

Implementacija 2D simulacije boćanja

Olivera Radovanović SV46/2021

Kratak opis problema

Projekat se sastoji od implementacije 2D simulacije boćanja. Potrebno je obezbediti korišćenje kinematike sa linearnim kretanjem, detekciju sudara, kao i ograničeno kretanje.

Pravila igre

Igra se sastoji od pravougaonog terena koji je ograničen, tako da kugla ne može da izađe iz njega. Svaki igrač ima po 3 kugle. Igra se igra dok jedan od igrača ne sakupi 13 poena. Na početku igrač po izboru baca malu specijalnu kuglu, a zatim jednu od svoje tri kugle. Igrači naizmenično bacaju svoje kugle, dok svih 6 kugli igrača ne bude na terenu. Igrač koji ima kuglu najbližu specijalnoj kugli dobija poen za svaku kuglu koja je bliža specijalnoj kugli u odnosu na najbližu kuglu protivnika (maksimalno 3). Pri bacanju kugli moguće je pomeriti svoju, tuđu ili specijalnu kuglu prilikom sudara sa bačnom kuglom.

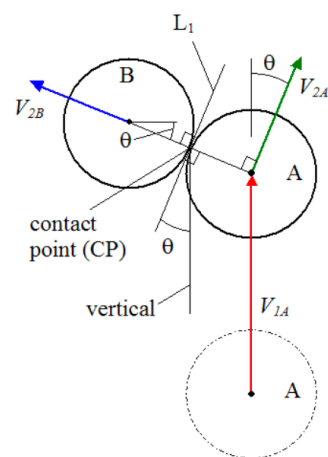
Detaljan opis problema

Kinematika sa linearnim kretanjem će biti korišćena za kretanje kugli po stazi, koristeći RK4 metodu.

Za detekciju sudara biće korišćena uprošćena geometrija za otkrivanje sudara, kao i prostorno indeksiranje - algoritam sweep and prune.

Sudar kugle sa drugim kuglama predstavlja sudar kruga sa krugom. Uvek će se, pre nego što se sudar dogodi, samo jedna kugla kretati (ona koja je bačena), dok će ostale koje se nalaze na terenu biti u stanju mirovanja. Pretpostavićemo da između kugli nema trenja i da su njihove mase jednake.

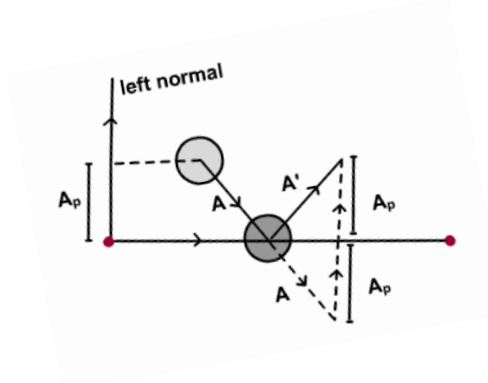
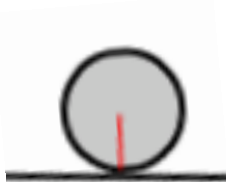
Sudar će se dogoditi ako rastojanje između dve kugle bude jednako zbiru poluprečnika tih kugli. Kada se sudar dogodi, kako su kugle istih masa, ugao između njihovih novih pravaca kretanje biće prav.



Sudar kugle sa ivicom staze predstavlja sudar kruga sa duži.

Sudar će se dogoditi ako rastojanje između centra kruga i duži bude jednak poluprečniku kugle. U tom slučaju krug se obija od duži i to pod istim uglom pod kojim je udario u nju.

Pri sudaru dolazi do gubitka impulsa kugle, pa ona usporava.



Programski jezik i biblioteke

Za implementaciju ovog projekta biće korišćena programski jezik Python i biblioteke pygame, numpy, math.

Primeri gotovih rešenja

Rešenja sličnih problema moguće je pronaći na sledećim linkovima:

<https://github.com/max-kov/pool>

Literatura

Materijali sa predavanja i vežbi

Materijali za predlog projekta

<https://code.tutsplus.com/tutorials/quick-tip-collision-detection-between-a-circle-and-a-line--active-10546>

shorturl.at/tvwMR