# **COLEGIUL NAȚIONAL DE INFORMATICĂ „TUDOR VIANU”**

LUCRARE DE ATESTAT

-pixel Star Wars-

Profesori coordinatori:

Coman Isabela

Toca Livia

Autor

Popa Petru

Clasa a 12-a G

2022-2023

**CUPRINS**

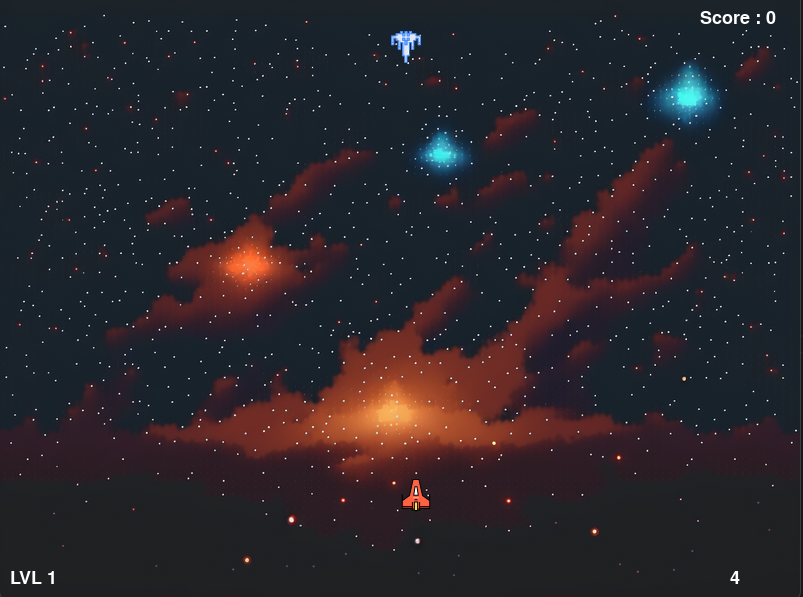
* Capitolul 1: Introducere
* Capitolul al 2-lea: Utilizarea aplicației
* Capitolul al 3-lea: Proiectarea aplicației
* Capitolul al 4-lea: Bibliografie

**CAPITOLUL 1**

**INTRODUCERE**

Aplicatia “pixel Star Wars” este un joc clasic de tipul Space Shooter in care jucatorul trebuie sa distruga cat mai multe nave inamice. Navele au o tematica inspirata de cele din filemele Star Wars.

Ea pune la dispozitie o nava jucatorului care poate fi imbunatatita pe parcursul luptei impotriva inamicilor, lupta ce nu are final. Am ales aceasta tema pentru ca imi plac jocurile clasice de tipul 8 bit.



**CAPITOLUL 2**

**UTILIZAREA APLICATIEI**

**RESURSE NECESARE**

Niciuna. Ruleaza pe orice calculator.

**INSTALARE**

Aplicația poate fi rulată prin fișierul executabil **„maine.exe”** aflat pe stick si care poate fi copiat pe orice calculator. Pentru a putea rula corect, este necesară copierea folderului **„pixelStarWars”**.





**PREZENTARE PENTRU UTILIZATOR**

Este un joc in care jucatorul trebuie sa distruga cat mai multe nave inamice in timpul alocat fiecarui nivel (10 secunde). Nivele cresc in dificultate pe masura ce mor din ce in ce mai multi inamici, nava jucatorului se inbunatateste dar si ajunge in noi galaxii.

**Inamici**



**Nave Jucator**





**CONTROALE**



Space – nava impusca cu lasere

Left Arrow – nava se misca la stanga

Right Arrow – nava se misca la drepata

A – nava se misca in continuu la stnga

D – nava se misca in continuu la dreapta

ESC – iesire din aplicatie

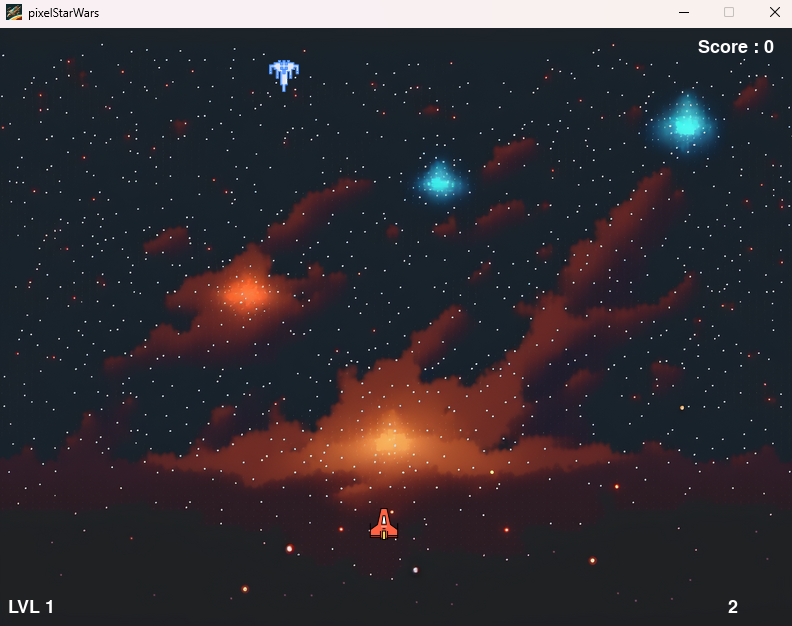
**PROGRES**

Score - numarul de inamici omorati

LVL - nivelul la care se afla jucatorul

Timp – in dreapta jos, cat timp mai este pana cand trebuie omorati toti inamicii, cand devine 0 jocul s-a terminat

Scor



Timp

Nivel

**CAPITOLUL 3**

**PROIECTARE**

Aplicatia este realizata folosind mediul de programare Microsoft Visual Studio Code, limbajul utilizat fiind Python. Organizarea aplicatiei este facuta pe baza tehnicilor moderne de programare utilizate in crearea aplicatiilor. Am scris codul utilizand clase cu scopul de a face un program usor de citit, modificat si imbunatatit.

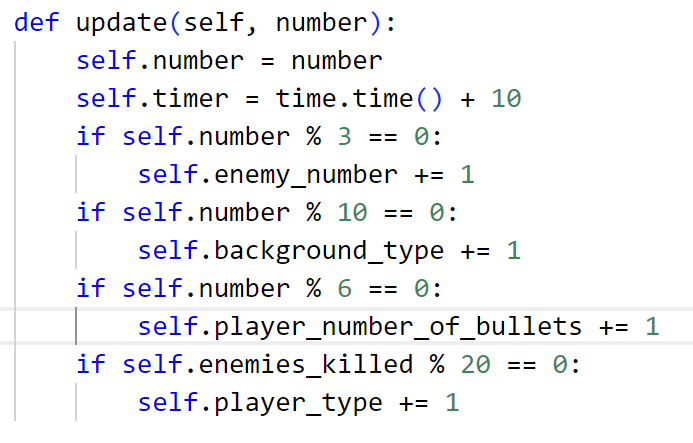
In plus au fost utilizate resurse create cu programe bazate pe inteligenta artificiala (stable diffusion, midjourney), altele au fost create manual iar altele luate de pe internet. Aceste sunt memorate in folderul “Assets”, fiecare resursa fiind plasata in folderul corespunzator: “BG”, “ Enemies”, “Lasers”, “Players”.

Ca principii de programare, am utilizat programare orientata pe obiecte.

Principalele clase si functiile lor sunt:

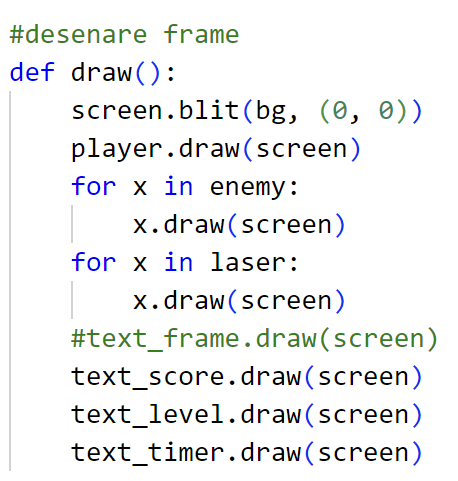
* Player
  + Initializare
  + Move
  + Draw
* Enemy
  + Initializare
  + Move
  + Draw
* Laser
  + Initializare
  + Move
  + Draw
* Text
  + Initializare
  + Update
  + Draw
* Level
  + Initializare
  + Update
  + Draw

Clasa level este cea prin care se fac modificarile la toate nivelele, in functia update jucatorul primeste imbunatatiri la nava dar si numarul de inamici si peisajul se schimba.

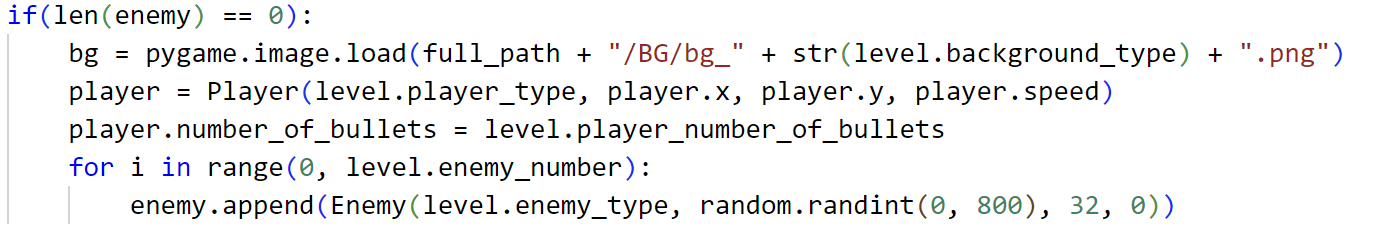


In main.py se afla codul pentru joc. L-am impartit in mai multe parti, fiecare avand functia ei.

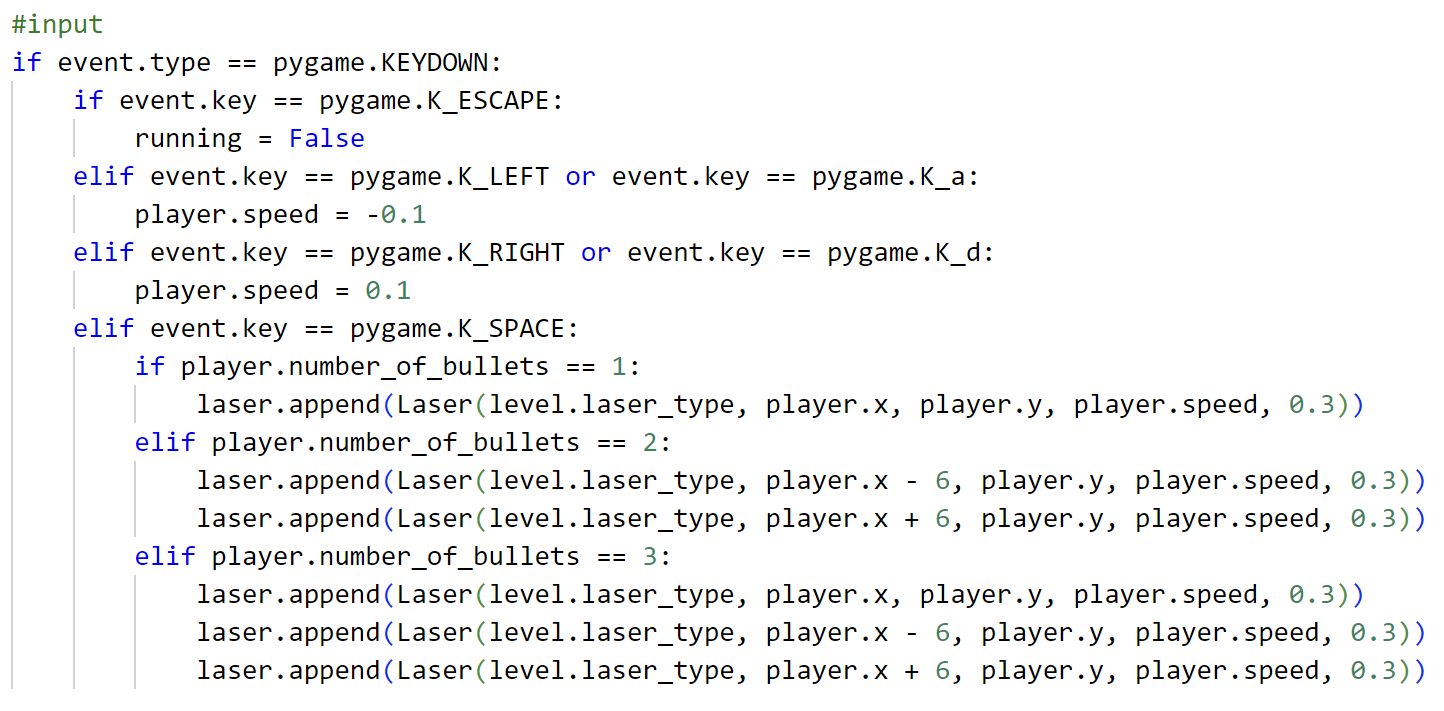
Functia draw face update la ecran pentru fiecare frame, apeland functiile de draw ale fiecarui obiect utilizand functia de blit din libraria pygame.



In loop-ul principal mai intai se verifica daca nivelul a fost castigat, caz in care se da udate la nivel si se adauga noi inamici in lista enemy.

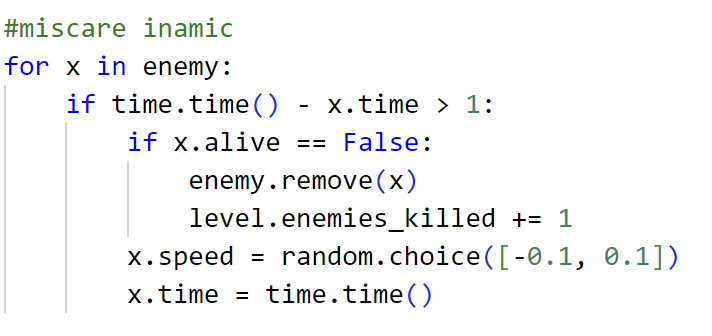


Dupa aceasta este verificat input-ul de la jucator si sunt create laserele care spre deosebire de alte jocuri pastreaza inertia navei prin transmiterea vitezei jucatorului.

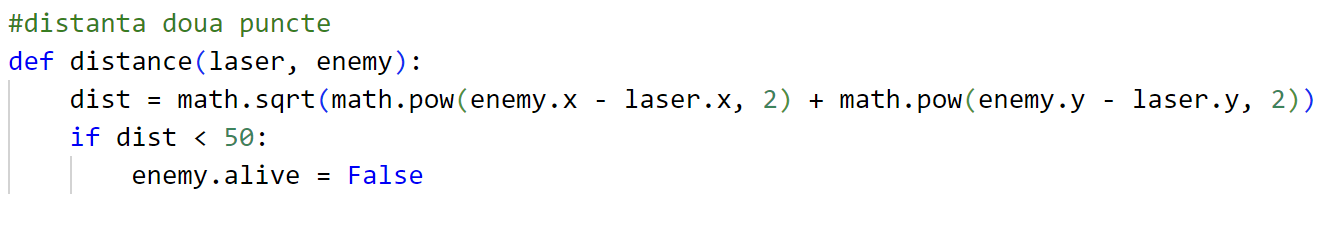


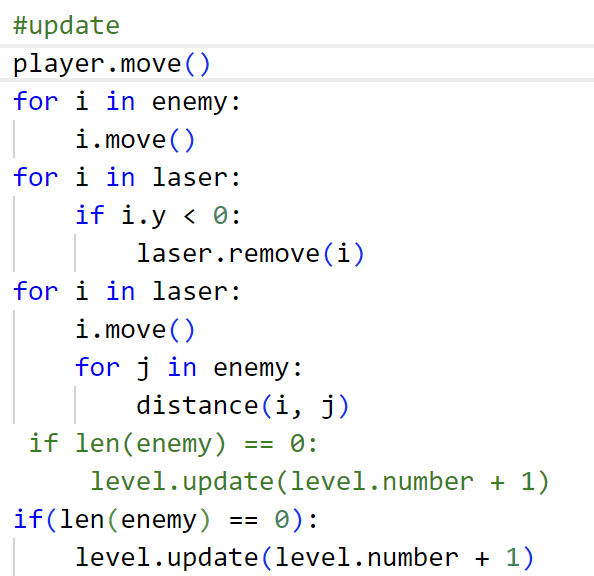
Randomizarea miscarii inamicilor este necesara pentru a nu fi un joc plictisitor. Am realizat-o prin crearea unei sanse de 50% de

a-si modifica directia de mers la fiecare secunda.



La finalul while-ului se verifica ce inamici au fost omorati prin functia distance care calculeaza distanta dintre fiecare laser si fiecare inamic si daca este indeajuns de mica inamicul este distrus. Pentru ca libraria pygame utilizeaza procesorul pentru a face toate caluclele (in loc de placa grafica), este necesar sa tin numarul de calculi cat mai mic posibil, motiv din care laserele sunt scoase imediat ce au iesit din ecran. Dupa aceasta se hotaraste daca nivelul s-a incheiat si se apeleaza functia draw pentru a desena frame-ul.





**CAPITOLUL 4**

**BIBLIOGRAFIE**

GeeksforGeeks

W3schools

StackOverflow