

ACADEMIA DE STUDII ECONOMICE

Facultatea: Cibernetică, Statistică și Informatică Economică

Specializarea: *Informatică economică*

## **Proiect de practică**

***Tema proiectului:* Implementarea unui joc de  
Minesweeper utilizand tehnologii web**

**Cadrul didactic coordonator:  
Prof. Univ. Dr. Claudiu Vințe**

**Student:  
Mașala Alexandra  
Grupa 1052**

**București  
2020**

# *Cuprins*

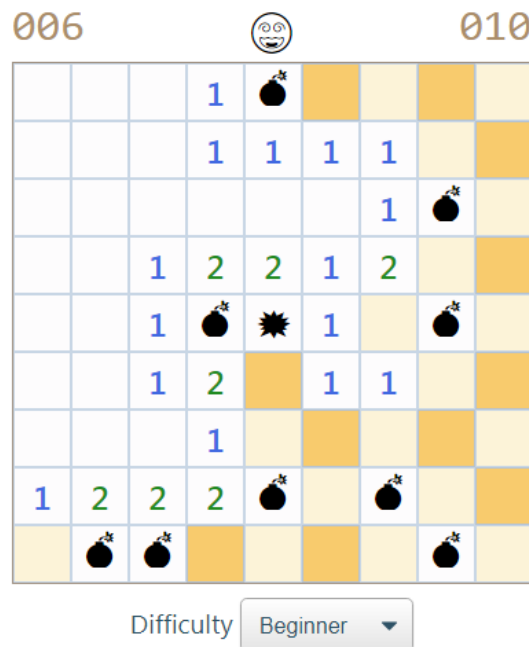
Introducere.....	3
1. Prezentare companie Ubisoft.....	4
2. Elaborarea aplicatiei Minesweeper.....	5
2.1. Proiectarea etapei de interfata si interactivitate .....	5
2.2 .NET, APIs, server-side.....	7
2.3. Baze de date, abordare, tabele, optimizari .....	8
Concluzii.....	9
Bibliografie.....	9

# Introducere

Acest document prezinta rezultatul stagiului meu de practica in cadrul proiectului Tap That Job, unde am avut ocazia sa studiez in amanunt programele de limbaj Javascript si CSS, dar si C#. Ca si aplicatii am folosit Visual Studio, Visual Studio Code si Postman. Participarea la acest eveniment a reprezentat a doua experienta de lucru profesional pentru mine.

Aplicatia realizata de mine drept tema in cadrul lui Tap That Job este un joc puzzle Minesweeper similar cu versiunea clasica adaugata pe Windows in 1990.

Versiunea finala ar trebui sa reflecte lucrul cu interfete, implementarea functionalitatilor si lucrul cu baza de date, pentru a putea pastra datele, dar avand in vedere timpul scurt alocat proiectului la finalul semestrului si organizarea afectata de pandemia COVID-19, aplicatia este in acest moment la un stadiu intermediar.



## **1. Prezentare companie Ubisoft**

Ubisoft este unul dintre liderii în entertainment, cu un portofoliu bogat de branduri cunoscute la nivel mondial. Iar misiunea echipelor care lucrează la studiourile Ubisoft din întreaga lume este să livreze experiențe de gaming originale și memorabile pentru toate platformele. Ubisoft București este un pilon important al industriei de gaming din România, fiind una dintre primele companii de acest gen, încă din 1992.

În departamentul Productie, cele mai cautate profile sunt cele de programator, game designer și level designer. În departamentul IT, ce creează și mentine buna funcționare a aplicațiilor și sistemelor informatice, sunt disponibile 10 familii de job-uri, cu focus pe poziții ca PHP Developer, Solutions Architect, .net Architect, Linux System Administrator sau BI Developer. Cei care aleg departamentul Quality Control trebuie să aibă atenție sporită la detalii pentru a le asigura utilizatorilor cea mai bună experiență de gaming.

Astfel, Ubisoft caută profesioniști pasionați de industria jocurilor video și IT, cu abilități de programare în C/C++, PHP, .NET, SQL server, creativi și dornici să facă parte din echipe ingenioase.

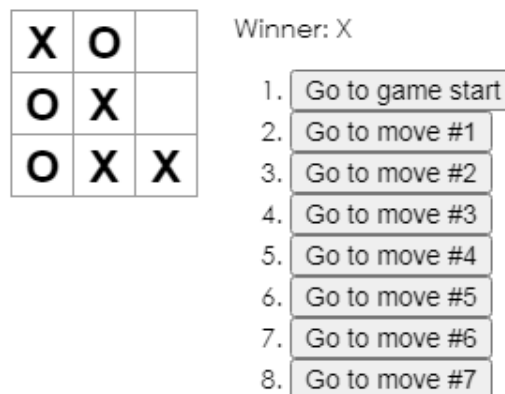
## 2. Elaborarea aplicatiei Minesweeper

### 2.1. Proiectarea etapei de interfata si interactivitate

In prima zi de activitate, reprezentantii Ubisoft ne-au oferit materiale initiale pe care sa le parcurgem cu scopul de a ne atinge obiectivul.

Pentru a incepe dezvoltarea jocului, am utilizat libraria JavaScript numita React cunoscuta drept o sintaxa speciala care permite utilizatorului sa combine HTML cu JavaScript.

Am invatat React ca incepator in JavaScript urmand un tutorial introductiv pentru a face un joc tic-tac-toe<sup>1</sup> deoarece este bine documentat si foarte usor de urmarit. Dupa completarea tutorialului am facut un joc asemanator ca aspect cu tic-tac-toe pentru a putea intelege bazele.



Mediul in care am programat interfetele, precum si functionalitati mai avansate, este NodeJS<sup>2</sup>. Instalarea NodeJS vine cu NPM<sup>3</sup> care e managerul de module pentru Node.

---

<sup>1</sup> <https://reactjs.org/tutorial/tutorial.html>

<sup>2</sup> <https://nodejs.org/en/>

<sup>3</sup> <https://www.npmjs.com/>

Ruland comanda pentru creare de aplicatii pe ReactJS in Visual Studio Code, am instalat toate dependintele, serverul de web, partea de compilare, debug, precum si alte tool-uri.:

```
npx create-react-app my-app
```

```
cd my-app
```

```
npm start
```

Dupa instalarea NodeJS, l-am rulat (in proiectul deja creat pe github<sup>4</sup>) cu “npx create-react-app ./“, apoi cu “npm start”.

Ultima comanda va da drumul la serverul de web, si permite accesarea proiectului initial de ReactJS la portul <http://localhost:3000>.

Pana la acest pas e un dummy app, dar e baza pe care am inceput sa construiesc jocul.

Tutorialul oficial pentru un joc de X si 0 a trebuit sa fie remodelat pentru un joc de Minesweeper astfel:

Un Minesweeper este o matrice mai mare, in care fiecare casuta/celula poate contine o bomba sau nu.

Bombele vor fi alocate random la inceputul jocului. Initial valorile tuturor celulelor sunt ascunse cu o anumita culoare.

In schema aplicatiei, avem 3 componente principale:

Game component - aici dam dimensiunile matricei (inaltime, latime), precum si numarul de bombe.

Cell component - e celula din matrice, care ascunde o valoare:

- daca celula e inca ascunsa, va arata o banala celula implicit colorata

---

<sup>4</sup> <https://github.com/alexmasala/minesweeper/tree/pr/padLeft>

- daca celula a fost dezvaluata, si e o bomba, aratam bomba ( Game over )

- daca celula a fost dezvaluata si nu e bomba, arata o celula alba

Board component - cea mai complexa componenta, este cea care contine logica jocului. In aceasta componenta se va crea matricea pe baza dimensiunilor trimise de Game Component, se vor alocata bombele random si va afisa plansa de joc. Pe langa matrice, e bine sa avem si un mineCount (cate bombe raman de descoperit) pentru determinarea scor-ului.

Atunci cand un joc este castigat, se va face un apel de API cu mineCount (precum si numele jucatorului sau orice alte detalii avem) pentru a fi salvate in baza de date.

Intr-o varianta avansata, pentru o a avea un joc mai interactiv - putem afisa numarul de bombe vecine, precum si sa putem pune stegulete pe celulele care ar putea fi bombe:

Cell component - adaugam posibile valori/stari pentru vecini si flags

- daca celula nu a fost dezvaluata, dar jucatorul a dat click dreapta pe ea (flagged it), va arata o imagine cu stegulet

- daca celula a fost dezvaluata si are bombe in jur, vom arata numarul de bombe din zona

- daca celula a fost dezvaluata si nu are bombe in jur, aratam o celula alba

## 2.2 .NET, APIs, server-side

Odata ce avem partea de react functionala vom face call catre cele 2 api endpoints ( una POST pentru salvarea scorului si alte detalii in data de baze, una GET pentru listarea scorului/leaderboardului din data de baze si afisarea in aplicatie a acestor date).

### Codul pentru trimiterea scorului catre API:

```
var sendScoreToAPI = () => {  
    //get player name from browser prompt  
    var playerName = prompt("Congrats for winning the game! Please enter your name: ",  
    "Alexa");
```

```

    if (playerName !== null) {
      var dataToSave = {
        playerScore: 10, //replace 10 with your actual variable (probably
this.state.gameScore or this.state.time)
        playerName: playerName,
        currentTime: new Date()
      };
      // Actual API call
      fetch(
        "https://api.example.com/minesweeper", // replace with the url to your API
        {
          method: 'POST',
          headers: {
            'Content-Type': 'application/json'
          },
          body: JSON.stringify(dataToSave)
        }
      )
      .then(res => res.json())
      .then(
        (result) => {
          alert('You saved your score');
        },
        // Note: it's important to handle errors here
        (error) => {
          alert('Bad API call');
          console.log(error);
        }
      )
    }
  }
}

```

#### **Precum si apelul functiei:**

```

//call API when the player wins a game, move this where it's needed
sendScoreToAPI();

```

## **2.3. Baze de date, abordare, tabele, optimizari**

Intrucit aplicatia in momentul actual nu a atins aceasta etapa, voi defini punctul de pornire al partii care tine de baze de date:

- Avem nevoie minimal de o tabela unde sa inregistram detalii despre jucatorii care incep un joc.
- De asemenea, odata pornit un joc, am mai avea nevoie de o tabela unde sa inregistram detalii despre sesiunea de joc: Id Sesiune Joc, Id Jucator, Start Time Game, End Time Game, Scor, etc.
- In Final va trebui sa construim un Leader Board, pe baza informatiilor din tabele, in care sa afisam top-ul Jucatorilor cu cele mai bune scor-uri



## Concluzii

Mediul de lucru ReactJS este un mediu accesibil, si, in opinia mea, cel mai bun atunci când vine vorba de dezvoltat aplicații web sau site-uri. Cu ajutorul sfatului din partea reprezentantilor companiei am învățat sa folosesc acest program si cele mai bune tool-uri pentru a realiza provizoriu ceea ce imi propusesem la inceputul acestui stagiu de practica.

Proiectul la care am lucrat nu este de complexitate mare, sau, insa pentru mine a reprezentat o noutate.

Consider ca am reusit sa dezvolt intr-un timp scurt si sa realizez o buna parte din ce mi-am propus pentru acest stagiu de practică. Ca rezolutie pentru viitor, imi doresc sa reftific anumite parti din aplicatie care ar putea ajuta sa duca acest puzzle interactiv mai departe, sa reusesc sa o stochez intr-o baza de date pentru a functiona asa cum trebuie.

## Bibliografie

<https://reactjs.org/tutorial/tutorial.html>

<https://nodejs.org/en/>

<https://www.npmjs.com/>

<https://www.bestjobs.eu/ro/firme/ubisoft-romania>