Colegiul Național „Grigore Moisil”

Proiect pentru atestat

„Visual Effect”

Profesor îndrumator: Autor: Rang Răzvan - Victor

Pintea Rodica Clasa: a XII – a A

Profil și Specializare: Matematică – Informatică intensiv Informatică

**Cuprins**

1. Introducere
2. Cerințe de sistem
3. Detalii pentru utilizator
4. Detalii tehnice
5. Bibliografie

I. Introducere

Proiectul are în vedere distrarea și relaxarea utilizatorului, dar și competiția dintre doi jucători. Acesta constă într-o cursă dintre două patrățele care se lupta pentru a nu se atinge de barele care vin spre ele. Între bare exista un spațiu prin care jucătorii pot trece, dar pe măsură ce jucătorii avansează, acest spațiu se miscșorează, iar viteza cu care vin barele crește. Jocul este realizat, în principal, pentru a fi jucat de către doi oameni, dar există un mod în care un singur jucător concurează cu un patrățel robot, care nu este la fel de provocator ca un prieten, dar este un bun antrenament pentru când nu există posibilitatea de a se juca în doi.

Motivația care a stat la baza conceperii acestui joc este una simpla, un joculeț de omorât timpul pentru cine vrea să se relaxeze sau vrea sa își încerce abilitățile de a-și încurca adversarul. Este un joc simplu, dar poate strica multe prietenii și în care se poate pierde mult timp datorită spiritului competitiv pe care îl aduce. Jocul poate fi jucat de oricine, de la copii la oameni învârstă, existând un buton de „Help” care explică care sunt controalele si care este scopul.

Aplicația este realizată în Visual Studio 2015, folosind limbajul de programare C# și principiile programării orientate pe obiecte (OOP). Limbajul C# sau C sharp a fost lansat publicului in iunie 2000 și oficial în primăvara anului 2002 și este un limbaj de programare care combină facilități testate de-a lungul timpului cu inovații de ultim moment. Creatorii acestui limbaj au fost o echipă de la firma Microsoft condusă de Anders Hejlsberg. Deși este creat de Microsoft, acesta nu este destinat doar platformelor Microsoft, existând compilatoare C# si pentru sisteme de operare precum Linux sau Macintosh. C# derivă din doua dintre cele mai de succes limbaje de programare: C și C++. De asemenea, limbajul este o „rudă” apropiată a limbajului Java.

II. Cerințe de sistem

**Cerințe hardware**

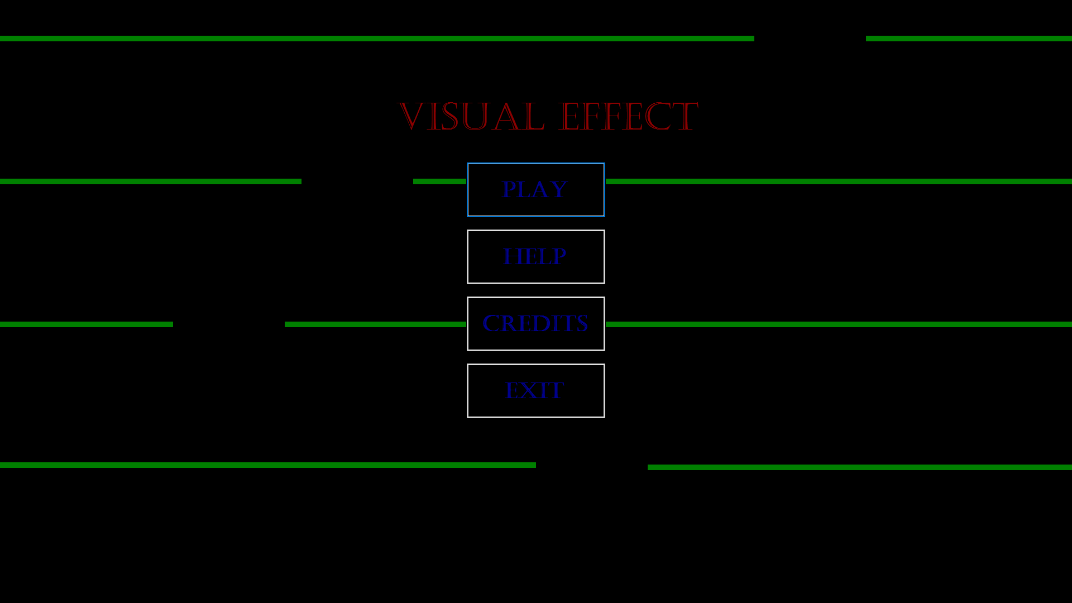
* Procesor construit pe 32-bit (x86) sau pe 64-bit (x64)
* Procesor Dual-core, min 2 gHz sau mai rapid
* 1 Gb RAM
* Spațiul ocupat este infim (80 KB)
* Placă video care suportă DirectX 9 cel puțin

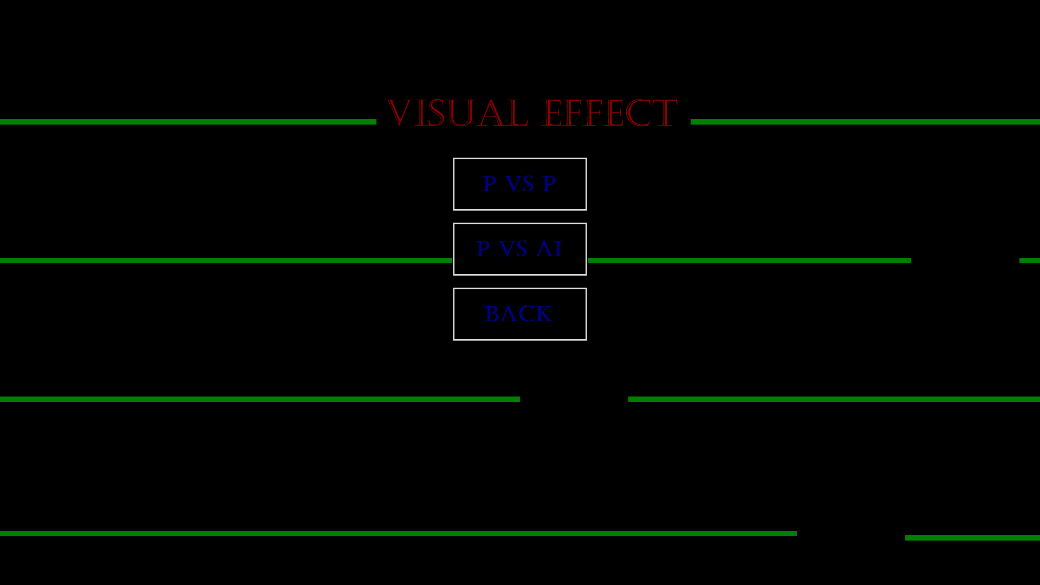
**Sisteme de operare**

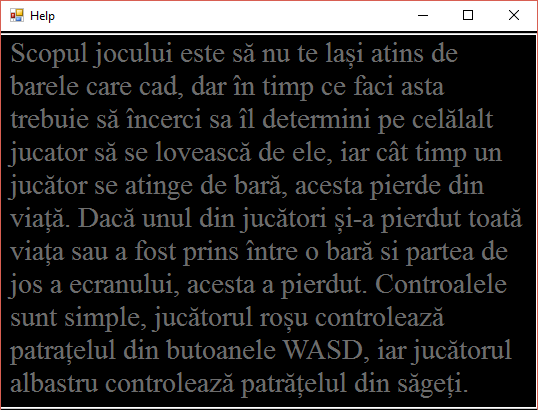
* Windows XP (toate SP-urile)
* Windows Server 2003 (SP2 și R2)
* Windows Vista
* Windows Server 2008 (SP2 și R2)
* Windows 7
* Windows 8
* Windows 8.1
* Windows 10

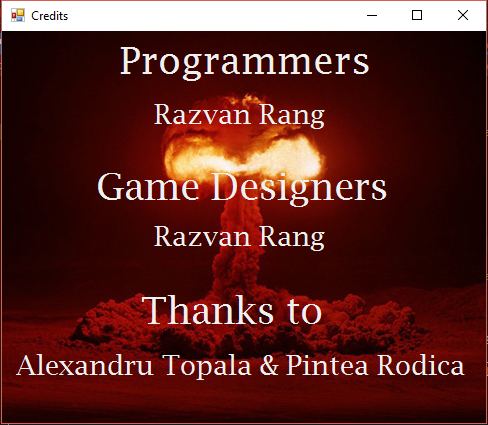
Aplicația merge pe orice variantă de windows, 32-bit sau 64-bit.

III. Detalii pentru utilizator

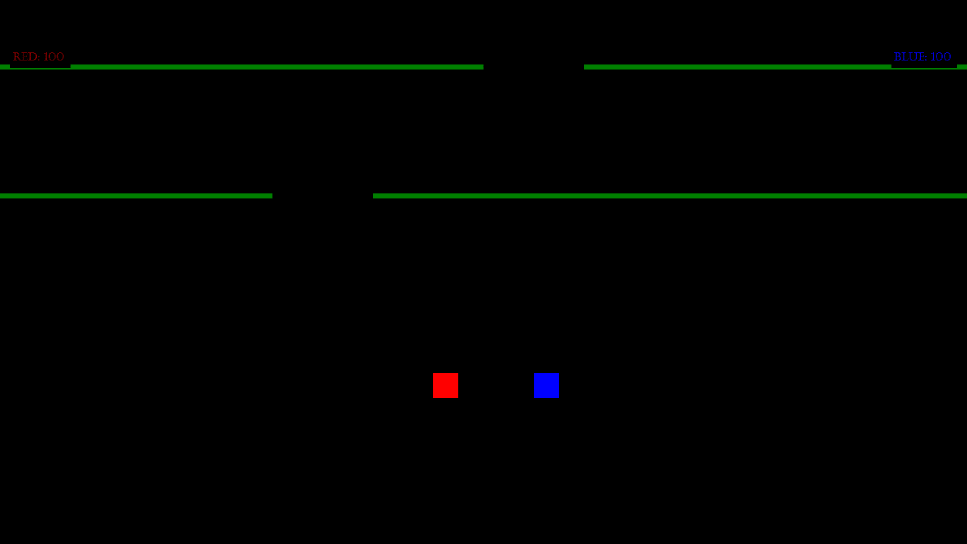


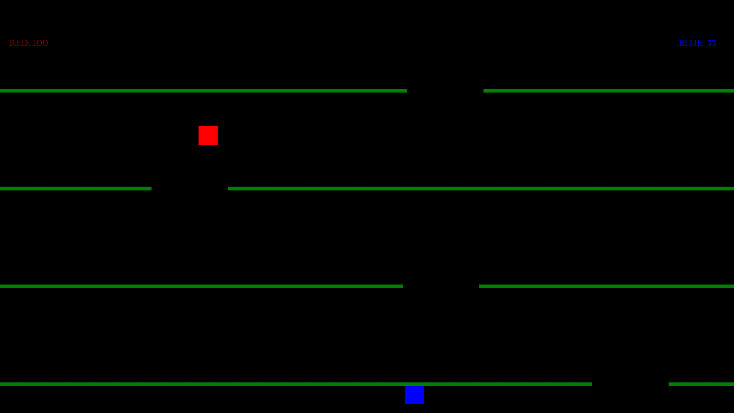
Jocul se deschide în full screen cu un meniu cu 4 butoane: Play, Help, Credits și Exit. Play duce la alte 3 butoane: P vs P (pentru a juca în doi), P vs AI (pentru a juca cu un bot) și Back pentru a ajunge înapoi la meniul principal.

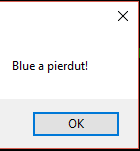
 Butonul de Help deschide o fereastră care explică scopul jocului și controalele acestuia.



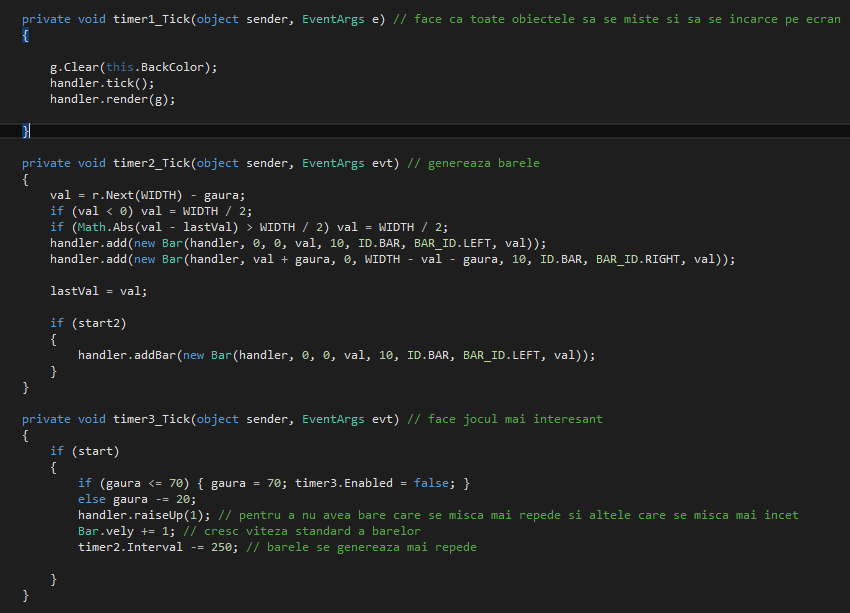
Dacă apăsăm pe butonul Credits, acesta va deschide o fereastră în care sunt numiți cei care au ajut at la realizarea aplicației. Iar butonul Exit, face ceea ce zice că face.

 Jocul începe cu cele două pătrate aliniate si gata de cursă, urmând ca după cateva secunde să înceapă barele să apară.



 Iar atunci când unul din jucători pierde, cum ar fi în poza alăturată, unde pătratul albastru a fost prins în partea de jos, va apărea un mesaj în care este scrisă culoarea pierzatorului.

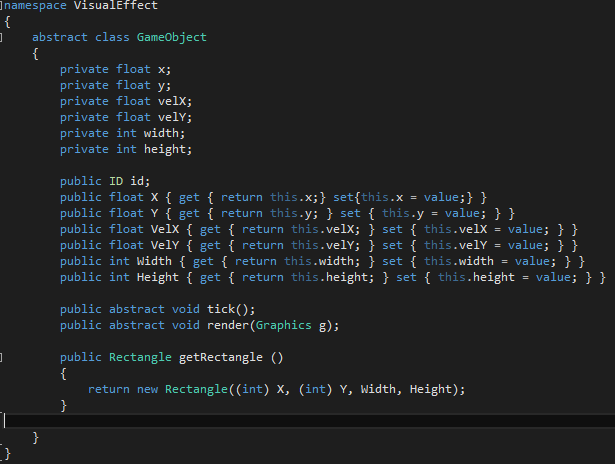
IV. Detalii Tehnice

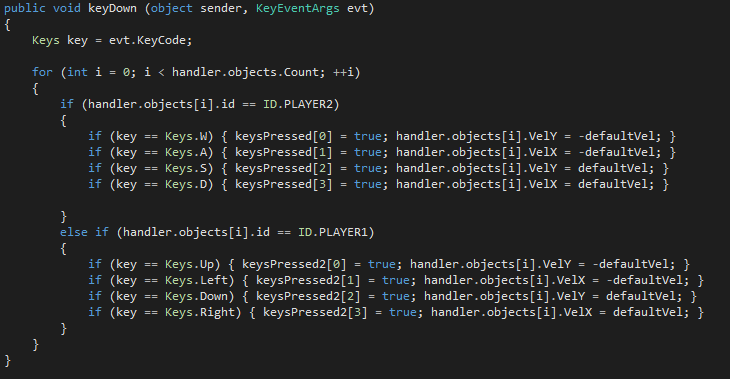
 În continuare voi prezenta câteva dintre principalele metode ale aplicației. Din punct de vedere teoretic, nu sunt utilizați algoritmi complexi, doar lucruri de bază și mult lucru pe obiecte.

Această secvență cuprinde metodele care fac jocul să se miște. Funcția timer1\_Tick() curăță ecranul de obiecte cu g.Clear(), handler.tick() actualizează poziția obiectelor pe ecran, handler fiind instanța clasei care îndosariază toate obiectele grafice din joc și handler.render(g) redesenează obiectele pe ecran, după ce le-a fost actualizată poziția, pe obiectul grafic al ferestrei, g.

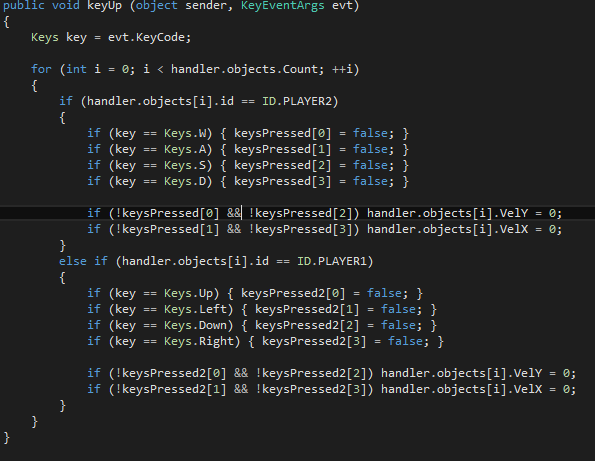
În timer2\_Tick() se generează barele la un interval prestabilit. În variabila val se reține o valoare aleatoare, care reprezintă punctul de unde va începe următorul spațiu, apoi se verifică dacă valoarea respectivă este mai mică decât 0 și în caz adevărat este înlocuită cu valoarea mijlocului ecranului, iar apoi este verificat dacă distanța dintre ultimele două spații este mai mare decât jumătatea ecranului pentru a preveni o distanță prea mare de parcurs de către un jucător între spații. După aceea sunt creeate instanțele de bare. Start2 este true dacă jucătorul se joacă singur cu bot-ul, iar în acest caz se reține bara din stânga pentru determinarea punctului unde va trebui să ajungă robotul sau false dacă sunt în joc doi oameni.

În timer3\_Tick() se face jocul mai greu. La un interval de 20 de secunde distanța dintre bare se micșorează, până la o valoare minimă, când timer-ul se oprește și viteza cu care vin barele crește, iar intervalul la care ele vin scade pentru a pastra aproximativ aceeași distanță dintre bare

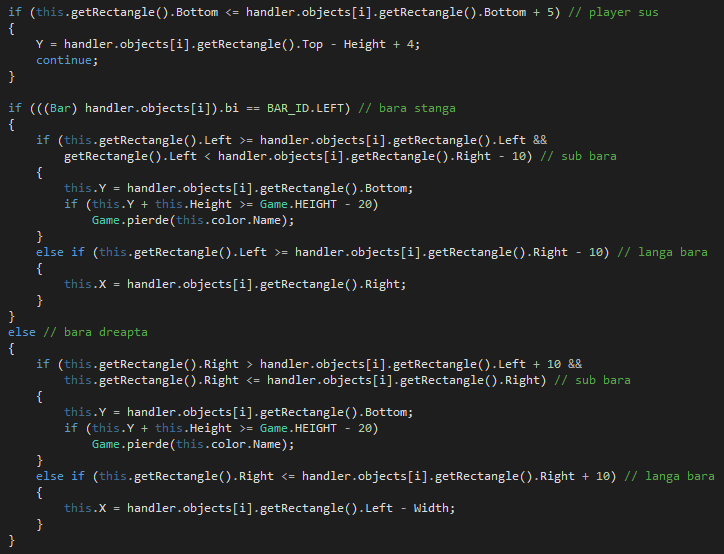
Clasa GameObject este o clasă abstractă și definește metodele de bază pe care le are fiecare obiect din aplicație. Această clasă va fi moștenită de toate clasele care definesc un obiect grafic în joc.



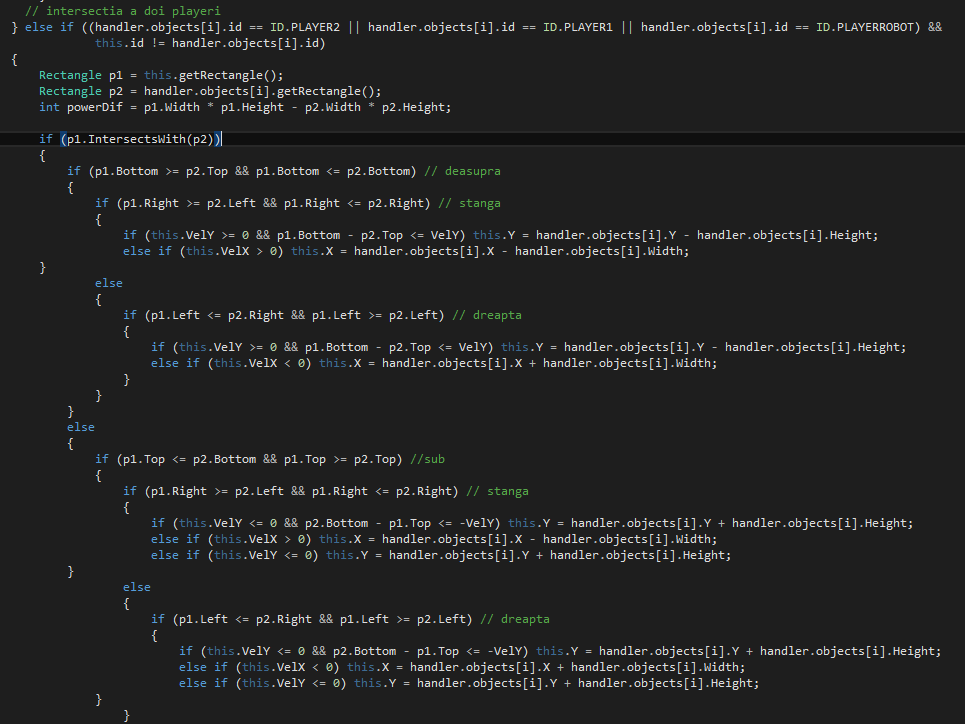
Metoda keyDown face parte din clasa KeyInput care conține tot ce ține cont de apăsarea tastelor în joc. Funcția keyDown este apelată automat atunci cand este apăsată o tastă și se verifică care a fost tasta și în funcție de tastă se modifică sau nu poziția unui pătrat cu o valoare prestabilită, defaultVel și se reține în vectorul keyPressed tasta apăsată.



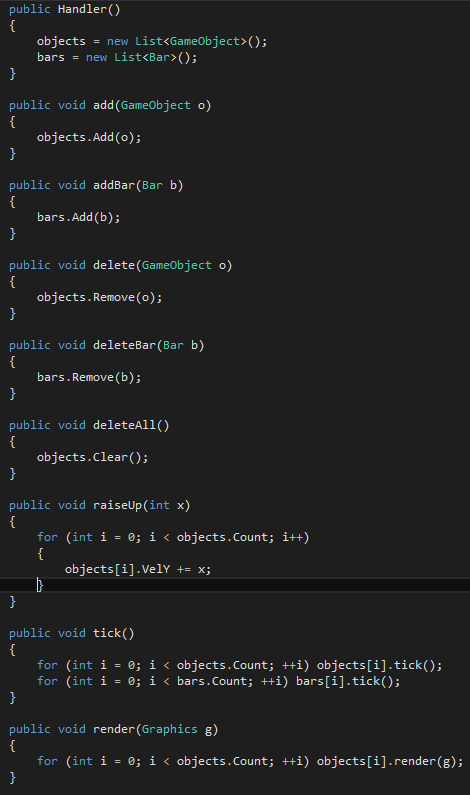
În metoda keyUp, care face parte tot din clasa KeyInput, este definit ce se întâmplă atunci când unul din jucători ridică degetul de pe o tastă. În acest moment vectorul keysPressed revine la starea inițială, aceea în care nu este apăsată vreo tasta și se resetează vitezele de miscare ale obiectelor la 0.

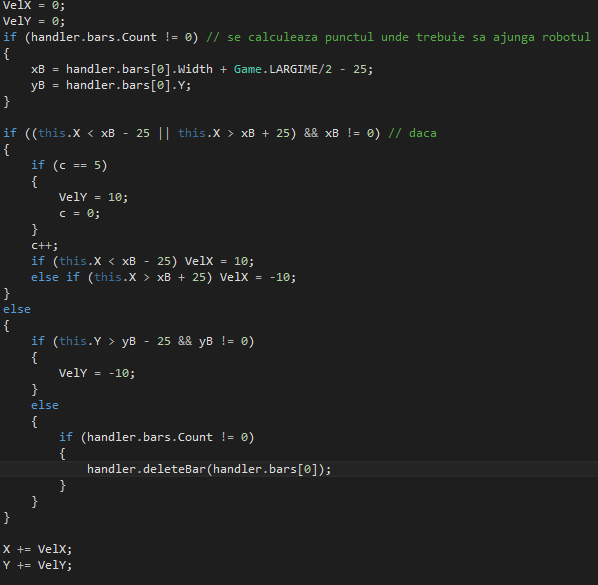


Această secvență de cod reprezintă algoritmul de deintersectare a unui pătrat cu o bară, prezentă în metoda tick() din clasa Player. Se verifcă pe rând poziția pătratului față de bară și în funcție de această poziție jucătorul este tras într-o anumită direcție pentru a nu se intersecta niciodată cu o bară și se mai verifică daca jucătorul este prins între o bară si partea de jos a ecranului, caz în care acesta pierde. Metoda getRectangle() face parte din clasa GameObject și întoarce o instanță de pătrat pentru a manevra mai usor coordonatele obiectelor din joc.



În secvența de mai sus este cuprinsă deintersectarea dintre doi jucători, p1 și p2 sunt instanțele de pătrate ale jucătorilor. Apoi se ia fiecare caz posibil de intersecție și se mută obiectele în așa fel încât să nu se intersecteze, dar să pară că se ating.

 Metodele alăturate fac parte din clasa Handler, cea care ține evidența tuturor obiectelor grafice din joc și care are ca atribute o listă de obiecte de tip GameObject, cel mai general tip de obiect și o listă de bare care este folosită în cazul în care se joacă modul P vs AI. Metodele mai aparte sunt raiseUp() care crește viteza de mișcare pe verticală a obiectelor atunci când se crește viteza în joc, metoda tick() actualizează toate obiectele grafice, iar render le desenează pe ecran.



Secvența de mai sus reprezintă calculul punctului unde trebuie să ajungă robotul, în variabilele xB și yB se rețin coordonatele punctului, apoi, în funcție de ele, robotul se deplasează spre punct, iar atunci cand se mișcă la stânga sau la dreapta, după 4 mutări se duce o data în jos pentru a nu rămâne lipit de bară, deoarece el nu se mișcă în două direcții simultan. La final, după ce trece de bară, aceasta este ștearsă din vector și robotul trece la următorul punct.

V. Bibliografie

* <https://support.microsoft.com/ro-ro/help/983509/description-of-visual-studio-2010-service-pack-1>
* <http://www.math.uaic.ro/~cgales/csharp/Curs1.pdf>
* <https://stackoverflow.com/>
* <https://www.youtube.com/watch?v=1gir2R7G9ws>