



**UNIVERSIDADE FEDERAL DE MATO GROSSO  
CAMPUS UNIVERSITÁRIO DO ARAGUAIA  
INSTITUTO DE CIÊNCIAS EXATAS E DA TERRA  
MESTRADO PROFISSIONAL EM ENSINO DE FÍSICA**

## **APRENDENDO O MOVIMENTO CIRCULAR UNIFORME COM O AUXÍLIO DO AMBIENTE VIRTUAL 3D VPYTHON**

Agnaldo Gonçalves Borges Junior

Produto educacional da dissertação de Mestrado Nacional Profissional em Ensino de Física, intitulada Aprendendo o Movimento Circular Uniforme com o Auxílio do Ambiente Virtual 3D Vpython, sob orientação do Prof. Dr Fabízio Myaki Alves, junto ao Programa de Pós-Graduação da Universidade Federal do Mato Grosso

Barra do Garças  
2020

## SUMÁRIO

APRESENTAÇÃO .....	3
RECOMENDAÇÕES PARA O USO DO AMBIENTE DE AULA.....	4
BOAS PRÁTICAS EM ATIVIDADES EXPERIMENTAIS.....	4
QUESTIONÁRIO 01 - DIAGNÓSTICO.....	6
GUIA AUXILIAR NA APLICAÇÃO DAS ANIMAÇÕES.....	11
I) ENGRENAGEM .....	11
II) TERRA .....	14
CÓDIGO FONTE DO SIMULADOR - ENGRENAGENS.....	16
FIGURA ILUSTRATIVA do simulador - Engrenagens com cursor mudança de velocidade angular .....	21
CÓDIGO FONTE DO SIMULADOR - TERRA (COM CURSOR DE MUDANÇA DA VELOCIDADE ANGULAR).....	22
FIGURA ILUSTRATIVA DO SIMULADOR MOVIMENTO ROTACIONAL DA TERRA - Com cursor de mudança da velocidade angular.....	28
CÓDIGO FONTE - TERRA (LANÇAMENTO DE OBJETOS).....	29
FIGURA ILUSTRATIVA DO SIMULADOR MOVIMENTO ROTACIONAL DA TERRA - Lançamento de objetos..	35
QUESTIONÁRIO 02 - AVALIATIVO.....	36
QUESTIONÁRIO 03 - AVALIAÇÃO METODOLÓGICA DAS ATIVIDADES .....	41
LINK PARA ACESSO ÀS SIMULAÇÕES DESTE PRODUTO .....	43

## APRESENTAÇÃO

O objetivo deste trabalho "produto educacional", é que o professor possa utilizar essa ferramenta pedagógica para discutir movimentos de corpos associados com trajetória curvilínea (em particular daqueles com trajetória circular), sendo a aplicação central em Movimento Circular Uniforme. Citamos como exemplo o movimentos de rotação da Terra e também conjuntos de engrenagens acopladas(uso geral em motores e outros).

O produto educacional desenvolvido nesse programa, tem como objetivo principal, auxiliar o professor em sala de aula na aplicação dos conteúdos, podendo ser utilizado como metodologia facilitadora para a compreensão, fixação de conteúdo assim como para visualização das aplicações em seu dia-a-dia.

Nosso trabalho utiliza o ambiente virtual 3D Vpython, onde o mesmo proporciona como foco o uso das animações de física, tendo como objetivos mais específicos:

- 1 - Atuar como instrumento facilitador da compreensão do conteúdo;
- 2 - Incentivar o aprendizado da teoria, promovendo discussões com os demais alunos;
- 3 - Tornar o momento de aula mais descontraídas promovendo um maior interesse dos alunos para disciplina assim como para o conteúdo;
- 4 - Mostrar a aplicação prática da física no cotidiano e na natureza;
- 5 - Desenvolver no aluno pleno interesse em aprender física;
- 6 - Proporcionar um espírito crítico da aplicação das leis da física;
- 7 - Propiciar aos alunos a capacidade de discutir e interagir uns com os outros através de grupo de discussão.

Assim nosso anseios é que esse trabalho de atividade prática/experimental, contribua para o auxilio no desenvolvimento da aplicação dos conceitos em sala

de aula, atuando assim como um método que auxilie tanto para o docente como para o aluno.

## **RECOMENDAÇÕES PARA O USO DO AMBIENTE DE AULA**

Preferencialmente usar em laboratório de informática da escola, porém pode ser utilizados em sala de aula usando equipamentos dos próprios alunos.

Não possui risco de acidente, exceto o aluno não cumpra as diretrizes de uso do espaço.

Lembre-se por mais que não haja risco de acidente a segurança e o sucesso ao realizar uma atividade experimental está apoiada em todos - o professor como orientador e o aluno como executor, devendo haver em toda equipe a responsabilidade da execução da mesma. Porém é bom lembrar que o professor é o maior responsável nessa atividade.

O sucesso de todo trabalho teórico/prático está no planejamento antecipado das atividades, teste antes do uso cada equipamento assim como os programas, para que na hora não ocorra surpresas desagradáveis.

## **BOAS PRÁTICAS EM ATIVIDADES EXPERIMENTAIS**

- ✓ Antes de iniciar uma atividade prática, é importante conhecê-la, assim como o ambiente de execução e os equipamentos.
- ✓ Evite uso de bebidas de qualquer natureza para que não corra o risco de danificar os equipamentos.
- ✓ Não permita o uso de qualquer outro equipamento que não seja o do desenvolvimento do simulador para que a atividade não se disperse.
- ✓ Evite conversas paralelas e brincadeiras, elas são desnecessárias, pois tira a atenção e o foco dos demais.
- ✓ Importante lembrar que você também é parte de uma equipe e a principal responsabilidade sobre todos e sobre o desenvolvimento da atividade é sua como professor.
- ✓ A segurança depende da ação de todos e do cumprimento das regras do local de uso, não mexa no que não lhe é permitido.

- ✓ Tome como hábito planejar as atividades que vai realizar, de modo a executá-lo com segurança.
- ✓ Conheça as principais características dos produtos que vai manipular, tenha em mãos este guia para consulta frequente.
- ✓ Proporcione ao seu aluno momentos de discussão, interação sobre a atividade, dê liberdade para que eles façam perguntas sanando todas as dúvidas sobre o assunto.

## QUESTIONÁRIO 01 - DIAGNÓSTICO

1 - O que é força centrípeta? Dê alguns exemplos em que ela está presente.

---

---

---

---

2 - Defina e explique o que são velocidades linear e angular.

---

---

---

---

3 - O que tem de característico no movimento circular uniforme de um objeto?

- a - ( ) a velocidade linear(vetor) do objeto não se altera;
- b - ( ) O valor (módulo) da velocidade linear é constante;
- c - ( ) O objeto não possui aceleração;
- d - ( ) Não existe força sobre o objeto.

**Justifique sua resposta:**

---

---

---

---

4 - Assinale quais as grandezas físicas abaixo que estão envolvidas no movimento circular uniforme?

- a - ( ) aceleração centrípeta ( $m/s^2$ );
- b - ( ) tempo (s);
- c - ( ) aceleração angular ( $rad/s^2$ );
- d - ( ) massa (kg)
- e - ( ) temperatura (Kelvin)
- f - ( ) velocidade angular ( $rad/s$ );

**Justifique sua resposta:**

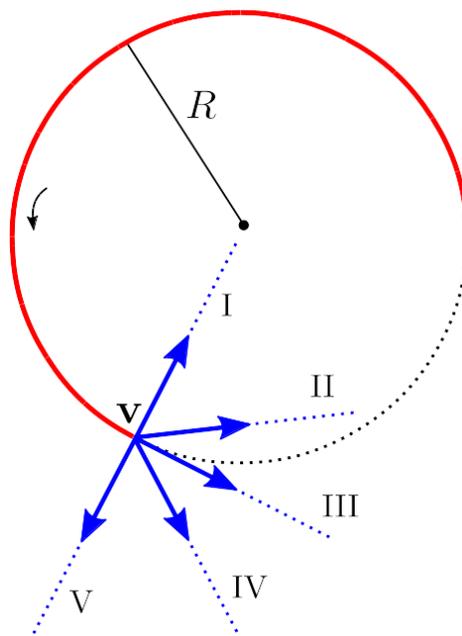
---

---

---

---

**5** - Um objeto com velocidade linear  $v$  realiza um movimento circular uniforme de acordo com a figura 1.



**Figura 1**

Se a ação que promove essa trajetória do objeto for cessada (desligada) no instante indicado na figura pelo seu vetor velocidade (seta azul), qual será a direção do movimento do objeto a partir desse instante?

- a - ( ) I;
- b - ( ) II;
- c - ( ) III;
- d - ( ) IV;
- e - ( ) V.

**Justifique a sua resposta:**

---

---

---

---

6 - Cite três situações do cotidiano em que ocorre o movimento circular.

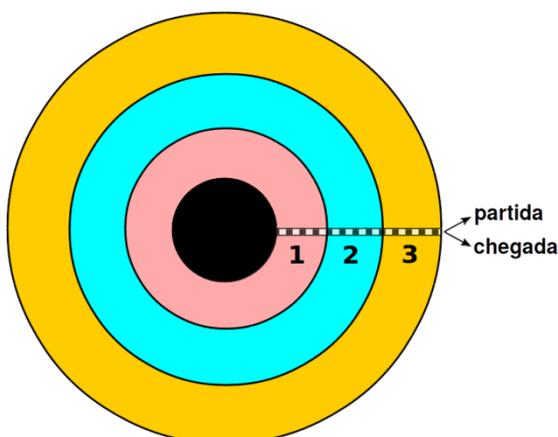
---

---

---

---

Três atletas posicionados em cada uma das raias 1, 2 e 3, disputarão uma corrida cujo percurso corresponde a uma volta completa na pista circular ilustrada na **figura 2** à seguir.



**Figura 2**

Com base nessas informações, responda as questões **7** e **8**.

**7** - Considerando que eles iniciam a corrida no mesmo instante percorrendo todo o trajeto com as mesmas velocidades lineares, qual(is) corredor(es) deve(m) vencer a corrida?

- a - ( ) O corredor da raia 3.
- b - ( ) Não existirá um único vencedor pois todos chegarão ao mesmo tempo.
- c - ( ) O corredor da raia 1.
- d - ( ) O corredor da raia 2.
- e - ( ) Não existirá um único vencedor pois os corredores das raias 1 e 2 chegarão juntos.

**Justifique sua resposta:**

---

---

---

---

**8** - Para que o corredor da raia 3 termine a corrida junto com o da raia 1, o que deve acontecer supondo que eles larguem no mesmo instante?

- a - ( ) Ele deve ter velocidade angular menor que o da raia 1.
- b - ( ) Ele deve ter velocidade angular maior que o da raia 1.
- c - ( ) Ambos devem ter a mesma velocidade angular.

**Justifique sua resposta:**

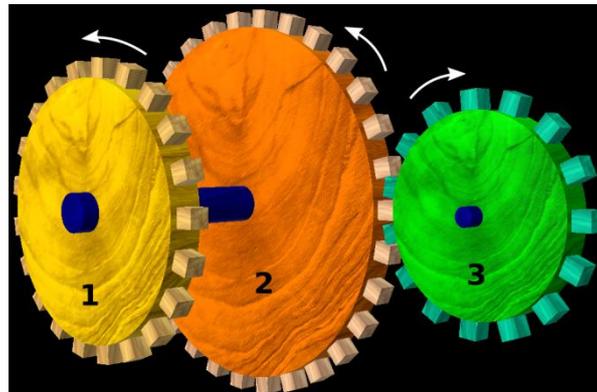
---

---

---

---

**A figura 3**, mostra três engrenagens de tamanhos (raios) diferentes realizando movimento circular uniforme, girando nos sentidos indicados na figura. As engrenagens 1 e 2 estão ligadas pelo mesmo eixo de rotação girando no mesmo sentido, e as engrenagens 2 e 3 estão acopladas pelos seus dentes e portanto girando em sentidos opostos. A engrenagem 2 possui o maior raio e a engrenagem 3 o menor raio.



**Figura 3**

Com base nessas informações responda as questões **9** e **10**.

**9** - Sobre os movimentos das engrenagens 1 e 2 podemos afirmar que em módulo:

- a - ( ) a velocidade angular da engrenagem 1 é menor que o da engrenagem 2.
- b - ( ) as velocidades angulares das engrenagens são iguais.
- c - ( ) as velocidades lineares das engrenagens são iguais.
- d - ( ) a velocidade linear da engrenagem 1 é maior que o da engrenagem 2.

**Justifique sua resposta:**

---

---

**10** - Sobre os movimentos das engrenagens 2 e 3 podemos afirmar que em módulo:

**a** - ( ) a velocidade angular da engrenagem 2 é maior que o da engrenagem 3.

**b** - ( ) as velocidades angulares das engrenagens são iguais.

**c** - ( ) as velocidades lineares das engrenagens são iguais.

**d** - ( ) a velocidade linear da engrenagem 2 é menor que o da engrenagem 3.

**Justifique sua resposta:**

---

---

---

---

## GUIA AUXILIAR NA APLICAÇÃO DAS ANIMAÇÕES

### I) ENGRENAGEM

Essa animação é constituída por três engrenagens rotuladas por 1 (laranja), 2 (verde) e 3 (amarela): duas delas (1 e 3) estão sob o eixo esquerdo e uma (2) no eixo direito conforme ilustrado na figura A. As engrenagens 1 e 2 estão ligadas através de seus dentes.

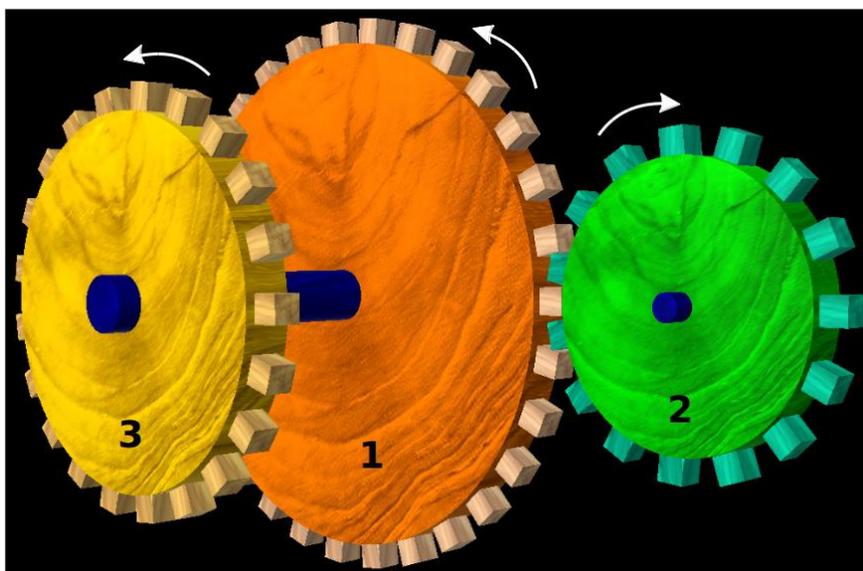


Figura A

Os vetores das grandezas físicas que descrevem a cinemática do movimento circular uniforme das engrenagens, velocidade linear ( $v$ ), raio ( $r$ ) e velocidade angular ( $\omega$ ), são mostradas na animação. Um clique do mouse permite visualizar os seus valores. Existe uma alavanca que permite alterar, de um valor mínimo até um máximo, a intensidade da velocidade angular de giro do eixo esquerdo. Já o eixo direito está livre para girar.

1). Com base nessas informações consideremos primeiro a situação em que os raios das três engrenagens são iguais. Desloque vagarosamente e gradativamente a posição da alavanca para mudar o valor da velocidade angular do eixo esquerdo e então observe e descreva o que ocorre com as velocidades  $v$  e  $\omega$  de cada engrenagem.

---

---

---

---

2). Analisemos agora uma segunda situação considerando as três engrenagens com raios diferentes. Observa-se alguma mudança em relação a primeira situação? Em caso afirmativo aponte as semelhanças e diferenças existentes no par de engrenagens 1 e 2 e também no par 1 e 3. De acordo com a teoria do movimento circular uniforme vista em sala de aula explique por que isso acontece?

---

---

---

---

3). As engrenagens 1 e 2 realizam uma volta completa no mesmo intervalo de tempo? E as engrenagens 1 e 3? Discuta.

---

---

---

---

4). Reduzindo o raio da engrenagem 2 pela metade do raio da engrenagem 1, como fica a relação (razão) das velocidades angulares dos seus respectivos eixos? E o que ocorre com essas velocidades se o raio da engrenagem 2 diminuísse ainda mais?

---

---

---

---

5). Aponte uma vantagem (ou vantagens) como aplicação propiciada por esse sistema de engrenagens

---

---

---

---

**OBS.: É importante dizer que todas as etapas (passos) apresentadas nesse guia sejam repetidas pelo aluno quantas vezes for necessárias para a compreensão do problema. E que em cada etapa o aluno faça uma reflexão sempre apoiado/embasado na teoria vista em sala de aula, podendo até mesmo consultá-la (revisá-la) novamente durante a sua interação com as animações.**

## II) TERRA

As próximas duas animações, tratam do movimento circular uniforme de três objetos em repouso localizados em diferentes pontos na superfície da Terra que gira com velocidade angular  $\omega$  constante em relação a um eixo passando pelo seu centro.

A primeira animação permite mudar o valor de  $\omega$  da Terra de zero até um valor limite. As grandezas vetoriais raio (cor vermelha), velocidade linear (cor laranja) e a força peso (cor azul), referentes a cada objeto, são mostradas na animação. Varie gradativamente o valor de “ $\omega$ ” e observe e analise as mudanças decorrentes.

1) Aumentando e posteriormente diminuindo o valor de “ $\omega$ ”, o que ocorre com a Terra? Discuta o que muda e o que não muda com o “ $\omega$ ”.

---

---

---

---

2). Para um dado “ $\omega$ ”, o tempo gasto (período) pelos três objetos completarem uma volta em torno da Terra é o mesmo ou não? Por que? Faça uma comparação com uma situação equivalente (análoga) presenciada no problema das engrenagens.

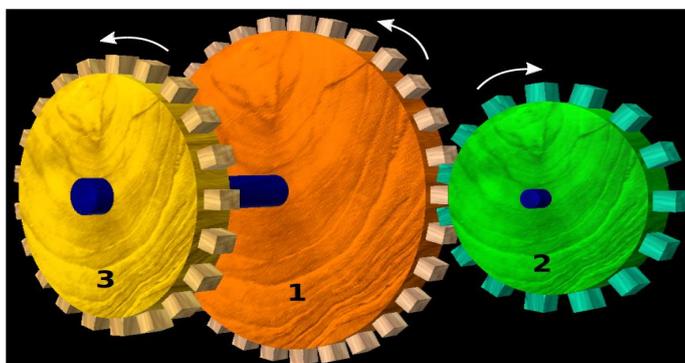


Figura A

---

---

---

---

3). No caso em que,  $\omega \neq 0$ , temos uma nova grandeza vetorial (na cor branca). Observe-a e descreva o seu comportamento para diferentes valores de " $\omega$ ". Discuta com o professor para saber mais detalhes sobre essa grandeza.

---

---

---

---

4). Consideremos que um quarto objeto seja colocado num ponto sobre um dos polos da Terra. De acordo com o movimento dos três objetos analisados anteriormente, o que podemos afirmar a respeito desse quarto objeto? Discuta.

---

---

---

---

5). Na segunda animação a velocidade angular da Terra está fixada. Supomos agora que tivéssemos alguma possibilidade de desligar as forças agindo sobre todos os objetos. Feito isso o que então aconteceria com os seus movimentos? Dê um clique no mouse durante a execução da animação para obter a resposta. Discuta o resultado obtido.

---

---

---

---

**OBS.: É importante dizer que todas as etapas (passos) apresentadas nesse guia sejam repetidas pelo aluno quantas vezes for necessárias para a compreensão do problema. E que em cada etapa o aluno faça uma reflexão sempre apoiado/embasado na teoria vista em sala de aula, podendo até mesmo consultá-la (revisá-la) novamente durante a sua interação com as animações.**

## CÓDIGO FONTE DO SIMULADOR - ENGRENAGENS

```
from visual import*

from visual.controls import*

import math

scene = display(width=1200,height=1500,background=(0.25,0.25,0.25))

scene.range = 2.2 # zoom da tela (quando as engrenagens possuem raios
diferentes)

scene.ambient = 0.6

scene.forward = (-0.2,0.0,-0.3) # posicionamento da câmera

'''
def omega(valor):
    w1 = valor
'''

gira = (0.0,0.0,1.0)

fixa1 = frame()

fixa2 = frame()

re1 = 0.6 # raio da engrenagem 1

nr = 1; re2 = re1 / nr # raio da engrenagem 2 (o sincronismo entre as engrenagens
ocorre apenas para: nr = 1, 2, 3, 5, 6)

re3 = re1 # raio da engrenagem 3

larg = 0.1 # espessura das engrenagens

hc1 = 0.6 * larg # dimensões de cada dente (cubo) - engrenagem 1

hc2 = hc1 # dimensões de cada dente (cubo) - engrenagem 2

hc3 = hc1 # dimensões de cada dente (cubo) - engrenagem 3

engrena1 = cylinder(frame = fixa1, pos = (-re1 - 0.65*hc1,0.0,0.0), axis =
(0.0,0.0,1.0), radius = re1, length = larg,
color = color.orange, material = materials.wood)

pino1 = cylinder(frame = fixa1, pos = (engrena1.pos.x,engrena1.pos.y,-
engrena1.length/2), axis = (0.0,0.0,1.0), radius = re1/10, length = 0.8,
```

```
color = color.blue, material = materials.wood)
```

```
engrena2 = cylinder(frame = fixa2, pos = (re2 + 0.65*hc2,0.0,engrena1.pos.z),  
axis = (0.0,0.0,1.0), radius = re2, length = larg,  
color = color.green, material = materials.wood)
```

```
pino2 = cylinder(frame = fixa2, pos = (engrena2.pos.x,engrena2.pos.y,-  
engrena2.length/2), axis = (0.0,0.0,1.0), radius = re2/10, length = 0.2,  
color = color.blue, material = materials.wood)
```

```
engrena3 = cylinder(frame = fixa1, pos = (-re1 - 0.65*hc1,0.0,0.6), axis =  
(0.0,0.0,1.0), radius = re3, length = larg,  
color = color.yellow, material = materials.wood)
```

```
n1 = 31 # número de dentes da engrenagem 1
```

```
n2 = n1 # número de dentes da engrenagem 2 (iguais)
```

```
n3 = n1 # número de dentes da engrenagem 3 (iguais)
```

```
for i in arange(0,n1): # estrutura de repetição que gera n1 dentes na engrenagem  
1
```

```
    dentes1 = box(frame = fixa1, pos = vector((re1 + hc1/2) * cos(2 * i * pi / (n1 - 1)),  
(re1 + hc1/2) * sin(2 * i * pi / (n1 - 1)), engrena1.length/2) + engrena1.pos,  
                axis = (cos(2 * i * pi / (n1 - 1)), sin(2 * i * pi / (n1 - 1)), 0.0), size = (hc1,  
hc1, hc1), material = materials.wood)
```

```
for i in arange(0,n2): # estrutura de repetição que gera n2 dentes na engrenagem  
2
```

```
    dentes2 = box(frame = fixa2, pos = vector((re2 + hc2/2) * cos(2 * i * pi / (n2 - 1)),  
(re2 + hc2/2) * sin(2 * i * pi / (n2 - 1)), engrena2.length/2) + engrena2.pos,  
                axis = (cos(2 * i * pi / (n2 - 1)), sin(2 * i * pi / (n2 - 1)), 0.0), size = (hc2,  
hc2, hc2), color = color.cyan, material = materials.wood)
```

```
for i in arange(0,n3): # estrutura de repetição que gera n3 dentes na engrenagem  
3
```

```
    dentes3 = box(frame = fixa1, pos = vector((re3 + hc3/2) * cos(2 * i * pi / (n3 - 1)),  
(re3 + hc3/2) * sin(2 * i * pi / (n3 - 1)), engrena3.length/2) + engrena3.pos,
```

```
axis = (cos(2 * i * pi / (n3 - 1)), sin(2 * i * pi / (n3 - 1)), 0.0), size = (hc3, hc3, hc3), color = (1.0, 1.0, 0.7), material = materials.wood)
```

```
tamalt = 220 # altura/comprimento para cima do controlado
```

```
tamlarg = 800 # largura do controlador
```

```
painel = controls(x = 1200, y = 0, width = tamalt, height = tamlarg, range = 100)
```

```
w1 = slider(pos=(0.0,-50.0), width = 20.0, length = 70.0, axis = (0,1,0), min = 0.0, max = 1.0, action = lambda: w1)
```

```
menu(pos=(0.0,40.0,0.0), height = 15, width = 84, text = 'Velocidade angular')
```

```
w1.value = 0.0
```

```
w2 = -w1.value * re1 / re2
```

```
t = 0.0; dt = 0.01
```

```
#####
```

```
phi = 0.0
```

```
if (nr == 1 or nr == 3 or nr == 5):
```

```
    fixa1.rotate(axis = gira, angle = pi / 2, origin = engrena1.pos) # altera a fase (ângulo) inicial da rotação da engrenagem
```

```
    phi = - math.pi/2 # fase inicial
```

```
#####
```

```
largvet = 0.037 * re1 # espessura/largura do vetor
```

```
fatv=1.0; a = engrena1.pos.x; b = engrena2.pos.x; c = engrena3.pos.x
```

```
fatw = 0.5
```

```
altura = 25
```

```
define = false # define a visibilidade dos valores das grandezas físicas
```

```
posic1 = arrow(frame = fixa1, pos=(engrena1.pos.x,engrena1.pos.y,engrena1.pos.z+engrena1.length),axis=(re1*cos(w1.value*t + phi),re1*sin(w1.value*t + phi),0.0), shaftwidth=largvet,color=color.magenta)# definindo vetor r raio da orbita (posicao da particula no eixo)
```

```
raio1_rot = label(pos = (a, 1.15, 0.0), text = 'r1 = %1.2f m' % re1, height = altura, opacity = 0, box = false, visible = define, color = color.magenta)
```

```

veloc1 = arrow(frame = fixa1, pos = (re1*cos(w1.value*t + phi) +
a,re1*sin(w1.value*t + phi), engrena1.length), axis = (-
fatv*w1.value*re1*sin(w1.value*t + phi),fatv*w1.value*re1*cos(w1.value*t + phi),
0.0),
shaftwidth = largvet, color=color.red)

```

```

veloc1_rot = label(pos = (a, raio1_rot.pos.y - 0.12, 0.0), text = 'v1 = %1.2f m/s' %
veloc1.axis.mag, height = altura, opacity = 0, box = false, visible = define, color =
color.red)

```

```

omega1 = arrow(pos = (engrena1.pos.x, engrena1.pos.y, pino1.length), axis =
(0.0, 0.0, fatw*w1.value), shaftwidth = largvet, color = color.white) # vetor
velocidade angular w1

```

```

omega1_rot = label(pos = (a, veloc1_rot.pos.y - 0.12, 0.0), text = 'w1 = %1.2f
rad/s' % w1.value, height = altura, opacity = 0, box = false, visible = define, color =
color.white)

```

```

posic2 = arrow(frame = fixa2, pos =
(engrena2.pos.x,engrena2.pos.y,engrena2.pos.z+engrena2.length), axis =
(re2*cos(w2*t),re2*sin(w2*t),0.0),shaftwidth = largvet, color = color.magenta)

```

```

raio2_rot = label(pos = (b, -0.8, 0.0), text = 'r2 = %1.2f m' % re2, height = altura,
opacity = 0, box = false, visible = define, color = color.magenta)

```

```

veloc2 = arrow(frame = fixa2, pos=(re2*cos(w2*t) +
b,re2*sin(w2*t),engrena2.length),axis=(0.0,fatv*w2*re2,0.0),shaftwidth=largvet,col
or=color.red)

```

```

veloc2_rot = label(pos = (b, raio2_rot.pos.y - 0.12, 0.0), text = 'v1 = %1.2f m/s' % -
veloc2.axis.mag, height = altura, opacity = 0, box = false, visible = define, color =
color.red)

```

```

omega2 = arrow(pos = (engrena2.pos.x, engrena2.pos.y, -pino2.length), axis =
(0.0, 0.0, fatw*w2), shaftwidth = largvet, color = color.white) # vetor velocidade
angular w2

```

```

omega2_rot = label(pos = (b, veloc2_rot.pos.y - 0.12, 0.0), text = 'w2 = %1.2f
rad/s' % w2, height = altura, opacity = 0, box = false, visible = define, color =
color.white)

```

```

posic3 = arrow(frame = fixa1, pos =
(engrena3.pos.x,engrena3.pos.y,engrena3.pos.z+engrena3.length), axis =
(re3*cos(w1.value*t + phi),re3*sin(w1.value*t + phi),0.0),shaftwidth = largvet, color
= color.magenta)

```

```

raio3_rot = label(pos = (c - 0.7, -0.95, 0.0), text = 'r3 = %1.2f m' % re3, height =
altura, opacity = 0, box = false, visible = define, color = color.magenta)

```

```

veloc3 = arrow(frame = fixa1, pos=(re3*cos(w1.value*t + phi) +
c,re3*sin(w1.value*t + phi), engrena3.pos.z + engrena3.length),axis=(-
fatv*w1.value*re3*sin(w1.value*t + phi),fatv*w1.value*re3*cos(w1.value*t +
phi),0.0),
shaftwidth = largvet,color=color.red)

```

```

veloc3_rot = label(pos = (c - 0.7, raio3_rot.pos.y - 0.12, 0.0), text = 'v3 = %1.2f
m/s' % veloc3.axis.mag, height = altura, opacity = 0, box = false, visible = define,
color = color.red)

```

```

omega3_rot = label(pos = (c - 0.7, veloc3_rot.pos.y - 0.12, 0.0), text = 'w3 = %1.2f
rad/s' % w1.value, height = altura, opacity = 0, box = false, visible = define, color =
color.white)

```

while True:

```

    if scene.mouse.clicked:
        scene.mouse.getclick()
        if (define == false):
            define = true
        else:
            define = false

```

```

    rate(100)

```

```

    painel.interact()

```

```

    fixa1.rotate(axis = gira, angle = w1.value*dt, origin = engrena1.pos)

```

```

    w2 = -w1.value * re1 / re2

```

```

    fixa2.rotate(axis = gira, angle = w2*dt, origin = engrena2.pos)

```

```

    veloc1.axis=(-fatv*w1.value*re1*sin(w1.value*dt + phi),fatv*w1.value*re1*cos(w1.value*dt + phi), 0.0) # atualização das componentes
do vetor mantenha a tangente à engrenagem

```

```

    raio1_rot.visible = define

```

```

    veloc1_rot.text = 'v1 = %1.2f m/s' % veloc1.axis.mag; veloc1_rot.visible = define

```

```

    omega1.axis.z = fatw * w1.value

```

```

    omega1_rot.text = 'w1 = %1.2f rad/s' % w1.value; omega1_rot.visible = define

```

```

    veloc2.axis=(-fatv*w2*re2*sin(w2*dt),fatv*w2*re2*cos(w2*dt),0.0)# atualização
das componentes do vetor mantenha a tangente à ingrenagem

```

```

    raio2_rot.visible = define

```

```
veloc2_rot.text = 'v2 = %1.2f m/s' % -veloc2.axis.mag; veloc2_rot.visible =  
define
```

```
omega2.axis.z = fatw * w2
```

```
omega2_rot.text = 'w2 = %1.2f rad/s' % w2; omega2_rot.visible = define
```

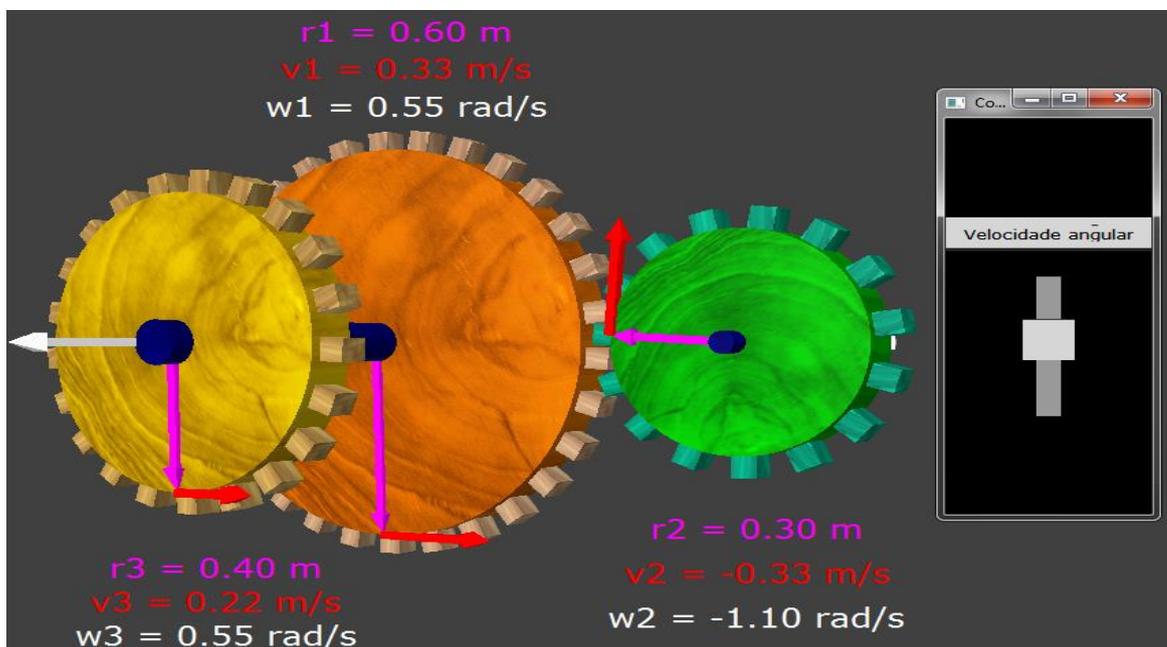
```
veloc3.axis=(-fatv*w1.value*re3*sin(w1.value*dt + phi),fatv*w1.value*re3*cos(w1.value*dt + phi),0.0) # atualização das componentes  
do vetor mantenha a tangente à ingrenagem
```

```
raio3_rot.visible = define
```

```
veloc3_rot.text = 'v3 = %1.2f m/s' % veloc3.axis.mag; veloc3_rot.visible = define
```

```
omega3_rot.text = 'w3 = %1.2f rad/s' % w1.value; omega3_rot.visible = define
```

**FIGURA ILUSTRATIVA do simulador - Engrenagens com cursor mudança de velocidade angular**



## CÓDIGO ONTE DO SIMULADOR - TERRA (COM CURSOR DE MUDANÇA DA VELOCIDADE ANGULAR)

```
from visual import*

from visual.controls import*

from visual.text import*

import math

scene = display(width=1200,height=1500,background=(0.25,0.25,0.25))# controla
o tamanho da tela da animação - (Width=largura - height=altura)

scene.range = 1.0 # zoom da tela

scene.forward = (0.1,0.1,-0.3)# posicionamento da câmera

scene.ambient = 0.6

'''
def omega(valor):
    w = valor
'''

gira = (0.0,1.0,0.0)

fixa = frame()

tamalt = 220 # altura/comprimento para cima do controlado

tamlarg = 800 # largura do controlador

painel = controls(x = 1200, y = 0, width = tamalt, height = tamlarg, range = 100)#
regula a posição do controlador - height=altura - width=largura

w = slider(pos=(0.0,-50.0), width = 20.0, length = 70.0, axis = (0,1,0), min = 0.0,
max = 0.05, action = lambda: w)

menu(pos=(0.0,40.0,0.0), height = 15, width = 84, text = 'Velocidade angular')

w.value = 0.0

r0 = 0.45; k = 1.5

raio = k * abs(w.value) + r0

fh = r0 * r0 / raio
```

```

terra = ellipsoid(frame = fixa, pos=(0.0,0.0,0.0), length = 2*raio, height = 2*fh,
width = 2*raio, color=color.white, opacity=0.7, material = materials.earth) # Terra

eixo = cylinder(frame = fixa, radius = 0.006, pos=(0.0,-1.05 * terra.height / 2,0.0),
axis=(0.0,1.05 * terra.height,0.0), color = color.white) # eixo de rotação da Terra

'''
rc = 0.01 * raio # espessura dos rastros

rastro = curve(frame = fixa, radius = rc, color = color.green, material =
materials.plastic) # rastro da órbita do equador

rastro1 = curve(frame = fixa, radius = rc, color = color.green, material =
materials.plastic) # rastro da órbita do meio

rastro2 = curve(frame = fixa, radius = rc, color = color.green, material =
materials.plastic) # rastro da órbita menor

'''

f = 0.4 # fator que determina a altura em y da órbita do meio

f1 = 0.8 # fator que determina a altura em y da órbita menor

#fh = falt / 2.0 # fator que define a altura da Terra em termos do seu raio no plano
x-z em y = 0

'''
delta = 1.0 + 4.0 * (fh * fh - 1.0) * (f * fh) * (f * fh)

raiz = (-1.0 + sqrt(delta)) / (fh * fh - 1.0) / 2.0

theta1 = arccos(sqrt(raiz))

delta1 = 1.0 + 4.0 * (fh * fh - 1.0) * (f1 * fh) * (f1 * fh)

raiz1 = (-1.0 + sqrt(delta1)) / (fh * fh - 1.0) / 2.0

theta2 = arccos(sqrt(raiz1))

rteta = raio

rteta1 = sqrt(1.0 - (f * fh) * (f * fh) + (fh * fh - 1.0) * cos(theta1) * cos(theta1)) * raio

rteta2 = sqrt(1.0 - (f1 * fh) * (f1 * fh) + (fh * fh - 1.0) * cos(theta2) * cos(theta2)) *
raio

'''

```

```
largvet = 0.022 * raio # espessura/largura dos vetores
```

```
fat = 10.0 # fator que determina o tamanho do vetor, serve para almentar o tamanho visual do mesmo
```

```
fce = fat
```

```
theta = math.pi/2.0
```

```
theta1 = arccos(f)
```

```
theta2 = arccos(f1)
```

```
posic = arrow(frame = fixa, pos = (0.0, 0.0, 0.0), axis = (raio, 0.0, 0.0), shaftwidth = largvet, color = color.yellow) # definindo vetor r raio da orbita (posicao da particula no eixo)
```

```
posic1 = arrow(frame = fixa, pos = (0.0, fh * cos(theta1), 0.0), axis = (raio * sin(theta1), 0.0, 0.0), shaftwidth = largvet, color = color.yellow) # definindo vetor r raio da orbita (posicao da particula no eixo)
```

```
posic2 = arrow(frame = fixa, pos = (0.0, fh * cos(theta2), 0.0), axis = (raio * sin(theta2), 0.0, 0.0), shaftwidth = largvet, color = color.yellow) # definindo vetor r raio da orbita (posicao da particula no eixo)
```

```
veloc = arrow(frame = fixa, pos = (raio, 0.0,0.0), axis = (0.0, 0.0, w.value * raio),shaftwidth = largvet,color = color.red) # coordenadas do vetor velocidade na origem
```

```
veloc1 = arrow(frame = fixa, pos = (raio * sin(theta1), fh * cos(theta1), 0.0), axis = (0.0, 0.0, w.value * raio * sin(theta1)), shaftwidth = largvet, color = color.red)# coordenadas do vetor velocidade órbita do meio
```

```
veloc2 = arrow(frame = fixa, pos = (raio * sin(theta2), fh * cos(theta2), 0.0), axis = (0.0, 0.0, w.value * raio * sin(theta2)), shaftwidth = largvet, color = color.red)# coordenadas do vetor velocidade órbita
```

```
massa = 0.013
```

```
g = 9.8 # valor da aceleração da gravidade
```

```
peso = arrow(frame = fixa, pos = ((1.0 + fce * fat * w.value * w.value ) * raio + massa * g, 0.0, 0.0), axis = (-massa * g, 0.0, 0.0), shaftwidth = largvet, color = color.cyan)
```

```
peso1 = arrow(frame = fixa, pos = ((raio + massa*g) * sin(theta1), (fh + massa*g) *  
cos(theta1), 0.0),  
axis = (-massa*g*sin(theta1), -massa*g*cos(theta1), 0.0), shaftwidth =  
largvet, color = color.cyan) # definindo o vetor força (peso), rastro médio
```

```
peso2 = arrow(frame = fixa, pos = ((raio + massa*g) * sin(theta2), (fh + massa*g) *  
cos(theta2), 0.0),  
axis = (-massa*g*sin(theta2), -massa*g*cos(theta2), 0.0), shaftwidth =  
largvet, color = color.cyan) # definindo o vetor força (peso), rastro menor
```

```
peso3 = arrow(frame = fixa, pos = (0.0, (fh + massa*g), 0.0), axis = (0.0, -massa*g,  
0.0), shaftwidth = largvet, color = color.cyan) # definindo o vetor força (peso),  
rastro menor
```

```
centrif = arrow(frame = fixa, pos = (raio, 0.0, 0.0), axis = (fce * fat * w.value *  
w.value * raio, 0.0, 0.0), shaftwidth = largvet) # definindo o vetor força centrífuga
```

```
centrif1 = arrow(frame = fixa, pos = (raio * sin(theta1), fh * cos(theta1), 0.0), axis =  
(fce * fat * w.value * w.value * raio * sin(theta1), 0.0, 0.0), shaftwidth = largvet) #  
definindo o vetor força centrífuga
```

```
centrif2 = arrow(frame = fixa, pos = (raio * sin(theta2), fh * cos(theta2), 0.0), axis =  
(fce * fat * w.value * w.value * raio * sin(theta2), 0.0, 0.0), shaftwidth = largvet) #  
definindo o vetor força centrífuga
```

```
t = 0.0; dt = 0.1 # tempo e o seu incremento
```

```
while True:
```

```
rate (100) # velocidade de visualização do movimento
```

```
painel.interact()
```

```
raio = k * abs(w.value) + r0
```

```
fh = r0 * r0 / raio
```

```
terra.length = 2*raio; terra.height = 2*fh; terra.width = 2*raio
```

```
eixo.pos.y = -1.05 * terra.height / 2
```

```
eixo.axis.y = 1.05 * terra.height
```

```
t = t + dt # define a atualização do tempo para o rastro
```

#####

posic.axis = (raio \* cos(w.value\*dt), 0.0, -raio \* sin(w.value\*dt)) # atualização das componentes do raio (r), vetor verde em cima da engrenagem

posic1.pos.y = fh \* cos(theta1)

posic1.axis = (raio \* sin(theta1) \* cos(w.value\*dt), 0.0, -raio \* sin(theta1) \* sin(w.value\*dt)) # atualização das componentes do raio (r), vetor verde em cima da engrenagem

posic2.pos.y = fh \* cos(theta2)

posic2.axis = (raio \* sin(theta2) \* cos(w.value\*dt), 0.0, -raio \* sin(theta2) \* sin(w.value\*dt)) # atualização das componentes do raio (r), vetor verde em cima da engrenagem

#####

veloc.pos = (raio \* cos(w.value\*dt), 0.0, raio \* sin(w.value\*dt)) # atualizar a posição x do vetor velocidade e raio(r), rastro da terra origem

veloc.axis = (fat \* w.value \* raio \* sin(w.value\*dt), 0.0, -fat \* w.value \* raio \* cos(w.value\*dt)) # atualização das componentes x do vetor mantenha a tangente do rastro da terra

veloc1.pos = (raio \* sin(theta1) \* cos(w.value\*dt), fh \* cos(theta1), raio \* sin(theta1) \* sin(w.value\*dt))

veloc1.axis = (fat \* w.value \* raio \* sin(theta1) \* sin(w.value\*dt), 0.0, -fat \* w.value \* raio \* sin(theta1) \* cos(w.value\*dt)) # atualização das componentes x do vetor mantenha a tangente do rastro da terra

veloc2.pos = (raio \* sin(theta2) \* cos(w.value\*dt), fh \* cos(theta2), raio \* sin(theta2) \* sin(w.value\*dt)) # atualizar a posição x do vetor velocidade e raio(r), rastro da terra origem

veloc2.axis = (fat \* w.value \* raio \* sin(theta2) \* sin(w.value\*dt), 0.0, -fat \* w.value \* raio \* sin(theta2) \* cos(w.value\*dt)) # atualização das componentes x do vetor mantenha a tangente do rastro da terra

#####

peso.pos.x = ((1.0 + fce \* fat \* w.value \* w.value) \* raio + massa \* g) \* cos(w.value\*dt)

peso.pos.z = (-(1.0 + fce \* fat \* w.value \* w.value) \* raio + massa \* g) \* sin(w.value\*dt)

```
peso.axis = (-massa * g * cos(w.value*dt), 0.0, massa * g * sin(w.value*dt))
peso1.pos = ((raio + massa*g) * sin(theta1) * cos(w.value*dt), (fh + massa*g) *
cos(theta1), -(raio + massa*g) * sin(theta1) * sin(w.value*dt))
```

```
peso1.axis.x = -massa * g * sin(theta1) * cos(w.value*dt)
```

```
peso1.axis.z = massa * g * sin(theta1) * sin(w.value*dt)
```

```
peso2.pos = ((raio + massa*g) * sin(theta2) * cos(w.value*dt), (fh + massa*g) *
cos(theta2), -(raio + massa*g) * sin(theta2) * sin(w.value*dt))
```

```
peso2.axis.x = -massa * g * sin(theta2) * cos(w.value*dt)
```

```
peso2.axis.z = massa * g * sin(theta2) * sin(w.value*dt)
```

```
peso3.pos.y = fh + massa*g
```

```
#####
```

```
centrif.pos = (raio * cos(w.value*dt), 0.0, -raio * sin(w.value*dt))
```

```
centrif.axis = (fce * fat * w.value * w.value * raio * cos(w.value*dt), 0.0, -fce * fat
* w.value * w.value * raio * sin(w.value*dt))
```

```
centrif1.pos = (raio * sin(theta1) * cos(w.value*dt), fh * cos(theta1), -raio *
sin(theta1) * sin(w.value*dt))
```

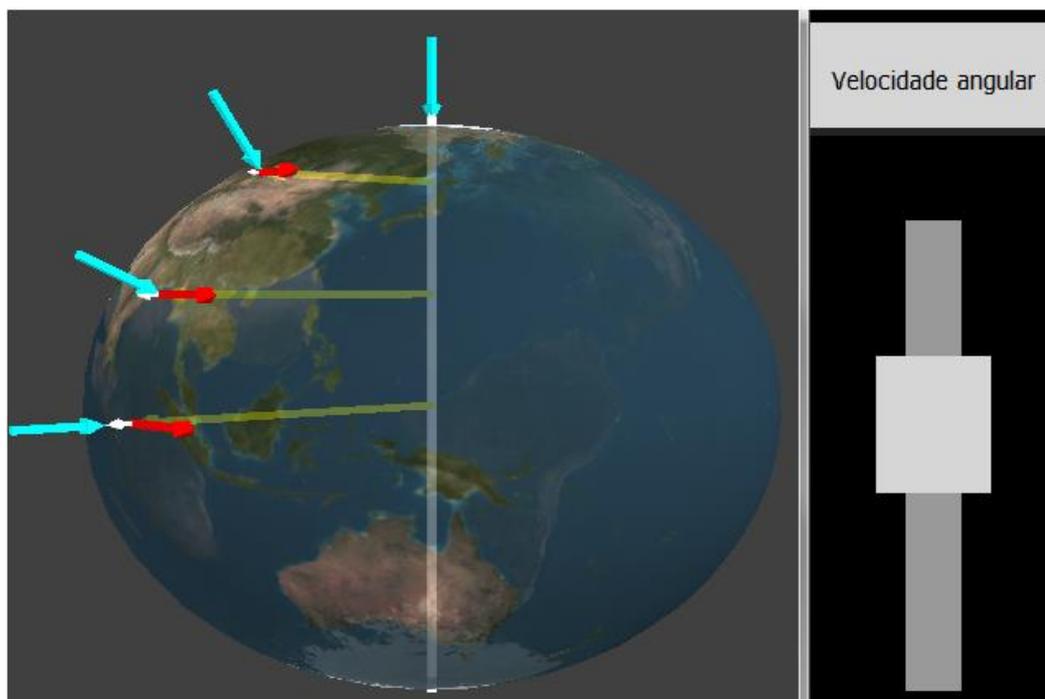
```
centrif1.axis = (fce * fat * w.value * w.value * raio * sin(theta1) * cos(w.value*dt),
0.0, -fce * fat * w.value * w.value * raio * sin(theta1) * sin(w.value*dt))
```

```
centrif2.pos = (raio * sin(theta2) * cos(w.value*dt), fh * cos(theta2), -raio *
sin(theta2) * sin(w.value*dt))
```

```
centrif2.axis = (fce * fat * w.value * w.value * raio * sin(theta2) * cos(w.value*dt),
0.0, -fce * fat * w.value * w.value * raio * sin(theta2) * sin(w.value*dt))
```

```
fixa.rotate(axis = gira, angle = w.value*dt, origin = terra.pos) # rotação da terra
```

**FIGURA ILUSTRATIVA DO SIMULADOR MOVIMENTO ROTACIONAL DA TERRA - Com cursor de mudança da velocidade angular**



## CÓDIGO FONTE - TERRA (LANÇAMENTO DE OBJETOS)

```
from visual import*

from visual.text import*

import math

scene.fullscreen = 1 # cena de tela cheia

scene.range = 1.0 # zoom da tela

scene.ambient = 0.6 # cor da terra mais esbrançada ou fundo preto

raio = 0.45 # raio do esferóide na linha do equador (plano x-z)

falt = 1.7

terra = ellipsoid(pos=(0.0,0.0,0.0), length = 2*raio, height = falt*raio, width = 2*raio,
color=color.white, opacity=0.7, material = materials.earth) # Terra

eixo = cylinder(radius = 0.006, pos=(0.0,-1.05*terra.height/2,0.0),
axis=(0.0,1.05*terra.height,0.0), color = color.white) # eixo de rotação da Terra

rc = 0.01 * raio # espessura dos rastros

rastro = curve(radius = rc, color = color.green, material = materials.plastic) # rastro
da órbita do equador

rastro1 = curve(radius = rc, color = color.green, material = materials.plastic) #
rastro da órbita do meio

rastro2 = curve(radius = rc, color = color.green, material = materials.plastic) #
rastro da órbita menor

f = 0.4 # fator que determina a altura em y da órbita do meio

f1 = 0.8 # fator que determina a altura em y da órbita menor

fh = falt / 2.0 # fator que define a altura da Terra em termos do seu raio no plano
x-z em y = 0

'''
delta = 1.0 + 4.0 * (fh * fh - 1.0) * (f * fh) * (f * fh)
```

```
raiz = (-1.0 + sqrt(delta)) / (fh * fh - 1.0) / 2.0
```

```
theta1 = arccos(sqrt(raiz))
```

```
delta1 = 1.0 + 4.0 * (fh * fh - 1.0) * (f1 * fh) * (f1 * fh)
```

```
raiz1 = (-1.0 + sqrt(delta1)) / (fh * fh - 1.0) / 2.0
```

```
theta2 = arccos(sqrt(raiz1))
```

```
rteta = raio
```

```
rteta1 = sqrt(1.0 - (f * fh) * (f * fh) + (fh * fh - 1.0) * cos(theta1) * cos(theta1)) * raio
```

```
rteta2 = sqrt(1.0 - (f1 * fh) * (f1 * fh) + (fh * fh - 1.0) * cos(theta2) * cos(theta2)) *  
raio  
'''
```

```
largvet = 0.027 * raio # espessura/largura dos vetores
```

```
fat = 10.0 # fator que determina o tamanho do vetor, serve para almentar o  
tamanho visual do mesmo
```

```
fce = fat
```

```
theta = math.pi/2.0
```

```
theta1 = arccos(f)
```

```
theta2 = arccos(f1)
```

```
posic = arrow(pos = (0.0, 0.0, 0.0), axis = (raio, 0.0, 0.0), shaftwidth = largvet,  
color = color.red) # definindo vetor r raio da orbita (posicao da particula no eixo)
```

```
posic1 = arrow(pos = (0.0, fh * raio * cos(theta1), 0.0), axis = (raio * sin(theta1),  
0.0, 0.0), shaftwidth = largvet, color = color.orange) # definindo vetor r raio da  
orbita (posicao da particula no eixo)
```

```
posic2 = arrow(pos = (0.0, fh * raio * cos(theta2), 0.0), axis = (raio * sin(theta2),  
0.0, 0.0), shaftwidth = largvet, color = color.magenta) # definindo vetor r raio da  
orbita (posicao da particula no eixo)
```

```
w = -0.05 # velocidade angular da Terra
```

```
#w = 0.0
```

```
veloc = arrow(pos = (raio, 0.0,0.0), axis = (0.0, 0.0, w * raio),shaftwidth = largvet,color = color.red) # coordenadas do vetor velocidade na origem
```

```
veloc1 = arrow(pos = (raio * sin(theta1), fh * raio * cos(theta1), 0.0), axis = (0.0, 0.0, w * raio * sin(theta1)), shaftwidth = largvet, color = color.orange)# coordenadas do vetor velocidade órbita do meio
```

```
veloc2 = arrow(pos = (raio * sin(theta2), fh * raio * cos(theta2), 0.0), axis = (0.0, 0.0, w * raio * sin(theta2)), shaftwidth = largvet, color = color.magenta)# coordenadas do vetor velocidade órbita
```

```
massa = 0.013
```

```
g = 9.8 # valor da aceleração da gravidade
```

```
peso = arrow(pos = ((1.0 + fce * fat * w * w ) * raio + massa * g, 0.0, 0.0), axis = (- massa * g, 0.0, 0.0), shaftwidth = largvet, color = color.cyan)
```

```
peso1 = arrow(pos = ((raio + massa*g) * sin(theta1), (fh * raio + massa*g) * cos(theta1), 0.0), axis = (-massa*g*sin(theta1), -massa*g*cos(theta1), 0.0), shaftwidth = largvet, color = color.cyan) # definindo o vetor força (peso), rastro médio
```

```
peso2 = arrow(pos = ((raio + massa*g) * sin(theta2), (fh * raio + massa*g) * cos(theta2), 0.0), axis = (-massa*g*sin(theta2), -massa*g*cos(theta2), 0.0), shaftwidth = largvet, color = color.cyan) # definindo o vetor força (peso), rastro menor
```

```
peso3 = arrow(pos = (0.0, (fh * raio + massa*g), 0.0), axis = (0.0, -massa*g, 0.0), shaftwidth = largvet, color = color.cyan) # definindo o vetor força (peso), rastro menor
```

```
centrif = arrow(pos = (raio, 0.0, 0.0), axis = (fce * fat * w * w * raio, 0.0, 0.0), shaftwidth = largvet) # definindo o vetor força centrífuga
```

```
centrif1 = arrow(pos = (raio * sin(theta1), fh * raio * cos(theta1), 0.0), axis = (fce * fat * w * w * raio * sin(theta1), 0.0, 0.0), shaftwidth = largvet) # definindo o vetor força centrífuga
```

```
centrif2 = arrow(pos = (raio * sin(theta2), fh * raio * cos(theta2), 0.0), axis = (fce * fat * w * w * raio * sin(theta2), 0.0, 0.0), shaftwidth = largvet) # definindo o vetor força centrífuga
```

```
t = 0.0; dt = 0.1 # tempo e o seu incremento
```

```
nulo = (0.0, 0.0, 0.0)
```

```

while True:

    rate (100) # velocidade de visualização do movimento

    t = t + dt # define a atualização do tempo para o rastro
    phi = w * t

    if scene.mouse.clicked:

        t = 0.0; dt = 1.e-2

        while (t <= 5.0):

            rate (100) # velocidade de visualização do movimento

            veloc.pos = veloc.pos + (veloc.axis.x * dt, 0.0, veloc.axis.z * dt)

            veloc1.pos = veloc1.pos + (veloc1.axis.x * dt, 0.0, veloc1.axis.z * dt)

            veloc2.pos = veloc2.pos + (veloc2.axis.x * dt, 0.0, veloc2.axis.z * dt)

            peso.axis = nulo; peso1.axis = nulo; peso2.axis = nulo; peso3.axis = nulo

            centrif.axis = nulo; centrif1.axis = nulo; centrif2.axis = nulo;

            rastro.append(pos = veloc.pos)

            rastro1.append(pos = veloc1.pos)

            rastro2.append(pos = veloc2.pos)

            t = t + dt

        break

    rastro.append(pos = (raio * cos(w*t), 0.0, raio * sin(w*t))) # rasto da terra, (o
negativo no seno inverte o movimento do rastro) - equador

    rastro1.append(pos=(raio * sin(theta1) * cos(w*t), fh * raio * cos(theta1), raio *
sin(theta1) * sin(w*t)))# rasto da terra, (o negativo no seno inverte o movimento do
rastro) - meio

    rastro2.append(pos=(raio * sin(theta2) * cos(w*t), fh * raio * cos(theta2), raio *
sin(theta2) * sin(w*t)))# rasto da terra, (o negativo no seno inverte o movimento do
rastro) - menor

#####

```

posic.axis = (raio \* cos(w\*t), 0.0, raio \* sin(w\*t)) # atualização das componentes do raio (r), vetor verde em cima da engrenagem

posic1.axis = (raio \* sin(theta1) \* cos(w\*t), 0.0, raio \* sin(theta1) \* sin(w\*t)) # atualização das componentes do raio (r), vetor verde em cima da engrenagem

posic2.axis = (raio \* sin(theta2) \* cos(w\*t), 0.0, raio \* sin(theta2) \* sin(w\*t)) # atualização das componentes do raio (r), vetor verde em cima da engrenagem

#####

veloc.pos = (raio \* cos(w\*t), 0.0, raio \* sin(w\*t)) # atualizar a posição x do vetor velocidade e raio(r), rastro da terra origem

veloc.axis = (-fat \* w \* raio \* sin(w\*t), 0.0, fat \* w \* raio \* cos(w\*t)) # atualização das componentes x do vetor mantenha a tangente do rastro da terra

veloc1.pos.x = raio \* sin(theta1) \* cos(w\*t) # atualizar a posição x do vetor velocidade e raio(r), rastro da terra origem

veloc1.pos.z = raio \* sin(theta1) \* sin(w\*t)

veloc1.axis = (-fat \* w \* raio \* sin(theta1) \* sin(w\*t), 0.0, fat \* w \* raio \* sin(theta1) \* cos(w\*t)) # atualização das componentes x do vetor mantenha a tangente do rastro da terra

veloc2.pos.x = raio \* sin(theta2) \* cos(w\*t) # atualizar a posição x do vetor velocidade e raio(r), rastro da terra origem

veloc2.pos.z = raio \* sin(theta2) \* sin(w\*t) # atualizar a posição z do vetor velocidade e raio(r), rastro da terra origem

veloc2.axis = (-fat \* w \* raio \* sin(theta2) \* sin(w\*t), 0.0, fat \* w \* raio \* sin(theta2) \* cos(w\*t)) # atualização das componentes x do vetor mantenha a tangente do rastro da terra

#####

peso.pos.x = ((1.0 + fce \* fat \* w \* w) \* raio + massa \* g) \* cos(w\*t)

peso.pos.z = ((1.0 + fce \* fat \* w \* w) \* raio + massa \* g) \* sin(w\*t)

peso.axis = (-massa \* g \* cos(w\*t), 0.0, -massa \* g \* sin(w\*t))

peso1.pos.x = (raio + massa\*g) \* sin(theta1) \* cos(w\*t)

peso1.pos.z = (raio + massa\*g) \* sin(theta1) \* sin(w\*t)

```

peso1.axis.x = -massa * g * sin(theta1) * cos(w*t)
peso1.axis.z = -massa * g * sin(theta1) * sin(w*t)

peso2.pos.x = (raio + massa*g) * sin(theta2) * cos(w*t)
peso2.pos.z = (raio + massa*g) * sin(theta2) * sin(w*t)

peso2.axis.x = -massa * g * sin(theta2) * cos(w*t)
peso2.axis.z = -massa * g * sin(theta2) * sin(w*t)

#####

centrif.pos = (raio * cos(w*t), 0.0, raio * sin(w*t))

centrif.axis = (fce * fat * w * w * raio * cos(w*t), 0.0, fce * fat * w * w * raio *
sin(w*t))

centrif1.pos = (raio * sin(theta1) * cos(w*t), fh * raio * cos(theta1), raio *
sin(theta1) * sin(w*t))

centrif1.axis = (fce * fat * w * w * raio * sin(theta1) * cos(w*t), 0.0, fce * fat * w *
w * raio * sin(theta1) * sin(w*t))

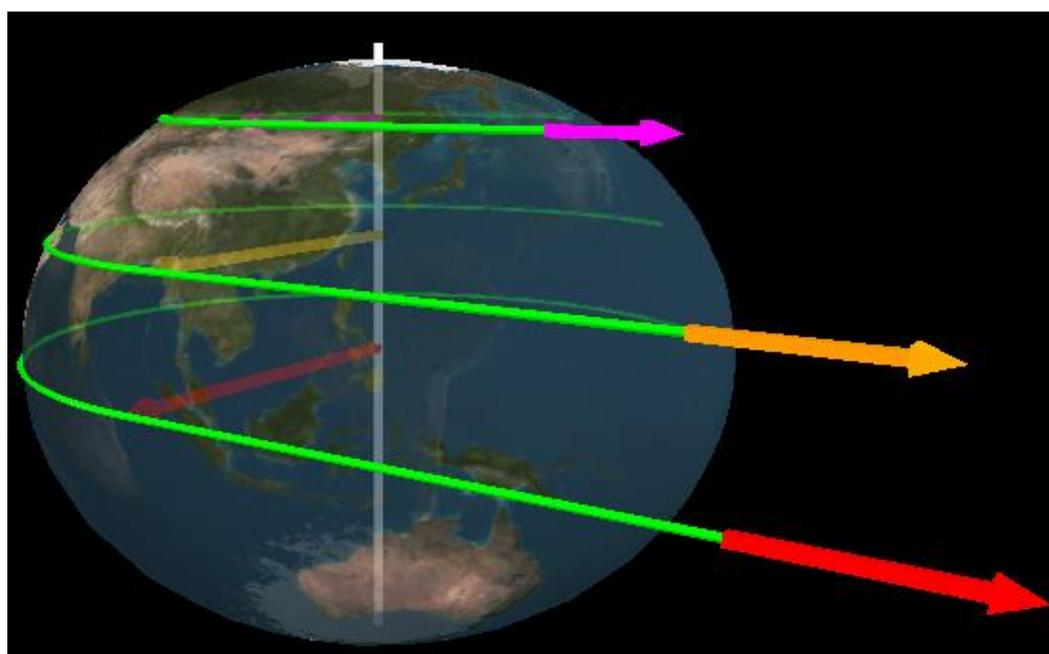
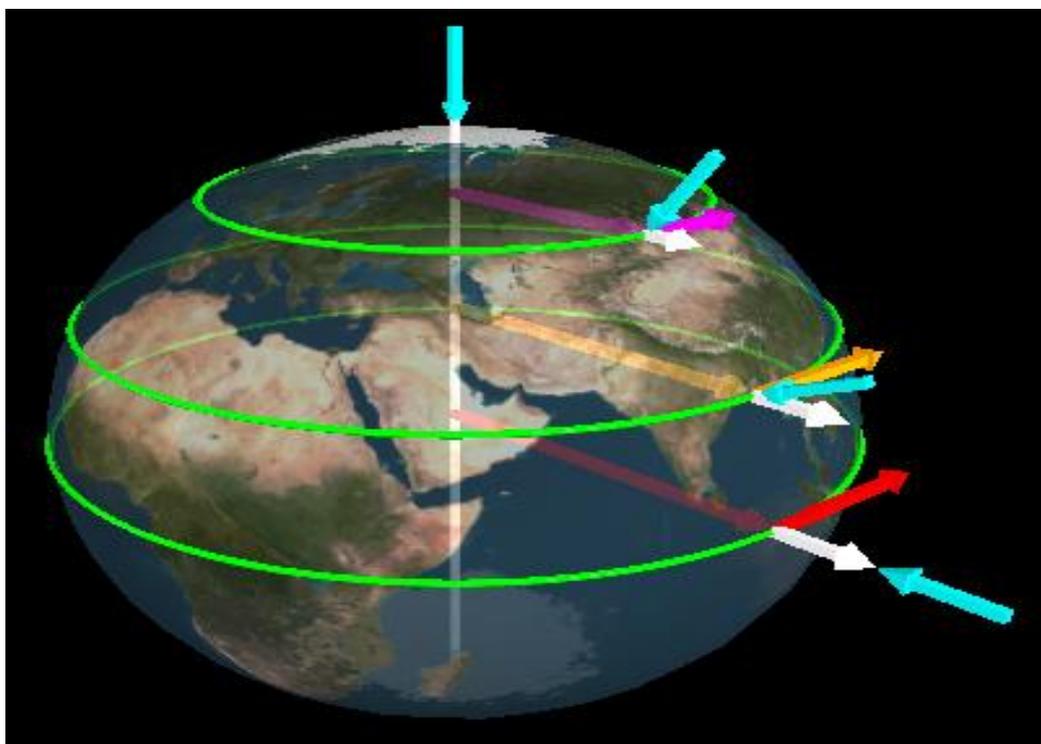
centrif2.pos = (raio * sin(theta2) * cos(w*t), fh * raio * cos(theta2), raio *
sin(theta2) * sin(w*t))

centrif2.axis = (fce * fat * w * w * raio * sin(theta2) * cos(w*t), 0.0, fce * fat * w *
w * raio * sin(theta2) * sin(w*t))

terra.rotate(axis = (0.0,1.0,0.0), angle = -w*dt, origin = (0.0,0.0,0.0)) # rotação
da terra

```

FIGURA ILUSTRATIVA DO SIMULADOR MOVIMENTO ROTACIONAL DA TERRA - Lançamento de objetos



## QUESTIONÁRIO 02 - AVALIATIVO

1 - Nome do aluno. (Não obrigatório)

---

2 - Com base nas aulas práticas com os simuladores, assinale a(s) situação(ões) em que ocorre o movimento circular uniforme:

- ( ) - Na caminhada de uma pessoa numa avenida reta.
- ( ) - Na roda de uma moto.
- ( ) - Ao beber uma água.
- ( ) - Em um carro contornando uma rotatória.
- ( ) - Na hélice de um ventilador.
- ( ) - No movimento de um satélite em órbita.

**Justifique sua resposta:**

---

---

---

---

3 - Observe a imagem abaixo da animação que representa uma situação estudada previamente e assinale a alternativa correta:

- ( ) Desprende-se da Terra por causa da alta velocidade adquirida.
- ( ) Desprende-se devido a força centrípeta.
- ( ) Desprende-se devido a ausência de forças.
- ( ) Nenhuma das alternativas estão corretas.

**Justifique sua resposta:**

---

---

---

---

4 - Assinale a proposição que representa corretamente o movimento circular uniforme de um objeto:

- ( ) Não existe força atuando sobre o objeto.
- ( ) A velocidade linear do objeto não se altera.
- ( ) O movimento circular do objeto é uniforme porque não possui aceleração.
- ( ) O valor (módulo) da velocidade linear é constante
- ( ) Nenhuma das alternativas estão corretas.

**Justifique sua resposta:**

---

---

---

---

5 - Observe a figura e responda: Três carros de corrida posicionados em cada uma das raias (1-rosa, 2-azul e 3-amarela). Considerando que eles iniciam a corrida no mesmo instante percorrendo todo o trajeto com a mesma velocidade linear, qual(is) carro(s) deve(m) vencer a corrida.

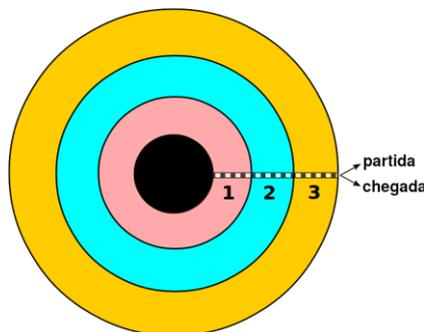


Figura 2

- ( ) O carro da raia 2.
- ( ) Não existirá um único vencedor pois todos chegarão ao mesmo tempo.
- ( ) O carro da raia 1.
- ( ) O carro da raia 3.
- ( ) Não existirá um único vencedor pois os carros das raias 2 e 3 chegarão juntos.
- ( ) Nenhuma das alternativas estão corretas.

**Justifique sua resposta:**

---

---

6 - Observando a imagem do simulador das engrenagens, marque a resposta correta. Com relação as engrenagens 1 e 2 (ambas acopladas pelo mesmo eixo), o que podemos afirmar sobre as velocidades.

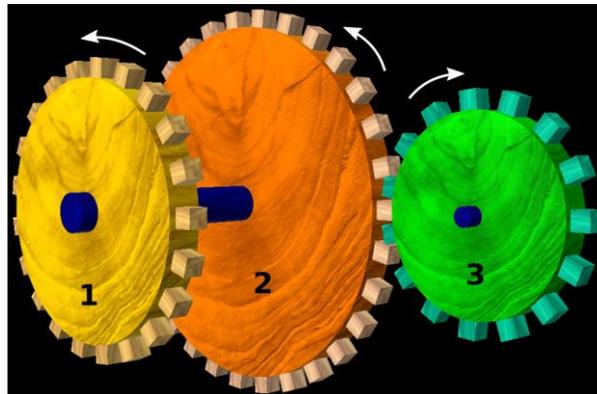


Figura 3

- ( ) As velocidades lineares das engrenagens são iguais.
- ( ) A velocidade angular da engrenagem 1 é menor que o da engrenagem 2.
- ( ) A velocidade linear da engrenagem 1 é maior que o da engrenagem 2.
- ( ) As velocidades angulares das engrenagens são iguais.
- ( ) Nenhuma das alternativas estão corretas.

**Justifique a sua resposta:**

---

---

---

---

7 - Continuando a observação da imagem do simulador das engrenagens(da questão anterior), marque a resposta correta. Com relação as engrenagens 2 e 3 (ambas acopladas pelos dentes), o que podemos afirmar sobre as velocidades.

- ( ) As velocidades angulares das engrenagens são iguais.
- ( ) A velocidade angular da engrenagem 2 é maior que o da engrenagem 3.
- ( ) A velocidade linear da engrenagem 2 é menor que o da engrenagem 3.
- ( ) As velocidades lineares das engrenagens são iguais.
- ( ) Nenhuma das alternativas estão corretas.

**Justifique sua resposta:**

---

---

8 - Marque o(s) item(ns) que julgar correto. Em uma pista de atletismo, onde posiciona um corredor em cada raia, supondo que ambos larguem juntos, o que deve acontecer para que o corredor da raia mais externa chegue junto com o da raia mais interna.

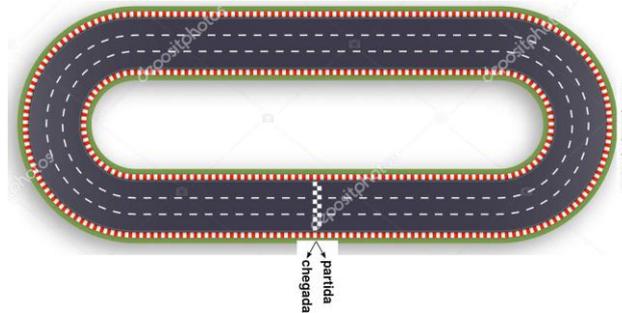


Figura 4

- ( ) Ele deve ter velocidade angular maior que o da raia 1.
- ( ) Ele deve ter velocidade angular menor que o da raia 1.
- ( ) Ambos devem ter a mesma velocidade angular.
- ( ) A alternativa 1 e 3.
- ( ) A alternativa 1 e 2.
- ( ) Todas as alternativas estão corretas.
- ( ) Nenhuma das alternativas estão corretas.

**Justifique sua resposta:**

---

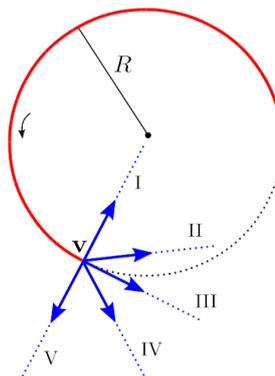


---



---

9 - Um objeto em Movimento Circular Uniforme, como mostra a figura abaixo, marque seta(azul) que melhor representa a força centrípeta desse objeto.



Figura

- ( ) I;
- ( ) II;
- ( ) V;
- ( ) II;
- ( ) IV.

**Justifique sua resposta:**

---



---

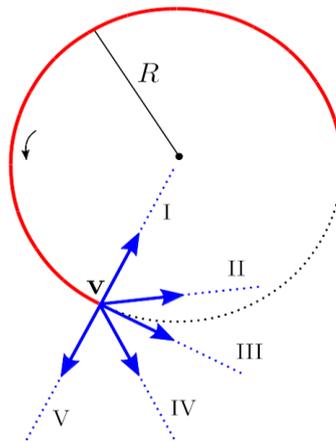


---



---

10 - Se a ação que promove essa trajetória do objeto for cessada (desligada) no instante indicado na figura pelo seu vetor velocidade (seta azul), qual será a direção do movimento do objeto a partir desse instante.



**Figura 1**

- ( ) II;
- ( ) I;
- ( ) III;
- ( ) V;
- ( ) IV.

**Justifique a sua resposta:**

---



---



---



---

## QUESTIONÁRIO 03 - AVALIAÇÃO METODOLÓGICA DAS ATIVIDADES

1 - A prática com simulador virtual contribuiu para seu aprendizado?

( ) Sim.

( ) Não.

2 - Enumere (em sua percepção) o quanto a prática com simulador contribuiu para seu aprendizado (nota crescente de 0 a 10 - sendo zero não contribuiu nada 10 contribui efetivamente com o aprendizado).

3 - A aula teórica contribuiu com seu aprendizado? (enumere com mesmos critérios da questão 2).

4 - Qual atividade contribuiu mais para o seu aprendizado em MCU.

( ) Teórica.

( ) Prática com os simuladores.

( ) As duas atividades - sendo uma complementar a outra (teórica + prática).

5 - Qual simulador contribuiu mais para o seu aprendizado.

( ) Engrenagens.

( ) Movimento da Terra.

( ) Os dois igualmente.

6 - Descreva as principais **dificuldades** no manuseio com o simulador.

---

---

---

---

7 - Descreva as **facilidades** que encontrou no manuseio do simulador.

---

---

---

---

**8** - O que você colocaria/adicionaria para melhorar a interatividade com o simulador

---

---

---

---

**9** - Sugestão(ões) para melhorar o desenvolvimento da aula.

---

---

---

---

## LINK PARA ACESSO ÀS SIMULAÇÕES DESTE PRODUTO

<https://drive.google.com/drive/folders/1axcdSn5af71lbvCBcbU1RxNDMMLJp3QZ?usp=sharing>