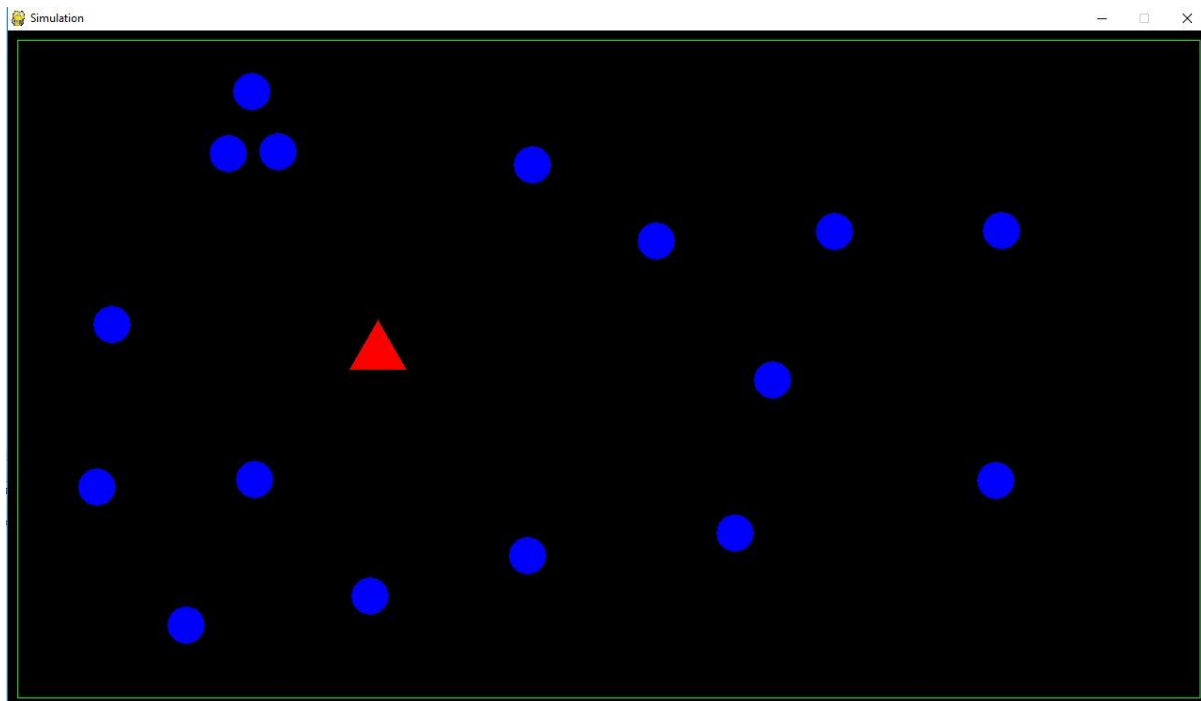


Simulacija kretanja poligona

Kratak opis

Projekat predstavlja dvodimenzionalnu simulaciju izolovanog fizičkog sistema koji se sastoji od poligona i krugova. Omogućuje korisniku da utiče na poziciju, masu i orijentaciju poligona, kao i drugih elemenata tog sistema. Korisnik pokreće poligon pomoću unosa sa tastature, podešava parametre sistema pomoću grafičkog interfejsa i mišem dodaje nove elemente. Kretanje je ograničeno u radnom prozoru.



Problemi

Potrebno je implementirati dvodimenzionalnu simulaciju kretanja poligona. Kretanje poligona (kao i ostalih elemenata) je ograničeno u radnom prozoru linijama.

U okviru projekta potrebno je implementirati i otkrivanje sudara poligon na krug, krug na krug, krug na liniju, poligon na liniju i opciono poligon na poligon.

Neophodno je voditi računa o masi tela, što je veoma bitan faktor prilikom kretanja, rotacije i udarca. U simulaciji je simulirano trenje okoline koje će usporavati sva kretanja svih tela sistema.

Detaljna specifikacija

Elementi rešenja

Korišćeni alati za rešavanje problema:

- Programski jezik Python 2.7
- Biblioteka pygame za iscrtavanje grafika
- Biblioteka aenum za bolju organizaciju koda
- Biblioteka PyQt za unos parametara
- Biblioteka numpy za proračune

Spisak problema

Kinematika

- Model
- Linearno kretanje (*Ojlerova metoda)
- Rotacija poligona (*Ojlerova metoda)

Otkrivanje kolizija

- Krug na krug
- Duž na krug
- Poligon na krug (SAT metoda*)
- Poligon na poligon (SAT metoda*) - opciono, entuzijastički (ako stignem)
- Prostorno indeksiranje - opciono, entuzijastički (ako stignem)

*Sve navedene metode su samo planirane za korišćenje (moguće su izmene pri izradi)

Linearno kretanje (Ojlerova metoda)

Kretanje bi se predstavilo diferencijalnim jednačinama koje se rešavaju Ojlerovom (opciono RK4) metodom

$$dx += v * dt$$

$$dv = f(t) / m * dt$$

Rotacija poligona (Ojlerova metoda)

Rotacija će se izvršavati preko diferencijalnih jednačina pomoću Ojlerove metode

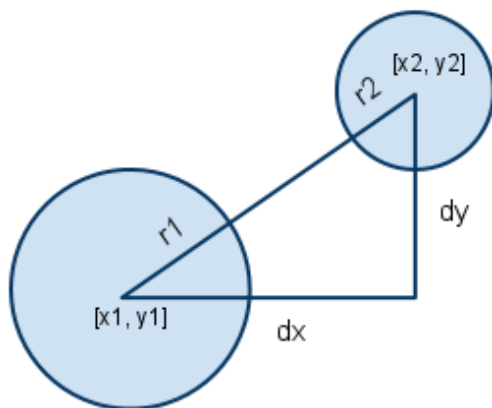
$$d\theta = \omega \cdot dt$$

$$d\omega = \tau(t) / I \cdot dt$$

Ova metoda će nam trebati prilikom udarca poligona sa krugom ili duži (ivicom radnog prozora) jer će se pri određenim udarcima poligon rotirati u zavisnosti od jačine i ugla udarca (kao na primeru datom ispod).

Kolizija krug na krug

Detekcija kolizije kruga na krug rešena je jednostavnom metodom poredjenja udaljenosti centara krugova i zbira njihovih poluprečnika. Ako je udaljenost centara manja ili jednaka od zbira njihovih poluprečnika, dešava se kolizija između krugova, u suprotnom nema kolizije.



Kolizija krug na duž

Kolizija se dešava ako se krug preseče sa duži. Ova kolizija se koristi radi implementacije ograničenog kretanja krugova po radnom prozoru.

Kolizija poligon na krug

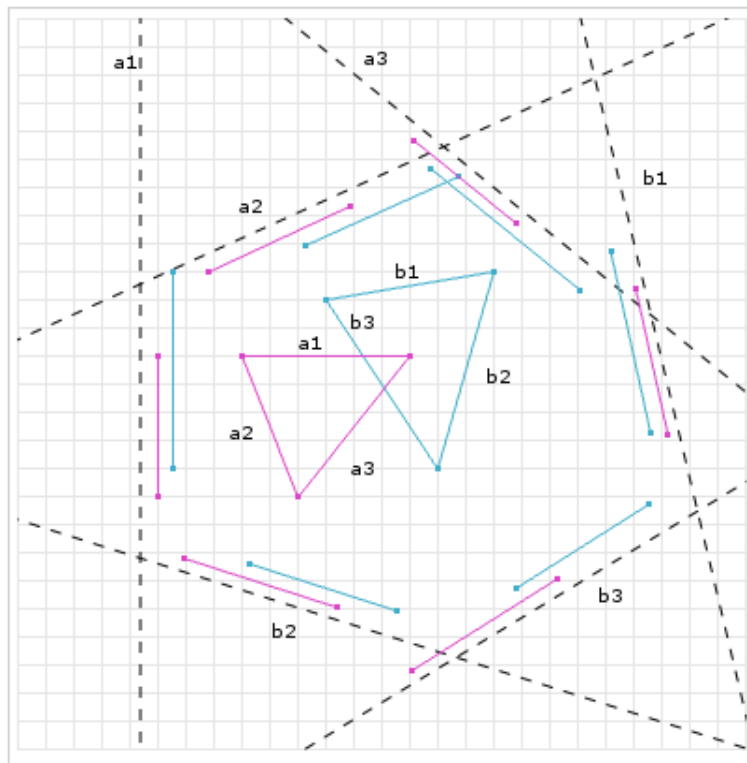
Koristiću SAT metodu za detekciju kolizija poligona na krug.

1. Generisanje pravi koje su paralelne sa stranicama poligona
2. Projekcija poligona i kruga na izgenerisane prave

3. Ako se sve projekcije imalo preklapaju, došlo je do kolizije

Kolizija poligon na poligon

Metoda SAT funkcioniše slično kao i za poligon na krug, samo što se generišu prave i za drugi poligon, pa se projekcije gledaju i za jedan i za drugi

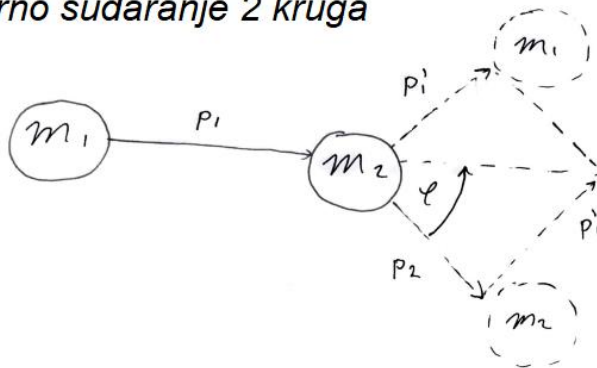


Prednost SAT metode je ako se projekcije na 1 pravi ne poklapaju, ostale se ne gledaju

Reakcija na sudare (ograničeno kretanje)

Kako elementi ne mogu da se kreću bilo gde (ne mogu da prolaze jedan kroz drugi niti da idu van okvira radnog prozora) mora postojati reakcija na udarac 2 ili više elemenata i reakcija na udarac elementa i ivice radnog prozora. Kada se 2 elementa sudare (sudar detektujemo kolizijom krug na poligon ili poligon na poligon objašnjenom iznad) oni se odbijaju jedan od drugog i njihova dalja brzina i pravac kretanja zavise od njihove mase, sile udarca i ugla pod kojim su se oni sudarili. Isto važi i za udarac elementa sa ivicom radnog prozora stim što se ivica neće pomerati, ali će doći do gubljenja kinetičke energije elementa, pa će se time element usporiti.

Linearno sudaranje 2 kruga



$$p = m \cdot \frac{dx}{dt} = m \cdot \dot{x}$$

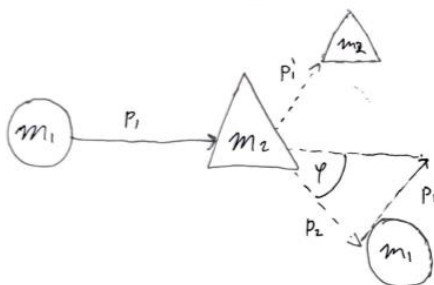
$$p_1 = p_1' + p_2$$

$$m_1 \cdot \dot{x}_1 = m_1 \cdot \dot{x}_1' + m_2 \cdot \dot{x}_2$$

$$(p_1')^2 = (p_1)^2 + (p_2)^2 - 2 p_1 \cdot p_2 \cdot \cos \varphi$$

$$F = \frac{dp}{dt} = m \frac{d\dot{x}}{dt} = m \frac{d^2x}{dt^2}$$

$$m = \text{const}$$



Linearno sudaranje poligona i kruga

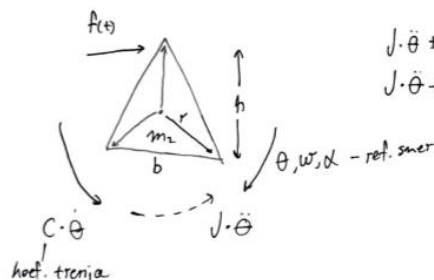
$$J = \frac{b \cdot h^2}{36} \cdot \frac{m}{P_b}$$

$$\omega = \dot{\theta}$$

$$\alpha = \dot{\omega} = \ddot{\theta}$$

$$\dot{\theta} = \frac{d\theta}{dt}$$

$$\ddot{\theta} = \frac{d^2\theta}{dt^2}$$



$$J \cdot \ddot{\theta} + c \cdot \dot{\theta} = f(t)$$

$$J \cdot \ddot{\theta} + c \cdot \dot{\theta} = m \cdot \frac{d^2x}{dt^2} \Rightarrow \ddot{\theta} = \frac{1}{J} \left[\frac{d^2x}{dt^2} \cdot m - c \cdot \dot{\theta} \right]$$

smene:

$$h_1 = \theta \quad h_1 = h_2$$

$$h_2 = \dot{\theta} \quad h_2 = \frac{1}{J} \left[\frac{d^2x}{dt^2} \cdot m - c \cdot h_2 \right]$$

za snižavanje reda

Ulazni parametri

Moguće je u toku simulacije promeniti parametre kao što su:

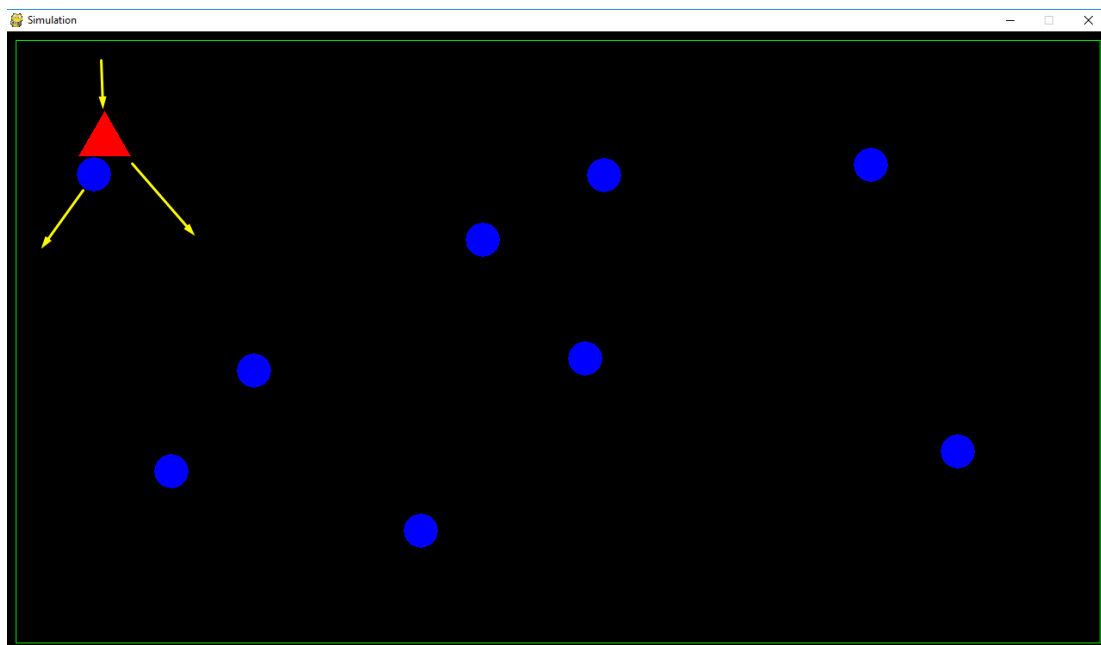
- Brzina kretanja poligona

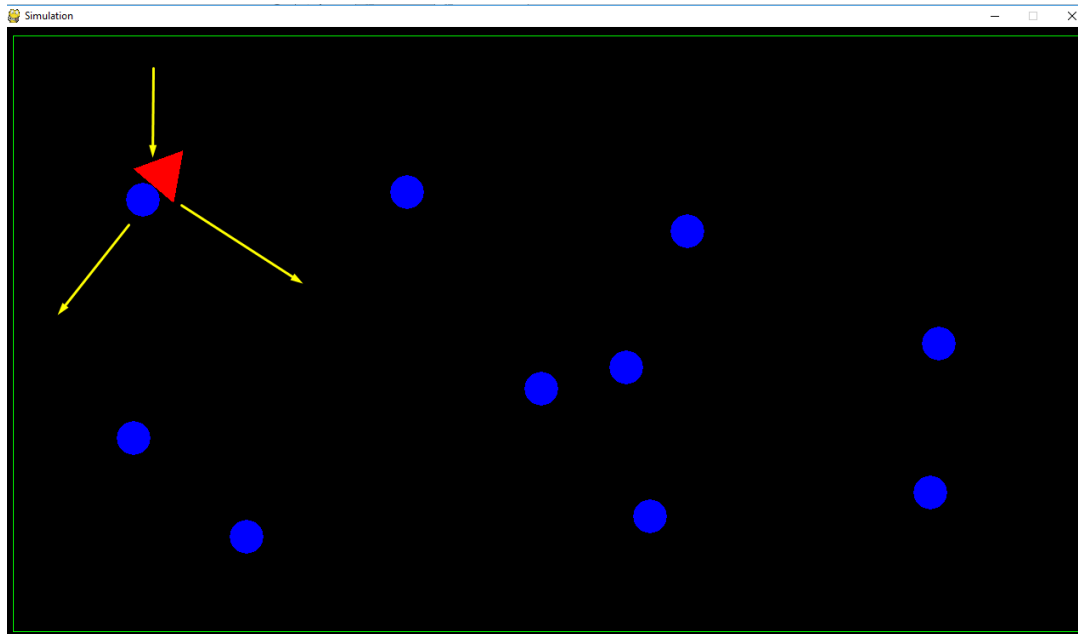
- Masa tela

- FPS

- Stepen poligona itd

Primer rotacije i sudaranja (poligon i kružnica)





Primeri gotovih rešenja - nisam našao

Reference

- https://www.youtube.com/channel/UCda_RJU9-xB0Hswcrjn4SKw
- <http://www.dyn4j.org/2010/01/sat/>
- https://en.wikipedia.org/wiki/Euler_method
- https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods
- <https://www.pygame.org/docs/>
- <https://www.tutorialspoint.com/pyqt/>
- <https://stackoverflow.com>
- Materijali sa vežbi i predavanja