

Android game with progress tracking: Arkanoid

Programarea Dispozitivelor Mobile

Autor: Botejaru Ștefan-Andrei

Facultatea: Matematică-Informatică

Coordonator Științific: Conf. Dr. Bocu Răzvan

Contents

[1. Tehnologii utilizate 3](#_Toc157290305)

[1.1 Unity 3](#_Toc157290306)

[1.2 Microsoft SQL Server 3](#_Toc157290307)

[1.3 .NET Web API 3](#_Toc157290308)

[2. Structura aplicației 4](#_Toc157290309)

[2.1 Arkanoid 4](#_Toc157290310)

[2.1.1 Elementele jocului 5](#_Toc157290311)

[2.1.2 Mecanicile jocului 7](#_Toc157290312)

[2.2 Web API 10](#_Toc157290313)

[2.2.1 Structurarea aplicației server 10](#_Toc157290314)

[2.2.2 Funcționalitatea server-ului 11](#_Toc157290315)

[2.2.3 Primirea de request-uri din Unity 13](#_Toc157290316)

[3. Concluzii 14](#_Toc157290317)

# 1. Tehnologii utilizate

## 1.1 Unity

Unity este o platformă de dezvoltare de jocuri și aplicații extrem de populară în domeniul său. Prin intermediul mediului său integrat, Unity oferă dezvoltatorilor posibilitatea de a crea conținut interactiv pentru diverse platforme, inclusiv desktop și dispozitive mobile. Datorită funcționalităților avansate în domeniul graficii, fizicii și suportului pentru multiple limbaje de programare, Unity s-a impus ca o alegere de top pentru proiecte variate, de la jocuri complexe la aplicații de realitate virtuală și augmentată.

Unity nu este doar o unealtă de dezvoltare pentru jocuri; este un ecosistem complex care promovează colaborarea și încurajează dezvoltatorii să partajeze resurse și să beneficieze de o comunitate activă. Cu capacități extinse de personalizare și acces la Asset Store, unde pot fi găsite și împărtășite resurse precum modele 3D, script-uri și efecte vizuale, Unity devine o platformă comprehensivă ce susține inovația și evoluția continuă a proiectelor.

## 1.2 Microsoft SQL Server

SQL Server este o platformă de gestionare a bazelor de date relationale dezvoltată de Microsoft, cunoscută pentru fiabilitatea și performanța sa în manipularea datelor critice. Această soluție oferă funcționalități avansate pentru stocarea, interogarea și securizarea datelor, facilitând dezvoltatorilor și administratorilor de baze de date crearea și gestionarea eficientă a aplicațiilor scalabile.

Microsoft SQL Server se remarcă prin funcționalitățile sale avansate, inclusiv servicii de analiză, raportare și integrare, care extind amprenta sa în construirea de soluții complexe. Cu abilitatea de a gestiona volume mari de date și de a susține atât arhitecturi cloud, cât și on-premise, SQL Server furnizează flexibilitate și scalabilitate pentru diverse scenarii de implementare. De asemenea, se distinge prin caracteristici puternice de securitate, gestionare a performanței și recuperare a datelor, contribuind la asigurarea integrității și disponibilității datelor în cadrul aplicațiilor și sistemelor.

## 1.3 .NET Web API

.NET Web API reprezintă un framework de dezvoltare flexibil și puternic, integrat în ecosistemul .NET, ce simplifică construirea și implementarea serviciilor web conform arhitecturii REST. Conceput pentru a oferi o abordare eficientă în crearea de API-uri HTTP, .NET Web API permite dezvoltatorilor să creeze cu rapiditate servicii web scalabile și interoperabile. Având suport nativ pentru formatele de date comune, precum JSON și XML, și integrându-se strâns cu celelalte componente .NET, acesta furnizează un mediu ușor de administrat pentru dezvoltarea soluțiilor web moderne și interconectate.

Dotat cu funcționalități precum gestionarea rutelor, autentificarea, autorizarea și suportul pentru formate de date personalizate, .NET Web API facilitează dezvoltarea eficientă a serviciilor web robuste și interoperabile.

# 2. Structura aplicației

Aplicația este formată din două părți: partea de client (jocul Arkanoid), cea cu care interacționează utilizatorul, și partea de server, ce este apelată de partea de client prin intermediul unui API web de tip REST, pentru a efectua operații asupra bazei de date în care se reține progresul fiecărui utilizator.

A close-up of a web site

Description automatically generated

Figura 1. Structura aplicației

## 2.1 Arkanoid

Arkanoid este un joc arcade clasic lansat în 1986, care a devenit rapid un joc popular. Jocul constă într-o platformă care se mișcă orizontal la baza ecranului, care poate fi controlată de către jucător pentru a reflecta o minge care se mișcă pe ecran.

Scopul jocului este de a distruge toate blocurile de pe ecran folosind mingea, fără a permite ca mingea să cadă. Jocul include mai multe nivele, fiecare cu o configurație diferită de blocuri și obstacole, și oferă un set de power up-uri, cum ar fi extinderea platformei de joc sau capacitatea de a folosi mai multe mingi simultan.

Această parte a aplicației a fost dezvoltată în game engine-ul Unity și a fost realizată pentru materia ”Dezvoltarea Aplicațiilor Mobile” în anul III la această facultate, și extinsă și actualizată pentru această materie, pentru a suporta funcționalitatea suplimentară oferită de un Web API.

A screenshot of a video game

Description automatically generated

Figura 2. Jocul Arkanoid

### 2.1.1 Elementele jocului

1. **Platforma** este controlată de utilizator prin atingere, are ca scop redirecționarea mingii înspre blocuri și de a le distruge, de asemenea are scopul de a împiedica mingea de a cădea din suprafața de joc



Figura 3. Platforma

1. **Mingea** este redirecționată de contactul cu pereții, blocuri și platformă, are scopul de a distruge blocurile generând astfel scor. Dacă aceasta părăsește suprafața de joc (cade sub platformă) se pierde o viață.



Figura 4. Mingea

1. **Blocurile** au scopul de a fi distruse pentru a genera scor și, în cazul distrugerii tuturor blocurilor din suprafața de joc, se va putea parcurge la nivelul următor. Fiecare bloc are nevoie de un număr diferit de lovituri: galben = 1 lovitură, portocaliu = 2 lovituri, roșu = 3 lovituri. După fiecare lovitură, modelul blocului se modifică.

A black and orange square

Description automatically generated

Figura 5. Blocurile

1. **Buff**-uri și **Debuff**-uri



Figura 6.Buff Extindere Platformă



Figura 7. Buff Multi-Ball



Figura 8. Debuff Scurtare Platformă

### 2.1.2 Mecanicile jocului

Inițial, utilizatorului îi este prezentată o casetă de text prin intermediul căreia acesta își va introduce numele de utilizator ce va fi reținut în baza de date, alături de toate scorurile obținute de acesta. La apăsarea butonului ”Save”, se va actualiza câmpul ”Scor Maxim” din interfața utilizatorului cu scorul maxim atins de acesta, în cazul în care utilizatorul nu este nou.

A screenshot of a login screen

Description automatically generated

Figura 9. Interfața inițială

După salvarea numelui de utilizator, jocul va începe la atingerea ecranului, mingea fiind lansată din punctul de pornire aflat în mijlocul platformei. Acest lucru este realizat cu ajutorul clasei ”Input” din framework-ul Unity.



Figura 10. Utilizare clasa ”Touch”

Poziția platformei pe ecran este relativă la punctul atins de utilizator, pe axa X. Pentru a limita mișcarea platformei în spațiul ecranului, sunt folosite două limite, una pentru fiecare parte a ecranului (left, right).



Figura 11. Limitarea deplasării platformei

Cu ajutorul framework-ului furnizat de Unity, detectarea coliziunilor dintre minge și platformă/pereți/blocuri se poate realiza cu ajutorul unei funcții prestabilite. Acest lucru este posibil atunci când unui obiect îi este atribuită starea de ”Collider” din interfața editorului.



Figura 12. Funcția pentru manipulareai coliziunii

Pentru a verifica ce obiect are coliziune, se folosește un tag atribuit obiectului în editorul Unity.



La distrugerea unui bloc, scorul crește cu 10 puncte. Dacă scorul actual este mai mare decât cel maxim, scorul maxim este de asemenea actualizat în interfața de utilizator cu valoarea scorului curent. Scorul este introdus în baza de date la încheierea jocului, fie pierdut, fie câștigat (mai multe detalii în subcapitolul următor). De asemenea, la distrugerea blocurilor, acestea prezintă șansa de 10% de a invoca un buff sau un debuff care durează 10 secunde, dacă acesta este prins de utilizator.

A screen shot of a computer program

Description automatically generated

Figura 13. Funcția pentru invocarea unui buff/debuff

Generarea nivelelor se face prin citirea unui fișier, ce prezintă *n* matrici pătratice ce au scopul de a reprezenta structura blocurilor pe fiecare nivel. Valorile de 0 indică spațiile goale, iar valorile de 1, 2 și 3 reprezintă câte lovituri va rezista blocul de pe acea poziție până va fi distrus.

Jocul se termină atunci când utilizatorul termină toate nivelele existente, sau când pierde cele 3 vieți. Pentru fiecare caz, utilizatorului îi este afișată fie interfața ”Ai câștigat”, fie ”Ai pierdut”.

O viață este pierdută în cazul în care mingea iese în afara spațiului de joc, iar în acest caz, nivelul este luat de la capăt.

A computer screen shot of a program

Description automatically generated

Figura 14. Pierderea unei vieți

## 2.2 Web API

Top of Form

Efectuarea de operații CRUD asupra bazei de date aferentă jocului Arkanoid este realizată printr-un API web de tip REST, dezvoltat cu ajutorul platformei .NET.

### 2.2.1 Structurarea aplicației server

API-ul web este dezvoltat utilizând structurarea pe straturi logice:

1. **Presentation Layer**, în care se găsesc controlerele API care gestionează cererile HTTP și interacționează cu Service Layer;
2. **Service Layer**, ce conține logica serviciilor apelate de controller. Acesta interacționează cu Data Access Layer;
3. **Data Access Layer**, în care se găsesc repository-urile ce se ocupă cu interacțiunea cu baza de date, folosind Object-Relational Mapping.



Figura 15. Structurarea pe nivele

În Visual Studio, acest lucru a fost posibil prin utilizarea a trei proiecte: un proiect de tip .NET Web API, ce reprezintă stratul de prezentare, și două de tip .NET Framework Class Library, ce reprezintă stratul de servicii și cel de date.

A screenshot of a computer

Description automatically generated

Figura 16. Structurarea proiectului în Visual Studio

### 2.2.2 Funcționalitatea server-ului

În stratul de acces pentru date, legătura la baza de date SQL Server a fost folosită folosind pachetul EntityFramework, ce a generat o clasă ”context” în care se află listele tuturor entităților, ce pot fi manipulate cu ajutorul claselor cu funcționalitatea de repository. Acestea efectuează operațiile de tip CRUD asupra bazei de date, și se pot implementa query-uri complexe cu ajutorul limbajului LINQ.

A screen shot of a computer code

Description automatically generated

Figura 17. Folosirea LINQ pentru a alege scorul maxim a unui utilizator

În stratul de servicii sunt apelate funcțiile de tip CRUD din layer-ul de acces de date, cu ajutorul unei clase în care sunt implementate interfețele claselor de tip repository, pentru a nu facilita accesul direct la stratul de date din stratul de prezentare.

A screen shot of a computer program

Description automatically generated

Figura 17. Folosirea serviciilor pentru a primi scorul maxim al unui utilizator

În cele din urmă, în startul de prezentare sunt declarate request-urile de tip HTTP prin intermediul claselor de tip controller. Funcțiile din controller sunt decorate cu tipul request-ului HTTP alături de adresa web ce o apelează, iar funcționalitatea este reprezentată de procesarea apelurilor la stratul de servicii.

A screen shot of a computer code

Description automatically generated

Figura 18. Utilizarea controller-ului pentru a trimite scorul maxim al unui utilizator

### 2.2.3 Primirea de request-uri din Unity

Request-urile la server au fost trimise prin utilizarea clasei ”UnityWebRequest” cu ajutorul adreselor web declarate în decoratoarele funcțiilor din controller. Procesarea răspunsurilor a fost realizată cu ajutorul unei clase open-source pentru manipularea unui text de tip JSON.

Pentru operațiile de tip GET, request-urile sunt ușor de trimis, deobicei parametrul trimis fiind un obiect de tip string sau un număr (ID).

A computer screen with text

Description automatically generated

Figura 19. Trimiterea unui UnityWebRequest pentru preluarea scorului maxim al unui utilizator

Pentru celelalte operații este necesară trimiterea unui parametru în format JSON. Astfel, există un pas suplimentar de inițializare a unui astfel de text, deoarece toate câmpurile unui obiect trebuie incluse în acest text.

A computer screen shot of a program code

Description automatically generated

Figura 20. Trimiterea unui UnityWebRequest pentru inserarea unui scor în baza de date

# 3. Concluzii

Acest proiect a servit ca o aprofundare a cunoștințelor dobândite de-a lungul programului de licență de la această facultate și m-a ajutat să înțeleg mai bine rostul și funcționalitatea unui API de tip REST, alături de modul în care acesta comunică cu o aplicație client.