MOONHACK 2019

PYTHON PROJEKT

CROATIAN

Uvod

20. srpnja 1969. godine Neil Armstrong napravio je prve korake na Mjesecu. Proslavom 50. obljetnice ovog nevjerojatnog postignuća imamo šansu razmišljati o poteškoćama i odvažnosti svakog dijela Mjesečevih misija. U ovome projektu, simulirat ćemo slijetanje na Mjesec stvaranjem Lunar Lander igre u Pythonu.

Upute

Lander će padati, a mi ćemo kontrolirati kut slijetanja strelicama i potisnik letjelice s tipkom za razmak.

Što će vam trebati:

Hardware

Kompjuter s internetskom vezom

Software

Trinket online editor

KORAK 1: PRIPREMANJE LANDER-a

Zadatci

- Otvorite Python projekt na adresi: https://trinket.io/python/ac3342d0a8.
  Ovdje ćete vidjeti tri kartice: main.py, terrain.py i landerClass.py. Mi ćemo programirati u main.py kartici; druge dvije kartice su postavljene za vas. Ako pritisnete tipku ▶ Run nasumično će generirati mjesečevu površinu i postaviti postolje za slijetanje. Pokušat ćemo sletjeti na ovu postaju za slijetanje.
- U vašoj main.py kartici, vidjet ćete da je već dodan neki kod. To su operatori uvoza (import na engleskom) koji donose kod koji ćemo kasnije trebati. Mi ćemo dodati sav naš kod nakon tih operatera uvoza.
- Prvo, trebamo kreirati Lander. Možemo to napravit stvaranjem nove *instance* landerClassa:

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

  Pokrenite svoj program s ▶ Run . Vidjet ćete da se Lander modul pojavljuje u samom vrh zaslona.

- Sljedeće, želimo kontrolirati naš Lander pomoću tipki. To možemo napraviti tako da koristimo Screen() .onkey() naredbu, koja će se registrirati kada pritisnete određene tipke.
  Počnimo sa strelicom lijevo kako bi naš Lander skrenuo lijevo:

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

  Naredba onkey () ima dva argumenta: prvi je naredba koju želimo izvršiti (u ovom slučaju, left naredba koja dio lander klase), druga je ključ koji tražimo (lijevu tipku – left na engleskom).
  Pokrenite svoj program s ▶ Run . Događa li se nešto kada stisnete tipku lijevo?

- Ništa se nije dogodilo jer smo rekli Python-u da odgovori na lijevu tipku, ali mu nismo rekli da sluša. Moramo koristiti listen() naredbe za popravak:

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

  Pokrenite svoj program s ▶ Run . Kada pritisnete lijevu tipku, vaš će se uređaj okretati nalijevo.

Napomena: Ako se ništa ne dogodi, provjerite jeste li fokusirani na prozor s rezultatima, to možete učiniti klikom na desnu stranu zaslona.

- Sada želimo dodati kod koji će skrenuti desno kada pritisnete desnu tipku i uključite i isključite potisnike kada pritisnete razmaknicu:

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

Pokrenite svoj program s ▶ Run . Sada biste trebali kontrolirati svoj Lander.

KORAK 2: SPUŠTANJE LANDERA

Zadatci

- Želimo da naš Lander padne zbog lunarne gravitacije. Napravit ćemo to uz pomoć while petlje. Petlja while uzima True ili False vrijednost ili drugi izraz koji ocjenjuje True ili False vrijednosti. Dodajte petlju koja će se nastaviti zauvijek:

```
11  while True:
```

- Sljedeće, moramo dodati kod koji će zapravo dovesti do pada Landera. Bez intervencije, brzina kojom pada Lander će se povećati zbog učinka ubrzanja gravitacije, pa ćemo dodati fiksno ubrzanje ACCELERATION iznosu yVel (y ubrrzanje - y velocity na engleskom) iz predhodne petlje:

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

- Ako sada pokrenemo naš kod, on još uvijek neće pasti, jer mu još nismo rekli što učiniti s yVel. Svaki put kad prođemo kroz našu petlju moramo oduzeti yVel s naše trenutne pozicije (oduzimamo jer idemo dolje):

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

Pokrenite svoj program s ▶ Run . Sada bi trebao pasti dok ne dosegne površinu, a zatim prestane.

KORAK 3: KONTROLIRANJE LANDER-a

Zadatci

- Trenutno možemo uključiti potisnike, ali oni zapravo ništa ne rade. Kod ćemo dodati u našu while petlju, brzinu ćemo smanjiti tako da smanjimo yVel ako su potisnici aktivirani:

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

Pokrenite svoj program s ▶ Run . Kada uključite svoje potisnike Lander će usporiti. Ako su dovoljno dugo uključeni, vaš će se Lander početi dizati.

- Sada uspješno koristimo naše potisnike kako bismo usporili naš Lander, ali što se događa ako okrenemo naš Lander pomoću strelica? Još uvijek idemo samo gore-dolje, ali želimo ići zajedno. Moramo podijeliti naš potisak u komponentu gore-dolje (y-os) i komponentu sa strane (x-os), ali kako možemo to riješiti? TRIGONOMETRIJA ZA POMOĆ! Ažurirajte svoj kod na sljedeće:

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

Pokrenite svoj program s ▶ Run . Vaš lander će i dalje ići samo gore i dolje! O ne!

- Do sada smo definirali xVel (x brzinu), ali zapravo nismo to koristili za preusmjeravanje našeg Landera. Dodajmo redak koda koji će koristiti naš xVel za premještanje našeg Lander-a duž osi x:

```
13      lander.sety(lander.ycor() - lander.yVel)
14      lander.setx(lander.xcor() + lander.xVel)
15      if lander.thrusters:
```

Pokrenite svoj program s ▶ Run . Sada bi trebali imati potpuno funkcionalan lunarni Lander koji se pomiče na x i y osi.

KORAK 4: SIGURNO SLIJETANJE…. ili ne

Zadatci

- U ovom trenutku, kada završimo našu petlju, naš program se zaustavlja. Mi želimo provjerite ako (if na engleskom) je naš Lander uspješno sletio na naš blok. Dodajte sljedeće u kod kako biste bili sigurni da je naš uređaj za slanje završio na bloku:

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

- Ako ne sletimo na podlogu, srušili smo se. Dodajte else izraz:

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

Pokrenite svoj program s ▶ Run . Ako završite u blizini pristaništa za slijetanje, sletjet ćete uspješno, inače ćete se srušiti i vaš modul Landera će se raspasti.

- To izgleda prilično dobro, ali trenutno ćemo imati uspješno slijetanje čak i ako dolazimo jako brzo. Dodajmo uvjet u našu if izjavu da kaže da smo uspješno sletjeli samo ako sletimo brzinom manjom od 2:

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

Pokrenite svoj program s ▶ Run .

IZAZOV: Više uvjeta sudara

Trenutno će se Lander srušiti ako prebrzo ide prema dolje, ali tamo su druge situacije u kojima se može srušiti. Možete li provjeriti jesu li x brzinu i smjer Landera u razumnim granicama?
Savjet: funkcija abs () uzima broj i daje vam apsolutnu vrijednost tog broja. To može biti korisno jer x brzina može biti ili negativno (lijevo) ili pozitivno (desno).

IZAZOV: Mars Lander

NASA te treba! Istražiti gravitacijsku silu na Marsu u odnosu na Mjesec i promijenite ACCELERATION vašeg Mjesečevog modula kako bi odgovarala uvjetima na Marsu.