

Enunciado – Projeto Final

Neste projecto vocês podem escolher um dos problemas disponíveis e realizar o projecto lá definido. Os projectos têm uma nota máxima possível, definida a partir da sua dificuldade.

Alguns projectos têm de ser feitos individualmente, outros podem ser feitos em grupo de 2 pessoas.

O projecto deve ser entregue até às 23:55 do dia 16/06/2020, e vale 60% da nota final da disciplina.

A entrega deve ser feita via Moodle, e deve incluir um link para um repositório Github onde está o projecto.

Será avaliado no projecto a funcionalidade (40%), o modelo físico (40%) e a qualidade do código (20%).

O código do projeto vai ser sujeito a análise através de ferramentas de análise sintática para detecção de plágios e cópias. Se houver fortes indícios de cópia, a componente de projeto vai ter nota zero.

Nota: Só devem escolher UM dos projectos.

FAQ:

Pode usar-se código não feitos por nós?

Não é permitida o uso de qualquer tipo de código não criado pelo grupo.

As únicas exceções são:

- Numpy (apesar de eu não recomendar)
- Classes de vector 2d e 3d que tenham sido desenvolvidas em projectos anteriores.
- A classe de vectores do pygame (faz parte do Pygame, mas achei melhor clarificar)

Existe nota mínima no projecto?

A nota mínima no projecto é de 7 valores.

Eu tenho uma pergunta que não foi esclarecida no enunciado!

Qualquer dúvida, enviem um email para o professor da cadeira. Se for relevante para todos, será acrescentada aqui.

Projecto A (Individual)

Cotação máxima = 12 V

Desenvolver uma aplicação em Python que permita resolver dois problemas de Física automaticamente.

Inicialmente a aplicação deve mostrar um menu para escolher o tipo de problema que se quer resolver.

```
What problem do you want to solve?
1. Floatation
2. Springs
>1
Firing at 30 degrees, at 30 m/s.
Viscosity is 1, gravity is 9.81
>
```

Existem dois problemas para resolver:

- 1) Imersão num fluido: este deve mostrar como *output* a que profundidade um certo objecto ficaria quando imerso num fluido, mais precisamente, a que profundidade o centro do objecto ficaria.
A aplicação deve dar a opção de mudar os parametros da simulação e ir resimulando à medida que os parametros mudam.

Os parametros da simulação são:

- Gravidade
- Densidade do fluido
- Densidade do objecto
- Volume do objecto – A forma do objecto é um cubo perfeito, por isso vai dar jeito saber que $V_{cubo} = L^3$

Um exemplo de execução seria algo como no screenshot abaixo (apenas exemplificativo):

```
Object properties: Mass=10, Density=0.2, Volume=3
Fluid has a density of 1, gravity is 9.81
Object would float at 6.667994203456506 meters
>set mass 15
Object properties: Mass=15.0, Density=0.2, Volume=3
Fluid has a density of 1, gravity is 9.81
Object would float at 10.917604803047386 meters
>set gravity 4
Object properties: Mass=15.0, Density=0.2, Volume=3
Fluid has a density of 1, gravity is 4.0
Object would float at 6.215610580355024 meters
>set volume 2
Object properties: Mass=15.0, Density=0.2, Volume=3
Fluid has a density of 1, gravity is 4.0
Object would float at 3.951899529080257 meters
>
```

- 2) Molas: este deve mostrar como *output* a resolução de um problema de molas, em que um objecto com uma certa massa está preso por uma mola ao tecto através de uma mola com um coeficiente e um comprimento em repouso definido e queremos saber em qual o comprimento final da mola para o sistema ficar em equilíbrio.



A aplicação deve dar a opção de mudar os parametros da simulação e ir resimulando à medida que os parametros mudam.

Os parametros da simulação são:

- Gravidade
- Massa do objecto
- Comprimento da mola em repouso
- Constante da mola

Um exemplo de execução seria algo como no screenshot abaixo (apenas exemplificativo):

```
what problem do you want to solve?
1. Flootation
2. Springs
>2
Object mass is 1.5, gravity is 9.81.
Base spring length is 0.5, constant = 0.2
Spring would stretch to 0.8932473303297541 meters
>set gravity 4
Object mass is 1.5, gravity is 4.0.
Base spring length is 0.5, constant = 0.2
Spring would stretch to 1.554858753384999 meters
>set constant 1
Object mass is 1.5, gravity is 4.0.
Base spring length is 0.5, constant = 1.0
Spring would stretch to 1.7642558200831833 meters
>set length 0.2
Object mass is 1.5, gravity is 4.0.
Base spring length is 0.2, constant = 1.0
Spring would stretch to 1.2688959513021925 meters
```

Projecto B (Individual ou Grupo)

Cotação máxima = 16 V

Desenvolver uma aplicação em Python+Pygame que simule o comportamento físico de um projétil balístico. Através de comandos na consola, devemos poder mudar os parametros da simulação (como no projecto A), e a representação visual da trajectória também deve ser alterada.

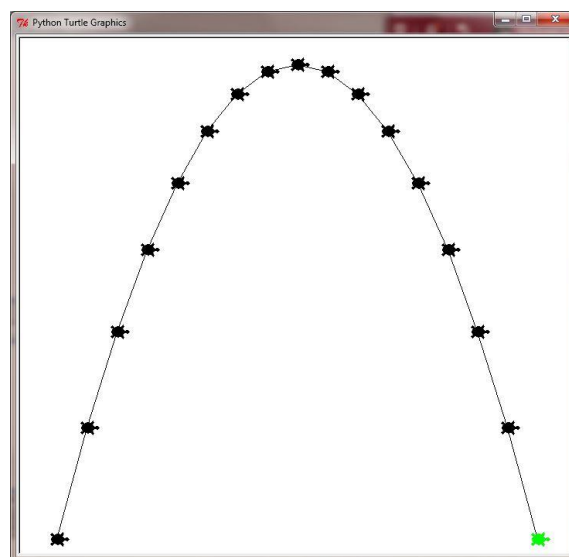
Os parametros da simulação são:

- Gravidade
- Ângulo de disparo
- Velocidade de disparo
- Constante de viscosidade do ar

Também deve ser possível acrescentar e remover forças externas (como um vector)

```
$ py test.py
Firing at 30 degrees, at 30 m/s.
Viscosity is 1, gravity is 9.81
>set speed 20
Firing at 30 degrees, at 20.0 m/s.
Viscosity is 1, gravity is 9.81
>set gravity 4
Firing at 30 degrees, at 20.0 m/s.
Viscosity is 1, gravity is 4.0
>addforce 100 20
Firing at 30 degrees, at 20.0 m/s.
Viscosity is 1, gravity is 4.0
>forces
0. F = (100.0, 20.0
Firing at 30 degrees, at 20.0 m/s.
Viscosity is 1, gravity is 4.0
>addforce 0 200
Firing at 30 degrees, at 20.0 m/s.
Viscosity is 1, gravity is 4.0
>forces
0. F = (100.0, 20.0
1. F = (0.0, 200.0
Firing at 30 degrees, at 20.0 m/s.
Viscosity is 1, gravity is 4.0
>removeforce 0
Firing at 30 degrees, at 20.0 m/s.
Viscosity is 1, gravity is 4.0
>forces
0. F = (0.0, 200.0
Firing at 30 degrees, at 20.0 m/s.
Viscosity is 1, gravity is 4.0
```

Consola da aplicação



Representação da trajectória balística

Projecto C (Individual ou Grupo)

Cotação máxima = 20 V

Desenvolver uma aplicação em Python+Pygame que simule o comportamento físico de uma nave espacial tipo a nave do “Asteroids”. Uma tecla deve rodar a nave para a esquerda, outra para a direita (dando aceleração angular – podem dar um valor *hard-coded* ao momento de inércia), uma tecla deve dar aceleração para onde a nave está orientada, outra deve dar aceleração no sentido contrário. O objectivo deste projecto não é fazer um movimento de nave giro para um jogo, mas sim um comportamento físico o mais correcto possível.

Esta nave deve ser representada visualmente utilizando o Pygame.

Também deve ser possível ter poços de gravidade (também eles visíveis) na simulação.

Exemplo visual (não representativo):

