**Cuprins**

Introducere

Manual de utilizare

Scenele din Unity

Secvențe de cod

Baza de date

Eventuale îmbunătățiri

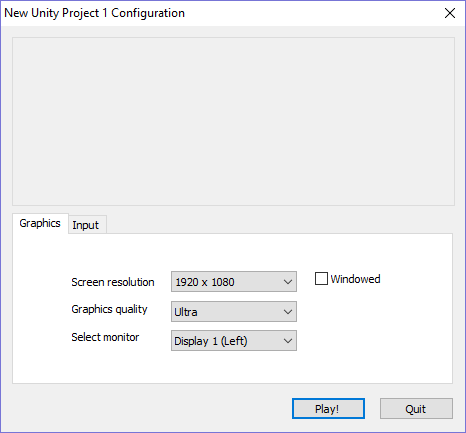
Bibliografie

**Introducere**

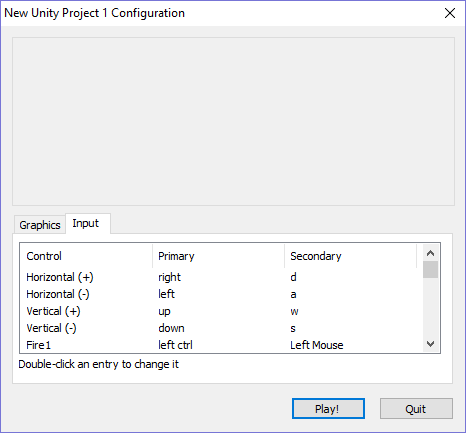
Aplicația creată constă într-un joc de tip shooter, în care jucătorul controlează un caracter și trebuie să distrugă inamicii aruncați spre el, astfel încât sa supraviețuiască pe cât de mult posibil.

Tema alesă este datorată pasiunii mele pentru jocuri video și dorința de a lucra în domeniul acestora.

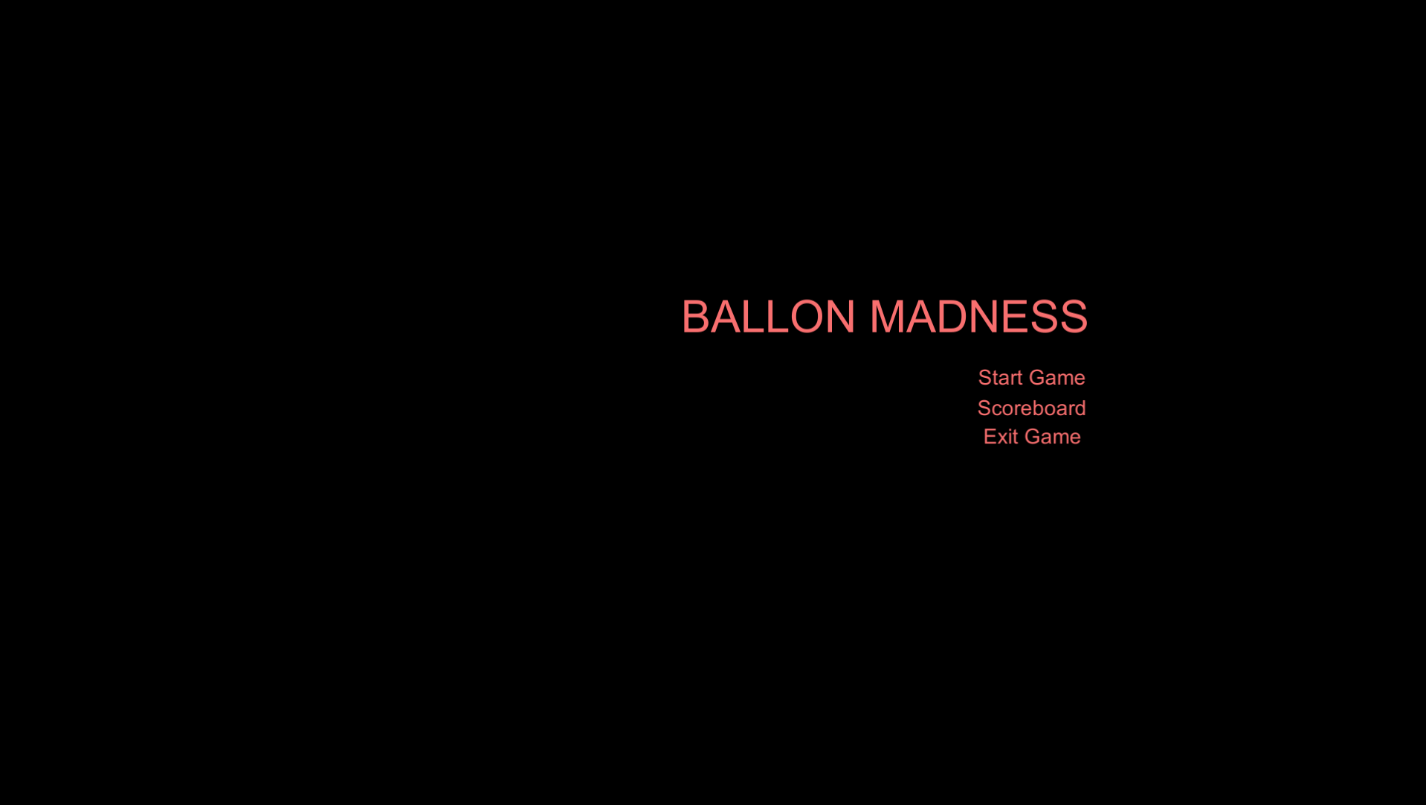
**Manual de utilizare**



După ce programul este lansat în execuție, imaginea de deasupra va apărea pe ecran. De aici, se poate selecta rezoluția în care să fie lansată aplicația (și dacă să fie în modul “fereastră”), calitatea acesteia, și monitorul pe care va fi lansată (în cazul în care utilizatorul are mai multe monitoare).

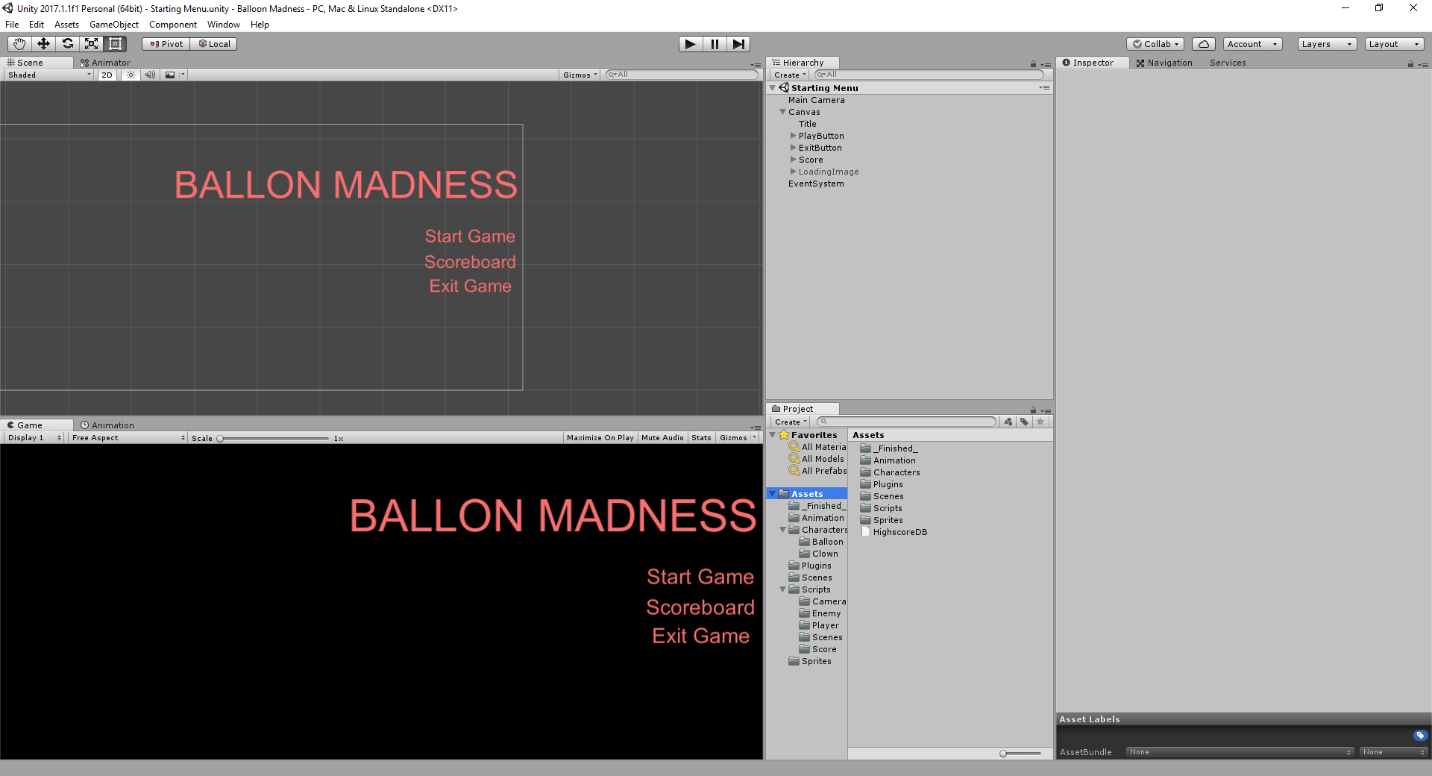


Trecând la tab-ul “Input”, utilizatorul își poate modifica controalele pe care urmează să le folosească în aplicație. Setările aplicate însă nu funcționează, deoarece codul este făcut pentru butoane specifice.

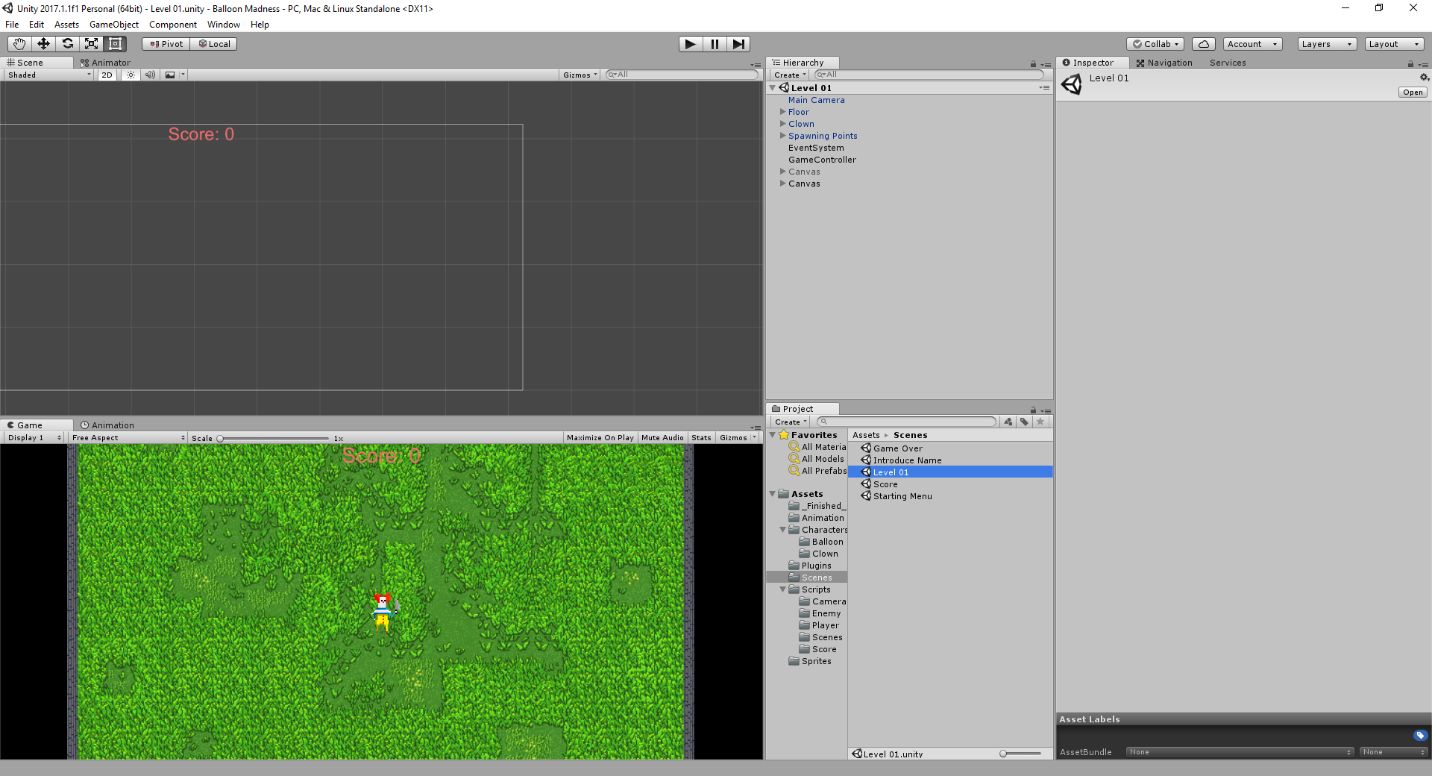


După ce aplicația este pornită, utilizatorul va fi întâmpinat de meniul principal. De aici, el poate porni jocul (“Start Game”), poate vizualiza scorurile (“Scoreboard”) sau poate ieși din aplicație (“Exit Game”).

Scene din Unity

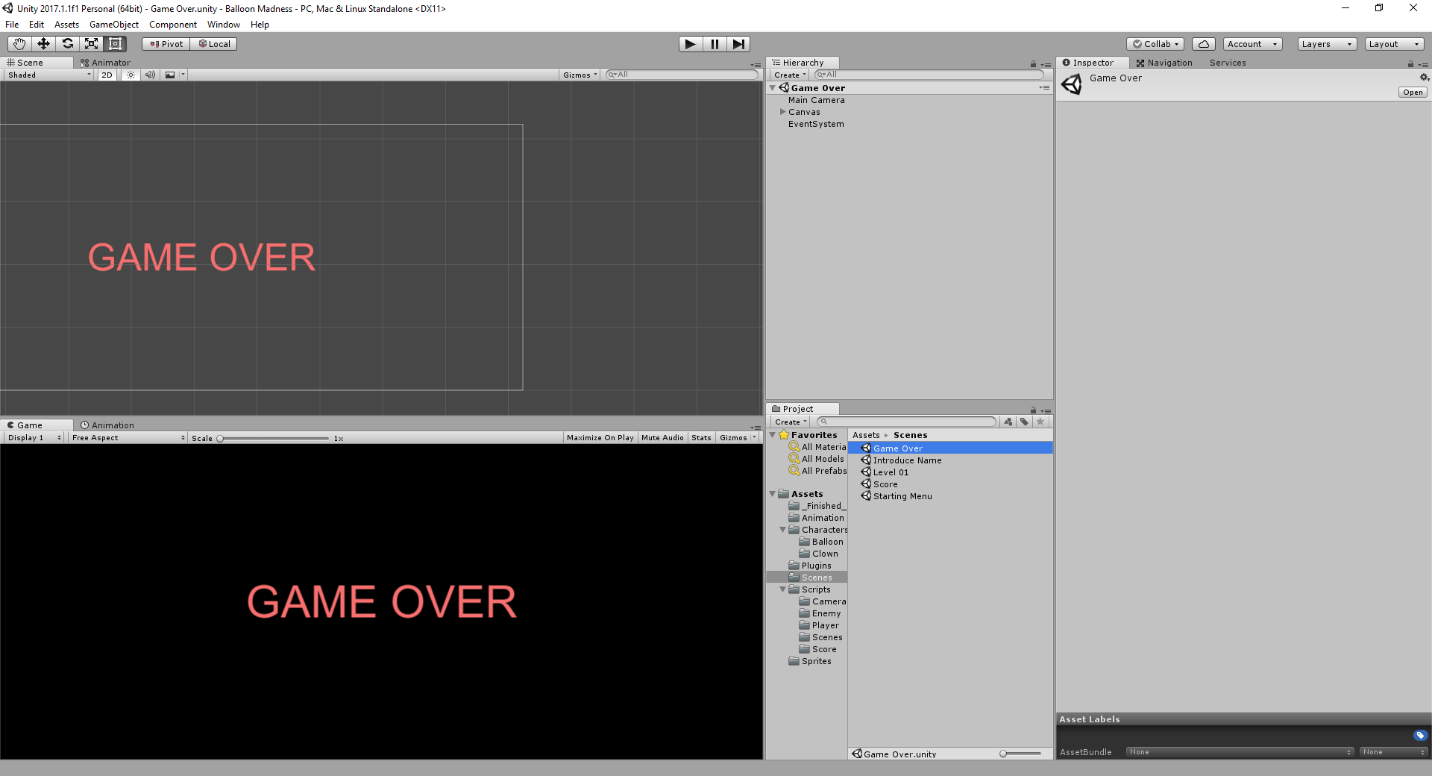


Prima scena prezentată este meniul principal, în care au fost folosite gameobjects de tip UI. Obiectul Canvas este un gameobject (ce conține o componentă de tip canvas), pe care sunt așezate de obicei toate celelalte elemente de tip UI.

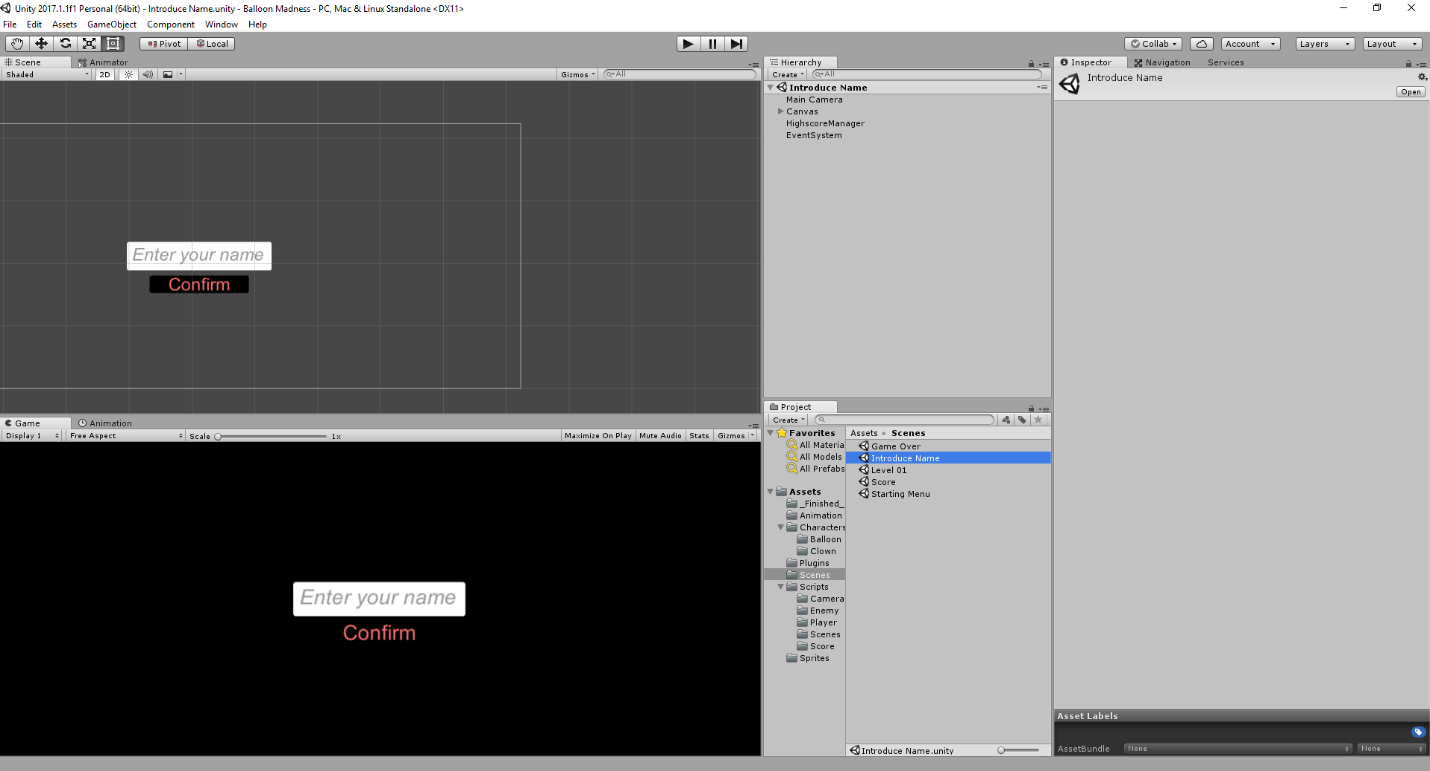


Cea de-a doua scenă este scena jocului propriu-zis. Caracterul jucătorului este un clovn, pe care îl vor urmări baloane furioase. Jucătorul trebuie sa arunce cu cuțite după baloane astfel încât să le spargă, folosind click stânga. Cuțitul va fi aruncat în direcția cursorului. Dacă jucătorul este atins de balon, jocul va fi gata și scena de Game Over va apărea pe ecran.

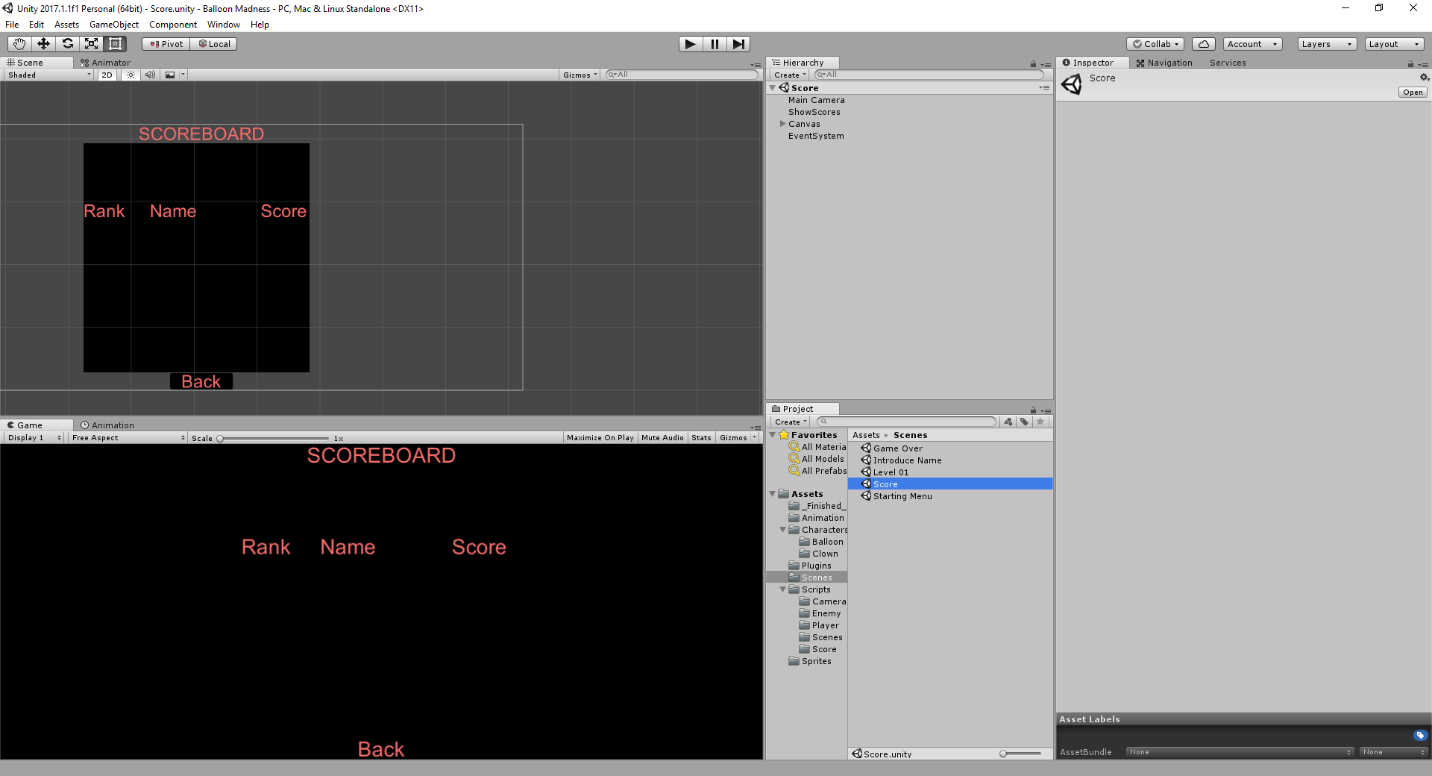
Scena conține mai multe gameobjects de tip sprite, unul pentru podea, unul pentru clovn (care conține neapărat o componentă Rigidbody 2D) și la fel și gameobject-urile care vor urma să fie create în urma spawn-ului baloanelor.



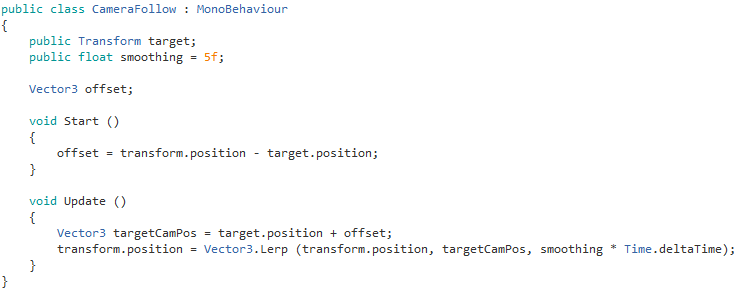
Cea de-a treia scenă este scena de Game Over, apărută în urma pierderii jucătorului. Aici jucătorul trebuie doar să dea click stânga, și scena urmatoare se va incărca.



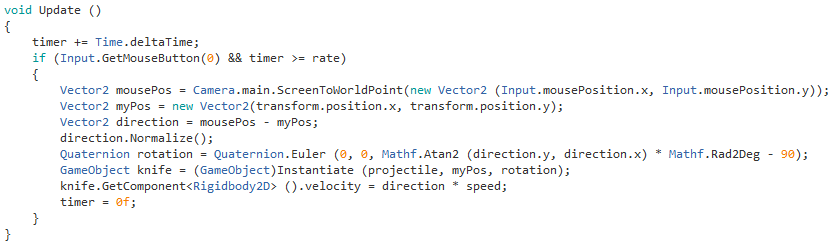
Cea de-a patra scenă este scena de după Game Over, ți anume cea de introducere a numelui. Jucătorul își introduce numele după ce a pierdut, iar astfel, apăsând pe butonul de confirmare, numele si scorul sunt salvate în baza de date. Jucătorul este apoi trimis în următoarea scenă, cea a scorurilor.

Cea de-a cincea și ultima scenă este cea a scorurilor. Aici jucătorul poate vedea scorurile reținute până acum (cele mai mari n scoruri). Apăsând pe butonul din josul ecranului “Back”, jucătorul revine la scena inițiala, cea a meniului principal.

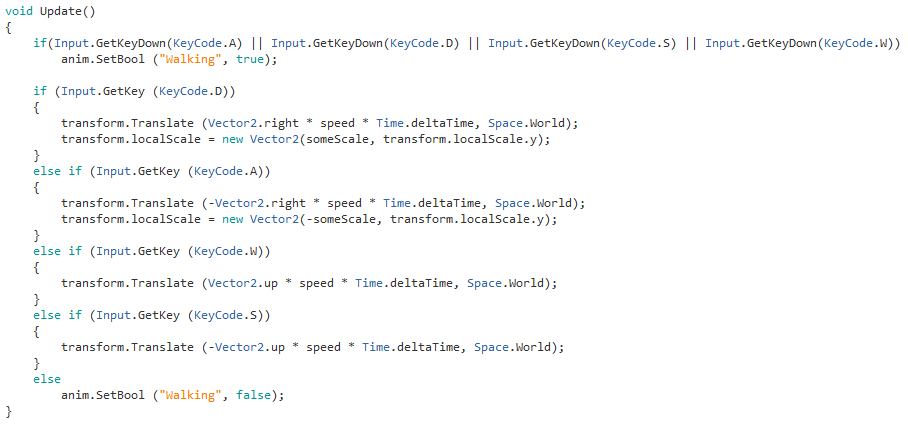
Secvențe de cod



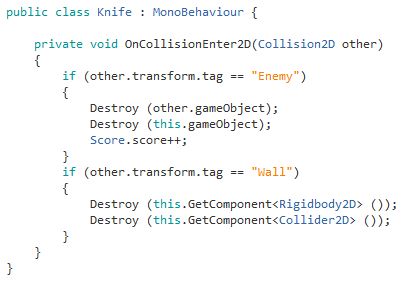
În secvența de cod de mai sus este setată camera care urmărește caracterul jucătorului (variabila de tip Transform). În funcția predefinită Start (funcție specifică Unity, în care variabilele sunt inițializate înainte de apelarea oricărei funcție de tip Update), este inițializat punctul de plecare al camerei. În funcția Update (funcție specifică Unity, care este apelată o data per frame) este actualizată poziția camerei după mișcarea caracterului, cu ajutorul unui vector de tip XYZ.



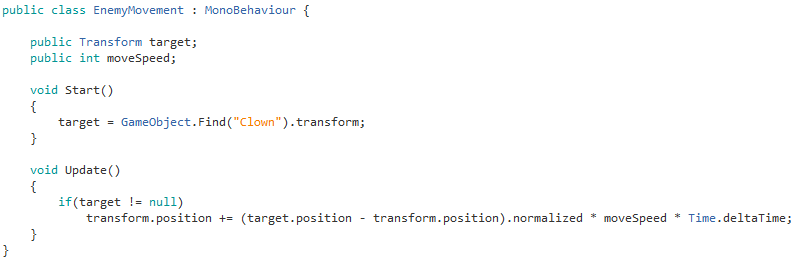
În această funcție Update sunt instanțiate obiectele proiectile aruncate de caracter. Acestea sunt setate astfel încat să fie aruncate în direcția mouse-ului odată cu click stânga. Algoritmul este însoțit de o variabilă float (numită “timer”) care reprezintă numărul de secunde între care pot fi aruncate mai multe proiectile succesive.



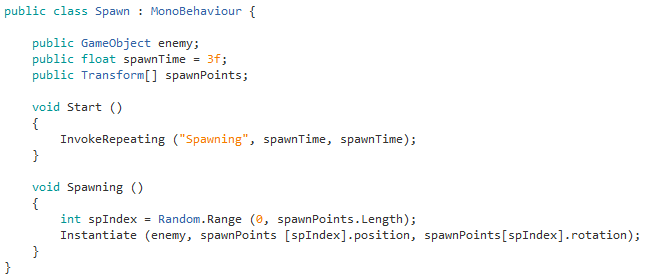
În secvența de mai sus au fost declarate controalele folosite pentru a controla caracterul. Cu ajutorului animatorului anim, jocul detectează când caracterul se află în mișcare, și activează animația pe perioada respectivă.



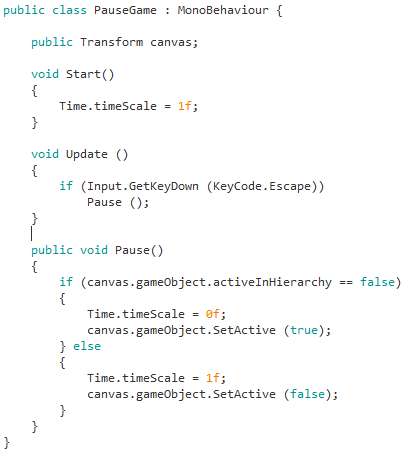
În această secvență, în funcția OnCollisionEnter2D (funcție specifică Unity, care este apelată când două obiecte de tip 2D intră în contact), atunci când proiectilul lovețte un inamic, inamicul respectiv este distrus, iar scorul jucătorului crește. În momentul în care proiectilul lovește un perete, gameobject-ul proiectilului își pierde componentele necesare astfel încat să rămână blocat în perete.



Mai sus este setată mișcarea inamicilor care urmăresc jucătorul.



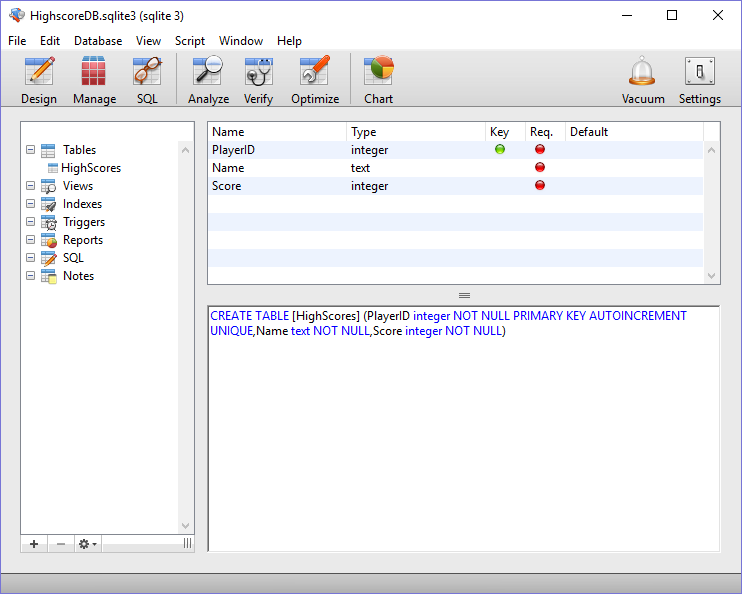
Algoritmul de mai sus se ocupă cu spawn-ul inamicilor. Funcția Spawning instanțiaza câte un gameobject inamic o dată la n secunde (spawnTime) într-unul dintre locurile de spawn create.



Acest algoritm se ocupă cu crearea meniului de pauza, deschis în timpu jocului apăsând tasta “Esc”. Odată cu apăsarea tastei, jocul din fundal va *îngheța*, iar meniul își va face apariția în mijlocul ecranului.

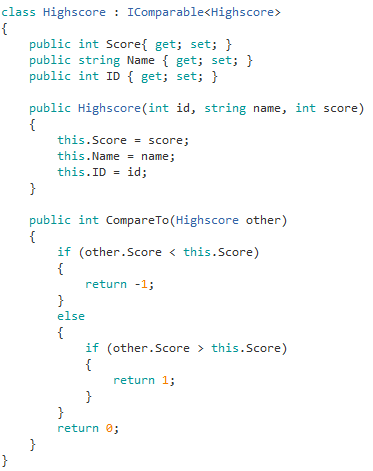
Baza de date

Programul în care a fost creată baza de date este SQLiteManager, acesta putând crea fișiere de tip sqlite3, care sunt compatibile cu Unity.





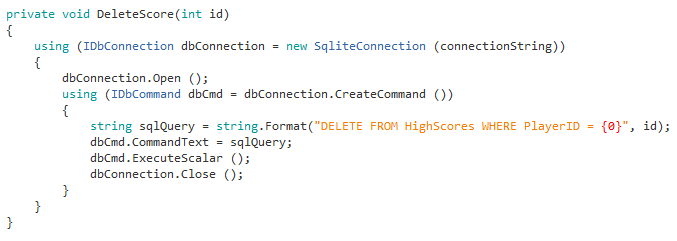
Funcția de mai sus se ocupă cu preluarea datelor din tabelul din baza de date care reține numele și scorul jucătorului. Datele preluate sunt adăugate într-o listă de tip Highscore (clasă creată în codul de mai jos) și urmează să fie utilizate în alte funcții.



Mai sus avem clasa creată pentru secvența de cod arătată anterior, inițializată folosind IComparable, ceea ce permite compararea elementelor lor. Compararea elementelor este facută efectiv cu ajutorul funcției CompareTo, care returnează 1, 0 sau -1.



Funcția InsertScore se ocupă cu inserarea scorurilor în baza de date. Dacă la inserarea unui nou scor, numărul de scoruri deja reținute depășește numărul maxim, cel mai mic scor este eliminat din listă în cazul în care noul scor este mai mare. Dacă noul scor nu va fi mai mare, atunci el nu va fi adăugat în listă.



Mai sus este reprezentată funcția de ștergere a unei date din tabel, folosită în funcția de inserare.

**Eventuale îmbunătățiri**

Aplicația ar putea fi îmbunătățite prin adăugarea unor efecte audio, care să sporească farmecul jocului.

O altă îmbunătățire ar putea fi creșterea nivelului de dificultate. Prin asta mă refer la reducerea intervalului în care baloanele sunt spawned, pe măsură ce trece timpul.

**Bibliografie**

https://www.youtube.com/channel/UCYbK\_tjZ2OrIZFBvU6CCMiA

https://www.youtube.com/channel/UCBrCPYAEhVCjYJJj1SMyzvA

https://www.youtube.com/user/Unity3D/featured

https://www.youtube.com/user/KnnthRA

https://gamedev.stackexchange.com/questions/97265/unity-basic-ai-enemy-follows-player-prevent-flocking-of-enemies-together

https://stackoverflow.com/questions/16844020/how-to-make-an-enemy-ai-that-would-follow-the-player-without-rotating-or-looking

https://forum.unity.com/threads/making-enemy-ai-follow-script-without-rotating-the-gameobject.255760/

https://answers.unity.com/questions/971104/unity-2d-c-script-follows-the-enemy-player.html

https://unity3d.com/learn/tutorials/topics/2d-game-creation/2d-game-development-walkthrough

https://answers.unity.com/questions/1190440/enemy-spawning-help-2d-platformer-c.html

https://answers.unity.com/questions/1373051/fire-projectile-towards-mouse-position-top-down-2d.html

https://answers.unity.com/questions/343014/make-an-arrow-stick-into-object-that-it-hits.html