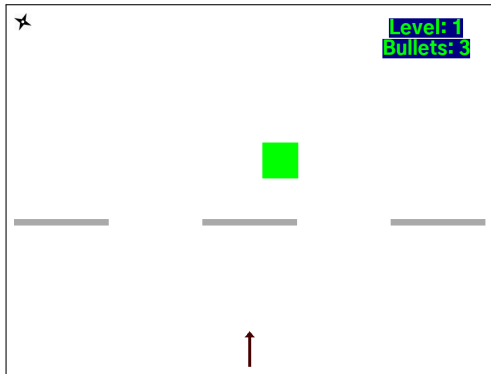


Gađanje mete

Vasilić Marko IN/2019
Savković Jovan IN/2019

Opis problema

2D igrice - gađanje mete
Igrač bira projektil kojim
gađa metu, ali na polju se
nalaze prepreke u vidu
debelih i tankih zidova.
Igrač mora da izabere
projektil koji će najbolje
obaviti posao. Šuriken će
uništiti tanke zidove, lopta
će se odbiti od debele
zidove. Treba kombinovati
projektile tako da se
oslobodi put do mete.



Kontrole

- ▶ Ciljanje:



- ▶ Pucanje:



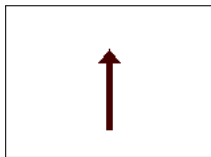
- ▶ Biranje municije:



- ▶ Resetovanje igrice:



- ▶ Zaustavljanje
aktivnog projektila:



Strelica za ciljanje

- ▶ Kinematika - pravolinijsko kretanje i rotaciono kretanje

Rešenje - RK4

- ▶ Detekcija kolizije

Rešenje - GJK

Kinematika

Pravolinijsko i rotaciono kretanje je predstavljeno kao diferencijalna jednačina drugog reda.

Pomoću RK4 algoritma se aproksimira vrednost datih funkcija kretanja pri svakom koraku igrice.

```
def rk4It(xi, yi, h, f):  
    k1 = f(xi, yi)  
    k2 = f(xi + h / 2, yi + 0.5 * k1 * h)  
    k3 = f(xi + h / 2, yi + 0.5 * k2 * h)  
    k4 = f(xi + h, yi + k3 * h)  
    return yi + h / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
```

Pravolinijsko kretanje

```
def update_pos(self): # metoda za azuriranje položaja pomoću rk4
    self.center[0] = rk4It( self.age,
                             self.center[0],
                             1,
                             lambda t, x: self.velocity[0]
                           )

    self.center[1] = rk4It( self.age,
                             self.center[1],
                             1,
                             lambda t, y: self.velocity[1]
                           )
```

Rotaciono kretanje

```
def update_angle(self): # metoda za azuriranje ugla rotacije pomoću rk4
    if self.angular_velocity != (0, 0):
        self.angle = rk4It( self.age,
                             self.angle,
                             1,
                             lambda t, theta: self.angular_velocity
                           )
```

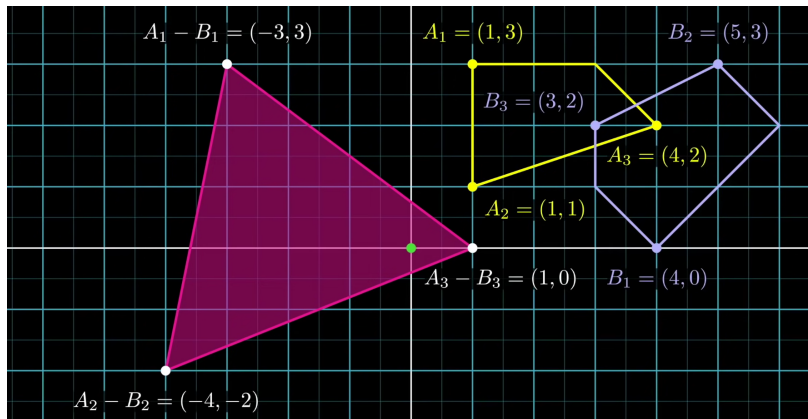
Detekcija kolizije

Detekcija kolizije funkcioniše za bilo koju kombinaciju krugova, poligona i duži

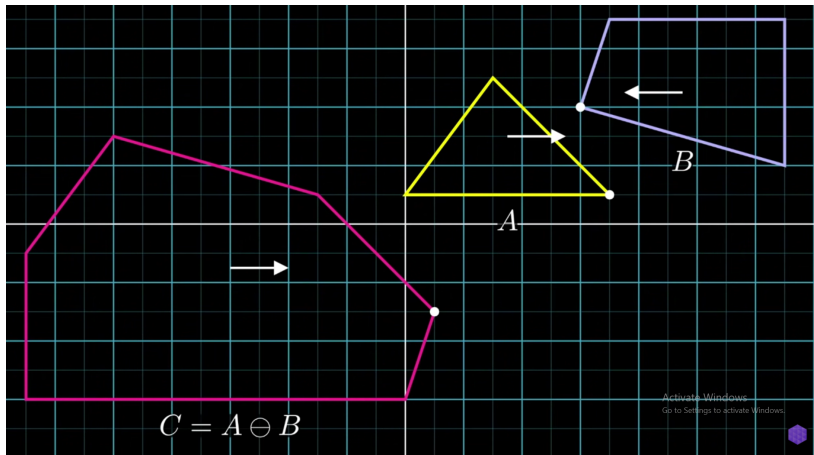
Za detekciju kolizije koristi se GJK algoritam.

```
def gjk(s1, s2):  
    if s1[0] == s2[0]:  
        return True  
    d = pygame.Vector2(-1, -1)  
    simplex = [support_point(s1, s2, d)]  
    d = ORIGIN - simplex[0]  
    for i in range(100):  
        A = support_point(s1, s2, d)  
        if A.dot(d) < 0:  
            return False  
        simplex.append(A)  
        if handleSimplex(simplex, d):  
            return True
```

Minkowski razlika



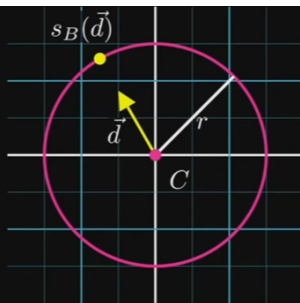
Određivanje support pointa



Support point za krug

$$s_B(\vec{d}) = C + r\vec{d}$$

*Assumes \vec{d} is a unit vector



Support point za poligon

$$v_0^T \vec{d} = 1.62$$

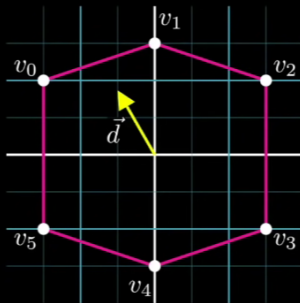
$$v_1^T \vec{d} = 1.3$$

$$v_2^T \vec{d} = 0.12$$

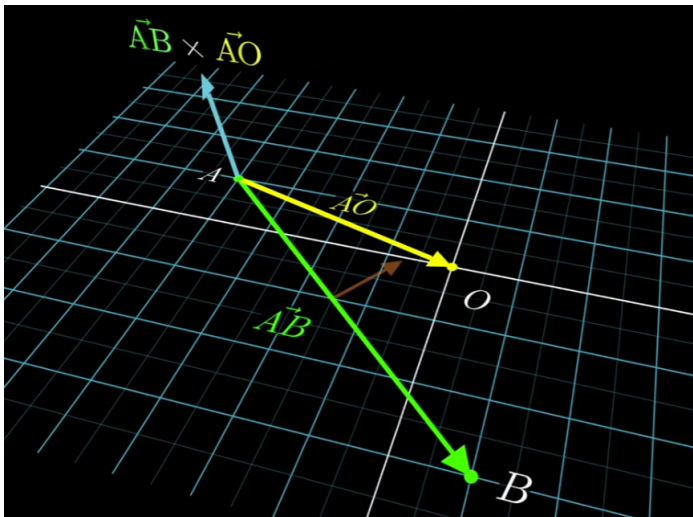
$$v_3^T \vec{d} = -1.62$$

$$v_4^T \vec{d} = -1.3$$

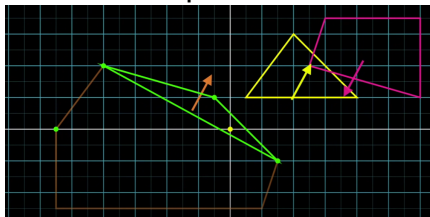
$$v_5^T \vec{d} = -0.12$$



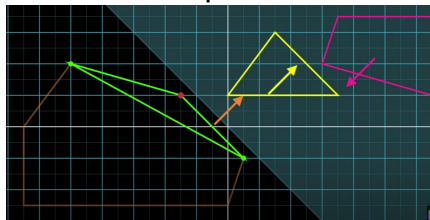
Određivanje vektorskog proizvoda



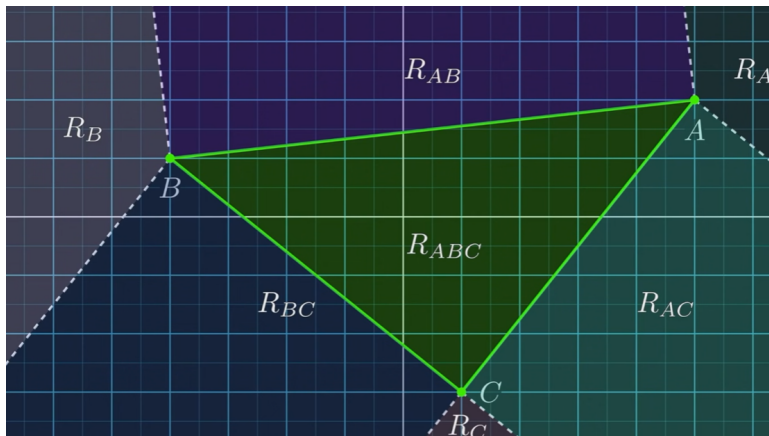
Uspešno



Neuspešno

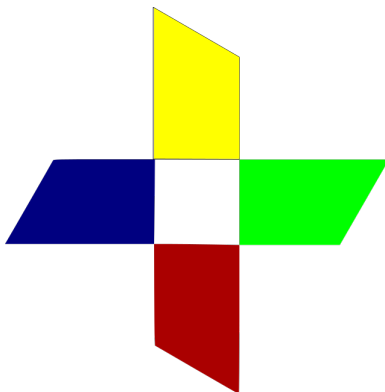


Određivanje da li trougao sadrži koordinatni početak



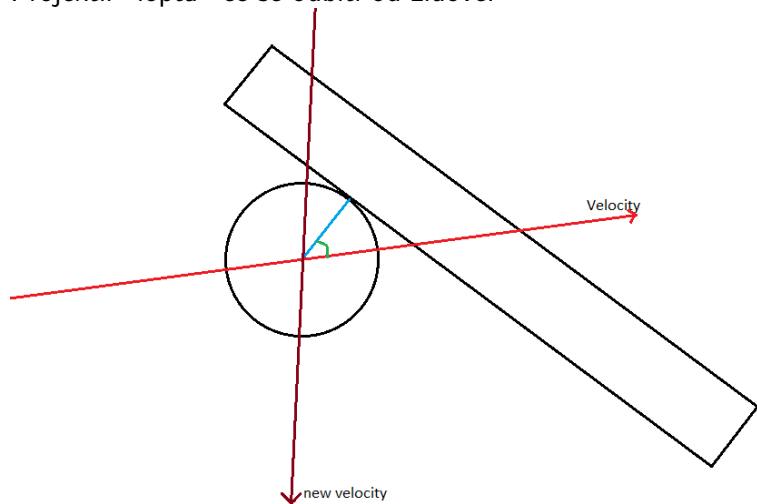
Detekcija kolizije - šuriken

Šuriken nije konveksan poligon, tako da gjk ne bi radio za njega. Ovaj problem je rešen tako što se šuriken posmatra kao skup 4 konveksna poligona. U koliko se detektuje kolizija sa bilo kojim od poligona koji ga sačinjavaju, smatra se da je šuriken došao u kontakt sa nečim.

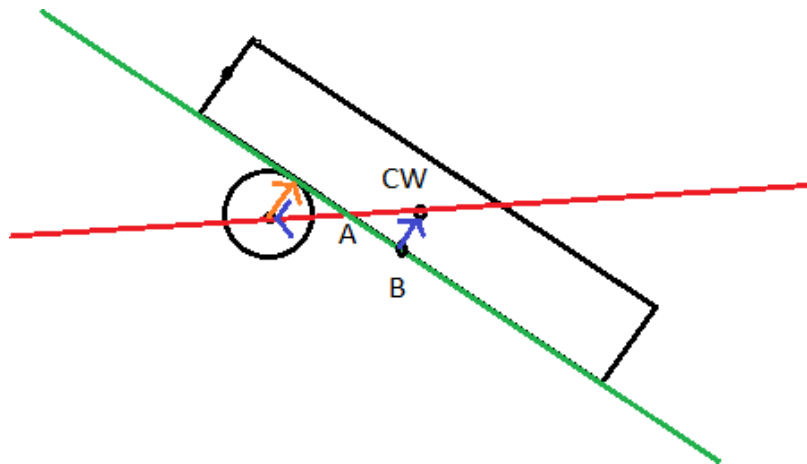


Odbijanje o zidove

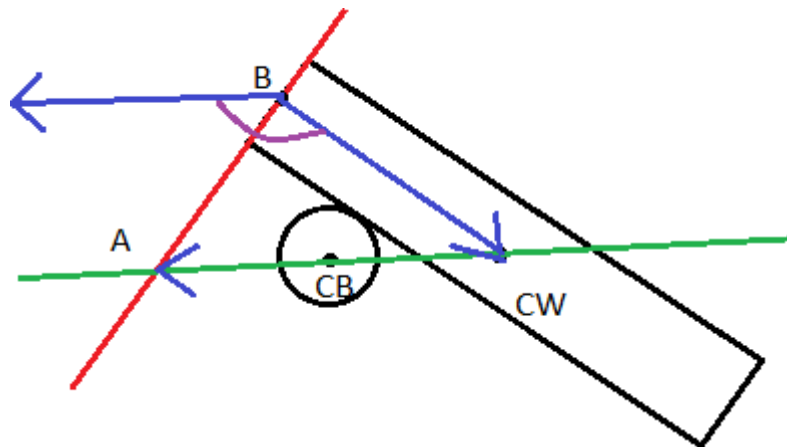
Projektil "lopta" će se odbiti od zidove.



Odbijanje o zidove



Odbijanje o zidove



Igrica se sastoji od 3 novoa koji služe da demonstriraju funkcionalnosti igre.

Za svaki nivo igrač ima na raspolaganju 3 projektila da pogodi metu, u koliko ne pogodi kreće ispočetka.

Nivo 1

Jednostavno gađanje mete.



Level: 1
Bullets: 3



Nivo 2

Igrač mora prvo da pomoću šurikena uništi jedan od tankih zidova, zatim da pogodi metu proizvoljnim projektilom.

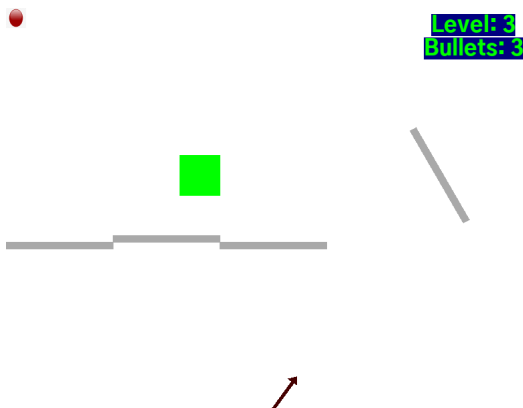


Level: 2
Bullets: 3



Nivo 3

Igrač mora da odbije lopticu od dijagonalni zid na desnoj strani ekrana da bi pogodio metu.



Hvala na pažnji