

# Simularea interacțiunilor fizice

student: Cristian-Andrei Sandu

coordonator: prof. dr. ing. Costin-Anton Boiangiu

Septembrie 2018

Universitatea POLITEHNICA  
din București



# Motivație

- aprofundarea unui subiect practic (care nu e studiat în facultate și care e folositor în industrie)
- dorința de a implementa o aplicație completă pentru a testa diferiți algoritmi și metode numerice

# Problema

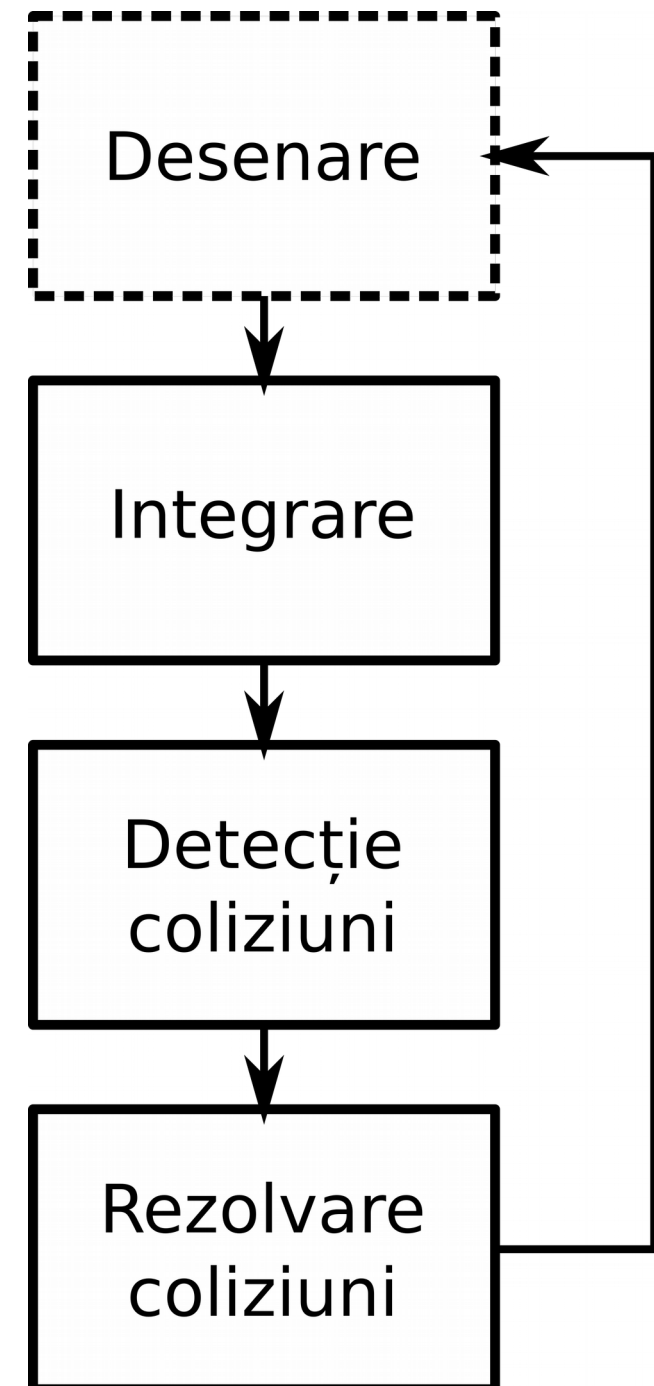
- simularea realistă a interacțiunilor mecanice dintre corpuri 3D (aproximarea cât mai bună a fenomenelor fizice din realitate)
- 2 componente principale ale soluției:
  - motorul de fizică
  - interfața cu utilizatorul

# Motorul de fizică


- componentă esențială a oricărui
  - joc video
  - simulator științific
  - software pentru animații sau efecte speciale
- oferit de obicei ca middleware (pentru a fi integrat în aplicații)
- logica din spatele interacțiunilor dintre corpuri
- aproximare a fenomenelor fizice reale

# Motorul de fizică (2)

- funcționarea unui motor de fizică



# Integrarea obiectelor

- actualizarea stării obiectelor 
  - poziția
  - orientarea
- calcul accelerație → actualizare viteză → noua poziție
- diferite metode numerice (Euler, RK4, Verlet etc.) de precizie și complexitate diferită

# Detecția coliziunilor

- 2 etape: preliminară și exactă
- algoritmi de complexitate diferită
- Se stabilesc punctele de coliziune și parametrii acestora:
  - punctele de interpenetrare maximă
  - direcția și distanța de separare

# Rezolvarea coliziunilor

- răspuns cât mai realist
- Trebuie rezolvată atât interpenetrarea, cât și vitezele corpurilor de după ciocnire.
- gruparea punctelor de coliziune în funcție de perechea de obiecte de care aparțin
- caching pentru contacte persistente
- rezolvare iterativă, prin aplicare de impulsuri normale sau tangențiale (în cazul frecărilor)



# Interfața cu utilizatorul

- interfață grafică + mouse + tastatură
- controlul obiectelor din simulare
- alterarea parametrilor de simulare
- controlul camerei
- încărcarea / salvarea de scene

# Rezultate

- aplicație **C++** cu **OpenGL**
- simularea **interacțiunilor mecanice**
- **corpuri solide 3D**
- dezvoltare personală

# Rezultate (2)

- integrare **Euler implicită**
- filtrarea corpurilor posibil aflate în coliziune cu **OBB-uri**
- algoritmi **GJK** și **EPA** pentru calcularea informațiilor de coliziune
- rezolvare coliziuni prin metoda **impulsurilor secvențiale**
- interfață grafică folosind **Dear ImGui**
- demo-uri pentru funcționalitățile aplicației

# Evaluare

- corpuri suportate: cub, paralelipiped dreptunghic, sferă, cilindru, capsulă
- 3 scenarii de test:
  - **stive** stabile de **maxim 4 obiecte**
  - simulare stabilă cu **maxim 64** de obiecte
  - **plan înclinat + frecări**
- interfață puternică și ușor de folosit
- aplicație **Windows** single-threaded
- memorie totală utilizată: ~**45.5 MB**
- dimensiune heap: ~**5900 KB**

# Dezvoltări ulterioare

- implementare algoritmi mai eficienți pentru detecția de coliziuni (arbori BSP, ierarhii de volume încadratoare)
- optimizarea lucrului cu heap-ul
- suport pentru mai multe forme 3D
- mai multe tipuri de constrângeri fizice
- suport pentru Linux
- reparare bug-uri

# Cuvinte cheie

- corpuri solide 3D
- simulare a fenomenelor reale
- coliziuni
- volume încadratoare
- frecări
- impulsuri secvențiale
- interfață grafică
- C++
- OpenGL