

## SPACE::SAT

Barcan Nicoleta Gabriela  
1210A

### Povestea jocului:

Acțiunea se desfășoară în Cosmos, la 210420 km de planeta Pământ.

După o expediție pe Lună, care a durat 12 zile, echipajul RoSatOne este pregătit să se întoarcă acasă cu noile descoperiri făcute. La întoarcere au întâmpinat o ploaie de asteroizi, în urma căreia rezervorul cu combustibil și zona cu mâncare au fost lovite. Au reușit să repare acele părți, dar au suferit pierderi majore. După ce au făcut un calcul, au realizat că nu au suficient combustibil ca să se întoarcă acasă, prin urmare căpitanul Matt Taylor s-a hotărât să meargă la satelitul navei ca să ceară ajutor de la bază. Din nefericire câțiva asteroizi mai mici au lovit satelitul, rezultând în despărțirea panourilor și a antenei.



### Prezentare joc:

Este un joc single player, 2D cu 2 niveluri. Sunt 2 părți din satelit care s-au desprins, antena și panoul drept, acestea s-au împărțit la rândul lor în alte bucăți mai mici datorită coliziunii cu asteroizii. Jucătorul va trebui să recupereze fiecare bucătică și să le aducă la navă pentru a putea fi unite, după care va trece la următorul nivel, unde dificultatea va crește. Finalul semnifică repararea satelitului și salvarea echipajului.

### **Reguli joc:**

Fiind în spațiu, jucătorul nu va vedea tot drumul, ci doar o parte apropiată lui (voi folosi noțiunea de cameră). Acesta va trebui să treacă de obstacole și să găsească obiectele, totodată ferindu-se de asteroizi.

Vor fi 5 piese pe fiecare nivel de acumulat, pe care o dată ce le are pe toate, poate să se întorcă la navă și să treacă la următorul nivel. Pe drum are posibilitatea să găsească și mâncare, ce va reprezenta puncte bonus.

### **Prezentare jucător:**

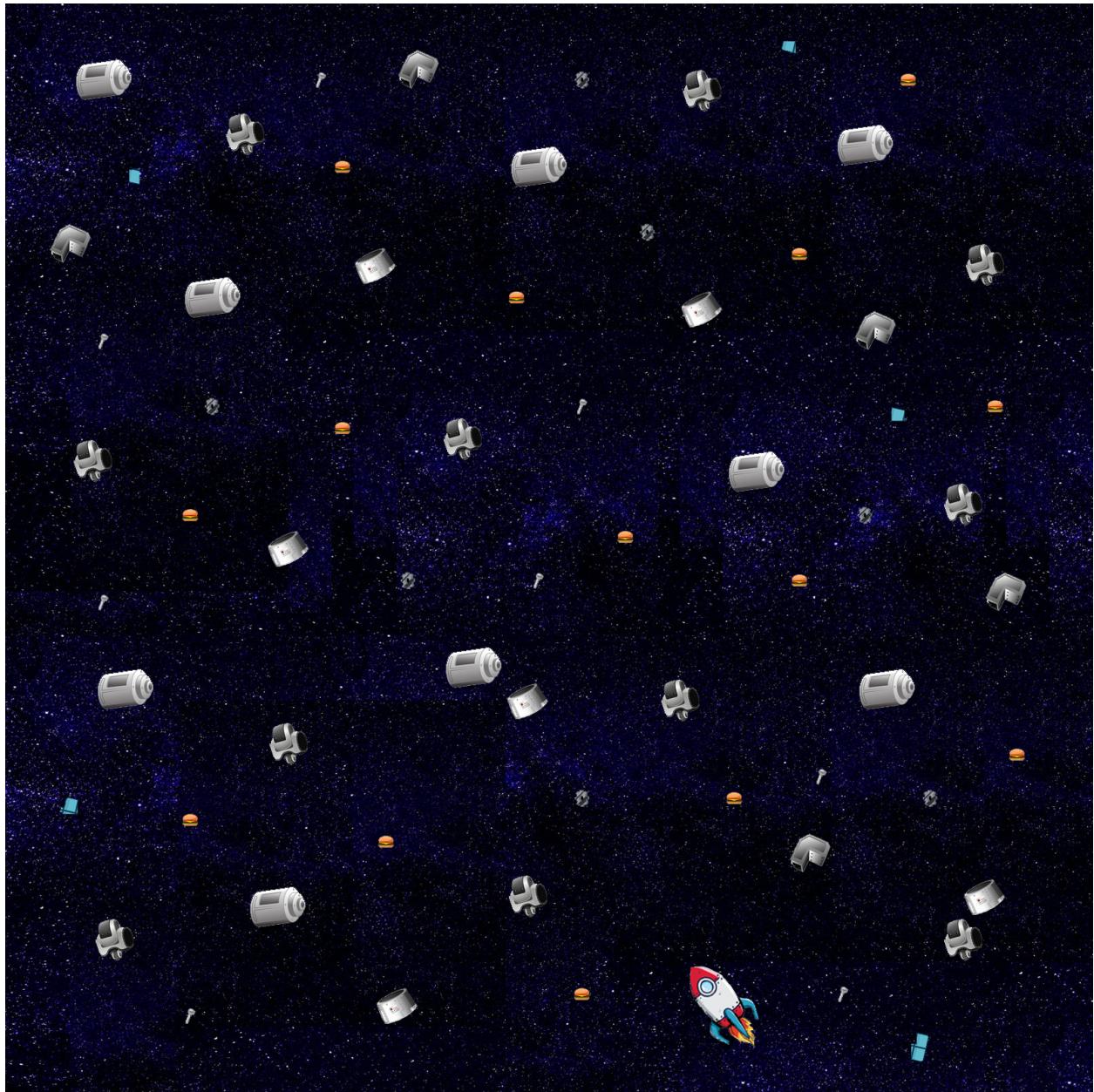
**Căpitánul:** El este protagonistul și singurul jucător. Personaj inspirat din viața reală, Matt Taylor sau cu numele său real, Matthew Graham George Thaddeus Taylor, este un astrofizician din Marea Britanie ce lucrează la ESA (European Space Agency), care este cel mai cunoscut pentru aterizarea pe Cometa 67P/ Churyumov-Gerasimenko din cadrul proiectului Rosetta.

### **Tabla jocului:**

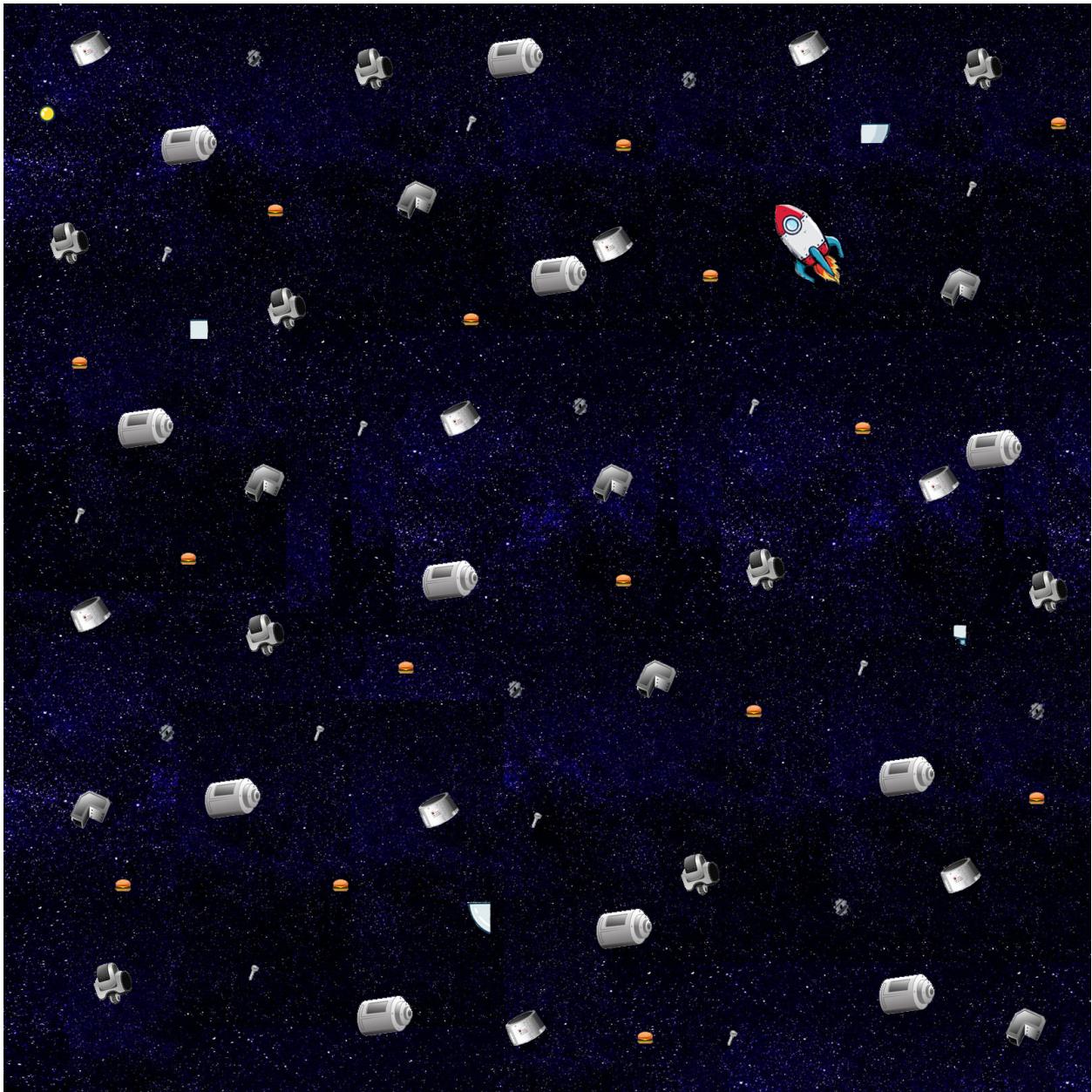
Imaginiile de jos sunt niște exemple de cum ar arăta harta finală, elementele sunt puse imitând Cosmosul real, adică împrăștiate datorită lipsei gravitației.

- ★ **Componente pasive:** sunt dalele cu stele (rol vizual, de background); cele care conțin componente din nava, precum: șuruburi, roți, 3 piese mai mari (rol de obstacole, șuruburile și roțile pot fi luate de jucător, dar ocupă spațiu) și nava spațială de la care pornește și la care trebuie să ajungă protagonistul)
- ★ **Componente active:** asteroizii sunt singurele elemente care pot omorî personajul, cu cât crește nivelul, cu atât crește și rapiditatea cu care se mișcă aceștia, astfel jucătorul fiind nevoie să se miște mai repede
- ★ **Structura tablei de joc:** Tabla prezintă o perspectivă ortogonală, fiind construită din dale de 32x32 pixeli. Este construită din 2500 de dale în total, fiind sub formă de pătrat. În afară de elementele enumerate mai sus, tabla conține și o bară, ce se află în partea de sus, care are rolul de a arăta jucătorului ce punctaj a acumulat câte piese a reușit să adune.

**Harta întreagă nivel 1:**



## Harta întreagă nivel 2:



## Mecanica jocului:

- ★ Game Points: pe traseu se pot aduna puncte, prin colectarea de mâncare
- ★ Comenzile/butoane utilizate: stânga-> D, dreapta-> A, față->W, spate->S, colectare->SPACE
- ★ User Interaction: fiind joc single player, nu există interacțiune cu alte personaje. Protagonistul interacționează cu piesele pe care le poate colecta (cele bune și rele), cu mâncarea care îi oferă puncte, cu nava unde trece la următorul nivel și cu asteroizii

**Game sprite:**

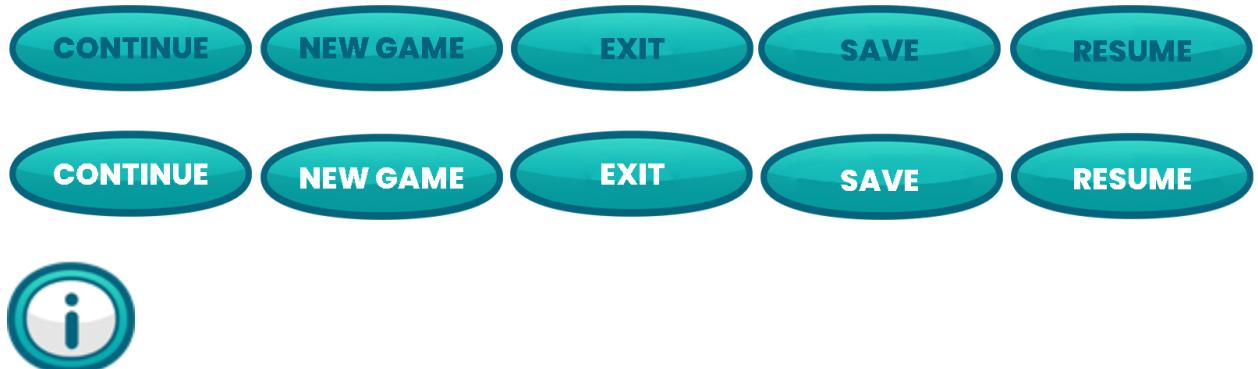
★ protagonistul:



★ Elementele hărții:



★ Meniul:



★ “Inamicul”:



#### Descriere fiecare nivel:

★ **Nivel 1:** presupune strângerea bucăților din panoul solar drept și aducerea acestora înapoi la navă, pentru a fi asamblate.

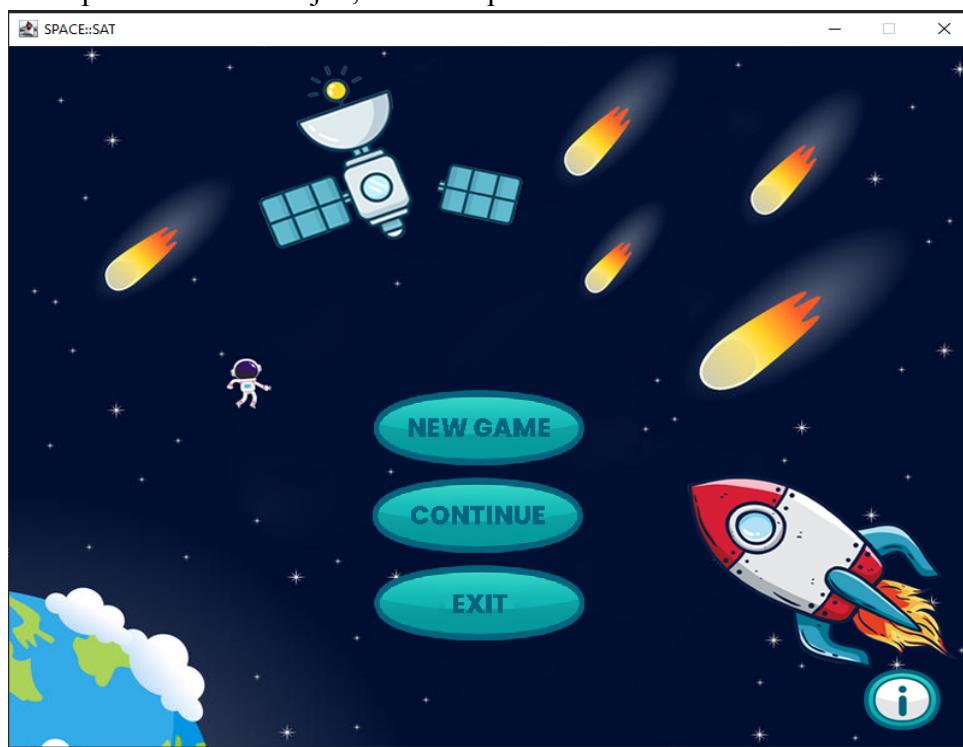
★ **Nivel 2:** traseul va fi asemănător, dar obstacolele și obiectele ce trebuie colectate vor fi plasate în alt mod. Va avea în vedere antena satelitului

Dificultatea va crește de la un nivel la altul, prin creșterea numărului de asteroizi care cad, o dată cu viteza lor.

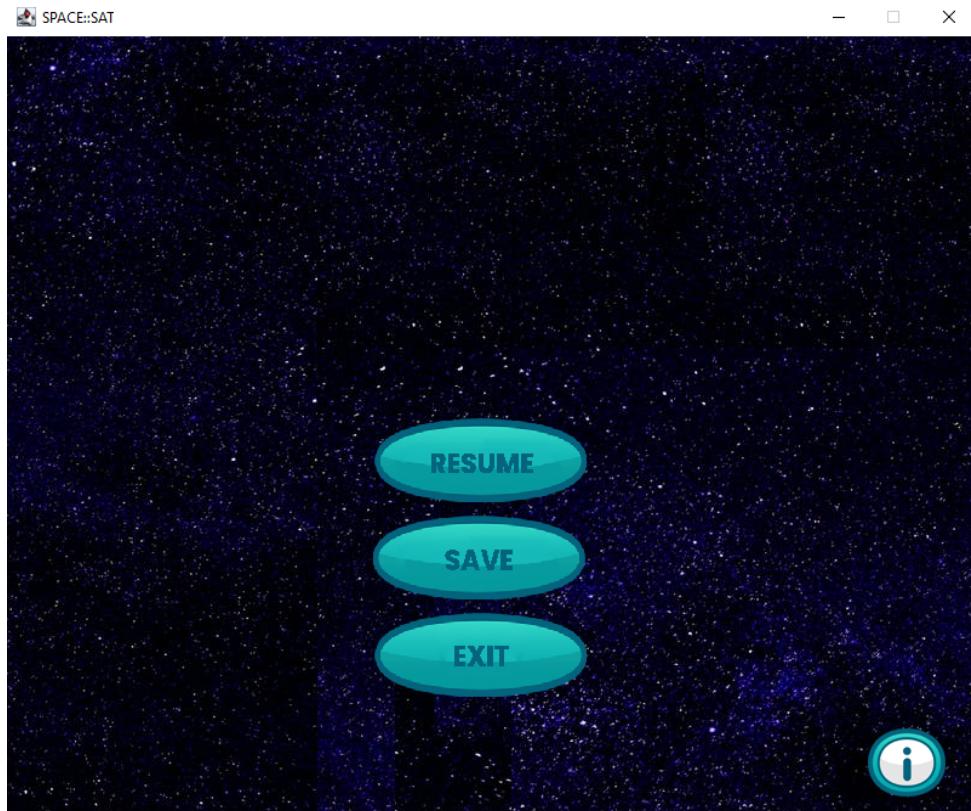
#### Descriere meniu:

Vor fi următoarele opțiuni:

- ★ New Game: pentru a începe un joc de la primul nivel
- ★ Continue: pentru a continua
- ★ Exit: pentru a ieși din joc (fără a salva progresul)
- ★ Save: pentru a salva în baza de date nivelul, punctajul și piesele adunate
- ★ Info: pentru a afla informații despre joc
- ★ Resume: pentru a reveni la joc, un fel de pause-resume



Pentru partea de meniu, am ales sa am diferite forme. Când se deschide jocul, intrăm în state-ul meniu, cel din poza de mai sus. Când suntem în joc și apăsăm tasta ESC, intrăm în state-ul de settings, care arata ca în poza de mai jos. Mai sunt două state-uri în care vor apărea butoanele de “new game” și “exit”.



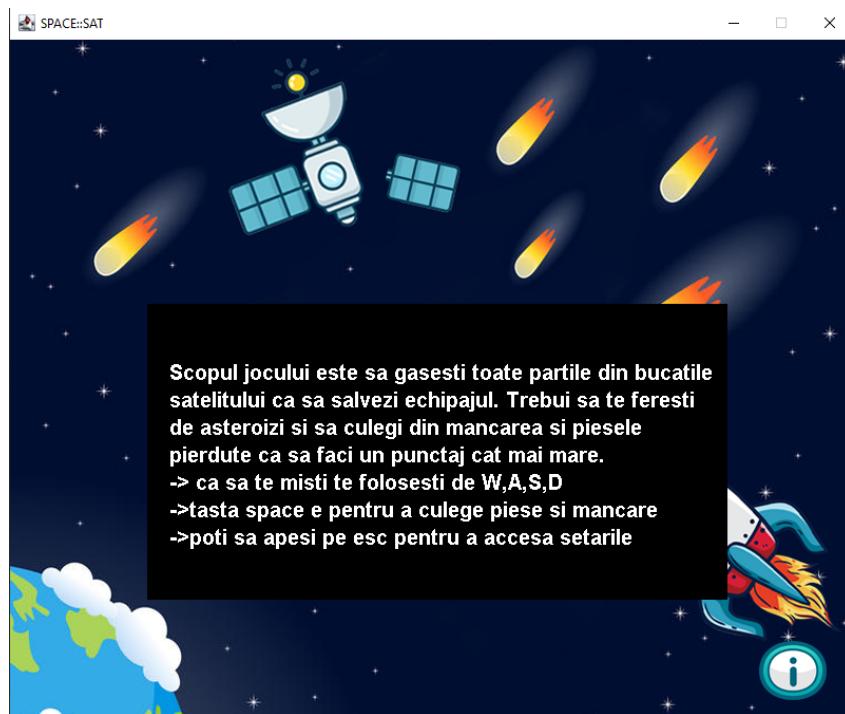
### **Metode de înfrângere/câștig:**

Câștigul se face dacă a trecut de fiecare nivel, găsind toate piesele satelitului. Pe traseu fiind și altfel de piese ce pot fi culese, playerul poate fi dus în eroare. Sunt 5 piese plasate pe traseu în fiecare nivel, o dată ce le-a colectat, playerul se duce la navă și este automat trimis la următorul nivel/în state-ul de câștig.

Înfrângerea se produce în momentul în care playerul intră în contact cu un asteroid, atunci se intră în state-ul de pierdut.

### Ipostaze din joc:

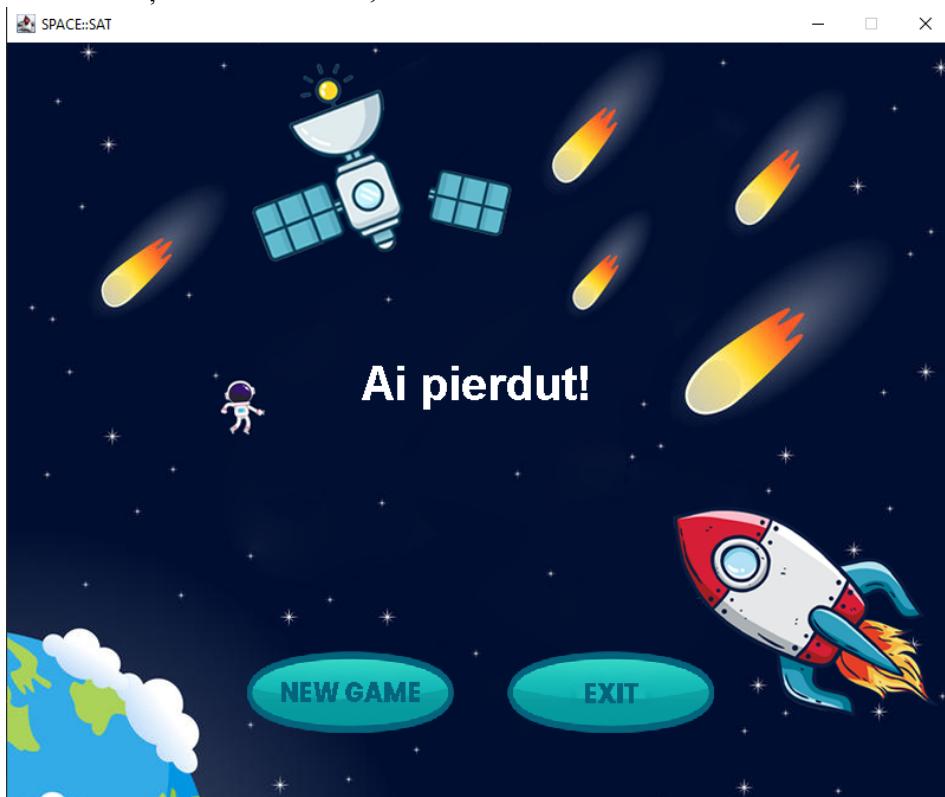
Jocul începe în state-ul de meniu, de unde utilizatorul poate să acceseze info



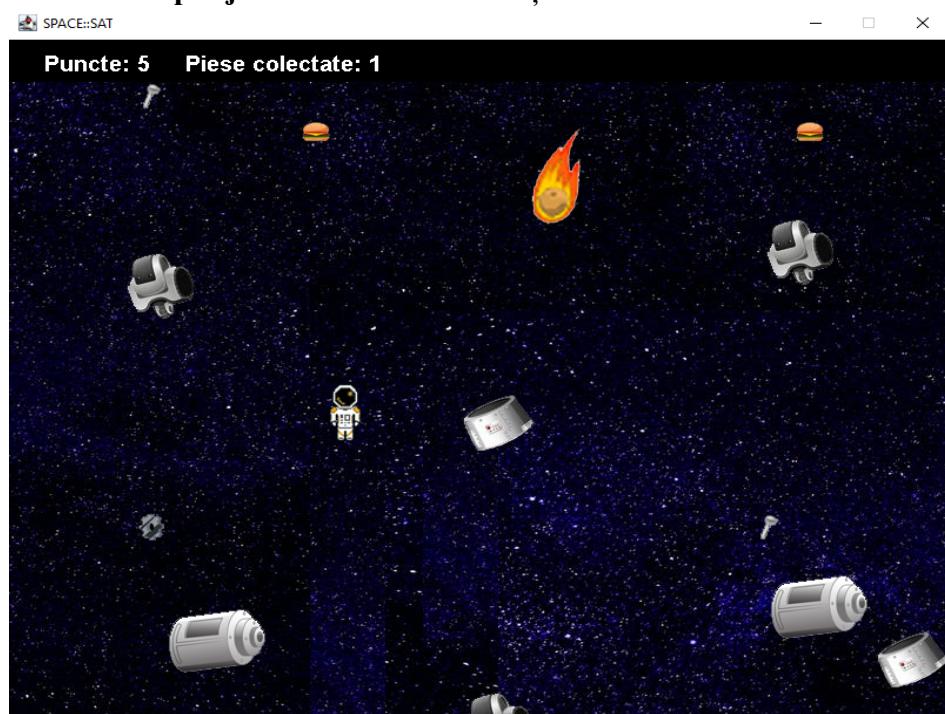
După care poate începe jocul, unde după ce culege cele 5 piese, trece la următorul nivel



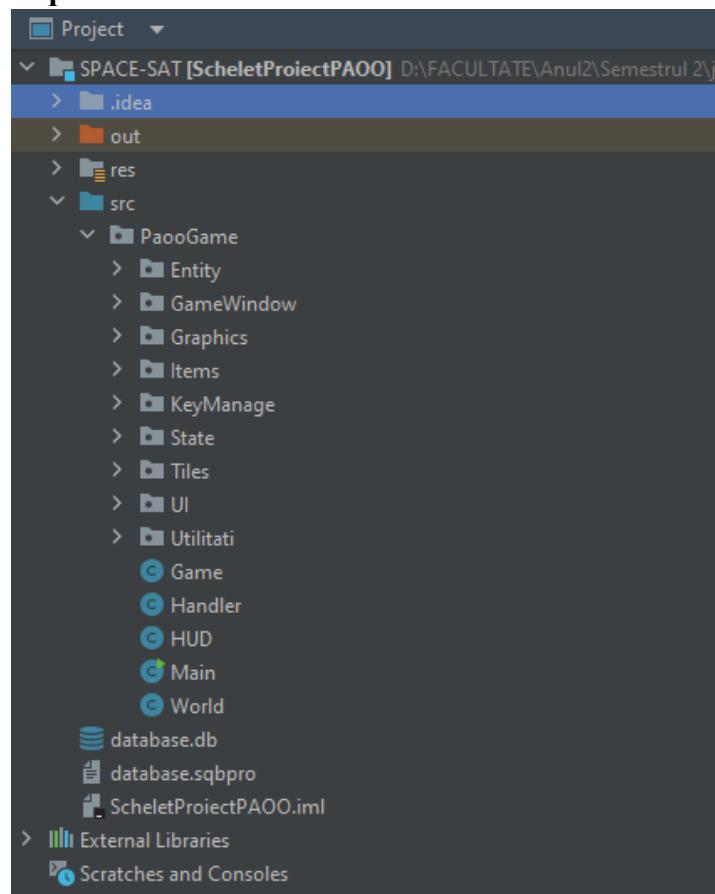
În cazul în care te lovește de un asteroid,



**Elemente de UI din timpul jocului sunt “Puncte” și “Piese colectate”:**



## Structura modulară a proiectului



## Baza de date:

Jocul realizează o conexiune cu SQLite, cu ajutorul lui DB Browser, unde avem o tabelă în care se salvează datele jocului înainte de apăsarea butonului SAVE. Tabela cu numele “State Save”, salvează punctajul (câte piese a cules pe drum), câte piese bune a cules (din cele 5) și nivelul la care a ajuns.

O privire din ansamblu:

The screenshot shows the Database Structure view in DB Browser for SQLite. The table "StateSave" is defined with the following schema:

```
CREATE TABLE "StateSave" ( "ID" INTEGER UNIQUE, "Nivel" INTEGER, "Punctaj" INTEGER, "Piese" INTEGER, PRIMARY KEY("ID" AUTOINCREMENT) )  
ID INTEGER  
Nivel INTEGER  
Punctaj INTEGER  
Piese INTEGER
```

Other tables listed in the structure include "sqlite\_sequence" and "sqlite\_stat1". Indices, Views, and Triggers are listed as 0.

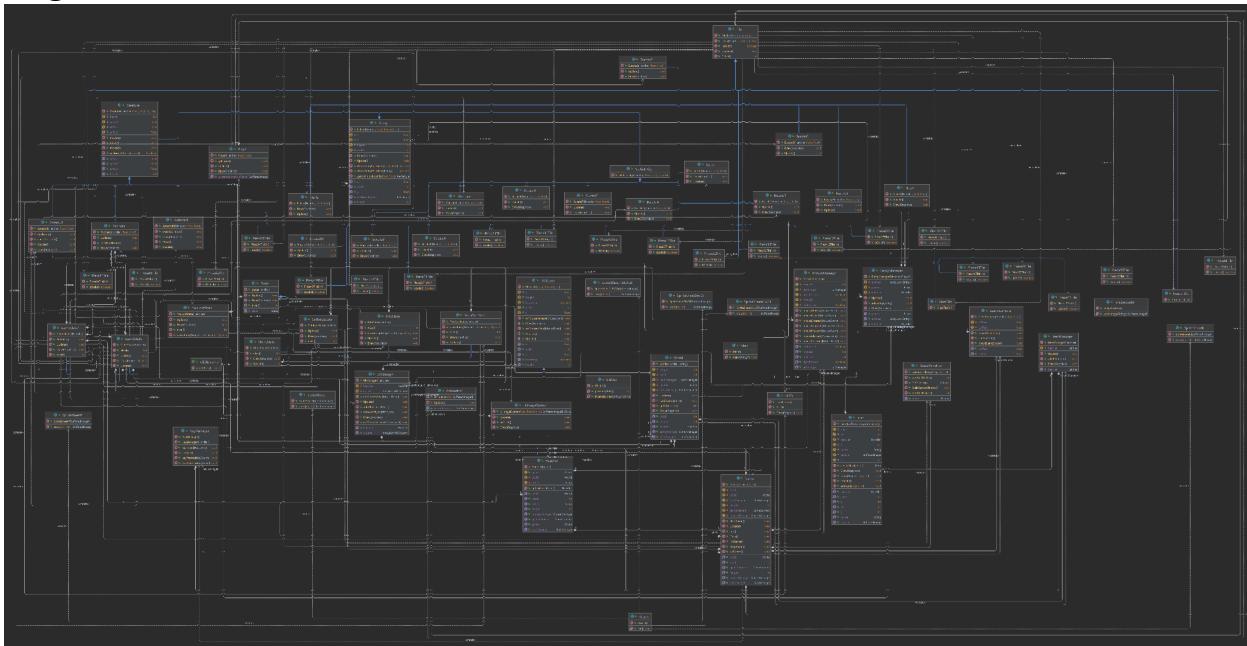
Tabela cu date ce se preiau automat

Database Structure      Browse Data

Table: StateSave

ID	Nivel	Punctaj	Piese
Filter	Filter	Filter	Filter
1	36	1	4
2	37	1	5
3	38	1	0
4	39	2	0
5	40	1	6

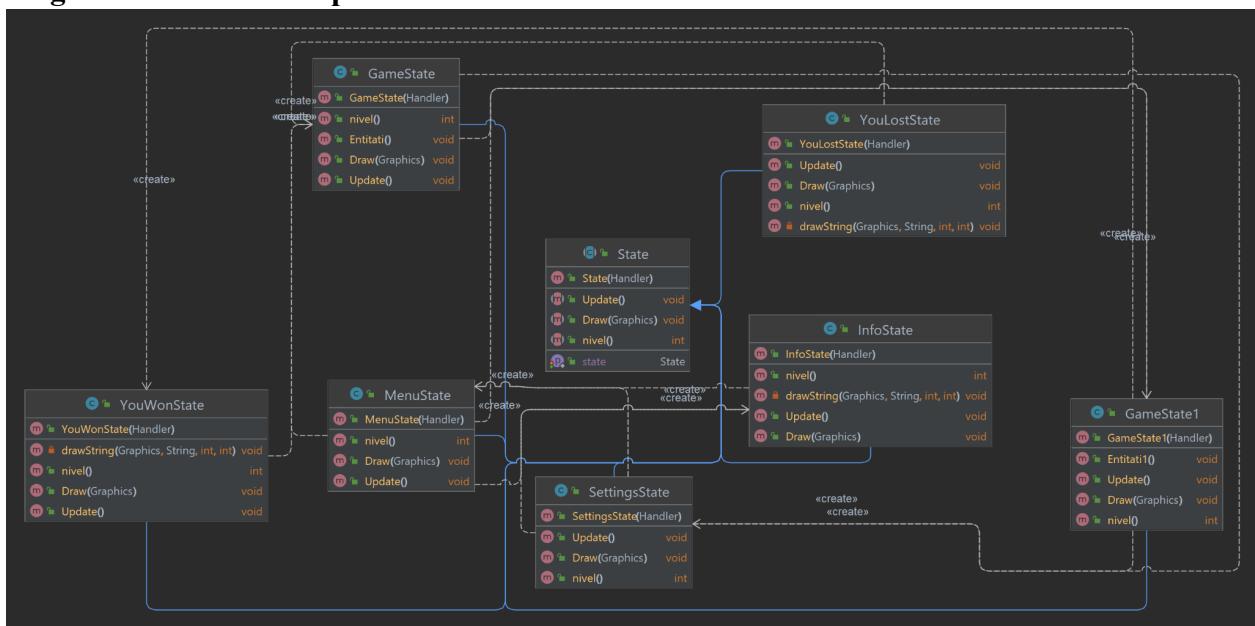
Diagrama UML:



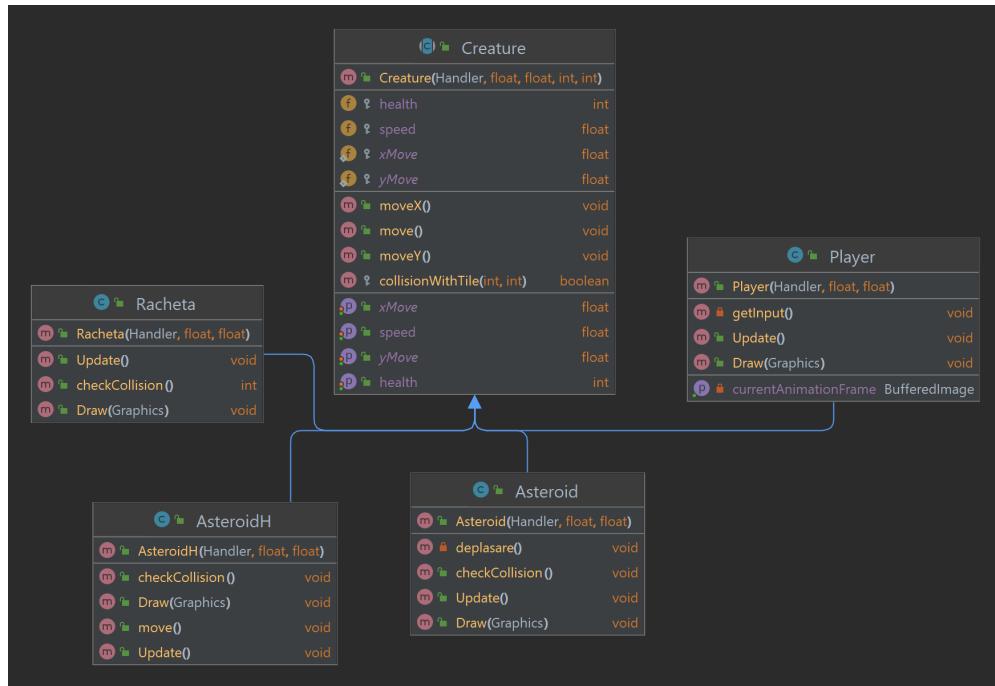
Pentru o mai bună vizionare a pozelor:

[https://drive.google.com/drive/folders/15WCTu5L4D\\_ohaMNQRH\\_0YJlc4gHFV4?usp=sharing](https://drive.google.com/drive/folders/15WCTu5L4D_ohaMNQRH_0YJlc4gHFV4?usp=sharing)

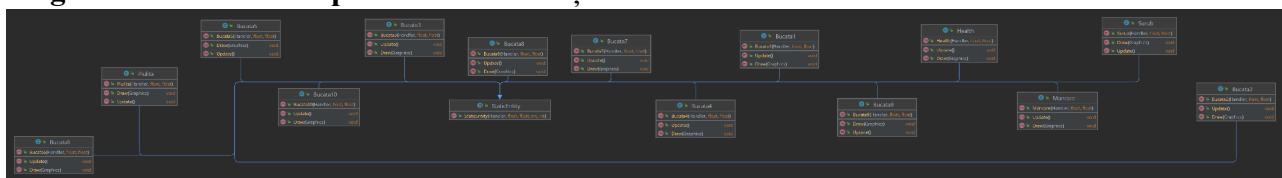
## Diagrama de clase corespunzătoare stărilor



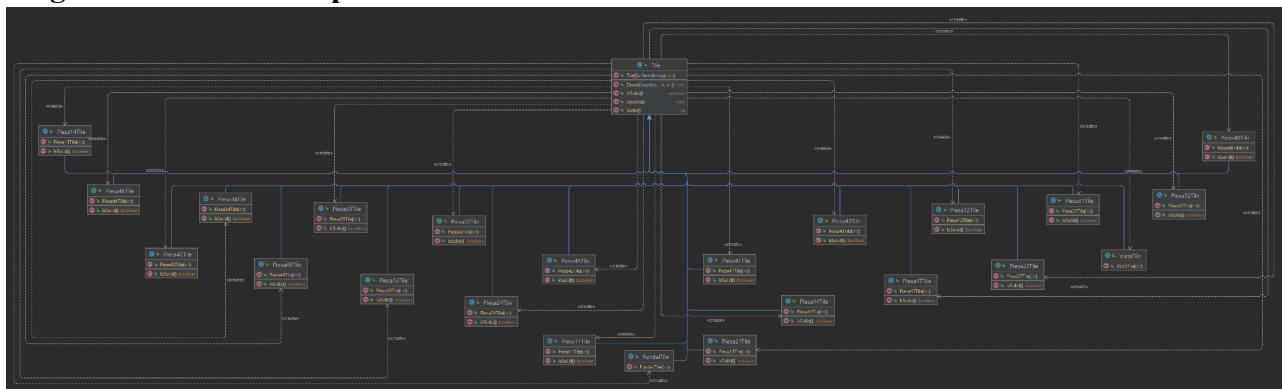
## Diagrama de clase corespunzătoare creaturilor (cea ce implică playerul, asteroidul și racheta)



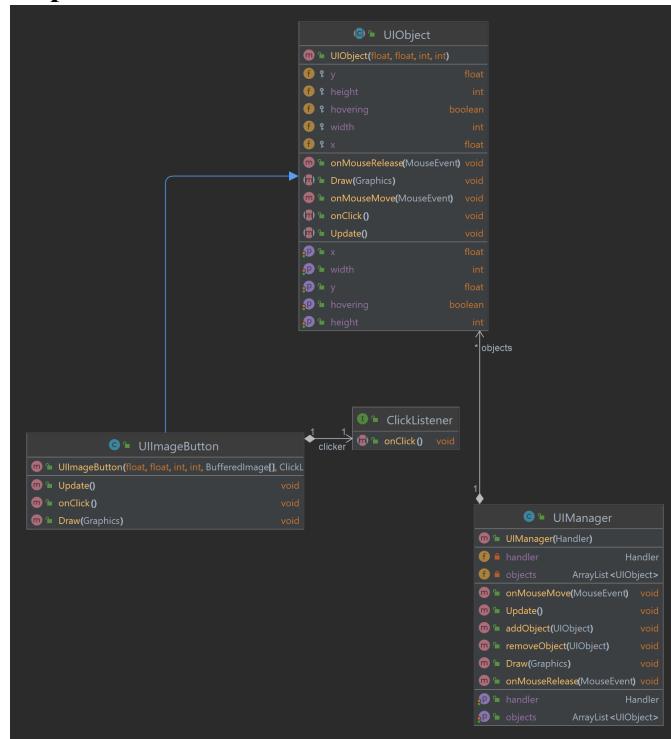
## Diagrama de clase corespunzătoare entităților statice



## Diagrama de clase corespunzătoare dalelor



## Diagrama de clase corespunzătoare User Interface



Printre cele mai importante clase se numără:

- ★ Handler: este o clasă ce gestionează elementele din clasa game și world, pentru a îmi fi mai ușoară apelarea metodelor din aceste clase în alte locuri. Pentru handler am utilizat şablonul de proiectare Singleton
- ★ KeyManager & MouseManager: sunt două clase care ne permit crearea unei legături între apăsarea tastelor W,A,S,D,SPACE,ESC și leftclick-ul de la mouse, pentru a mișca playerul, respectiv pentru a accesa butoanele din meniu. Acestea au implementate metodele publice keyPressed, keyReleased și mousePressed, mouserReleased, mouseMoved
- ★ World: este clasa care ne dă posibilitatea de a ne afișa harta, datorită metodei loadWorld, care primește un string. Cu aceasta pot citi dintr-un fișier text id-urile pe care le-am dat fiecărui tip de tile ca să le desenez pe harta și poziția de început a jucătorului
- ★ GameCamera: cu această clasă îmi pot fixa fereastra de 800x640px pe player, ca atunci când acesta se mișcă pe hartă, utilizatorul să poată vedea unde merge, altfel playerul ar fi ieșit din ecran
- ★ State: clasa abstractă care implementează şablonul de proiectare State, am clasele GameState, GameState1, MenuState, SettingsState, InfoState etc. care extind această clasă, iar în Game am request de a seta state-ul jocului prin metoda setState. Cu această clasă pot să modific state-ul jocului cum am nevoie

## Bibliografie:

- ★ Protagonistul e bazat pe o persoană reală, de aici am luat câteva informații de bază:  
[https://en.wikipedia.org/wiki/Matt\\_Taylor\\_\(scientist\)](https://en.wikipedia.org/wiki/Matt_Taylor_(scientist))
- ★ Aplicația Tiled pentru construcția hărții: <https://www.mapeditor.org/>
- ★ Pt imaginea asteroizilor:  
<https://pngtree.com/element/down?id=NTIyNzUzMMA==&type=1&time=1647627178&token=YzlmNTImMDFmYmQyZWE0OGE0YTg2NDgzNTE4NDk2Y2M=>
- ★ Pt mâncare:  
[https://pngtree.com/freepng/hand-drawn-cute-cartoon-burger-with-food-elements\\_4056721.html](https://pngtree.com/freepng/hand-drawn-cute-cartoon-burger-with-food-elements_4056721.html)
- ★ Personajul: <https://images.app.goo.gl/EjfP8HektVwZWrb39>
- ★ Pentru editurile imaginii jocului și a mai multor elemente, am utilizat Adobe Photoshop
- ★ Alte informații:  
<https://codereview.stackexchange.com/questions/73959/multiple-random-falling-objects-animation-in-java>