

Fizička simulacija igrice

SV 27/2020 - NEMANJA DUTINA

SV 18/2020 - MILICA SLADAKOVIĆ



O čemu je riječ ?

Osnova – igrica Leo's fortune

- Popularna igrica na IOS platformama.
- Stvorenje Leo, krećući se po podlozi sa raznim preprekama, dolazi do logičkih zagonetki čijim rješavanjem nastavlja kretanje do krajnjeg cilja.
- **Mi se nismo bavili logikom, već fizikom ove igrice.**
- Modelovali smo ravan teren sa poligonskim preprekama i dvije vrste metaka.

Pravila igre

- Kretanje kuglice kontroliše korisnik, tako što ubrzava (\rightarrow), usporava (\leftarrow) i skače (\uparrow).
- Loptica se kreće po Njutnovim zakonima kretanja (linearno kretanje i rotacija)
- Postoje dvije vrste metaka – linijski i bombice (njihovo kretanje je ograničeno)
- Udarom od linijske metke, kuglica se usporava (održanje impulsa – neelastičan sudar)
- Udarom od bombice, umire
- Na podlozi se nalaze poligonske prepreke od koje se odbija, ako ih ne preskoči (elastični sudar)

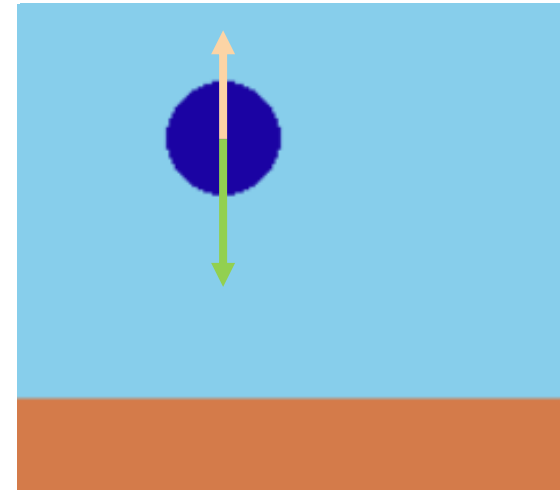
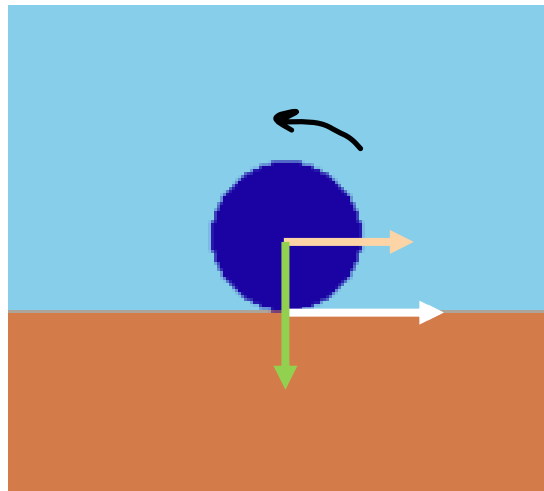
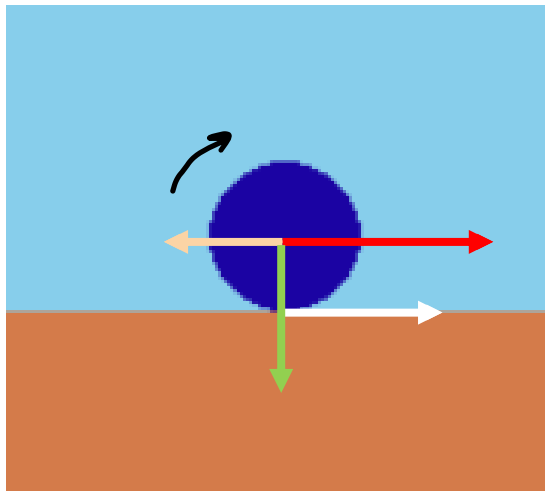
Elementi implementacije

- Fizičke veličine i sva kretanja - Milica
- Kolizije i rezultati kontakata - Nemanja

Fizika

Njutnovi zakoni kretanja

- Kretanje po podlozi – sila koju zadaje korisnik, inercija
- Kretanje u vazduhu – skok
- Na tijelo uvijek djeluje sila trenja, otpor vazduha i gravitacija.



RK4

- Iz brzine dobijamo trenutni položaj kuglice
- Brzinu računamo RK4 metodom
- $dv_{axis} = \frac{\Sigma F}{m} dt \quad dr_{axis} = v_{axis} dt$
- Cilj RK metoda je da se izvodi parametrizuju
- Koristili smo RK4 jer se pokazala najefikasnijom i preciznom.

Kolizije

KOLIZIJE

- KAKO BISMO MOGLI DA OBEZBEDIMO PONAŠANJE OBJEKATA KOJE ODGOVARA NJIHOVOM PONAŠANJU U REALNOM SVETU, POTREBNO NAM JE DA OBEZBEDIMO NJIHOVO MEĐUSOBNO DELOVANJE.
- OVO DELOVANJE IZMEĐU OBJEKATA, POSTIŽEMO DETEKCIJOM KOLIZIJA I KOREKCIJOM POLOŽAJA OBJEKATA NA OSNOVU DELOVANJA KOLIZIJA.



1. DETEKCIJA KOLIZIJA

Detekcija kolizija predstavlja proces otkrivanja kolizija i sastoji se iz dve faze:

1. *Broad phase*

2. *Narrow phase*

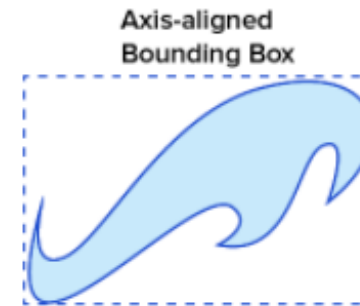
Broad phase je faza detekcije kolizije u kojoj otkrivamo parove objekata koji su u potencijalnoj koliziji.

Narrow phase je faza detekcije kolizije u kojoj proveravamo da li su potencijalni parovi objekata stvarno u koliziji.

1.1. BROAD PHASE

Ova faza podrazumeva pronalazak parova objekata koji se potencijalno nalaze u koliziji.

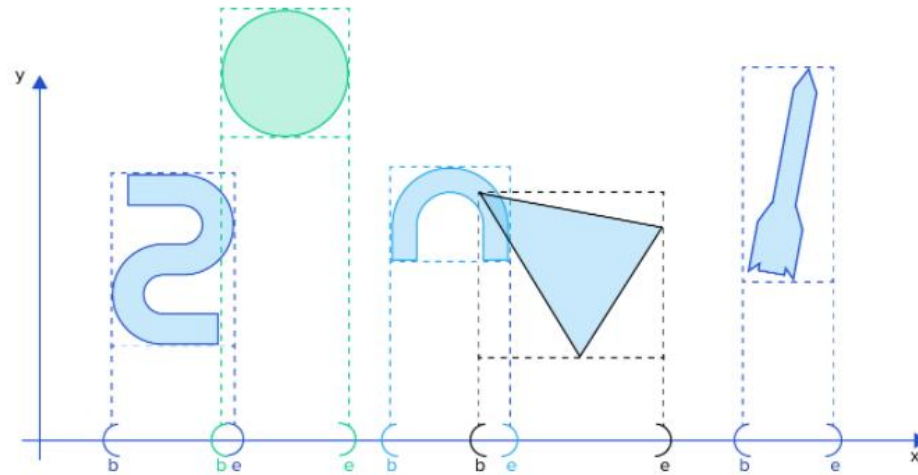
Za otkrivanje potencijalnih kolizija, koristili smo Axis-Aligned Bounding Box (AABB) mehanizam.



Ovaj mehanizam podrazumeva pronalazak minimalnog pravougaonika koji u sebi sadrži svaki objekat, te se potencijalne kolizije proveravaju koristeći novonapravljene pravougaonike. Provera da li se dva pravougaonika seku, mnogo je lakša od provere za komplikovanije oblike, te se samim tim provera olakšava.

1.1.1. SWEEP AND PRUNE

Uvođenjem AABB sistema smo umnogome olakšali proces traženja potencijalnih kolizija. Međutim, ovaj proces još uvek podrazumeva provere za svaka dva AABB objekta, što je dosta spor proces.



Proces ubrzavamo algoritmom za podelu prostora koji nazivamo *sweep and prune*.

Ovaj algoritam podrazumeva podelu prostora na intervale u kojima se nalaze pojedinačni objekti, te proveru kolizije samo ukoliko se intervali nekih objekata seku.

1.2. NARROW PHASE

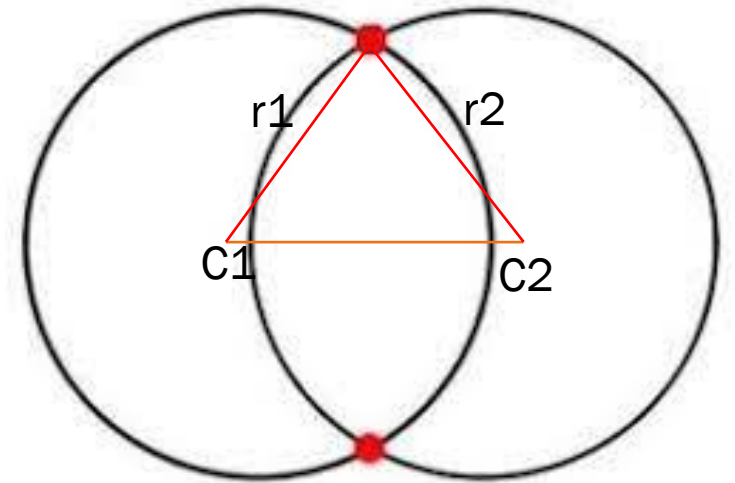
Ova faza podrazumeva proveru kolizije za svaki od parova objekata koji se potencijalno nalaze u koliziji.

Faza je podeljena na tri slučaja:

1.2.1. Duž na krug – proveru da li krug i duž imaju zajedničku tačku rešavanjem sistema jednačina

1.2.2. Krug na krug – proveru da li se centri krugova nalaze na manjoj udaljenosti od zbira poluprečnika

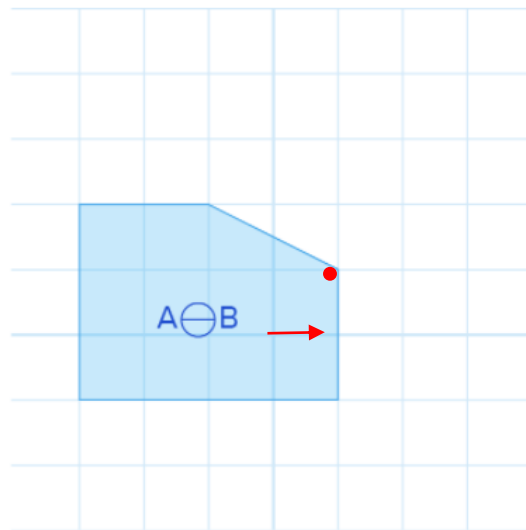
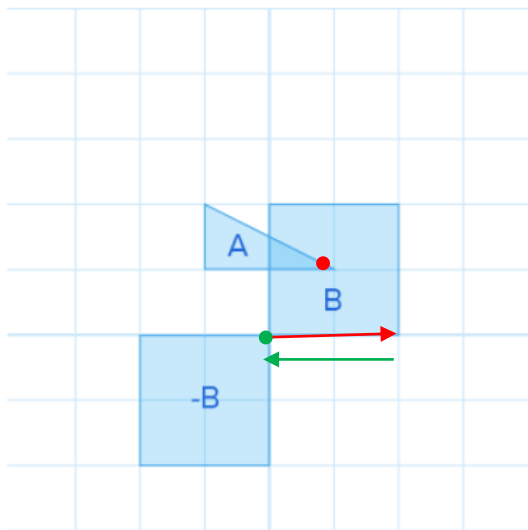
1.2.3. Poligon na krug – GJK + EPA algoritam



GILBERT-JOHNSON-KEERTHI (GJK) ALGORITAM

Algoritam koji na osnovu Minkowski razlike određuje da li je došlo do kolizije ili ne.

Na osnovu support funkcije pronalaze se tačke na Minkowski razlici koje formiraju simplex. Ukoliko se centar koordinatnog sistema nalazi unutar tog simplexa, objekti se seku. Ukoliko se centar ne nalazi unutar simplexa, objekti se ne seku.



GJK algoritam koristi support funkciju koja racuna tacku na Minkowski razlici za zadati vektor.

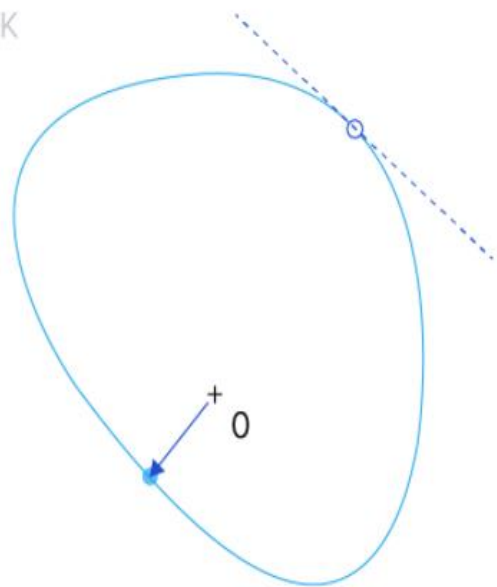
EXPANDING POLYTOPE ALGORITHM (EPA)

EPA algoritam se koristi u paru sa GJK algoritmom kako bi se odredio vektor koji predstavlja pravac i intenzitet sudara, tj. koliko bi se jedan od objekata morao pomeriti kako sudara ne bi bilo.

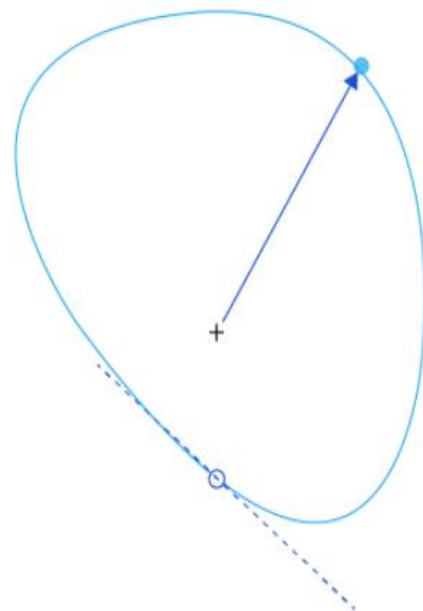
Algoritam koristi simplex koji je dobijen GJK algoritmom za traženje tog vektora.

Ovaj algoritam proširuje simplex dok ne dođe do tačke koja pripada Minkowski razlici a koja je najbliža centru koordinatnog sistema. Vektor određen centrom i tom tačkom je traženi vektor.

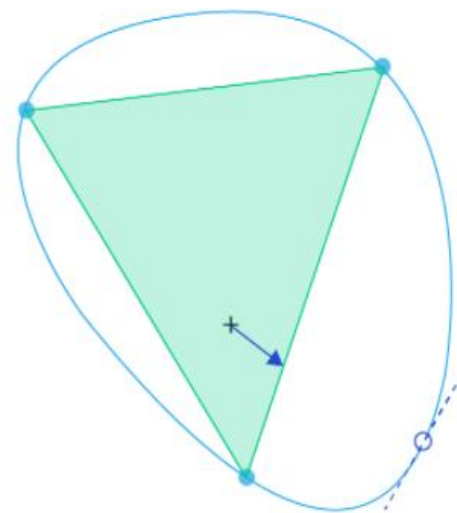
1. GJK



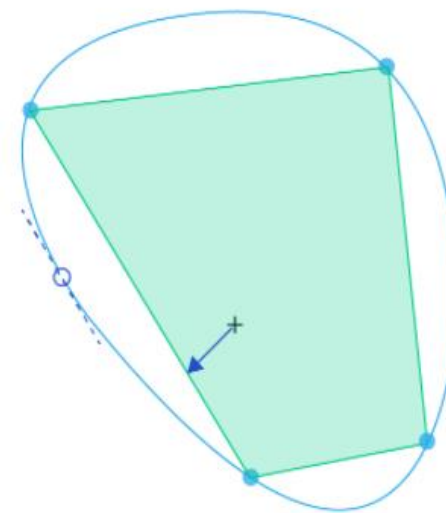
2. GJK



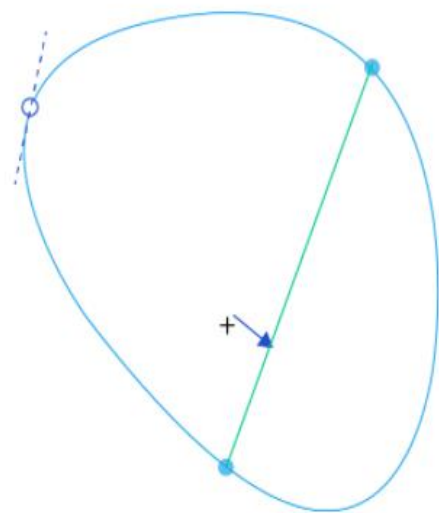
5. EPA



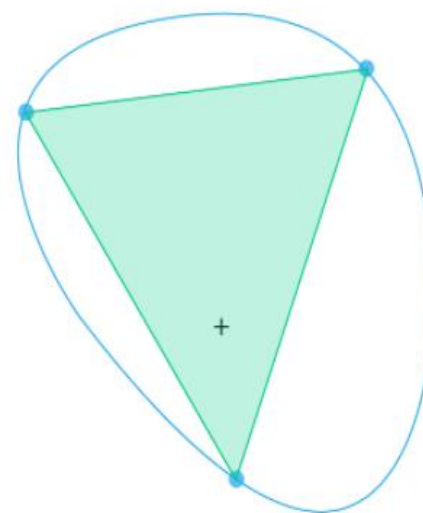
6. EPA



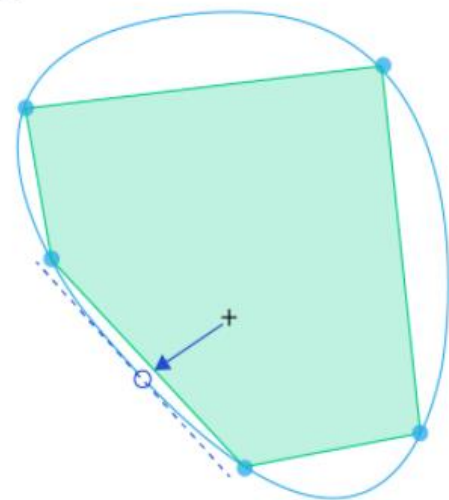
3. GJK



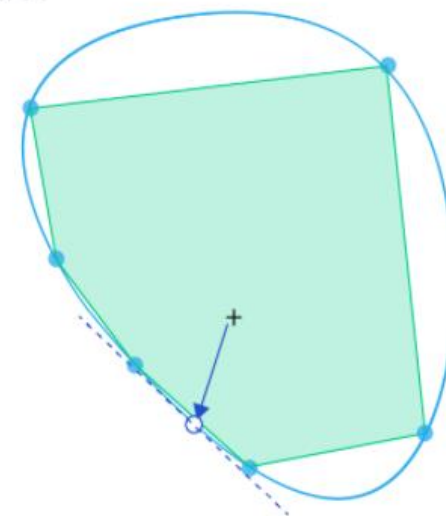
4. GJK



7. EPA



8. EPA



Zaključak

Nedostaci

- Usporeenje u skoku – veliki broj iteracija prilikom kojih se u svakoj traže potencijalne kolizije
- Rješenje – više radne memorije ?