Will Legg

04-11-2024

REFLECTION PROMPT: This prompt is a bit different and part of a prompt from a previous week, as it is often constructive. We're including a weird, and hopefully fun spin on it at the end. Consider a member of your current group,

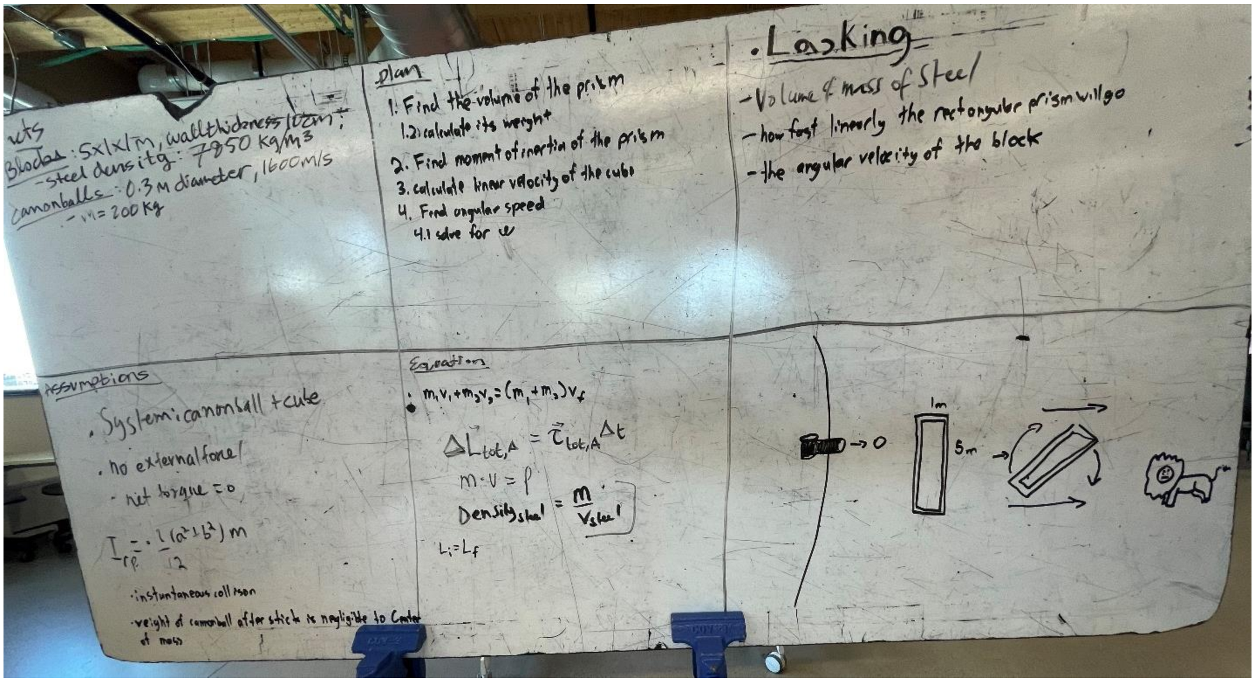
1. who is someone that you think is contributing positively as a group member, and describe what it is they are doing. Eleazer from our group is very strong with his calculation skills and also has many ideas about how we can solve any problem. He always begins thinking out his ideas and talking with us in the group to try to make his approach as direct and thorough as possible. Nicholas and Mike are very outgoing and positive during our solving process and they definitely encourage discussion within our group.

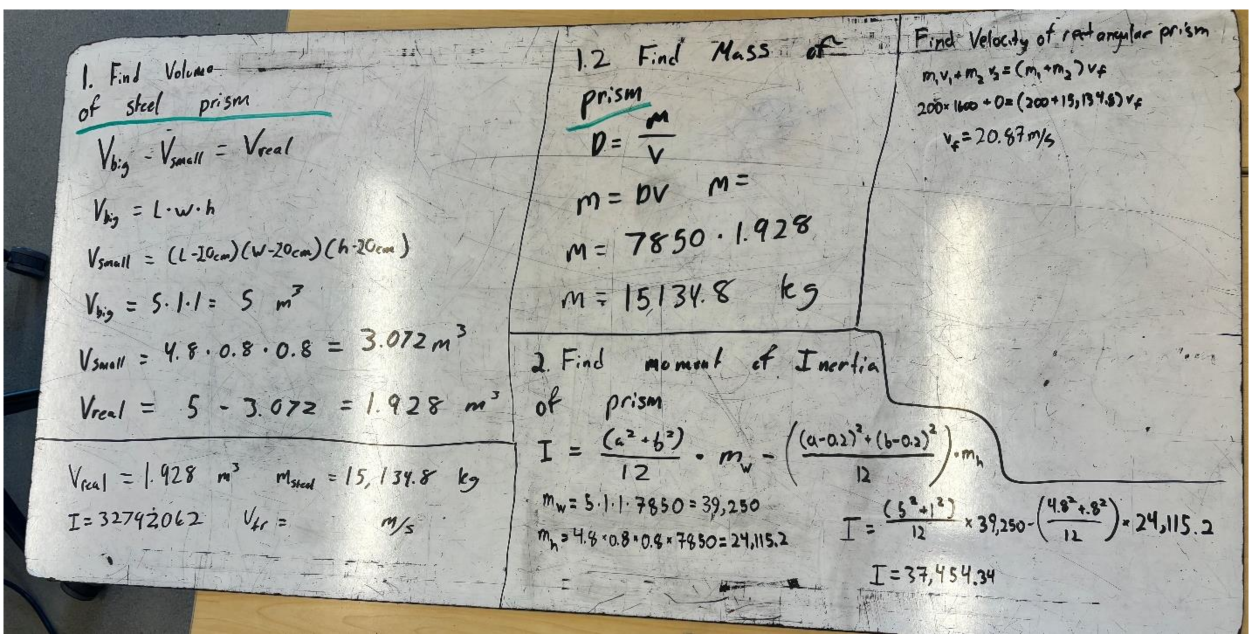
2. What have you learned from them that you think you will apply the next time you are working in groups in another class. Moving forward I think I will always try to encourage as much discussion as possible within a group and try my best to make sure that everyone is at the same level of understanding. I'll also try my best to balance between my brainstorming vs. actual generation of plan, and not try to move directly from brainstorming to solving. All of my groupmates have been great fun to work with and they're all great problem solvers. I think one of the reasons we've found some success as a group is because they're easy to talk to and are open to having a laugh throughout our time we've spent in the trenches of the boartiger physics realm.

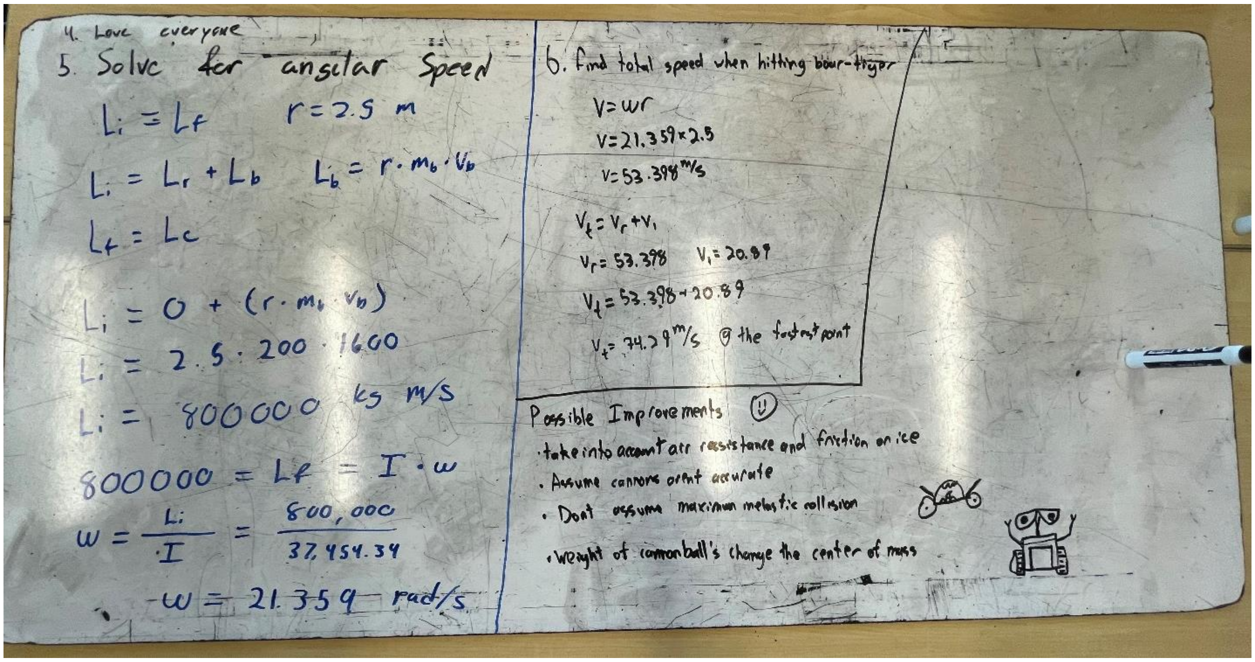
Finally, the weird bit, 3. what is their fictional origin story that has resulted in them becoming excellent at contributing to a group in this way (bitten by radioactive spider, struck by lightening...)?

I have a feeling that my groupmates were probably exposed to a radioactive green ooze in the sewers of New York City and taken in as pupils of a very knowledgeable rat, maybe named "*Spring*er" or something similar. Instead of learning the ways of combat, they learned how to become some of the best neo-physicians that I've met so far.

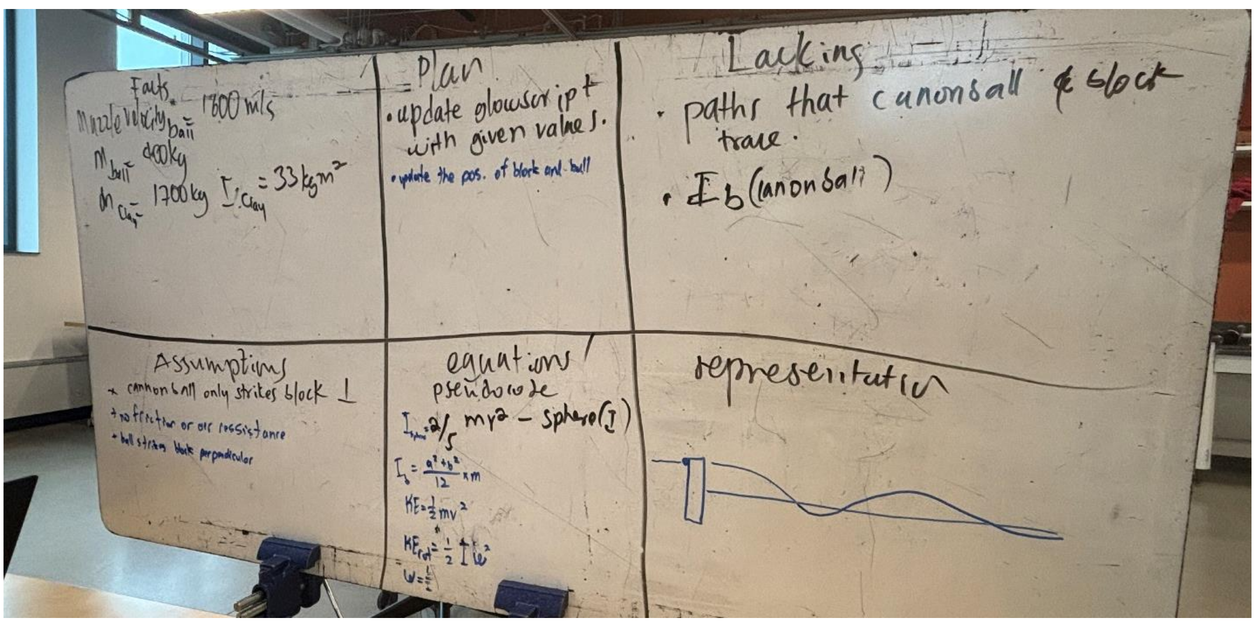
PASTE IMAGES OF YOUR TUESDAY WHITEBOARDS HERE:





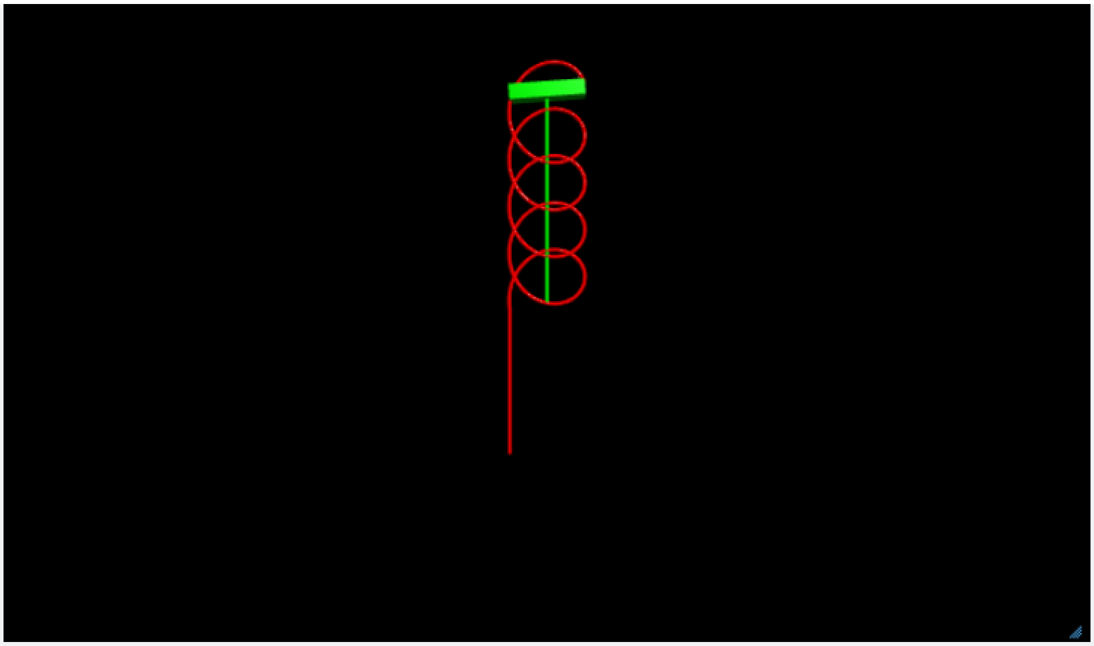


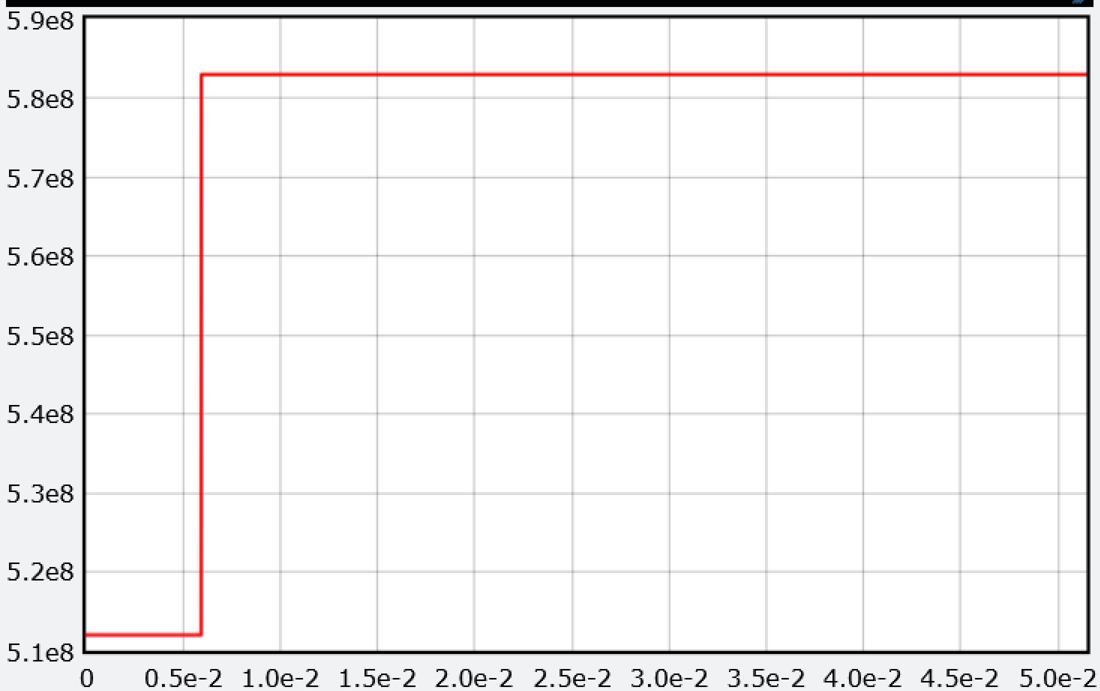
PASTE IMAGES OF YOUR THURSDAY WHITEBOARDS HERE:



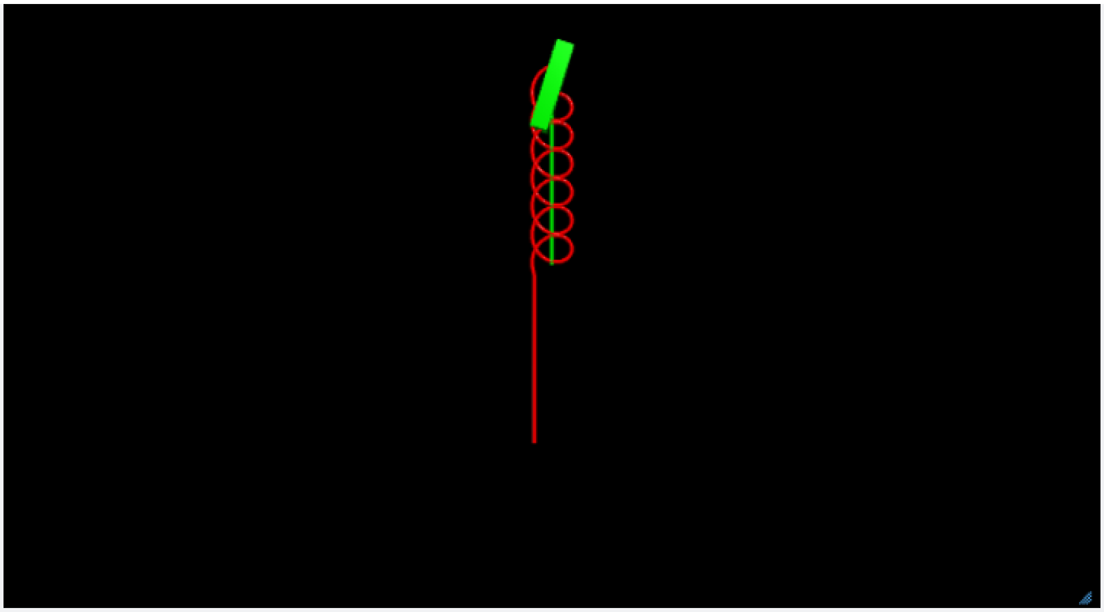
```
Web VPython 3.2
# GlowScript 2.9 VPython

get_library('https://cdn.rawgit.com/PERLMSU/physutil/master/js/physutil.js')


#Window setup
scene.center = vec(0,0,0)
scene.range = 5

#Objects
block = box(pos=vec(0,0,0), length=5, width=1, height=1, color=color.green,make_trail=True)
ball = sphere(pos=vector(-2.45,-10,0), radius=0.15, color=color.red,make_trail=True)

#Parameters and Initial Conditions
mball = 400
mblock = 1700
mtot = mball + mblock

vball = vector(0,1600,0)
# total translational velocity (block + ball) using conservation of momentum
vtrans = (mball*vball)/(mball + mblock)

pball = mball*vball
ptot = pball

Lball = cross(ball.pos,pball)
Ltot = Lball

# moment of inertia for block
Iblock = 33

#Time and time step
t = 0
tf = 10
dt = 0.00001

#Graph

Egraph = PhysGraph(numPlots = 1)
E = 0

print("Position on x-axis where ball collides w/ block: " + ball.pos.x + " m" + "\n")

```

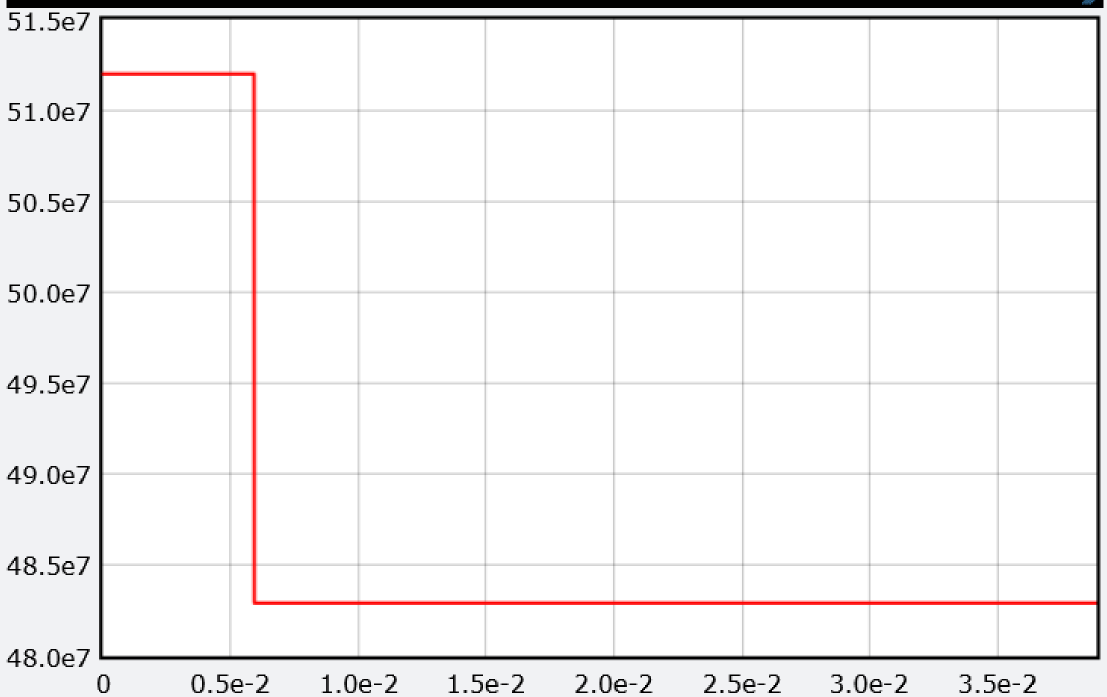
```
#Calculation Loop
while t<tf:
    rate(500)

    if ball.pos.y<-0.5 or abs(ball.pos.x)>2.5 or abs(ball.pos.z)>0.5:

        ball.pos = ball.pos + (pball/mball)*dt


        # calculate Energy of system prior to collison
        KEb = (1/2)*(mball)*(mag(vball)**2)

        Egraph.plot(t,KEb)


    else:
        # update position of block/ball based on translational velocity
        block.pos = block.pos + (vtrans*dt)
        ball.pos = ball.pos + (vtrans)*dt

        # calculate moment of inertia for ball/block
        Iball = mball * (mag(block.pos - ball.pos))**2
        # Total moment of inertia
        Itot = Iball + Iblock

        # update position of ball block based on rotational velocity
        block.rotate(angle=(mag(Ltot)/(Itot))*dt,axis=Ltot/mag(Ltot),origin=block.pos)
        ball.rotate(angle=(mag(Ltot)/(Itot))*dt,axis=Ltot/mag(Ltot),origin=block.pos)

        # calculate energy of system after collision
        KEtrans = (1/2)*(mtot)*(mag(vtrans)**2)
        KErot = (1/2)*((mag(Ltot)**2)/(Itot))
        KE = KEtrans + KErot
        Egraph.plot(t, KE)
    t = t + dt


print("KE of system before collision: " + KEb + " J" + "\n")
print("KE of system after collision: " + KE + " J" + "\n")
```





Position on x-axis where ball collides w/ block: -2.45 m





Position on x-axis where ball collides w/ block: -1 m