

Maramureș, Baia Mare

Colegiul Național “Vasile Lucaciu”

Clasa a XI-a, matematică informatică-intensiv informatică

Membri: **Covaci Alexandra, Alex Manu**

Coordonator: **prof**. **Alin Maidan**

**Capitolul I – Mediul de dezvoltare utilizat**

Microsoft Visual Studio este un mediu de dezvoltare integrat (IDE) de la Microsoft. Acesta este utilizat pentru a dezvolta programe pentru Microsoft Windows, precum și pentru site-uri web, aplicații web și servicii web. Visual Studio utilizează platforme de dezvoltare software Microsoft, cum ar fi API Windows, Windows Forms, Windows Presentation Foundation, Windows Store și Microsoft Silverlight.

Visual Studio acceptă diferite limbaje de programare, precum C++, C#, VB sau J#.

Limbajul C# a fost dezvoltat de o echipă restrânsă de ingineri de la Microsoft, echipă din care s-a evidențiat Anders Hejlsberg. C# este un limbaj simplu, cu circa 80 de cuvinte cheie și 12 tipuri de date predefinite. El permite programarea structurată, modulară și orientată obiectual, conform perceptelor moderne ale programării profesioniste.

Principiile de bază ale programării pe obiecte sunt elemente fundamentale ale programării C#. În mare, limbajul moștenește sintaxa și principiile de programare din C++. Sunt o serie de tipuri noi de date sau funcțiuni diferite ale datelor din C++, iar în spiritul realizării unor secvențe de cod sigure, unele funcțiuni au fost adăugate (de exemplu, interfețe și delegări), diversificate (tipul struct), modificate (tipul string) sau chiar eliminate (spre exemplu, pointerii către funcții). Chiar dacă derivă din limbajul C++, limbajul C# include influențe și din alte limbaje, mai ales Java.

În C# se scriu aplicații pentru sisteme complexe care funționează sub o mare varietate de sisteme de operare, cât si pentru sisteme dedicate. Acestea din urmă se întind pe o arie largă, de la dispozitive portabile, cum ar fi ceasurile digitale, telefoanele mobile, MP3 playere, până la dispozitive staționare ca semafoarele de trafic.

Unity este un game engine dezvoltat de Unity Technologies folosit pentru a dezvolta jocuri video pentru PC, console și dispozitive mobile. Printre limbajele de programare suportate se numără C# și JavaScript.

**Capitolul 2 – Descriere generală a proiectului**

Tema proiectului este un joc de tip arcade, care are ca scop conducerea extraterestrului Yumfo spre salvarea planetei de la invazie.

Este un joc de divertisment care este destinat tuturor persoanelor indiferent de vârstă.

Jocul începe cu povestea extraterestrului Yumfo trăind liniștit pe planeta lui cand aceasta este invadată de alți extratereștrii, el este răpit si dus pe nava lor. El reușește să scape din celula unde este închis. În prima parte Yumfo caută o cale de scăpare de pe nava lor. În cea de a doua parte Yumfo evadează și ajunge într-o padure de pe planeta sa. Apoi el se întreaptă spre orașul său natal pentru a restabili liniștea. În continare aventurile sale continuă.

Yumfo este un extraterestru care are abilitatea de a trage cu lasere şi de asemena are şi o sabie laser care îl ajută în lupta sa împotriva extratereştriilor invadatori. Acesta are 3 vieţi pe care le poate creşte prin colectare pe parcursul jocului sau le poate pierde atunci când se loveşte de capcane sau este omorât de extratereştrii după care revine la checkpoint, dar în cazul în care rămâne fără vieţi, atunci Yumfo pierde jocul şi trebuie să reînceapă nivelul.

Prima parte a jocului se încheie atunci când Yumfo ajunge la nivelul 10 şi îl învinge pe unul dintre extratereştrii superiori ai jocului. Cea de a doua parte a jocului se termină atunci când Yumfo îi învinge pe cei doi extratereştrii giganţi care nu îl lasă să ajungă în oraşul lui. În a treia parte Yumfo ajunge în orașul său natal unde întâlnește roboți răi. Pe tot parcusul orașului Yumfo trebuie sa îi infrânga pe roboții cei răi. După ce face asta, se intâlnește cu Yumfo Genesis, șeful roboților și al extratereștriilor răi care au invadat planeta lui, acesta îl capturează și îl duce în lagărul său. În partea a patra el îi întâlnește pe semenii lui capturați de Yumfo Genesis și dorește să îi ajute sa evadeze. Astfel el caută o cale de scăpare și trebuie să îi înfrângă pe toți extratereștrii supremi de la fiecare parte pentru a se putea întoarce la viața sa de înainte.

**Capitolul 3 – Resurse necesare**

Cerințe minime de sistem:

* OS: Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, Android 2.3.1+
* Placa video: DX9 (shader model 2.0)
* CPU: SSE2 instruction set
* Web Player: necesită un browser care suportă pluginuri, precum Safari, Mozilla, IE.

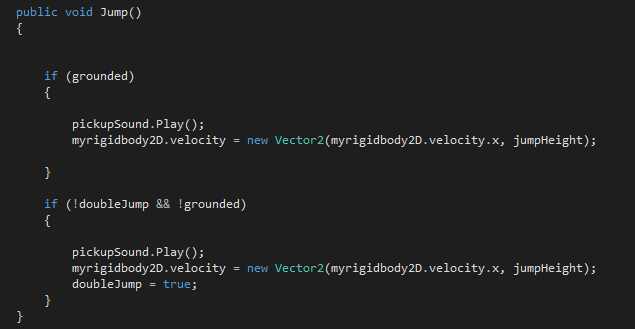
**Capitolul 4 – Instalarea**

Pentru a nu complica lucrurile inutil, vă punem la dispoziție arhiva proiectului care se descarcă de la adresa (<http://yumfo.wecode.ro/>) la sectiunea download. Apoi aveți posibilitatea să o dezarhivați în orice locație doriți referirea la fisiere fiind de tip relativ si astfel fiind posibilă rularea proiectului de oriunde. Dimensiunea mai mare se datorează faptului că vă punem la dispoziție atât fișiere necesare pentru sisteme pe 32 cât și pe 64 de biți, cunoscute fiind problemele legate de incompatibilitatea aplicaților generate cu UNITY între cele 2 tipuri de sisteme de operare.

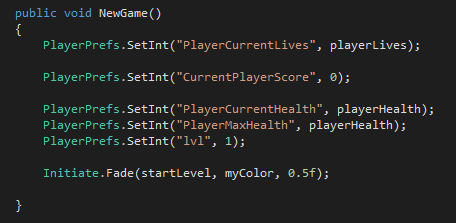
De asemenea jocul vine şi cu versiunea de android care poate fii descărcată gratuit de pe Magazin Play căutând numele jocului, Yumfo. Jocul poate să fie jucat online pe site-ul jocului (<http://yumfo.wecode.ro/>) prin apăsarea butonului Play de la secţiunea download.

**Capitolul 5 – Detalii tehnice de implementare**

Codurile jocului au fost preluate și adaptate de pe <https://www.youtube.com/watch?v=-ixk2uxEc94&list=PLiyfvmtjWC_Up8XNvM3OSqgbJoMQgHkVz> .

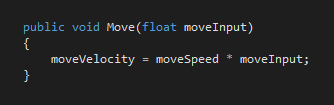


În imaginea de mai sus, descriem subprogramul necesar pentru săritura extraterestrului. În primul rând verificăm dacă Yumfo se află pe pământ (if (grounded) ), în acest caz se porneşte sunetul de săritură şi se efectuează săritura la o înălţime aleasă de noi prin variabila jumpHeight. În următoarea condiţie oferim posibilitatea jucătorului să aplice săritura dublă. Pentru acest lucru verificăm dacă sub player nu este pământ şi a efectuat doar o săritură. În acest fel, se porneşte din nou sunetul de săritură, se realizează din nou săritura la înălţimea aleasă de noi şi variabila doubleJump ia valoarea true.



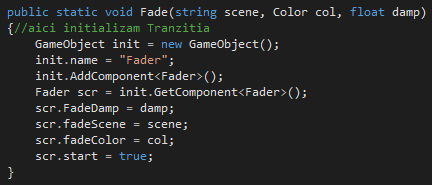
Funcţia de mai sus este ataşată de un buton, unde, la fiecare apăsare a sa, se îndeplinesc următoarele condiţii :

* Prima linie din subprogram are rolul de a apela o altă clasă care iniţializează “PlayerCurrentlives” ia valoarea variabilei playerLives;
* “CurrentPlayerScore” se iniţializează cu valoarea 0;
* “PlayerCurrentHealth” ia valoarea variabilei playerHealth;
* “PlayerMaxHealth” se iniţializează cu valoarea variabilei playerHealth;
* Prin iniţializarea variabilei “lvl” se alege culoarea lui Yumfo selectând un număr de la 1 la 3;
* „Initiate” iniţializează un efect de tip fade către nivelul salvat în stringul „startLevel”, având culoarea („mycolor”) aleasă de noi şi viteza de 0,5 f.

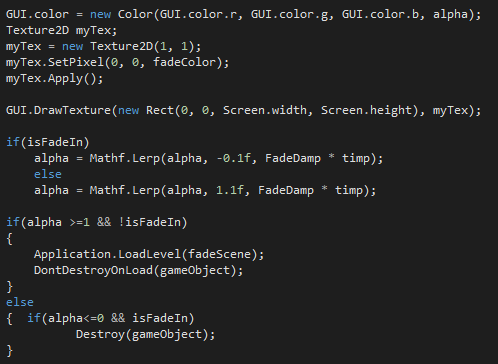


Subprogramul public void Move primeşte prin intermediul variabilei « moveInput » direcţia de mers a lui Yumfo. Deplasarea se va realiza prin produsul dintre o viteză aleasă de noi şi valoarea 1, dacă merge spre dreapta, respectiv, -1, dacă merge spre stânga.

Efectul de tranziție pentru nivele a fost preluat și adaptat din următorul video: <https://www.youtube.com/watch?v=YHyCzCkkp9I>.



Secvența de cod de mai sus are rolu de a inițializa de fiecare dată un **GameObject** cu numele **Fader** care să facă tranziția între scene cu o anumită **culoare** și o anumită **viteză** de derulare, acesta fiind inițializat doar la apelul **Initiate.Fade(level,culoarealeasa,vitezaderulare);**.



Codul de mai sus reprezintă mecanismul tranziției, care setează culoarea fedului prin codul de culori rgb si alpha reprezentânt nivelul de transparență, apoi creeaza seteaza textura sa pe dealungul intreguilui ecran. După care urmează să verifice ce fel de tranziție sa facă deoarece există tranziție de de început unde transparența este la 255 (adica nu există transparentă), și tranzitie de final unde nivelul de transparență este 0.

**Capitolul 6 – În loc de concluzii**

Deoarece am dorit ca jocul să fie în totalitate original, am căutat pe google un nume care nu a mai fost folosit înainte.

Grafica este realizată personal de noi în totalitate cu ajutorul programelor Paint si Photoshop. Sunetele sunt realizate cu ajutorul unui program Bfrx ( <http://www.bfxr.net/> ), iar fundalul sonor care apare pe tot parcursul jocului este realizat de un coleg de clasa , Lovasz Norbert, care a inregistrat piesa la clape si a modificat-o pe calculator ulterior.

Video-ul de promovare a fost inregistrat si editat de către noi.

Imaginiile care conturează povestea jocului sunt făcute de o prietenă, Cora Săsăran.

Font-ul jocului a fost descărcat de pe site-ul <http://www.dafont.com/from-cartoon-blocks.font> .

În concluzie, jocul este finalizat , dar există posibilitatea de a apărea pe parcursul timpului mici îmbunătățiri.

**Bibliografie**

[www.wikipedia.org/](https://www.wikipedia.org/) (informații despre visual studio , unity și C# )

<https://unity3d.com/> (programul în care am realizat jocul )

https://msdn.microsoft.com/en-us/library/kx37x362.aspx (informații despre microsoft )

<http://docs.unity3d.com/Manual/index.html> (idei de rezolvare a problemelor cu ajutorul forumurilor )

<https://www.youtube.com/watch?v=-ixk2uxEc94&list=PLiyfvmtjWC_Up8XNvM3OSqgbJoMQgHkVz>

<https://www.youtube.com/watch?v=YHyCzCkkp9I>

<http://www.dafont.com/from-cartoon-blocks.font>

<http://www.bfxr.net/>

**Cuprins**

* Informații despre echipa noastră (pagina 1)
* Capitolul 1– Mediul de dezvoltare utilizat (pagina 2)
* Capitolul 2 – Descriere generală a proiectului (pagina 3)
* Capitolul 3 – Resurse necesare (pagina 4)
* Capitolul 4 – Instalarea (pagina 5)
* Capitolul 5 – Detalii tehnice de implementare (paginile 6-8)
* Capitolul 6 – În loc de concluzii (pagina 9)
* Bibliografie (pagina 10)