

Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnică a Moldovei

Catedra Automatica și tehnologii informaționale

Medii interactiva de dezvoltare a produselor soft

Lucrare de laborator # 6

Lucru in echipa. Aplicatie complexa

A efectuat:

st. gr. TI-141
M. Ghelas

A verificat:

lect. asistent
Irina Cojanu

Chișinău 2016

1. Scopul Lucrării:

- Lucru în echipă
- Aplicarea tehnicilor, limbajelor de programare și IDE-uri studiate în laboratoarele precedente.

2. Obiective:

- Crearea unei aplicații complexe în echipă.
- Divizarea sarcinilor pe membrii echipei

3. Implementarea

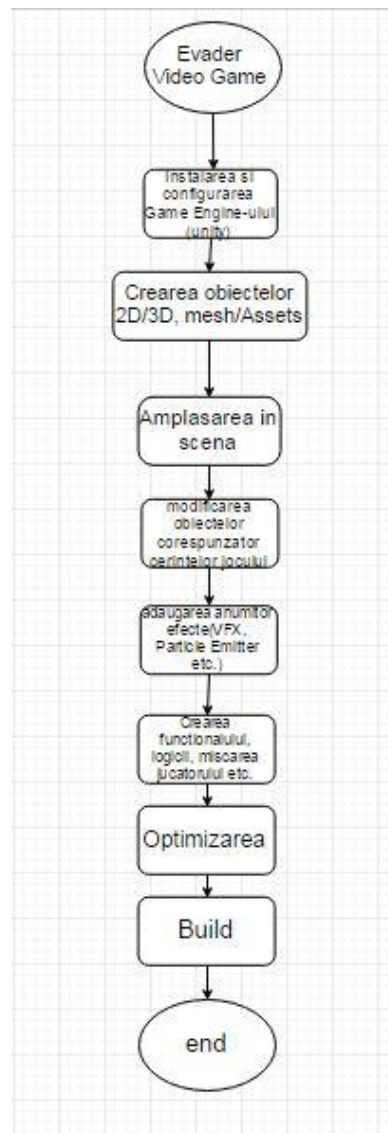
3.1 Taskuri și puncta

- Joc tip Arcade elaborate în game Engine-ul Unity3D.

Link către repository original: <https://github.com/teacateodor/Laboratorul-6>

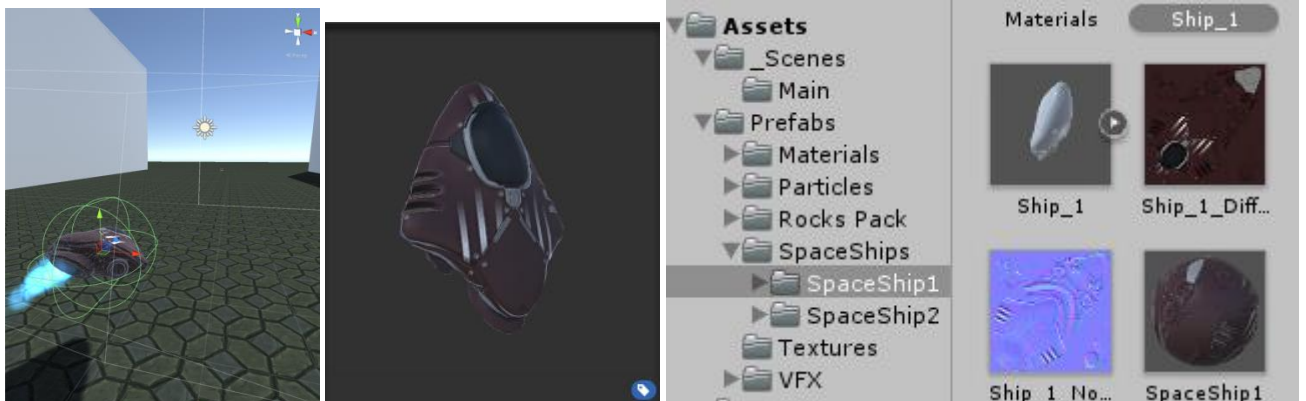
Link către fork: <https://github.com/Erdboden/Laboratorul-6>

Am decis să cream un joc bazat pe aptitudinile capitate din laboratoarele precedente (laboratorul nr. 5 în deosebite) și tutoriale de pe site-ul oficial Unity3D (unity3d.com). Acest joc face parte din categoria Arcade, de la a 3-a persoană unde jucătorul controlează o navă spațială și trebuie să ocolească obstacolele în formă de pietre care se vor genera random de sus și se vor mișca către jucător. Ideea principală a mecanicii jocului este că jucătorul stă pe loc, dar vom crea iluzia de mișcare prin pietrele ce se vor mișca către jucător și pământul care de asemenea se va mișca înapoi. Jocul va continua până la infinit până nu se va distruge nava spațială a jucătorului. De asemenea există și un sistem de scor bazat pe timp.

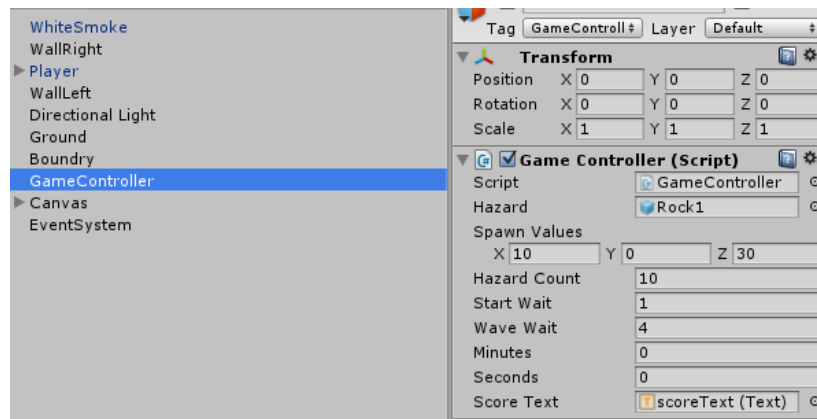


Unele obiecte (prefabs sau mesh) au fost luate de pe store-ul oficial Unity sau de pe alte portale de modele 3d.

Pentru inceput am adaugat toate elementele necesare in editor, adica modelele 3d, texturele, materialele etc. si le-am amplasat pe scena.



Acum ramine de facut functionalul jocului. Incepem cu scriptul principal gameController care permanent va fi in scena si va coordona toate evenimentele jocului. Pentru aceasta trebuie sa cream un obiect invizibil pe scena si sa-i atribuim script-ul acestui obiect.



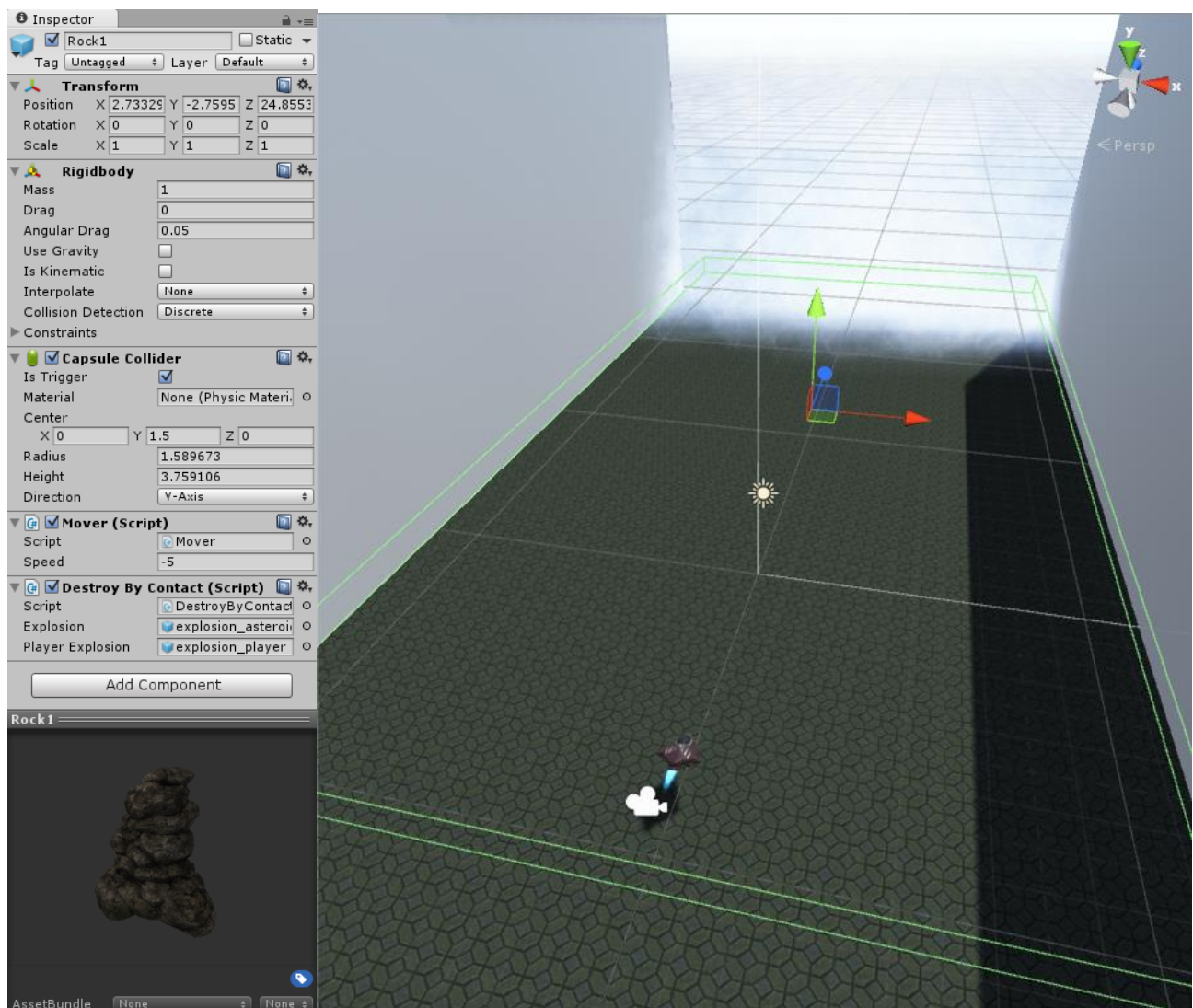
În principiu scriptul va include totul ce e legat de apariția obstacolelor în scenă, și anume apariția lor în anumite puncte. De asemenea se va face restart jocului de fiecare dată când jucătorul pierde și scorul.

Acum avem obstacolele dar ele nu se mișcă. Cream scriptul pentru mișcarea lor Mover.cs

Atribuim acest script obstacolelor și le facem ca prefabs.

De asemenea dacă ele se vor mișca în urmă, jucătorul nu le va vedea, dar ele totuși vor fi în joc, deci trebuie să le distrugem atunci când a trecut de jucător.

Pentru aceasta cream un obiect nou invizibil în scenă dar cu un collider. Obstacolele vor apărea în collider iar la ieșirea din obiect obstacolele vor dispărea. Cream scriptul DestroyByBoundry.cs și atribuim obiectelor ce vor reprezenta obstacolele



Zona verde din a doua figură reprezintă obiectul invizibil cu un Collider.

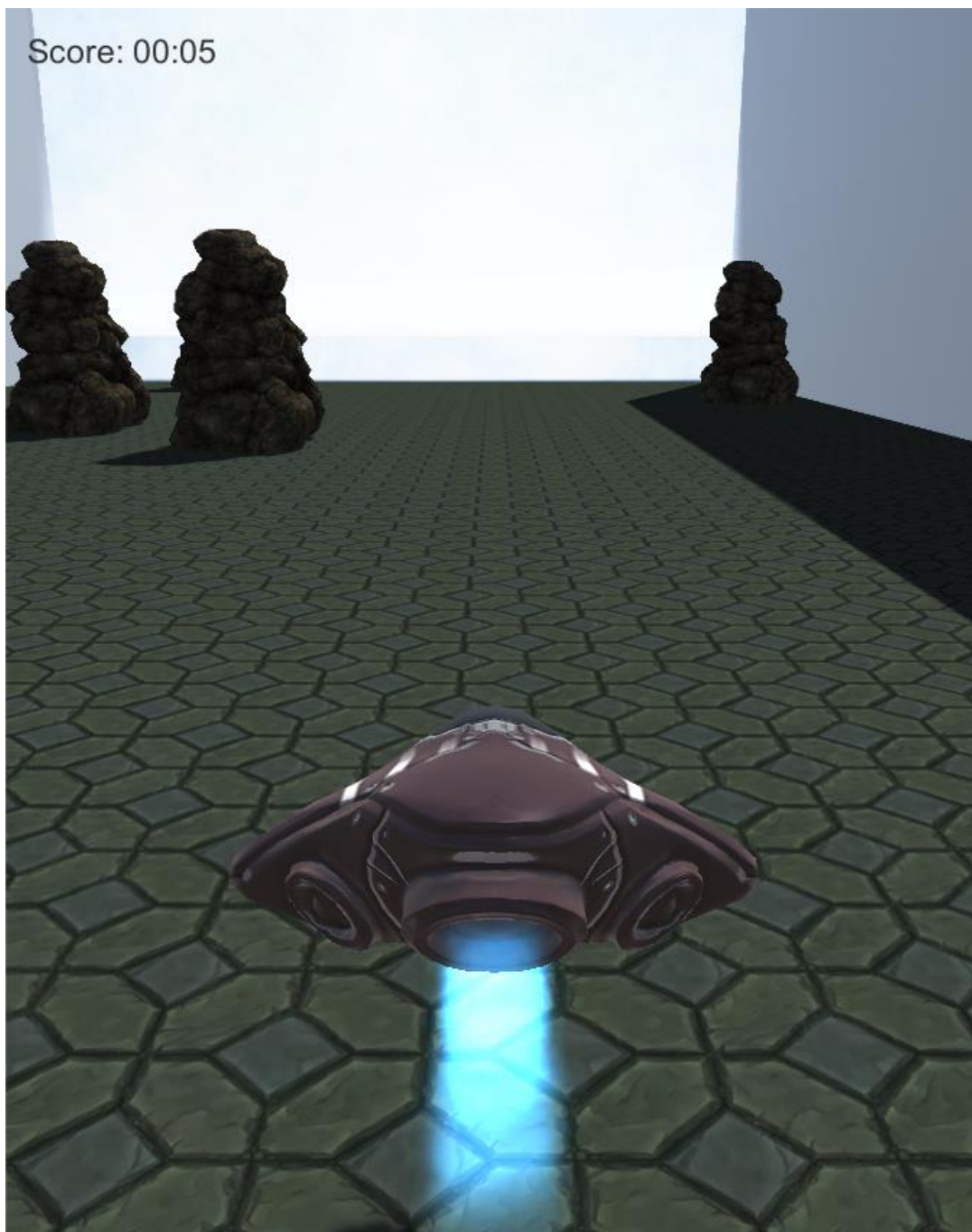
Obstacolele apar, dar jucatorul inca sta pe loc. Trebuie sa cream miscarea jucatorului nostru. Cream scriptul PlayerController.cs. Vom folosi sagetile pentru miscare in stinga si dreapta. Deasemenea trebuie de prevazut ca jucatorul sa nu iasa in afara zonei amenajate, deci nava spatiala nu trebuie sa iasa dupa pereti.

Jucatorul deja poate sa se miste, insa nui nici un sens in joc deoarece nava trece prin obstacole. Pentru aceasta trebuie sa scriem scriptul DestroyByContact.cs unde se va prevedea distrugerea navei atunci cind se va ciocni cu un obstacol. Aici cel mai mult vom folosi metoda cu etichete. Eticheta la nava va fi player, si atunci cind colliderele la obiectul cu eticheta Player si la obiectul obtsacol se vor ciocni jocul se va restarta.

Atribuim acest script obiectului obstacol.

A ramas de creat iluzia de miscare prin miscarea planului pe care se afla nava. Pentru aceasta cream scriptul BGScroller.cs . Planul se va misca in urma cu o viteza constanta anumita data de programator si cind pozitia planului pe Z va fi 0 atunci se va reseta pozitia initiala a lui si se va repeta acelasi proces.

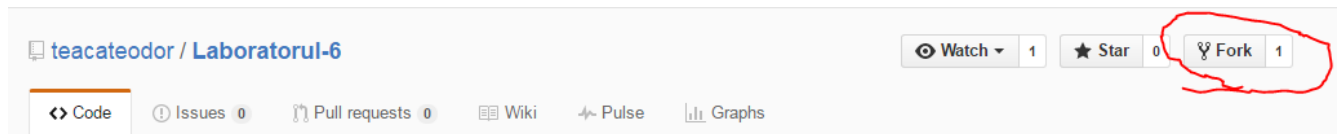
Jocul final:



Acest proiect treptat a fost adaugat pe platforma github.com.

Procesul

Initial, s-a creat un repository curat special pentru acest proiect. Dupa aceasta s-a creat prima iteratie a jocului, s-a incarcat pe github(Repositoryul se afla pe contul lui Teaca Teodor). Apoi eu am facut un “fork” acestui proiect direct de pe site-ul github.com:



Si acest fork l-am clonat pe calculatorul local. Ambii am creat un branch pe care am lucrat. Fiecare schimbare in proiect se face pe branch-ul propriu, si pull request catre branch-ul principal master. Deasemenea era nevoie de actualizat proiectul de fiecare data cind un membru a echipei a schimbat ceva.

Concluzie

La aceasta lucrare de laborator am lucrat in echipa cu Teaca Teodor folosind serviciul web de hosting a proiectelor Github. Cu ajutorul acestui serviciu paralel am lucrat la proiect si l-am dezvoltat. A fost o experienta noua pentru noi atit in mediul de dezvoltare a jocurilor(anume in editorul Unity3d) cit si folosirea github-ului la acest proiect mai complex decit celelalte laboratoare.