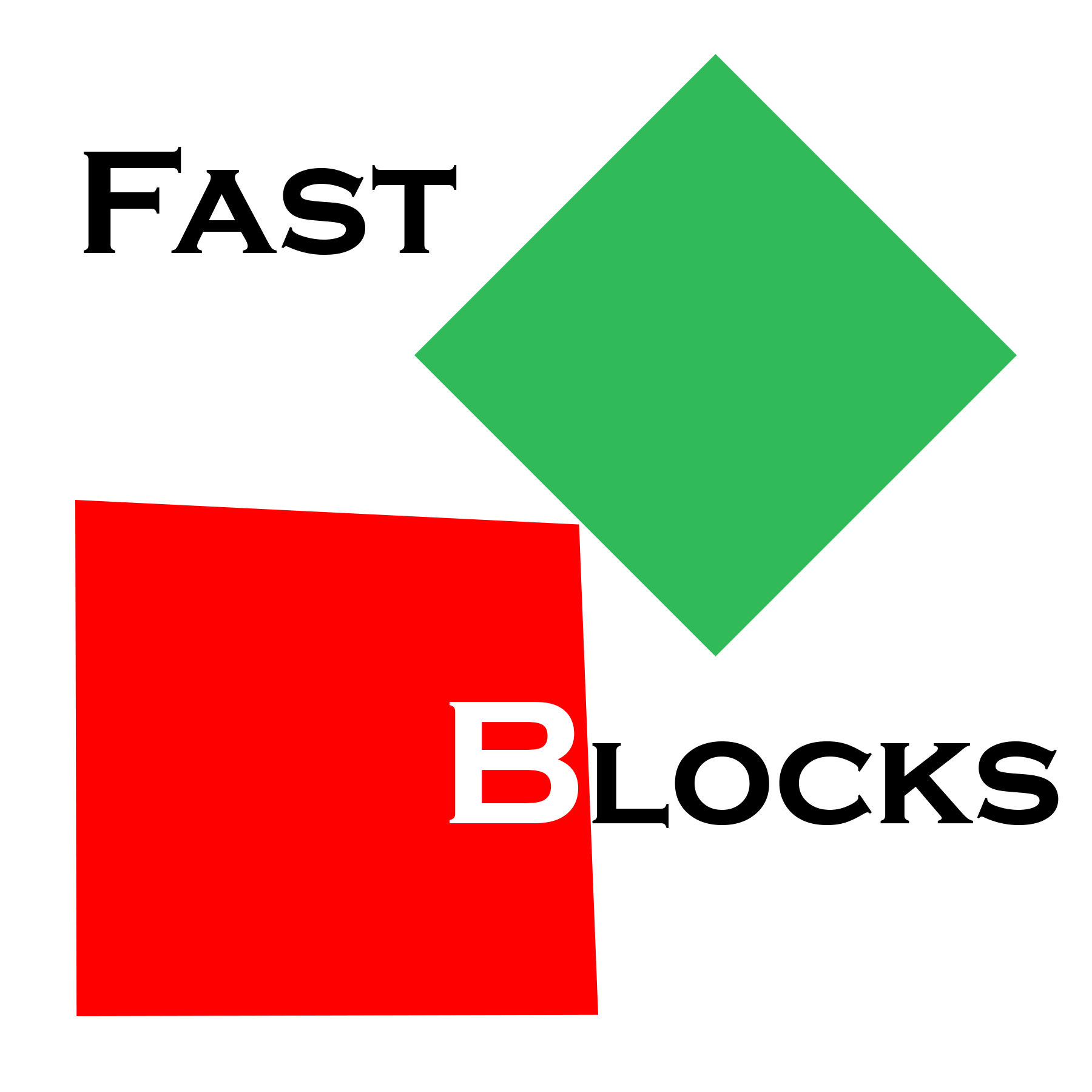
Lucrare de atestat

Disciplina informatică

2021



**CUPRINS**

**I. Introducere..............................................................4**

1. Alegerea proiectului

2. Specificații software

3. Scopul aplicației

**II. Descrierea aplicației................................................7**

4. Fereastra principală

5. Simularea ciocnirii

6. Graficul coliziunii corpurilor

7. Calcularea momentului coliziunii

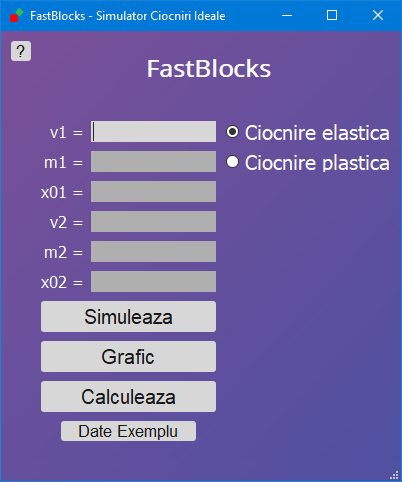
**III. Concluzii...............................................................13**

**I. Introducere**

**1.Alegerea proiectului**

Proiectul finalizat constă în realizarea unei aplicații în limbajul de programare Python. Aplicația permite utilizatorului să vizualizeze ciocnirea între două corpuri cu masa și viteza variabile, graficul vitezei corpurilor și aflarea momentului și poziției coliziunii lor.

Datorită pasiunii comune atât pentru informatică, cât și pentru fizică, am ales ca temă a proiectului ciocnirea dintre două corpuri, un fenomen important care aparține de mecanica clasică, ramură a fizicii. Tema a fost valorificată printr-o interfață simplă și intuitivă.

Ciocnirea reprezintă interacțiunea de scurtă durată dintre două sau mai multe corpuri sau particule, în urma căreia se modifică vitezele și direcțiile de mișcare ale acestora. Într-un astfel de proces, impulsul total se conservă. Ciocnirile a două corpuri pot fi plastice, însemnând o deformare permanentă a corpurilor, sau elastice, implicând lipsa deformării corpurilor. Aplicația FastBlocks abordează ciocnirea a două corpuri în condiții ideale, lipsite de forțe exterioare, precum frecarea cu aerul sau suprafața pe care acestea alunecă și gravitația.

Deși a reprezentat un obstacol semnificativ datorită lipsei de experiență cu sintaxa, am decis încă de la început realizarea aplicației în limbajul Python. Python este un limbaj de programare dinamic multi-paradigmă care pune accentul pe curățenia și simplitatea codului. Python servește ca limbaj pentru tipul de programare orientată pe obiecte, dar permite și programarea imperativă, funcțională sau procedurală. Alegerea noastră a fost influențată în principal de actualitatea limbajului, abilitățile în limbajul Python fiind un atu căutat pe piața muncii.

**2. Specificații software**

Pentru a putea accesa aplicația FastBlocks, utilizatorii trebuie să aibă atât versiunea 3.6 sau una mai nouă a limbajului Python instalată, cât și patru librării specifice limbajului. Instalarea se poate face de pe site-ul oficial al organizației, <https://www.python.org/>.

Librăriile necesare sunt:

**PyQt5** este o librărie utilă la crearea interfețelor grafice. Aceasta conține o aplicație similară cu Visual Studio, numită QtDesigner, care, prin plasarea elementelor grafice pe un plan, generează codul necesar afișării lor. Codul generat nu este în totalitate utilizabil, așadar a trebuit reorganizat.

Librăria **Matplotlib** permite crearea animațiilor interactive care depind de ecuații matematice și prelucrarea datelor numerice.

**NumPy** este o librărie a limbajului Python folosită în prelucrarea vectorilor. Librăria oferă o alternativă listelor din forma de bază a limbajului Python, care nu sunt deloc eficiente din punctul de vedere al timpului de executare.

**Pygame** este o librărie alcătuită din mai multe module, folosită la crearea animațiilor interactive complexe, precum jocurile video.

**3.Scopul aplicației**

Programul oferă utilizatorilor un mod inedit de a interacționa cu ciocnirea dintre două corpuri.

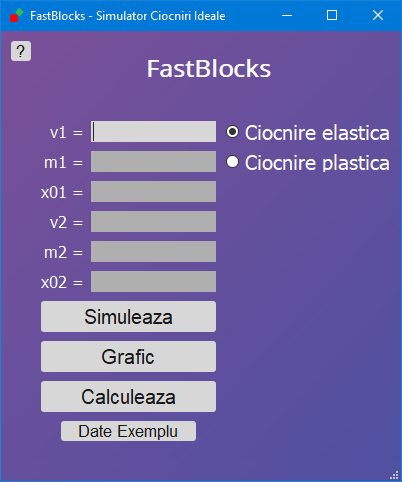
Aplicația constituie o unealtă utilă atât pentru elevii, cât și pentru profesorii de fizică din învățământul gimnazial și liceal, având la dispoziție un nou mod de a interacționa cu, respectiv prezenta un fenomen fundamental din mecanica clasică.

Utilizatorii beneficiază de un timp redus de rulare, datorită eficienței limbajului Python.

**II. Descrierea aplicației**

Programul FastBlocks a cunoscut un lung proces de dezvoltare, din cauza lipsei de experiență în programarea în limbajul Python.

**4. Fereastra principală**

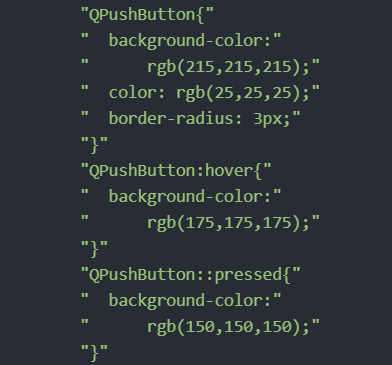


La momentul deschiderii aplicației, utilizatorul este întâmpinat de o fereastră principală atractivă și usor de înțeles. Din punct de vedere grafic, elementele principale ale programului sunt șase câmpuri care permit introducerea datelor, etichete cu scop informativ și trei butoane care permit valorificarea datelor introduse în diferite moduri. De asemenea, butonul "Date Exemplu" introduce automat un set de date pentru o demonstrare mai rapidă a funcționalității programului.

În colțul din stânga sus a ferestrei există un buton informativ destinat tuturor utilizatorilor care atenționează despre dependența simulării ciocnirii de valoarea datelor introduse.

Pentru o organizare mai bună, am împărțit codul în două fișiere. "MW\_Setup.py" conține toate elementele grafice și design-ul acestora, în timp ce "Main Code.py" reprezintă partea funcțională a programului.

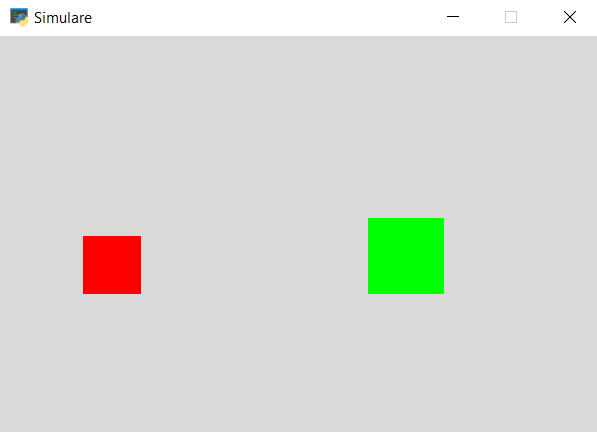
Design-ul programului este unul propriu, realizat cu ajutorul limbajului CSS (Cascading Style Sheets).



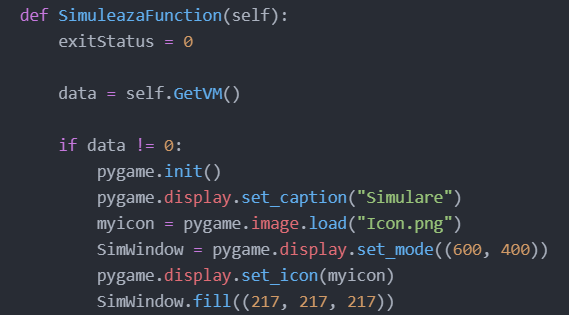
Design-ul unui buton

**5. Simularea ciocnirii**

Prin intermediul unui buton „Simulează” se deschide o fereastră care afișează o simulare în două dimensiuni, implementată prin librăria Pygame. Din punct de vedere vizual, în fereastra deschisă apar două pătrate de culori diferite. Dimensiunea, direcția de mișcare, viteza și poziția lor iniţială variază în funcție de datele introduse în câmpurile de preluare a datelor. În funcție de tipul de ciocnire selectată, simularea va înfățișa ori ciocnirea plastică, ori ciocnirea elastică dintre pătrate.

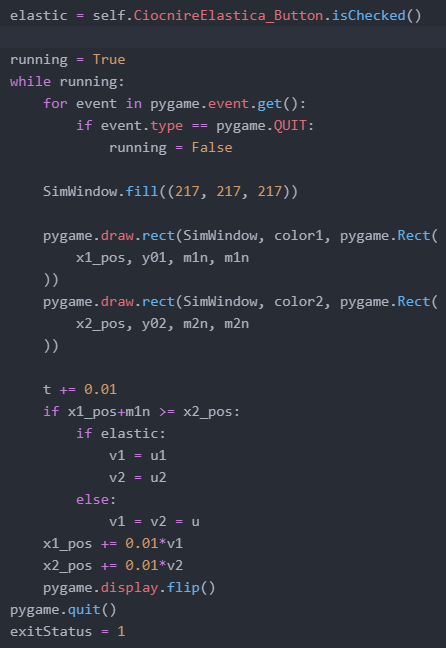


Fereastra "Simulează"

****

Crearea ferestrei

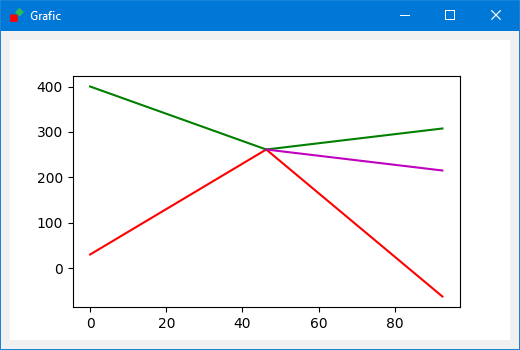
Funcția destinată ciocnirii preia datele din câmpurile de introducere a datelor. În funcție de datele preluate și cu ajutorul librăriei Pygame, se construiește simularea grafică, care se reactualizează la fiecare parcurgere a funcţiei "while". La momentul ciocnirii, funcția recalculează vitezele.



Functia "Simuleaza"

**6. Graficul poziției corpurilor**

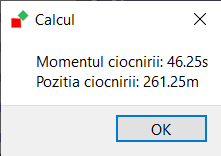
La apăsarea butonului "Grafic" se va deschide o fereastră care afișează graficul poziției celor două pătrate din plan în funcție de timp. Graficul poziției corpurilor a fost implementat și afișat cu ajutorul librăriilor Matplotlib și Numpy, care permit gestionarea și manipularea mai ușoară a datelor și construirea interfeței grafice necesară prezentării acestora. Graficul prezintă un design simplist și ușor de înțeles pentru orice utilizator.



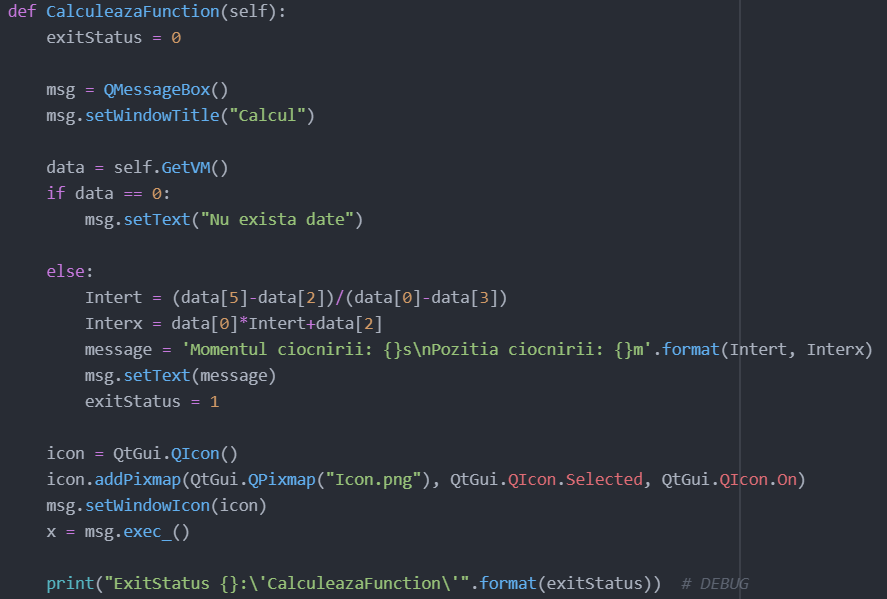
Interfața graficului

**7. Calcularea momentului coliziunii**

Acționarea butonului "Calculează" determină afișarea momentului și poziției exacte a ciocnirii celor două corpuri prin intermediul unei ferestre mici.



Fereastra "Calcul"

****

Funcția "Calculează"

**III. Concluzii**

Am realizat acest program în limbajul Python. Pe parcursul proiectului am dobândit abilități noi și utile, precum împărțirea unui cod în două fișiere, programarea orientată pe obiecte, introducerea elementelor de design grafic în cod și folosirea mai multor librării în cadrul aceluiași proiect, abilități care, deși nu intră în programa școlară, sunt esențiale pentru un programator.

Punctele forte ale programului sunt timpul redus de rulare, design-ul atractiv și interfața plăcută. Punctul slab al programului este lipsa de portabilitate, întrucât mărimea ferestrei nu se adaptează la ecranul utilizatorului, iar pentru a rula aplicația este necesară instalarea limbajului Python și a celor patru librării.

Pe parcursul acestei documentații am incercat să prezentăm cât mai bine proiectul și să explicăm atât modul de utilizare al aplicației, cât și logica din spatele codului și structurii acestuia.

**Bibliografie**

* <https://www.python.org/>
* <https://www.w3schools.com/>
* <https://doc.qt.io/>
* <https://pypi.org/project/PyQt5/>
* <https://matplotlib.org/>
* <https://www.pygame.org/news>
* <https://numpy.org/>