

MOONHACK 2019

PYTHON PROJECT PORTUGUESE



Moonhack Python 2019


Introdução

Em julho de 1969, Neil Armstrong deu os primeiros passos na lua. Nós então celebramos o 50º aniversário desta conquista incrível, nós temos a chance de refletir na dificuldade e na audácia que foi cada parte das missões lunares. Neste projeto, nós iremos simular o pouso na lua criando um jogo da Cápsula Lunar em Python.



Passo 1: Preparando a Cápsula

✔ Lista de atividades

- Abra o projeto inicial em Python usando link <https://trinket.io/python/ac3342d0a8>. Aqui você irá ter três abas `main.py`, `terrain.py` e `landerClass.py`. Nós iremos escrever nosso código no arquivo `main.py`; as outras duas abas serão configuradas por você. Se você clicar no botão  Run, será gerado aleatoriamente uma superfície lunar e sua plataforma de pouso. Nós tentaremos pousar nessa plataforma.

Em sua aba `main.py`, você verá que algum código já foi adicionado lá. Este comando `import` irá buscar alguns códigos que iremos precisar no futuro. Nós iremos adicionar todo nosso código após esse comando de importação.

Primeiramente, nós precisamos criar uma cápsula. Nós poderemos fazer isso criando uma nova *instância* da nossa classe `landerClass`:

```
6 lander = landerClass()
```

(execute) seu programa. Você irá ver que a cápsula lunar aparece bem no topo da tela.

A seguir, nós queremos controlar a cápsula usando o teclado. Nós podemos fazer isso usando o comando `Screen().onkey()`, que irá registrar quando nós pressionarmos certas teclas. Vamos começar usando a tecla Seta para esquerda para virar para esquerda:

```
7 Screen().onkey(lander.left, "Left")
```

O comando `onkey()` recebe dois *argumentos*: o primeiro é o comando que você quer executar (que no caso é, comando `left` que já foi configurado como parte da classe `lander`), o segundo é a tecla que nós estaremos aguardando ser pressionada (a tecla `Seta para esquerda`) (execute) seu programa. Algo aconteceu quando você pressionou a tecla `Seta para esquerda`?


Nada aconteceu porque apesar de termos dito ao Python para responder a ação de pressionar a `Seta para esquerda`, nós na verdade não temos nenhum comando que capture a ação de apertar a tecla. Nós então precisamos usar o comando `listen()` para consertar isso:

```
8 Screen().listen()
```

(execute) seu programa. Quando você pressionar a tecla `Seta para esquerda`, seu módulo deve rotacionar para esquerda. **Note:** Se nada acontecer, certifique-se que você está olhando a janela de resultados, você pode fazer isso clicando na parte lateral direita da sua tela.

- Agora queremos adicionar código que irá virar para direita quando pressionarmos a tecla `Seta para direita`, e ligar e desligar os propulsores quando pressionarmos a tecla espaço:

```
7 Screen().onkey(lander.left, "Left")
8 Screen().onkey(lander.right, "Right")
9 Screen().onkey(lander.toggleThrust, "Space")
10 Screen().listen()
```

 (execute) seu programa. Você conseguirá ter controle da sua cápsula.

Passo 2: Fazendo a Cápsula cair

✔ Lista de atividades

- Queremos fazer que nossa cápsula seja afetada pela gravidade lunar. Nós iremos fazer isso usando um loop `while`. Um loop `while` recebe um valor `True` (verdadeiro) ou `False` (falso), ou outra expressão que resulte em `True` ou `False`. Adicione um loop que continuará para sempre:

```
11 while True:
```

- Agora nós precisaremos adicionar um código que na verdade fará nossa cápsula cair. Sem intervenção, a velocidade de queda irá aumentar de acordo com a aceleração da gravidade, então nós iremos configurar uma quantidade pré-determinada de `ACELERAÇÃO` à variável `yVel` (velocidade em y) usando o loop anterior:

```
11 while True:
12     lander.yVel += lander.ACCELERATION
```



Se nós executarmos o código agora, ela ainda não cairá porque nós não definimos que iremos usar `yVel` como parâmetro. Toda vez que rodarmos nosso loop, precisaremos dizer para subtrair o valor de `yVel` da posição atual (iremos subtrair porque ela está indo para baixo):

```
11 while True:
12     lander.yVel += lander.ACCELERATION
13     lander.sety(lander.ycor() - lander.yVel)
```



(execute) seu programa. A cápsula deve cair em direção ao solo e lentamente adicionando velocidade (o aumento é bem lento, lembre-se que nós estamos na Lua!)



Agora, nós estamos caindo direto na superfície da Lua. Queremos apenas continuar a se mover até alcançarmos a superfície da Lua. Para fazer isso iremos precisar mudar o loop `while` para usar a função `aboveLine()`. A função `aboveLine()` é definida no arquivo `terrain.py`. Ela pega o valor das coordenadas `x` e `y`, e retorna verdadeiro (true) se o ponto estiver acima da linha, e falso (false) e ele estiver abaixo dela. Modifique seu código com o trecho a seguir:

```
11 while terrain.aboveLine(lander.xcor(),lander.ycor()):
12     lander.yVel += lander.ACCELERATION
13     lander.sety(lander.ycor() - lander.yVel)
```



(execute) seu programa. Ela agora irá cair até alcançar a superfície, e então irá parar.

Passo 3: Controlando a Cápsula

✔ Lista de atividades



Até o momento, nós podemos ligar os propulsores, mas eles não fazem nada na realidade. Nós iremos adicionar um código em nosso loop while que irá reduzir a velocidade dos propulsores quando forem ativados:

```
13     lander.sety(lander.ycor() - lander.yVel)
14     if lander.thrusters:
15         lander.yVel -= lander.THRUST
```



(execute) seu programa. Quando você ligar os propulsores, isso irá desacelerar sua cápsula. Se eles forem ativados por muito tempo, sua cápsula irá subir.



Nós agora estamos usando nossos propulsores com sucesso para desacelerar nossa cápsula, mas o que acontece se virarmos nossa cápsula com as teclas de seta? Nós estaremos apenas indo para cima e para baixo, mas queremos ir para as laterais também. Precisamos dividir nossos propulsores em componente cima-e-baixo (eixo y), componente laterais (eixo x), mas como podemos fazer isso funcionar?

TRIGONOMETRIA AO RESGATE! Atualize seu código com o trecho abaixo:

```
14     if lander.thrusters:
15         heading = radians(lander.heading())
16         lander.yVel -= lander.THRUST*sin(heading)
17         lander.xVel += lander.THRUST*cos(heading)
```



(execute) seu programa. Você deve agora ter uma cápsula lunar totalmente funcional que se move tanto no eixo x quanto no eixo y.

Passo 4: Pousando com segurança... ou não

✔ Lista de atividades



Até agora, quando terminarmos nosso loop, o programa apenas para. Queremos checar se (`if`) nossa cápsula pousou com sucesso na plataforma. Adicionando o código a seguir irá fazer com que tenhamos certeza que ela terminou na plataforma:

```
19     if terrain.onPad(lander.xcor()):
20         lander.landed()
```



Se não pousarmos na plataforma, nós temos batemos. Adicione uma cláusula senão (`else`):

```
19     if terrain.onPad(lander.xcor()):
20         lander.landed()
21     else:
22         lander.crash()
```



(execute) seu programa. Se você terminou próximo à plataforma, você pousou com sucesso, do contrário você irá bater com sua cápsula na plataforma e irá partir em pedaços.



Isso parece muito bom, mas atualmente ainda teremos um pouso bem-sucedido, mesmo se estivermos chegando muito rápido. Vamos adicionar uma condição à nossa declaração if que nos diz que só pousamos com sucesso se chegarmos a menos de 2 unidades de velocidade:

```
19 if terrain.onPad(lander.xcor()) and lander.yVel < 2:
```

 Run (execute) seu programa.

Parabéns! Você terminou seu projeto! Quer mais? Tente fazer os desafios a seguir e experimente outros projetos do Moonhack!

Desafio: Mais condições de derrota

Atualmente, a cápsula irá bater se você descer rápido demais, mas existem outras situações que poderão fazer você bater. Você poderia checar se a velocidade X e o rumo da cápsula (a direção em que ele está apontando) estão dentro dos limites razoáveis? Dica: A função `abs()` pega um número e retorna para você o valor absoluto dele. Ela pode ser bem útil porque a velocidade X pode ser tanto negativa (indo para esquerda), quanto positiva (indo para direita).

Desafio: Cápsula marciana

NASA precisa de você! Pesquise a força gravitacional em Marte em relação a da Lua e mude a `ACELERAÇÃO` para que sua cápsula lunar reflita isso.