de joc

```
.playMP
{
    font-size: 20px;
    width: 100px;
    position: relative;
    cursor: pointer;
    border: solid 2px black;
    border-radius: 4px;
    text-align: center;
    background-color: #c1c1c1;
    font-weight: bold;
    color: black;
}

.button {
    font-size: 20px;
    width: 250px;
    margin: auto;
    margin-top: 20px;
    padding: 20px;
    cursor: pointer;
    border: solid 2px black;
    border-radius: 4px;
    text-align: center;
    background-color: #c1c1c1;
    font-weight: bold;
    color: black;
}
```

- **.playMp, .button:** sunt clase care permit utilizatorului să aibă o interacțiune mai plăcută, prin apăsarea unor butoane stilizate.

```
.buttonhelp{
  width: 70px;
  padding: 5px 10px;
  border: solid 2px black;
  border-radius: 4px;
  text-align: center;
  background-color: #c1c1c1;
  font-weight: bold;
  color: black;
}

.button:hover, .buttonhelp:hover {
  background-color: #FFBF46;
  color: black;
}
```

- **.buttonHelp:** este clasa pentru div-ul care stochează indiciul cuvântului curent.

```
#wordHolder {
  margin-left: 150px;
  margin-top: 50px;
}

#clueHolder {
  color: black;
  font-size: 18px;
  font-weight: bold;
  margin-top: 130px;
  margin-left: 150px;
  cursor:pointer;
}

#guesses {
  color: black;
  font-size: 24px;
  margin-top: 10px;
  margin-left: 150px;
}
```

```
#pixHolder {
    margin: 0 auto;
    display: block;
    width: 75px;
    float: center;
    overflow: hidden;
}

#pixHolder img {
    position: relative;
}
```

- **#wordHolder:** este zona de pe ecran care va conține spațiile goale în care va merge cuvântul.
- **#clueHolder:** va afișa indiciul curent
- **#guessses:** va fi zona în care se afișează caracterele încercate, dar care nu fac parte din cuvânt.
- **#pixHolder:** va ține imaginea bărbatului spânzurat. Aceasta se va deplasa cu 75 de pixeli la fiecare caracter greșit.

```
#feedback {
    margin-top: 20px;
    font-size: 36px;
    color: #404040;
    text-align: center;
}
```

- **#feedback:** jucătorul va fi informat dacă a câștigat sau dacă a rămas fără nicio încercare.

```
.tile {
    height: 50px;
    width: 50px;
    float: left;
    margin-top: 5px;
    margin-right: 10px;
    padding-bottom: 7px;
    background-color: white;
    text-align: center;
    font-size: 42px;
    color: #333;
    font-weight: bold;
    font-family: 'Amatic SC', cursive;
}
```

- **.tile:** este stilul pentru fiecare căsuță în care se vor afla literele din cuvânt.

```
#finalMessage {
  text-align: center;
  font-size: 40px;
  color: rgb(0, 0, 0);
  width: 90%;
  margin: auto;
  margin-top: 100px;
}
```

- **#finalMessage:** este folosit pentru mesajul afișat la sfârșitul jocului, când utilizatorul ghicește toate cuvintele.

3. Fișierul JavaScript

Definirea variabilelor globale

```
var questionBankMP = new Array;
var questionBankSP = new Array;
var questionBank = new Array;
var wordArray = new Array;
var previousGuesses = new Array;
var currentWord;
var currentClue;
var wrongAnswerCount;
```

- **questionBankMP:** este folosit pentru a stoca șirul introdus de către utilizator în modul Multiplayer.
- **questionBankSP:** este un vector ce funcționează ca o bază de date ce conține cuvintele și indiciile inserate de către programator
- **questionBank:** este un vector ce va fi folosit atât în modul Singleplayer cât și în cel Multiplayer.
- **wordArray:** stochează cuvântul actual din timpul jocului
- **previousGuesses:** este vectorul ce conține caracterele încercate deja de utilizator
- **currentWord:** este cuvântul pe care trebuie să-l ghicească utilizatorul, ce va fi afișat în caz că nu reușește
- **currentClue:** este indiciul specific fiecărui cuvânt
- **wrongAnswerCount:** este utilizat pentru a număra caracterele greșite ce nu se află în cuvânt.

```
function titleScreen() {
    $('#gameContent').append('<div id="startButtonSP" class="button">ÎNCEP  
EȚI JOCUL SINGLEPLAYER</div> ');
    $('#gameContent').append('<div id="startButtonMP" class="button">ÎNCEP  
EȚI JOCUL MULTIPLAYER</div> ');
    $('#startButtonSP').on("click", function () {
        $('#gameContent').css("display", "block");
        questionBank = questionBankSP;
        gameScreenSP();
    });
    $('#startButtonMP').on("click", function () {
        gameScreenMP();
    });
} //display game
```

- **titleScreen():** funcție care creează un ecran de titlu. Aici există cele 2 butoane care sunt folosite pentru a selecta modul de joc Singleplayer sau Multiplayer, în cazul singleplayer se face atribuirea vecotrului de cuvinte specifice.

```
function gameScreenMP() {
    $('#gameContent').empty();
    $('#gameContent').css("display", "grid");
    $('#gameContent').append('<div id="writeWords"><p>Va rugăm sa introduc  
eți cuvântul în căsuța text de mai jos: </p><input type="text" id = "myWord" p  
laceholder="scrie" autocomplete="off"></div><br/>');
    $('#gameContent').append('<div id = "startMP" class = "playMP"> TRIMITE  
CUVÂNTUL </div> ');

    $('#gameContent').append('<div id= "arrowHolder"> <img id = "arrow" sr  
c="back.png"> <p style = "text-align:center"> Înapoi </p> </div>');
    $('#arrow').on("click", function () {
        location.reload();
    });

    $('#startMP').on("click", function () {

        if ($("#myWord").val().length >= 14) {
            alert("Vă rugăm să introduceți alt cuvânt, deoarece ați depăși  
t 14 caractere!");
            //jump back to writing the word
            gameScreenMP(); // sterge sirul tastat anterior in input
        }
    })
}
```

```

        else {
            if ($("#myWord").val().length == 0) {
                alert("Vă rugăm să introduceți un cuvânt!");
            }
            else {
                var x = document.getElementById('myWord').value;
                var indiciu = "NU PRIMIȚI INDICIU IN ACEST MOD";
                questionBankMP = [[x, indiciu]];
                questionBank = questionBankMP;
                $("#gameContent").css("display", "block");
                gameScreenSP();
            }
        }
    });
}

```

- **gameScreenMP()**: creează un ecran pentru modul **Multiplayer**. Acesta are posibilitatea de a se întoarce la ecranul de pornire prin apăsarea săgeții. De asemenea, poate tasta cuvântul pe care trebuie să-l ghicească jucătorul numărul doi. Cuvântul trimis trebuie să aibă cel puțin un caracter și maxim 14. Apoi se apelează funcția **gameScreenSP()** ce conține celelalte funcționalități.

```

function gameScreenSP() {
    $('#gameContent').empty();
    $('#gameContent').append('<div id="pixHolder"><br/></div>');
    $('#gameContent').append('<div id="wordHolder"></div>');
    $('#gameContent').append('<div id="clueHolder"></div>');
    $('#gameContent').append('<div id="guesses">Litere încercate:</div>');
    $('#gameContent').append('<div id="feedback"></div>');

    $('#gameContent').append('<div id= "arrowHolder">  <p style = "text-align:center"> Înapoi </p> </div>');
    $('#arrow').on("click", function () {
        location.reload();
    });

    getWord();
    var numberOfTiles = currentWord.length;
    wrongAnswerCount = 0;
    previousGuesses = [];

    for (i = 0; i < numberOfTiles; i++) {
        $('#wordHolder').append('<div class="tile" id=t' + i + '></div>');
    }
}

```



```

    $('#clueHolder').append(' <div id="startButtonSP" class="buttonhelp" style="margin:initial;">Indiciu</div> ');
    $('#startButtonSP').on("click", function () {
        document.getElementById("clueHolder").innerHTML = "Indiciu: " + currentClue;
    })

    $(document).on("keyup", handleKeyUp);
} //gameScreenSP

```

- **gameScreenSP()**

- se golește containerul #gameContent
- Se adaugă următoarele secțiuni: #pixHolder, #worHolder, #pixHolder, #clueHolder, #guesses, #feedback, #arrowHolder cu semnificațiile de mai sus.
- Se apelează funcția getWord() care va alege un cuvânt aleatoriu din vectorul questionBank. getWord() va genera variabila currentWord și vom folosi lungimea acestui cuvânt pentru a determina câte casete goale trebuie afișate.

```

function getWord() {
    var rnd = Math.floor(Math.random() * questionBank.length);
    currentWord = questionBank[rnd][0];
    currentClue = questionBank[rnd][1];
    questionBank.splice(rnd, 1);
    wordArray = currentWord.split("");
} //getword

```

- **getWord()**: se definește o variabila **rnd** și este folosită pentru a selecta o valoare aleatorie între 0 și lungimea vectorului de cuvinte. Cu această variabilă, se stabilește cuvântul curent și indicele curent. Apoi se elimină cuvântul și indicele tocmai ales.

```

function handleKeyUp(event) {
    if (event.keyCode > 64 && event.keyCode < 91) {
        var found = false;
        var previouslyEntered = false;
        var input = String.fromCharCode(event.keyCode).toLowerCase();

        for (i = 0; i < previousGuesses.length; i++) { if (input == previousGuesses[i]) { previouslyEntered = true; } }

        if (!previouslyEntered) {
            previousGuesses.push(input);

            for (i = 0; i < wordArray.length; i++) {

```

```

        if (input == wordArray[i]) { found = true; $('#t' + i).append(input); }

    } //for

    if (found) { checkAnswer(); }
    else { wrongAnswer(input); }
} //if
} //handlekeyup

```

- **handleKeyUp()**: se permite introducerea de la tastură a unui caracter. Se definesc două variabile locale **found** și **previouslyEntered** cu următoarele semnificații: found pentru a verifica dacă litera se găsește în cuvânt și previouslyEntered va stabili dacă litera a fost trimisă anterior și pentru a nu o mai lua în calcul.

```

function checkAnswer() {
    var currentAnswer = "";
    for (i = 0; i < currentWord.length; i++) {
        currentAnswer += ($('#t' + i).text());
    }
    if (currentAnswer == currentWord) {
        victoryMessage();
    };
} //checkanswer

function victoryMessage() {
    $(document).off("keyup", handleKeyUp);
    $('#feedback').append("Ai castigat!<br><br><div id='replay' class='button'>CONTINUAȚI</div>");
    $('#replay').on("click", function () {
        console.log(questionBank.length);
        if (questionBank.length > 0) {

            gameScreenSP()
        }
        else { finalPage() }
    });
}

```

- **checkAnswer()**: verifică dacă întregul cuvânt a fost rezolvat, comparând cuvântul curent alcătuit din caracterele tastate de utilizator și verifică dacă este identic cu

cuvântul aflat în baza de date. Atunci se folosește funcția **victoryMessage()** pentru a afișa un mesaj de victorie.

- **victoryMessage()**: afișează mesajul de victorie și cu un buton pentru a continua jocul. În cazul în care au rămas cuvinte ne jucate în baza de date se pornește automat un nou joc.

```
function wrongAnswer(a) {
    wrongAnswerCount++;
    var pos = (wrongAnswerCount * -75) + "px"
    $('#guesses').append(" " + a);
    $('#hangman').css("left", pos);
    if (wrongAnswerCount == 6) {
        defeatMessage();
    }
} //wronganswer

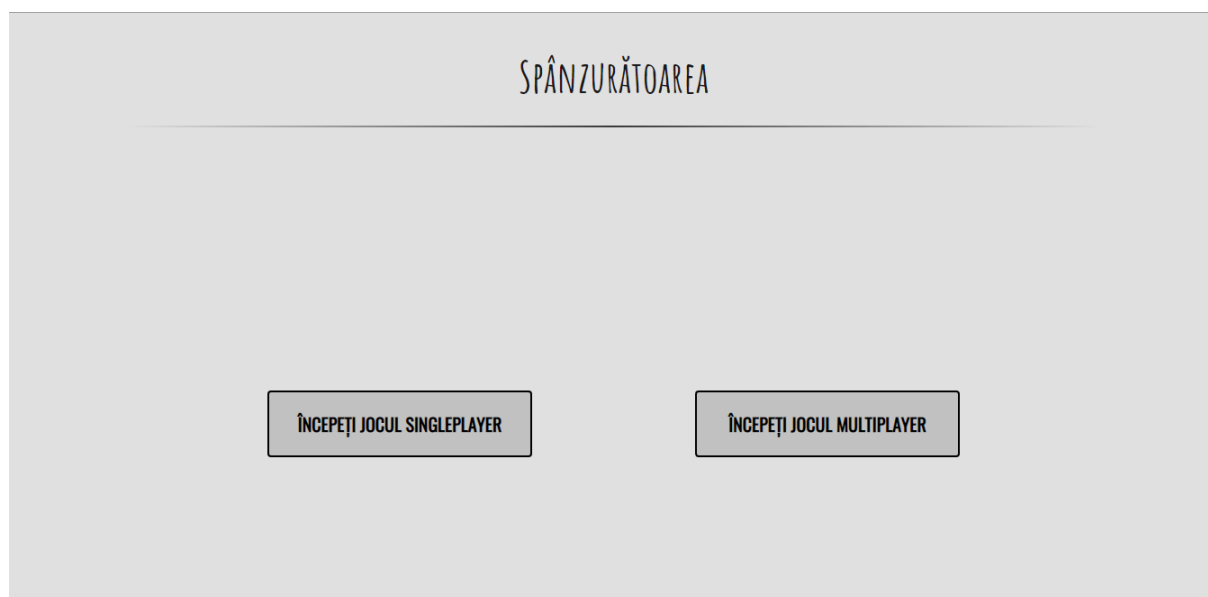
function defeatMessage() {
    $(document).off("keyup", handleKeyUp);
    $('#feedback').append("Ai murit!<br>(Răspunsul este: " + currentWord +
    "<div id='replay' class='button' style='margin-top:20px;'>CONTINUĂȚI</div>");
    $('#replay').on("click", function () {
        if (questionBank.length > 0) {
            gameScreenSP();
        }
        else { finalPage() }
    });
}
```

- **wrongAnswer()**: în cazul în care litera nu apare în cuvânt. Se incrementează variabila wrongAnswerCount, folosită pentru a deplasa imaginea stiker-ului spânzurat. Dacă numărul de caractere introduse greșit atinge valoarea 6 se afișează mesajul de eșec, **defeatMessage()**.
 - **defeatMessage()**: afișează mesajul și răspunsul pe care jucătorul nu l-a ghicit. Dacă au mai rămas cuvinte nefolosite, se începe un nou joc.

```
function finalPage() {
    $('#gameContent').empty();
    $('#gameContent').append('<div id="writeWords"><p>Felicitări! Ați câștigat jocul, ghicind toate cuvintele din joc! </p>');
    $('#gameContent').append('<div id="backToHome" class="button">(RE)ÎNCEPEȚI UN ALT JOC</div>');
    $('#backToHome').on("click", function () {
        location.reload(); }); //finalpage
}
```

- **finalPage()**: Se afișează mesajul când utilizatorul ghicește toate cuvintele din joc.

Cap.V – Capturi de ecran



SPÂNZURĂTOAREA



Înapoi

P E R _ E _ E R E _ _ A

Indiciu: A nu te da batut

Litere încercate: j g k

SPÂNZURĂTOAREA



Înapoi

P E R S E V E R E N T A

Indiciu: A nu te da batut

Litere încercate: j k g

Ai câștigat!

CONTINUĂȚI

SPÂNZURĂTOAREA



Înapoi

J _ _ _ S _ R I _ T

Indiciu: Limbaj de programare

Litere încercate: k w q n l f

Ai murit!
(Răspunsul este: javascript)

CONTINUĂȚI

SPÂNZURĂTOAREA

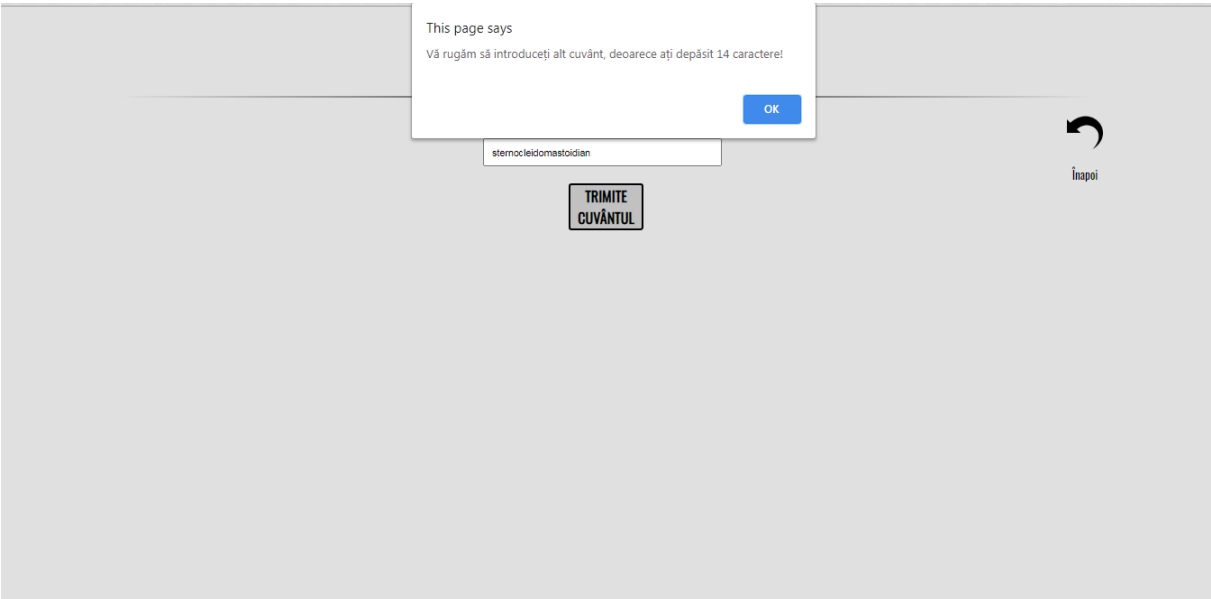
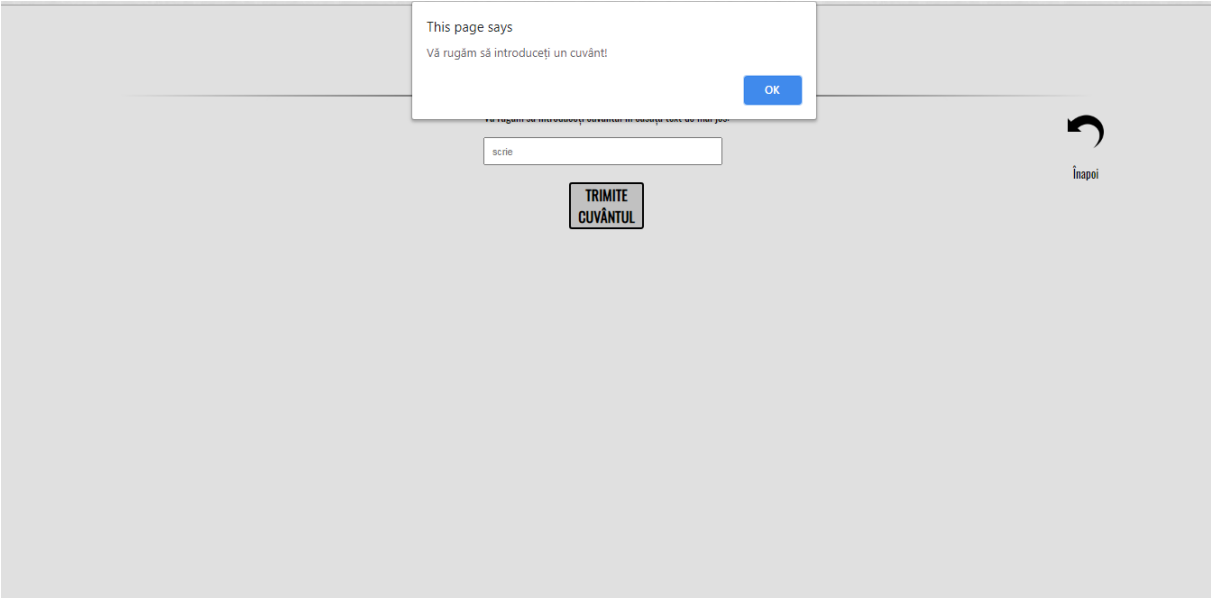
Va rugăm să introduceți cuvântul în căsuța text de mai jos:

scrie

TRIMITE
CUVÂNTUL



Înapoi



SPÂNZURĂTOAREA



Înapoi

M A S I N A

Indiciu: NU PRIMIȚI INDICIU ÎN ACEST MOD

Litere încercate: t l k f

Ai câștigat!

CONTINUAȚI

SPÂNZURĂTOAREA

Felicitări! Ai câștigat jocul, ghicind toate cuvintele din joc!

(RE)ÎNCEPEȚI UN ALT JOC

Bibliografie

- www.TutorialeHTML.com
- <http://www.flashbynight.com/tutes/hangmanhtml5/>
- <https://www.link-academy.com/curs-introducere-html-css#>
- <https://medium.com/tutoreel-articole-%C8%99i-g%C3%A2nduri/de-ce-s%C4%83-%C3%AEnve%C8%9Bi-javascript-3f9ba82ac0af>
- <https://web.ceiti.md/lesson.php?id=16>