Caiet de practică

An II

Numele firmei: ECO SOLUTII ONLINE SRL -TechBolide, Bucuresti

Reprezentant practica: Pirvu Stefan

Student: Brânzea Ana-Maria

Grupa:40317A

1. Prezentarea generală a stagiului de practică

*Despre companie*

TechBolide se prezintă ca un dezvoltator specializat în sisteme software și tehnologii IoT (Internet of Things), focalizat pe furnizarea de soluții integrate pentru Waste Management și Automatizarea Proceselor de Afaceri. Prin implementarea de sisteme software complexe, această companie creează conexiuni între persoane, echipamente și fluxuri de lucru, în vederea obținerii unor performanțe superioare pentru clienți în fiecare zi.

TechBolide este un producător european. A fost lansat din pasiunea câtorva antreprenori care au dorit să ofere cele mai bune soluții și flexibilitate.

Compania are numeroase divizii, fiecare având scopul său specific:

* MANAGEMENTUL DESEURILOR

Soluții SMART IoT și software concepute în conformitate cu principiile economiei circulare pentru a atinge obiectivele UE pentru atingerea tintelor de colectare si reciclare.

Produse: Cantar automat industrial, cantar sasiu, cantar pentru stivuitor, sisteme de colectare separata, senzori de volum, sisteme RFID, aplicatie mobila, management activelor, solutii software dedicate.

* AUTOMATIZARI

Produse:

* Smart Production – Software pentru Gestionarea Productiei si Managementul Stocurilor
* WMS (Warehouse Management System)
* SCM (Supply Chain Management)
* Automatizari Procese de Business – solutii personalizate la cerere, care raspund 100% cerintelor clientului
* HARDWARE

Ofera solutii hardware integrate si complete, accesorii personalizate.

Produse:

* Sisteme de cantarire deseuri pentru autogunoiere (pe lift si pe sasiu)
* Senzori de volum
* Eco Insule (pubele inteligente pentru colectarea selectiva a deseurilor)
* Sisteme RFID pentru managementul deseurilor
* SOFTWARE

Oferă o gama larga de solutii software și aplicații ERP pentru a atinge potențialul maxim în afacerea clientului.

Produse:

* Software Gestionare Productie
* Warehouse Management System
* Supply Chain Management System
* Dezvoltare software la comanda

Perioada de practică s-a desfășurat în departamentul de dezvoltare software al companiei.

1. Descrierea activităţilor desfăşurate în stagiul de practică

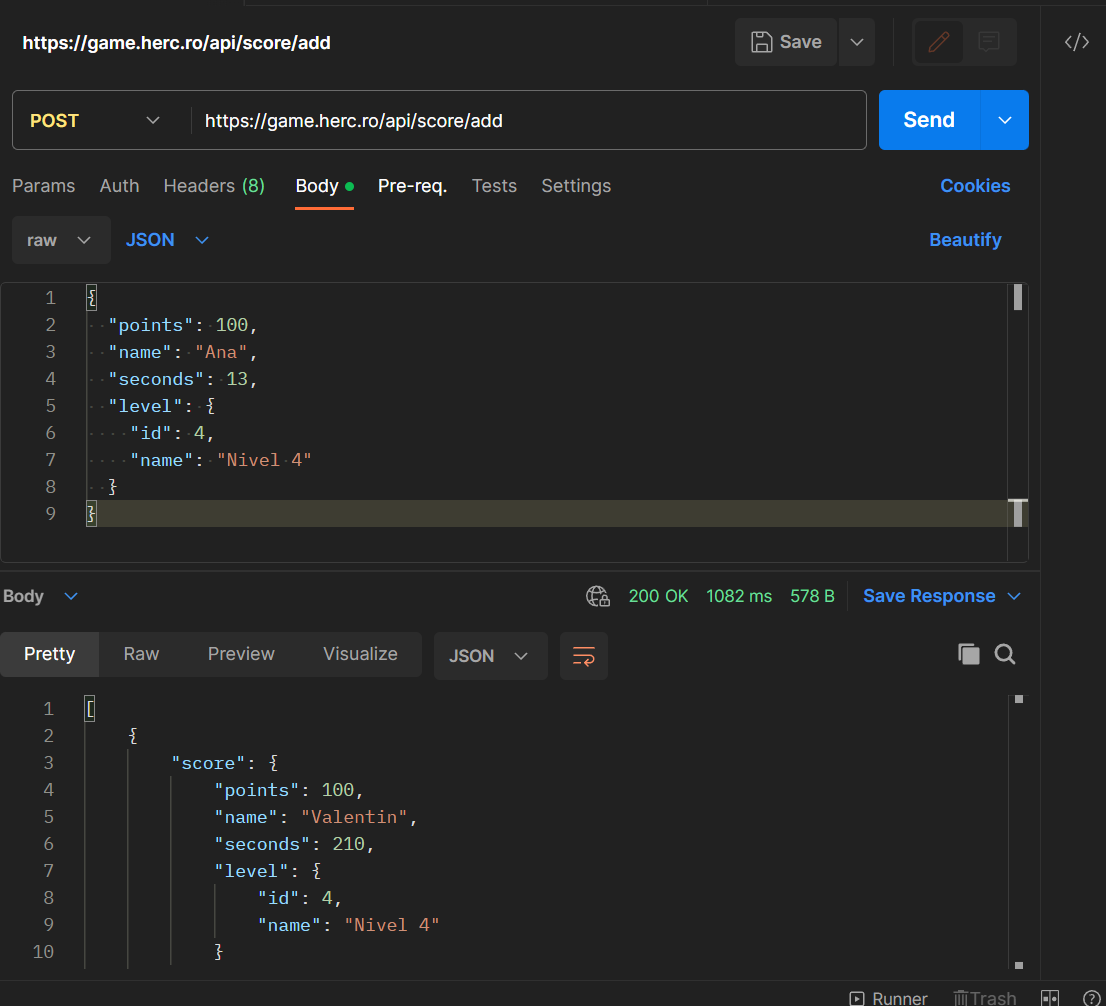
Stagiul de practică s-a desfășurat în cadrul departamentului de dezvoltare software al companiei, având ca punct de plecare un proiect inițiat de îndrumătorul nostru. Proiectul consta în dezvoltarea unui joc de tip "Candy Crush" folosind tehnologia Angular. Acest proces de dezvoltare a fost structurat în patru etape principale:

* Etapa de Analiză: În prima etapă, am efectuat o analiză detaliată a cerințelor proiectului. Am identificat funcționalitățile esențiale ale jocului și am stabilit cerințele tehnice și de design.
* Etapa de Proiectare: În această fază, am elaborat design-ul și arhitectura jocului. Am creat interfețe utilizator și am stabilit structura de bază a aplicației pentru a asigura o implementare eficientă.
* Etapa de Implementare: Următoarea etapă a constat în dezvoltarea efectivă a jocului. Am scris codul sursă în Angular, implementând funcționalitățile, interacțiunile și logica de joc necesare.
* Etapa de Test: După ce am finalizat dezvoltarea, am efectuat o serie de teste pentru a verifica corectitudinea și eficiența jocului. Am identificat și am remediat eventualele erori sau bug-uri pentru a asigura o experiență de joc fără probleme.

În prima etapă a proiectului, echipa noastră a efectuat o analiză amănunțită pentru a înțelege și pentru a se documenta cu privire la Angular, un framework de dezvoltare web. Această analiză ne-a permis să ne familiarizăm cu sintaxa Angular și să înțelegem cum să îl folosim în cadrul proiectului nostru.

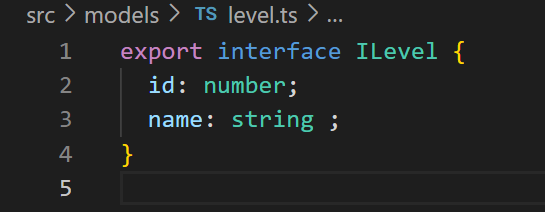
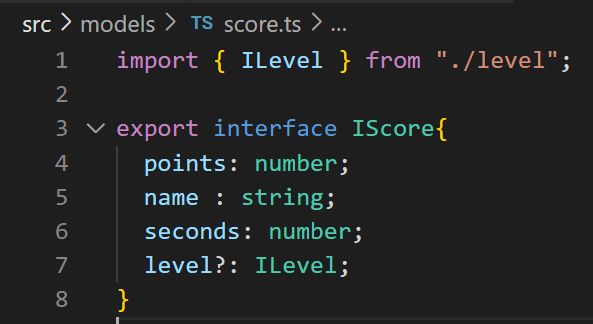
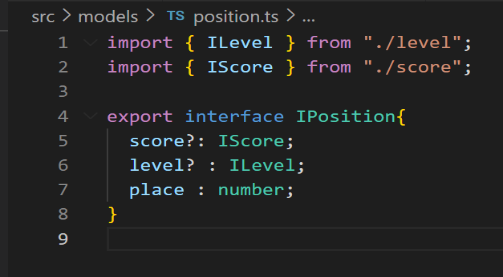
În paralel cu învățarea Angular, am analizat detaliat cerințele proiectului. Acesta viza crearea unui joc de tip Candy Crush, care să includă o tablă de joc de dimensiune 8x8. Pe această tablă, urmau să fie generate aleatoriu cinci tipuri diferite de elemente, asemănătoare cu cele din jocul Candy Crush. În plus, am identificat funcțiile și caracteristicile esențiale care trebuiau implementate pentru a face jocul funcțional, inclusiv sistemul de punctare și un scoreboard care să afișeze progresul jucătorilor pe nivele.

Pentru a ne pregăti pentru interacțiunea cu API-urile necesare proiectului, am explorat și folosit aplicația Postman. Cu ajutorul acesteia, am testat metodele POST și GET pentru API-uri. Pentru solicitările POST către link-ul https://game.herc.ro/api/score/add, am creat obiecte JSON cu informații precum punctele obținute, numele jucătorului, timpul în secunde și nivelul atins. Acest proces ne-a ajutat să ne asigurăm că putem înregistra și transmite corect informațiile necesare către serverul jocului.



API (Interfață de Programare a Aplicației) este un set de reguli și protocoale care permit aplicațiilor sau serviciilor să comunice și să interacționeze între ele. API-urile permit dezvoltatorilor să acceseze funcționalități sau date dintr-o aplicație sau serviciu și să le utilizeze în cadrul altor aplicații sau servicii fără a cunoaște detaliile interne ale implementării respectivei aplicații sau servicii.

Inițial, am definit interfețele TypeScript pentru niveluri, scoruri și poziții, astfel încât să putem utiliza API-urile corespunzătoare. Mai jos puteți vedea modul în care au fost definite aceste interfețe:



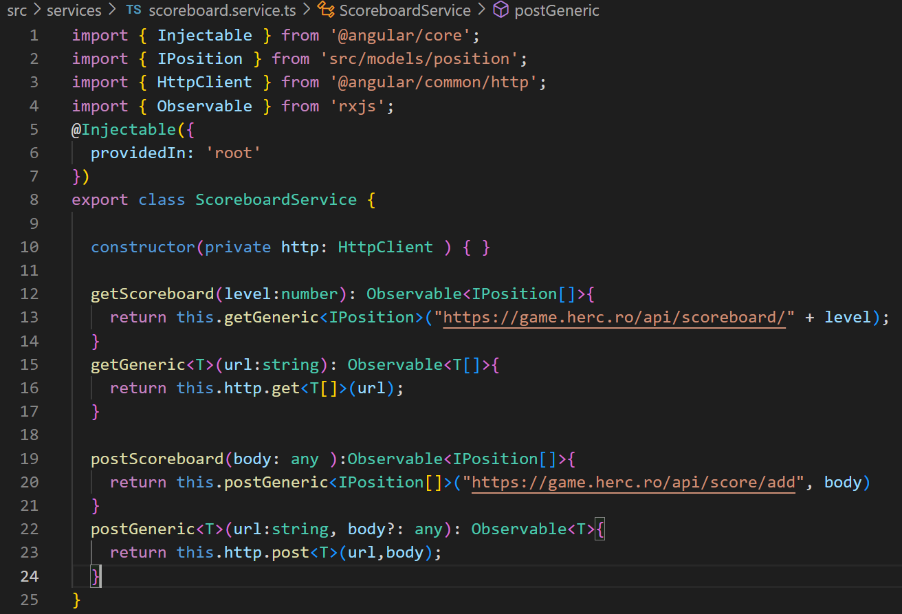
Aceste interfețe sunt folosite pentru a structura și formata datele pe care le trimitem sau le primim de la API-urile proiectului. Ele definesc structura și tipurile de date care sunt utilizate în comunicația cu API-urile, ceea ce face ca dezvoltarea și manipularea datelor să fie mai ușoară și mai sigură.

**Scoreboard**

Pentru crearea panoului de scoruri în aplicația Angular, am folosit următoarele componente:

* score-board.component.ts: Acest fișier reprezintă componenta Angular dedicată panoului de scoruri. În cadrul acestei componente, am definit variabile și servicii necesare pentru obținerea și afișarea scorurilor. În constructor, am injectat serviciul ScoreboardService pentru a facilita comunicarea cu API-ul panoului de scoruri. Aici am efectuat o solicitare GET către serviciu pentru a obține scorurile și le-am stocat în variabila tabela. Această componentă Angular este legată la un șablon HTML și un fișier CSS pentru a afișa datele corespunzătoare.



* score-board.component.spec.ts: Acest fișier conține teste unitare pentru ScoreBoardComponent, care este componenta Angular asociată panoului de scoruri. Testele au fost create pentru a confirma că componenta este inițializată corect și funcționează așa cum este preconizat.
* scoreboard.service.ts: Acest fișier conține implementarea ScoreboardService, care este un serviciu Angular. Serviciul utilizează HttpClient pentru a face cereri către API-uri pentru a obține și a trimite date legate de scoruri. Aceasta include două funcții principale: getScoreboard pentru a obține tabelul de scoruri pentru un anumit nivel și postScoreboard pentru a trimite un nou scor la server.

Pentru a conferi un aspect vizual atrăgător și plăcut pentru utilizatori, am efectuat și procesul de stilizare grafică a tabelului de scoruri folosind platforma FIGMA. FIGMA este o aplicație de design și prototipare online care permite proiectarea interfețelor grafice pentru aplicații și site-uri web. Acest instrument oferă dezvoltatorilor și designerilor o platformă pentru a crea și colabora asupra aspectului și comportamentului unei aplicații înainte de implementarea efectivă.

În cadrul proiectului nostru, am folosit FIGMA pentru a defini aspectul vizual al scoreboard-ului, inclusiv paleta de culori, fonturile utilizate și dispunerea generală a elementelor. Am avut posibilitatea să experimentăm și să ajustăm aspectul interfeței pentru a ne asigura că oferă o experiență plăcută utilizatorilor.

În ceea ce privește implementarea pe HTML, scoreboard-ul este structurat ca o tabelă cu două rânduri distincte. Primul rând este rezervat antetului tabelului, care conține informații generale despre fiecare coloană din tabel, cum ar fi "Jucător," "Puncte," "Secunde" și "Nivel." Rândurile ulterioare sunt folosite pentru a afișa datele efective ale scorurilor.

Pentru a afișa datele din variabila tabela în HTML, am folosit o directivă Angular numită \*ngFor. Această directivă permite iterarea prin fiecare înregistrare din tabela și afișarea lor într-un rând separat al tabelului. În felul acesta, pentru fiecare înregistrare, sunt create patru coloane pentru a afișa numele jucătorului, punctele obținute, secundele jucate și nivelul atins.

Această abordare face ca afișarea scorurilor să fie dinamică, astfel încât, pe măsură ce noi înregistrări sunt adăugate în tabela, ele vor fi afișate automat în tabelul de scoruri al aplicației. Astfel, utilizatorii pot vedea întotdeauna cele mai recente date și scoruri în timp real.

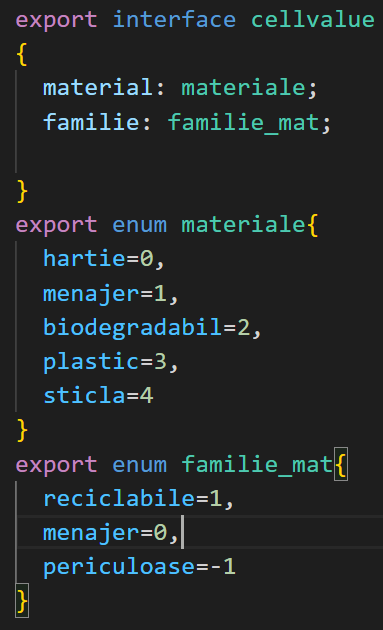


Aspectul scoreboard-ului final:



**Game**

Am creat mai multe componente și modele pentru jocul în sine. Printre acestea, am definit componenta board-cell, componenta game, și modelul cellvalue.ts. Modelul cellvalue.ts conține definiții pentru diverse atribute ale unei celule din joc.

 Modelul cellvalue.ts: Acest model definește structura și caracteristicile unei celule din joc. El conține o interfață numită cellvalue care specifică atribute precum "material" și "familie."

* "Material" este un atribut care indică tipul de material din care este făcută celula, iar acesta poate fi unul din valorile definite în enumerarea materiale. De exemplu, o celulă poate fi făcută din "plastic" sau "sticlă."
* "Familie" este un alt atribut care descrie familia de material a celulei și poate fi unul din valorile definite în enumerarea familie\_mat. Această informație poate fi folosită pentru a determina dacă celula este reciclabilă, menajeră sau periculoasă, în funcție de regulile jocului sau de logica specifică a aplicației.

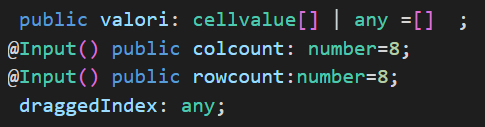
Această componentă, denumită BoardCellComponent, are rolul de a afișa celulele jocului în funcție de imaginile corespunzătoare. Componenta primește date despre caracteristicile unei celule, inclusiv tipul de material din care este făcută celula, și folosește aceste date pentru a alege imaginea potrivită pentru afișare:

* @Input() public val: cellvalue | undefined;: Această proprietate este decorată cu @Input(), ceea ce înseamnă că poate primi date de la componenta părinte. Proprietatea val reprezintă caracteristicile unei celule din joc și poate să fie o valoare de tip cellvalue sau undefined.
* getImageUrl(): Această metodă este responsabilă pentru alegerea URL-ului imaginii care trebuie afișată pentru celula respectivă. În funcție de val, se alege o imagine corespunzătoare materialului din care este făcută celula sau se alege una aleatoriu.
* Imaginile sunt stocate în folderul /assets/images/ și sunt numite în funcție de tipurile de materiale din joc. De exemplu, există imagini pentru "biodegradabile," "hartie," "sticla," "menajere," și "plastic." Această componentă utilizează aceste imagini pentru a crea URL-ul corespunzător și a afișa imaginea potrivită pentru celula curentă.

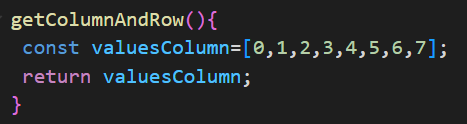


GameComponent reprezintă partea esențială a jocului și conține logica principală care îl face să funcționeze. Această secțiune s-a dovedit a fi cea mai challenging și solicitantă din întreaga perioadă de practică. Mai jos, voi oferi o explicație detaliată a ceea ce se întâmplă în această componentă:

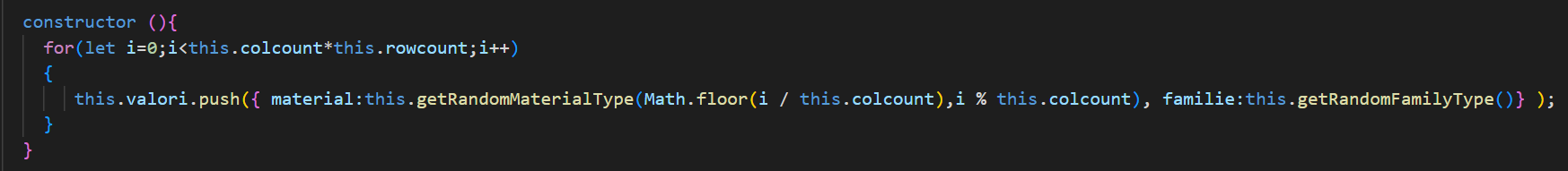
* public valori: cellvalue[] | any =[];: Această proprietate stochează caracteristicile celulelor din joc sub formă de obiecte cellvalue. Practic, reprezintă tabla de joc și este o matrice bidimensională care păstrează starea actuală a jocului.
* @Input() public colcount: number = 8; și @Input() public rowcount: number = 8;: Aceste proprietăți permit specificarea numărului de coloane și rânduri ale tablei de joc și pot fi configurate din exterior. În mod implicit, tabla are 8 coloane și 8 rânduri.



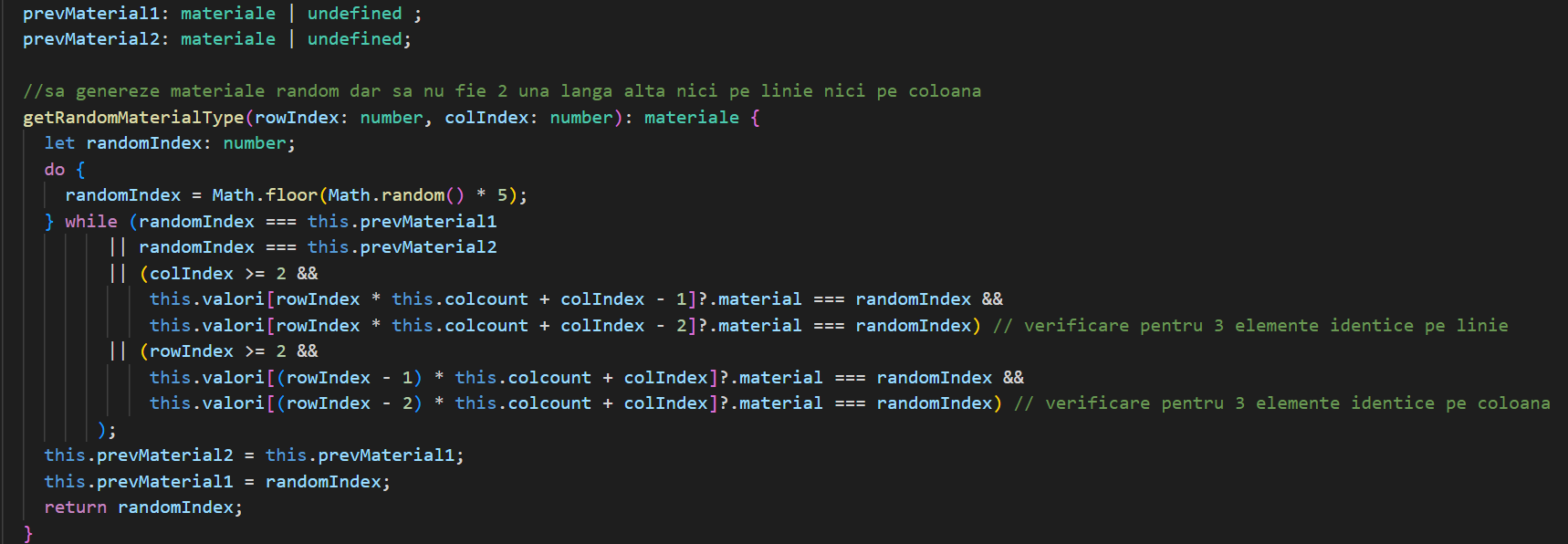
* getColumnAndRow(): Această metodă returnează un array cu valorile posibile ale indicilor coloanelor, utilă pentru afișarea corectă a coloanelor în interfața de joc.



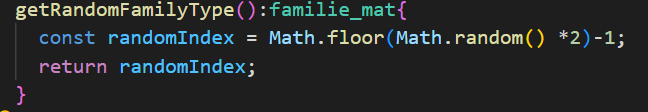
* Constructorul: În constructor, se inițializează tabla de joc (valori) prin umplerea sa cu caracteristici ale celulelor generate aleatoriu. Fiecare celulă primește un material și o familie aleatoare, astfel creând o stare inițială a jocului.



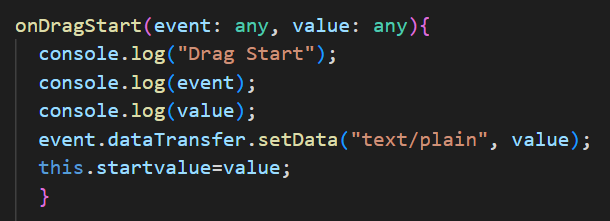
* getRandomMaterialType(rowIndex: number, colIndex: number): materiale: Această metodă generează aleatoriu un tip de material pentru o celulă specifică în funcție de poziția sa pe tablă. Se asigură că două celule consecutive pe aceeași linie sau coloană nu au același material.



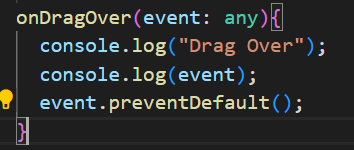
* getRandomFamilyType(): familie\_mat: Această metodă generează aleatoriu o familie de material pentru o celulă. Fiecare celulă poate face parte dintr-una din cele trei familii posibile: "reciclabile," "menajere," sau "periculoase."



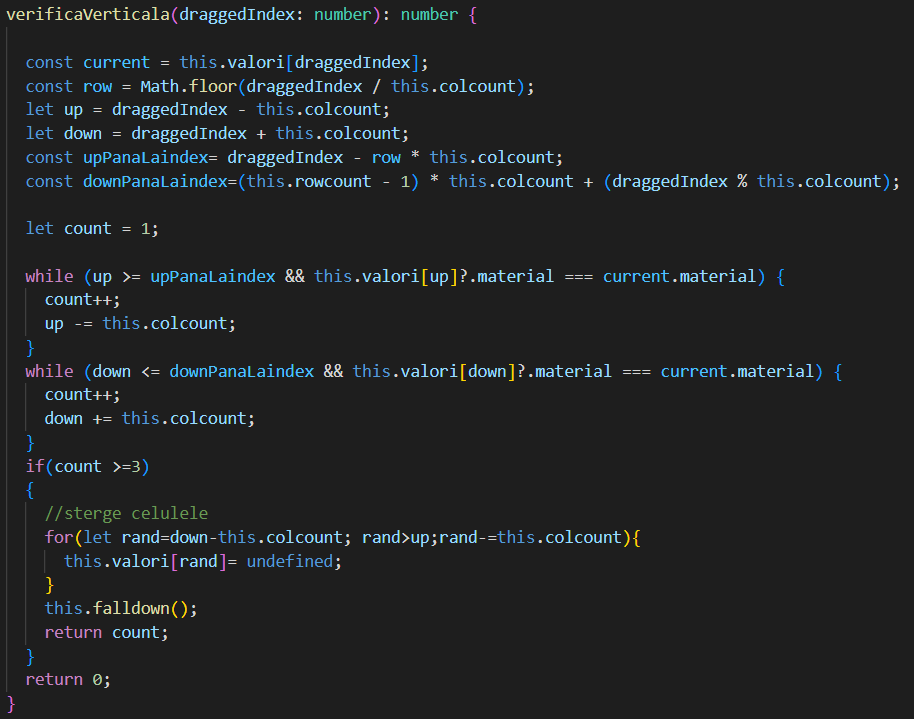
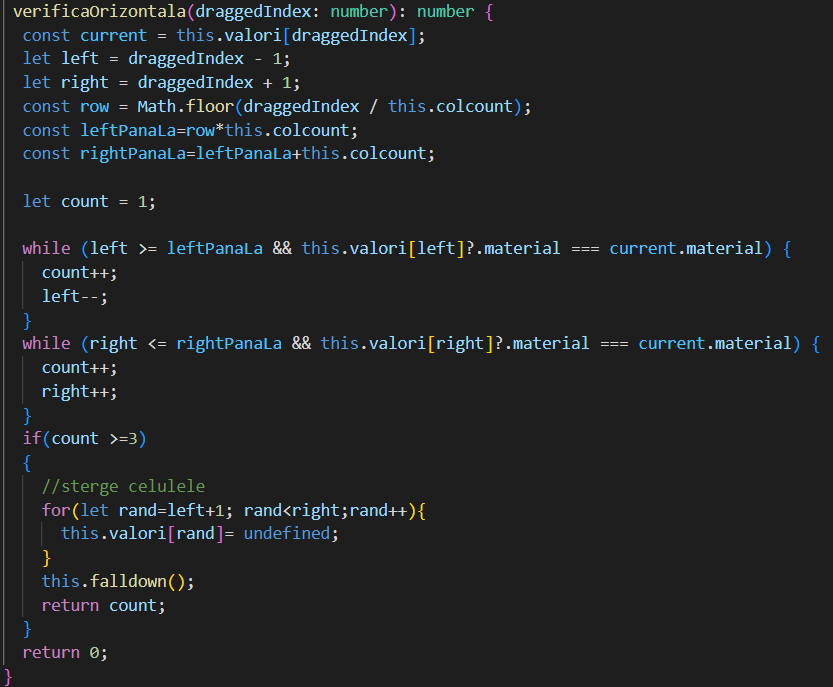
* onDragStart(event: any, value: any): Această metodă este apelată atunci când un utilizator începe să tragă o celulă. Ea setează datele care trebuie transferate în timpul tragerii pentru a identifica celula și începutul tragerii.



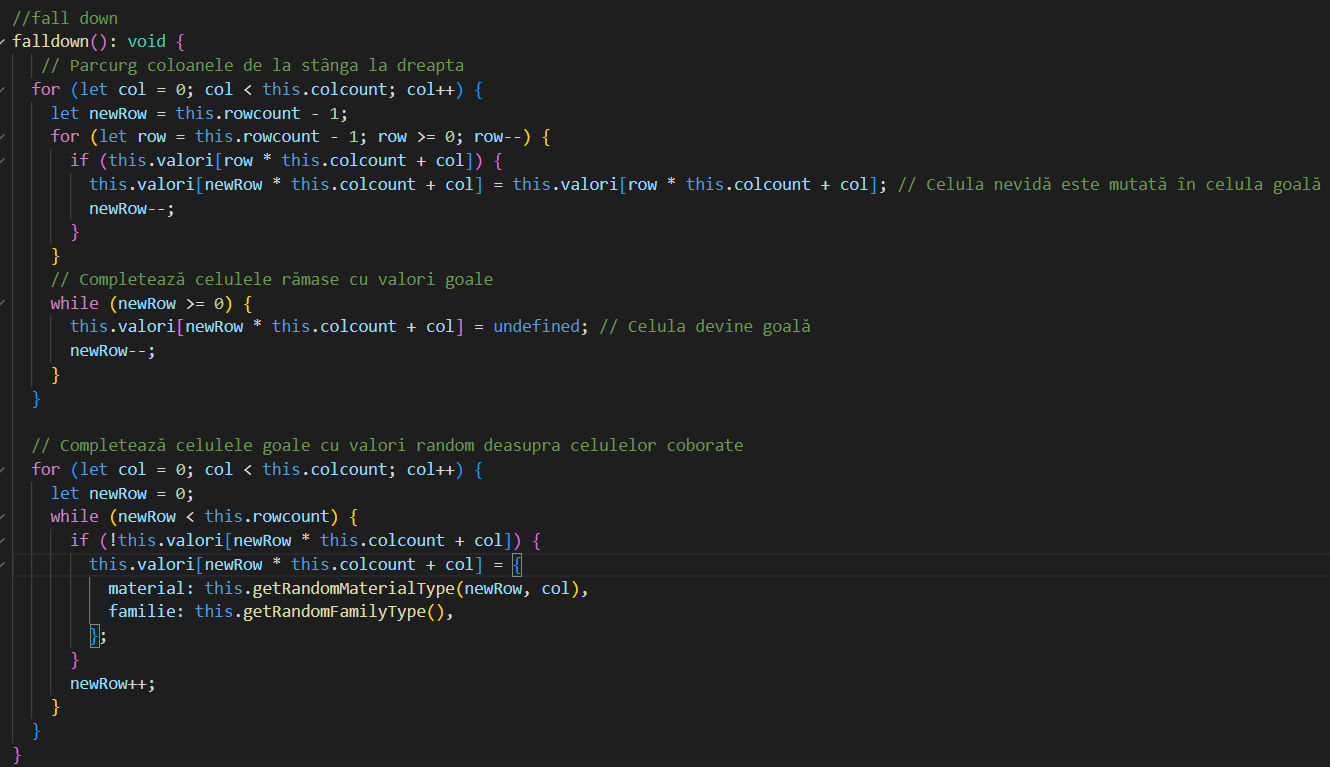
* onDragOver(event: any): Această metodă este apelată atunci când o celulă este trasă peste alta în timpul tragerii. Ea previne comportamentul implicit al browserului pentru a permite continuarea tragerii.



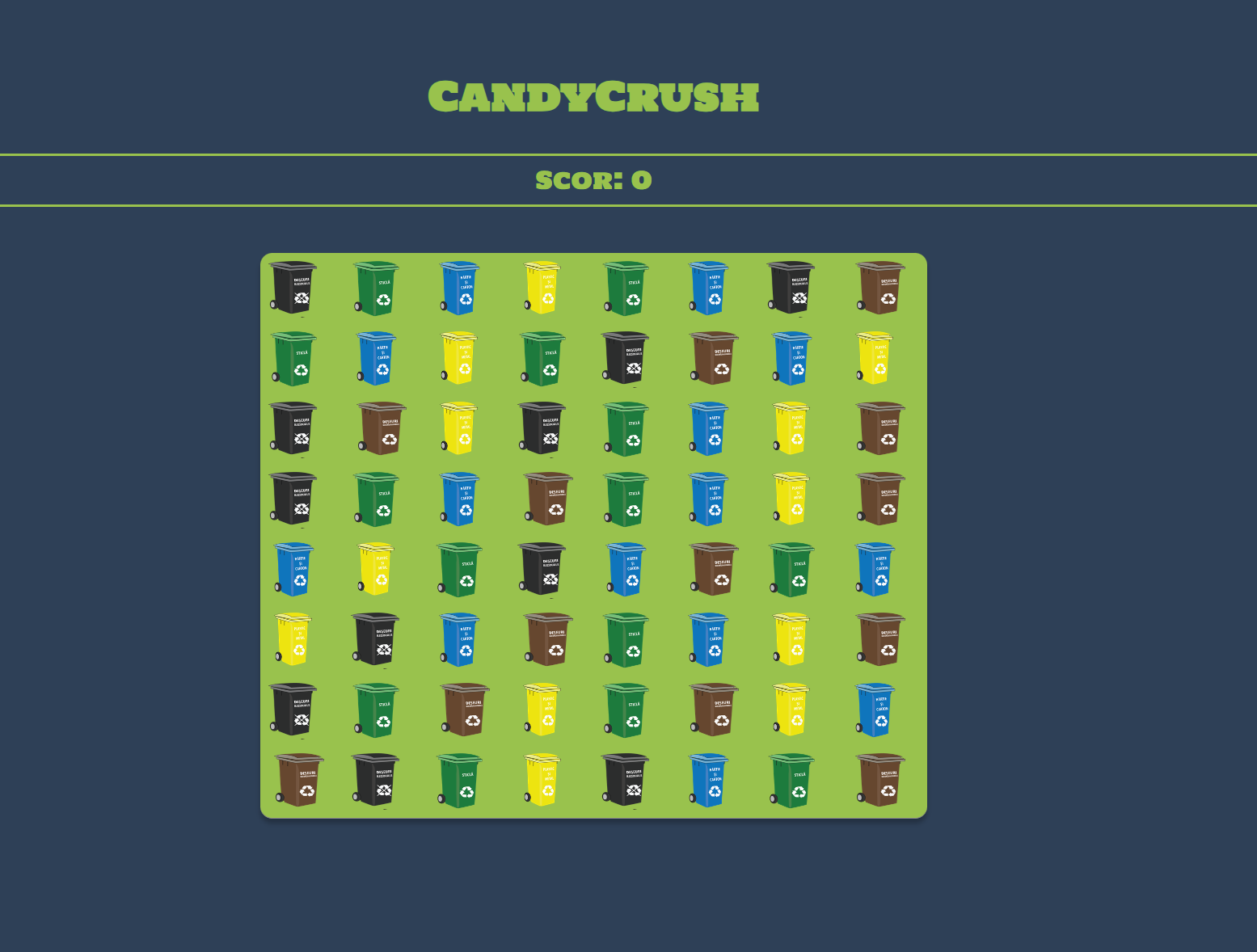
* onDrop(event: any, index: number): Această metodă este apelată atunci când o celulă este eliberată în timpul tragerii. Ea realizează schimbul de poziții între celule dacă tragerile sunt valide și verifică dacă se formează grupuri de cel puțin 3 elemente de același tip. Dacă se formează grupuri, acestea sunt eliminate, iar celulele de deasupra lor cad în locurile goale.
* verificaScor(draggedIndex: number): number: Această metodă verifică dacă se formează grupuri de cel puțin 3 elemente de același tip în orizontală sau verticală și calculează scorul. De asemenea, gestionează eliminarea celulelor și căderea celorlalte pentru a forma noi grupuri.
* verificaOrizontala(draggedIndex: number): number și verificaVerticala(draggedIndex: number): Aceste metode verifică dacă se formează grupuri de cel puțin 3 elemente de același tip în orizontală, respectiv verticală, în jurul unei celule. Ele returnează numărul de elemente în grup, iar grupurile eliminate contribuie la scorul jucătorului.



* falldown(): Această metodă se ocupă de căderea celulelor din partea de sus a tablei atunci când grupurile de cel puțin 3 elemente identice sunt eliminate. Pe lângă căderea celulelor, ea se ocupă și de generarea de noi celule cu valori aleatorii pentru a umple spațiile goale create. Această abordare repetată asigură fluxul continuu al jocului, unde grupurile sunt formate, eliminate și înlocuite cu altele, oferind jucătorului șansa de a obține un scor cât mai mare.



Interfata finala a jocului:



1. Concluzii şi propuneri

În urma perioadei de practică, pot afirma cu încredere că am obținut rezultate semnificative în procesul meu de învățare. Această experiență mi-a dezvoltat competențele și cunoștințele în domeniul dezvoltării software și al utilizării tehnologiilor Angular, API-urilor și gestionării proiectelor.

Principalele aprecieri privind rezultatele învățării sunt:

* Cunoașterea Angular: Am avut oportunitatea de a lucra în profunzime cu framework-ul Angular, înțelegând sintaxa și funcționalitățile acestuia. Am dobândit cunoștințe solide în dezvoltarea de aplicații web folosind acest framework.
* Interacțiunea cu API-uri: Prin intermediul proiectului meu de practică, am dobândit abilități în comunicarea cu API-uri externe și în gestionarea cererilor HTTP. Am învățat cum să integrez date din surse externe în aplicații web și cum să manipulez aceste date în mod eficient.
* Proiectare și dezvoltare software: Am avut ocazia de a participa la toate etapele dezvoltării software, de la analiza inițială la proiectare, implementare și testare. Această experiență m-a ajutat să înțeleg importanța planificării și structurării corecte a proiectelor software.
* Colaborarea în echipă: Practica mi-a oferit o perspectivă valoroasă asupra lucrului în echipă. Am învățat să comunic eficient cu colegii de echipă, să împărtășesc idei și să contribui la rezolvarea problemelor în mod colectiv.

În ceea ce privește dezvoltarea profesională personală, această experiență de practică mi-a întărit pasiunea pentru dezvoltarea software și m-a motivat să doresc să rămân mereu conectată la noutățile tehnologice și să mă implic în proiecte variate. Mai mult, am dobândit abilități îmbunătățite de gestionare a sarcinilor și de prioritizare a activităților, într-un mediu de lucru dinamic.

În concluzie, perioada de practică a fost o oportunitate valoroasă de învățare și dezvoltare profesională. Am atins obiectivele propuse și am acumulat experiență relevantă în domeniul meu de interes. Abia aștept să continui să explorez și să aduc contribuții în industria software în viitorul meu profesional, dar sunt, de asemenea, deschisă să explorez și să experimentez și alte domenii din sfera tehnologiei informației.

1. Tabel centralizator al activităților desfășurate

|  |  |  |
| --- | --- | --- |
| Data | Descrierea activității pe scurt | Limbaje de programare/ Produse  software/ Tehnologii studiate |
| 24.05.2023 | Învățarea și aprofundarea cunoștințelor despre Angular. | Angular, Typescript, html, css |
| 25.05.2023 | Înțelegerea funcționării API-urilor. | Postman, Angular |
| 26.05.2023 | Înțelegerea cerințelor proiectului. | Angular, Typescript |
| 27.05.2023 | Implementarea API-urilor pentru sistemul de punctaj (scoreboard). | Angular,Typescript |
| 28.05.2023 | Elaborarea unei configurații grafice pentru tabloul de scoruri. | Figma, Css,html |
| 1.06.2023 | Explorarea conceptelor | Facade, Decorator, Dependency injection,Factory |
| 2.06.2023 | Procesul de elaborare al jocului Candy Crush. | Angular |
| 3.06.2023 | Elaborarea codului pentru funcționalitatea jocului. | Angular |
| 4.06.2023 | Elaborarea codului pentru funcționalitatea jocului. | Angular |
| 5.06.2023 | Elaborarea codului pentru funcționalitatea jocului. | Angular |